



POLITECNICO DI MILANO

DIPARTIMENTO DI INGEGNERIA CIVILE E AMBIENTALE
DOTTORATO IN INGEGNERIA AMBIENTALE E DELLE INFRASTRUTTURE

PROGETTAZIONE E REALIZZAZIONE DEL
SISTEMA INFORMATIVO TERRITORIALE
“GEOGRAFIE DELL’ITALIA MEDIEVALE (XIII – XV SEC.)”

Tesi di Dottorato di:

Cynthia Zambrano

Relatore e tutor:

Prof.ssa Federica Migliaccio

Correlatore:

Dott.ssa Daniela Carrion

Coordinatore:

Prof. Fernando Sansò

2013 – XXV CICLO

INDICE

INTRODUZIONE	7
Inquadramento del progetto	7
Stato dell'arte: applicazioni in campo storico e archeologico	9
Sommario del lavoro di Tesi.....	14
1 PROGETTAZIONE DEL DATABASE GEOREFERENZIATO DEI DATI STORICI.....	17
1.1 I dati del Quaternus declaracionum	18
1.2 Perché creare un database per i dati storici?	22
1.3 Metodologia di progettazione di un sistema informativo e di una base di dati	24
1.4 Studio di fattibilità del geoservizio	28
1.5 Analisi dei requisiti del database di dati storici	30
1.6 Progettazione concettuale del database storico	32
1.7 Progettazione logica del database storico	39
1.8 Scelta del DBMS con supporto di gestione ai dati spaziali: Postgresql e PostGIS	43
1.9 Progettazione fisica del database storico	45
1.10 Popolamento del database storico	47
2 .PROGETTAZIONE DEL GEOSERVIZIO “GEOGRAFIE DELL’ITALIA MEDIEVALE (XIII – XV SEC.)”.....	51
2.1 Un’applicazione web per la consultazione della base di dati storici.....	52
2.2 Architettura di un’applicazione web	52

2.3	Scelte di progettazione del “lato server” del geoservizio	56
2.4	Progettazione del “lato client” (interfaccia utente del geoservizio).....	58
2.4.1	Fase 1 – definizione degli obiettivi della applicazione web	60
2.4.2	Fase 2 – definizione delle caratteristiche generali dell’interfaccia utente.	60
2.4.3	Fase 3 – definizione delle tecnologie per la realizzazione dell’interfaccia web	63
2.5	Sintesi delle tecnologie scelte per la realizzazione del geoservizio	73
3	REALIZZAZIONE E PUBBLICAZIONE DEL GEOSERVIZIO DEI DATI STORICI.....	75
3.1	L’applicazione web “Geografie dell’Italia Medievale (XIII – XV secolo)”.	76
3.1.1	Navigazione del geoservizio: modalità “WebGIS”	76
3.1.2	Navigazione del geoservizio: modalità “Mostra tabella”	86
3.2	Implementazione delle funzionalità di ricerca nella modalità “Mostra tabella”.....	89
3.2.1	Funzionalità di ricerca semplice	90
3.2.2	Funzionalità di ricerca avanzata	92
3.3	Procedura di autenticazione e di caricamento dell’applicazione web.....	96
3.4	Validazione del prototipo del geoservizio per i dati storici	101
4	CONCLUSIONI E SVILUPPI FUTURI.....	103
4.1	Conclusioni	103
4.2	Sviluppi futuri	105
	INDICE DELLE FIGURE.....	107
	INDICE DELLE TABELLE	109

LISTA DEGLI ACRONIMI.....	111
BIBLIOGRAFIA	113
SITOGRAFIA.....	119
APPENDICE A	121
APPENDICE B.....	131

INTRODUZIONE

INQUADRAMENTO DEL PROGETTO

L'utilizzo dei Sistemi Informativi Geografici (*Geographical Information Systems*, GIS) ha conosciuto negli ultimi venti anni uno sviluppo molto rapido; in senso generale, si può dire che questo crescente interesse dipenda dal fatto che molte delle informazioni comunemente disponibili oggi siano georeferenziabili, cioè collegabili ad una posizione nello spazio identificabile tramite coordinate geografiche. Una tale abbondanza di informazioni comporta la crescente esigenza di nuovi strumenti per poterle gestire, organizzare e tradurre in dati immediatamente leggibili e rappresentabili cartograficamente. I software GIS consentono appunto questo e, in virtù di ciò, sono diventati uno strumento di grande utilità per chiunque debba compiere analisi sul territorio. In tal senso, i GIS sono strettamente ed indissolubilmente legati al supporto cartografico e in essi confluiscono ed interagiscono molti aspetti riguardanti la gestione di basi di dati, le tecniche di rappresentazione grafica e le metodologie di indagine statistica. I vantaggi dell'adozione di tali strumenti prescindono dall'ambito di applicazione e consentono nuove opportunità di ricerca sia nelle discipline tecnico-analitiche che in quelle umanistiche.

Nell'ambito della Ricerca Storica, ad esempio, è disponibile una grande quantità di dati, raccolti da fonti eterogenee e con una forte connotazione territoriale: gli strumenti GIS ben si prestano quindi ad analisi di dati storici provenienti dallo studio delle geografie politiche. Questo tipo di indagine può beneficiare di un supporto cartografico che deve essere strutturato in forma digitale per poter essere facilmente gestito e aggiornato.

Il presente lavoro di Tesi si inquadra all'interno del Progetto di Ricerca "Organizzazione del territorio, occupazione del suolo e percezione dello spazio nel Mezzogiorno medievale (secoli XIII – XV). Sistemi informativi per una nuova cartografia storica" (PRIN¹ 2009) il cui principale obiettivo consiste nell'approfondimento del nesso "potere – territorio" nell'area del Mezzogiorno mediante lo studio dei linguaggi politici e delle carte antiche. A questo progetto partecipano quattro unità di ricerca, di cui tre di storia medievale (l'Università Federico II di Napoli, capofila del progetto, l'Università del Salento e l'Università di Bologna). La quarta unità, costituita dall'Unità di Ricerca del Politecnico di Milano, si è posta come obiettivo offrire alle Unità degli Storici ulteriori strumenti di studio e analisi dei dati da affiancare a quelli già esistenti, al fine di arrivare a nuove interpretazioni dei dati ad oggi disponibili.

Per rendere omogeneo l'utilizzo dei dati storici in ambiente GIS e di consentirne il controllo, il confronto, la correlazione, la sovrapposizione e la condivisione, era di fondamentale importanza che le collezioni di dati delle diverse Unità di Ricerca fossero organizzate secondo schemi uniformi in un archivio comune, senza equivoci sul formato e sul significato dei dati.

Un importante lavoro sull'uniformazione dei dati era già stato svolto durante il progetto "*Geografie e linguaggi politici alla fine del Medioevo*" (PRIN 2006), in cui l'Unità dell'Università del Salento, in collaborazione con l'Unità del Politecnico di Milano, aveva delineato una struttura tabellare capace di identificare la geografia insediativa, amministrativa ed economica del territorio. Successivamente gli Storici dell'Università del Salento avevano utilizzato questo modello per archiviare in fogli Excel i dati provenienti dal documento storico *Quaternus Declaracionum* di Francesco Agello, dopo averli opportunamente georeferenziati mediante le coordinate piane di ogni località storica e classificati tramite un coefficiente numerico chiamato "grado di affidabilità". Nell'ambito degli obiettivi del lavoro svolto dal gruppo di

¹ Progetto di Ricerca di Interesse Nazionale.

ricerca del Politecnico, una delle finalità di questa Tesi è stata progettare e realizzare un database il cui modello dati fosse riconducibile alla struttura tabellare di partenza: questa nuova organizzazione delle informazioni consente di svolgere interrogazioni incrociando in modo automatico dati di diversa natura e provenienza, sfruttando tutte le potenzialità di questo tipo di sistemi.

Un'altra finalità che ci era prefissata è stata quella di rendere possibile la condivisione delle informazioni fra le diverse Unità di Ricerca mediante la realizzazione di un geoservizio web, chiamato "*Geografie dell'Italia Medievale (XIII – XV secolo)*", che consentisse la navigazione e l'interrogazione dei dati storici in una duplice modalità: mediante un supporto cartografico con strumenti GIS e sotto forma di viste e tabelle interattive mediante un'interfaccia appositamente studiata. L'applicazione è stata ideata e sviluppata nell'ambito di questa Tesi cercando di renderne più semplice ed intuitivo possibile l'utilizzo da parte degli utenti storici, astraendo dagli aspetti più tecnici del funzionamento di una base di dati. Il geoservizio, attualmente disponibile solo ai partecipanti al progetto, sarà reso pubblico alle fine delle operazioni di collaudo e validazione.

STATO DELL'ARTE: APPLICAZIONI IN CAMPO STORICO E ARCHEOLOGICO

L'impiego di strumenti GIS in campo storico è una tendenza piuttosto recente: solo verso la seconda metà degli anni '90 è stata coniata la definizione HGIS (*Historical Geographic Information System*). La tardiva adozione di queste soluzioni è da ricercarsi in parte nella difficoltà di modificare metodologie di ricerca storiche ampiamente consolidate, ma anche nella difficoltà tecnica di integrare efficacemente dati storici, spesso molto eterogenei nel formato e nella provenienza. Per questi motivi, la realizzazione di un GIS per la navigazione e consultazione di dati storici è

un'operazione complessa (Grosso et al., 2009) che richiede la sinergia di personale con competenze diverse (storici, cartografi e informatici).

In Italia si possono trovare alcuni esempi di applicazioni di metodologie GIS all'analisi della cartografia storica (Balletti, 1998; Balletti, 2000; Balletti et al., 2000; Baiocchi e Lelo, 2002; Barzaghi e Carrion, 2011); si tratta però di lavori più rivolti all'analisi di tipo storico-urbanistico ed alla trattazione delle metodologie di correlazione e adeguamento delle cartografie storiche a quelle odierne. Non ci sono, pertanto, molti esempi di GIS che abbiano realmente come oggetto il dato storico in quanto tale o la sua strutturazione in un'adeguata base di dati (Ardisson e Rinaudo, 2005).

In ambito internazionale hanno gettato le basi per questo tipo di indagine i lavori di (Berman, 2005) e quelli di (Gregory et al. 2002; Gregory, 2003; Gregory e Ell., 2008). Berman mise in evidenza la difficoltà di confrontare indicazioni di luoghi, distanze e aree di appartenenza tratte da fonti storiche con le nostre informazioni moderne, problema aggravato se i documenti storici si contraddicono perché rispecchiano la visione del territorio in esame di due giurisdizioni in contesa. Gregory sottolineò le potenzialità degli strumenti GIS nella ricerca storica come, ad esempio, la possibilità di riconoscere *pattern* spaziali in un singolo set di dati oppure la consultazione di più set di dati in funzione della posizione geografica.

I dati storici possono essere compresi essenzialmente in tre categorie (Paselli, 2009):

- **cartografie storiche:** la documentazione è rappresentata da piante, planimetrie, ma anche antiche litografie, mappe catastali, incisioni e affreschi, ecc; in questi casi la costruzione di un Sistema Informativo Territoriale passa dalla georeferenziazione della mappa storica mediante la vettorializzazione delle unità minime significative (isolati, particelle catastali, ecc) a cui sono integrate le informazioni storiche associate (abitanti, valore dell'edificio, altezza, ecc);

- **reperti archeologici:** resti di civiltà antiche e non, che permettono una ricostruzione storica dell’ambiente, del paesaggio e della città; i primi GIS per la ricerca storica rientrano in questa categoria: infatti, per poter costruire una mappa o ricreare un modello virtuale del reperto archeologico è necessario eseguire il rilievo topografico dell’area d’interesse, che costituisce la base georeferenziata su cui viene costruito lo strumento GIS;
- **dati storici:** possono essere dati documentali come manoscritti, registri, atti notarili, catasti medioevali, epigrafi, testi a stampa, ecc; in questi casi la georeferenziazione può avvenire, per esempio, mediante l’informazione relativa all’ubicazione riportata nella fonte storica (Berti, 2005) e il più delle volte può risultare molto difficoltosa se l’indicazione della localizzazione è imprecisa (si ha solo l’indicazione di una circoscrizione o distretto di appartenenza), ambigua (esistono più località con lo stesso nome) o se non si trova corrispondenza con nessuna località attuale.

Tra gli esempi che rientrano nella categoria dei GIS che comprendono mappe storiche possiamo ricordare il WebGIS del Catasto Gregoriano², realizzato dal Dipartimento di Studi Urbani dell’Università “Roma Tre” insieme alla Fondazione Cariplo, all’Archivio di Stato di Roma e alla Soprintendenza Comunale di Roma. Il WebGIS permette la navigazione dei dati vettoriali e raster del Catasto Pontificio di Roma e dei relativi brogliardi, ovvero i registri dei proprietari, negli anni tra il 1818 e il 1824 e agevola l’interrogazione degli attributi associati alle particelle catastali.

A Parigi, il Laboratorio di Studi Medievali Occidentali (Lamop) ha portato a termine con grande rigore scientifico il progetto ALPAGE³, un lavoro di georeferenziazione di 910 mappe catastali prodotte da Vassoit tra il 1810 e il 1836 al fine di ricostruire ed analizzare gli spazi urbani della Parigi ottocentesca (Noizet et al., 2008; Noizet e Grosso, 2011). Al termine del lavoro, agli elementi geometrici e geografici sono state

² URL del progetto (gennaio 2013) <http://www.dipsuwebgis.uniroma3.it/index.html>.

³ URL del progetto (gennaio 2013) <http://alpage.tge-adonis.fr/index.php/fr/>.

integrate tutte le informazioni di tipo storico ed è stato realizzato un WebGIS per diffondere le informazioni tra scuole e università.

Per quanto riguarda i GIS archeologici, un importante progetto è Fasti Online⁴, un progetto che raccoglie gli scavi archeologici condotti dal 2000 in tredici Paesi con particolare attenzione all'Italia, Bulgaria, Romania, Macedonia, Malta e Albania. Questa iniziativa è l'evoluzione della rivista Fasti Archaeologici, pubblicata tra il 1946 e il 1987 dall'Associazione Internazionale di Archeologia Classica (AIAC) contenente notizie brevi molto utili sugli scavi riguardanti l'Impero romano; i costi crescenti e i ritardi della pubblicazione della rivista hanno portato a cercare un modo innovativo per registrare e diffondere i risultati delle indagini archeologiche recenti: i Fasti Online sono il risultato di questo sforzo. Il progetto è attualmente finanziato dal Ministero dei Beni e le Attività Culturali, la Society for the Promotion of Roman Studies e la Classical Association.

Per quanto riguarda le applicazioni GIS relative ai dati storici, un interessante progetto è il *Great Britain Historical Geographical Information System* (GBHGIS) condotto dal Dipartimento di Geografia dell'Università di Portsmouth. Originariamente l'obiettivo del lavoro era quello di archiviare l'enorme mole di informazioni che derivavano da schede di censimento, gazzette storiche, contratti di matrimonio, decessi, racconti di viaggi, statistiche elettorali, elenchi fiscali in un database al fine di ottimizzarne la consultazione. Successivamente, a partire dal 2001, si è cercato di integrare funzionalità tipiche dei GIS con l'introduzione del sistema di gestione dati Oracle Spatial, anche se molte informazioni contenute nel database non avevano un riferimento territoriale. La soluzione adottata non è propriamente un GIS, bensì un sistema flessibile capace di collegare tra di loro, tramite delle regole, le unità territoriali (cappelle, distretti locali, distretto amministrativi, nazioni). Dalla pagina web che ospita il progetto⁵ è possibile eseguire ricerche sui luoghi di interesse inserendo il nome di una località oppure un'estensione territoriale: i risultati sono

⁴ URL del progetto (gennaio 2013) <http://www.fastionline.org/index.php?view=home>.

⁵ URL del progetto (gennaio 2013) <http://www.visionofbritain.org.uk/>.

ordinati in base alle unità territoriali (dalla più piccola unità, la parrocchia, alla più grande, la nazione).

Un'interessante applicazione degli strumenti GIS che ha consentito la ricostruzione parziale degli scenari della guerra civile romana tra Giulio Cesare e Pompeo è stato portato avanti dall'Universitat de Barcelona (Rubio-Campillo, 2009). A partire dai racconti del *De Bello Civili* di Cesare sono stati raccolti tutti i parametri dello scenario (vicinanza a fonti d'acqua, posizione degli accampamenti, ecc) in un GIS e, al fine di ipotizzare le decisioni prese dai due comandanti così come i luoghi dello scontro, tutte le informazioni sono state processate attraverso la teoria matematica dei giochi, una teoria utile per descrivere il comportamento umano in termini matematici. In ultimo, un progetto estremamente significativo è stato realizzato dall'Università di Harvard e dall'Università di Fudan: China HGIS⁶ è un database geografico che si distingue per l'utilizzo delle tecnologie GIS con particolare attenzione all'ordinamento delle informazioni secondo una sequenza temporale. Il progetto, iniziato nel 2001 e pubblicato nella sua versione iniziale nel 2002, raccoglie dati storici di carattere prevalentemente amministrativo del periodo dinastico cinese dal 221 AC al 1911 DC (Henderson e Berman, 2003). Utilizzando il software Time Map sono state realizzate animazioni che hanno evidenziato l'evoluzione temporale dei dati nei vari *layer*: particolarmente suggestiva è la rappresentazione degli spostamenti di Ouyi Zhixu, un maestro e monaco buddista vissuto nell'ultima dinastia Ming, ricostruiti sulla base di informazioni storiche e biografiche⁷. La correlazione del dato temporale e dell'informazione spaziale è estremamente suggestiva se si considerano le potenzialità in ambito scientifico e storico rese possibili da analisi non solo di tipo geografico/spaziale ma anche di tipo temporale.

Il lavoro svolto in questa Tesi si inquadra tra le applicazioni GIS relative ai dati storici; in particolare le informazioni oggetto di questo studio sono state estratte dal

⁶ URL del progetto (gennaio 2013) <http://www.fas.harvard.edu/>.

⁷ URL di documentazione (gennaio 2013)

http://www.fas.harvard.edu/~chgis/work/docs/papers/CGA_Wkshp2009_Lex_9apr09.pdf

Quaternus declaracionum di Francesco Agello, un documento di riepilogo degli ingressi e delle uscite tributarie dello stato orsiniano alla fine del Medioevo. Auspicabilmente la banca dati continuerà a crescere con nuove integrazioni, rendendo questo strumento sempre più rilevante nell'ambito della Ricerca storica Medievale: questo processo in parte è già stato avviato, infatti sono già stati integrati nel database storico alcuni dati tributari relativi al Ducato di Milano.

SOMMARIO DEL LAVORO DI TESI

Il presente lavoro di Tesi si articola in tre sezioni principali:

- nel primo Capitolo “PROGETTAZIONE DEL DATABASE GEOREFERENZIATO DEI DATI STORICI” viene descritta la tipologia e l’origine dei dati oggetto di questo lavoro, si illustra il percorso seguito durante la definizione del database georeferenziato nelle diverse fasi progettuali (concettuale, logica e fisica) e, infine, vengono descritte le procedure di creazione e popolamento della base di dati;
- nel secondo Capitolo “PROGETTAZIONE DEL GEOSERVIZIO GEOGRAFIE DELL’ITALIA MEDIEVALE (XIII – XV SEC.)” si descrive la fase di ricerca e studio preliminare, il percorso di ideazione dell’interfaccia grafica utente per poi illustrare l’architettura *client-server* finalizzata alla realizzazione dell’applicazione web, con particolare attenzione all’individuazione delle più opportune soluzioni tecnologiche per la realizzazione del geoservizio;
- nel terzo Capitolo “REALIZZAZIONE E PUBBLICAZIONE DEL GEOSERVIZIO” sono approfonditi i passaggi più significativi della realizzazione tecnica dell’applicazione web nelle sue componenti architettoniche: l’interfaccia utente (lato client) e gli *script* (lato server) di caricamento e interrogazione dei dati nell’applicazione; in particolare sono

descritte le diverse modalità con cui è possibile usufruire del servizio, la procedura di autenticazione e i principi di funzionamento degli strumenti implementati.

1 PROGETTAZIONE DEL DATABASE GEOREFERENZIATO DEI DATI STORICI

Nel presente capitolo sarà illustrata la procedura di progettazione del database georeferenziato dei dati storici provenienti da due fascicoli di Francesco di Agello del *Quaternus Declaracionum*, in cui sono registrati gli ingressi e uscite esattoriali dello Stato orsiniano negli anni 1448 e 1460.

La progettazione della base di dati storici ha seguito la metodologia standard che si articola in tre fasi da seguire in cascata: Progettazione Concettuale, Progettazione Logica, Progettazione Fisica. In questa fase si è sfruttata la struttura tabellare di archiviazione dei dati che l'Unità di Ricerca di Lecce e l'Unità di Ricerca del Politecnico di Milano avevano definito nell'attività di ricerca condotta per il progetto “*Geografia e linguaggi politici alla fine del Medioevo*” (PRIN 2006) per delineare la geografia insediativa, amministrativa ed economica del territorio. Le informazioni erano state archiviate in tabelle con struttura comune che riportavano per ogni località storica le coordinate piane, ed era stato assegnato un coefficiente numerico chiamato “grado di affidabilità” che variava da 1 a 5 all'aumentare del livello di precisione.

In questo capitolo viene inoltre descritto lo studio di fattibilità in cui sono state considerate una suite di tecnologie open source e una suite di tecnologie proprietarie. Dall'analisi è emerso come più conveniente l'utilizzo di una *suite* di tecnologie open source; questa scelta evita di gravare sul budget complessivo del progetto e consente di creare soluzioni scalabili e personalizzabili.

Infine viene descritta la procedura di creazione e il popolamento del database mediante *script* SQL sviluppati appositamente per questo lavoro .

1.1 I DATI DEL QUATERNUS DECLARACIONUM

Nell'ambito del Progetto “*Geografia e linguaggi politici alla fine del Medioevo*” (PRIN 2006) si è cercato di ricostruire, almeno parzialmente, la consistenza dello stato orsiniano, utilizzando per lo più documenti storici di tipo fiscale. In particolare, nella prima fase del lavoro è stato considerato un documento di riepilogo di tutte le registrazioni di ingressi e uscite del dominio orsiniano: il *Quaternus declaracionum*. Tale documento veniva redatto dai maestri razionali, i revisori del Principe Orsini, i quali erano tenuti ad esaminare l'operato degli amministratori nelle curie locali. Percettori, credenzieri, doganieri, fondichieri, erari locali e generali, tesorieri e capitani, alla fine del proprio mandato, erano tenuti a sottoporre al vaglio dei maestri i registri da loro compilati, al fine di documentare l'attività svolta (Bulgarelli, 1993; Bulgarelli 1996; Bulgarelli, 1997; Comba e Naso, 1994; Ginatempo e Sandri, 1990; Mainoni, 2001). Dopo aver collazionato le carte e verificato la corrispondenza delle informazioni riportate dai documenti, i razionali annotavano sinteticamente su un fascicolo le operazioni eseguite da ogni singolo amministratore, ripartendole in due sezioni, *introitus* (ingressi) ed *exitus* (uscite), e calcolando alla fine le somme di denaro e i beni spettanti alla curia. Ciascun fascicolo, che poteva riportare i rendiconti di uno o più funzionari operanti in una stessa località, veniva poi rilegato, insieme ad altri fascicoli, in un volume che costituiva il *Quaternus declaracionum* relativo ad un determinato anno. Il computo del tempo nel *Quaternus Declaracionum* è di tipo indizionale⁸, nei documenti storici viene indicato l'anno all'interno di un ciclo di quindici anni numerati progressivamente da uno all'ultimo anno del ciclo a conclusione del quale il conto riprende da uno, ossia il primo anno del nuovo ciclo. I due *Quaterni declaracionum* considerati in questo lavoro sono stati compilati negli anni della settima e ottava indizione dal razionale Francesco de Agello, sono rilegati

⁸ L'indizione è un computo del tempo che non ha alcuna relazione col movimento degli astri. Essa è parte della Data nei documenti della tarda antichità, medievali e, in alcuni luoghi, moderni: indica l'anno all'interno di un ciclo di anni numerati progressivamente da 1 all'ultimo anno del ciclo a conclusione del quale il conto riprende da 1, ossia il primo anno del nuovo ciclo.

in un unico volume di 465 carte e sono attualmente conservati presso l'Archivio di Stato di Napoli nel fondo Diversi della Sommaria.

Una lettura attenta del registro consente di notare come il primo *quaternus* abbia una struttura omogenea e coerente, poiché i singoli funzionari, sebbene operanti in centri diversi della provincia di Terra d'Otranto, sono convocati dal maestro razionale a rendicontare le proprie attività durante l'anno della settima indizione. Il secondo *quaternus* riprende perfettamente la struttura del primo fino a c.⁹. 159v¹⁰. Gli amministratori tarantini, infatti, sono tutti convocati al controllo durante l'ottava indizione. Nelle carte successive non viene più rispettato il criterio dell'anno di computo, ma sembra che esse siano divise in sezioni, ciascuna dedicata ad una funzione amministrativa diversa, svolta in anni indizionali differenti (dalla quattordicesima alla quarta indizione). Questa configurazione eterogenea può far supporre che le cc. 160-228v non facessero originariamente parte del *quaternus*, ma che siano state rilegate insieme alle altre in un secondo momento, senza che gli archivisti prestassero troppa attenzione al fatto che i controlli del razionale erano stati effettuati in anni diversi dall'ottava indizione.

Il registro 248 si configura, dunque, come una fonte ricca di informazioni relative all'intero dominio orsiniano per un arco temporale piuttosto ampio. Nell'attività di ricerca condotta per il progetto PRIN 2006-2008 "Geografia e linguaggi politici alla fine del Medioevo", i due *quaterni* sono stati studiati e analizzati dall'Unità di Ricerca di Lecce al fine di far emergere i tratti peculiari della geografia territoriale dello stato orsiniano. Le Unità di Ricerca del Politecnico di Milano e l'Unità dell'Università del Salento hanno elaborato una struttura tabellare capace di delineare la geografia insediativa, amministrativa ed economica del Principato che consentisse, in un secondo momento, di predisporre la struttura di una base di dati (Carrion e Migliaccio, 2009). Per valorizzare le informazioni presenti nel registro 248, è stata

⁹ carta

¹⁰ verso

costruita una matrice di dati su fogli di lavoro, individuando una serie di campi significativi:

- toponimo del centro preso in considerazione, così come è riportato dalla fonte,
- numero di fuochi soggetto al pagamento dell'imposta diretta,
- distretto erariale,
- località sede di tesoreria generale,
- località sede di capitaneato,
- località sede di castellania,
- località sede di dogana,
- località sede di fondaco,
- nome dell'erario generale,
- nome dell'erario locale,
- nome del tesoriere generale,
- nome del tesoriere locale,
- nome del capitano,
- nome del castellano,
- importo versato da ciascun centro per il pagamento dell'imposta diretta,
- importo versato da ciascun centro per il pagamento del focatico,
- importo versato da ciascun centro per il pagamento delle collette,
- quantitativo di tomoli di sale¹¹ acquistato,
- importo versato per il mantenimento delle condotte militari del re,
- importo versato da ciascun centro per il diritto di platea,
- importo versato da ciascun centro per il diritto di affida,
- importo versato da ciascun centro per la bagliva.

¹¹ A Taranto i cittadini avevano il diritto di usufruire delle saline, ma con l'arrivo dei Normanni e l'introduzione del feudalesimo le saline divennero oggetto di donazione, passando così ai baroni che ne affidavano la gestione ai salinarii. Poi con Federico II il commercio del sale divenne monopolio statale; mentre con gli Aragonesi venne introdotta una riforma fiscale che prevedeva la distribuzione di un tomolo di sale in cambio di 5 carlini per ogni famiglia e questa tassa prese il nome di tassa focatica.

L'elenco completo con i 47 campi previsti nella struttura tabellare è disponibile nell'Appendice A.

Ogni centro inserito nella tabella di dati è stato georeferenziato, attraverso l'attribuzione di coordinate piane, ed è stato assegnato un coefficiente numerico chiamato "grado di affidabilità" che varia da 1 a 5 all'aumentare del livello di precisione della georeferenziazione (Carrion et al., 2008). Ad esempio, in corrispondenza di un edificio ben identificabile come una chiesa o un convento è stato assegnato il grado di precisione massimo pari a 5, mentre in corrispondenza di piccoli agglomerati abitativi il coefficiente sarà pari a 4 e così via.

La scelta del tipo di geometria da utilizzare è stata a lungo discussa con gli Storici, in quanto risulta molto difficile dalle fonti definire con precisione i confini geografici di ogni località. Gli Storici erano divisi in due diverse correnti di pensiero: quelli che preferivano una geometria puntuale molto più cautelativa e quelli che avrebbero preferito ipotizzare un'estensione geografica per ogni luogo storico. Per il primo sviluppo dell'applicazione è stato deciso di abbracciare l'approccio più cautelativo ovvero utilizzare la geometria di tipo puntuale, nel prossimo futuro con l'aiuto degli Storici si potrà implementare la geometria poligonale.

Inoltre, a ciascuna località sono state collegate tutte le informazioni ricavate dall'analisi dei documenti, specificando la fonte documentaria portatrice del dato informativo e l'anno di riferimento che è di tipo indizionale. Questo complesso lavoro di schedatura delle fonti e di sistematizzazione dei dati in tabelle, svolto dai colleghi dell'Università del Salento, permetterà di eseguire analisi capaci di far emergere i caratteri distintivi della geografia politica del dominio orsiniano.

1.2 PERCHÉ CREARE UN DATABASE PER I DATI STORICI?

L'approccio di gestione dei dati, impiegato a questo punto del lavoro, è chiamato *file-based* e sfrutta la presenza di archivi o file per memorizzare i dati in modo persistente sulla memoria di massa di un *computer*: un *file* consente di memorizzare e ricercare dati, ma fornisce solo semplici meccanismi di accesso e condivisione. Eventuali dati di interesse per più programmi sono replicati tante volte quanti sono i programmi che li utilizzano, con evidente ridondanza e possibilità di incoerenza.

Per superare questo tipo di inconvenienti, era stato deciso già alla fine del progetto “*Geografia e linguaggi politici alla fine del Medioevo*” (PRIN 2006) di costruire una banca dati gestendo in modo integrato e flessibile le informazioni, limitando i rischi di ridondanza e incoerenza (Atzeni et al., 1999). La soluzione individuata consiste nella creazione di una base di dati che verrà gestita da appositi sistemi software, detti DBMS (*Database Management System*). Un DBMS è un sistema software in grado di gestire collezioni di dati che siano grandi, condivise e persistenti, assicurando la loro affidabilità e privacy.

