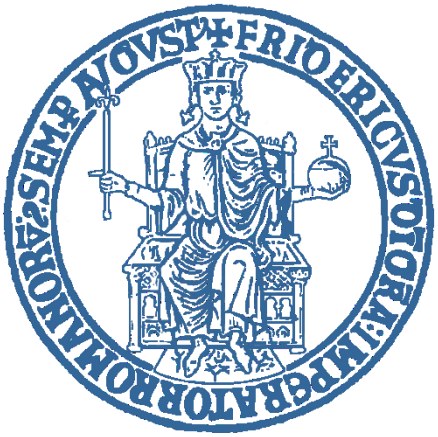
Università degli Studi di Napoli "Federico II"



Dottorato di Ricerca in Sanità Pubblica e Medicina Preventiva

Strumenti di business intelligence a supporto dell'analisi e gestione dei dati sanitari georeferenziati ed applicazione allo studio di patologie nella Terra dei Fuochi.

.

Relatore

Chiar.ma Prof. Maria Triassi

Dottorando

Dr. Mario Alessandro Russo

Correlatore

Chiar.mo Prof. Mario Cesarelli

Indice

GLOSSARIO

API (Application Programming Interface) = in informatica, 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.

Applications server = In informatica un application server (a volte abbreviato con la sigla AS) è una tipologia di server che fornisce l'infrastruttura e le funzionalità di supporto, sviluppo ed esecuzione di applicazioni nonché altri componenti server in un contesto distribuito. Si tratta di un complesso di servizi orientati alla realizzazione di applicazioni ad architettura multilivello ed enterprise, con alto grado di complessità, spesso orientate per il web (applicazioni web).

Bigdata = Big data ("grandi dati" in inglese) è un termine adoperato per descrivere l'insieme delle tecnologie e delle metodologie di analisi di dati massivi[1]. Il termine indica la capacità di estrapolare, analizzare e mettere in relazione un'enorme mole di dati eterogenei, strutturati e non strutturati, per scoprire i legami tra fenomeni diversi e prevedere quelli futuri.

Business Intelligence = In termini molto semplici e volutamente pratici si può descrivere un'applicazione di BI come uno strumento software che, acquisendo e manipolando masse di dati presenti su database o anche archivi de-strutturati, fornisce report, statistiche, indicatori, grafici costantemente aggiornati, facilmente adattabili e configurabili persino dall'utente (senza necessità di intervento del tool administrator)

Cartella Clinica Elettronica (CCE) = La Cartella Clinica Elettronica è una raccolta di dati clinici del paziente messa su supporto informatico integrato al servizio delle Aziende Ospedaliere, introdotta dall’emendamento sulle Semplificazioni approvato dalle commissioni Affari Costituzionali e Attivita’ produttive della Camera in vigore dal 12 febbrai 2012. Tale cartella sanitaria digitale, funziona al pari della cartella clinica cartacea ma molto più pratica

e sicura, in quanto tutti i dati medici e clinici, dagli esami di laboratorio effettuati alla storia clinica del paziente, dalle prescrizioni mediche alla spesa sanitaria.

Clustering = clustering o analisi dei gruppi (dal termine inglese cluster analysis introdotto da Robert Tryon nel 1939) è un insieme di tecniche di analisi multivariata dei dati volte alla selezione e raggruppamento di elementi omogenei in un insieme di dati.

CRAN = CRAN è l'acronimo di Comprehensive R Archive Network ovvero un sistema per documentare e rendere disponibili i moduli aggiuntivi al software statistico R. È una rete di server FTP e di server web che offrono la versione aggiornata di R, assieme alla documentazione ed ai moduli aggiuntivi.

Dashboard = Dashboard (in italiano cruscotto) è un'applicazione per il sistema operativo macOS sviluppata dalla Apple Inc., che, all'occorrenza, consente di attivare con un tasto delle mini-applicazioni, chiamate widget, e farle successivamente sparire dal desktop quando non servono più.

Dataframe = Un dataframe è una lista di vettori (le variabili), che devono avere tutti la stessa lunghezza (numero di casi), ma possono essere di tipo diverso: variabili nominali (fattori), variabili cardinali (vettori numerici), etc.

Data Mart = Un data mart è un raccoglitore di dati specializzato in un particolare soggetto che contiene un'immagine dei dati che permette di formulare strategie sulla base degli andamenti passati.

Data Mining = Il data mining è l'insieme di tecniche e metodologie che hanno per oggetto l'estrazione di una informazione o di una conoscenza a partire da grandi quantità di dati (attraverso metodi automatici o semi-automatici) e l'utilizzo scientifico, industriale o operativo di questa informazione.

Data Warehouse = In informatica un data warehouse (acronimo DW o DWH, traducibile come "magazzino di dati") è un archivio informatico contenente i dati di un'organizzazione, progettati per consentire di produrre facilmente analisi e relazioni utili a fini decisionali-aziendali.

Extraction, Transformation, Loading (ETL) = In informatica Extract, Transform, Load (ETL) è un'espressione in lingua inglese che si riferisce al processo di estrazione, trasformazione e caricamento dei dati in un sistema di sintesi (data warehouse, data mart...).

GUI = L'interfaccia grafica utente, nota anche come GUI (dall'inglese Graphical User Interface), comunemente abbreviata in interfaccia grafica, è un tipo di interfaccia utente che consente all'utente di interagire con la macchina controllando oggetti grafici convenzionali.

ICD-IX = La classificazione ICD (dall'inglese International Classification of Diseases; in particolare, International Statistical Classification of Diseases, Injuries and Causes of Death) è la classificazione internazionale delle malattie e dei problemi correlati, stilata dall'Organizzazione mondiale della sanità (OMS-WHO). L'ICD è uno standard di classificazione per gli studi statistici ed epidemiologici, nonché valido strumento di gestione di salute e igiene pubblica.

ICPC = L'ICPC è una classificazione progettata per le cure primarie di tutto il mondo e soddisfa la possibilità di accogliere gli episodi di cura di ogni paziente dai loro punti di partenza, che spesso consistono in sintomi o disturbi non specifici, inoltre consente la classificazione di diagnosi cliniche specifiche e degli interventi.

ICT = Le tecnologie dell'informazione e della comunicazione (in inglese Information and Communications Technology[1], in acronimo ICT), sono l'insieme dei metodi e delle tecnologie che realizzano i sistemi di trasmissione, ricezione ed elaborazione di informazioni

(tecnologie digitali comprese), ampiamente diffusisi a partire dalla cosiddetta Terza rivoluzione industriale.

IT = Information technology – tecnologie dell'informazione

Layout = In informatica, l'impaginazione e la struttura grafica di un sito web o di un documento.

Licenza GNU GPL = La GNU General Public License, comunemente indicata con l'acronimo GNU GPL o semplicemente GPL, è una licenza fortemente copyleft per software libero, originariamente stesa nel 1989 da Richard Stallman per patrocinare i programmi creati per il sistema operativo GNU. Infatti, a differenza di altre licenze libere non-copyleft, un'opera protetta da GNU GPL deve rimanere libera, ovvero col susseguirsi delle modifiche deve continuare a garantire ai suoi utenti le cosiddette quattro libertà.

Lista = Una lista in R è un oggetto che consiste di un insieme ordinate di altri oggetti, che sono chiamati componenti della lista.

Georeferenziazione = la georeferenziazione è la tecnica che permette di associare ad un dato, in formato digitale, delle coordinate che ne fissano la posizione sulla superficie terrestre.

MMG-PLS = Medici di Medicina Generale e i Pediatri di Libera scelta

OLAP = OLAP è l'acronimo dell'espressione On-Line Analytical Processing, che descrive un insieme di tecniche che consentono di una fotografia di informazioni (ad esempio quelle di un database relazionale) in un determinato momento e trasformare queste singole informazioni in dati multidimensionali.

OLTP = L'online transaction processing (OLTP) è un insieme di tecniche software utilizzate per la gestione di applicazioni orientate alle transazioni.

Pattern recognition = Il riconoscimento di pattern (in inglese, pattern recognition) è una sottoarea dell'apprendimento automatico. Esso consiste nell'analisi e identificazione di pattern all'interno di dati grezzi al fine di identificarne la classificazione. La maggior parte della ricerca nel campo riguarda metodi di apprendimento supervisionato e non supervisionato.

Plug-in = Il plugin in campo informatico è un programma non autonomo che interagisce con un altro programma per ampliarne o estenderne le funzionalità originarie.

Query = In informatica il termine query viene utilizzato per indicare l'interrogazione da parte di un utente di un database, strutturato tipicamente secondo il modello relazionale, per compiere determinate operazioni sui dati (selezione, inserimento, cancellazione dati, aggiornamento ecc.).

Shiny = Shiny consiste in un'applicazione web pensata appositamente per R, con la quale è possibile trasformare i comandi per l'analisi statistica in veri e propri software.

Software open source = In informatica, il termine inglese open source (che significa sorgente aperta) indica un software di cui gli autori (più precisamente, i detentori dei diritti) rendono pubblico il codice sorgente, favorendone il libero studio e permettendo a programmatori indipendenti di apportarvi modifiche ed estensioni.

Spatial Dataframe = dataframe contenenti le informazioni geografiche quali coordinate, sistema di riferimento e sistema di proiezione.

1 Introduzione

1.1 Cartella Clinica Elettronica e Fascicolo Sanitario Elettronico

E' ormai noto anche in letteratura che la gestione ed il controllo della salute si basano sempre di più sull'utilizzo, la trasmissione e il confronto di una grande quantità di dati, informazioni e conoscenze eterogenee [TANG, Paul C., et al. 2006]. Il bisogno di scambiare dati è aumentato vertiginosamente, sia all'interno della singola struttura sanitaria (tra i diversi soggetti e tra unità operative specializzate), sia tra strutture anche geograficamente distanti. La diffusione e lo sviluppo degli strumenti di Information and Communication Technology (ICT) è ormai maturo per soddisfare le crescenti necessità di memorizzazione, elaborazione e trasmissione dei dati clinici, in un contesto più ampio di informatizzazione del sistema sanitario. L'innalzamento dei costi e la complessità dell'organizzazione richiedono infatti un adeguato sistema informativo, che garantisca l'efficienza (attraverso l'ottimizzazione dell'organizzazione locale), l'efficacia (attraverso la pianificazione e il controllo) e l'adeguatezza delle prestazioni effettuate rispetto alle migliori pratiche.

La Cartella Clinica è lo strumento utilizzato per la gestione dei dati clinici di un assistito, dati che vengono raccolti durante gli incontri con gli operatori sanitari, per la prevenzione o in occasione di episodi di malattia. La Cartella clinica nella sua versione classica (ovvero cartacea) è divenuta sempre più ingestibile in quanto caratterizzata dalla presenza di documenti provenienti da moltissime fonti, e risulta pertanto sempre più difficile reperire ed organizzare in tempi rapidi le informazioni necessarie.

Per tali motivi è necessario ripensare sia i metodi impiegati finora per memorizzare e organizzare l'informazione clinica, sia le procedure per scambiare e mettere in comune i dati tra operatori sanitari. Questo processo di ristrutturazione ed innovazione è ormai inevitabile e porterà in un prossimo futuro a una cartella clinica completamente informatizzata (ovvero una Cartella Clinica Elettronica) perfettamente integrata nel sistema informativo sanitario. Per completare questo processo di transizione vengono richiesti:

- Un trattamento uniforme di dati clinici e amministrativi sui singoli e sulle strutture sanitarie, di letteratura scientifica;

- Protocolli, nell'ambito di sistemi informativi sempre più complessi ed estesi, con bisogni informativi e di comunicazione estremamente intensi e diversificati.

Tuttavia, nonostante gli sforzi in tale direzione esiste ad oggi una confusione tra Cartella Clinica Elettronica ed il Fascicolo Sanitario Elettronico così come proposto dal Servizio Sanitario Inglese, all'interno dei sistemi informativi sanitari o clinici.

Infatti la sostanziale differenza tra cartella e fascicolo elettronico può essere ricercata all'interno delle definizioni stesse, di seguito riportate:

- La *Cartella Clinica Elettronica Locale*: è limitata ad una singola struttura sanitaria e questa soluzione viene chiamata generalmente Electronic Patient Record (EPR) o anche Electronic Medical Record (EMR);
- La *Cartella Clinica Elettronica (CCE)* costituisce un'evoluzione della Cartella Clinica Cartacea (CCC) ovvero è lo strumento per la gestione organica e strutturata dei dati riferiti alla storia clinica di un paziente in regime di ricovero o ambulatoriale, garantendo il supporto dei processi clinici (diagnostico-terapeutici) e assistenziali nei singoli episodi di cura e favorendo la continuità di cura del paziente tra diversi episodi di cura afferenti alla stessa struttura ospedaliera mediante la condivisione e il recupero dei dati clinici in essi registrati. Le funzioni principali della CCE, riprendendo gli standard di Joint Commission International sono:
    - Supportare la pianificazione e la valutazione delle cure (predisposizione del piano diagnostico-terapeuticoassistenziale);
    - Costituire l'evidenza documentale dell'appropriatezza delle cure erogate rispetto agli standard;
    - Essere lo strumento di comunicazione volto a facilitare l'integrazione operativa tra i professionisti sanitari coinvolti in uno specifico piano diagnostico-terapeutico-assistenziale al fine di garantire continuità assistenziale;
    - Costituire una fonte dati per studi scientifici e ricerche cliniche, attività di formazione e aggiornamento degli operatori sanitari, valutazione delle attività

assistenziali ed esigenze amministrativo-legali nonché rispondere a esigenze di cost-accounting;

- Supportare la protezione legale degli interessi del paziente, dei medici e dell'azienda sanitaria: deve cioè consentire di tracciare tutte le attività svolte per permettere di risalire (rintracciabilità) ai responsabili, alla cronologia e alle modalità di esecuzione.