Di seguito le principali caratteristiche delle basi di dati e dei DBMS:

- le basi di dati possono essere di grandi dimensioni;
- le basi di dati sono condivise, nel senso che applicazioni e utenti diversi devono poter accedere, secondo opportune modalità, a dati comuni. In questo modo si riduce la ridondanza dei dati, poiché si evitano ripetizioni e conseguentemente si riduce la possibilità di inconsistenze: se ogni dato è memorizzato sul calcolatore in modo univoco, non è possibile incorrere in disallineamenti;
- le basi di dati sono persistenti, cioè hanno un tempo di vita che non è limitato a quello delle singole esecuzioni dei programmi che le utilizzano;
- i DBMS garantiscono l'affidabilità, cioè la capacità, in caso di malfunzionamenti dell'*hardware* o del *software*, di conservare intatto il

- contenuto della banca dati (o almeno di permetterne la ricostruzione). A questo scopo i DBMS forniscono diverse funzionalità di salvataggio e ripristino (backup e recovery);
- i DBMS garantiscono la privacy dei dati; ciascun utente registrato viene abilitato a svolgere solo determinate azioni sui dati, attraverso meccanismi di autorizzazione;
 - i DBMS sono programmi efficienti, ovvero sono in grado di svolgere operazioni utilizzando un insieme di risorse, in termini di tempo e spazio, accettabile per gli utenti; naturalmente l'efficienza dipende fortemente dalle scelte prese nella fase di progettazione della base di dati e dal dimensionamento del sistema informatico che ospita la banca dati;
 - i DBMS devono essere efficaci, cioè rendere produttive le attività dei suoi utenti; l'attività di progettazione della banca dati e delle sue applicazioni mira a garantire una buona efficacia del sistema.

Per la realizzazione del database di dati storici si è deciso di ricorrere al “modello relazionale” (Codd, 1970) dei dati, attualmente il più diffuso che consente di organizzare i dati in insiemi di record a struttura fissa. In tale modello una relazione viene rappresentata per mezzo di una tabella, le cui righe rappresentano i record e le cui colonne rappresentano attributi dei record.

Il modello relazionale offre diversi vantaggi a confronto con i modelli di organizzazione di dati precedenti:

- è basato su una solida e consolidata teoria matematica (la teoria degli insiemi) con quello che ne consegue in termini di operazioni;
- è flessibile. Non ci sono riferimenti espliciti (non ci sono puntatori) che legano le tabelle fra di loro. Ogni tabella si può collegare a qualsiasi altra definita nel database utilizzando gli operatori che verranno illustrati nei prossimi paragrafi;

- garantisce l'indipendenza dall'organizzazione fisica dei dati. Chi interroga un DBMS non ha necessità di conoscere dove e come sono registrati in memoria di massa i dati: dal suo punto di vista esistono solo le relazioni;
- garantisce l'indipendenza dell'organizzazione logica dei dati dalle procedure di accesso ad essi. Le tabelle non nascono per rispondere a determinate query ma per descrivere la realtà di interesse;
- garantisce un elevato livello di sicurezza. I dati sono gestiti, in modo centralizzato, dal DBMS e le procedure lo interrogano per ottenere quelli di proprio interesse. Il DBMS può quindi tenere nota di chi accede e a quali dati;
- garantisce una bassa e controllata ridondanza. La duplicazione dei dati può essere limitata a solo a casi indispensabili per l'ottimizzazione della gestione (ridondanza strategica). Qualora si dovesse presentare una ridondanza la relazione può sempre essere divisa in due sotto relazioni con dati non ridondanti.

1.3 METODOLOGIA DI PROGETTAZIONE DI UN SISTEMA INFORMATIVO E DI UNA BASE DI DATI

La progettazione di una base di dati costituisce solo una delle componenti del processo di sviluppo di un sistema informativo complesso e va quindi inquadrata in un contesto più ampio, quello del ciclo di vita dei sistemi informativi (Figura 1.1).

Il ciclo di vita di un sistema informativo comprende, generalmente, le seguenti attività.

- **Studio di fattibilità**, serve a definire in maniera per quanto possibile precisa, i costi delle varie alternative possibili e a stabilire le priorità di realizzazione delle varie componenti del sistema. Questa fase sarà descritta nel paragrafo 1.4.

- **Raccolta e analisi dei requisiti**, consiste nell'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, ma generalmente informale, dei dati coinvolti e delle operazioni su di essi (Teorey et al., 2005). Questo passaggio sarà approfondito nel paragrafo 1.5.
- **Progettazione**, si divide generalmente in progettazione dei dati e progettazione delle applicazioni. Nella prima si individua la struttura e l'organizzazione che i dati dovranno avere (paragrafi 1.6, 1.7, 1.8 e 1.9), nell'altra si definiscono le caratteristiche dei programmi applicativi (paragrafo 2.3 e 2.4).
- **Implementazione**, consiste nella realizzazione del sistema informativo secondo la struttura e le caratteristiche definite nella fase di progettazione. Viene costruita e popolata la base di dati (paragrafo 1.10) e viene prodotto il codice dei programmi (paragrafo 2.4).
- **Validazione e collaudo**, serve a verificare il corretto funzionamento e la qualità del sistema informativo. In questa fase il sistema informativo diventa operativo ed esegue compiti per i quali era stato originariamente progettato. Se non si verificano malfunzionamenti o revisioni delle funzionalità del sistema, questa attività richiede solo operazioni di gestione e manutenzione.

Questo processo non è quasi mai strettamente sequenziale in quanto spesso, durante l'esecuzione di una delle attività citate, è necessario rivedere decisioni prese nell'attività precedente.

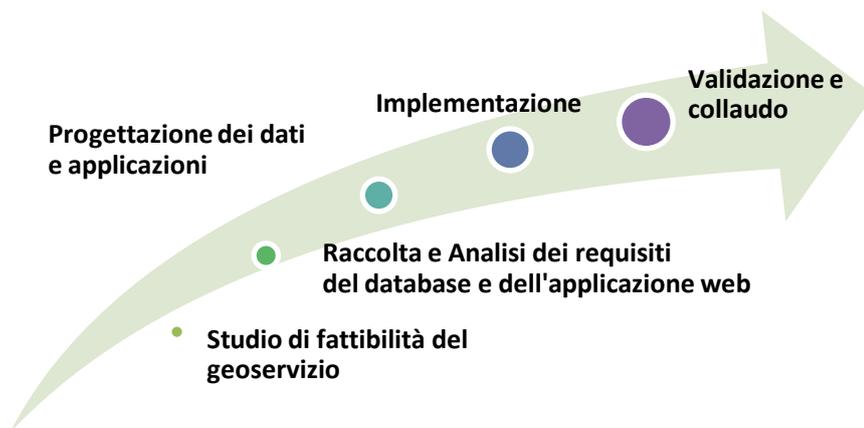


Figura 1.1 - Le cinque fasi di progettazione di un sistema informativo. La fase di progettazione dei dati è a sua volta suddivisa in tre fasi (Figura 1.2).

Le basi di dati costituiscono in effetti solo una delle componenti di un sistema informativo che include anche i programmi applicativi, le interfacce grafiche utente e altri programmi di servizio. Comunque, il ruolo centrale che i dati hanno in un sistema informativo giustifica ampiamente uno studio autonomo relativo alla progettazione delle basi di dati. In questo capitolo saranno approfonditi solo gli aspetti dello sviluppo dei sistemi informativi che riguardano da vicino il progetto delle basi di dati ovvero la terza fase del ciclo di vita dei sistemi informativi. Questa maniera di procedere è coerente con l'approccio allo sviluppo dei sistemi informativi basato sui dati, in cui l'attenzione è centrata sui dati e sulle loro proprietà. Questo approccio prevede prima la progettazione della base di dati e, successivamente, la realizzazione delle applicazioni che la utilizzano.

Negli anni si è consolidata una metodologia di progettazione delle basi di dati che soddisfa le proprietà di qualità del prodotto finale, facilità d'uso e generalità rispetto alle applicazioni. Tale metodologia è articolata in tre fasi, la Progettazione Concettuale, la Progettazione Logica e la progettazione fisica (Jardine, 1977) da seguire in cascata (Figura 1.2) 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).

Nella fase di progettazione è bene fare 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, che contenga cioè tutte le informazioni necessarie per eseguire le operazioni previste.



Figura 1.2 - Le tre fasi della progettazione di un database.

Lo scopo della **progettazione concettuale** di una banca dati consiste nel rappresentare le specifiche raccolte nella fase dell’analisi dei requisiti in termini di una descrizione “formale”, completa e indipendente dai criteri di rappresentazione utilizzati nei sistemi di gestione delle basi di dati. Il prodotto finale di questa fase è chiamato schema concettuale e fa riferimento a un modello concettuale dei dati. I modelli concettuali consentono di descrivere l’organizzazione dei dati ad un alto livello di astrazione, senza tenere conto degli aspetti implementativi (Atzeni et al., 1999). Lo schema concettuale rappresenta quindi il contenuto informativo della banca dati, senza tenere conto delle modalità con le quali queste informazioni verranno codificate in un sistema reale, né dell’efficienza dei programmi che faranno uso di queste informazioni.

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. Questa fase è del tutto indipendente dal DBMS che verrà utilizzato per la creazione e gestione del database, il software può quindi essere scelto a posteriori. Nella **progettazione fisica** si fa uso dello schema logico e delle specifiche sulle operazioni per ottimizzare le prestazioni del sistema. In questa fase, è necessario tenere conto delle caratteristiche del particolare sistema di gestione di basi di dati utilizzato (DBMS).

Il risultato della progettazione di una base di dati non si riduce al solo schema fisico, ma è costituito anche dallo schema concettuale e dallo schema logico. Lo schema concettuale fornisce infatti una rappresentazione della base di dati di alto livello, che può essere molto utile a livello documentativo, mentre lo schema logico fornisce una descrizione concreta del contenuto della base di dati che, prescindendo dagli aspetti documentativi, può essere utile come riferimento per le operazioni di interrogazione e aggiornamento (Atzeni et al., 1999).

1.4 STUDIO DI FATTIBILITÀ DEL GEOSERVIZIO

Per la realizzazione del geoservizio sono state considerate due alternative: una *suite* di tecnologie *open source* e una *suite* di tecnologie ESRI (Tabella 1.1). Entrambe le soluzioni prevedono l'utilizzo di programmi molto diffusi e con un'attiva comunità di utilizzatori e sviluppatori che garantiscono aggiornamenti frequenti delle tecnologie e assistenza agli utenti.

La sintesi dello studio di fattibilità è riportata in Tabella 1.2.

Tabella 1.1 – Suite di tecnologie considerate: alternativa open source e ESRI.

	OSS	ESRI
DBMS	PostgreSQL + PostGIS o SpatiaLite	PostgreSQL + PostGIS + ArcSDE
GIS Server	QGIS Server o Geoserver	ArcGIS Server
Web Mapping	OpenLayers o Leaflet	ArcGIS API ¹² for JavaScript

Tabella 1.2 – Analisi dei vantaggi (in verde) e svantaggi (in arancione) tra una suite di GIS open source e la suite ESRI (ArcGIS Desktop, Server e ArcSDE).

suite GIS open source	suite ESRI
nessun costo di acquisto licenze	licenze ArcGIS Desktop, Server e ArcSDE
indipendenza dai fornitori	dipendenza dai fornitori ESRI
soluzioni flessibili, scalabili, personalizzabili	soluzioni poco flessibili e poco personalizzabili
ricca documentazione su internet	ricca documentazione online
comunità attiva di sviluppo	assistenza con tempi più lenti
mercato instabile ¹³	mercato stabile
contribuzione alla diffusione ed evoluzione dell'informatica libera in applicazioni GIS	non richiede personale esperto per la manutenzione
richiede personale esperto	richiede personale esperto per l'installazione e lo sviluppo di applicazioni web
tempi di realizzazione maggiori	tempi di realizzazione inferiori

¹² Con il termine *Application Programming Interface* API (Interfaccia di Programmazione di un'Applicazione) si indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma.

¹³ Il modello attuale di distribuzione del software open source è rischioso in un mercato aggressivo come quello dell'IT: il pericolo è che il distributore scelto esca dal mercato e non siano garantiti gli aggiornamenti.

Dall'analisi di fattibilità è emerso che una soluzione completamente *open source* offre molti più vantaggi rispetto all'alternativa proprietaria. Così è stato deciso che per evitare di impegnare le Unità di Ricerca coinvolte nel progetto all'acquisto di licenze (che sarebbero risultate molto onerose sul budget complessivo del progetto) si è optato per l'uso di tecnologie *free e open source*. Le tecnologie *open source* presentano un altro vantaggio: consentono la realizzazione di soluzioni scalabili, cioè che possono essere estese, modificate e migliorate da chiunque ne abbia le capacità e le competenze. Inoltre avendo a disposizione il codice sorgente dei programmi è possibile fare delle personalizzazioni e togliere funzionalità superflue, ottenendo un sistema snello e dimensionato alle proprie esigenze. D'altra parte le soluzioni *open source* richiedono personale con maggiore conoscenza informatica non solo in fase di sviluppo ma anche per la manutenzione del servizio e in mancanza di queste risorse può risultare necessario rivolgersi a personale esterno a pagamento. Infine, per capire la sinergia tra le tecnologie *open source* coinvolte è necessario un "periodo di addestramento", che influisce in maniera rilevante sui tempi di realizzazione e sviluppo.

1.5 ANALISI DEI REQUISITI DEL DATABASE DI DATI STORICI

Il database dei dati storici dovrà consentire la raccolta, consultazione ed interrogazione dei dati da parte delle Unità di Ricerca coinvolte nel progetto. I dati che il database dovrà contenere provengono da fonti di diversa origine, risalgono a un'età compresa nell'arco temporale tra il XIII e il XVI secolo e riguardano lo Stato Orsinano, che comprendeva il Principato di Taranto, la Contea di Lecce e la Contea di Soleto nel Regno di Napoli. Per il prototipo sono stati considerati i dati estratti dalla fonte *Quaternus Declaracionum*, forniti dall'Unità di Ricerca di Lecce. Tali dati sono prevalentemente di tipo tributario e fiscale, tuttavia da essi è stato possibile estrarre

anche informazioni relative a personaggi storici che lavoravano presso uffici di riscossione delle tasse quali: castellanie, centri capitaneali, centri giurisdizionali, tesorerie, centri vicariali. Tali ufficiali potevano ricoprire una o più cariche amministrative e, a seconda dei casi, prendono il nome di erari (generali o locali), baiuli, capitani, vicari, tesorieri (di riferimento e non), castellani ed esattori.

La banca dati dovrà essere di tipo relazionale per i motivi discussi nel paragrafo 1.4, passando così da una semplice struttura monotabellare ad una struttura dati che possa essere gestita come un database. In questo modo gli utilizzatori del sistema avranno la possibilità di eseguire sui dati interrogazioni con sintassi più complesse, di selezionare dati in base a uno o più criteri e di consultare l'archivio in maniera più snella ed efficiente.

I dati presentano una componente spaziale, essendo stati georeferenziati (dove possibile) secondo la località attuale corrispondente al toponimo latino. Il sistema dovrà consentire di ottenere rappresentazioni di tipo cartografico dalle informazioni presenti nel database, in modo da offrire un ulteriore punto di vista sulle informazioni storiche. Questo nuovo punto di vista potrà essere sfruttato per l'interpretazione da parte degli storici, che potranno trarre vantaggio dalla visualizzazione della distribuzione sul territorio delle località a cui sono riferiti i dati degli archivi da loro consultati.

È particolarmente importante per gli storici che da tutte le informazioni sia sempre possibile risalire al documento da cui è stata estratta l'informazione in esame, ovvero deve essere sempre presente l'indicazione dell'archivio, del fondo e del numero del pezzo, secondo il sistema di catalogazione dei manoscritti storici.

La base di dati dovrà presentare una struttura flessibile e modulare, in modo da non escludere l'inserimento di altri dati provenienti da nuove fonti storiche di tipo fiscale e non. In particolare si desidera poter integrare in futuro un ipotetico database relativo ai personaggi storici riconosciuti come storicamente esistiti per i quali si dispone di una scheda anagrafica completa. Per consentire una più agevole consultazione e

interrogazione dei dati alfanumerici e geografici, sarà progettata e realizzata un'interfaccia grafica utente o GUI (*Graphical User Interface*), realizzata con tecnologie web *client-side*. Tale interfaccia, che verrà chiamata “Geografie dell’Italia Medievale, XIII – XV secolo” sarà descritta in dettaglio nel Capitolo 2.

1.6 PROGETTAZIONE CONCETTUALE DEL DATABASE STORICO

La progettazione concettuale di una base di dati consiste nella costruzione di uno schema Entità-Relazioni in grado di descrivere al meglio le specifiche sui dati di una applicazione. Il modello Entità-Relazioni (Chen, 1976), indicato anche con l'abbreviazione ER, è un modello concettuale dei dati e, come tale, fornisce una serie di costrutti, atte a descrivere la realtà di interesse in una maniera facile da comprendere e che prescinde dai criteri di organizzazione dei dati nei calcolatori.

I costrutti principali del modello ER sono:

- le entità;
- gli attributi semplici o composti con relative cardinalità;
- gli identificatori;
- le relazioni con relative cardinalità;
- le generalizzazioni (e il caso particolare dei “sottoinsiemi”).

Le **entità** rappresentano classi di oggetti che hanno proprietà comuni ed esistenza “autonoma” ai fini dell'applicazione di interesse. In uno schema, ogni entità ha un nome che la identifica univocamente e viene rappresentata mediante un rettangolo. Gli elementi contenuti nelle entità prendono il nome di occorrenze. Un'occorrenza di un'entità è un oggetto o istanza della classe che l'entità rappresenta.

Ad esempio “Brindisi” è un’occorrenza dell’entità denominata “Luogo attuale”, così come “Brundisium” è un’occorrenza di “Luogo”; un’occorrenza di entità non è un valore che identifica l’oggetto ma è l’oggetto stesso.

Seguendo la metodologia illustrata nel paragrafo 1.2 in questa fase si è ragionato sulla struttura monotabellare di partenza per individuare le entità da rappresentare nello schema ER. Tutte le entità individuate si possono classificare in due gruppi: quelle che presentano una componente spaziale e quelle puramente alfanumeriche. La componente cartografica, dove presente, è stata esplicitata aggiungendo come attributi alle entità le coordinate Est e Nord, definite rispettivamente anche x e y. L’elenco di tutte le entità individuate, suddivise tra spaziali e non spaziali, è riportato nelle Tabelle 1.3 e 1.4. In queste tabelle è riportata anche una descrizione sintetica per ogni entità.

Tabella 1.3 – Elenco delle entità “spaziali” e relativa descrizione sintetica

<i>ENTITA'</i>	DESCRIZIONE
Luogo	raccoglie informazioni sui luoghi storici così come indicati nelle fonti.
Luogo attuale	raccoglie i dati relativi al toponimo attuale dove si localizza l’insediamento medievale citato nel documento storico.
Unità fiscale	centri di appartenenza fiscale dei toponimi.
Ufficio	uffici amministrativi di riscossione delle tasse: castellanie, centri capitaneali, centri giurisdizionali, tesorerie, centri vicariali, dogane, centri di affida e centri di fondaco.

Tabella 1.4 - Elenco delle entità “non spaziali” e relativa descrizione

<i>ENTITA'</i>	DESCRIZIONE
Distretto erariale	raccoglie le unità fiscali di appartenenza dei toponimi.
Fonte	documenti storici da cui sono state estratte le informazioni
Imposte	raccoglie le informazioni relative alle tasse: tipo di imposta, quantitativo, ecc.
Ufficiale	funzionari, così come rinvenuti nei documenti storici. Un ufficiale può ricoprire uno o più delle seguenti cariche: erario (generale o locale), baiulo, capitano, vicario, tesoriere (eventualmente di riferimento), castellano ed esattore. Un personaggio storico può comparire più volte se nel corso della sua vita ha ricoperto diverse cariche oppure se ha ricoperto la stessa carica per diversi anni.
Personaggio	Entità che raccoglie i dati relativi ai personaggi storici accertati. Possibile punto di collegamento con altre banche dati che archiviano informazioni sui personaggi storici.

In ogni entità sono presenti informazioni di interesse ai fini dell'applicazione, che prendono il nome di attributi e descrivono le proprietà elementari dell'entità. Un attributo associa a ciascuna occorrenza di entità un valore appartenente a un insieme, detto “dominio”, che contiene i valori ammissibili per l'attributo.

Per esempio, l'attributo “grado_affidabilità” dell'entità “Luogo” sopra citato può assumere solo valori interi tra 1 e 5 e analogamente l'attributo “toponimo_ita” di “Luogo attuale” può avere come dominio l'insieme di tutti gli insediamenti attuali italiani che possono essere riconducibili per nome o per posizione geografica agli insediamenti latini, così come riportati nelle fonti storiche.

Inoltre ogni entità è identificata in maniera univoca da un attributo semplice (con significato fisico o appositamente creato) oppure da un insieme di attributi, che prende il nome di **identificatore**.

Nelle Figure 1.3 e 1.4 viene mostrato come vengono rappresentati graficamente entità e relativi attributi: per ogni entità nello schema sono specificati gli identificatori, che si distinguono dagli attributi perché identificano in modo univoco le occorrenze di un'entità. Per quest'applicazione sono stati scelti identificatori semplici (un solo attributo) ed interni (non sono necessarie informazioni relative ad altre entità). Gli identificatori si distinguono graficamente per via del cerchio pieno e in questa applicazione il nome è nella forma "id_" + nome dell'entità.

Si osservi in Figura 1.3 che sono stati individuati due attributi "composti", ovvero che raccolgono attributi che presentano una certa affinità nel significato e uso: si tratta degli attributi "coordinate" e "tempo". Il primo è formato dagli attributi semplici "est_x" e "nord_y" mentre il secondo è composto da cinque attributi quali "tipo_anno", "indizione", "anno", "giorno", "mese", "anno". Infatti, nella gestione della componente temporale si è dovuto tener conto che nel *Quaternus Declaracionum* è stato usato un sistema di computo del tempo di tipo "indizionale". Per non perdere nessuna informazione relativa alla componente tempo, infatti, oltre agli attributi "giorno", "mese" e "anno" è necessario l'attributo "tipo_anno", un attributo booleano che indica se l'anno è indizionale oppure solare, e l'attributo "indizione", presente solo nel caso di anno indizionale, che indica il numero dell'anno calcolato all'interno di un ciclo di 15 anni.

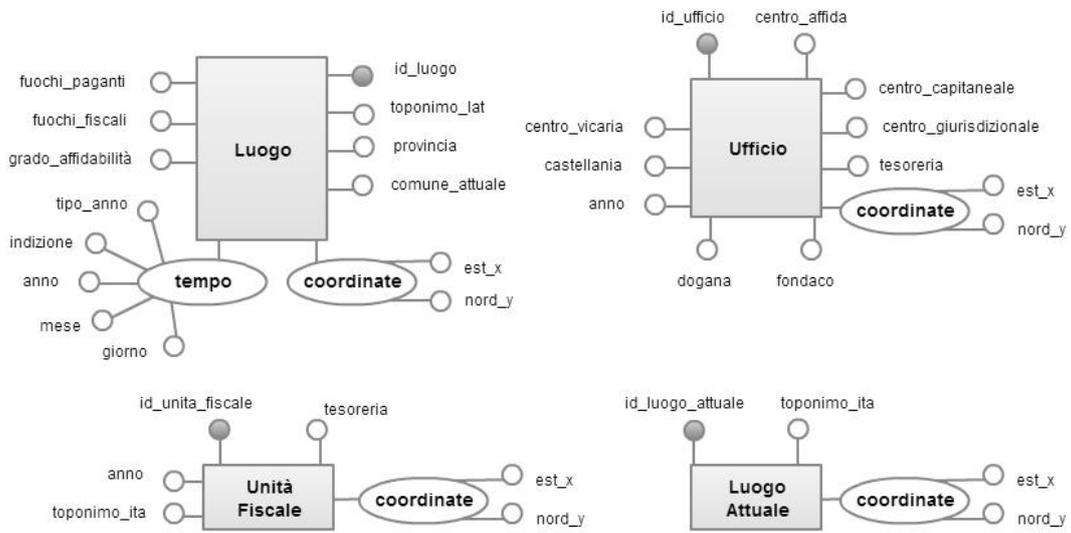


Figura 1.3 – Entità con riferimenti spaziali e relativi attributi

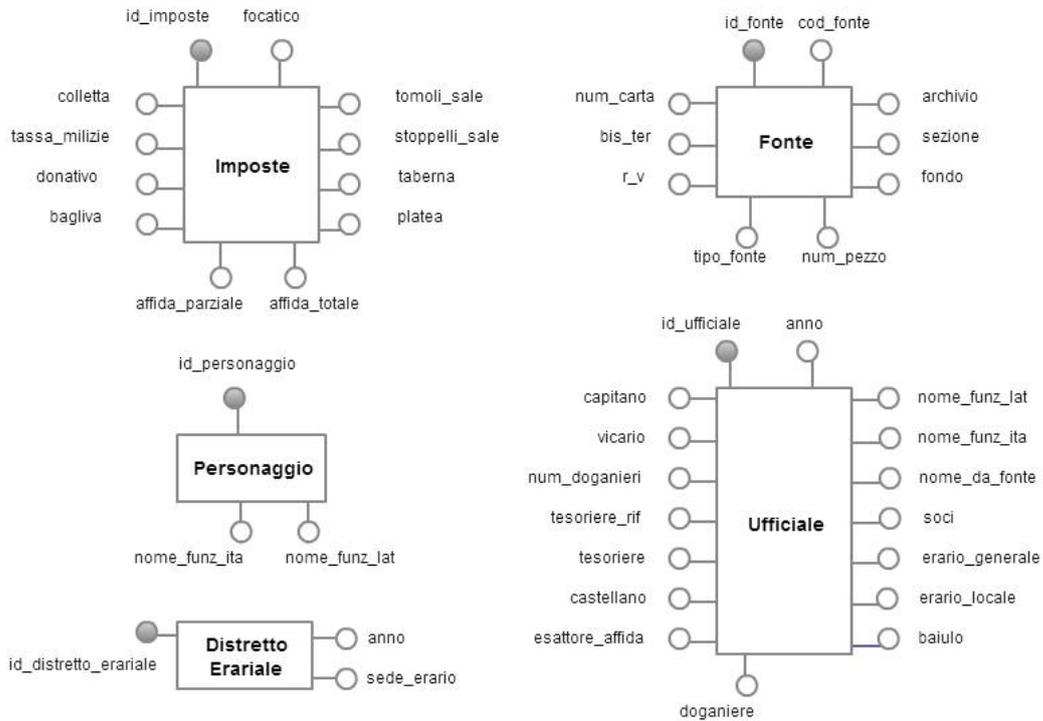


Figura 1.4 – Entità con informazioni esclusivamente alfanumeriche e relativi attributi

Per l'entità LUOGO (Figura 1.4) sono presenti due attributi relativi al pagamento del focatico: "fuochi paganti" e "fuochi fiscali". Secondo il sistema imposto nello stato orsiniano i fuochi, ovvero le unità familiari produttrici di reddito, dovevano al Principe un ducato d'oro ciascuno. Raramente il numero di fuochi fiscali corrispondeva al numero di fuochi paganti, quest'ultimo infatti era il risultato di accordi tra potere centrale e singoli insediamenti.

I nomi degli attributi derivano principalmente dai nomi delle colonne della struttura tabellare usata dagli storici per archiviare le informazioni. Meritano un approfondimento invece gli attributi dell'entità "Fonte", per la quale si è deciso di suddividere le diverse informazioni che compongono la segnatura di un documento storico negli attributi "tipo_fonte", "archivio", "fondo", "num_pezzo", "num_carta", "r_v" (*recto* o *verso*). L'identificatore dell'entità "Fonte" invece è un codice alfanumerico derivato calcolato a partire dagli attributi. Ad esempio, il codice F1010010248100210 può essere scomposto come in Tabella 1.5.

Tabella 1.5 – Struttura dell'identificatore dell'entità Fonte

<i>Attributo</i>	<i>Valore</i>	<i>Significato</i>
archivio	101	Archivio di Stato di Napoli
sezione	0	Nessuna sezione
fondo	1	Diversi Sommaria II
num_pezzo	0248	248
num_carta	1002	2
bis_ter	1	<Null>
r_v	0	recto

In Figura 1.5 viene presentato lo schema ER completo per il database storico progettato.

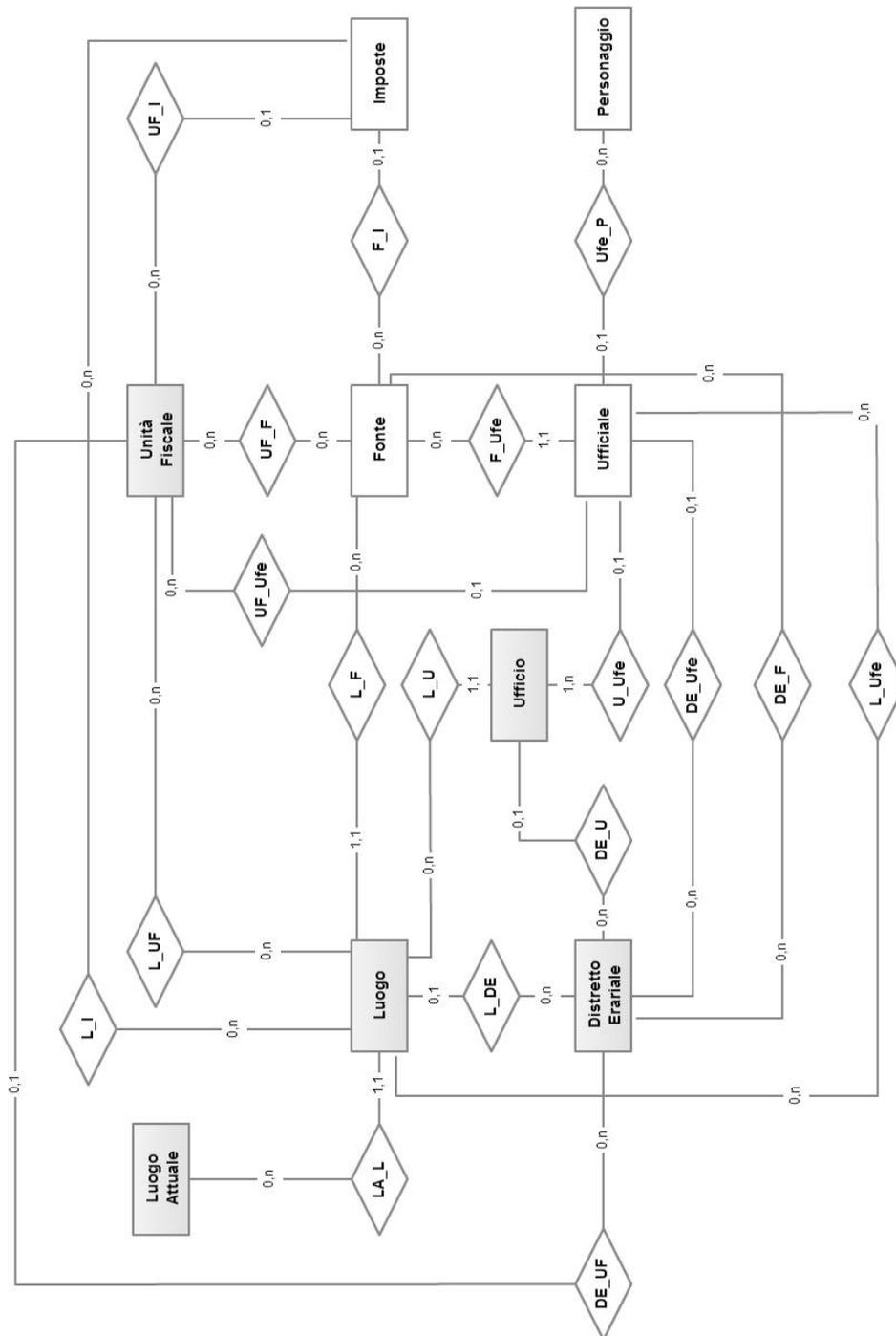


Figura 1.5 – Modello concettuale del database storico

Come già accennato nell'analisi dei requisiti (paragrafo 1.5), è particolarmente importante poter risalire in ogni momento al documento storico da cui sono state estratte le informazioni. Per venire incontro a questa esigenza si è deciso di creare molte associazioni con l'entità "Fonte". Le nuove associazioni con l'entità "Fonte" hanno generato grafi chiusi (chiamati anche "cicli") e, in alcuni casi, ridondanza di informazioni. Questa ridondanza è voluta e necessaria.