La CCE è pertanto un sistema informatico che contiene tutte le informazioni necessarie per la gestione di un processo diagnostico-terapeutico-assistenziale che di norma comprende informazioni di assessment clinico (anamnesi) e infermieristico (rilevazione dei fabbisogni infermieristici), esame obiettivo, diario clinico integrato (medico e infermieristico), referti di prestazioni ambulatoriali e di altri esami diagnostico-specialistici (ad es. laboratorio, anatomia patologica, radiologia...) gestione del ciclo del farmaco e delle attività di nursing, gestione del percorso chirurgico, gestione della lettera di dimissione con eventuali suggerimenti per il MMG-PLS e di continuità assistenziale, vari documenti amministrativi quali ad esempio i consensi informati. Si ritiene opportuno precisare, riprendendo le indicazioni del documento di Linee Guida della regione Lombardia, che "la CCE si configura quindi come un sistema informatico integrato aziendale, da intendersi come trasversale alle varie tipologie di regimi clinico-sanitari di accesso e ai vari processi di cura, in sostituzione della cartella clinica cartacea, che da un lato ne rispetti i requisiti e le funzioni, e dall'altro risolva alcune criticità ad essa legate, offrendo opportunità di aumentare il valore attraverso l'integrazione con altri strumenti informatici. È importante infatti riconoscere allo strumento elettronico una sua dignità che ne determina anche una forte differenza nel modo di assolvere alle sue funzioni rispetto allo strumento cartaceo. Lo strumento elettronico oggi è in grado di assolvere a tutti i compiti formalmente definiti per la cartella clinica cartacea ma è necessario e auspicabile che lo faccia in modo diverso, ovvero secondo la logica di una efficace ed efficiente gestione elettronica del dato;

- Il *Fascicolo Sanitario Elettronico Personale* [HÄYRINEN et al. 2008], consiste in forme più complete di servizio che prevedono una qualche modalità di integrazione e di accesso in rete, su dati provenienti da applicazioni cliniche eterogenee. Queste

varie forme vengono denominate genericamente in inglese Electronic Health Record (EHR) o Fascicolo Sanitario Elettronico (FSE).

Le differenze risultano ancora più evidenti analizzando le finalità di ciascuno degli strumenti presentati:

1. *Finalità delle Cartelle Cliniche Elettroniche*: l'obiettivo principale è quello di supportare i clinici nello svolgimento dell'attività ospedaliera con funzionalità quali:
    a. La movimentazione dei pazienti (ricoveri, dimissioni, trasferimenti);
    b. La gestione dell'evoluzione clinica all'interno dei singoli episodi di cura (equivalente della cartella clinica cartacea);
    c. La gestione degli ordini ai servizi diagnostici (Order Management);
    d. La gestione dei posti letto;
    e. Supporto attività infermieristica;
    f. Permettere la condivisione delle informazioni sanitarie a tutti gli attori che si occupano della cura dei pazienti nella singola organizzazione Interfacciare i sistemi direzionali ospedalieri per supportare gli usi secondari dell'informazione (gestione amministrativa, ricerca, analisi epidemiologiche, …).
2. *Finalità del FSE*: l'obiettivo principale del Fascicolo Sanitario elettronico possono essere così riassunte:

    a. Permettere tramite la rete la condivisione delle informazioni sanitarie a tutti gli attori del sistema sanitario che si occupano della cura dei cittadini ;

    b. Fornire una visione integrata e contestualizzata della storia e della documentazione sanitaria prodotta fino a quel momento per un determinato cittadino (struttura Patient Centric), con modalità pensate ad hoc per i processi di diagnosi e cura;

    c. Creare una "rete delle reti" che raccoglie e rende disponibili tutte le informazioni raccolte nelle "reti verticali" (Reti di Patologia..etc.) ed in quelle "orizzontali" più generali (medicina di assistenza primaria, prevenzione, etc.)

d. Rendere fruibili le informazioni anche al cittadino in modo diretto (es. referti online);

e. Abilitare la comunicazione con i sistemi direzionali di Data Warehouse per supportare gli usi secondari dell'informazione (politica sanitaria, gestione amministrativa, educazione, ricerca, analisi epidemiologiche, …).

1.1.1 La Cartella Clinica Elettronica per MMG: caratteristiche e vantaggi

La cartella clinica elettronica o informatizzata è divenuta strumento indispensabile di lavoro per il Medico di Medicina Generale (MMG) per migliorare le sue possibilità assistenziali. Le possibilità offerte dalla cartella clinica elettronica sono :

- Avere a disposizione i dati anamnestici e le terapie in corso;
- Fornire con una certa rapidità certificati, ricette ripetibili;
- Seguire nel tempo i problemi e partecipare a indagini epidemiologiche come previsto dagli accordi regionali.

La cartella clinica in medicina generale deve essere differente da quella utilizzabile in ambito ospedaliero perché diverso è il tempo e il campo di utilizzo. Così, mentre in ospedale può essere utile focalizzare l'attenzione sugli eventi prossimi che portano a un certo iter diagnostico verso una determinata patologia, in medicina generale deve essere utilizzata una cartella clinica per problemi, in cui la raccolta dei dati ruoti intorno al problema per cui è stata richiesta la visita, e sia possibile distinguere tra problemi attivi, per i quali deve ancora essere trovata una soluzione, e problemi inattivi, ovvero già risolti.

La cartella clinica in Italia non è considerata di proprietà della ASL, come in altri Paesi (per esempio la Gran Bretagna), ma del medico curante; questo ha ovviamente facoltà di trasmetterla a specialisti, sostituti, medici ospedalieri coinvolti nella cura del paziente. In genere non viene lasciata al paziente per evitare dimenticanze, smarrimenti, incongrue consultazioni mentre è buona norma lasciare al paziente gli originali degli esami e delle visite specialistiche affinché possano servire in caso di emergenza come fonte di dati per altri medici.

Una cartella medica orientata per problemi particolarmente utile è quella ideata da L.L. Weed nel 1969 [WEED et al. 1964] per la formazione degli studenti di medicina americani, e ispirata a principi di praticità (orientamento per problemi), comprensione, completezza e onestà (dovendo annotare le situazioni che hanno portato a una determinata decisione) e adattabilità alla computerizzazione, nonostante quest'ultima non sia considerata indispensabile.

La cartella clinica orientata problemi è così suddivisa :

- Dati di base
- Lista dei problemi, attivi e inattivi
- Diario clinico, compilato tenendo conto di dati generali oggettivi (polso, pressione arteriosa, obiettività particolari), dati soggettivi (sintomi), ipotesi diagnostiche con relative strategie diagnostiche e terapeutiche
- Allegati eventuali per il monitoraggio di disturbi cronici, per attività preventive, per riportare esami strumentali e di laboratorio, visite specialistiche o altro

Nelle informazioni di base devono essere annotati oltre ai dati del paziente, la sua attività lavorativa, eventuali allergie, vaccinazioni, sieropositività, terapie, abitudini di vita, occupazione lavorativa, eventuali problemi familiari o sociali, oltre a un'anamnesi ostetrico-ginecologica per la donna.

La lista dei problemi riassume la situazione clinica del paziente mentre il diario clinico è un'ipotesi di lavoro in cui vengono annotati i sintomi riferiti dal paziente, l'obiettività, le ipotesi diagnostiche e le procedure diagnostiche o terapeutiche; qualora venga eseguito un esame strumentale questo può venire annotato nel diario clinico sotto la data così da essere prontamente evidenziato con il risultato conseguito annotato come obiettività. Gli allegati sono utili per monitorare determinate patologie croniche e ricordarsi dei controlli da effettuare per quella data patologia.

Tale tipo di cartella presuppone notevoli capacità sintetiche e talora viene considerata poco flessibile. E' importante comunque utilizzare un linguaggio comprensibile perché se si creano diversità di interpretazione e di linguaggio fra diversi medici può essere difficile la comunicazione tramite cartella.

Si ricorda inoltre come sia fondamentale rivedere periodicamente le cartelle (per esempio annualmente) e ripulirle di tutti quei dati che possono essere divenuti superflui e costituire così inutili digressioni nella lettura della cartella.

Vantaggi della cartella clinica elettronica

Nella Tabella seguente sono esposti in maniera sintetica i pro e i contro della Cartella Clinica Elettronica secondo quanto riportato in un report del Sole-24 ore Sanità.

PROBLEMA	CARTELLA CLINICA CARTACEA	CARTELLA CLINICA ELETTRONICA
Fisicità del cartella	Originale e in copia esistono solo dove sono conservate.	Dipende dalla presenza di un hardware e di un software adeguati, quindi dipende dai punti dove è memorizzata, dai relativi back-up. Può essere resa accessibile da punti remoti.
Accessibilità	E' fisicamente presente nel punto in cui viene usata.	E' possibile prevedere un punto di accesso da qualsiasi punto della rete.
Risorse	E' più economica.	Richiede investimenti nell'hardware, nel software, nella manutenzione, negli aggiornamenti e nell'addestramento. I maggiori costi sarebbero compensati dalle maggiori prestazioni e da una migliore qualità della assistenza.
Adattamento	Esistono molte varietà: la cartella cartacea si adatta all'esigenze dell'utilizzatore.	Le cartelle elettroniche possono essere realizzate in diversi modi dai progettisti che ne definiscono l'interfaccia, l'architettura e la funzionalità con forti implicazioni sul supporto e sul trasferimento di informazioni cliniche da un sistema all'altro. Inoltre e' l'utente (medico o infermiere) che si deve adattare al sistema scelto da usare.
Manutenzione	Non richiede una particolare manutenzione, a parte la custodia e l'archiviazione coi suoi problemi per la tutela alla riservatezza.	Ha precise necessità su manutenzione tecnica, aggiornamenti, preservazione dell'integrità dei dati, che richiedono diverse abilità organizzative e specifiche risorse.
Addestramento	L'uso della cartella clinica è istitutivo e tramandato didatticamente con schemi tradizionali.	Non usabile senza una formazione di base nell'uso del computer e addestramento specifico sull'applicazione utilizzata.

Identificazione del paziente	Conoscendo il paziente è difficile non abbinare correttamente il paziente con la sua cartella. Tuttavia un paziente o alcuni suoi esami possono essere confusi con quelli di un altro che presenti dati anagrafici simili oppure la saltuarietà dei contatti, specialmente per le schede sanitarie o per le schede cliniche, può portare a produrre due cartelle diverse.	Il sistema elettronico rende più difficile gli scambi degli esami e delle cartelle.
Accessi indesiderati	Può essere consultata solo nel luogo fisico dove si trova. Non evidenzia gli accessi di semplice consultazione.	Se in rete può essere interrogata e visualizzata anche a distanza se non sono attuate e rispettate misure adeguate di sicurezza. Va tenuto presente che un sistema elettronico mantiene però sempre traccia di tutti gli accessi effettuati, per cui sono facilmente evidenziati eventuali intromissioni e manomissioni. Per la diffusione di computer portatili e palmari ci sono rischi di furti e quindi di eventuali entrate illecite nei sistemi di gestione della cartelle cliniche.
Integrità	Per una manomissione o distruzione totale o parziale deve essere presente la cartella clinica.	Un accesso illegale al sistema può distruggere o alterare i dati da una postazione remota. Permette copie più sicure, in particolare evitando alterazioni o distruzioni accidentali.
Attribuzione	La firma o la calligrafia permettono l'identificazione dell'autore.	La firma elettronica garantisce l'inalterabilità del contenuto di un documento e l'identità dell'autore.
Immissione d dati clinici	E' più semplice anche se ci possono essere diversi moduli con campo e spazi predisposti.	Anche qui i dati possono essere strutturati o espressi come narrativa libera. Inoltre i documenti e le immagini ricevuti in forma elettronica possono essere collegati a tutta la cartella o a un elemento di essa. Può essere effettuata più rapidamente la ricerca della voce disponibile più appropriata.
Consultazione elaborazione dei dati	Il recupero di dati è più indaginoso.	In recupero dei dati per consultare, confrontare, analizzare o valutare) è più agevole anche per grosse moli di dati. Può dare diverse viste sui dati, a seconda del livello di accesso consentito e del compito corrente. Può dare segnalazione di errori, aiutare nelle

		decisioni, ricordare scadenze.
Interpretazione dati	Il significato dei contenuti riflette direttamente le intenzioni e la capacità di esprimersi dell'autore con considerevole libertà di espressione. Può creare ambiguità legate a abbreviazioni o appunti stenografici.	Pone limiti, più o meno rigidi, sui dati che è possibile rappresentare. Può però facilitare l'uso appropriato di testo e codici, orientare per problemi o addirittura gestioni avanzate dei documenti ricevuti.
Accuratezza d dati	E' demandata alla raccolta e il controllo avviene per semplice lettura e verifica mnemonica.	I sistemi elettronici permettono di effettuare controlli al momento dell'immissione dei dati e di verificarne tempestivamente la loro incompletezza o alcune incongruenze.

Tabella 1. Elenco dei vantaggi e degli svantaggi della CCE in funzioine del problema.