In uno schema ER per ogni entità partecipante ad una relazione viene specificata una **cardinalità** di relazione. Essa è una coppia di numeri naturali che specifica il numero minimo e massimo di istanze di relazione a cui una istanza dell'entità può partecipare. È possibile indicare con la costante N un numero generico maggiore di uno quando la cardinalità non è nota con precisione. Nel diagramma ER, la cardinalità di relazione di una entità etichetta l'arco che lega l'entità alla relazione.

Dall'analisi delle cardinalità delle relazioni dello schema ER, si può osservare che alcune sono di tipo 1:N ed altre invece presentano cardinalità N:N. In particolare, poiché un luogo storico nel nostro database è individuato dal nome, dalla fonte storica e dall'anno è logico supporre che a un insediamento attuale possano corrispondere più luoghi storici. Per questo motivo la relazione tra le entità "Luogo" e "Luogo attuale" è di tipo 1:N. Analogamente la relazione tra le entità "Personaggio" e "Ufficiale" è sempre di tipo 1:N, poiché un personaggio storico può assumere più cariche amministrative nel corso della sua carriera.

1.7 PROGETTAZIONE LOGICA DEL DATABASE STORICO

L'obiettivo della progettazione logica è quello di costruire uno schema logico in grado di descrivere, in maniera corretta ed efficiente, tutte le informazioni contenute nello schema ER prodotto nella fase di progettazione concettuale. Non si tratta di una semplice traduzione da un modello a un altro perché, prima di passare allo schema logico, lo schema ER va ristrutturato per soddisfare due esigenze: quella di

“semplificare” la traduzione e quella di “ottimizzare” il progetto. La semplificazione si rende necessaria perché non tutti i costrutti del modello ER hanno una traduzione naturale nei modelli logici. Per esempio, mentre una entità può essere facilmente rappresentata da una relazione del modello relazionale (avente gli stessi attributi dell’entità), per gli attributi composti esistono diverse possibilità. Inoltre, mentre la progettazione concettuale ha come obiettivo la rappresentazione accurata e naturale dei dati di interesse dal punto di vista del significato che hanno nell’applicazione, la progettazione logica costituisce la base per l’effettiva realizzazione dell’applicazione e deve tenere conto, per quanto possibile, delle sue prestazioni: questa necessità può portare a una ristrutturazione dello schema concettuale che renda più efficiente l’esecuzione delle operazioni previste. Pertanto, è necessario prevedere sia un’attività di riorganizzazione, sia un’attività di traduzione dal modello concettuale a quello logico (Atzeni et al., 1999).

Lo schema logico completo è riportato in Figura 1.6.

I dati di ingresso di questa fase sono lo schema concettuale prodotto (Figura 1.5) e il carico applicativo¹⁴ previsto. La semplificazione dello schema ER consisterà principalmente nell’eliminazione degli attributi composti “tempo” e “coordinate” presenti nelle entità “Luogo”, “Ufficio”, “Unità Fiscale” e “Luogo Attuale” (Figura 1.3). Questa ristrutturazione si rende necessaria perché il modello relazionale non permette di rappresentare in maniera diretta questo tipo di attributo. Gli attributi composti sono stati scissi negli attributi semplici che li costituiscono.

Si osservi che le associazioni di tipo N:N sono state trasformate in nuove tabelle, chiamate tabelle “ponte”, mentre le associazioni con cardinalità 1:N sono state implementate aggiungendo la chiave esterna nell’entità che partecipa con cardinalità 1 (ad esempio nell’associazione Ufficiale-Ufficio la chiave esterna è aggiunta all’entità “Ufficiale” che partecipa con cardinalità 1).

L’elenco delle tabelle “ponte” è riportato in Tabella 1.7.

¹⁴ Il carico applicativo è la stima del tempo di esecuzione delle operazioni di principale interesse e lo spazio di memoria necessario per archiviare le informazioni.

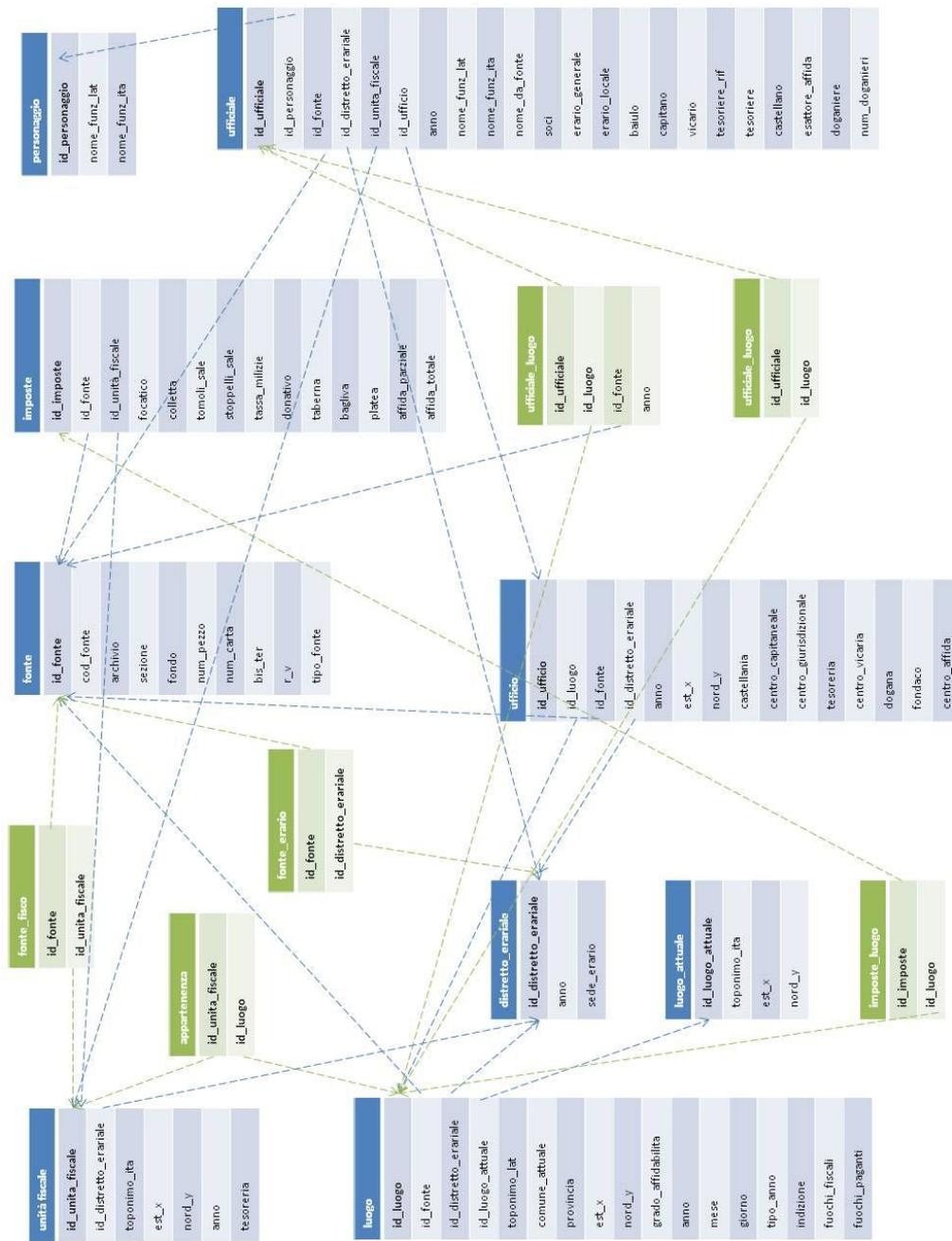


Figura 1.6 – Schema logico della base dati dei dati storici. In verde le tabelle “ponte” e in grassetto le chiavi primarie (o “primary key”) delle entità.

Tabella 1.6 – Elenco delle associazioni con cardinalità N:N

<i>Associazione</i>	<i>Nome tabella “ponte”</i>
Fonte - Unità Fiscale	fonte_fisco
Fonte - Distretto Erariale	fonte_erario
Imposte - Luogo	imposte_luogo
Luogo – Unità Fiscale	appartenenza
Ufficiale - Luogo	ufficiale_luogo

Si osservi che, rispetto al modello concettuale, sono state aggiunte due nuove associazioni: Fonte – Ufficio e Fonte – Ufficiale_Luogo, che consentono di avere sempre il riferimento del documento storico da cui sono state estratti i dati, informazione molto importante per gli Storici. Tuttavia le nuove associazioni hanno incrementato la rigidità e la ridondanza dei dati nel database. Ad esempio, la chiave esterna “id_fonte” nell’entità “Ufficio” può essere ricavata dalla associazione Fonte - Luogo – Ufficio. La presenza di ridondanze, tuttavia, non è sempre indesiderabile e va valutata caso per caso dopo avere eseguito un’approfondita analisi dei vantaggi e svantaggi: in questo caso il vantaggio principale è la riduzione degli accessi necessari per calcolare il dato derivato, lo svantaggio maggiore è una maggiore occupazione della memoria (che in questo caso possiamo considerare un costo trascurabile in quanto si tratta di pochi *kilobyte*).

Per lo schema logico è stata creata una documentazione di supporto, che dovrebbe facilitare l’interpretazione dello schema stesso e descrivere le proprietà dei dati che non possono essere esplicitati nello schema logico. Tale documentazione prende il nome di Dizionario dei dati ed è riportato nell’Appendice B.

1.8 SCELTA DEL DBMS CON SUPPORTO DI GESTIONE AI DATI SPAZIALI: POSTGRESQL E POSTGIS

Prima di procedere con l'ultima fase della progettazione di una banca dati (la progettazione fisica) è necessario analizzare i diversi DBMS e scegliere quello più opportuno. La scelta di un DBMS è influenzata da fattori tecnici (ad esempio le strutture di memorizzazione e di accesso ai file offerte dal DBMS), da fattori economici e organizzativi (ad esempio, costo del sistema e dell'assistenza). Negli ultimi anni sono diventate molto utilizzate le tecnologie gratuite e *open source*, soprattutto per architetture di modeste dimensioni e con pochi utenti.

Nel caso in esame molte informazioni presentano una componente spaziale, la scelta del software da utilizzare sarà quindi vincolata alla possibilità di rappresentare le informazioni di tipo geografico. In ambito GIS uno Spatial DBMS rappresenta il mezzo più indicato per la memorizzazione dei dati geografici. Uno Spatial DBMS è essenzialmente costituito da un DBMS di tipo convenzionale che deve però anche essere in grado di supportare lo speciale *datatype* GEOMETRY. Inoltre, per eseguire le ricerche sui dati geografici in maniera efficiente e senza la scansione di ogni record del database sono necessari gli indici spaziali. L'indicizzazione velocizza la ricerca organizzando i dati secondo un albero di ricerca che può essere percorso velocemente per l'individuazione di un record specifico. Tramite Spatial Index RTree¹⁵ (Guttman, 1984; Manolopoulos, 2006) uno Spatial DBMS è in grado di ottimizzare i tempi di accesso ai dati GEOMETRY.

¹⁵ Gli R-tree sono strutture dati albero usate per indicizzare le coordinate spaziali (X, Y) di dati geografici. La struttura dati divide lo spazio in MBR (*minimum bounding rectangles*) innestati gerarchicamente e quando possibile sovrapposti. R-Tree divide i dati in rettangoli, sotto-rettangoli e così via. È una struttura di indicizzazione auto-ottimizzante.

L'R-tree è stato proposto da Antonin Guttman nel 1984 e ha trovato un uso significativo in contesti sia teorica che applicata (Manolopoulos et al., 2006). Gli R-tree sono molto utilizzati dai sistemi di navigazione nelle richieste che coinvolgono informazioni spaziali, come ad esempio: "trova tutti i musei a circa 2 km dalla mia posizione attuale", "trova tutti i segmenti su strada in 2 km di mia posizione" o "trova il distributore di benzina più vicino".

Le specifiche OGC-SFS (*Open Geospatial Consortium - Simple Features Interface Standard*) definiscono i formati WKT e WKB¹⁶ per la rappresentazione dei dati GEOMETRY e definiscono inoltre un esteso set di funzioni SQL che consentono l'analisi geometrico-spaziale dei dati. Alla data di redazione di questo testo nell'ambito *open source* i più diffusi DBMS per l'archiviazione e gestione delle informazioni spaziali sono PostGIS e SpatiaLite.

SpatiaLite rappresenta un'estensione per dati spaziali assai simile a PostGIS: tuttavia SpatiaLite si basa sul DBMS SQLite. In questo modo si ottiene un vero e proprio Spatial DBMS che supporta praticamente l'intera gamma delle specifiche SQL ed OGC-SFS, ma che è anche nel contempo assai leggero, strutturalmente semplice e privo di qualsiasi complessità. Questa soluzione è fortemente indicata per usi di tipo Personal/Desktop dove la massima leggerezza, semplicità ed immediatezza d'uso rappresentano un indiscutibile vantaggio.

La soluzione PostgreSQL integrata dall'estensione PostGIS è invece largamente consigliabile laddove un'architettura *client-server* è indispensabile, come nel caso in esame, mentre una certa complessità sistemistica non rappresenta un problema.

PostgreSQL è disponibile sia nella versione compilata (che comprende i file eseguibili o binari) sia nella versione da compilare (comprende solo i sorgenti) per i Sistemi Operativi Linux, Windows, Mac OS X e Solaris. In questo lavoro è stata utilizzata la versione compilata per la distribuzione Linux Ubuntu.

¹⁶ Il *Well-known text* (WKT) è un linguaggio creato per rappresentare: oggetti di geometria vettoriale su una mappa, un Sistema di Coordinate di Riferimento (una proiezione), una trasformazione tra sistemi di coordinate.

Un equivalente binario, il *Well-Known Binary* (WKB) è generalmente usato per salvare le stesse informazioni in un database. Il formato è mantenuto dall'Open Geospatial Consortium (OGC).

1.9 PROGETTAZIONE FISICA DEL DATABASE STORICO

Nell'ambito del processo di progettazione di una base di dati, la fase finale è quella della "progettazione fisica", che, ricevendo in ingresso lo schema logico, produce in uscita lo schema "fisico" della base di dati. L'attività di progettazione fisica di una base di dati relazionale può essere molto complessa, perché oltre alle scelte relative alle strutture fisiche, può essere necessario definire molti parametri, che vanno dalle dimensioni iniziali dei file alle possibilità di espansione (Atzeni et al., 2003). A parte l'attività di specifica degli schemi delle relazioni, con i tipi di dati (chiamati anche *datatype*) ammessi per i vari attributi e dimensioni dei campi, la progettazione fisica può essere ricondotta a un'attività di individuazione degli indici. Gli indici sono speciali strutture di accesso a relazioni che vengono definiti su attributi delle relazioni stesse e garantiscono l'accesso diretto, e quindi efficiente, ai dati quando l'interrogazione si basa su tali attributi.

Per la creazione della banca dati con relative tabelle, attributi, indici, ecc sono stati creati degli *script* in linguaggio SQL, grazie ai quali è possibile in qualunque momento ripristinare il database utilizzando una copia di *backup*.

Ad esempio per la tabella "Personaggio" si avrà:

```
CREATE TABLE public.personaggio (id_personaggio
INTEGER NOT NULL, nome_funz_lat VARCHAR(75),
nome_funz_ita VARCHAR(75),CONSTRAINT personaggio_pk
PRIMARY KEY (id_personaggio))
```

Inoltre, sono state create viste utili per agevolare la consultazione dei dati. Gli script di creazione delle viste hanno la forma:

```
CREATE OR REPLACE VIEW vista
(id_luogo,id_luogo_attuale,...)
as select vista1.id_luogo,
vista1.id_luogo_attuale,... ,
from vista1 full join ufficiale_luogo on
vista1.id_luogo = ufficiale_luogo.id_luogo...
```


Dall'analisi dello schema fisico si può osservare che in generale i tipi di dato utilizzati sono i seguenti: *integer*, *varchar*, *double*, *boolean* e *bigint*.

Generalmente le operazioni più delicate in una base di dati relazionale sono quelle di selezione (che corrisponde all'accesso di uno o più record sulla base dei valori di uno o più attributi) e di join (che richiede di combinare tuple di relazioni diverse sulla base di valori di uno o più attributi di ognuna di tali relazioni). Ciascuna delle due operazioni può essere eseguita in modo molto più efficiente se sui campi interessati è definito un indice, che permette un accesso diretto. Poiché quasi sempre la chiave di una relazione è coinvolta in operazioni di selezione o di join (o di entrambe), è ragionevole definire, su ciascuna relazione, un indice in corrispondenza della relativa chiave; la maggior parte dei DBMS (PostGIS incluso) costruisce questo indice automaticamente.

Inoltre, possono essere definiti ulteriori indici su altri campi su cui vengono effettuate operazioni di selezione oppure su cui è richiesto un ordinamento in uscita, un indice, infatti, ordina logicamente i record di un file, rendendo nullo il costo di un ordinamento. Questo accorgimento può risultare molto utile nel caso delle tabelle "personaggio", "luogo" e "luogo attuale", per le quali è ragionevole un ordinamento dei record in ordine alfabetico sui rispettivi campi "nome_funz_lat" (nome latino del funzionario), "toponimo_lat" (nome latino del toponimo) e "toponimo_ita" (nome del toponimo attuale).

1.10 POPOLAMENTO DEL DATABASE STORICO

Col termine "popolamento del database" si intende l'attività di inserimento dei dati. Normalmente la memorizzazione di una singola informazione corrisponde all'inserimento di una o più righe in una o più tabelle del database. L'ordine delle operazioni non è puramente casuale, infatti l'inserimento delle righe deve essere fatto in modo da rispettare i vincoli d'integrità referenziale imposti sulle tabelle. Ad

esempio, poiché non può esistere una chiave esterna senza che sia stata prima inserita la riga a cui essa si riferisce, prima di poter inserire una riga nella tabella in cui è presente una chiave esterna dovrà essere stata inserita la corrispondente riga nella tabella a cui la chiave esterna punta. Nel caso in cui un vincolo venga violato, il DBMS impedirà l'operazione di inserimento facendola fallire (Agosti et al., 1997).

Le operazioni di popolamento della base di dati storici sono state automatizzate tramite degli *script* in linguaggio SQL. In tali script è possibile individuare due passaggi fondamentali.

- Copia dei dati da file di testo in formato csv nelle tabelle del database. I file csv sono organizzati secondo la struttura delle tabelle create nel database e delimitati dal carattere “;”

```
COPY personaggio FROM
'D:\\condivisa_tesi\\db_storico\\personaggio.
csv'
WITH DELIMITER AS ';'
NULL AS '<Null>';
```

- Creazione delle geometrie spaziali e definizione del Sistema di Riferimento e di Coordinate mediante l'uso delle funzioni di PostGIS. A tutte le tabelle che prevedono l'archiviazione delle coordinate è stato aggiunto un nuovo campo “geom” contenente la geometria puntuale, nel sistema di riferimento WGS84, UTM fuso 33 (codice SRID¹⁷ = 32633)

```
SELECT AddGeometryColumn
('luogo_attuale', 'geom', 32633, 'POINT', 2);
UPDATE luogo_attuale
SET geom = ST_SetSRID(ST_Makepoint(est_x,
nord_y), 32633);
```

¹⁷ Il codice SRID (Spatial Reference System Identifier) identifica univocamente le definizioni dei sistemi di coordinate. I codici dell'European Petroleum Survey Group (EPSG), dal 2005 gestiti dall'associazione internazionale Oil & Gas Producers (OGP) Surveying & Positioning Committee, sono molto diffusi e considerati standard di riferimento.

Infine sono state importate nella banca dati dei dati storici tre strati informativi vettoriali in formato *shapefile*, relativi ai confini amministrativi comunali, provinciali e regionali dell'anno 2011. Questi dati, originariamente nel sistema ED50, sono stati trasformati in WGS84, UTM fuso 33. Eventuali errori di trasformazione risultano trascurabili ai nostri scopi; si ricordi, infatti, che i dati utilizzati in questa applicazione sono stati georeferenziati in base alle informazioni estratte dai documenti storici, che non ci consentono una georeferenziazione di precisione.

2.PROGETTAZIONE DEL GEOSERVIZIO “GEOGRAFIE DELL’ITALIA MEDIEVALE (XIII – XV SEC.)”

Nel capitolo precedente è stata descritta la progettazione della base di dati che consentirà la gestione e consultazione dei dati storici provenienti dai *quaderni* di Francesco Agello. Insieme alla qualità e completezza della base di dati, è altrettanto importante che la navigazione e interrogazione dei dati sia per gli utenti semplice e intuitiva (Nielsen, 2000; Krug, 2006). Poiché gli strumenti di amministrazione di una base di dati hanno come target di riferimento utenti molto esperti, per facilitare le operazioni a tutti i partecipanti al progetto “*Geografia e linguaggi politici alla fine del Medioevo*” (PRIN 2006), si è deciso di realizzare un’interfaccia grafica utente (nota anche come GUI, dall’inglese *Graphical User Interface*) fruibile da web. In questo capitolo sarà descritta la fase di progettazione dell’architettura *client – server* e dell’interfaccia dell’applicazione web che prende il nome di “*Geografie dell’Italia Medievale (XII – XV secolo)*”. L’interfaccia grafica utente è stata realizzata con tecnologie tipiche del web che garantiscono accessibilità indipendentemente dalle caratteristiche dei PC degli utenti. Inoltre, poiché alcuni dei dati storici presentano una componente spaziale che ben si presta a essere rappresentata su un supporto cartografico, nell’applicazione web sono stati integrati strumenti tipici dei Sistemi Informativi Geografici.

2.1 UN'APPLICAZIONE WEB PER LA CONSULTAZIONE DELLA BASE DI DATI STORICI

L'applicazione web o geoservizio “*Geografie dell'Italia Medievale (XII – XV secolo)*” dovrà adempiere ai seguenti compiti:

- permettere la visualizzazione dei geodati su un supporto cartografico mediante funzionalità di *zoom* e *pan*;
- conoscere la posizione (coordinate) e il significato degli oggetti visualizzati (legenda);
- consentire di mostrare o nascondere i vari strati informativi (*layers*);
- interrogare la mappa ed il database associato; l'interrogazione può essere effettuata su un singolo strato o su più strati informativi contemporaneamente;
- agevolare la ricerca di informazioni d'interesse tramite appositi riquadri di ricerca.

L'applicazione web in questione avrà due modalità di consultazione dei dati: la modalità “WebGIS” e la modalità “Mostra Tabella”, da cui sarà possibile selezionare la tabella del database che si vuole visualizzare. Per la realizzazione dell'applicazione sarà necessario appoggiarsi a tecnologie diverse, alcune lavoreranno sul lato *client*, ovvero sui *browser* dei diversi utenti, altre invece si appoggeranno al lato *server*, la macchina su cui sarà installato il sistema informativo realizzato in questo lavoro.

2.2 ARCHITETTURA DI UN'APPLICAZIONE WEB

Con la definizione *Web Application* si intende un'applicazione residente in un *Server Web* alla quale si accede tramite un programma *client* che può essere un *browser* internet o un altro programma con funzioni di navigazione operante secondo gli standard del *World Wide Web*.

Il termine *client-server* si riferisce normalmente all'architettura di una applicazione *software* costituita da due programmi distinti e fra loro indipendenti (Fu e Sun, 2011):

- un programma *server*, che fornisce una serie di servizi;
- un programma *client*, che richiede ed utilizza tali servizi.

Un programma *client* consiste fondamentalmente in una "interfaccia utente" che accetta le richieste di un utilizzatore umano, ne verifica la correttezza e compone un "messaggio" di richiesta al *server*, attendendo adeguata risposta.

Quando il *server* risponde, il *client* può convertire il messaggio di risposta nella forma più opportuna affinché possa essere compresa dall'utilizzatore.

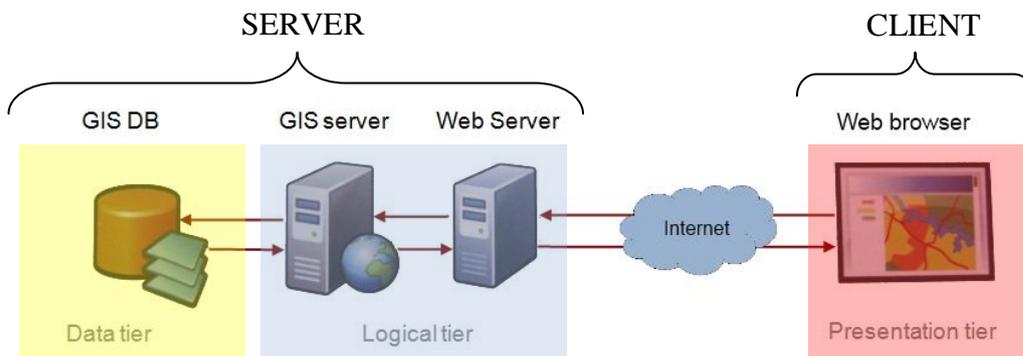


Figura 2.1 – Componenti dell'architettura base di un'applicazione web.

Alcuni dei principali vantaggi derivano dal fatto che *client* e *server* hanno funzioni specifiche radicalmente differenti e sono in larga misura indipendenti fra loro: l'unica cosa che li "lega" è l'insieme di messaggi che possono scambiarsi. Questo comporta che parte *client* e parte *server* possono essere sviluppate separatamente ed ottimizzate successivamente in maniera indipendente. Inoltre, è possibile sostituire i *client* o i *server* con programmi di tipo diverso, purché accettino lo stesso "linguaggio" di messaggi (protocolli).

In un'applicazione web tradizionale ad ogni modifica/interazione da parte dell'utente corrispondeva un caricamento della pagina: poiché molto codice HTML (*HyperText Markup Language*) della prima pagina corrispondeva a quello della pagina appena ricaricata, vi era uno spreco di banda oltre a un aumento dei tempi di attesa da parte

dell'utente. Per superare questi problemi negli ultimi venti anni si è sviluppata una nuova tecnica che migliora lo scambio di informazioni tra lato *server* e lato *client* che prende il nome di AJAX (acronimo di *Asynchronous JavaScript and XML*). AJAX è una tecnica di sviluppo che coinvolge più tecnologie per la realizzazione di applicazioni web interattive (*Rich Internet Application*). Lo sviluppo di applicazioni HTML con AJAX si basa su uno scambio di dati in *background* fra *web browser* e *server*, che consente l'aggiornamento dinamico di una pagina web senza esplicito ricaricamento da parte dell'utente.

AJAX è asincrono nel senso che i dati extra sono richiesti al *server* e caricati in *background* senza interferire con il comportamento della pagina esistente. Normalmente le funzioni richiamate sono scritte con il linguaggio JavaScript, che sarà presentato nel paragrafo 2.6. Tuttavia, e a dispetto del nome, l'uso di JavaScript e di XML (*eXtensible Markup Language*) non è obbligatorio, come non è necessario che le richieste di caricamento debbano essere necessariamente asincrone. Il rovescio della medaglia è che in questo modo si perde la linearità tra richiesta dell'utente e risposta del server e il risultato è più difficile da gestire sia per lo sviluppatore che per l'utente. Infatti, i primi potrebbero trovarsi in difficoltà qualora l'operazione asincrona dovesse attendere necessariamente una risposta al fine di completare una serie di operazioni più o meno sensibili mentre i secondi potrebbero non avere idea di cosa stia accadendo alla pagina e quindi potrebbero chiuderla ignari di aver richiesto un'informazione.

AJAX è una tecnica multi-piattaforma utilizzabile su molti sistemi operativi, architetture informatiche e *browser* web. Le applicazioni web che usano AJAX richiedono *browser* moderni quali: Firefox, Opera, Konqueror, Safari, Internet Explorer e Chrome.

L'architettura *client-server* si presta naturalmente ad applicazioni come quella che vogliamo realizzare, in cui si prevede che un gruppo di utenti accedano ai servizi di consultazione della banca dati.

n questa applicazione web è stato fatto largo uso della tecnica AJAX per consentire l'interazione tra l'interfaccia utente e il *server* che ospita il database: tutte le richieste alla banca dati vengono trasformate in chiamate AJAX, secondo il flusso di operazioni descritto in Figura 2.2.

- L'utente esegue una richiesta al database mediante l'interfaccia grafica utente (richiesta di visualizzazione di dati, ricerche, ecc). Il codice JavaScript interpreta la richiesta e compone la chiamata AJAX con i parametri da inviare al *server*.
- Lo script in linguaggio *server* PHP (che sarà presentato nel paragrafo 2.3) interpreta i parametri ricevuti dal *client* via AJAX, compone la *query* SQL e invia l'interrogazione al database.
- La risposta del database viene inviata allo script PHP, che si incarica di formattare i dati in un linguaggio di *markup*, nel nostro caso in formato JSON.
- La risposta del *server* viene interpretata da JavaScript che la rappresenta nell'interfaccia della *Web Application*, predisposta per ricevere il messaggio in un formato dati specifico (XML o JSON).

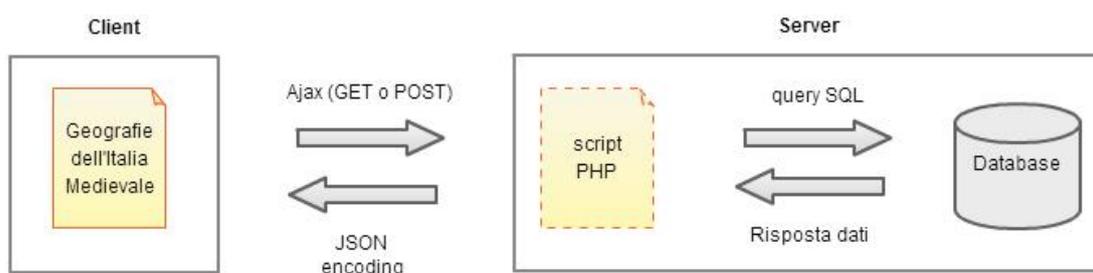


Figura 2.2 – Schema di funzionamento delle chiamate AJAX nell'applicazione web "Geografie dell'Italia Medievale".

2.3 SCELTE DI PROGETTAZIONE DEL “LATO SERVER” DEL GEOSERVIZIO

In questa fase si valutano i vari programmi che consentono l'esecuzione di *script* sul lato *server*: i processi si svolgeranno sul *server* e saranno indipendenti dall'ambiente di esecuzione dell'utente (*browser*, sistema operativo e impostazioni e preferenze). Questi programmi comprendono: il Sistema Operativo, il DBMS, il Web Service e il GIS Server.

Come già detto nel paragrafo 2.3, per questo progetto si è deciso di fare uso di tecnologie esclusivamente *free* e *open source*, per evitare di impegnare gli enti di ricerca partecipanti all'acquisto di licenze e a ulteriori spese di manutenzione del software. Coerentemente con questa posizione si è optato per un Sistema Operativo gratuito e a codice aperto che al tempo stesso garantisca da una parte piena compatibilità con gli strumenti scelti finora e dall'altra facilità d'uso e manutenzione. Per questi motivi la scelta individuata è Ubuntu, nella sua versione LTS (*Long Term Support* ossia supporto a lungo termine)10.4.

Ubuntu è un sistema operativo GNU/Linux nato nel 2004, basato su Debian, che si focalizza sull'utente e sulla facilità d'utilizzo. È prevista una nuova versione ogni sei mesi. Questo sistema è pubblicato come *software* libero sotto licenza GNU GPL ed è quindi gratuito e liberamente modificabile. Ogni versione ha un supporto ufficiale di 18 mesi con *patch* di sicurezza e aggiornamenti, ad eccezione delle versioni LTS che hanno un periodo di supporto esteso. Queste versioni hanno un periodo di supporto di 5 anni sia per i *desktop* che per i *server* (precedentemente alla versione 12.04 LTS il supporto per la versione *desktop* era di soli 3 anni). L'aggiornamento viene supportato da una versione LTS all'altra. È prevista la pubblicazione di una nuova versione LTS ogni due anni.

Al momento in cui si è iniziato a lavorare su questo progetto la versione LTS più recente era la 10.04, nel prossimo futuro si prevede un passaggio alla versione LTS più recente, la 12.04.

Per quanto riguarda il DBMS, ovvero il software di gestione della base di dati, è stato scelto PostgreSQL integrato dall'estensione per la gestione dei dati spaziali PostGIS (come già spiegato al paragrafo 1.8), soluzione pienamente compatibile con il Sistema Operativo Ubuntu 10.04. In particolare, per questo progetto si è usato PostgreSQL 8.4 con l'estensione PostGIS 2.0.

La pubblicazione dei dati è stata affidata a QGIS Server, un *server WMS (Web Map Service)* versione 1.3.0 *open source* che deriva da Quantum GIS, l'applicazione *desktop*, e ne utilizza le librerie. QGIS Server consente di creare mappe WMS in cui le informazioni geografiche sono restituite esattamente come nell'applicazione *desktop*, permettendo così di configurare stili anche molto complessi in modo semplice e grafico (WYSIWYG, *What-You-See-Is-What-You-Get*), senza bisogno di editare *file* di configurazione. Infatti, mappe e template di stampa possono essere creati in QGIS *desktop* e pubblicati utilizzando il modulo FastCGI¹⁸ del *webserver* Apache.

Il progetto è in sviluppo da qualche anno e si fa apprezzare per la semplicità e rapidità con cui si possono realizzare servizi OGC; le prestazioni non sono paragonabili ad applicativi quali UMN MapServer o GeoServer, ma il tutto può venir superato se si utilizza un sistema di *cache*¹⁹ (come ad esempio MapProxy) tra il *client* ed il *server* pre-generando le *tile* se i servizi sono consultabili essenzialmente tramite interfaccia

¹⁸ Il modulo FastCGI è la versione più recente di *Common Gateway Interface*, una tecnologia standard usata dai *webserver* per interfacciarsi con applicazioni esterne. Ogni volta che un *client* richiede al *webserver* un URL corrispondente ad un programma CGI, il server lo esegue in tempo reale, generando dinamicamente informazioni.

¹⁹ Un servizio di mappa può essere ottimizzato pre-generando immagini statiche o piastrelle (*tile*) a diversi livelli di scala. La cartella che contiene le immagini costituisce la *cache*. Questa tecnica è molto utilizzata nei servizi di mappa consultabili da web perché la restituzione di una piastrella richiede molto meno tempo che generare l'immagine della mappa su richiesta.

web. QGIS Server è stato preferito a MapServer e GeoServer anche perché consente una manutenzione più semplice e intuitiva direttamente dall'applicazione *desktop*.

Il linguaggio di programmazione lato *server* che verrà utilizzato per fare le interrogazioni al database è PHP, un linguaggio interpretato, con licenza *open source*, originariamente concepito per la realizzazione di pagine web dinamiche.

L'elaborazione di codice PHP sul *server* produce codice in un linguaggio di *markup* (XML, HTML, JSON) da inviare al *browser* dell'utente che ne fa richiesta.

Per questo lavoro sarà predisposto sul lato server un controllo di accesso all'applicazione web che permetterà solo agli utenti autorizzati di visualizzare le informazioni. Tale controllo rimarrà attivo finché il servizio non sarà reso pubblico nella sua versione definitiva.

2.4 PROGETTAZIONE DEL “LATO CLIENT” (INTERFACCIA UTENTE DEL GEOSERVIZIO)

In questo contesto, l'interfaccia utente riveste un ruolo fondamentale. Essa ha il compito di “filtrare” la complessità, presentando all'utente un'immagine semplificata del servizio, e congruente con i compiti che deve svolgere. Una buona interfaccia non solo nasconde all'utilizzatore la complessità interna del sistema, ma ne riduce la complessità funzionale, mettendo a disposizione dell'utente funzioni di più alto livello, in grado di effettuare compiti complessi con un grado di automatismo maggiore. Ciò viene realizzato integrando numerose funzionalità semplici in funzionalità più potenti, con il risultato di semplificare il dialogo fra l'utente e il sistema (Polillo, 2010).

L'efficacia dell'interfaccia grafica risulta quindi determinante per la validità e utilità complessiva del progetto. L'interfaccia, infatti, rappresenta il punto di accesso privilegiato al database: ogni errore nella progettazione dell'interfaccia si rifletterà negativamente sull'esperienza d'uso dell'utente, causando rallentamenti e frustrazioni

che influiranno in maniera negativa sui risultati che si intendono ottenere. Risulta pertanto necessario definire con estrema cura le caratteristiche generali dell'interfaccia utente. Il percorso delineato prevede una serie di fasi, alcune delle quali da effettuarsi contemporaneamente, e un'analisi dei risultati ottenuti, nonché dei problemi incontrati e delle possibili soluzioni da implementare, al termine di ciascuna fase. Più in dettaglio:

- definizione degli **obiettivi**: si delineano le principali funzionalità d'uso della *web application* in base alla tipologia di utenti. Si rimanda al paragrafo 2.4.1 per approfondimenti;
- definizione delle **caratteristiche generali** dell'interfaccia utente: in base alle informazioni raccolte in precedenza e alle caratteristiche dei dati da visualizzare e manipolare viene disegnata l'interfaccia utente nelle sue caratteristiche generali. Questa fase verrà descritta in dettaglio nel paragrafo 2.4.2;
- definizione degli **strumenti software** che verranno impiegati: in questa fase si stabiliscono le piattaforme software su cui sarà possibile utilizzare la *web application* (ovvero quali *browser* verranno supportati) e gli strumenti di sviluppo da impiegare per la programmazione dell'interfaccia utente. Questa fase verrà descritta in dettaglio nel paragrafo 2.4.3;
- sviluppo e collaudo di una **versione prototipale**: viene implementata una prima versione dell'applicazione prevalentemente orientata alla consultazione del database. Questa fase verrà descritta in dettaglio nel Capitolo 3;
- sviluppo della **versione definitiva**: sulla base delle esperienze maturate grazie alla versione sperimentale, e con il contributo dei ricercatori coinvolti nel progetto, viene sviluppata la versione definitiva. Questa voce non sarà approfondita in questa Tesi e costituisce uno degli sviluppi futuri di questo lavoro.

2.4.1 FASE 1 – DEFINIZIONE DEGLI OBIETTIVI DELLA APLICAZIONE WEB

L'interfaccia grafica utente dovrà agevolare le operazioni di manutenzione, aggiornamento e consultazione della banca dati. Questo implica che saranno predisposte delle funzionalità che consentiranno la modifica (inserimento e/o eliminazione di nuovi dati) e l'interrogazione dei dati già esistenti. La versione prototipale comprenderà solo le funzioni di consultazione dei dati secondo due modalità:

- modalità “WebGIS”, da cui sarà possibile visualizzare i dati su una base cartografica, fare operazioni di zoom e pan, consultare la legenda dei diversi strati informativi, interrogare i dati direttamente in mappa oppure mediante appositi box;
- modalità “Mostra Tabella”, che consentirà la consultazione della componente prettamente alfanumerica; sarà possibile visualizzare tabelle e viste predisposte per la consultazione dei dati, fare delle ricerche mediante un apposito *query builder*, ovvero uno strumento che consente di creare delle interrogazioni selezionando i campi d'interesse e concatenando più criteri di ricerca.

2.4.2 FASE 2 – DEFINIZIONE DELLE CARATTERISTICHE GENERALI DELL'INTERFACCIA UTENTE

In base alle informazioni raccolte nella fase precedente è stato creato lo schema di funzionamento dell'interfaccia grafica utente, schematizzato in Figura 2.4.

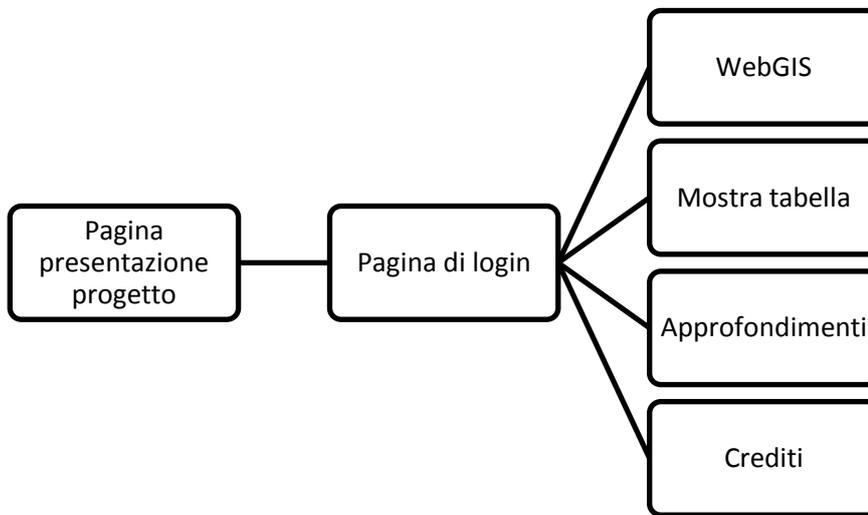


Figura 2.3 – Mappa del sito che consentirà accesso e navigazione al database.

Come si può osservare la navigazione parte da una pagina web di presentazione del progetto; tale pagina è pubblica ovvero accessibile a tutti i visitatori e oltre a contenere una breve descrizione degli obiettivi del progetto, i principali attori e altre informazioni di approfondimento permette l’accesso ai dati del database, mediante un’apposita pagina dedicata alla *Login*. I dati possono essere visualizzati e consultati da due *tab* o schede, una per la modalità “WebGIS”, la seconda per la modalità “Mostra Tabella” (vedi paragrafo 2.4.1). Nella fase prototipale sono previste altre due schede, la prima raccoglierà approfondimenti sui dati, quali metadati che potrebbero essere utili per gli storici nelle loro analisi e bibliografia di riferimento, mentre la seconda sarà dedicata ai *Crediti*, ovvero l’elenco di tutte le Unità di Ricerca coinvolte con relativi URL alle pagine ufficiali, l’elenco dei partecipanti con corrispondenti informazioni di contatto e i *banner* degli Enti che hanno finanziato la ricerca.

In questa fase risulta molto utile creare delle anteprime grafiche del sito, che nel linguaggio del *web design* prendono il nome di *mockup*. I *mockup* realizzati per le schede WebGIS e “Mostra Tabella” sono riportati in Figura 2.5 e 2.6.

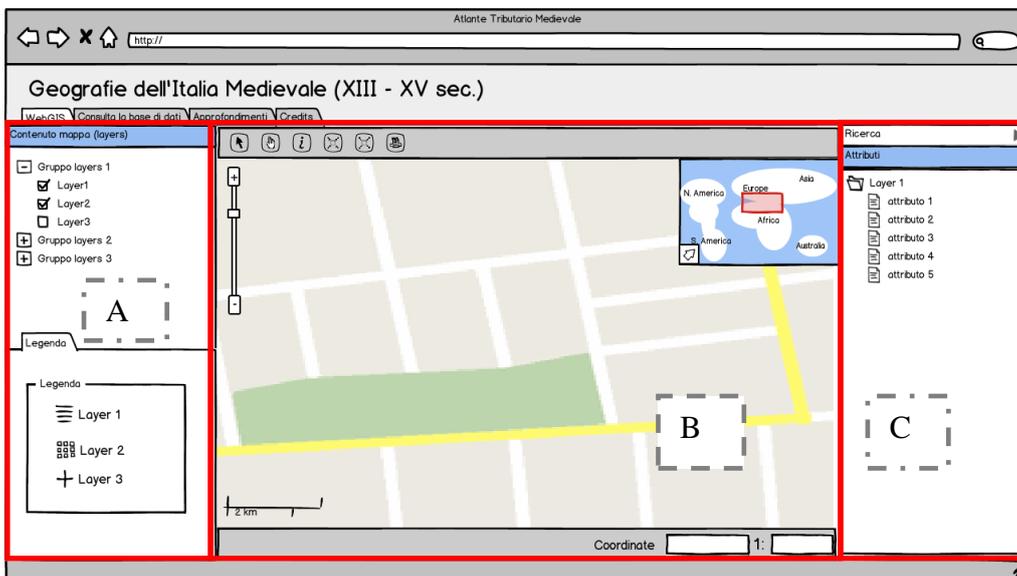


Figura 2.4 – Mockup della scheda “WebGIS”. L’area di lavoro può essere suddivisa in tre parti, una a sinistra (A), una centrale (B) e una a destra (C).

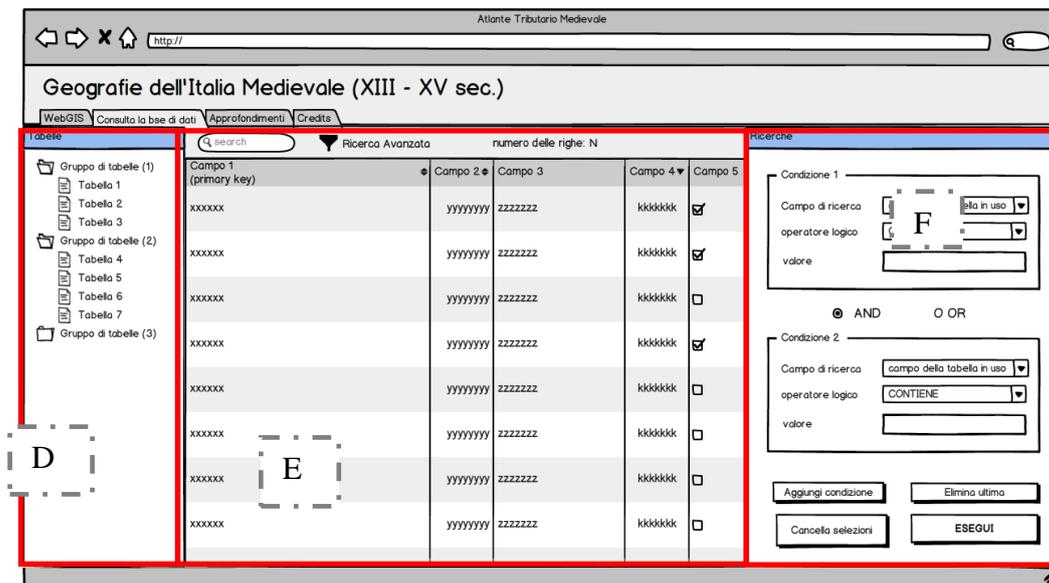


Figura 2.5 – Mockup della scheda “Mostra tabella”. Anche in questo caso l’area di lavoro può essere suddivisa in tre parti (D, E ed F).

Il mockup è una bozza iniziale dell’interfaccia e del layout di un progetto, serve non solo per porre le fondamenta per la creazione di una vera e propria interfaccia grafica,

ma è anche lo strumento più immediato ed efficace per far capire ai vari collaboratori come funzionerà l'applicazione e definire l'interfaccia grafica utente in maniera approssimata. Il vantaggio principale è quello di ottenere un riscontro da parte dei vari utenti finali, che consenta eventualmente di perfezionare esigenze specifiche di progetto, migliorare l'usabilità di un prodotto o persino valutare linee di progettazione alternative. I *mockup* risultano particolarmente importanti nella fase di progettazione delle componenti del sito per le quali modifiche a uno stato avanzato dei lavori può risultare molto difficile e oneroso in termini di tempo, risorse e prestazioni.

Come si può osservare dalle Figure 2.5 e 2.6 la pagina web presenta una sezione in alto chiamata *header* che riporta il nome dell'applicazione, e il resto della pagina è suddivisa in *tab* o schede che a loro volta possono essere suddivise in tre parti. L'area a sinistra è dedicata alla selezione dei dati siano essi strati cartografici come nella scheda "WebGIS" (A) o tabelle/viste come nella scheda "Mostra Tabella" (D). L'area centrale presenta una barra degli strumenti in alto che raccoglie alcuni semplici strumenti di navigazione e ricerca e nell'area restante contiene la mappa nella modalità "WebGIS" (B) o la tabella/vista selezionata (E). Infine l'area a destra contiene gli strumenti per le ricerche avanzate e mostra i risultati ottenuti (C e F).

I *mockup* delle Figure 2.5 e 2.6 sono stati realizzati con Balsamiq Mockup, un programma che consente di creare schizzi in stile *hand made* (come se fossero realmente disegnati a mano) con interfacce anche complesse. Il programma, che è disponibile su tutti i sistemi operativi (iOS, Linux e Windows), è basato sulle azioni di *drag and drop* e si trova in due versioni: una demo online gratuita e una versione a pagamento in cui è possibile estendere il numero e tipo di controlli della libreria di base con altri disponibili gratuitamente sul sito.

2.4.3 FASE 3 – DEFINIZIONE DELLE TECNOLOGIE PER LA REALIZZAZIONE DELL'INTERFACCIA WEB

L'interfaccia grafica utente per la *web application* "Geografie dell'Italia Medievale (XIII – XV sec.)" definita nelle sue caratteristiche generali nel paragrafo precedente

avrà il compito di semplificare le operazioni di gestione e consultazione del database da parte degli storici. Per garantire la maggior diffusione sulle diverse piattaforme OS e sulla maggior parte dei *browser*, si è deciso di realizzare un'applicazione web senza utilizzo di *plugin* aggiuntivi quali Flash o Silverlight. La scelta è ricaduta su tre linguaggi di programmazione web che insieme si stanno consolidando come il *kit* più diffuso in internet per realizzare pagine web (Figura 2.7): HTML, Javascript e CSS. Tale *kit* di sviluppo prende il nome di HTML5.



–

Figura 2.6 – I principali linguaggi di sviluppo web utilizzati per la realizzazione dell'applicazione web progettata.

Infatti, un documento web può essere scomposto in tre componenti distinte e separate:

- strato contenuto (HTML);
- strato presentazione (CSS);
- strato comportamento (JavaScript e librerie).

STRATO CONTENUTO - HTML

Il contenuto è l'informazione che si vuole trasmettere all'utente. Comprende testo, immagini, suoni, video o altro. L'informazione è descritta strutturalmente e semanticamente attraverso il linguaggio HTML (*HyperText Markup Language*) che non è un vero e proprio linguaggio di programmazione in quanto non prevede alcuna definizione di variabili, strutture dati, funzioni, strutture di controllo ma solamente un'alberatura di *markup* che descrive le modalità di impaginazione o visualizzazione grafica (*layout*) del contenuto, testuale e non, di una pagina web attraverso *tag* di formattazione. Tuttavia, l'HTML supporta l'inserimento di *script* e oggetti esterni quali immagini o filmati. Punto HTML (.html) o punto HTM (.htm) sono le estensioni comuni per riconoscere i documenti in questo formato (Pilgrim, 2010).

Un'ulteriore ed importante caratteristica di HTML è che esso è stato concepito per definire il contenuto logico e non l'aspetto finale del documento. I dispositivi e i *browser* che possono accedere ad un documento HTML sono molteplici e non sempre dotati di potenti capacità grafiche. Proprio per questo gli sviluppatori di HTML hanno optato per un linguaggio che descrivesse dal punto di vista logico, piuttosto che grafico, il contenuto dei documenti. Questo significa che non esiste alcuna garanzia che uno stesso documento venga visualizzato in egual modo su due dispositivi. Se da una parte questo ha imposto in passato dei forti limiti agli sviluppatori di pagine Web, ha dall'altro garantito la massima diffusione di Internet ed evitato che essa diventasse un medium di élite.

Attualmente i documenti HTML sono in grado di incorporare molte tecnologie, che offrono la possibilità di aggiungere al documento ipertestuale controlli più sofisticati sulla resa grafica, interazioni dinamiche con l'utente, animazioni interattive e contenuti multimediali. Si tratta di linguaggi come CSS, JavaScript, XML, JSON, o di altre applicazioni multimediali di animazione vettoriale o di streaming audio o video.

È consentito incorporare codice JavaScript e codice CSS in un file HTML, tuttavia è consigliabile tenere separati i tre strati di contenuto, presentazione e comportamento in file distinti.

STRATO PRESENTAZIONE – CSS

Lo strato presentazione definisce l'aspetto e la resa grafica dei contenuti che possono apparire in modo diverso su differenti dispositivi (Meyer, 2011). Ad esempio per una stessa pagina web possono essere predisposti diversi *template* di visualizzazione a seconda che l'utente acceda alla pagina da uno *smartphone*, *tablet* o dal proprio *laptop*. L'aspetto dei contenuti è definito attraverso l'uso dei fogli di stile CSS (Cascading Style Sheets) che permettono di specificare, attraverso i selettori, come visualizzare/presentare gli elementi del contenuto.

L'introduzione del CSS si è resa necessaria per separare i contenuti dalla formattazione e permettere una programmazione più chiara e facile da utilizzare, sia per gli autori delle pagine HTML che per gli utenti. L'HTML infatti, presenta dei limiti sul lato della pura presentazione, non essendo nato a questo particolare scopo ma è stato piegato a fare cose che intrinsecamente non era in grado di fare. Le lacune del linguaggio HTML sono coperte dai fogli CSS, ad esempio è possibile dare al testo delle pagine web un aspetto da *word-processor*: non solo con il colore o i font, ma con un sistema di interlinea pratico e funzionale, riuscendo a spaziare lettere e parole, impostando stili diversi per titoli e paragrafi, sfruttando i benefici dell'indentatura o della giustificazione.

Così come è possibile distanziare gli elementi della pagina in maniera semplice e intuitiva con un potente meccanismo di gestione dei margini o ancora aggiungere uno sfondo e collocarlo nella posizione desiderata (operazione particolarmente difficoltosa con HTML).

In generale, la manutenzione di un sito che prevede una gestione separata della formattazione in fogli di stile risulta molto semplificata: con poche e mirate

modifiche al codice CSS è possibile cambiare contemporaneamente la formattazione in più parti della pagina web.

STRATO COMPORTAMENTO – JAVASCRIPT

Il comportamento ovvero l'interazione con l'utente di un sito internet avviene mediante il linguaggio di *scripting* JavaScript. Con interazione si intende ad esempio la semplice validazione dei dati immessi dall'utente in un *form* oppure anche la gestione di un'interfaccia simile alle applicazioni *standalone*.

La caratteristica principale di JavaScript è quella di essere un linguaggio interpretato: il codice non viene compilato, ma interpretato dall'interprete che è incluso nel *browser* che si sta utilizzando. Il linguaggio definisce le funzionalità tipiche dei linguaggi di programmazione ad alto livello (strutture di controllo, cicli, ecc.) e consente l'utilizzo del paradigma orientato agli oggetti.

Inoltre in JavaScript il codice viene eseguito direttamente sul *client* e non sul *server*. Il vantaggio di questo approccio è che, anche con la presenza di *script* particolarmente complessi, il *server* non viene sovraccaricato a causa delle richieste dei *client*. Di contro, nel caso di *script* che presentino un sorgente particolarmente grande, il tempo per lo scaricamento può diventare abbastanza lungo. Un altro svantaggio è il seguente: ogni informazione che presuppone un accesso a dati memorizzati in un database remoto deve essere rimandata ad un linguaggio *server* (PHP, Java, Python, ecc) che effettui esplicitamente la transazione, per poi restituire i risultati ad una o più variabili JavaScript; operazioni del genere richiedono il caricamento della pagina stessa. Con l'avvento di AJAX tutti questi limiti sono stati superati (paragrafo 2.2).

A differenza di altri linguaggi, quali il C o il Java, che permettono la scrittura di programmi completamente *standalone*, JavaScript viene utilizzato soprattutto in quanto linguaggio di *scripting*, integrato, quindi, all'interno di un altro programma. L'idea di base è che il programma ospite (quello che ospita ed esegue lo *script*) fornisca allo *script* un'API (*Application Program Interface*) ben definita, che consente

l'accesso ad operazioni specifiche, la cui implementazione è a carico del programma ospite stesso. Lo *script*, quando eseguito, utilizza riferimenti a questa API per richiedere (al programma ospite) l'esecuzione di operazioni specifiche, non previste dai costrutti del linguaggio JavaScript in sé. Un *browser* tipicamente incorpora un interprete JavaScript; quando viene visitata una pagina web che contiene il codice di uno *script* JavaScript, quest'ultimo viene portato in memoria ed eseguito dall'interprete contenuto nel *browser*.

Le interfacce che consentono a JavaScript di rapportarsi con un *browser* sono chiamate DOM (*Document Object Model*). Molti siti web usano la tecnologia JavaScript lato *client* per creare potenti applicazioni web dinamiche.

Un uso principale del Javascript in ambito Web è la scrittura di piccole funzioni integrate nelle pagine HTML che interagiscono con il DOM del *browser* per compiere determinate azioni non possibili con il solo HTML statico, come controllare i valori nei campi di input, nascondere o visualizzare determinati elementi, ecc.

In questo strato è utile l'uso di librerie JavaScript che semplificano il lavoro: ExtJS ha consentito di integrare elementi come griglie, bottoni, barre di strumenti, all'interfaccia grafica utente, mentre OpenLayers ha permesso l'inclusione di funzionalità di *web mapping* (Hazard, 2011).

ExtJS è un *framework* di sviluppo JavaScript utile a realizzare interfacce utente (GUI) per applicazioni web rilasciato dalla società Sencha Inc. ExtJS, è disponibile con doppia licenza: una versione *free* con licenza GPL3 ed una versione commerciale con licenza proprietaria. Negli ultimi anni (tra il 2007 ed il 2011) il *framework* ha subito una rapida evoluzione: nell'ultima versione ExtJS 4.0. sono stati introdotti numerosi miglioramenti rispetto alle versioni precedenti relativamente alla struttura delle classi e riorganizzato il *framework* in modo da sposare completamente il paradigma di programmazione MVC (*model-view-controller*)²⁰. Nonostante questa strutturazione

²⁰ Model-View-Controller (MVC, talvolta tradotto in italiano Modello-Vista-Controllo) è un pattern architetturale molto diffuso nello sviluppo di interfacce grafiche di sistemi *software object-oriented*. Il

del codice richieda un lavoro progettuale più impegnativo i vantaggi sono numerosi: il modello, infatti, facilita il lavoro di manutenzione e la comprensione del codice anche da parte di altri programmatori e inoltre è garantita più flessibilità (ad esempio, nel caso di modifiche al database l'applicazione web può essere adattata modificando opportunamente la componente modello e non tutto il codice).

In questo lavoro è stata utilizzata la più recente versione rappresentata da ExtJS 4.1, che è stata ufficialmente rilasciata il 24 Aprile 2012.

L'utilizzo di ExtJS presenta numerosi aspetti positivi, in primo luogo è possibile creare applicazioni web con *layout* molto avanzati anche senza avere approfondite conoscenze in ambito CSS, ma soprattutto è possibile ottenere questo risultato in un tempo molto limitato e scrivendo pochissime righe di codice. In poche parole ExtJS permette di fare in poche righe di codice quello che utilizzando HTML, JavaScript e fogli di stile CSS si potrebbe ottenere solo con *script* molto grandi e ore di lavoro.

Inoltre ExtJS permette di utilizzare funzionalità AJAX per l'aggiornamento dell'interfaccia utente, agganciare l'applicazione web ad utili strumenti web come jQuery e Google Gears, creare grafici di vario tipo come istogrammi, grafici a torta e diagrammi di dispersione, dotati di un layout molto accattivante. Queste funzionalità permettono allo sviluppatore di creare rapidamente applicazioni web molto vicine alle applicazioni *desktop* con le quali siamo abituati a lavorare tutti i giorni. Inoltre è importante il fatto che le applicazioni realizzate utilizzando ExtJS sono realmente *cross-browser*; da diversi test risulta, infatti, che il *framework* sia compatibile con il 95% dei browser oggi disponibili (“Un’introduzione a ExtJS”, DTraining).

pattern è basato sulla separazione dei compiti fra i componenti software che interpretano tre ruoli principali:

- il *model* fornisce i metodi per accedere ai dati utili all'applicazione;
- il *view* visualizza i dati contenuti nel *model* e si occupa dell'interazione con utenti e agenti;
- il *controller* riceve i comandi dell'utente (in genere attraverso il *view*) e li attua modificando lo stato degli altri due componenti

Questo schema, fra l'altro, implica anche la tradizionale separazione fra la logica applicativa a carico del *controller* e del *model*, e l'interfaccia utente a carico del *view*.