Per concludere, oltre ai già citati vantaggi va ricordato che il sistema di archiviazione e dematerializzazione delle cartelle cliniche apporta dei grandi vantaggi ai servizi ospedalieri in termini di:

- Risparmio: ambientale e di costi con l'abbattimento dell'utilizzo della carta;
- Archiviazione e giurisdizione: la prassi tradizionale di gestione documentale ha bisogno di essere rimodernata e questo avviene all'insegna dei nuovi media digitali;
- Soluzioni tecniche: garantiscono una maggiore efficienza ed efficacia nello svolgimento dei processi accostando questi ultimi ad una garanzia autenticità, integrità, accessibilità e sicurezza.

Queste sono caratteristiche comuni al processo di dematerializzazione e archiviazione digitale fortemente incoraggiate anche dal ministero della Salute attraverso un documento riportante le "*Linee guida per la dematerializzazione della documentazione clinica in diagnostica per immagini*". Tale documento è stato redatto da un gruppo di lavoro che ha visto la partecipazione, oltre che di esperti del Ministero e di altre Amministrazioni centrali,

anche di rappresentanti di associazioni e società scientifiche di settore. Il documento predisposto ha l'obiettivo di fornire ai Direttori Generali, ai Direttori Sanitari, ai Direttori/Responsabili dei Sistemi Informativi e dei Dipartimenti e Unità Operative (U.O.) di Diagnostica per Immagini, Radiologia, Medicina Nucleare, le linee guida per poter gestire la documentazione clinica testuale e iconografica, ottenuta direttamente in formato digitale, nel rispetto delle attuali normative, nonché prescrivere direttive pratiche e di fattibilità per realizzare la completa dematerializzazione dei referti e delle immagini di diagnostica.

1.2 Business Intelligence

Il termine Business Intelligence (BI) [CHEN et al. 2012] è stato coniato nel 1989 da Howard Dresner, analista di Gartner Group [BUCHANAN et al. 2006], per indicare una classe di applicazioni e strumenti informatici in grado di venire incontro all'esigenza sempre crescente da parte dei manager, di definire e gestire in maniera più incisiva i flussi informativi di un'azienda o di un'organizzazione. Da allora il termine viene utilizzato, talvolta con un abuso, per indicare l'insieme di tutti gli strumenti e sistemi che vengono impiegati per generare la reportistica direzionale e per il supporto alle decisioni.

Le organizzazioni raccolgono informazioni a differenti livelli di dettaglio per poter trarre valutazioni e stime sia riguardo al proprio contesto aziendale che rispetto al mercato di riferimento. Le informazioni raccolte sono poi gestite attraverso un sistema di BI per incrementare il vantaggio competitivo e/o per migliorare aspetti organizzativi e strategici. Nei casi più comuni le informazioni vengono raccolte per scopi direzionali interni e per facilitare il controllo di gestione, ovvero i dati raccolti vengono elaborati per poi essere utilizzati per supportare, sulla base delle informazioni di contesto o di eventuali previsioni, le decisioni di chi occupa ruoli direzionali (ad esempio per capire l'andamento delle performance dell'azienda, generare stime previsionali, ipotizzare scenari futuri e future strategie di risposta) [TURBAN et al. 2011]. Ovviamente le informazioni possono essere analizzate anche a differenti livelli di dettaglio per essere impiegate per migliorare qualsiasi altra funzione dell'azienda o dell'organizzazione (marketing, commerciale, finanza, personale ecc.). Le fonti informative sono generalmente interne, provenienti dai sistemi informativi aziendali ed integrate tra loro a seconda delle esigenze. In senso più ampio possono anche essere utilizzate informazioni provenienti da fonti esterne all'organizzazione come ad

esempio esigenze della base dei clienti, pressione stimata degli azionisti, trend tecnologici o culturali fino al limite delle attività di spionaggio industriale [ERVURAL et al. 2017].

Pertanto, ogni sistema di BI deve avere sempre un obiettivo preciso che deriva dalla vision e dagli obiettivi della gestione strategica dell'azienda.

In un'accezione più tecnologica il termine BI consiste in un insieme di tecniche informatiche e di processi di business che consentono di raccogliere ed analizzare informazioni digitali e le tecnologie necessarie per la realizzazione dei suddetti processi [CHAUDHURI et al. 2011]. Gli strumenti software utilizzati hanno l'obiettivo di permettere alle persone di prendere decisioni strategiche fornendo informazioni precise, aggiornate e significative nel contesto di riferimento. Ci si può riferire ai sistemi di BI anche con il termine "sistemi per il supporto alle decisioni", anche se l'evoluzione delle tecniche utilizzate rende la terminologia suscettibile di ammodernamenti.

In termini prettamente informatici con BI si identificano i sistemi di analisi di dati in formato digitale con lo scopo di perseguire i seguenti obiettivi:

1. Analisi e aggregazione dei dati utilizzando un data warehouse;
2. Individuazione di pattern di regolarità all'interno dei dati che consentano di identificarne la classificazione attraverso strumenti di Data Mining;
3. La rappresentazione delle informazioni ottenute come output dall'esecuzione dei suddetti processi informatici in grado di trasformare i dati e le informazioni (prelevati dal campo) in conoscenza (ad un maggiore livello di astrazione).

Di fatto, il Sistema Informativo Direzionale è costituito da un insieme di processi, tecnologie ed applicazioni che trasformano i dati in informazione, l'informazione in conoscenza e la conoscenza in piani aziendali cha appunto vengono definiti BI.

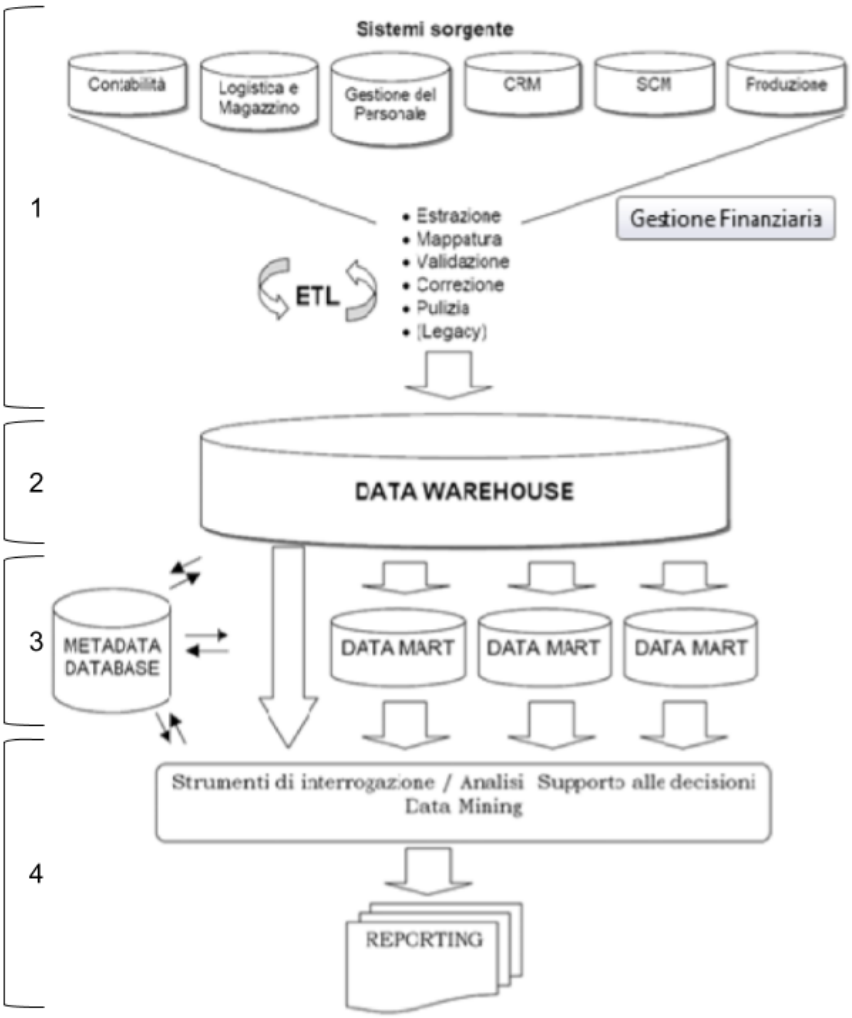


Figura 1 - Processo di Business Intelligence

La Figura 1 mostra in maniera sintetica un tipico esempio dove sono impiegati sistemi e processi caratteristici della BI:

1. I dati in diversi formati e provenienti da sistemi eterogenei vengono elaborati attraverso strumenti di tipo Extraction, Transformation, Loading (ETL) per poter essere caricati in un Data Warehouse (DW);
2. Il DW è costituito da una collezione di dati di supporto al processo decisionale, che risulti consistente, integrato, orientato al soggetto e rappresentativo dell'evoluzione temporale dei dati, di cui fornisce una visione unificata e corretta [INMON et al. 2005.]. Un sistema di DW, oltre ad essere una struttura di memorizzazione dati più evoluta del semplice database, è un processo complesso che parte dall'estrazione dei

dati operativi per arrivare alla trasformazione degli stessi sino alla presentazione delle informazioni. Questo processo prende nome di Data Warehousing. Il DW, quindi, non è un prodotto da acquistare ed installare in azienda, ma un vero e proprio Sistema Informativo Direzionale che ricorre a tecnologia software ed hardware. Nel momento iniziale di implementazione di un Sistema Informativo Direzionale è essenziale una Business Analysis, cui segue la fase di sviluppo del "magazzino" fisico e degli applicativi; tali fasi vengono realizzate ad inizio progetto e reiterate a seconda delle modifiche intervenute nel business, nelle esigenze conoscitive o nelle tecnologie disponibili.

3. Il Data Warehouse pertanto è composto da sottoinsiemi dei dati, i data mart, rappresentati secondo uno schema multidimensionale (data cube) costituito da:
    - Fatti: il concetto oggetto dell'analisi;
    - Misure: una proprietà atomica di un fatto;
    - Dimensioni: una prospettiva lungo la quale si analizza il fatto;
    - Gerarchie: Aggregazione delle istanze dei fatti.
4. Gli strumenti di interrogazione, analisi, supporto alle decisioni e alle operazioni di Data Mining ricevono in ingresso il contenuto informativo dei Data Mart. In tale fase del processo di creazione della BI è possibile estrarre "conoscenza" dai dati. Attraverso appositi strumenti software è possibile effettuare in concreto l'analisi dei dati. Si tratta di strumenti che, partendo dalle interrogazioni multidimensionali previste, consentono di presentare di volta in volta i dati e le informazioni ottenute producendo l'output finale attraverso report testuali o grafici (dashboard).

Il processo di Data Warehousing deve essere supportato da strumenti e tecnologie che interrogano le basi di dati direzionali aziendali, sia di tipo relazionale che di tipo multidimensionale, e consentono l'elaborazione dei dati secondo schemi/modelli noti come tecnologie OLAP. OLAP è l'acronimo dell'espressione On-Line Analytical Processing, che descrive un insieme di tecniche che consentono di una fotografia di informazioni (ad esempio quelle di un database relazionale) in un determinato momento e trasformare queste singole informazioni in dati multidimensionali. Eseguendo successivamente delle interrogazioni sui dati è possibile ottenere risposte in tempi decisamente ridotti dal momento che il database di un sistema Online Transaction Processing (OLTP) non è adatto a consentire analisi articolate. I database OLAP tipicamente includono due tipi di dati:

1. Le misure: ovvero i dati numerici le quantità e le medie utilizzate per prendere decisioni aziendali consapevoli;
2. Le dimensioni: ovvero le categorie utilizzate per organizzare le misure.

I database OLAP consentono di organizzare i dati con diversi livello di dettaglio, utilizzando le stesse categorie impiegate per l'analisi dei dati. Nelle interrogazioni OLAP di un data warehouse o di un data mart le dimensioni di un indicatore sono assi di una matrice multidimensionale, detta ipercubo. Ogni lato dell'ipercubo rappresenta una dimensione ed ogni sottocubo contiene dati aggregati di un certo indicatore o delle dimensioni considerate. A titolo esemplificativo si veda la rappresentazione grafica proposta nella Figura 2.

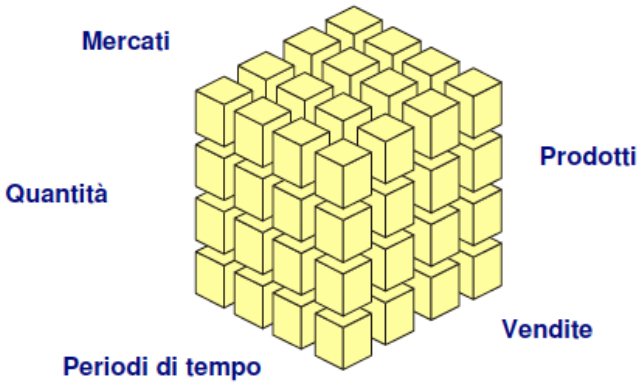


Figura 2 - Ipercubo OLAP

Per tale esempio sono state scelte le dimensioni tempo, quantità, mercati, prodotti, vendite per l'indicatore fatturato. Ogni dimensione raccoglie al suo interno altri elementi (ad esempio la dimensione tempo dell'indicatore fatturato avrà al suo interno l'elemento giorno, settimana, mese, trimestre, anno etc). La possibilità di suddividere ogni dimensione in diversi livelli di dettaglio aggregati è il requisito principale di un sistema OLAP.

Si noti che il processo di Data Warehousing è caratterizzato dalla presenza di soggetti aventi ruoli differenti. Al di là delle figure professionali (come ad esempio il database Administrator, data scientist, il programmatore delle applicazioni ecc.), stabilmente impiegate in azienda, dotate di competenze e conoscenze tecniche cioè utilizzo di linguaggi informatici e di software di gestione delle basi dati, vi sono soggetti che presidiano il sistema di controllo gestionale come ad esempio i controller. Il controller è una figura di mediazione tra i

manager/decision-maker, che sono gli utenti finali, e i tecnici dei Sistemi Informativi. Le tecnologie OLAP vengono utilizzate in azienda per far si che i controller ed i decision maker possano reperire le informazioni di cui hanno bisogno in modo autonomo ed interattivo, pur non essendo esperti di IT. Il manager pensa e "domanda direttamente" alla macchina che fornisce le risposte grazie alle tecnologie di interrogazione dei database direzionali. Il decision maker interroga il database, cioè crea della query, senza conoscere linguaggi di programmazione particolari come l'SQL, ma semplicemente disponendo di una interfaccia utente (GUI), fatta di icone, menù a tendina e semplici operazioni di navigazione. E' sufficiente che l'utente definisca le dimensioni di interesse.

In conclusione, l'utilizzo di strumenti di BI basati su data warehouse risulta cruciale per migliorare l'efficienza dei sistemi informativi, velocizzare ed ottimizzare il processo decisionale, consentendo inoltre una riduzione significativa dei costi. Inoltre, grazie a strumenti statistici e di Data Mining possono essere estratte informazioni implicite, precedentemente sconosciute e potenzialmente utili dai dati, rendendo possibili operazioni di clustering e pattern recognition, ed operazioni di forecasting, utili a determinare, con buona approssimazione, i valori futuri di determinate variabili estratte dai dati. Un esempio di impiego potrebbe essere rappresentato dal monitoraggio delle performance attraverso cruscotti interattivi, che consentano di tradurre i dati in informazioni evidenziando le criticità ed accelerando il processo di *decision making*.

1.3 Business Intelligence in Sanità

Secondo l'Osservatorio ICT in Sanità della School of Management del Politecnico di Milano "*L'innovazione digitale della Sanità italiana oggi è una soluzione obbligata, l'unica in grado di modernizzare il sistema e permettergli di reggere l'impatto della crescita della domanda, fermando il decadimento in atto di qualità ed efficienza*". Questa esigenza di innovazione tecnologica è fortemente spinta anche dalle linee guida del Ministero della Salute che stabilisce che "*l'adozione di soluzioni basate sulle tecnologie dell'informazione e delle comunicazioni (ICTs) diventa un'operazione strumentale, finalizzata al miglioramento dell'appropriatezza, dell'efficienza e dell'efficacia, attraverso l'efficientamento complessivo del SSN*".

Negli ultimi anni la BI è stata recepita in ambito sanitario come uno strumento fondamentale per la gestione e l'organizzazione di personale medico e pazienti e per la razionalizzazione dei fondi pubblici e delle spese che devono assolutamente rientrare nel bilancio dell'anno. Si tratta, in sostanza, di software e strumenti digitali che garantiscono una gestione ottimale di un ospedale o di un piccolo studio, perché permettono di mettere insieme ed analizzare dati economici, clinici, gestionali e di marketing offrendo la possibilità di valutare degli opportuni indicatori. Se aumentano gli indicatori relativi alle prestazioni delle strutture sanitarie, allora si richiede a queste ultime un salto di qualità o un miglioramento in determinati settori. L'identificazione di aree interne dove migliorare le performance va accompagnata da una gestione finanziaria oculata e da una altrettanto meditata riforma nella gestione dei pazienti durante il loro percorso di cura sia all'interno che all'esterno delle strutture cliniche.

Nel settore healthcare il potere dell' "analytics" deve ancora dispiegare pienamente il suo impatto. Se è vero che il settore Salute si caratterizza per la dinamica medico-paziente, è anche vero che la tecnologia ne costituisce l'intelaiatura. Diverse informazioni possono essere estratte dall'analisi dei dati in ambito sanitario: informazioni su come bilanciare i costi, migliorare i trattamenti, aderire agli standard di settore e, inoltre, definire i margini per un'ulteriore crescita. Uno sforzo computazionale di elevato impatto richiede una strategia altrettanto raffinata ed efficace, che abbracci le capacità tecnologiche di ultima generazione, delle infrastrutture adeguate e garantisca una forte governance dei dati; questo potrebbe rappresentare un ostacolo all'introduzione e diffusione della BI in ambito sanitario.

L'impiego della di BI in ambito sanitario viene spesso associato all'insieme degli strumenti appartenenti alla categoria degli analytics, del data warehousing, degli strumenti di visualizzazione. Tuttavia la BI andrebbe associata, più che a specifici strumenti, ad una strategia che risponda puntualmente al bisogno fondamentale di una struttura per i dati clinici [METTLER et al. 2009]. Già la Gartner ha evidenziato come la mancanza di una strategia di BI razionalizzata rappresenti uno dei nove ostacoli all'evoluzione dell'healthcare [Gartner et al. 2014]. Di certo, la consapevolezza dei limiti rappresenta il punto cruciale, dal quale partire per apportare miglioramenti, significativi e mirati, nella qualità di un processo, sempre più orientato al valore aggiunto verso il paziente [Gartner et al. 2015].

Secondo quanto riportato dalla Gartner diversi possono essere gli elementi invalidanti nello sviluppo di tale strategia:

- **Estrema delicatezza dei dati**. Le organizzazioni sanitarie si trovano a dover trattare informazioni sensibili e confidenziali, spesso sottoposte a regolamenti atti a tutelarne la riservatezza. Gli operatori sanitari sono quindi chiamati a gestire un sistema complesso che tenga conto, nei vari passaggi, del riserbo di tali informazioni;
- **Difficoltà nell'accesso ai dati**. Molte strutture sanitarie possono arrivare a utilizzare vari sistemi di cartelle cliniche elettroniche Electronic Health Record (EHR) simultaneamente, gestite sia in-house che da provider esterni. Farle interagire implica la necessità di conoscere le specifiche di ciascun database e di scrivere un codice aggiuntivo che permetta agli archivi che raccolgono i dati di comunicare tra loro. In tale contesto, gli operatori sanitari si troveranno, molto probabilmente, a dover affidare l'attività di integrazione di dati a una società esterna di BI. Ci si rende conto di quanto complesso sia il processo da gestire, senza considerare la ritrosia di molti fornitori di cartelle cliniche elettroniche a condividere i loro codici sorgenti;
- **La qualità dei dati.** I dati possono avere origini molto diverse (ERP, ADT, EHR), essere indirizzati verso dipartimenti differenti (es. radiologia, cardiologia, farmacia) e presentarsi sotto diverse forme (video, testo, numeri, immagini), provenienti da smartwatch, dispositivi wearable e altre applicazioni mHealth. Considerando l'eterogeneità dei dati disponibili, e molto spesso relativi allo stesso paziente, e il fatto che l'implementazione di un sistema basato su cartelle elettroniche sia un processo costoso e a lungo termine, che induce molto di frequente le strutture sanitarie a dotarsi, in principio, di sistemi non esattamente di ultima generazione, è evidente quanto la raccolta dati e il loro processing da parte del team di BI possano risultare pressoché epici;
- **L'inconsistenza dei dati**. Gli specialisti della salute, ovvero medici, chirurghi, personale paramedico, immettono dati in modi differenti. Sistemi come le cartelle cliniche elettroniche, e altri simili, incamerano queste informazioni in molteplici caselle. Il problema sorge quando lo specialista ha bisogno del dato. La flessibilità nell'immissione dell'informazione contribuisce a palesare l'anello debole nella gestione dei dati, che consiste nella presenza di informazioni ridondanti, e spesso discordanti, appartenenti al profilo dello stesso paziente.

La capacità di elaborare i dati sanitari in tempo reale può portare molteplici benefici ed è crescente la consapevolezza che la BI e la gestione integrata dei dati possano rappresentare la chiave di volta per far compiere al mondo della salute un notevole progresso, consentendo migliorie su diverse linee direttrici:

- Medicina personalizzata, in cui ogni paziente viene curato non in base a protocolli ma in relazione ai propri dati biomedici specifici;
- Telemedicina, grazie alle opportunità di integrazione offerte dall'Internet of Things e dai dispositivi indossabili, per individuare la cura giusta al momento giusto in ottica di deospedalizzazione;
- Ricerca medica, in quanto la velocità di elaborazione dei dati, resi anonimi, è in grado di ridurre esponenzialmente i cicli di sviluppo delle cure;
- Una migliore gestione finanziaria: Spesso il problema più impegnativo che si trovano ad affrontare le strutture sanitarie è proprio quello della gestione dei costi. Fortunatamente software specifici permettono ora di tenere sotto controllo il budget, monitorare le spese e gestire il bilancio, utilizzando grafici che permettono subito di capire quali sono i problemi della gestione finanziaria che vanno risolti immediatamente per assicurare al paziente una buona gestione della struttura;
- Una migliore gestione delle strutture localizzate sul territorio. La BI permette infatti di costruire un sistema sanitaria più efficiente e sicuro, monitorando da un lato i servizi erogati dalle singole strutture e dall'altro la storia clinica del paziente, dando così la possibilità di scovare eventuali frodi;
- Invio rapido di flussi sanitari alle Regioni: La nuova regolamentazione prevede infatti che strutture sanitarie, studi medici e specialisti inviino direttamente al Sistema Sanitario una documentazione relativa alle proprie spese. Software specifici permettono così di trasmettere una mole di dati molto ampia, proveniente persino da piattaforme diverse tra loro, ma facilmente trasmissibile con un semplice click.

Un caso concreto che dimostra i vantaggi di impiegare sistemi di BI è stato dimostrato già nel 2007 dall'Azienda Ospedaliero Universitaria Meyer di Firenze, specializzata nella cura dei bambini dalla nascita all'adolescenza. L'Ospedale nell'anno 2012 ha realizzato 3.980

ricoveri ordinari chirurgici e 2.300 casi di Day Surgery e rappresenta un centro di riferimento nazionale per numerose specialità sia mediche che chirurgiche. L'Azienda, a partire dal 2007 ha sviluppato, in collaborazione con Oslo, un sistema di Business Intelligence per il reporting direzionale, grazie alla creazione di un Data Warehouse con l'obiettivo di:

1. Utilizzare i dati sanitari e amministrativi per la governance aziendale, per monitorare i costi e migliorare l'efficienza;
2. Realizzare una reportistica completa e aggiornata.

Il progetto integra al suo interno:

- La contabilità analitica;
- I dati sanitari inerenti i ricoveri e la specialistica ambulatoriale;
- Sistema di analisi dei costi e delle attività;
- Gli applicativi aziendali (CUP, RIS/PACS, LIS, ecc.);
- Una reportistica completa e aggiornata, a partire dai dati presenti in Azienda.

Tra i benefici conseguiti a seguito vi è la riduzione dei consumi interni di prodotti, con un calo nel 2012 del 5,10% (pari a oltre 700.000 euro) del consumo di farmaci rispetto al 2011. Inoltre, l'implementazione del sistema ha comportato una riduzione delle giacenze pari a 820.000 euro nel 2011 e 470.000 euro nel 2012 un valore complessivo di 1,3 milioni di euro.

1.4 Business Intelligence - strumenti software

Relativamente a software e sistemi, ovvero alla tecnologia, la BI è rappresentata da una vasta serie di prodotti che presentano caratteristiche affini, spesso modulari tanto da rendere complesso la scelta e la gestione dei software da utilizzare. Effettuare una scelta corretta degli strumenti da analizzare non è semplice e richiede la conoscenza di una grande quantità di realtà per comprendere quali prodotti possono soddisfare al meglio i requisiti specifici richiesti dall'attività che si vuole realizzare.

I prodotti presenti sul mercato possono essere classificati in tre categorie:

- **Prodotti storici:** Oracle Corporation, Microsoft Corporation, IBM, SAP e MicroStrategy;

- **Prodotti indipendenti:** Qlik, Spotfire e Tableau;
- **Prodotti open source:** Pentaho, TIBCO Jaspersoft BI e Eclipse BIRT Project.

Date le finalità di ricerca la nostra analisi si è concentrata sulla categoria di software open source. Tra le suite considerate alcune offrono sia la versione Community, quindi gratuita, sia delle versioni più complete aventi componenti e funzionalità aggiuntive, disponibili attraverso abbonamenti e licenze non più libere, ma commerciali. I prodotti valutati sono quelli attualmente considerati tra i migliori per la loro tipologia e che presentano un peso importante anche all'interno del mercato dei prodotti di BI.

Pentaho

Pentaho è una società relativamente giovane fondata nel 2004 da cinque soci con sede in Orlando, Stati Uniti d'America. Questa società offre una suite di BI Open Source in grado di coprire tutto l'ampio ventaglio di potenzialità della BI. All'interno del prodotto, si possono trovare funzionalità di reporting anche in modalità self-service, cubi OLAP per l'analisi multidimensionale del dato, cruscotti e dashboard per visualizzare in modo semplice e intuitivo i principali indicatori in grado di valutare l'andamento dell'azienda, integrazione dei dati attraverso funzionalità ETL per facilitare l'integrazione dei dati provenienti da sorgenti differenti, data mining [https://www.pentaho.com].