Un ulteriore vantaggio dell'utilizzo di ExtJS è rappresentato dal sito di Sencha che, nella sezione dedicata ad ExtJS, offre una buona documentazione online in lingua inglese corredata da numerosi forum di discussione. La community di utilizzatori di ExtJS è vasta ed attiva e dal sito è possibile accedere (gratuitamente) a forum moderati da sviluppatori professionisti di Sencha. I forum sono molto frequentati, in genere il tempo di attesa per ricevere repliche ai propri *post* è di qualche ora, inoltre è frequente trovare vecchi *post* che possono essere di grande aiuto a chi è alle prime armi e si scontra con problemi comuni. Inoltre nella sezione “Sample and Demos” è possibile trovare numerosi esempi funzionanti delle varie funzionalità offerte dal *framework*. Per ogni esempio è possibile accedere al codice sorgente e questo può essere un ottimo punto di inizio per capire come utilizzare ExtJS.

I principali aspetti negativi legati ad ExtJS sono senz'altro la sua complessità e la sua pesantezza. Come tutti gli strumenti sofisticati infatti ExtJS non è molto immediato da utilizzare, nonostante si possano realizzare in breve tempo componenti quali *form* e *griglie* con un aspetto molto piacevole. Per arrivare a padroneggiare tutte le potenzialità che il *framework* offre è necessario molto lavoro (e una certa dose di frustrazione iniziale) anche secondo sviluppatori esperti che hanno già avuto esperienze con altri *framework* JavaScript. L'aspetto sicuramente più critico di ExtJS sono i tempi di caricamento che possono risultare abbastanza elevati. Il rilascio di ExtJS 4.1 ha parzialmente rimediato a questo problema, apportando un buon incremento delle prestazioni, anche se le prestazioni di altri prodotti con funzionalità simili (come ad esempio jQuery, il più diffuso *framework* JavaScript) rimangono distanti. In rete è possibile trovare numerose discussioni relative a questo argomento; è opinione diffusa tra gli sviluppatori che JQuery sia decisamente più leggero e “performante” di ExtJS, ma anche molto meno ricco di funzionalità. Idea abbastanza comune è che entrambi i prodotti siano molto validi, anche se adatti ad usi differenti: il maggior pregio di JQuery è la leggerezza, per questo rappresenta un'ottima soluzione quando si vuole aggiungere funzionalità a un sito esistente, al contrario per

un'applicazione web complessa ExtJS è senza dubbio l'ideale in quanto fornisce una libreria completa di componenti paragonabile a molte librerie di applicazioni *desktop*. OpenLayers è una libreria JavaScript di tipo *open source* con licenza derivata dalla licenza BSD²¹ per visualizzare mappe interattive nei browser web, essendo una libreria lato *client* non richiede *software* specifico, né particolari impostazioni lato *server*.

OpenLayers è multi piattaforma, ovvero è compatibile con la maggioranza dei *browser* e offre ai programmatori robuste API per poter accedere a diverse fonti d'informazioni cartografiche in Internet come: *Web Map Service* (WMS), mappe di tipo commerciale (Google Maps, Bing, Yahoo), *Web Feature Service* (WFS), diversi formati vettoriali, mappe del progetto OpenStreetMap, e altre funzionalità geospaziali. Inizialmente le librerie sono state sviluppate da MetaCarta nel giugno del 2006 come risposta al servizio Google Maps, diventando presto un framework molto diffuso e “maturo”, grazie alla collaborazione di un'appassionata e attiva comunità di sviluppatori. Da novembre 2007 questo progetto è entrato a fare parte dei progetti della *Open Source Geospatial Foundation*. Il sito ufficiale offre una discreta documentazione e una buona raccolta di esempi.

È possibile utilizzare il *framework* Openlayers in due modalità:

- richiamando le API online, disponibili dal sito di Openlayers,
- richiamando le API in locale, preventivamente scaricate dal sito ufficiale.

²¹ Le licenze BSD sono una famiglia di licenze permissive per software. Molte sono considerate libere ed *open source*. Il loro nome deriva dal fatto che la licenza BSD originale fu usata originariamente per distribuire il sistema operativo Unix Berkeley Software Distribution (BSD), una revisione libera di UNIX sviluppata presso l'Università di Berkeley. Le licenze BSD risultano più deboli di altre, ad esempio della licenza GNU General Public License (GNU GPL), nel garantire il perpetuarsi di tali libertà nel tempo non avendo fra i propri obiettivi quello di proteggere la libertà del software cui sono applicate ma semplicemente di rendere per tutti il software completamente libero, accessibile e modificabile. Chi infatti modifichi un programma protetto da licenze BSD, può ridistribuirlo usando la stessa o qualunque altra licenza (anche non libera), senza avere l'obbligo di ridistribuire le modifiche apportate al codice sorgente.

Le funzionalità che Openlayers mette a disposizione sono molteplici e soddisfano una vasta tipologia di utenti. Lo sviluppatore alle prime armi può contare su una libreria “matura”, molta documentazione a riguardo e sviluppatori esperti che si rendono disponibili nei forum a rispondere alle domande. D’altra parte lo sviluppatore esperto ha a disposizione strumenti molto potenti su cui costruire funzionalità avanzate e può personalizzare ogni componente della mappa: strati informativi, controlli, ecc. Inoltre, trattandosi di uno strumento open source, è possibile cambiare il codice sorgente adattandolo alle proprie esigenze senza preoccuparsi di verificare i termini di licenza. Openlayers, infatti, non è legata a nessuna tecnologia proprietaria.

Per la gestione delle informazioni spaziali lato client è stata considerata anche la libreria JavaScript Leaflet, sviluppata da CloudMade e da un attivo gruppo di collaboratori. Leaflet è un *framework* moderno e ideale per la realizzazione di mappe interattive, veloci e “leggere”, ottimizzate sui *browser* moderni e molto adatte alla loro consultazione da dispositivi *mobile*, quali *smartphones* e *tablets*.

Leaflet, tuttavia, non è una vera alternativa a Openlayers: infatti, questa libreria consente di integrare in maniera semplice le principali funzionalità di navigazione della mappa ma non copre le funzionalità più avanzate. Ad esempio, non sono agevolate le operazioni di *editing*, in quanto non sono disponibili strumenti tipo Draw Feature e l’Editing Toolbar. Per non limitare ulteriori sviluppi della applicazione web si è deciso di fare affidamento su una libreria molto più matura e completa, quale è OpenLayers, che rimane la libreria più utilizzata nello sviluppo di WebGIS.

Inoltre poiché Leaflet è un prodotto molto recente (la prima *release* risale al 13 maggio 2011), risulta più difficile trovare esempi su Leaflet che su Openlayers.

2.5 SINTESI DELLE TECNOLOGIE SCELTE PER LA REALIZZAZIONE DEL GEOSERVIZIO

A conclusione di questo capitolo si riassumono le scelte tecnologiche per la realizzazione del geoservizio. Nelle Tabella 2.1 e 2.2 sono riportati rispettivamente le tecnologie scelte per il lato *server* e per il lato *client*.

Tabella 2.1 – Riassunto delle tecnologie scelte per il lato server dell'applicazione web.

TIPOLOGIA SW	SOFTWARE SCELTO
Sistema Operativo	Ubuntu (Linux) 10.4
DBMS	PostgreSQL e PostGIS
Web Server	QGIS Server
Linguaggio sviluppo	PHP

Tabella 2.2 – Riepilogo delle tecnologie utilizzate per la realizzazione del "lato client" ovvero l'interfaccia del servizio.

		LIBRERIE / FRAMEWORK	
STRATO	LINGUAGGIO	NOME	FUNZIONALITA'
Contenuto	HTML		
Presentazione	CSS		
Comportamento	JavaScript	Openlayers	Web Mapping
		ExtJS	GUI tools

3 REALIZZAZIONE E PUBBLICAZIONE DEL GEOSERVIZIO DEI DATI STORICI

In questo Capitolo verranno approfonditi gli aspetti più rilevanti della realizzazione e sviluppo dell'applicazione web "Geografie dell'Italia Medievale (XIII – XV secolo). In particolare saranno descritte le interfacce grafiche utente e i principi di funzionamento degli strumenti realizzati.

In particolare verrà descritto il processo di configurazione e personalizzazione del client web di QGIS, applicativo che costituisce l'ossatura del WebGIS realizzato per l'applicazione in esame così come saranno presentate le diverse modalità di ricerca e interrogazione che possono essere sfruttate dal geoservizio.

Verrà presentato il processo di parametrizzazione dell'applicazione che ha molto semplificato la procedura di aggiornamento e integrazione di nuove tabelle e viste. Sono state realizzate due modalità di ricerca: la prima, definita "semplice", consente di cercare una stringa nei campi definiti "searchable" nella tabella/vista attiva, mentre la seconda permette di creare delle interrogazioni più complesse, concatenando diversi criteri di ricerca.

I due tipi di ricerca sono implementati in maniera differente: la ricerca semplice viene eseguita dal server (per cui risulta più veloce se la tabella presenta un numero elevato di righe), mentre la ricerca avanzata è un filtro lato client eseguita mediante una funzionalità di ExtJS. È stato necessario implementare un diverso meccanismo per la ricerca avanzata per motivi di sicurezza.

Inoltre verrà presentato il meccanismo di autenticazione degli utenti registrati e la procedura di caricamento dell'applicazione web. Le credenziali fornite dagli utenti

rimangono archiviate per tutta la durata della visita o sessione per evitare che siano richieste all'utente a ogni interrogazione al database.

A conclusione del Capitolo sono riportate e riassunte alcune delle considerazioni degli Storici sul prototipo realizzato: la valutazione è stata complessivamente positiva e sono state avanzate molte richieste di sviluppi futuri.

3.1 L'APPLICAZIONE WEB "GEOGRAFIE DELL'ITALIA MEDIEVALE (XIII – XV SECOLO)"

Nel Capitolo 2 è stata descritta la fase di progettazione del geoservizio "Geografie dell'Italia Medievale (XIII – XV secolo)": sono state definite l'interfaccia grafica utente, l'architettura client-server e sono state valutate le diverse tecnologie necessarie a implementare il servizio. In questo capitolo sarà descritto lo sviluppo dell'applicazione web sulla base delle scelte tecnologiche precedentemente illustrate.

L'applicazione web ha il compito di semplificare la navigazione e consultazione dei dati contenuti nel database storico, mostrando all'utente un'interfaccia grafica in due modalità: "WebGIS" e "Mostra tabella" (paragrafi 3.1.1 e 3.1.2). La navigazione del servizio è già stata descritta nel paragrafo 2.4.2 e ha inizio da una pagina di presentazione del progetto. Durante la fase prototipale l'accesso è stato ristretto ai soli partecipanti al progetto PRIN 2009 ed è consentito solo in seguito ad una procedura di autenticazione (paragrafo 3.2); in futuro, con il consenso degli Storici, l'accesso ai dati del database storico sarà esteso anche ai non partecipanti al progetto.

3.1.1 NAVIGAZIONE DEL GEOSERVIZIO: MODALITÀ "WEBGIS"

Nella modalità "WebGIS" è possibile navigare e consultare su un supporto cartografico i dati georeferenziati del database storico; in particolare è possibile visualizzare i seguenti strati informativi: "luogo (storico)", "luogo attuale", "ufficio", "unità fiscale" e i confini delle Regioni italiane dell'Istat (Figura 3.1). La simbologia

dei singoli *layer*, così come le diverse scale di visualizzazione, sono state definite in un progetto QGIS (Figura 3.2), successivamente pubblicato tramite QGIS Server.



Figura 3.1 – Interfaccia del geoservizio “Geografie dell’Italia Medievale (XIII – XV secolo)”.

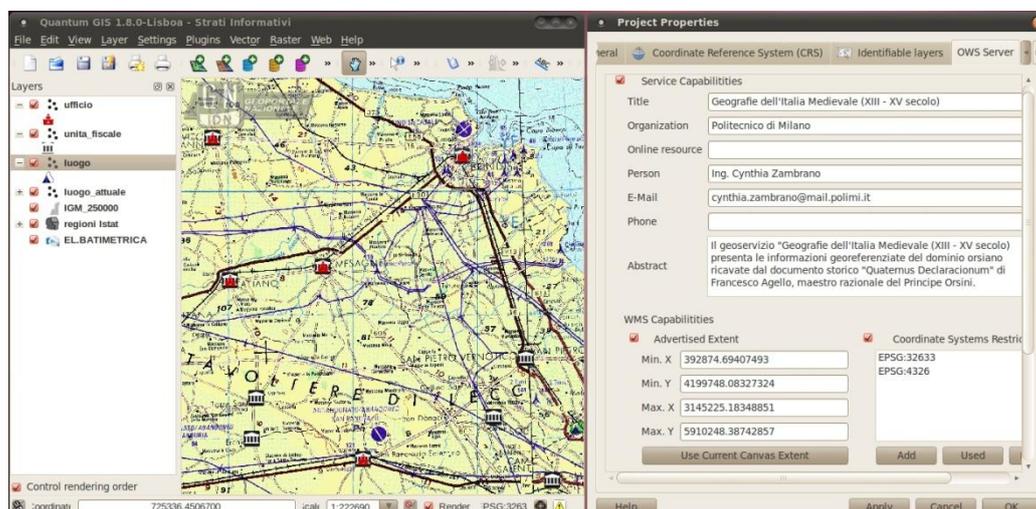


Figura 3.2 – Progetto QGIS in cui sono state definite le simbologie dei layer, le scale di visualizzazione e altre proprietà della mappa.

Nel progetto QGIS sono stati rinominati alcuni campi dei *layer* con opportuni alias, così come sono stati “nascosti” campi poco significativi per gli utenti del servizio come ad esempio tutte le chiavi esterne (*foreign key*) delle tabelle degli attributi, utili per la gestione del database ma poco rilevanti per gli utenti finali. In Figura 3.2 si può osservare che per il servizio WMS realizzato “Geografie dell’Italia Medievale (XIII – XV secolo)” sono stati specificati il sistema di riferimento e altre informazioni relative al servizio WMS.

Come base cartografica di supporto nel WebGIS è stata importata la carta corografica alla scala 1:250000 del Geoportale Nazionale – Ministero dell’Ambiente e della Tutela del Territorio e del Mare, disponibile come servizio WMS²² anche nello stesso sistema di riferimento dei dati del database storico (WGS84, UTM fuso 33). La carta corografica infatti è volta a rappresentare una regione sotto il profilo sia fisico che antropico, indicando le eventuali relazioni di interdipendenza tra i fatti osservati.

L’interfaccia grafica utente del servizio in modalità WebGIS deriva principalmente da QGIS web client, un applicativo basato su GeoExt, una libreria JavaScript unione delle funzionalità grafiche di ExtJS e delle funzionalità di *web mapping* di OpenLayers. Questo pacchetto client utilizza i parametri dei servizi WMS generati mediante QGIS Server per creare un WebGIS con le funzionalità di base (albero degli strati informativi, legenda, funzionalità di navigazione, funzionalità di interrogazione, ecc). In generale si è cercato di personalizzare la grafica ed è stato spostato il pannello dei risultati delle interrogazioni nell’area di destra per renderlo conforme alla scheda “Mostra tabella”. Inoltre è stata implementata qualche personalizzazione e corretto un errore di visualizzazione nel *browser* Chrome. Il problema riscontrato si manifestava nelle interrogazioni in mappa: l’oggetto selezionato era evidenziato in una posizione traslata rispetto a quella corretta. L’errore è stato corretto nascondendo e ridisegnando il *layer* vettoriale degli elementi selezionati.

²² Servizio disponibile all’URL (gennaio 2013)
http://wms.pcn.minambiente.it/ogc?map=/ms_ogc/WMS_v1.3/raster/IGM_250000.map.

Nelle impostazioni generali di QGIS web client sono stati definiti il sistema di riferimento (epsgcode), l'unità di misura (units), la scala massima (maxScale) e minima (minScale), parametri utilizzati nella costruzione della mappa del WebGIS.

```
var epsgcode = 32633;
var MapOptions = {
  projection: new OpenLayers.Projection("EPSG:"+epsgcode),
  units: "meters",
  maxScale:50,
  minScale: 11000000,
  transitionEffect:"resize"
}
```

Nell'estratto di codice sottostante sono specificate altre proprietà quali i *layer* (selectedLayers) da mostrare nell'applicazione web, il formato dell'immagine (format), la risoluzione (dpi) e lo zoom all'avvio dell'applicazione.

```
geoExtMap = new GeoExt.MapPanel({
  zoom: 3.6,
  layers: [
    thematicLayer = new OpenLayers.Layer.WMS (layer, wmsURI, {
      layers:selectedLayers,
      format:"image/jpeg",
      dpi:screenDpi
    }, {
      buffer:0,
      singleTile:true,
      ratio:1,    } ),...],
  map: MapOptions,
  id: "geoExtMapPanel",
  renderTo: MapPanelRef.body
```

Di seguito alcune delle principali funzionalità di QGIS web client che sono state utilizzate nell'applicazione web (Figura 3.3 e 3.4):

- albero degli strati informativi da cui è possibile impostare la visibilità dei *layer*;
- Strumenti di navigazione dei dati geografici quali zoom in, zoom out, zoom precedente, zoom su tutti i dati, pan e indicatori delle coordinate e della scala;
- Legenda dei singoli strati informativi;
- Mappa di *overview* dell'area visualizzata;
- Ricerca nelle tabelle degli attributi dei *layer*;
- Funzionalità di interrogazione dei dati sulla mappa;
- Strumento di misura della distanza tra due o più punti e strumento di misura dell'area di un poligono;
- Stampa il layout e ricerca della località con il servizio Geonames²³.

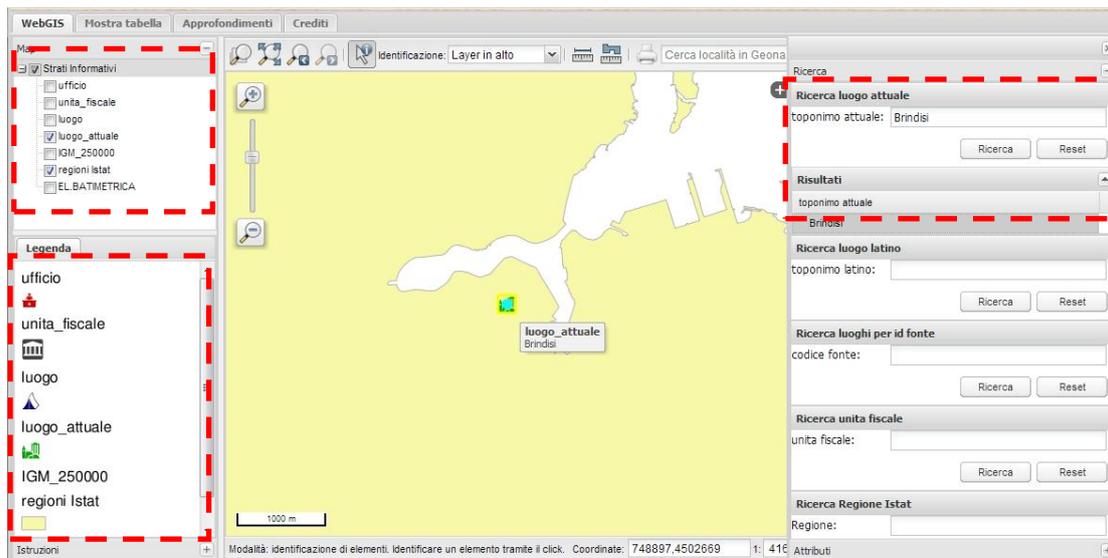


Figura 3.3 – Scheda WebGIS. In senso antiorario: albero dei layer, legenda, riquadri di ricerca e corrispondente griglia dei risultati.

²³ Geonames è un database di dati cartografici disponibile e accessibile tramite servizi web con licenza Creative Commons.



Figura 3.4 – Dettaglio degli strumenti nella modalità WebGIS.

In senso antiorario: strumenti di zoom e pan, slider della scala, coordinate e scala, mappa di overview, stampa e ricerca località, strumenti di misura di distanze e aree, strumenti di interrogazione in mappa.

Per agevolare la ricerca di informazioni quali il toponimo italiano, il toponimo latino, il codice della fonte, le unità fiscali e la Regione di appartenenza sono stati predisposti opportuni riquadri di ricerca, sfruttando le funzionalità del web client di QGIS. Queste funzionalità sono state configurate a livello di codice negli *script* JavaScript di QGIS web client. Tutti i criteri/riquadri di ricerca sono poi stati aggiunti al pannello di ricerca `mapSearchPanelConfigs`.

```
var mapSearchPanelConfigs = {};
mapSearchPanelConfigs[urlProject] = [TopoItaSearch,
TopoLatSearch, idFonteSearch, idUnitaFiscale,
regioneIstat];
```

Per ogni riquadro è necessario indicare nel codice JavaScript il campo e il *layer* su cui si desidera eseguire la ricerca e impostare la struttura della tabella dei risultati.

A titolo di esempio si riporta l'estratto di codice corrispondente al primo criterio di ricerca: la ricerca per nome del toponimo attuale.

```

var TopoItaSearch = {
  title: "Ricerca luogo attuale",
  query: 'simpleWmsSearch',
  useWmsRequest: true,
  queryLayer: "luogo_attuale",
  formItems: [
    {
      xtype: 'textfield',
      name: 'toponimo_ita',
      fieldLabel: "toponimo attuale",
      allowBlank: false,
      blankText: "Inserire il toponimo da cercare"
    }
  ],
  gridColumns: [
    {
      header: 'toponimo attuale',
      dataIndex: 'toponimo_ita'
    }
  ],
  selectionLayer: 'luogo_attuale',
  selectionZoom: 9
};...

```

La variabile `TopoItaSearch` è un oggetto che contiene le proprietà del criterio di ricerca; tra di esse le più interessanti sono:

- `queryLayer`, il layer su cui sarà effettuata la ricerca;
- `formItems`, in cui viene definito il tipo di dato da cercare (numerico, stringa, ecc) e il campo in cui eseguire la ricerca;
- `gridColumns`, in cui viene definita la struttura della tabella dei risultati della ricerca;
- `selectionLayer`, layer su cui sono evidenziati i risultati della ricerca;

- selectionZoom, scala di zoom²⁴ a cui viene eseguito l'ingrandimento in mappa, una volta selezionato uno dei risultati.

In Figura 3.5 è riportato il risultato di una generica ricerca nell'interfaccia grafica utente.



Figura 3.5 – Ricerca nella modalità WebGIS; a destra il pannello di ricerca con la griglia dei risultati, a sinistra il risultato evidenziato nel layer “luogo attuale”.

Nell'applicazione web sono state integrate funzionalità di interrogazione degli elementi in mappa, già predisposte da QGIS web client. L'utente può effettuare interrogazioni mediante due modalità selezionabili dal menù a tendina evidenziato in Figura 3.6:

- interrogazione sul “Layer in alto”, ovvero sul primo strato informativo nell'albero dei *layer*;
- interrogazione su “Tutti i layer visibili”, ovvero su tutti gli strati informativi selezionati nell'albero dei *layer*.

I risultati sono riportati nel pannello “Attributi” suddivisi per strato informativo. I nomi degli attributi corrispondono agli *alias* impostati nel progetto QGIS da cui deriva il servizio.

²⁴ Le scale di zoom sono definite nel codice JavaScript di QGIS web client.

Inoltre per consentire agli utenti di avere la possibilità di avere informazioni sugli elementi in mappa in maniera più agevole sono stati introdotti dei *tooltip* o suggerimenti sugli oggetti. Le informazioni che compaiono al passaggio del cursore sugli elementi in mappa sono state aggiunte in campi dal nome “tooltip” nelle tabelle degli attributi dei *layer* “unità fiscale”, “luogo”, “luogo attuale” e “regioni Istat”.

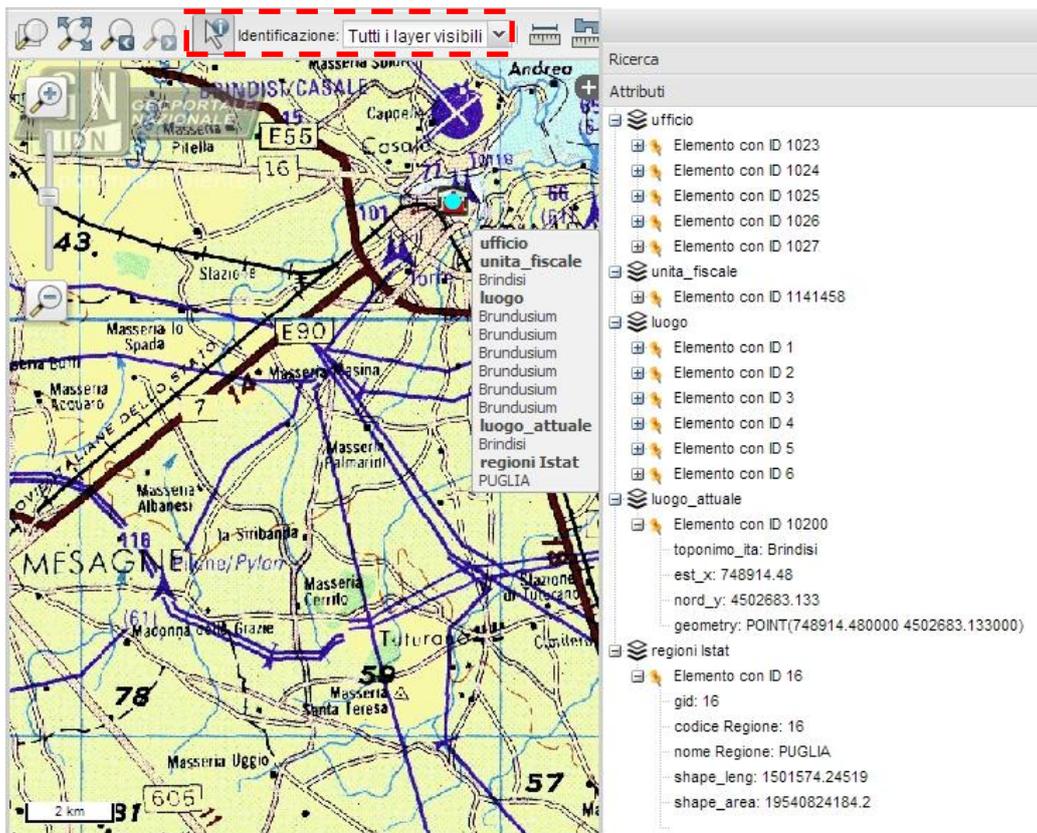


Figura 3.6 – Funzionalità di interrogazione sulla mappa. In rosso è evidenziato il menù a tendina che consente di selezionare la modalità di ricerca. A destra i risultati dell’interrogazione.

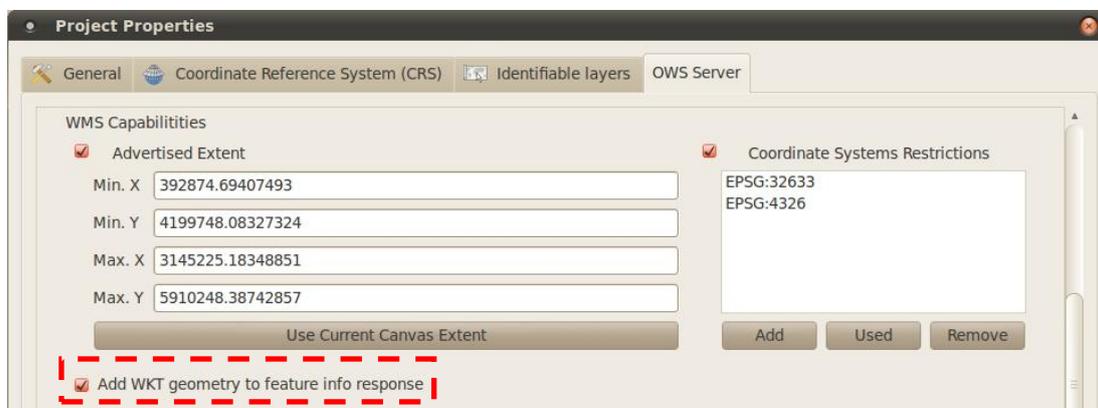
Gli oggetti selezionati in un’interrogazione sono rappresentati in un *layer* vettoriale predisposto ad accogliere i risultati delle interrogazioni. I risultati sono evidenziati in mappa mediante una simbologia definita a livello di codice JavaScript, come si può osservare nell’estratto successivo.

```

var symbolizersHighLightLayer = {"Point":{
  pointRadius: 4,
  graphicName: "circle",
  fillColor: "#00FFFF",
  fillOpacity: 1,
  strokeWidth: 3,
  strokeOpacity: 1,
  strokeColor: "#00FFFF"
}}...

```

Per rappresentare gli oggetti di geometria vettoriale e il sistema di riferimento dei dati, QGIS utilizza il linguaggio di *markup* WKT (*Well Known Text*)²⁵. Solitamente alle richieste di interrogazione dei dati (“GetFeatureInfo”) non viene restituita la stringa WKT della geometria, tuttavia per poter visualizzare i risultati delle interrogazioni in QGIS web client è necessario rendere visibile questa informazione dal pannello delle proprietà del progetto (Figura 3.7).



*Figura 3.7 – Proprietà del progetto QGIS.
In evidenza l'opzione che permette di rendere visibili i risultati.*

²⁵ Paragrafo 1.8 nota 5.

3.1.2 NAVIGAZIONE DEL GEOSERVIZIO: MODALITÀ “MOSTRA TABELLA”

La seconda scheda del servizio “Geografie dell’Italia Medievale (XIII – XV secolo)” consente di consultare e fare ricerche sul contenuto alfanumerico delle tabelle e delle viste del database storico; fra queste sono presenti anche le tabelle degli attributi di strati informativi “luogo”, “luogo attuale”, “ufficio fiscale”, “unità fiscale” e “distretto erariale”. Come si vede nella Figura 3.8, nella colonna di sinistra sono elencate le tabelle e le viste che si è ritenuto utile rappresentare nell’applicazione web. Sulla destra è invece rappresentato il contenuto della tabella/vista selezionata.