La suite di Pentaho offre due tipologie di prodotti: la **Community Edition (CE)** e l'**Enterprise Edition (EE)**. La CE è la soluzione open source offerta da questa suite e, nonostante sia la versione gratuita, contiene al suo interno una serie di prodotti che offrono una valida alternativa ad altre soluzioni di BI. La EE fornisce delle componenti e dei programmi aggiuntivi che rendono il prodotto più potente e più competitivo anche per le imprese medio-grandi. La versione enterprise si ottiene attraverso un abbonamento annuale che comprende anche servizi di assistenza aggiuntivi. L'ammontare della licenza varia a seconda dei servizi richiesti dalle singole aziende, di conseguenza è difficile eseguire una stima.

In seguito verranno analizzati i principali prodotti che compongono le due tipologie di suite proposte da Pentaho. Si possono suddividere questi prodotti in due ulteriori tipologie:

gli applications server e le desktop application. I principali prodotti application server offerti da entrambe le suite sono:

- **Pentaho BI Platform**: viene più generalmente chiamato BI Platform e, recentemente, è stato anche rinominato in Business Analytics Platform (BA Platform). Include una serie di funzionalità che consistono nella gestione della sicurezza, esecuzione di report, visualizzazione di dashboard, script per la definizione di regole di business e analisi OLAP. Questa applicazione viene eseguita in Apache Java Application Server.
- **Pentaho Analysis Services (Mondrian)**: è un application server scritto in java in grado di eseguire analisi OLAP (Online Analytical Processing). Supporta il linguaggio XML e MDX (Multidimensional Expressions) per le query ed è capace di leggere da sorgenti SQL e da altre sorgenti, aggregando i dati in una memoria cache.
- **Pentaho Data Access Wizard**: il plug-in è fornito assieme a tutti i server e permette all'utente di creare nuove sorgenti dati da utilizzare per tutto il sistema a partire da altri database o da file CSV presenti nel server. All'utente viene fornita la possibilità di creare un modello dei dati che descrive come i campi delle tabelle sono in relazione tra di loro. Questi modelli che possono essere creati dall'utente, sono messi a disposizione di altri prodotti che ne permettono l'interrogazione tramite query.
- **Pentaho Schema Workbench:** è un'interfaccia che consente di creare e testare schemi Mondrian di cubi OLAP in maniera visuale. Il Mondrian engine processa le richieste MDX (MultiDimensional eXpressions) mediante schemi ROLAP (Relational OLAP). Questi schemi sono costituiti da modelli di metadati in formato XML creati in una specifica struttura dall'engine. L'interfaccia fornisce le seguenti funzionalità:
    - Editor di schemi integrato con data source sottostante per la convalida;
    - Test delle query MDX rispetto allo schema e al database;
    - Esplorazione del database sottostante.

Oltre a questi servizi offerti dalla Community Edition, sono resi disponibili agli abbonati

- della versione Enterprise altre applicazioni server:
- **Pentaho Dashboard Designer (PDD)**: plug-in che consente agli utenti di creare dashboard che forniscono una visione centralizzata degli indicatori chiave di performance e altri aspetti del business. E' supportata un'interfaccia che permette agli utenti di creare dashboard in modo grafico, trascinando i vari oggetti di interesse.

- **Pentaho Analyzer**: fornisce una piattaforma web grazie alla quale l'utente può creare graficamente query MDX ed esportare la tabella risultante in formato pdf o xls. E' noto per poter lavorare su Apple attraverso il browser Safari.
- **Pentaho Interactive Reporting (PIR)**: permette di creare report ad hoc in modo
- grafico trascinando gli oggetti di interesse.
- **Pentaho Mobile**: novità introdotta recentemente che permette di usufruire delle principali funzionalità di analisi OLAP e gestione report e dashboard attraverso piccoli schermi touch screen.
- Oltre agli application server, Pentaho offre una serie di applicazioni desktop per gli utenti, tutte disponibili sia nella CE che nella EE:
- **Pentaho Data Integration (PDI)**: generalmente chiamato Kettle, consiste in un motore ETL che permette l'estrazione, l'integrazione e il caricamento di dati provenienti da sorgenti differenti attraverso un'interfaccia grafica semplificata. Essa permette all'utente di bypassare il puro codice e utilizzare oggetti grafici per rappresentare il flusso di lavoro.
- **Pentaho for Big Data**: è un tool ETL che si basa su Kettle; permette di eseguire job su grandi quantità di dati provenienti da sorgenti come Amazon e altre fonti di dati NoSQL.
- **Pentaho Report Designer**: è un prodotto che permette la generazione di report attraverso interrogazioni di sorgenti differenti. Anche in questo caso, l'utente è facilitato dalla presenza di un'interfaccia grafica semplificata.
- **Pentaho Data Mining**: permette l'elaborazione dei dati, analisi di regressione lineare, metodi di classificazione, interpolazioni etc. Sulla base dei modelli individuati dall'attività di data mining, gli utenti possono eseguire una previsione degli eventi futuri.
- **Pentaho Metadata Editor (PME)**: usato per creare modelli di business; rappresenta un livello di astrazione dell'architettura fisica delle sorgenti. E' uno strumento molto importante a livello aziendale e la sua presenza in una soluzione open source è sicuramente un aspetto positivo.
- Pentaho è un prodotto già molto utilizzato nelle varie realtà aziendali ed è in continuo sviluppo. Presenta delle caratteristiche molto importanti soprattutto per le PMI e può rappresentare un'ottima alternativa a delle soluzioni offerte da prodotti magari molto più costosi.

TIBCO Jaspersoft BI

Jaspersoft BI è una suite di BI caratterizzata da funzionalità complete, architettura leggera e flessibile e dai costi contenuti, se non nulli. Jaspersoft è un progetto nato nel 2001 e portato avanti grazie a finanziamenti provenienti da importanti società.

Nel 2014, Jaspersoft è stata acquistata da TIBCO, società californiana che produce soluzioni software per le aree di Business Management e BI.

Jaspersoft fornisce servizi di reporting, dashboard, analisi e integrazione dei dati, adatti a qualsiasi soluzione di business. Un aspetto importante che caratterizza questo prodotto e che permette alle aziende di migliorare i processi decisionali, è l'utilizzo di un'interfaccia utente web-based che facilita l'uso degli strumenti di analisi messi a disposizione per l'utente. Jaspersoft BI suite include i seguenti prodotti [https://www.jaspersoft.com] :

- **JasperReports Library**: è uno dei motori di reporting open source più popolari, è scritto interamente in Java ed è in grado di utilizzare dati provenienti da qualsiasi sorgente per creare documenti esportabili o stampabili in più formati, quali HTML, PDF, Excel, OpenOffice e Word.
- **Jaspersoft iReport Designer**: tool in grado di creare layout molto sofisticati contenenti grafici, immagini, sottoreport, testi, tabelle a campi incrociati, etc. E' possibile accedere ai vari database attraverso diverse forme di connessione e pubblicare i report in molti formati differenti come PDF, XML, XLS, CSV, HTML, TXT, e DOCS. A partire dall'inizio del 2016, questo componente è stato sostituito da Jaspersoft Studio il quale è comunque in grado di leggere e modificare i report creati con iReport Designer.
- **JasperReports Server**: è un server autonomo e integrabile; fornisce reporting e analisi che possono essere integrati in un'applicazione web o mobile. Questo componente è ottimizzato per condividere, proteggere e gestire i report e le viste realizzate con questa piattaforma. Comprende, inoltre, una serie di strumenti per la creazione e la visualizzazione di dashboard altamente interattivi, grazie ai quali è

possibile rappresentare i dati in maniera grafica. E' anche possibile navigare all'interno delle informazioni rappresentate, al fine di analizzare le varie aree aziendali di interesse.

- **Jaspersoft ETL**: costituisce il tool back-end della piattaforma e consente l'estrazione, l'integrazione e il caricamento dei dati da sorgenti differenti in un DW o in uno specifico Data Mart. Permette di definire i flussi di lavoro e può collegarsi sia a sorgenti proprietarie che a quelle aperte.
- **Jaspersoft OLAP**: permette di manipolare, modellare e visualizzare qualsiasi tipo di dato attraverso un'analisi di tipo multidimensionale (OLAP), al fine di individuare i problemi e i trend nell'azienda e aiutare i decision maker a prendere migliori decisioni in tempi più brevi.

Esistono cinque versioni della suite di Jaspersoft: Community, Reporting, AWS, Professional ed Enterprise. Per le finalità di quest'analisi, vale la pena soffermarsi solamente sulla versione Community e su quella Enterprise. La prima, che comprende le funzionalità di base della piattaforma Jaspersoft, è soggetta ad una licenza AGPL e rientra quindi nella tipologia di software open source, mentre la seconda è quella più completa che, però, prevede una licenza commerciale basata sul numero di core CPU forniti all'utente. Chiaramente, questa versione offre molte funzionalità aggiuntive, come il supporto tecnico professionale, reporting ad hoc, visualizzazioni basate su Flash e molto altro.

La suite BI Jaspersoft è sicuramente una piattaforma orientata al front-end e, quindi, al reporting e all'analisi dei dati, ma presenta al suo interno una serie di tool che la rendono completa in ambito BI, come lo strumento ETL per la creazione di Data Warehouse e Data Mart e la gestione dei metadati. Non mancano, però, alcuni aspetti negativi, messi in evidenza anche dallo studio effettuato da Gartner del mercato degli strumenti BI: la complessità di apprendimento da parte degli utenti, nonostante la presenza di interfacce grafiche, e l'assistenza e il supporto al cliente che non sono all'altezza di altri software concorrenti nel mercato.

Eclipse BIRT Project

BIRT (BI and Reporting Tools) Project, rappresenta un progetto open source fornito dalla piattaforma tecnologica BIRT per creare visualizzazioni di dati e report, che possono essere incorporate in ricche applicazioni client e web. E' un progetto software di alto livello

all'interno della Eclipse Foundation, un consorzio indipendente e no-profit di fornitori del settore software e una comunità open source [https://eclipse.org/birt].

BIRT comprende due componenti principali: un report designer basato su Eclipse e un componente runtime per la generazione di report implementabile in qualsiasi ambiente Java. Include, inoltre, un motore per produrre e integrare grafici all'interno delle applicazioni. Le varie tipologie di visualizzazione dei dati che è possibile inserire nelle applicazioni sono: liste di dati, grafici (rappresentazioni grafiche dei dati), tabelle a campi incrociati (visualizza i dati in due dimensioni), lettere e documenti. Spesso un report potrebbe necessitare di più tipologie tra quelle sopra citate; in questi casi i report vengono definiti composti.

Il progetto BIRT comprende molti componenti e molti plug-in che vanno a completare le funzionalità delle soluzioni offerte. Di seguito verranno presentati i componenti:

- **BIRT Report Designer**: è un componente di Eclipse usato per la progettazione di report salvati in un formato XML. Questo componente può essere scaricato come una piattaforma rich client (RCP), cioè uno strumento di programmazione che rende più facile l'integrazione di più componenti software indipendenti. Il trattamento dei dati avviene principalmente sul lato client, oppure come un all in one download di Eclipse.
- **Design Engine**: è il motore responsabile della creazione e della modifica dei report. L'API (Application Programming Interface) del Design Engine viene utilizzata da questo componente per la creazione delle presentazioni XLM.
- **Report Engine**: svolge la funzionalità di generare i report. Utilizzando l'API del Report Engine, il motore può essere integrato all'interno di qualsiasi applicazione Java.
- Charting Engine: è utilizzato per generare grafici sia in stand-alone che all'interno di report BIRT. Grazie all'API del Charting Engine, il Design e il Report Engine permette di fornire grafici di vario tipo.
- **BIRT Viewer**: il progetto BIRT fornisce questo componente che permette di visualizzare i report all'interno di Eclipse. L'output fornito da questo viewer può essere in formato HTML, PDF, XLS, DOC, PPT. Inoltre, l'utente può esportare i dati del report in formato CSV e stamparli.

BIRT è, attualmente, una delle piattaforme più usate per la visualizzazione e il reporting dei dati. Tra gli aspetti positivi di BIRT, c'è sicuramente la capacità di integrarsi con molte

sorgenti dati in diversi ambienti. In particolar modo, è in grado di integrare dati provenienti da database SQL, che sono tra i più diffusi, ma anche da altri tipi di sorgenti.

Per concludere, in Tabella 2, viene proposta un sinottico riassuntivo del processo di analisi dei prodotti e delle soluzioni tecnologiche per la BI di tipo Open Source. Dal lavoro svolto è emerso che tutti i prodotti sono caratterizzati da un supporto al reporting molto valido e all'altezza, se non superiore, a quelli offerti da altri prodotti presenti sul mercato. In questo ambito, il prodotto migliore risulta essere Jaspersoft, anche se, da quanto emerge dalle valutazioni generali fatte, Pentaho è considerato generalmente migliore rispetto agli altri due prodotti open source

PRODUTTORE	**ETL e DW**	**Reporting**	**Dashboarding**	**Predictive Analysis**
PENTAHO	Pentaho Data Integration	Pentaho Interactive Reporting Pentaho Report	Pentaho BI Pentaho Dashboard Designer	Pentaho Data Mining
JASPERSOFT	Jaspersoft ETL	JasperReports Library Jaspersoft iReport Designer JasperReports Server	JasperReports Server	-
BIRT	-	Eclips BIRT project	Eclips BIRT project	-

Tabella 2 - Sinottico riepilogativo dei sistemi di BI Open Source

L'aspetto che fa la differenza è costituito dai componenti di back end; Pentaho infatti, offre uno strumento ETL (Pentaho Data Integration) nettamente superiore alle altre soluzioni offerte sia da BIRT che da Jaspersoft e offre agli utenti un'alternativa che si avvicina molto a quelle dei prodotti storici. BIRT risulta essere un prodotto valido soprattutto nella parte di reporting, anche se, messo a confronto con i diretti concorrenti, è caratterizzato da una

valutazione complessivamente inferiore, in quanto non offre una suite completa come Pentaho. Pertanto, sulla base di quanto analizzato, per la parte di BI si è scelto di usare la suite CE di Pentaho.

1.5 Infrastruttura sistema R e IT

R è un ambiente integrato per l'analisi dei dati nato nel 1993 dall'elaborazione del linguaggio di programmazione S (ideata da John Chambers presso i Bell Laboratories) ad opera di Robert Gentleman e Ross Ihaka, colleghi presso l'Università di Auckland. L'interfaccia grafica RStudio, basato sullo stesso linguaggio di programmazione, è distribuito con la licenza GNU GPL, e disponibile per diversi sistemi operativi (ad esempio Unix, GNU/Linux, macOS, Microsoft Windows). La caratteristica open source del software ha permesso a un gran numero di sviluppatori di programmare appositi pacchetti da caricare in R per utilizzare diverse tipologie di tecniche statistiche per l'analisi dati (R Core Team 2016).

Le caratteristiche principali di RStudio sono :

- la capacità di gestione e manipolazione dei dati;
- l'accesso ad un vasto insieme di strumenti integrati per l'analisi statistica;
- la potenzialità grafiche particolarmente flessibili;
- la possibilità di adoperare un vero e proprio linguaggio di programmazione orientato ad oggetti che consente l'uso di strutture condizionali e cicliche, nonché di funzioni create dall'utente.

Grazie a queste caratteristiche RStudio sta diventando il riferimento sia non solo per l'accademia: Bank of America, Facebook, Ford, NewScientist,The New York Times, FDA sono solo alcune delle aziende che utilizzano R per la gestione dei loro dati. L'ambiente R è diviso in 2 parti concettuali: il sistema R "base" che si scarica da CRAN, e tutto il resto. R è suddiviso in un certo numero di pacchetti, tra cui quelli base necessarie per eseguire R e le funzioni fondamentali. Gli altri pacchetti contenuti nel sistema "base" includono utils, stats, datasets, graphics, grDevices, grid, methods, tools, parallel, compiler, splines, tcltk, stats4. In funzione delle proprie finalità è possibile scaricare, usare e implementare numerosi pacchetti, tra cui rgdal, sp, ggplot, shiny, etc.. Le entità che R crea e manipola sono note come oggetti.

Questi possono essere variabili, array di numeri, caratteri stringhe, funzioni o più in generale strutture costruite a partire da tali componenti. Tra gli oggetti più usati in questo lavoro di tesi ci sono le liste e i dataframe. Per questo motivo di seguito definiremo che cosa sono le liste e i dataframe.

1.5.1 Oggetti di R: liste e dataframe

Una lista in R è un oggetto che consiste di un insieme ordinate di altri oggetti, che sono chiamati componenti della lista. Un semplice esempio di lista è:

Lst<- list(nome = "Ugo", moglie = "Maria", nu.figli = 3, eta.figli = (4,7,9))

Le componenti sono sempre numerate e ci si può riferire ad esse singolarmente attraverso degli indici. In particolare, per l'esempio visto si ha Lst[[1]], Lst[[2]], Lst[[3]], Lst[[4]]. Inoltre per la quarta componente si possono individuare le sottocomponenti Lst[[4]] [1], Lst[[4]] [2] e Lst[[4]] [3]. Le componenti di una lista possono essere richiamate anche specificando il loro nome; ad esempio lst$nome è lo stesso di Lst[[1]] e corrisponde alla stringa Ugo.

Un dataframe è una lista di vettori (le variabili), che devono avere tutti la stessa lunghezza (numero di casi), ma possono essere di tipo diverso: variabili nominali (fattori), variabili cardinali (vettori numerici), etc.. Un dataframe può essere costruito inserendo i dati direttamente in R, oppure, come più comunemente avviene, importando i dati da altre applicazioni. Una volta caricato, un dataframe diventa un oggetto del Workspace (ambiente di lavoro), e può essere richiamato nei comandi per poter svolgere l'analisi dei dati. All'interno del workspace possono essere disponibili diversi dataframe alla volta. Questo tipo di strutture dati si prestano bene al processamento e alla gestione di informazioni complesse come quelle presenti in un dato georeferenziato o da georeferenziare. L'utilizzo di un software sviluppato in RStudio può risultare complesso a chi non ha confidenza con questo linguaggio di programmazione, tuttavia, grazie ad alcuni pacchetti e librerie rilasciate dagli sviluppatori è possibile creare delle interfacce grafiche *user friendly* utili per rendere il software usufruibile anche a chi non ha particolari competenze di programmazione o di informatica. I software con interfaccia *user – friendly* possono eseguire un numero limitato e controllato di operazioni prefissate, ma in modo rapido e generalizzabile. La maggior parte dei software statistici con interfaccia utente esistenti sono di tipo commerciale, e non sempre

consentono di condurre sia l'analisi statistica che la geo-referenziazione dei risultati come richiesto in questo studio. Pertanto, a partire dalla conoscenza del linguaggio R si è deciso di progettare ed implementare la maggior parte delle operazioni necessarie in un vero e proprio software interattivo, ancora una volta grazie alle risorse sviluppate nel più vasto ambiente open source di RStudio sfruttando i pacchetti e le librerie già esistenti e disponibili nelle repository.

1.5.2 La georeferenziazione: definizione e applicazione in RStudio

In generale la georeferenziazione è la tecnica che permette di associare ad un dato, in formato digitale, delle coordinate che ne fissano la posizione sulla superficie terrestre. Questo processo può essere più o meno facilitato in funzione del software, del database e del tipo di geometria che si intende assegnare ai dati. Per esempio, nel caso di un'informazione di tipo puntuale, al dato sono assegnate le due coordinate spaziali. Pertanto la struttura di questo file consisterà in n righe, che rappresentano il numero di oggetti che si intende georeferenziare, e in x colonne, di cui le prime due obbligatorie contenenti le coordinate geografiche (latitudine e longitudine), mentre le altre colonne possono contenere qualsiasi tipo di informazioni in formato alfanumerico. Questo file può essere salvato come file di testo (txt) e importato in diversi software commerciali di georefereziazione sia open source (QuantumGis) che non (ArcGis®). Dopo aver importato il file di testo nei software di georeferenziazione è possibile creare ed esportare lo strato informativo come shape file. Il formato shape è stato sviluppato e regolato da Esri, allo scopo di accrescere l'interoperabilità fra i sistemi ESRI e altri GIS. Di fatto è diventato uno standard per il dato vettoriale spaziale. Con "shapefile" si indica di norma un insieme di file con estensione .shp, .dbf, .shx, .qpj e .prj che hanno in comune il prefisso dei nomi. Dal punto di vista geometrico è possibili georeferenziare punti, linee e poligoni in funzione del dato di origine e dello scopo del lavoro (Fig. 3).

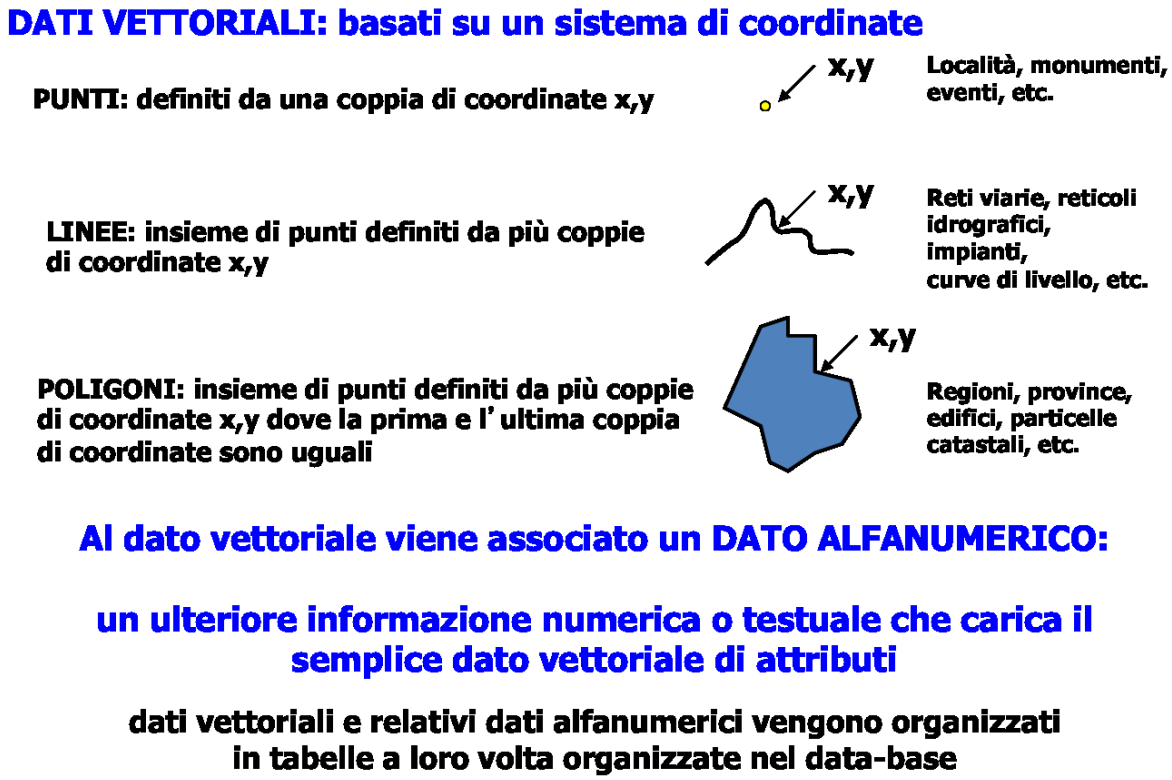


Figura. 3. Definizione dei dati vettoriali.

In questa tesi, sebbene nelle CCE siano presenti gli indirizzi dei pazienti, i dati sanitari saranno associati al comune di residenza del paziente per motivi di privacy. Un comune è rappresentato come un poligono, ovvero una figura geometrica piana delimitata da una linea spezzata chiusa; pertanto, ogni poligono è rappresentato non da una singola coordinata ma da un insieme di coordinate afferenti allo stesso gruppo (group) dove le prime e ultime coordinate coincidono. Quindi, ogni attributo deve essere duplicato n volte per tutte le n righe afferenti allo stesso gruppo. Mentre questa operazione avviene in maniera nascosta e automatica nei software commerciali *open source* e non, nel caso della creazione, e quindi del pre-processamento dei dati da georeferenziare, questa operazione avviene da linea di comando mediante l'utilizzo di pacchetti già presenti nelle repository R, quali rgdal, sp e broom.

Per procedere alla georeferenziazione dei dati è necessario innanzitutto procedere con la creazione degli Spatial Dataframe, ovvero un dataframe contenenti le informazioni

geografiche quali coordinate, sistema di riferimento e sistema di proiezione. Questo tipo di struttura dati consiste nell'unione di:

- un dataframe, una "tabella" contenente i dati (attributi) da georeferenziare
- shape file contenenti le informazioni geografiche come i confini amministrativi regionali, provinciali e comunali.