The screenshot shows the 'Geografie dell'Italia Medievale' web application interface. The main content area displays a table with 143 records. The table has the following columns: id fonte, cod fonte, archivio, fondo, num pezzo, num carta, r/v, and tipo fonte. The records list various sources from the Archivio di Stato di Napoli, specifically from the Diversi Sommaria II series.

id fonte	cod fonte	archivio	fondo	num pezzo	num carta	r/v	tipo fonte
F 10.100.10248.100210	ASN-OS II-248-2f	Archivio di Stato di Napoli	Diversi Sommaria II	248	2	r	Quaternus Declaracionum
F 10.100.10248.100311	ASN-OS II-248-3v	Archivio di Stato di Napoli	Diversi Sommaria II	248	3	v	Quaternus Declaracionum
F 10.100.10248.100510	ASN-OS II-248-5r	Archivio di Stato di Napoli	Diversi Sommaria II	248	5	r	Quaternus Declaracionum
F 10.100.10248.100710	ASN-OS II-248-7r	Archivio di Stato di Napoli	Diversi Sommaria II	248	7	r	Quaternus Declaracionum
F 10.100.10248.100711	ASN-OS II-248-7v	Archivio di Stato di Napoli	Diversi Sommaria II	248	7	v	Quaternus Declaracionum
F 10.100.10248.102211	ASN-OS II-248-22v	Archivio di Stato di Napoli	Diversi Sommaria II	248	22	v	Quaternus Declaracionum
F 10.100.10248.102310	ASN-OS II-248-23r	Archivio di Stato di Napoli	Diversi Sommaria II	248	23	r	Quaternus Declaracionum
F 10.100.10248.102311	ASN-OS II-248-23v	Archivio di Stato di Napoli	Diversi Sommaria II	248	23	v	Quaternus Declaracionum
F 10.100.10248.102410	ASN-OS II-248-24r	Archivio di Stato di Napoli	Diversi Sommaria II	248	24	r	Quaternus Declaracionum
F 10.100.10248.102411	ASN-OS II-248-24v	Archivio di Stato di Napoli	Diversi Sommaria II	248	24	v	Quaternus Declaracionum
F 10.100.10248.102510	ASN-OS II-248-25r	Archivio di Stato di Napoli	Diversi Sommaria II	248	25	r	Quaternus Declaracionum
F 10.100.10248.102711	ASN-OS II-248-27v	Archivio di Stato di Napoli	Diversi Sommaria II	248	27	v	Quaternus Declaracionum
F 10.100.10248.102810	ASN-OS II-248-28r	Archivio di Stato di Napoli	Diversi Sommaria II	248	28	r	Quaternus Declaracionum
F 10.100.10248.102811	ASN-OS II-248-28v	Archivio di Stato di Napoli	Diversi Sommaria II	248	28	v	Quaternus Declaracionum
F 10.100.10248.102910	ASN-OS II-248-29r	Archivio di Stato di Napoli	Diversi Sommaria II	248	29	r	Quaternus Declaracionum
F 10.100.10248.102911	ASN-OS II-248-29v	Archivio di Stato di Napoli	Diversi Sommaria II	248	29	v	Quaternus Declaracionum
F 10.100.10248.103211	ASN-OS II-248-32v	Archivio di Stato di Napoli	Diversi Sommaria II	248	32	v	Quaternus Declaracionum
F 10.100.10248.103410	ASN-OS II-248-34r	Archivio di Stato di Napoli	Diversi Sommaria II	248	34	r	Quaternus Declaracionum
F 10.100.10248.103411	ASN-OS II-248-34v	Archivio di Stato di Napoli	Diversi Sommaria II	248	34	v	Quaternus Declaracionum

Figura 3.8 – Scheda “Mostra tabella”.

Il codice JavaScript e PHP per la realizzazione di questa scheda è stato progettato e implementato in modo da far dialogare l’interfaccia grafica (lato client) con gli *script* lato server in linguaggio PHP mediante chiamate asincrone AJAX (paragrafo 2.2). L’utilizzo di AJAX ha consentito di rendere il caricamento di tabelle e viste dinamico e “parametrizzato”²⁶. Per gestire in maniera semplice la visibilità di ciascuna tabella, il nome visualizzato nell’interfaccia web e i campi in cui effettuare le ricerche sono

²⁶ Il processo di caricamento dinamico sarà spiegato in maniera più dettagliata nel paragrafo 3.3.

state create due nuove tabelle nel database storico, la prima per la gestione di tabelle e viste, la seconda per la gestione dei relativi campi. Nella prima tabella “tables_settings” (Figura 3.9) sono riportate tutte le tabelle e viste del database storico; per ognuna sono stati definiti un alias (“display_text”), il tipo²⁷ (“type”) e le proprietà booleane “visible”, “editable” e “searchable”. Ad esempio nella tabella “Appartenenza” la proprietà “visible” è FALSE, di conseguenza non comparirà nell’elenco delle tabelle nell’interfaccia web.

La struttura della tabella di gestione dei campi di tabelle e viste “fields_settings” (Figura 3.10) è analoga: oltre alle proprietà “visible”, “editable” e “searchable” vi sono un campo per gli alias (campo “column_name”) e uno per il tipo di dato (campo “data_type”) utile per le funzioni di ricerca (paragrafo 3.2.2).

	name [PK] text	display_text text	visible boolean	editable boolean	type text	searchable boolean
1	appartenenza	appartenenza	FALSE	FALSE	table	FALSE
2	distretto_erariale	distretto erariale	TRUE	TRUE	table	TRUE
3	fonte	fonte storica	TRUE	FALSE	table	TRUE
4	imposte	imposte medievali	TRUE	FALSE	table	TRUE
5	luogo	insediamento latino	TRUE	FALSE	table	TRUE
6	luogo_attuale	luogo attuale	TRUE	TRUE	table	TRUE
7	personaggio	personaggio storico	TRUE	TRUE	table	TRUE
8	tabellone	tabella originale	TRUE	FALSE	view	TRUE
9	ufficiale	ufficiale	TRUE	FALSE	table	TRUE
10	ufficio	ufficio fiscale	TRUE	FALSE	table	TRUE
11	unita_fiscale	unita fiscale	TRUE	FALSE	table	TRUE
12	vista1	luogo-unita fiscale	TRUE	FALSE	view	TRUE

Figura 3.9 – Estratto della tabella “tables_settings”.

I campi impostati come “non visibili” risultano quindi nascosti al caricamento della tabella o vista, ma possono essere resi visibili manualmente operando sui controlli dell’interfaccia grafica utente (Figura 3.11). Per tutti i campi delle tabelle e viste sono inoltre disponibili funzioni di ordinamento crescente e decrescente²⁸.

²⁷ Questa informazione consente di organizzare e classificare dinamicamente tabelle e viste negli alberi “Tabelle consultabili” e “Viste di consultazione”.

²⁸ Queste funzioni sono “metodi” tipici degli oggetti *grid* di ExtJS, che sono stati usati in questa applicazione per rappresentare le tabelle e le viste.

	table_name [PK] informat	column_name [PK] informatio_n_s	data_type information_schen	udt_name informati	visible boolean	searchabl boolean	editable boolean
175	ufficio	nord_y	double precision	float8	FALSE	FALSE	FALSE
176	ufficio	tesoreria	boolean	bool	TRUE	TRUE	FALSE
177	unita_fiscale	anno	integer	int4	TRUE	TRUE	FALSE
178	unita_fiscale	est_x	double precision	float8	FALSE	FALSE	FALSE
179	unita_fiscale	geom	USER-DEFINED	geometry	FALSE	FALSE	FALSE
180	unita_fiscale	id_distretto_erariale	integer	int4	FALSE	FALSE	FALSE
181	unita_fiscale	id_unita_fiscale	integer	int4	TRUE	TRUE	FALSE
182	unita_fiscale	nord_y	double precision	float8	FALSE	FALSE	FALSE
183	unita_fiscale	tesoreria	boolean	bool	TRUE	TRUE	FALSE
184	unita_fiscale	tooltip	text	text	FALSE	FALSE	FALSE
185	unita_fiscale	toponimo_ita	character varying	varchar	TRUE	TRUE	FALSE
186	vista1	anno	integer	int4	TRUE	TRUE	FALSE

Figura 3.10 – Estratto della tabella “fields_settings”.

La parametrizzazione della procedura di caricamento delle tabelle e viste rende più semplice la gestione e il mantenimento futuro dell’applicazione web: infatti per ogni nuova tabella o vista che si desidera visualizzare nell’applicazione sarà sufficiente aggiungere una nuova riga nella tabella “tables_settings”, e nella tabella “fields_settings” tante righe quanti sono i campi della nuova tabella/vista affinché la nuova tabella del database storico compaia nell’interfaccia dell’applicazione.

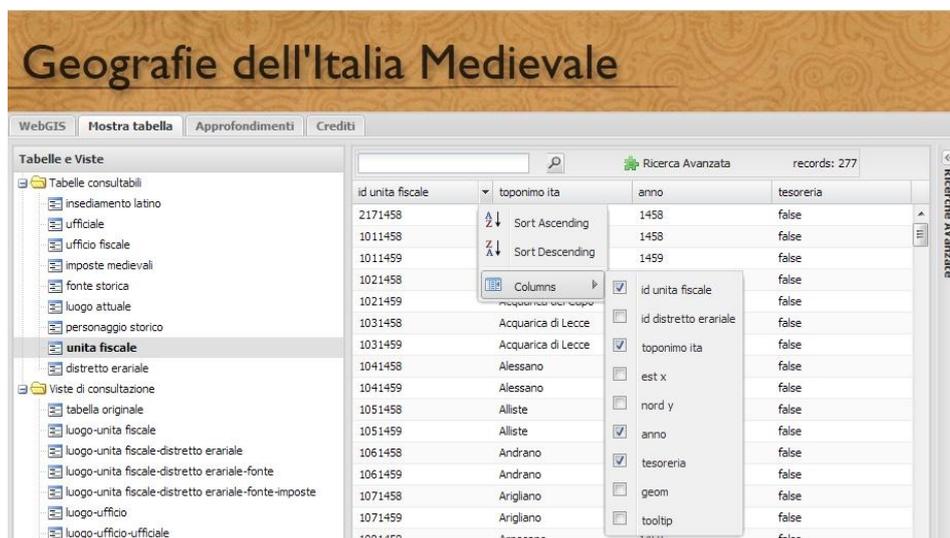
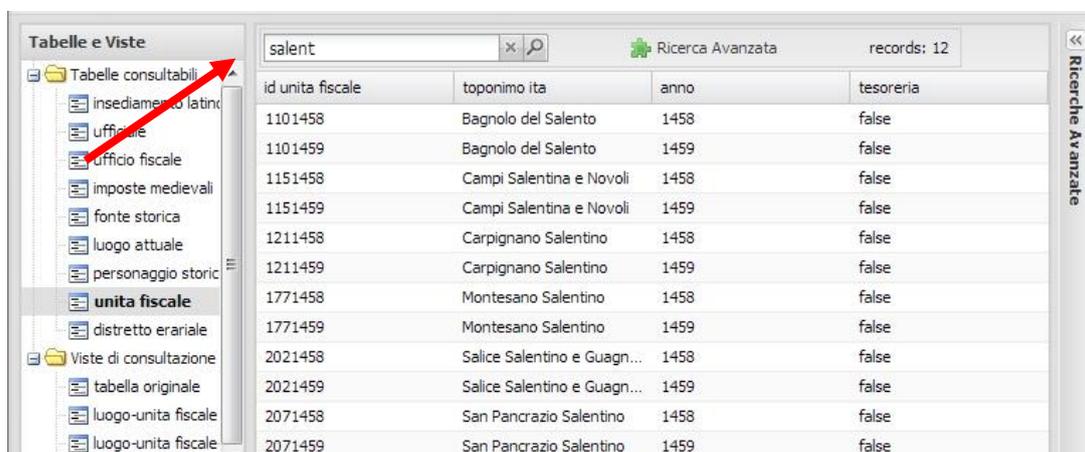


Figura 3.11 – Oltre alle funzioni di ordinamento crescente e decrescente è possibile aggiungere o nascondere campi della tabella/vista attiva.

3.2 IMPLEMENTAZIONE DELLE FUNZIONALITÀ DI RICERCA NELLA MODALITÀ “MOSTRA TABELLA”

Per consentire agli utenti di eseguire interrogazioni sulle tabelle del database visibili dall'applicazione web, sono stati implementati due diversi strumenti di ricerca che agiscono sulla tabella/vista attiva. Le due modalità di ricerca sono state progettate e realizzate per coprire esigenze diverse: nella ricerca “semplice” (Figura 3.12) l'utente inserisce un valore da cercare ed esegue un filtro veloce sui *record* della tabella attiva che presentano il valore inserito; nella ricerca “avanzata” (Figura 3.13) l'utente può comporre più interrogazioni o *query* e concatenarle mediante gli operatori logici AND/OR.

Entrambe le modalità di ricerca fanno sì che nell'interfaccia siano visibili solo le righe corrispondenti ai criteri impostati e aggiornano in maniera dinamica il contatore del numero delle righe posto sopra la tabella/vista attiva.



id unita fiscale	toponimo ita	anno	tesoreria
1101458	Bagnolo del Salento	1458	false
1101459	Bagnolo del Salento	1459	false
1151458	Campi Salentina e Novoli	1458	false
1151459	Campi Salentina e Novoli	1459	false
1211458	Carpignano Salentino	1458	false
1211459	Carpignano Salentino	1459	false
1771458	Montesano Salentino	1458	false
1771459	Montesano Salentino	1459	false
2021458	Salice Salentino e Guagn...	1458	false
2021459	Salice Salentino e Guagn...	1459	false
2071458	San Pancrazio Salentino	1458	false
2071459	San Pancrazio Salentino	1459	false

Figura 3.12 – Esempio di ricerca semplice.

id unita fiscale	toponimo_ita	anno	tesoreria
1101459	Bagnolo del Salento	1459	false
1151459	Campi Salentina e Novoli	1459	false
1211459	Carpignano Salentino	1459	false
1771459	Montesano Salentino	1459	false
2021459	Salice Salentino e Guagnano	1459	false
2071459	San Pancrazio Salentino	1459	false

Ricerche Avanzate

Condizione 1

Campo di ricerca:

Operatore logico:

Valore:

Concatena condizione: AND OR

Condizione 2

Campo di ricerca:

Operatore logico:

Valore:



Figura 3.13 – Esempio di ricerca avanzata.

3.2.1 FUNZIONALITÀ DI RICERCA SEMPLICE

La ricerca semplice avviene mediante il campo di input sopra la tabella/vista attiva e permette di cercare una stringa di testo in tutti i campi in cui è consentito eseguire ricerche (ovvero definiti “searchable” nella tabella di gestione “fields_settings”). La ricerca semplice viene eseguita dal server che riceve dall’applicazione web due parametri: \$value (il testo da cercare) e searchFields (una stringa contenente i campi della tabella/vista in cui eseguire la ricerca concatenati dal carattere “.”).

L’estratto dello script index.php sotto riportato separa i diversi campi su cui eseguire la ricerca con il comando explode e utilizza il valore \$value inserito dall’utente nella casella di testo per creare la stringa d’interrogazione nella variabile \$query, iterando su ogni campo in cui cercare.

```

function getList()
{
    $query = 'SELECT * FROM '.$_REQUEST['table'];
    if (isset($_REQUEST['searchFields'] )
    {
        $campiRicerca =
explode(".", $_REQUEST['searchFields']);
        $value = $_REQUEST['valoreInserito'];
        $query .= ' WHERE ';
        for ($i=0; $i< count($campiRicerca); $i++)
        {
            $query.= $campiRicerca[$i]. " ILIKE '%" . $value . "%'
OR ";
        }
        $query = substr($query, 0, -3);
    }
    $result = "";
    $result = pg_query($query)
...

```

Ad esempio l'interrogazione mostrata in Figura 3.12 corrisponde a

```

$query = 'SELECT * FROM unita_fiscale WHERE toponimo_ita
ILIKE "salent" '

```

in quanto l'unico campo di tipo stringa definito "searchable" nella tabella "unità fiscale" è "toponimo_ita".

Il risultato dell'interrogazione viene successivamente codificato in formato JSON e inviato all'interfaccia grafica, che lo mostra sotto forma di tabella.

Per proteggere il database da richieste non consentite o pericolose (inserite con la tecnica *SQL Injection*²⁹) è stato imposto un controllo nella casella di input che filtra e

²⁹ *SQL injection* è una tecnica mirata a colpire le applicazioni web che si appoggiano su un database di tipo SQL; sfrutta l'inefficienza dei controlli sui dati ricevuti in input ed inserisce codice maligno all'interno di una query SQL. Questa tecnica consente, ad esempio, di autenticarsi con ampi privilegi in aree protette del sito anche senza essere in possesso delle credenziali d'accesso e di visualizzare e/o alterare dati sensibili.

trasforma tutti i caratteri speciali inseriti nel carattere innocuo “_”. Tale controllo, ad esempio, trasforma il carattere speciale “;” (che potrebbe essere usato per concatenare al valore da cercare una richiesta in linguaggio SQL) nel carattere *jolly*³⁰ “_” del DBMS PostgreSQL. Grazie a questa soluzione è possibile eseguire ricerche contenenti caratteri speciali quali apici e virgolette senza mettere a rischio la sicurezza e l’integrità del database.

3.2.2 FUNZIONALITÀ DI RICERCA AVANZATA

Nella modalità avanzata l’utente è guidato nella composizione della ricerca composta da multipli criteri concatenati da operatori logici; per ciascun criterio è necessario definire i seguenti parametri:

- il campo di ricerca, selezionabile con il supporto di un menù a tendina (*comboCampi*) che contiene solo i campi “searchable” (sui quali è possibile effettuare ricerche) della tabella/vista attiva;
- l’operatore di confronto, anch’esso selezionabile da un menù a tendina (*comboOperatore*) che varia a seconda del tipo di dato del campo di ricerca selezionato (Tabella 3.1);
- la stringa o il numero (*valoreDaCercare*) da inserire nella casella di testo su cui è stato imposto il controllo che la casella non sia vuota. Questo parametro non è presente nel caso l’operatore di confronto sia booleano.

Gli elementi dell’interfaccia grafica che compongono una condizione di ricerca sono raccolti in *fieldset*; di conseguenza saranno presenti nel pannello di ricerca avanzata tanti *fieldset* quante sono le condizioni di ricerca impostate dall’utente.

³⁰ Carattere che all’interno di una stringa non rappresenta sé stesso bensì un insieme di altri caratteri o sequenze di caratteri.

Tabella 3.1 – Operatori di confronto proposti,
in dipendenza dal tipo di dato del campo selezionato.

TIPO DI DATO	stringa	numerici	booleani
Operatore di confronto selezionabile	diverso da	diverso da	is true
	=	=	is false
	contiene	<	
	non contiene	>	
		<=	
		>=	

Alla selezione del pulsante “Aggiungi condizione” viene creato e aggiunto al pannello di ricerca avanzata un *container* che comprende il selettore AND/OR (*radio button*) e un nuovo *fieldset*. La creazione dei riquadri di ricerca è stata parametrizzata in modo che non ci sia un limite al numero di criteri di ricerca che si desidera impostare. Ad esempio, per l’assegnazione del nome degli elementi grafici sono state definite le seguenti formule:

```
'searchFieldset'+n,  
'comboCampi'+n,  
'comboOperatore-'+n e  
'valoreDaCercare'+n, .
```

Dove l’indice n corrisponde all’n-esima condizione.

Analogamente, alla selezione del pulsante “Rimuovi ultima condizione” viene tolto dal pannello l’ultimo container (Figura 3.14) inserito.

The image shows a user interface for an advanced search panel. It consists of two main sections: 'FIELDSET' (highlighted in green) and 'CONTAINER' (highlighted in red). The 'FIELDSET' section contains 'Condizione 1' with a dropdown menu for 'Campo di ricerca' set to 'toponimo_ita', a dropdown for 'Operatore logico' set to '=', and a text input for 'Valore' containing 'Acaya'. The 'CONTAINER' section contains 'Concatena condizione:' with radio buttons for 'AND' and 'OR' (selected), 'Condizione 2' with a dropdown for 'Campo di ricerca' set to 'anno', a dropdown for 'Operatore logico' set to '>', and a text input for 'Valore' containing '1458'. Below these sections are buttons: 'Aggiungi condizione', 'Rimuovi ultima condizione', 'Nessuna selezione', and 'Esegui'.

Figura 3.14 – Componenti del pannello di ricerca avanzata.

Di seguito viene riportato un estratto del codice JavaScript che crea i nuovi *fieldset*. Come si può osservare, i dati che popolano il menù a tendina `comboCampi` provengono da `searchableFieldsStore`, un oggetto contenente i campi su cui è consentita la ricerca nella tabella attiva. Questo oggetto si aggiorna ogni volta che viene cambiata la tabella/vista attiva.

```
Ext.create('Ext.form.FieldSet', {
  id: 'searchFieldset'+n,
  title: 'Condizione '+n,
  items: [{
    id: 'comboCampi'+n,
    xtype: 'combobox',
    fieldLabel: 'Campo di ricerca',
    store: searchableFieldsStore,
    listeners: {select: {
      fn: function(combo, value) {
        datatypeField = value[0].data.type;
        updateQueryStore (datatypeField);
      }
    }}
  ]
});
```

Inoltre al momento della selezione di uno dei campi nel menù `comboCampi` viene chiamata la funzione `updateQueryStore()`, che aggiorna `OperatoriQueryStore`, l'oggetto contenente i possibili operatori di confronto per il campo selezionato.

```
{
  id: 'comboOperatore-'+n,
  xtype: 'combobox',
  fieldLabel: 'Operatore',
  store: OperatoriQueryStore,
  listeners:{
    select:{
      fn:function(combo,value){
        if (combo.value =='IS TRUE' || combo.value =='IS
FALSE')
        {
          Ext.getCmp('valoreDaCercare'+n).hide();
        }
      }
    }
  }
}
```

Si osservi che nel caso in cui l'operatore di confronto sia booleano (IS TRUE, IS FALSE) la casella di testo `valoreDaCercare` non è necessaria e viene nascosta con il comando `hide()`.

Una volta impostati correttamente i criteri di ricerca, l'utente chiede all'applicazione web di eseguire l'interrogazione: quest'azione richiama due funzioni in cascata `creaFiltro()` e `eseguiFiltro()`.

La funzione `creaFiltro()` legge i parametri impostati dall'utente e li organizza nell'array di oggetti (`condizioni`), dove ogni oggetto corrisponde a una singola condizione impostata dall'utente. Ad esempio, per la ricerca in Figura 3.14 si avrà un array composto come in Figura 3.15. Tutti gli oggetti tranne il primo (`condizioni[0]`) presentano la proprietà "congiungi" in cui viene archiviato l'operatore logico necessario per concatenare le condizioni di ricerca.

```

▼ condizioni : Array[2]
  ▼ 0: Object
    operatore: "="
    prop: "toponimo_ita"
    valore: "Acaya"
    ► __proto__: Object
  ▼ 1: Object
    congiungi: "OR"
    operatore: ">"
    prop: "anno"
    valore: "1458"
    ► __proto__: Object
    length: 2
    ► __proto__: Array[0]

```

Figura 3.15 – Array delle condizioni impostate nella ricerca di Figura 3.14.

L’output della funzione `creaFiltro()` è utilizzato dalla funzione `eseguiFiltro()` che, per ogni riga della tabella/vista attiva, prima valuta le singole condizioni (che possono essere vere o false) e successivamente le confronta per dare un unico esito. Alla fine del processo sono presentate in tabella solo le righe con esito complessivo `true`.

Questo metodo di ricerca avviene sfruttando esclusivamente le funzionalità del lato client; non ci sono nuove chiamate al database e la ricerca viene svolta filtrando dati già presenti nella tabella attiva. Si è preferito usare questa tecnica per motivi di sicurezza: infatti se la ricerca fosse eseguita lato server la possibilità di concatenare codice SQL avrebbe potuto consentire la modifica o la cancellazione dei dati dal database storico.

3.3 PROCEDURA DI AUTENTICAZIONE E DI CARICAMENTO DELL’APPLICAZIONE WEB

Per poter procedere con le operazioni di validazione del prototipo del geoservizio è stato deciso che, per tutta la durata della fase prototipale, l’accesso al geoservizio “Geografie dell’Italia Medievale (XII – XV secolo)” sarà consentita solo con

opportune credenziali (*username* e *password*), secondo il seguente processo di autenticazione (Figura 3.16).

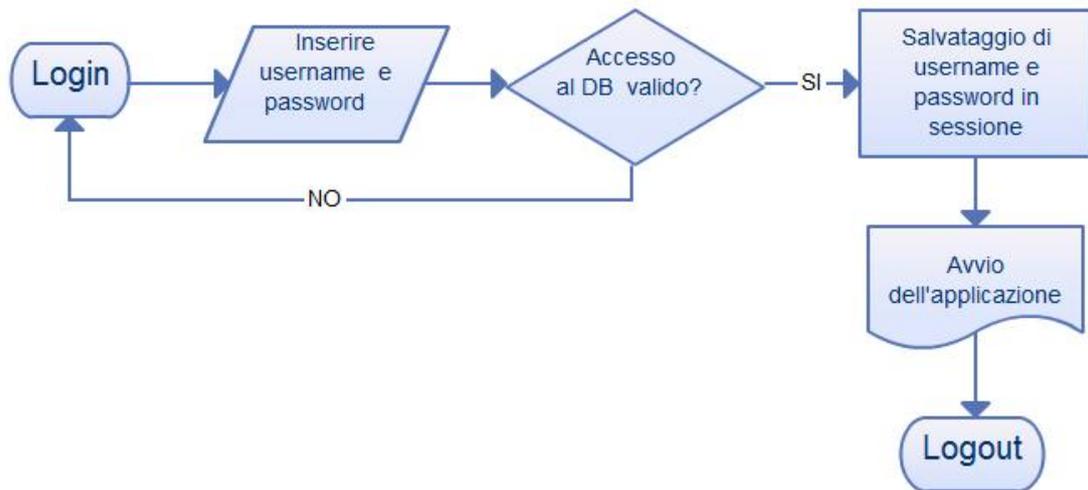


Figura 3.16 – Diagramma di flusso della procedura di autenticazione.

Nella pagina principale di presentazione del progetto è disponibile un *form* da cui è possibile accedere con un'utenza presente nel database. Se il procedimento di autenticazione non va a buon fine viene restituito un messaggio di errore e l'utente è rimandato alla pagina di login, se invece le credenziali sono riconosciute vengono archiviate *username* e *password* per la durata di tutta la sessione e si avvia il caricamento dell'applicazione web. Il salvataggio delle credenziali è stato introdotto per evitare di chiedere continuamente all'utente di inserirle ad ogni chiamata al database storico.

La procedura di avvio dell'applicazione web è stata parametrizzata il più possibile: grazie alle tabelle di gestione "tables_settings" e "fields_settings" che contengono le informazioni necessarie per il caricamento, non devono essere apportate modifiche a livello di codice.

Il caricamento dell'applicazione web può essere schematizzato in quattro passaggi fondamentali (Figura 3.17).

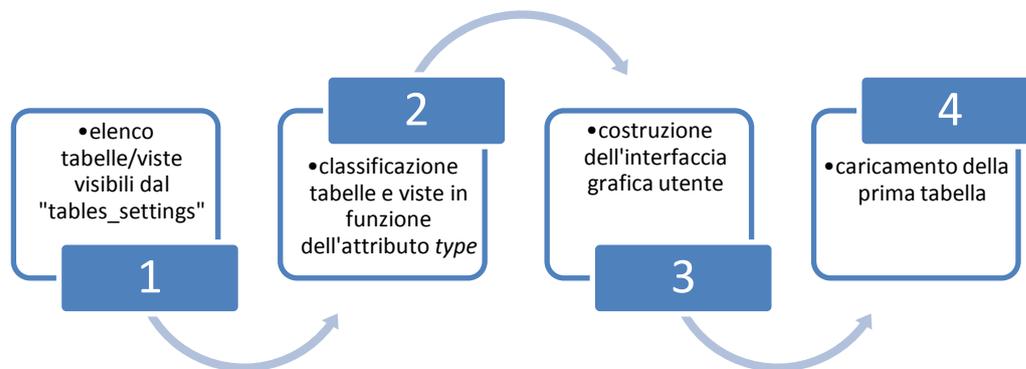


Figura 3.17 – Passaggi fondamentali nella procedura di caricamento dell'applicazione web.

Letture dalla tabella di gestione “tables_settings” dell’elenco delle tabelle/viste da mostrare nella scheda “Mostra tabella” dell’applicazione web. La lettura avviene mediante una chiamata AJAX in cui il codice PHP esegue una *query* sul database (task GET_TABLE_STRUCTURE, funzione tables_gui_structure) che restituisce in output un oggetto in codice JSON (tabelleStoriche) contenente l’elenco di tutte le tabelle/viste e per ognuna l’elenco dei relativi campi e proprietà. Di seguito l’estratto del codice PHP che esegue l’interrogazione al database:

```

function tables_gui_structure()
{...
    $tableStructure = "";
    $query = "SELECT * FROM tables_settings";
    $result = pg_query($query);
    ...
}
  
```

e un estratto della risposta restituita al client (tabelleStoriche) in formato JSON. Le prime righe riassumono le proprietà della tabella “luogo” mentre nella proprietà “fields” sono riportati tutti i campi con le corrispondenti caratteristiche.

```

[
  {
    id:"luogo",
    tableName:"luogo",
    type:"table",
    displayText:"insediamento latino",
    visible:true,
    searchable:true,
    editable:false,
    fields:
    [
      {
        name:"id_luogo",
        type:"string",
        mapping:"id_luogo",
        searchable:true,
        editable:false,
        visible:true,
      },
      {
        name:"id_luogo_attuale",
        type:"int",
        mapping:"id_luogo_attuale",
        searchable:true,
        editable:false,
        visible:true,
      }
    ]
  }
]

```

A partire dalla risposta `tabelleStoriche` sono filtrate le tabelle o viste che presentano la proprietà `“visible” = true`. Inoltre ogni elemento di `tabelleStoriche` viene classificato in tabella o vista in funzione dell’attributo `type` e aggiunto all’opportuno albero.

```

if (tabelleStoriche[i].visible) {
switch(tabelleStoriche[i].type){
case "view":
addViewToTree(
tabelleStoriche[i].id,
tabelleStoriche[i].displayText);
break;
case "table":
addTableToTree(
tabelleStoriche[i].id,
tabelleStoriche[i].displayText);
break;
}
}...
createTreeStore();

```

Viene costruita l'interfaccia grafica utente e le due schede “Mostra tabella” e “WebGIS”, inserita nell'applicazione come pagina esterna (*iframe*).

```

xtype: 'tabpanel',
id: 'tabpanel',...
items:[{
xtype: 'panel',
id: 'WebGIS',
title:'WebGIS',
html:'<iframe id="WebGIS_Iframe"
src="url_qgiswebclient?map=/.../projectName.qgs&
format=image/png&visibleLayers=listOfLayers"
frameborder="0"> </iframe>',{
xtype: 'panel',
id: 'database',
layout: 'border',
title: 'Mostra tabella'

```

Infine viene creata e caricata la tabella/vista predefinita, che può essere impostata a livello di codice nella variabile `activeTable`. Le funzioni che provvedono alla creazione e caricamento di tale tabella sono `initModel()`, `createStore()` e `createTable()`.