Sia i Dataframe che gli shape file sono importabili in RStudio; i Dataframe tramite le funzioni base di R, mentre gli shape file e mediante pacchetti specifici per la georeferenziazione quali "sp" e "rgdal". Questi ultimi pacchetti forniscono un solido set di classi per differenti tipi di dati spaziali supportando direttamente le classi di oggetti di carattere spaziale (*e.g.*, leggendoli o scrivendoli), o convertendoli in una classe ad essi appropriati. In generale, i dati geografici hanno due dimensioni proiettate su una superficie piana (mappa) o su una sfera (la Terra). Aspetto centrale dei dati spaziali è che essi posseggono un sistema di riferimento, il quale è codificato in un oggetto di classe CRS. Il CRS è una stringa di caratteri che descrive il sistema di riferimento e di proiezione in modo da essere "capito" dalla libreria PROJ.4 disponibile nel pacchetto rgdal. Tutte le classi spaziali derivano da una classe spaziale base che fornisce solo i confini e le informazioni circa il sistema di riferimento e di proiezione. E' fondamentale che i dati spaziali su cui si effettuano operazioni abbiano lo stesso sistema di riferimento e di proiezione. A tal proposito è possibile convertire tutti gli oggetti spaziali importati in RStudio nello stesso sistema di riferimento e di proiezione mediante la funzione spTransform prevista nel pacchetto rgdal. Tra gli oggetti di classe spaziale i principali sono gli SpatialPoints che riferiscono ai punti, le SpatialLines che si riferiscono alle linee e gli SpatialPolygons che fanno riferimento ai poligoni. La costruzione di uno Spatial dataframe non fa altro che estendere l'oggetto Spatial (-Points, o Lines, o Polygons) con uno slot di dati, al cui interno può essere salvato un dataframe contenente differenti e vari attributi. In questo modo è possibile importare e integrare qualsiasi file di testo con la componente spaziale e, quindi, georeferenziare i dati con la creazione dello SpatialPolygons dataframe mediante la funzione *merge*. La particolarità di questi tipi di dati è che, riferendosi a poligoni (es. comuni), che dal punto di vista geometrico sono costituiti da n linee chiuse ognuna con una sua coordinata, è necessario duplicare gli attributi n volte quante sono le linee usate per comporre il poligono. Una volta

creati è possibile rappresentare in maniera grafica gli Spatial Dataframe grazie a uno dei pacchetti più usato in R quale GGPLOT2. In generale, le analisi di tipo esplorativo richiedono l'utilizzo di elaborazione di immagini dettagliate che permettano di studiare i risultati ottenuti in maniera chiara. GGPLOT2 mette a disposizione dell'utente funzioni avanzate per lo sviluppo di grafici, i quali possono essere completamente personalizzabili a seconda del tipo di interrogazione che si vuole effettuare sui dati aggiungendo funzioni dalla linea di comando usata per creare il grafico principale [Wickham, 2009]. E' inoltre possibile rendere dinamici i grafici prodotti in GGPLOT2 grazie al pacchetto Plotly. Plotly consente la creazione e la condivisione di "*data visualization*" ed offre anche uno strumento per svolgere analisi statistiche. Inoltre questo pacchetto offre la possibilità di disegnare funzioni personalizzate, e una shell Python integrata. Le visualizzazioni interattive offerte da Plotly possono essere create direttamente in R. Grazie a Plotly, gli utenti R possono facilmente creare un grafico interattivo in R con poche righe di comando, o rendere interattivo, mediante l'apposita funzione *ggplotly*, quello statico ottenuto in GGPLOT. La creazione degli Spatial Dataframe e la produzione della mappa avviene da linea di comando, rappresentando un limite per la diffusione e l'utilizzo del software.

Per rendere i software user friendly è possibile costruire delle Graphic User Interface (GUI) grazie a librerie e pacchetti come Shiny.

1.5.3 Sviluppo GUI: Shiny app

Shiny consiste in un'applicazione web pensata appositamente per R, con la quale è possibile trasformare i comandi per l'analisi statistica in veri e propri software [Chang et al. 2017]. I software statistici con **interfaccia utente (user-friendly)** consentono di rendere più rapida ed agevole l'estrazione di informazioni di svariato tipo dai dati rispetto a quanto si potrebbe ricavare attraverso l'utilizzo di funzioni da semplice linea di comando. Infatti, le funzioni da linea di comando hanno il vantaggio di poter (in linea di principio) eseguire azioni diverse, ma devono essere costantemente adattate a seconda del tipo di dati su cui effettuare le analisi e del tipo di risultato che emerge. Viceversa, i software con interfaccia user – friendly possono eseguire un numero limitato e controllato di operazioni prefissate, ma in modo rapido e generalizzabile.

Più in generale, i vantaggi di sfruttare una tipologia di programma geo-spaziale che si basa sulla logica del "point & click" e sulla visualizzazione di mappe e strati informativi, sono diversi ed estremamente utili:

- grazie a un'interfaccia utente costituita da sezioni ben definite, da menu a tendina ed input di comandi intuitivi, risulta più facile a chiunque voglia interrogare i dati scegliere il tipo di analisi da eseguire senza dover conoscere un linguaggio di programmazione;
- data la velocità con cui si possono ricavare informazioni, si aumenta il numero di indagini che è possibile effettuare a parità di tempo, e quindi aumenta la possibilità di ricavare un numero di conclusioni maggiore;
- la possibilità di inserire grafici esplorativi di tipo interattivo rende più facile estrarre informazioni riguardo gruppi di unità statistiche o singoli campioni, visto che anche solo passando il cursore su di un elemento del grafico è possibile da subito conoscere l'identificativo di quel particolare campione, il gruppo a cui appartiene, o altre informazioni rilevanti.

La maggior parte dei software di georeferenziazione sono di tipo commerciale con qualche eccezione open-source, tuttavia, in rari casi, è stata prevista un interfacciamento con piattaforme e strumenti di BI come richiesto in questo studio. Pertanto, a partire dalla conoscenza del linguaggio *R* e dopo aver identificato tutte le funzioni necessarie per gli raggiungere gli obiettivi prefissati in questo studio, è stato sviluppato un vero e proprio software interattivo con un'interfaccia grafica *user-friendly* grazie alle risorse sviluppate nel più vasto ambiente open source di *RStudio*. La realizzazione di un software è costituita da due fasi principali. La prima è quella di **progettazione**, che prevede la definizione di strutture di dati su cui il software deve operare. Bisogna infatti specificare quali file possono essere importati all'interno del software e soprattutto il tipo di formato tabellare in cui i dati devono trovarsi, perché i successivi comandi devono operare esclusivamente su una struttura organizzativa dei dati predefinita. Nella fase di progettazione è anche prevista la definizione di tutte le funzioni che il software darà la possibilità di svolgere per condurre l'analisi geo-statistica dei dati, iniziando così anche a capire come strutturare l'interfaccia utente attraverso cui è possibile eseguire tutti i comandi in maniera organizzata e soprattutto logica.

La fase successiva, quella di **implementazione**, prevede la traduzione dei codici scritti in un linguaggio di programmazione capace di generare degli output basati sulle funzioni geo-

statistiche presenti nel software. Questi output possono essere di tipo grafico o dei veri e propri file scaricabili: una volta definito il tipo di output che si vuole ottenere, bisogna associare ad esso delle funzioni statistiche sotto forma di linguaggio di programmazione *R* che si adattano al tipo di input, riescono ad interpretarlo e applicare su di esso una funzione definita dal programmatore. La peculiarità di un software risiede nel fatto che quando l'input cambia, cambia rapidamente anche il risultato che si ottiene.

Il pacchetto Shiny mette a disposizione dell'utente tutta una serie funzioni che permettono i) di costruire l'interfaccia utente (tipo web-based) attraverso cui utilizzare il software, ii) di poter interagire con l'interfaccia per l'acquisizione di input, scelte di parametri e analisi da effettuare, iii) di poter eseguire le istruzioni richieste (comandi) al fine di eseguire calcoli, analisi o produrre grafici, iv) di mostrare i risultati (grafici e/o tabelle) ottenuti.

2 Architettura del Data Warehouse

2.1 Introduzione

L'utilizzo dei Data Warehouse (DW) nasce con l'obiettivo di rendere l'informazione aziendale accessibile, consistente, affidabile e usabile per il supporto alle decisioni. I DW sono costituiti da una base di dati che si distingue da un comune database per i seguenti punti caratteristici:

- Utilizzata principalmente per il supporto alle decisioni direzionali;
- Integrata al livello aziendale più alto:
- Orientata ai dati e non alle applicazioni;
- Estesa ad un ampio orizzonte temporale;
- Non volatile: i dati sono caricati ed acceduti fuori linea;
- Mantenuta separatamente dalle basi di dati operazionali.

L'obiettivo principale nell'uso di un DW è quello di eseguire interrogazioni (query) relativamente semplici che operano su una grande mole di dati (spesso dell'ordine dei terabyte); una volta definito il concetto sul quale effettuare l'analisi (fatto) e gli attributi attraverso cui si vogliono fare le ricerche (dimensioni), le interrogazioni applicheranno le regole attraverso cui arrivare al dato richiesto (misura). Il modello logico dei dati più adatto è quello di una struttura multidimensionale (data cube) nella quale le chiavi di ricerca rappresentano le dimensioni della struttura (cubo); le dimensioni possono a loro volta avere una struttura gerarchica mentre le celle del cubo rappresentano i valori metrici da misurare.

L'architettura di un DW si basa su diverse forme standard, sviluppate principalmente su uno, due e tre livelli [INMON et al. 2005] [DEVLIN et al. 1996]. Tralasciando l'architettura a un livello basata su un data warehouse virtuale ovvero implementato come una vista multidimensionale dei dati operazionali generata da un apposito strato di elaborazione intermedio. Tali livelli sono:

- *Livello delle sorgenti:* che rappresenta i database di interesse aziendali, relazionali o legacy siano essi interni o esterni all'organizzazione;
- *Livello dell'alimentazione:* in cui i dati memorizzati nelle sorgenti devono essere estratti, ripuliti (per eliminare le inconsistenze e completare eventuali parti mancanti)

e integrati (per fondere sorgenti eterogenee secondo uno schema comune). I cosiddetti strumenti ETL (Extract, Transform and Loading) permettono di integrare schemi eterogenei, nonché di estrarre, trasformare, ripulire, validare, filtrare e caricare i dati dalle sorgenti nel DW. Dal punto di vista tecnologico vengono trattate problematiche tipiche dei servizi informativi distribuiti, come la gestione di dati inconsistenti e delle strutture dati incompatibili;

- *Livello del Warehouse:* che raccoglie le informazioni in un singolo "contenitore" (il data warehouse), centralizzato logicamente. Esso può essere direttamente consultato, ma anche usato come sorgente per costruire dei Data Mart orientati verso specifiche aree dell'impresa. Accanto al DW, il contenitore dei metadati mantiene informazioni sulle sorgenti, sui meccanismi di accesso, sulle procedure di pulitura ed alimentazione, sugli utenti, sugli schemi dei Data Mart, ecc.;
- *Livello di Analisi:* che permette la consultazione efficiente e flessibile dei dati integrati per la stesura dei report nonché per le attività di analisi e di simulazione.

La progettazione di un data warehouse può avvenire secondo due modalità diverse:

- top-down: previsione contemporanea delle esigenze di tutti gli utenti; analisi di tutte le fonti informative necessarie; lunga fase di analisi da cui discenderanno tutti i data mart previsti dalla procedura;
- bottom-up: costruzione incrementale ottenuta assemblando i vari data mart costruiti; primi risultati visibili in tempi brevi; dall'individuazione dei dati necessari ad ottenere il primo data mart, si ottengono le prime entità logiche utilizzabili in seguito anche per lo sviluppo di data mart diversificati.

La progettazione prevede generalmente varie fasi ma non esiste un modello progettuale standard da poter seguire nelle fasi architetturali. Le caratteristiche principali su cui in genere basa la progettazione dell'architettura di un DW sono:

- *Separazione*: in modo da tenere l'elaborazione analitica e quella transazionale il più possibile separate;
- *Scalabilità*: l'architettura hardware e software deve poter essere facilmente ridimensionata a fronte della crescita nel tempo dei volumi di dati da gestire ed elaborare e del numero di utenti da soddisfare;
- *Estensibilità*: deve essere possibile accogliere nuove applicazioni e tecnologie senza riprogettare integralmente il sistema;

- *Sicurezza*: il controllo sugli accessi è essenziale a causa della natura strategica dei dati memorizzati;
- *Amministrabilità*: la complessità dell'attività di amministrazione non deve risultare eccessiva.

2.2 Progettazione e popolamento del Data Warehouse

Nel presente paragrafo viene descritta la progettazione dei componenti della piattaforma alimentata dal Data Warehouse, e i cui componenti architetturali vengono riportati per blocchi in Figura 4.

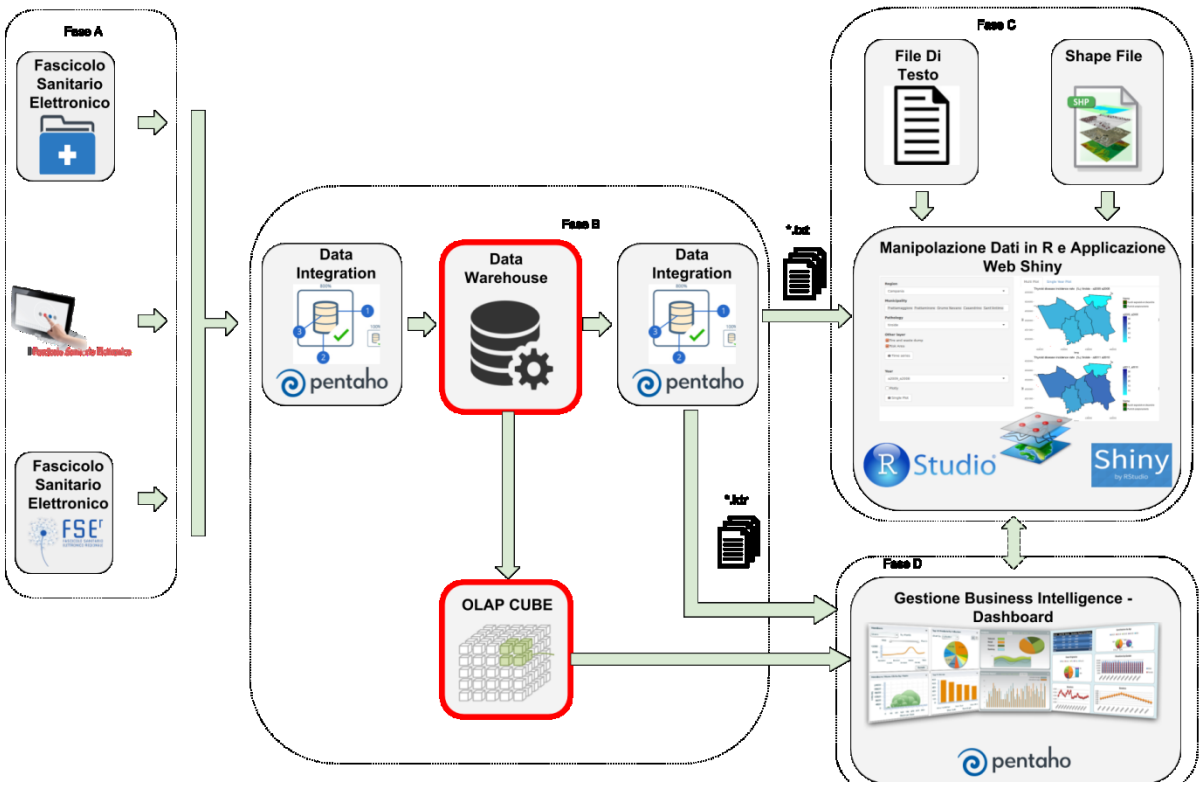


Figura 4 - Architettura piattaforma

Per il progetto è stato scelto un'architettura a due livelli in grado di evidenziare la separazione tra il livello sorgenti e quello del DW stesso come riportato in Figura 5.