3.4 VALIDAZIONE DEL PROTOTIPO DEL GEOSERVIZIO PER I DATI STORICI

Nell'ambito della progettazione e dello sviluppo di nuovi strumenti informatici per utenti di formazione umanistica è importante tenere in considerazione il concetto di usabilità, secondo cui l'immediatezza d'uso è proporzionale a quanto il modello del progettista (ovvero le idee di questi riguardo al funzionamento dello strumento) si avvicina a quello dell'utente. Al fine di ottenere una validazione di quanto realizzato, il prototipo del geoservizio "Geografie dell'Italia Medievale (XIII – XV sec.)" è stato presentato all'Unità di Ricerca di Lecce.

Il riscontro da parte degli Storici è stato complessivamente positivo, in particolare è stata apprezzata la duplice modalità di consultazione dei dati ("Mostra tabella" e "WebGIS"), tanto da richiederne un ulteriore sviluppo. A seguito di un primo utilizzo dello strumento è stato suggerito che una ricerca eseguita in una delle due modalità si riflettesse automaticamente sull'altra. È interessante osservare come la duplice modalità di consultazione, dapprima assente nel loro *modus operandi* sia stata accolta positivamente e sia al centro di nuove richieste.

Gli strumenti sviluppati e implementati nell'applicazione web sono risultati di immediato uso, ad eccezione della funzionalità di "Ricerca avanzata" per la quale è stata necessaria qualche indicazione sull'utilizzo. In seguito gli hanno lavorato autonomamente sulle *query*, ad esempio selezionando le località appartenenti a un determinato distretto erariale ed evidenziandone la distribuzione geografica in mappa.

La rappresentazione geografica delle località storiche nella modalità WebGIS si è dimostrata utile per individuare eventuali errori di georeferenziazione: ad esempio, durante l'incontro gli Storici hanno notato che le località San Pietro in Lama e di Cocumola fossero state georeferenziate in posizioni non corrette. Analogamente la verifica dello strumento si è rivelata utile anche nell'individuazione di errori nella modalità "Mostra tabella".

La funzionalità implementata nel WebGIS di calcolo di distanze tra punti e aree di poligoni disegnati dall'utente è stata accolta con entusiasmo e ha portato a un'ulteriore richiesta: gli Storici desidererebbero avere la possibilità di tracciare ipotetici confini amministrativi sotto forma di poligoni sovrapposti alla cartografia di base. Questa nuova funzionalità sarà implementata non appena saranno state abilitate le funzionalità di scrittura nel database.

Una delle osservazioni sollevate dagli Storici durante la fase di sviluppo del prototipo riguardava l'estetica dell'interfaccia dell'applicazione web; nello specifico era stato richiesto di sostituire il carattere tipografico in quanto riconducibile al periodo gotico. La nuova proposta che prevede l'utilizzo di un carattere "neutro" è stata accettata. Qualche incertezza permane sul titolo dell'applicazione che potrebbe diventare "Geografie del Mezzogiorno basso medievale".

Tra gli sviluppi futuri si riporta che saranno integrati nel WebGIS nuovi strati informativi quali l'idrografia e il modello digitale del terreno dell'area in esame, così come l'Atlante di Rizzi Zannoni (1803) come nuova cartografia di base. L'integrazione di nuovi supporti cartografici potrebbe consentire di identificare, e quindi di georeferenziare, insediamenti ormai inglobati dai grossi centri urbani e pertanto non più presenti nella cartografia moderna.

4 CONCLUSIONI E SVILUPPI FUTURI

4.1 CONCLUSIONI

Il presente lavoro di Tesi si colloca nell'ambito del Progetto di Ricerca di Interesse Nazionale 2009 "Organizzazione del territorio, occupazione del suolo e percezione dello spazio nel Mezzogiorno medievale (secoli XIII – XV). Sistemi informativi per una nuova cartografia storica" che vede la partecipazione di quattro Unità di Ricerca³¹ per la progettazione e la realizzazione di nuovi strumenti per l'analisi e la consultazione dei dati storici.

Nello specifico, gli obiettivi di questo lavoro di Tesi possono essere così riassunti:

- progettazione e realizzazione di una base di dati georeferenziata per l'archiviazione di registrazioni esattoriali dello Stato Orsiano relative agli anni 1448 – 1460 estratte dal documento storico "Quaternus Declaracionum";
- progettazione e realizzazione con tecnologie web di un geoservizio per la consultazione e lo scambio di informazioni alfanumeriche e cartografiche tra le diverse Unità di Ricerca medievale.

L'approccio seguito nel corso dell'intero lavoro riconosce il ruolo centrale dei dati all'interno di un sistema informativo territoriale: proprio al fine di individuare le migliori soluzioni progettuali si è scelto di svolgere in prima istanza l'analisi dei requisiti e la definizione del database; solo successivamente è stata progettata e realizzata la componente tecnologica del geoservizio che consente agli storici un'agevole consultazione dei dati.

³¹ Le quattro Unità di Ricerca sono: l'Università Federico II di Napoli, capofila del progetto, l'Università del Salento, l'Università di Bologna e il Politecnico di Milano.

La fase di analisi e raccolta dei requisiti ha richiesto una stretta collaborazione con le Unità di Ricerca degli Storici per meglio individuare le caratteristiche dei dati, le relazioni tra di essi e le più comuni operazioni di analisi svolte dai ricercatori. Nello specifico, la struttura del database è stata progettata per archiviare dati di tipo storico, in cui la componente temporale è gestita secondo il sistema indizionale.

Nel corso della progettazione si è costantemente tenuto conto della possibilità di espansione futura dalla base di dati con l'inclusione di nuove fonti storiche di tipo fiscale e non: si è preferita ogni volta che fosse possibile una struttura modulare, al fine di mantenere estensibile e aggiornabile il database.

La centralità di concetti come flessibilità e modularità nel corso della fase di progettazione del database si sono riflessi anche nei passaggi successivi: per agevolare l'integrazione di nuovi dati e di funzionalità future sono state evitate tecnologie proprietarie in favore di soluzioni Open Source consolidate e molto diffuse, che garantiscono la presenza di aggiornamenti periodici, come il DBMS PostgreSQL con l'estensione PostGIS per l'integrazione della componente cartografica. Analogamente, le funzionalità di *web mapping* e l'interfaccia utente sono state realizzate con tecnologie e standard web maturi ed ampiamente documentati e che rendono semplice l'integrazione di nuove funzionalità e l'interoperabilità con altri sistemi.

Queste scelte consentiranno in futuro di estendere il progetto con nuovi moduli e funzionalità, ad esempio per la gestione dei metadati o delle immagini ottenute come scansioni dei documenti storici.

Se dal punto di vista tecnico si è cercato di ottimizzare le procedure in vista di una auspicabile crescita futura in termini di dati archiviati e di funzionalità dello strumento, per quanto riguarda la progettazione dell'interfaccia si è scelto di dare priorità alla semplicità d'utilizzo. Nel corso della fase di definizione dell'interfaccia l'applicazione web è stata progettata e realizzata con particolare riguardo all'usabilità

da parte degli utenti finali: sono stati realizzati e discussi prototipi dell'interfaccia grafica utili per selezionare solo le informazioni significative per gli utenti.

Per agevolare l'uso da parte di utenti non esperti nell'utilizzo di strumenti di questo tipo, sono state realizzate procedure che nascondono quando possibile gli aspetti più tecnici e complessi. Per questi motivi quest'applicazione web è indicata per essere utilizzata in ambito umanistico e in particolar modo in ambito storico.

L'organizzazione dei dati in un database e lo sviluppo dell'applicazione WebGIS per consultarli hanno consentito di affiancare ai dati in forma tabellare anche rappresentazioni di tipo cartografico, in grado di favorire nuove interpretazioni dei dati da parte degli Storici. Il prototipo del geoservizio è stato presentato all'Unità di Ricerca di Lecce, il cui riscontro è stato positivo. In particolare sono state apprezzate le due modalità di consultazione dei dati che hanno consentito di individuare le località che godevano di sgravi fiscali e centri obbligati a pagare un importo maggiore del dovuto (Pizzuto, 2009).

Al fine di rendere quanto più semplici possibile le operazioni di inserimento dati e manutenzione dell'applicazione web, è stata dedicata molta cura alla parametrizzazione delle logiche di funzionamento secondo cui vengono trattate all'interno dell'applicazione le varie tabelle/viste, i rispettivi campi e le varie proprietà secondo cui consentire o meno operazioni di filtro, ricerca e, in futuro, modifica dei dati. Praticamente ogni elemento dell'interfaccia utente è gestibile dal pannello di amministrazione del database storico attraverso tabelle di configurazione, riducendo al minimo la necessità di interventi a livello di codice.

4.2 SVILUPPI FUTURI

Gli Storici hanno dimostrato grande interesse verso una futura integrazione nel database di dati di tipo tributario provenienti da altre fonti storiche come ad esempio il *Liber Focorum* o il *Liber Decimae*. Alcuni dati relativi al Ducato di Milano sono

già stati integrati nel database storico parallelamente allo svolgimento di questo lavoro. Quest'operazione, pur comportando modifiche ad una parte della struttura della banca dati, non ha richiesto interventi sulle logiche funzionali o modifiche al codice: sarà sufficiente aggiornare le tabelle di configurazione del database per vedere rappresentati nell'interfaccia anche i nuovi dati. Auspicabilmente la banca dati continuerà a crescere con nuove integrazioni, rendendo questo strumento sempre più rilevante nell'ambito della Ricerca storica Medievale.

In una fase successiva, allo scopo di agevolare ulteriormente la condivisione delle informazioni tra le Unità di Ricerca degli Storici, saranno abilitate le funzionalità di modifica delle informazioni del database attraverso il geoservizio pubblicato: le informazioni alfanumeriche saranno editabili dall'apposita scheda dell'applicazione web mentre le informazioni cartografiche potranno essere modificate da appositi strumenti disponibili nella scheda "WebGIS".

Inoltre, un aspetto che in parte è già stato tenuto in considerazione nel corso di questo lavoro ma che sarà meglio definito nel prossimo futuro, sarà l'utilizzo di carte storiche (l'Atlante di Rizzi Zannoni del 1803 oppure la cartografia IGM dei primi anni del'900) come supporto cartografico del WebGIS. Ciò potrebbe consentire di identificare, e quindi di georeferenziare, insediamenti ormai inglobati dai grossi centri urbani e pertanto non più presenti nella cartografia moderna.

Un'ultima considerazione a chiusura di questo lavoro è la speranza che strumenti e soluzioni simili a quanto qui presentato possano rendere sempre più rapido, preciso ed intuitivo lo studio di dati georeferenziati, non solo in ambito storico ma anche per altre materie umanistiche. Questo lavoro vuole essere una dimostrazione delle potenzialità che lo strumento GIS può offrire a supporto della ricerca e, soprattutto, dei benefici in ambito scientifico di un approccio collaborativo ed interdisciplinare.

INDICE DELLE FIGURE

Figura 1.1 - Le cinque fasi di progettazione di un sistema informativo. La fase di progettazione dei dati è a sua volta suddivisa in tre fasi (Figura 1.2).	26
Figura 1.2 - Le tre fasi della progettazione di un database.....	27
Figura 1.3 – Entità con riferimenti spaziali e relativi attributi	36
Figura 1.4 – Entità con informazioni esclusivamente alfanumeriche e relativi attributi	36
Figura 1.5 – Modello concettuale del database storico.....	38
Figura 1.6 – Schema logico della base dati dei dati storici. In verde le tabelle “ponte” e in grassetto le chiavi primarie (o “primary key”) delle entità.....	41
Figura 1.7 – Schema fisico del data base dei dati storici realizzato con il software SQL Power Architect.....	46
Figura 2.1 – Componenti dell’architettura base di un’applicazione web.	53
Figura 2.2 – Schema di funzionamento delle chiamate AJAX nell’applicazione web “Geografie dell’Italia Medievale”.	55
Figura 2.3 – Mappa del sito che consentirà accesso e navigazione al database.	61
Figura 2.4 – Mockup della scheda “WebGIS”. L’area di lavoro può essere suddivisa in tre parti, una a sinistra (A), una centrale (B) e una a destra (C).	62
Figura 2.5 – Mockup della scheda “Mostra tabella”. Anche in questo caso l’area di lavoro può essere suddivisa in tre parti (D, E ed F).....	62
Figura 2.6 – I principali linguaggi di sviluppo web utilizzati per la realizzazione dell’applicazione web progettata.	64
Figura 3.1 – Interfaccia del geoservizio “Geografie dell’Italia Medievale (XIII – XV secolo)”.....	77

Figura 3.2 – Progetto QGIS in cui sono state definite le simbologie dei layer, le scale di visualizzazione e altre proprietà della mappa.....	77
Figura 3.3 – Scheda WebGIS. In senso antiorario: albero dei layer, legenda, riquadri di ricerca e corrispondente griglia dei risultati.	80
Figura 3.4 – Dettaglio degli strumenti nella modalità WebGIS. In senso antiorario: strumenti di zoom e pan, slider della scala, coordinate e scala, mappa di overview, stampa e ricerca località, strumenti di misura di distanze e aree, strumenti di interrogazione in mappa.	81
Figura 3.5 – Ricerca nella modalità WebGIS; a destra il pannello di ricerca con la griglia dei risultati, a sinistra il risultato evidenziato nel layer “luogo attuale”.	83
Figura 3.6 – Funzionalità di interrogazione sulla mappa. In rosso è evidenziato il menù a tendina che consente di selezionare la modalità di ricerca. A destra i risultati dell’interrogazione.....	84
Figura 3.7 – Proprietà del progetto QGIS. In evidenza l’opzione che permette di rendere visibili i risultati.....	85
Figura 3.8 –Scheda “Mostra tabella”.....	86
Figura 3.9 – Estratto della tabella “tables_settings”.....	87
Figura 3.10 – Estratto della tabella “fields_settings”.....	88
Figura 3.11 – Oltre alle funzioni di ordinamento crescente e decrescente è possibile aggiungere o nascondere campi della tabella/vista attiva.....	88
Figura 3.12 –Esempio di ricerca semplice.....	89
Figura 3.13 –Esempio di ricerca avanzata.....	90
Figura 3.14 – Componenti del pannello di ricerca avanzata.	94
Figura 3.15 – Array delle condizioni impostate nella ricerca di Figura 3.14.	96
Figura 3.16 – Diagramma di flusso della procedura di autenticazione.	97
Figura 3.17 – Passaggi fondamentali nella procedura di caricamento dell’applicazione web.....	98

INDICE DELLE TABELLE

Tabella 1.1 – Suite di tecnologie considerate: alternativa open source e ESRI.	29
Tabella 1.2 – Analisi dei vantaggi (in verde) e svantaggi (in arancione) tra una suite di GIS open source e la suite ESRI (ArcGIS Desktop, Server e ArcSDE).....	29
Tabella 1.3 – Elenco delle entità “spaziali” e relativa descrizione sintetica.....	33
Tabella 1.4 - Elenco delle entità “non spaziali” e relativa descrizione	34
Tabella 1.5 – Struttura dell’identificatore dell’entità Fonte	37
Tabella 1.6 – Elenco delle associazioni con cardinalità N:N	42
Tabella 2.1 – Riassunto delle tecnologie scelte per il lato server dell’applicazione web.....	73
Tabella 2.2 – Riepilogo delle tecnologie utilizzate per la realizzazione del“lato client”ovvero l’interfaccia del servizio.....	73
Tabella 3.1 – Operatori di confronto proposti, in dipendenza dal tipo di dato del campo selezionato.....	93

LISTA DEGLI ACRONIMI

AIAC	Associazione Internazionale di Archeologia Classica
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface (Interfaccia di Programmazione di un'Applicazione)
ArcSDE	ESRI Spatial Database Engine
BSD	Berkeley Software Distribution
CGI	Common Gateway Interface
CSS	Cascading Style Sheets
CSV	Comma Separated Values (tipologia di formato dati)
DB	Data Base
DBMS	Data Base Management System
ED50	European Datum 1950
EPSG	European Petroleum Survey Group
ER	Entità – Relazioni (diagramma)
ESRI	Environmental Systems Research Institute
GBHGIS	Great Britain Historical Geographical Information System
GIS	Geographical Information System
GNU	General Public License
GPL	General Public System
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HW	Hardware
IT	Information Technology
JS	JavaScript (linguaggio di programmazione)
JSON	JavaScript Object Notation
LAMOP	Laboratorio di Studi Medievali Occidentali
LTS	Long Term Support
MVC	Model-view-controller
OGC	Open Geospatial Consortium
OGP	Oil & Gas Producers

OS	Sistema Operativo
OSS	Open-source software
PHP	Hypertext Preprocessor (in origine “Personal Home Page”)
PRIN	Progetto di Ricerca Nazionale
SHP	Shapefile (estensione di un file)
SQL	Structured Query Language
SRID	Spatial Reference System Identifier
SW	Software
UTM	Universal Transverse of Mercator
WEB	World Wide Web
WFS	Web Service Feature
WGS84	World Geodetic System 1984
WKB	Well-Known Binary
WKT	Well-known text
WMS	Web Map Service
WYSIWYG	What-You-See-Is-What-You-Get
XML	eXtensible Markup Language

BIBLIOGRAFIA

Agosti M., Benfante L. e Melucci M. (1997). OFAHIR: “On-the-Fly” Automatic Authoring of Hypertexts for Information Retrieval. In S. Spaccapietra, F. Maryansky (Eds), *Searching for Semantics: Data Mining, Reverse Engineering*. Proc. of the 7th IFIP 2.6 Working Conference on Database Semantics (DS-7), Leysin, Svizzera, 7-10 Ottobre 1997.

Ardissone P. e Rinaudo F. (2005). A GIS for the Management of Historical and Archaeological Data. *Proceedings of the CIPA 2005, XX International Symposium*, Torino, Italia, 26 Settembre – 1 Ottobre 2005.

Atzeni P., Ceri S., Paraboschi S. e Torlone R. (1999). *Basi di dati. Modelli e linguaggi di programmazione*, Milano, Mc Graw Hill.

Atzeni P., Ceri S., Fraternali P., Paraboschi S. e Torlone R. (2003). *Basi di dati. Architetture e linee di evoluzione*. Milano, Mc Graw Hill.

Baiocchi V. e Lelo K. (2002). Confronto di cartografie storiche con cartografie attuali per l'area del centro storico di Roma. *Atti della VI Conferenza Nazionale ASITA*, Perugia, Italia, 5-8 Novembre 2002.

Balletti C. (1998). An Informative System for the Management of Historical Maps. 5th Hellenic Cartographic Conference “Large Scale Cartography: City Maps”, XEEE, Salonicco, 25-27 Novembre 1998.

Balletti C., Guerra F. e Monti C. (2000). Analytical Methods and New Technologies for Geometrical Analysis and Georeferenced visualization of Historical Maps. *International Archives of Photogrammetry and Remote Sensing*, Lubiana, vol. XXXII, part 6W8/1, ISPRS WG VI/3 e IV/3, pp. 1-7.

- Balletti C. (2000). Analytical and Quantitative Methods for the Analysis of the Geometrical Content of Historical Cartography. *International Archives of Photogrammetry and Remote Sensing*, Amsterdam, vol. XXXIII, part B5, pp. 30 - 37.
- Barzaghi, R. e Carrion, D. (2011). Il GIS come strumento di ricerca: il contributo della cartografia. *Ricerche storiche*, Firenze, vol. 1, pp.45-59.
- Berman L. M. (2005). Boundaries or Networks in Historical GIS: Concepts of Measuring Space and Administrative Geography in Chinese History. *Historical Geography*, Special issue: Emerging trends in Historical GIS, Baton Rouge, USA vol. 33, pp. 118-133.
- Burrough P. (1986). *Principles of geographical information systems for land resource assessment*. Oxford, U.K, Clarendon Press.
- Burrough P. e McDonnell R. (1998). *Principles of geographical information systems*. Oxford University Press.
- Bulgarelli Lukacs A. (1993). L'imposta diretta nel Regno di Napoli in età moderna, Milano, Franco Angeli.
- Bulgarelli Lukacs A. (1996). Coscienza e controllo della periferia attraverso lo strumento fiscale: l'esperienza del Regno di Napoli, in *Comunità e poteri centrali negli antichi stati italiani: alle origini dei controlli amministrativi*, a cura di Mannori L., Napoli, pp. 242-275.
- Bulgarelli Lukacs, A. (1997). Il Sannio in età moderna: fisco ed economia in *Archivio Storico del Sannio*, Napoli, Ed. Scientifiche Italiane, II, n. 1/97, pp. 53-72.
- Carrion, D., Cengarle, F., Somaini, F. (2008). Un approccio alla georeferenziazione di una carta antica. *Atti della 12a Conferenza nazionale ASITA*, L'Aquila, 21 - 24 Ottobre 2008, pp.655-660.
- Carrion, D. e Migliaccio, F. (2009). Il Principato di Taranto in un GIS: problemi e potenzialità. In *Domini del principe di Taranto in età orsiniana (1399-1463)* a cura di Somaini F. e Vetere B. Galatina (LE), Italy, Congedo Editore, pp. 37-60.

- Chen P.P.-S. (1976). The Entity-Relationship Model: Toward a Unified View of Data. *ACM on Database Systems, Special issue: papers from the international conference on very large data bases, Framingham, USA, Settembre 22–24, 1975, Vol. 1 n.1*, pp. 9-36.
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM, New York, USA, Vol. 13, n.6*, pp. 377-387.
- Comba R. e Naso I. (1994). *Demografia e società nell'Italia medievale (secoli IX-XIV)*. Cuneo, Soc. Studi Stor. Archeologici, Da Cuneo all'Europa.
- Fu P. e Sun J. (2011). *WebGIS, Principles and applications*. Esri Press. ISBN 978-1-58948-245-6.
- Ginatempo M. e Sandri L. (1990). *L'Italia delle città. Il popolamento urbano tra Medioevo e Rinascimento (secoli XII-XV)*. Firenze, Le Lettere, collana Le vie della storia.
- Gregory I.N., Bennet C., Gilham V.L. e Southall H.R. (2002). The Great Britain Historical GIS: From Maps to Changing Human Geography. *The Cartographic Journal*. Vol. 39 n.1, 2002, pp. 37-49.
- Gregory I.N. (2003). *A Place in History. A Guide to Using GIS in Historical Research*. Oxford, Oxbow Books.
- Gregory I.N. e Ell P. (2008). *Historical GIS: Technologies, Methodologies, and Scholarship*. Cambridge University Press, Studies in Historical Geography.
- Grosso E., Bouju A. e Mustière S. (2009). Data Integration GeoService: A First Proposed Approach Using Historical Geographic Data. *W2GIS, vol. 5886 of Lecture Notes in Computer Science, Maynooth, Irlanda, Springer*, pp. 103-119.
- Guttman, A. (1984). R-Trees: A Dynamic Index Structure for Spatial Searching. *Proceedings of the 1984 ACM SIGMOD international conference on Management of data - SIGMOD '84* .doi:10.1145/602259.602266. ISBN 0897911288. pp. 47.
- Hazard E. (2011). *Getting Started with Openlayers*, Packt Publishing - Birmingham Mumbai.

- Henderson M., Berman L. M. (2003). *Methods for Space-Time Analysis: Examples form CHGIS*. ESRI Users Conference, San Diego, Luglio, 2003.
- Jardine D.A. (1977). *The ANSI/SPARC DBMS Model*. Co. Montreal, Canada. North-Holland Pub.
- Knowles, A.K. (2000). Introduction: Historical GIS, the spatial turn in Social Science History. *Madinson, USA, Social Science History* vol. 24, n.3, pp. 451-470.
- Knowles A.K. (2002). *Past Time, Past Place: GIS for history A collection of twelve case studies on the use of GIS in historical research and education*. Redlands, USA, ESRI Press, xx, pp. 202.
- Krug S. (2006). *Don't make me think. Un approccio di buon senso all'usabilità web*. Milano, Tecniche Nuove, Hops-Tecnologie.
- Laurini R. e Thompson D. (1992). *Fundamentals of Spatial Information Systems*. Academic Press, New York.
- Laurini, R. (2001). *Information Systems for Urban Planning: A Hypermedia Cooperative Approach*. Londra, Gran Bretagna, Taylor and Francis.
- Longley P.A., Goodchild M.F., Maguire D.J. e Rhind, D.W. (2005) *Geographic Information Systems and Science*. Chichester: Wiley.
- Mainoni P. (2001). *Politiche finanziarie e fiscali nell'Italia settentrionale (secoli XIII-XV)*, Milano, Unicopli.
- Manolopoulos Y., Nanopoulos A. e Theodoridis Y. (2006). *R-Trees: Theory and Applications*. Series in Advanced Information and Knowledge Processing, Springer.
- Meyer E.A. (2011). *CSS – Guida pocket*. Milano, Tecniche Nuove, O'Reilly.
- Migliaccio F. (2008). *Sistemi informativi territoriali e cartografia*. Milano, Maggioli Editore. Politecnica.
- Mogorovich P. e Mussio P. (1988). *Automazione del Sistema Informativo territoriale. Elaborazione Automatica dei Dati Geografici*. Milano, Masson, vol.2 pp.503-8.
- Monti C. (2011). *La cartografia dall'antichità fino al XVIII secolo*. Milano, Maggioli Editore. Politecnica.

- Murgante B. (2008). *L'informazione geografica a supporto della pianificazione territoriale*. Milano, Franco Angeli.
- Nielsen J. (2000). *Web usability*. Milano, Apogeo.
- Noizet H., Dallo A., Blary G.-X., Costa L. e Pouget F. (2008). ALPAGE: towards the setting-up of a collaborative tool, *Archeologia e Calcolatori* n. XIX, Firenze, Edizioni All'Insegna del Giglio, pp. 87-102.
- Noizet H., Grosso E. (2011). The ALPAGE project: Paris and its suburban area at the intersection of history and geography (9th-19th century). In *Digital proceedings of the 25th International Cartographic Conference (ICC'11)*, Parigi, Francia.
- O'Sullivan D. e Unwin D (2002) *Geographic Information Analysis*, New York, USA, John Wiley & Sons.
- Olaya V. (2012). *Sistemas de Información Geográfica*. Madrid, Spagna, Bubok.
- Ott, T. e Swiaczny, F. (2001). *Time-integrative GIS. Management and analysis of spatio-temporal data*. Berlin / Heidelberg / New York. Springer.
- Pilgrim M. (2010). *HTML5 – Guida operativa*, Milano, Tecniche Nuove, O'Reilly.
- Pizzuto S. (2009). Il Quaternus declaracionum di Francesco de Agello (1450-1461). Un contributo allo studio della geografia politica del Principato di Taranto in età orsiniana. In Somaini F., Vetere B. *I domini del Principe di Taranto in età orsiniana (1399-1463)*. Galatina, Congedo, pp. 61-76.
- Polillo R. (2010). "Facile da usare", Milano, Apogeo.
- Rubio-Campillo, X. (2009). Teoria de jocs aplicada a la historia de la guerra moderna. De Balaguer a Almenar, 1710. *Manuscripts. Revista d'Història Moderna* pp. 143–160.
- Teorey, T.; Lightstone, S. e Nadeau, T.(2005). *Database Modeling & Design: Logical Design*. Burlington, Gran Bretagna, Morgan Kaufmann Press.
- Tomlin C.D. (1990). *Geographic information systems and cartographic modeling*. New Jersey, Prentice-Hall, Englewood Cliffs.

Worboys M. e Duckham M. (2004). GIS: a computing perspective. Boca Raton, USA, CRC Press.

SITOGRAFIA

ArcGIS for Server versus Open Source

<http://mappinggis.com/2012/07/13/arcgis-for-server-vs-open-source/>

Introduzione alla teoria dei database relazionali, SQL e ambiente MySQL

<http://ennebi.solira.org/>

Documentazione ExtJS

<http://www.dtraining.info/component/content/article/37-il-blog-di-dtraining/80-una-introduzione-a-extjs>

Documentazione GeoExt

<http://www.geoext.org/>

Documentazione linguaggio PHP per PostgreSQL

<http://php.net/manual/en/book.pgsql.php>

Documentazione OpenLayers

<http://openlayers.org/>

Oltre MapServer

http://www.rigacci.org/wiki/doku.php/tecnica/gps_cartografia_gis/beyond_mapserver

OpenLayers API

<http://dev.openlayers.org/releases/OpenLayers-2.10/doc/apidocs/files/OpenLayers-js.html>

OpenLayers o Leaflet

<http://mappinggis.com/2012/12/05/openlayers-o-leaflet/>

ORBIS - The Stanford Geospatial Network Model of the Roman World

<http://orbis.stanford.edu/>

Por que utilizar OpenLayers y GeoExt

<http://mappinggis.com/2012/11/30/por-que-utilizar-openlayers-y-geoext/>

Progettazione di un'interfaccia grafica

<http://www.urp.it/cpusabile/index59f0.html>

Progetto ALPAGE

<http://alpage.tge-adonis.fr/index.php/fr/>

Progetto China HGIS

<http://www.fas.harvard.edu/>

Progetto Fasti Online

<http://www.fastionline.org/index.php?view=home>

Progetto Great Britain Historical Geographical Information System (GBHGIS)

<http://www.visionofbritain.org.uk/>

Progetto QGIS Web Client

<https://github.com/qgis/qgis-web-client>

QGIS Server – A WMS for the masses

<http://linfiniti.com/2010/08/qgis-mapserver-a-wms-server-for-the-masses/>

QGIS Server Tutorial

http://hub.qgis.org/projects/quantum-gis/wiki/QGIS_Server_Tutorial

Rappresentazione degli spostamenti di Ouyi Zhixu

http://www.fas.harvard.edu/~chgis/work/docs/papers/CGA_Wkshp2009_Lex_9apr09.pdf

WebGIS del Catasto Gregoriano

<http://www.dipsuwebgis.uniroma3.it/index.html>.

WebGIS in Archeologia

<http://www.archeoserver.it/space/Metodologie/WebGIS/WebGIS+in+Archeologia>

Wikipedia - l'Enciclopedia Libera

<http://it.wikipedia.org>

Workshop PostGIS – Indicizzazione spaziale

<http://www.padido.eu/gfoss/postgis/indexing.html>

APPENDICE A

ELENCO COMPLETO DEI CAMPI SIGNIFICATIVI DELLA STRUTTURA TABELLARE DI ARCHIVIAZIONE DEI DATI PROVENIENTI DAL QUATERNUS DECLARACIONUM

NOME TOPONIMO ITALIANO: Nome attuale del toponimo ove si localizza l'insediamento medievale.

NOME TOPONIMO LATINO: Nome latino del toponimo, riportato nella forma grafica con cui viene indicato nel documento in esame.

NOME CITTA' ATTUALE: Nome della città a cui appartiene il toponimo (nei casi in cui si tratti di siti a continuità di vita, tale nome viene a coincidere con il nome del toponimo italiano).

PROVINCIA: Provincia di appartenenza della città, indicata con la sigla, espressa con due lettere maiuscole.

APPARTIENE A: Nei casi in cui il centro dipenda fiscalmente da un altro centro si indica il toponimo di appartenenza

COORDINATE X: Coordinate geografiche piane, espresse da codice numerico di sei cifre (esempio: 768.580). Tali coordinate sono funzionali alla georeferenziazione dei centri presi in esame.

COORDINATE Y: Coordinate geografiche piane, espresse da codice numerico di sette cifre (esempio: 4.458.413). Tali coordinate sono funzionali alla georeferenziazione dei centri presi in esame.

COEFFICIENTE DI PRECISIONE: Coefficiente attribuito alle coordinate, relativo al grado di precisione raggiunto nella georeferenziazione.

- Toponimi con scarsissime notizie: Scarso grado di precisione;
- Toponimi incerti: Basso grado di precisione;
- Città: Medio grado di precisione;
- Piccoli agglomerati abitativi: Alto grado di precisione;
- Edifici isolati: Massimo grado di precisione.

ID FONTE: Codice alfanumerico identificativo della fonte in esame. Di solito tale sigla è composta dall'indicazione di archivio, segnatura e numero del documento. (esempio: ASN DS II Reg. 249 ovvero Archivio di Stato di Napoli, Diversi Sommaria, II numerazione, Registro 249). In alcuni casi tale sigla è seguita da una virgola e dall'indicazione delle carte in cui si trova l'informazione registrata nel record.

TIPOLOGIA FONTE: Indica la tipologia della fonte esaminata:

- Registro erariale
- Libro di fuochi

ANNO: Codice numerico recante l'indicazione dell'anno a cui fa riferimento la fonte. Nel caso di anno indizionale si indica l'anno solare su cui si estende maggiormente l'anno indizionale.

CODICE: Codice numerico binario (**0/1**) indicante la tipologia del computo annuale:

- **0:** Anno Solare (1 Gennaio – 31 Dicembre);
- **1:** Anno Indizionale (1 settembre – 31 agosto anno solare successivo).

MESE: Ove presente, viene riportato il mese relativo alla registrazione dei dati elencati nel record.

GIORNO: Codice numerico indicante, ove presente, il giorno relativo alla registrazione dei dati presenti nel record.

INDIZIONE: Codice numerico compreso tra **1** e **15** indicante il numero di Indizione; tale codice esplicita l'anno specifico all'interno del periodo Indizionale di durata quindicennale.

Numero dei FUOCHI FISCALI: Codice numerico indicante il numero dei fuochi imponibile, vale a dire, il numero di fuochi che effettivamente versava al principe l'imposta diretta. Ai fuochi censiti attraverso le numerazioni venivano sottratti i fuochi esenti (nobili ed ecclesiastici). Ciascuna comunità poteva ottenere dal principe di essere tassata per un minor numero di fuochi.

FUOCHI de corpore: Codice numerico indicante il numero dei fuochi imponibile, vale a dire, il numero di fuochi che effettivamente versava al principe l'imposta diretta. Ai fuochi censiti attraverso le numerazioni venivano sottratti i fuochi esenti (nobili ed ecclesiastici). Ciascuna comunità poteva ottenere al principe di essere tassata per un minor numero di fuochi.

TASSA FOCATICO: Codice numerico indicante l'importo complessivo, espressa in grana, dell'imposta sui fuochi versata dal centro al principe.

Il focatico corrisponde all'imposta diretta introdotta da Alfonso d'Aragona con la riforma fiscale del 1443. Tale sistema era fondato sull'imposizione di 1 ducato d'oro, pari a 5 tarì e 10 grana, per ogni fuoco.

COLLETTA: Codice numerico indicante l'entità complessiva, espressa in grana, dell'imposta diretta versata dal centro al principe.

La colletta era la forma di prelievo fiscale utilizzata nel periodo angioino, dapprima richiesta come imposta straordinaria, con Carlo I assunse un carattere fisso e annuale. Fu sostituita dal focatico, in seguito alla riforma fiscale attuata da Alfonso d'Aragona, ma all'interno del Principato di Taranto, le comunità potevano essere sottoposte all'una o all'altra forma di prelievo fiscale.

THOMOLI SALE: Codice numerico indicante la quantità complessiva di tomoli di sale acquistata da ciascun centro.

TASSA MILIZIA: Codice numerico indicante l'entità complessiva, espressa in grana, della tassa per le milizie versata dal centro al principe.

DONATIVO: Codice numerico indicante l'entità complessiva, espressa in grana, della tassa "Pro dono consueto" versata dal centro al principe.

TAVERNA: Codice numerico indicante l'entità complessiva, espressa in grana, della tassa "pro iure tabernae", ovvero il diritto di gestire taverne.

DISTRETTO ERARIALE: Codice numerico (1/2/3/4) indicante l'Area Fiscale di appartenenza. **Terra d'Otranto** risulta divisa in **tre** distretti:

1. Da Taranto a Lecce: sede dell'erario del distretto: **TARANTO;**

2. Da Lecce a Leuca: sede dell'erario del distretto: **LECCE;**

3. Distretto di Soletto: sede dell'erario del distretto: **SOLETO**

4. Distretto di Bari: sede dell'erario del distretto: **BARI**

NOME LATINO DELL'ERARIO DEL DISTRETTO: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi in latino, riportati nella forma grafica emersa dal documento.

NOME ITALIANO DELL'ERARIO DEL DISTRETTO: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi nella trasposizione italiana del nome riportato nella fonte.

NOME LATINO DELL'ERARIO LOCALE: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi in latino, riportati nella forma grafica emersa dal documento.

AB: NOME ITALIANO DELL'ERARIO LOCALE: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi nella trasposizione italiana del nome riportato nella fonte.

AC. NUMERO BAIULI: Codice numerico indicante il numero dei baiuli che reggono la BAGLIVA nell'anno di riferimento della fonte.

AD: SEDE CAPITANIALE: Codice numerico binario (0/1) indicante l'Assenza/Presenza di sede capitanale.

- **0:** NO – Sede capitaneale assente;
- **1:** SI – Sede capitaneale presente.

AE. ELENCO LOCALITA' DEL CAPITANATO: Elenco località facenti parte della stessa giurisdizione capitaneale.

AF. NOME LATINO DEL CAPITANO: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi in latino, riportati nella forma grafica emersa dal documento.

AG. NOME ITALIANO DEL CAPITANO: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi nella trasposizione italiana del nome riportato nella fonte.

AH. CENTRO DI VICARIE: Codice numerico binario (0/1) indicante l'Assenza/Presenza di centro vicariale.

- **0:** NO – Sede Vicariale assente;
- **1:** SI – Sede Vicariale presente.

AI. NOME LATINO DEL VICARIO: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi in latino, riportati nella forma grafica emersa dal documento.

AJ. NOME ITALIANO DEL VICARIO: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi nella trasposizione italiana del nome riportato nella fonte.

AK. SEDE DI TESORERIA: Codice numerico binario (0/1) indicante l'Assenza/Presenza di sede di Tesoreria.

- **0:** NO – Sede di Tesoreria assente;
- **1:** SI – Sede di Tesoreria presente.

AL. TESORIERE DI RIFERIMENTO: Nel caso in cui nel centro in esame non sia presente la Tesoreria, si indica la sede del Tesoriere di riferimento.

AM. NOME LATINO DEL TESORIERE GENERALE: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi in latino, riportati nella forma grafica emersa dal documento.

AN. NOME ITALIANO DEL TESORIERE GENERALE: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi nella trasposizione italiana del nome riportato nella fonte.

AO. NOME LATINO DEL TESORIERE LOCALE: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi in latino, riportati nella forma grafica emersa dal documento.

AP. NOME ITALIANO DEL TESORIERE LOCALE: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi nella trasposizione italiana del nome riportato nella fonte.

AQ. SEDE DI CASTELLANIA: Codice numerico binario (0/1) indicante l'Assenza/Presenza di sede di Castellania.

- **0:** NO – Sede di Castellania assente;
- **1:** SI – Sede di Castellania presente.

AR. NOME LATINO DELCASTELLANO: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi in latino, riportati nella forma grafica emersa dal documento.

AS. NOME ITALIANO DEL CASTELLANO: Codice alfabetico indicante Cognome e provenienza seguito da virgola e nome, espressi nella trasposizione italiana del nome riportato nella fonte.

AT. SEDE DI DOGANA: Codice numerico binario (0/1) indicante l'Assenza/Presenza di sede di Dogana.

- **0:** NO – Sede Dogananale assente;
- **1:** SI – Sede Dogananale presente.

AU. NUMERO DOGANIERI: Codice numerico indicante il numero dei doganieri operanti nel centro nell'anno di riferimento della fonte.

AV. SEDE DI FONDACO: Codice numerico binario (0/1) indicante l'Assenza/Presenza di sede di Fondaco.

- **0:** NO – Sede di Fondaco assente;
- **1:** SI – Sede di Fondaco presente.

AW. BAGLIVA: Codice numerico indicante l'entità complessiva, espressa in grana, dei proventi ricavati dalla Bagliva nell'anno di riferimento del documento in questione.

AX. PLATEA: Codice numerico indicante l'entità complessiva, espressa in grana, dello "ius plateae" versata al principe per il diritto sul commercio.

AY. AFFIDA – PROVENTI PARZIALI: Codice numerico indicante l'entità complessiva, espressa in grana, della tassa "Pro Iure Affidae" versata dal centro in esame a quello avente tale diritto, in quanto fornitore di pascolo. [...]

AZ. LOCALITA' – A CHI PAGANO: Nome della località, seguita da trattino e nome del funzionario a cui il centro in esame versa l'imposta. (sono indicati sia la località che il soggetto al quale si paga)

BA. CENTRO DI AFFIDA: Codice numerico binario (0/1) indicante l'Assenza/Presenza di sede Centro di Affida.

- **0:** NO – Centro di Affida assente;
- **1:** SI – Centro di Affida presente.

AFFIDA TOTALE: Codice numerico indicante l'entità complessiva, espressa in grana, della tassa "Pro Affida" incamerata dal centro avente tale diritto.

APPENDICE B

DIZIONARIO DEI DATI

ENTITÀ: fonte

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_fonte	Codice identificativo della fonte esaminata	Codice alfanumerico
cod_fonte	Codice composto della fonte esaminata	Codice alfanumerico, rappresentato da una sigla composta da: archivio, fondo, num_pezzo, num_carta, r_v Esempio: ASN-DS II-248-112v ASN = Archivio di Stato di Napoli, DS II = Diversi Sommaria II, 248 = Registro 248, 112 = carta 112, v = verso
archivio	Indicazione per esteso del nome dell'archivio in cui si trova il documento esaminato	Stringa alfabetica Es. Archivio di Stato di Napoli

sezione	Indicazione per esteso dell'eventuale sezione in cui si trova il documento esaminato	Stringa alfabetica Es. Sezione di Archivio di Stato di Assisi
fondo	Indicazione per esteso del fondo a cui appartiene il documento esaminato	Stringa alfabetica Es. Diversi Sommaria II numerazione
num_pezzo	Numero del documento	Numero intero che si riferisce al documento esaminato
num_carta	Numero della carta o del fascicolo appartenente alla fonte in esame	Numero intero
bis_ter	Indicazione aggiuntiva su num_carta	Codice alfabetico: bis, ter, quater, ...
r_v	Indicazione del lato della carta definita da num_carta	Codice alfabetico: r = recto v = verso rv = recto e verso
tipo_fonte	Tipologia della fonte esaminata	Stringa alfabetica corrispondente ad esempio a: Registro erariale Libro di fuochi Quaternus Declaracionum

ENTITÀ: distretto_erariale

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_distretto_erariale	Codice indicante il Distretto Erariale di appartenenza del luogo	<p>Codice numerico a 5 cifre: la prima cifra attualmente è compresa fra 1 e 4; i valori da 1 a 3 riguardano i tre Distretti di Terra d'Otranto, 4 è riferito a Bari:</p> <p>Da Taranto a Lecce (sede dell'erario del distretto: TARANTO);</p> <p>Da Lecce a Leuca (sede dell'erario del distretto: LECCE);</p> <p>Distretto di Soleto (sede dell'erario del distretto: SOLETO);</p> <p>Distretto di Bari (sede dell'erario del distretto: BARI).</p> <p>Le successive 4 cifre riguardano l'anno in cui il distretto erariale è citato nella fonte.</p>
anno	Indicazione dell'anno solare a cui fa riferimento la fonte.	Numero intero di quattro cifre

sede_erario	<p>Nota: nel caso di anno indizionale si indica l'anno solare su cui si estende maggiormente l'anno indizionale</p> <p>Sede dell'erario del Distretto = 1 – Sede = distretto Taranto Distretto = 2 – Sede = Lecce Distretto = 3 – Sede = Soletto Distretto = 4 – Sede = Bari</p>
-------------	---

ENTITÀ: luogo_attuale

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_luogo_attuale	Codice identificativo del luogo attuale corrispondente al toponimo italiano	Codice numerico
toponimo_ita	Nome attuale del toponimo ove si localizza il luogo medievale. Le coordinate sono riferite a questo attributo	Stringa alfabetica
est_x	Coordinata cartografica metrica Est UTM WGS84 (fuso 33) per la georeferenziazione degli insediamenti presi in esame	Valore numerico espresso in metri

nord_y	<p>Coordinata cartografica Valore numerico espresso in metrica Nord UTM WGS84 metri (fuso 33) per la georeferenziazione degli insediamenti presi in esame</p>
grado_affidabilita	<p>Grado di affidabilità Coefficiente numerico in una associato alla scala ordinale di valori da 1 a georeferenziazione del 5: toponimo italiano, 1: Toponimi con scarsissime definito da un coefficiente notizie: numerico in una scala <u>scarso</u> grado di precisione; ordinale di valori 2: Toponimi incerti: <u>basso</u> grado di precisione; 3: Città: <u>medio</u> grado di precisione; 4: Piccoli agglomerati abitativi: <u>alto</u> grado di precisione; 5: Edifici isolati: <u>elevato</u> grado di precisione.</p>

ENTITÀ: luogo

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_luogo	Codice identificativo del luogo corrispondente al toponimo latino	Codice alfanumerico
id_luogo_attuale	Codice identificativo del luogo attuale corrispondente al toponimo italiano	Codice numerico
id_fonte	Codice identificativo della fonte esaminata	Codice alfanumerico
cod_fonte	Codice alfanumerico della fonte esaminata	Codice alfanumerico, rappresentato da una sigla composta da: archivio, fondo, num_pezzo, num_carta, r_v Esempio: ASN-DS II-248-112v ASN = Archivio di Stato di Napoli, DS II = Diversi Sommaria II, 248 = Registro 248, 112 = carta 112, v = verso
id_distretto_erariale	Codice indicante il Distretto Erariale di appartenenza del luogo	Codice numerico a 5 cifre: la prima cifra attualmente è compresa fra 1 e 4; i valori da 1 a 3 riguardano i tre Distretti di Terra d'Otranto, 4 è riferito a

		<p>Bari:</p> <p>Da Taranto a Lecce (sede dell'erario del distretto: TARANTO);</p> <p>Da Lecce a Leuca (sede dell'erario del distretto: LECCE);</p> <p>Distretto di Soleto (sede dell'erario del distretto: SOLETO);</p> <p>Distretto di Bari (sede dell'erario del distretto: BARI).</p> <p>Le successive 4 cifre riguardano l'anno in cui il distretto erariale è citato nella fonte.</p>
toponimo_lat	Nome latino del toponimo, riportato nella forma grafica con cui viene indicato nel documento in esame	Stringa alfabetica
toponimo_ita	Nome attuale del toponimo ove si localizza il luogo medievale. Le coordinate sono riferite a questo attributo	Stringa alfabetica
comune_attuale	Nome del Comune a cui appartiene il toponimo (nei	Stringa alfabetica

	<p>casi in cui si tratti di siti a continuità di vita, tale nome viene a coincidere con il nome del toponimo italiano)</p>	
provincia	<p>Sigla provinciale della Provincia di appartenenza del Comune</p>	Due lettere maiuscole
est_x	<p>Coordinata cartografica metrica Est UTM WGS84 (fuso 33) per la georeferenziazione degli insediamenti presi in esame</p>	Valore numerico espresso in metri
nord_y	<p>Coordinata cartografica metrica Nord UTM WGS84 (fuso 33) per la georeferenziazione degli insediamenti presi in esame</p>	Valore numerico espresso in metri
grado_affidabilita	<p>Grado di affidabilità che si presume sia possibile associare alla georeferenziazione del toponimo italiano, definito da un coefficiente numerico in una scala ordinale di valori</p>	<p>Coefficiente numerico in una scala ordinale di valori da 1 a 5: 1: Toponimi con scarsissime notizie: <u>scarso</u> grado di precisione; 2: Toponimi incerti: <u>basso</u> grado di precisione; 3: Città:</p>

		<p><u>medio</u> grado di precisione;</p> <p>4: Piccoli agglomerati abitativi:</p> <p><u>alto</u> grado di precisione;</p> <p>5: Edifici isolati:</p> <p><u>elevato</u> grado di precisione.</p>
anno	Indicazione dell'anno solare a cui fa riferimento la fonte.	Numero intero di quattro cifre
	Nota: nel caso di anno indizionale si indica l'anno solare su cui si estende maggiormente l'anno indizionale	
mese	Indicazione del mese a cui fa riferimento la fonte.	Numero di due cifre
giorno	Indicazione del giorno a cui fa riferimento la fonte.	Numero di due cifre
tipo_anno	Tipologia del computo annuale	<p>Codice numerico di una cifra:</p> <p>0 = Anno Solare (1 Gen. – 31 Dic.);</p> <p>1 = Anno Indizionale (1 Set. – 31 Ago. dell'anno solare successivo).</p> <p>Bisognerà tenere conto di tutte le diverse tipologie di anno solare e anno indizionale.</p>
indizione	Numero di Indizione: definisce l'anno specifico	Numero intero compreso tra 1 e 15.

fuochi_fiscali	<p>all'interno del periodo indizionale di durata quindicennale</p> <p>Numero dei fuochi fiscali, Valore numerico intero ovvero numero dei fuochi imponibili censiti.</p> <p>Nota: ai fuochi censiti attraverso le numerazioni venivano sottratti i fuochi esenti (nobili ed ecclesiastici). Ciascuna comunità poteva ottenere dal principe di essere tassata per un minor numero di fuochi</p>
fuochi_paganti	<p>Numero dei fuochi che Valore numerico intero effettivamente versavano al principe l'imposta diretta.</p>

ENTITÀ: unita_fiscale

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_unita_fiscale	<p>Codice identificativo del centro di appartenenza fiscale</p> <p>Nota. Per quanto riguarda il centro fiscale, un luogo può fare riferimento: Al luogo stesso che è anche centro fiscale, a un altro luogo che è centro fiscale, a più insediamenti che sono centri fiscali (fra cui può esserci il luogo stesso)</p>	Codice numerico
id_distretto_erariale	Codice indicante il Distretto Erariale di appartenenza del luogo	<p>Codice numerico a 5 cifre: la prima cifra attualmente è compresa fra 1 e 4; i valori da 1 a 3 riguardano i tre Distretti di Terra d'Otranto, 4 è riferito a Bari:</p> <p>Da Taranto a Lecce (sede dell'erario del distretto: TARANTO);</p> <p>Da Lecce a Leuca (sede</p>

		<p>dell'erario del distretto: LECCE);</p> <p>Distretto di Soleto (sede dell'erario del distretto: SOLETO);</p> <p>Distretto di Bari (sede dell'erario del distretto: BARI).</p> <p>Le successive 4 cifre riguardano l'anno in cui il distretto erariale è citato nella fonte.</p>
toponimo_ita	<p>Nome attuale del toponimo ove si localizza l'unità fiscale. Le coordinate sono riferite a questo attributo</p>	Stringa alfabetica
est_x	<p>Coordinata cartografica metrica Est UTM WGS84 (fuso 33) per la georeferenziazione degli insediamenti presi in esame</p>	<p>Valore numerico espresso in metri</p>
nord_y	<p>Coordinata cartografica metrica Nord UTM WGS84 (fuso 33) per la georeferenziazione degli insediamenti presi in esame</p>	<p>Valore numerico espresso in metri</p>

anno	Indicazione dell'anno a cui fa riferimento la fonte	Numero di quattro cifre
tesoreria		Booleano (0 / 1)

ENTITÀ: ufficio

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_ufficio	Codice identificativo di un ufficio per il quale si sono trovate informazioni in una certa fonte	Codice numerico
id_luogo	Codice identificativo del luogo corrispondente al toponimo latino	Codice alfanumerico
id_fonte	Codice identificativo della fonte esaminata	Codice alfanumerico
id_distretto_erariale	Codice indicante il Distretto Erariale di appartenenza del luogo	Codice numerico a 5 cifre: la prima cifra attualmente è compresa fra 1 e 4; i valori da 1 a 3 riguardano i tre Distretti di Terra d'Otranto, 4 è riferito a Bari: Da Taranto a Lecce (sede dell'erario del distretto: TARANTO); Da Lecce a Leuca (sede

		<p>dell'erario del distretto: LECCE);</p> <p>Distretto di Soleto (sede dell'erario del distretto: SOLETO);</p> <p>Distretto di Bari (sede dell'erario del distretto: BARI).</p> <p>Le successive 4 cifre riguardano l'anno in cui il distretto erariale è citato nella fonte.</p>
anno	<p>Indicazione dell'anno solare a cui fa riferimento la fonte.</p> <p>Nota: nel caso di anno indizionale si indica l'anno solare su cui si estende maggiormente l'anno indizionale</p>	<p>Numero intero di quattro cifre</p>
est_x	<p>Coordinata cartografica metrica Est UTM WGS84 (fuso 33) per la georeferenziazione degli insediamenti presi in esame</p>	<p>Valore numerico espresso in metri</p>
nord_y	<p>Coordinata cartografica metrica Nord UTM WGS84 (fuso 33) per la georeferenziazione degli</p>	<p>Valore numerico espresso in metri</p>

	insediamenti presi in esame	
castellania	Tipo di ufficio	Booleano (0/1)
centro_capitaneale		Booleano (0/1)
centro_giurisdizionale		Booleano (0/1)
tesoreria		Booleano (0/1)
centro_vicaria		Booleano (0/1)
dogana		Booleano (0/1)
fondaco		Booleano (0/1)
centro_affida		Booleano (0/1)

ENTITÀ: personaggio

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_personaggio	Codice identificativo di un personaggio per il quale si sono trovate informazioni in una certa fonte	Codice numerico
nome_funz_lat	Nome latino dell'ufficiale	Stringa alfabetica
nome_funz_ita	Traduzione italiana del nome latino dell'ufficiale	Stringa alfabetica

ENTITÀ: ufficiale

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_ufficiale	Codice identificativo di un ufficiale per il quale si sono trovate informazioni in una certa fonte	Codice numerico
id_personaggio	Codice identificativo di un personaggio per il quale si sono trovate informazioni in una certa fonte	Codice numerico
id_fonte	Codice identificativo della fonte esaminata	Codice alfanumerico
id_distretto_erariale	Codice indicante il Distretto Erariale di appartenenza del luogo	Codice numerico a 5 cifre: la prima cifra attualmente è compresa fra 1 e 4; i valori da 1 a 3 riguardano i tre Distretti di Terra d'Otranto, 4 è riferito a Bari: Da Taranto a Lecce (sede dell'erario del distretto: TARANTO); Da Lecce a Leuca (sede dell'erario del distretto: LECCE); Distretto di Soleto (sede dell'erario del distretto:

		SOLETO); Distretto di Bari (sede dell'erario del distretto: BARI). Le successive 4 cifre riguardano l'anno in cui il distretto erariale è citato nella fonte.
id_unita_fiscale	Codice identificativo del centro fiscale a cui sono pagate le imposte	Codice numerico
id_ufficio	Codice identificativo di un ufficio per il quale si sono trovate informazioni in una certa fonte	Codice numerico
anno	Indicazione dell'anno solare a cui fa riferimento la fonte. Nota: nel caso di anno indizionale si indica l'anno solare su cui si estende maggiormente l'anno indizionale	Numero intero di quattro cifre
nome_funz_lat	Nome latino dell'ufficiale	Stringa alfabetica
nome_funz_ita	Traduzione italiana del nome latino dell'ufficiale	Stringa alfabetica
nome_da_fonte	Nome dell'ufficiale come è stato rinvenuto nella fonte	Stringa alfabetica
soci	Assenza/presenza di soci,	Booleano (0 / 1)

	rilevabile dal suffisso “et socii” o “et sociis” in nome_da_fonte	
erario_generale	Ruolo dell’ufficiale	Booleano (0 / 1)
erario_locale		Booleano (0 / 1)
baiulo		Booleano (0 / 1)
capitano		Booleano (0 / 1)
vicario		Booleano (0 / 1)
tesoriere_rif		Booleano (0 / 1)
tesoriere		Booleano (0 / 1)
castellano		Booleano (0 / 1)
esattore_affida		Booleano (0 / 1)
doganiere		Booleano (0 / 1)
num_doganieri	Numero dei doganieri operanti nel centro nell’anno di riferimento della fonte	Valore numerico intero

ENTITÀ: imposte

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_imposte	Codice identificativo delle imposte versate da un luogo a un determinato centro fiscale in un determinato anno, secondo la fonte a cui si fa riferimento	Codice numerico
id_fonte	Codice identificativo della fonte esaminata	Codice alfanumerico
id_unita_fiscale	Codice identificativo del centro	Codice numerico

<p>focatico</p>	<p>fiscale a cui sono pagate le imposte</p> <p>Importo complessivo dell'imposta sui fuochi versata dal luogo al centro fiscale.</p> <p>Il focatico corrisponde all'imposta diretta introdotta da Alfonso d'Aragona con la riforma fiscale del 1443. Tale sistema era fondato sull'imposizione di 1 ducato d'oro, pari a 5 tari e 10 grana, per ogni fuoco</p>	<p>Valore numerico intero, espresso in grana</p>
<p>colletta</p>	<p>Importo complessivo dell'imposta definita "colletta" dal luogo al centro fiscale.</p> <p>La colletta era la forma di prelievo fiscale utilizzata nel periodo angioino; dapprima richiesta come imposta straordinaria, con Carlo I assunse un carattere fisso e annuale. Fu sostituita dal focatico, in seguito alla riforma fiscale attuata da Alfonso d'Aragona, ma all'interno del Principato di Taranto le comunità potevano essere</p>	<p>Valore numerico intero, espresso in grana</p>

	sottoposte all'una o all'altra forma di prelievo fiscale	
tomoli_sale	Quantità complessiva di tomoli di sale acquistata da ciascun luogo. Il tomolo era una misura di capacità per i solidi in uso a Napoli	Valore numerico indicante il numero di tomoli (ed eventualmente mezzi tomoli)
stoppelli_sale	Quantità complessiva di stoppelli di sale acquistata da ciascun luogo.	Valore numerico indicante il numero di stoppelli (ed eventualmente mezzi stoppelli). È presente raramente nel database
tassa_milizie	Entità complessiva della tassa per le milizie versata dal centro al principe	Valore numerico intero, espresso in grana
donativo	Entità complessiva della tassa "Pro dono consueto" versata dal centro al principe	Valore numerico intero, espresso in grana
taberna	Entità complessiva della tassa "Pro iure tabernae", ovvero il diritto di gestire taverne	Valore numerico intero, espresso in grana
bagliva	Entità complessiva dei proventi ricavati dalla Bagliva nell'anno di riferimento del documento in questione	Valore numerico intero, espresso in grana
platea	Entità complessiva dello "ius	Valore numerico intero,

	plateae” versato al principe per espresso in grana il diritto sul commercio	
affida_parziale	Entità complessiva della tassa “Pro Iure Affidae” versata dal centro in esame a quello avente tale diritto, in quanto fornitore di pascolo [...]	Valore numerico, espresso in grana
affida_totale	Entità complessiva della tassa “Pro Affida” incamerata dal centro avente tale diritto	Valore numerico, espresso in grana

ENTITÀ: fonte_erario

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_fonte	Codice identificativo della fonte esaminata	Codice alfanumerico
id_distretto_erariale	Codice indicante il Distretto Erariale di appartenenza del luogo	Codice numerico a 5 cifre: la prima cifra attualmente è compresa fra 1 e 4; i valori da 1 a 3 riguardano i tre Distretti di Terra d’Otranto, 4 è riferito a Bari: Da Taranto a Lecce (sede dell’erario del distretto: TARANTO); Da Lecce a Leuca (sede dell’erario del distretto:

	<p>LECCE);</p> <p>Distretto di Soleto (sede dell'erario del distretto: SOLETO);</p> <p>Distretto di Bari (sede dell'erario del distretto: BARI).</p> <p>Le successive 4 cifre riguardano l'anno in cui il distretto erariale è citato nella fonte.</p>
--	--

ENTITÀ: fonte_fisco

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_fonte	Codice identificativo della fonte esaminata	Codice alfanumerico
id_unita_fiscale	<p>Codice identificativo del centro di appartenenza fiscale</p> <p>Nota. Per quanto riguarda il centro fiscale, un luogo può fare riferimento: Al luogo stesso che è anche centro fiscale, a un altro luogo che è centro fiscale, a più insediamenti che sono centri fiscali (fra cui può esserci il luogo stesso)</p>	Codice numerico

ENTITÀ: appartenenza

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_unita_fiscale	<p>Codice identificativo del centro di appartenenza fiscale</p> <p>Nota. Per quanto riguarda il centro fiscale, un luogo può fare riferimento:</p> <p>Al luogo stesso che è anche centro fiscale,</p> <p>a un altro luogo che è centro fiscale,</p> <p>a più insediamenti che sono centri fiscali (fra cui può esserci il luogo stesso)</p>	Codice numerico
id_luogo	<p>Codice identificativo del luogo corrispondente al toponimo latino</p>	Codice alfanumerico

ENTITÀ: ufficiale_luogo

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_ufficiale	Codice identificativo di un ufficiale per il quale si sono trovate informazioni in una certa fonte	Codice numerico
id_luogo	Codice identificativo del luogo corrispondente al toponimo latino	Codice alfanumerico
id_fonte	Codice identificativo della fonte esaminata	Codice alfanumerico
anno	Indicazione dell'anno solare a cui fa riferimento la fonte. Nota: nel caso di anno indizionale si indica l'anno solare su cui si estende maggiormente l'anno indizionale	Numero intero di quattro cifre

ENTITÀ: imposte_luogo

<i>Attributo</i>	<i>Definizione</i>	<i>Valore, formato, unità di misura</i>
id_imposte	Codice identificativo delle imposte versate da un luogo a un determinato centro fiscale in un determinato anno, secondo la fonte a cui si fa riferimento	Codice numerico
id_luogo	Codice identificativo del luogo corrispondente al toponimo latino	Codice alfanumerico