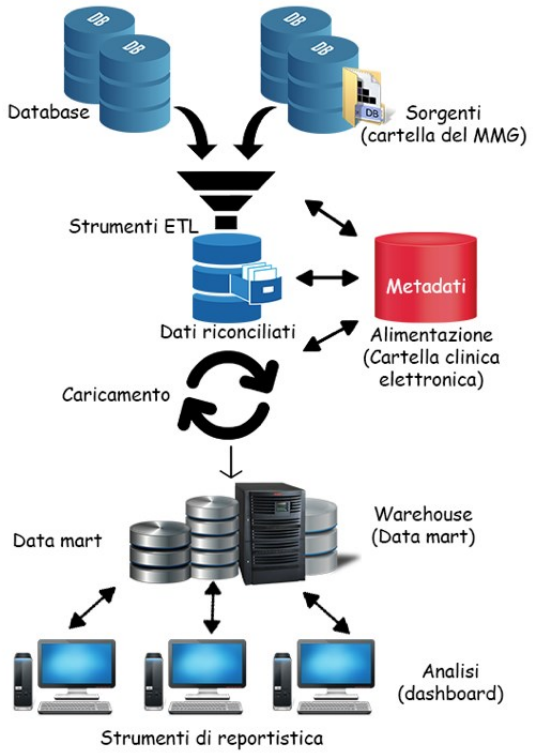


Figura 5 - Architettura a due livelli DW

Con riferimento alla Figura 5, il blocco (detto primario) che va sotto il nome di DW si occupa di raccogliere i dati di sintesi provenienti dai sistemi che alimentano il warehouse a livello centrale. A corredo del warehouse primario ci sono i cosiddetti Data Mart che invece possono essere visti come piccoli warehouse locali che replicano (ed eventualmente sintetizzano ulteriormente) le informazioni di interesse. Tale soluzione ha il vantaggio di snellire le fasi di progettazione ma determina uno schema complesso per l'accesso ai dati e ingenera il rischio di inconsistenza tra i Data Mart.

Nel caso applicativo oggetto della tesi, i blocchi della Figura 4 sono stati caratterizzati come segue:

- *Livello delle sorgenti:* rappresenta l'insieme dei sistemi informativi o dei repository che gestiscono e contengono le informazioni relative alle cure dei pazienti (cartella dello specialista, del MMG, sistema informativo ospedaliero, ecc.). Tali sistemi alimentano in modo diretto la Cartella Clinica Elettronica;

- *Livello di alimentazione:* La Cartella Clinica Elettronica stessa rappresenta il livello di alimentazione poiché contiene seppur in una logica transazionale tutte le informazioni relative al singolo paziente e ai servizi erogati, oltre ai metadati utili per l'indicizzazione dei documenti clinici strutturati contenuti nei repository;
- *Livello del warehouse*: questo livello rappresenta la parte più consistente del lavoro, oltre alle modalità di caricamento dei dati contenuti nella Cartella, descritti nei prossimi paragrafi;
- *Livello di analisi:* rappresentato dagli strumenti di data mining, OLAP e di reportistica utili per l'analisi dei dati contenuti nel data warehouse.

Nel progetto della piattaforma oggetto del lavoro di tesi, si è utilizzato per il data warehouse l'approccio bottom-up, individuando un primo data mart da sviluppare ed identificando di conseguenza le entità logiche necessarie. Si è partiti dall'individuazione del *fatto* di interesse, sono state poi applicate ad esso le necessarie *dimensioni* sulle quali si sono effettuate le *misure*. Nella Tabella 3 viene riportato il dettaglio.

FATTO	**POSSIBILI DIMENSIONI**	**POSSIBILI MISURE**	**STORICITA'**
Assistiti affetti da patologia (Assistiti Malati)	Patologia Luogo Sesso Età	Numero Assistiti Impegno di Spesa	5 anni

Tabella 3 - primo data mart

Partendo da queste considerazioni si è passati a definire un possibile albero degli attributi che mostra sinteticamente le dipendenze funzionali così come gli identificatori e le associazioni (si veda Figura 6).

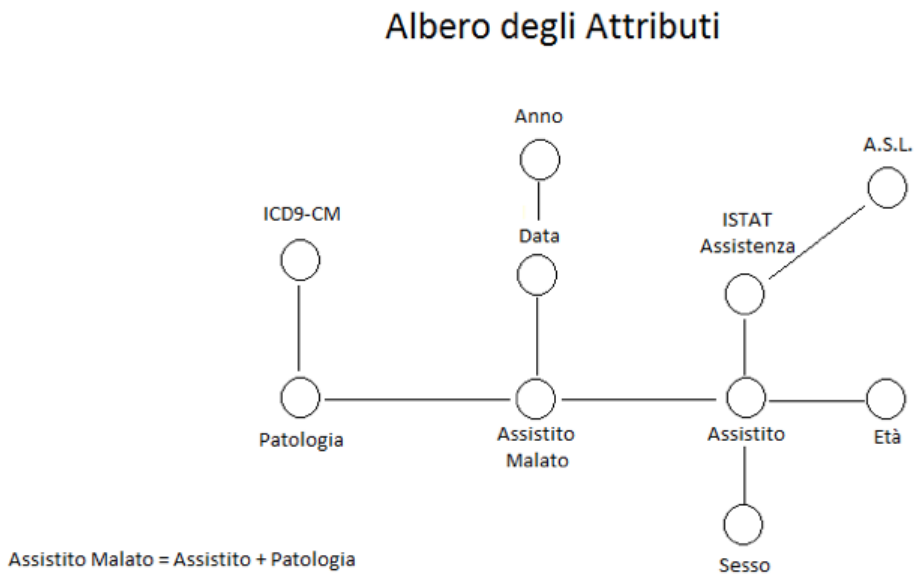


Figura 6 - Albero degli Attributi

In questa rappresentazione, ogni vertice corrisponde ad un attributo del concetto da cui ha origine, mentre ogni radice corrisponde ad un identificatore del *fatto*. Da questa rappresentazione è poi possibile ricavare il primo *diagramma a stella* che corrisponde alla struttura logica del data mart per il *fatto* identificato (si veda Figura 7).

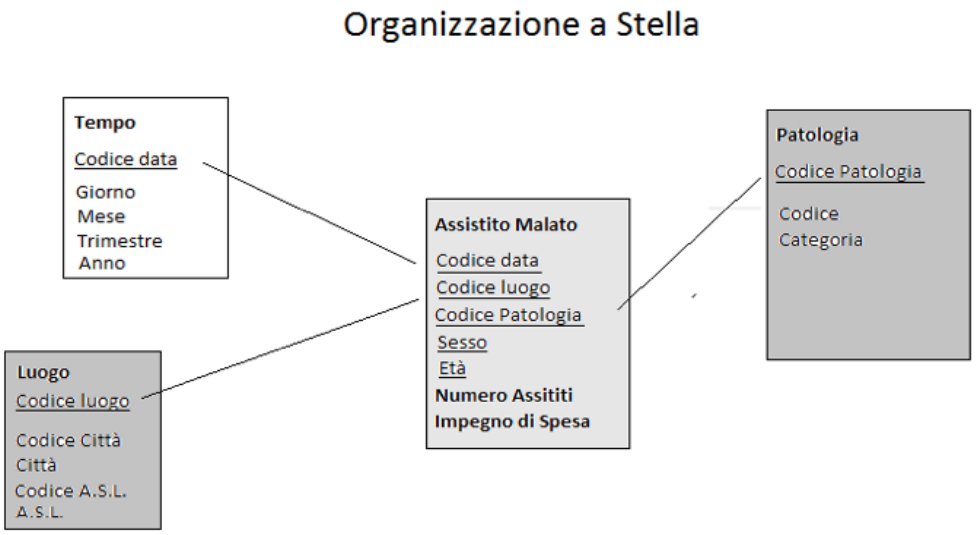


Figura 7 - Organizzazione a Stella

In questo diagramma sono mostrate sia le chiavi di ricerca (*dimensioni*) che le aggregazioni che è possibile effettuare (*misure*) su tali chiavi. Nel diagramma, per brevità, non vengono mostrate le dimensioni sesso ed età.

Dimensioni

Le dimensioni principali individuate sono, oltre al *Tempo*, il *Luogo* e la *Patologia*. Il *Luogo* è rappresentativo della posizione geografica dove l'assistito vive e viene pertanto assistito nelle sue eventuali patologie mentre la *Patologia* è identificata mediante una codifica internazionale appropriata. Per quanto riguarda la dimensione *Luogo*, è stata utilizzata la codifica ISTAT dei comuni d'Italia raggruppati secondo le rispettive AA.SS.LL. di appartenenza, anch'esse codificate secondo i codici ISTAT. Per la dimensione Patologia è stata utilizzata la codifica ICD9-CM per l'attributo *Codice* e la codifica ICPC-2R per l'attributo *Categoria*. La codifica internazionale ICPC-2R mappa *uno-a-molti* i codici ICD9-CM in modo da poterli raggruppare in categorie più comprensibili rispetto ai singoli capitoli di cui è composta la codifica ICD9-CM.

Entità

Le entità da individuare per poter rispondere alle esigenze espresse da un data mart dipendono fondamentalmente dai dati operazionali disponibili. In molti casi, infatti, alla mancanza di uno o più dati necessari è possibile "sopperire" con algoritmi logici che consentono almeno di approssimare il/i dato/i mancante/i.

Il Sistema realizzato ottiene i dati di ingresso dalle cartelle cliniche dei medici di medicina generale (di seguito MMG) raccolti in una base dati per la medicina di rete. Le tabelle che costituiscono le basi dati degli MMG contengono le Entità Operazionali riportate in Figura 8.

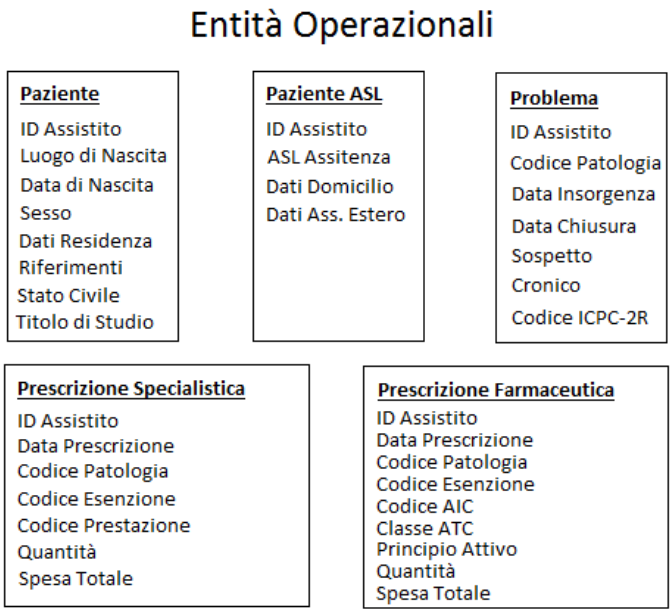


Figura 8 - Entità Operaionali

Analizzando i dati contenuti in tali tabelle, sono state fatte le seguenti considerazioni:

- *Luogo*: il luogo di assistenza è stato identificato in base al domicilio e/o alla residenza dell'assistito, ricavabile dai rispettivi indirizzi che i MMG memorizzano nelle proprie cartelle cliniche
- *Patologia*: le eventuali patologie di cui gli assistiti sono affetti sono anch'esse memorizzate nelle cartelle cliniche dei MMG ma analizzando questo particolare dato si è osservato che, per la maggior parte dei casi, la data di insorgenza della patologia era quasi sempre mancante o impostata con un valore di default tipico di ciascuna cartella clinica; ciò impediva al data mart di applicare la dimensione *Tempo* in modo corretto. Per ovviare a questa mancanza di dati, si è cercato nella base dati disponibile qualche altra informazione che potesse legare l'assistito alle sue patologie individuando tale legame nelle prescrizioni che il MMG effettua per i suoi assistiti alle quali associa la codifica della patologia per la quale prescrive. Pertanto è stato implementato un algoritmo in grado di determinare le patologie di un assistito partendo dalla considerazione secondo la quale un assistito è affetto da una particolare patologia se presenta almeno due prescrizioni fatte per la stessa patologia nell'anno considerato.

Da quanto detto, sono state quindi identificate le seguenti entità come base del DW da cui estrarre il data mart in sviluppo. E' importante notare come l'entità Prescrizione non appaia nel diagramma a stella mostrato prima ma risulti necessaria in base all'analisi fatta sui dati disponibili.

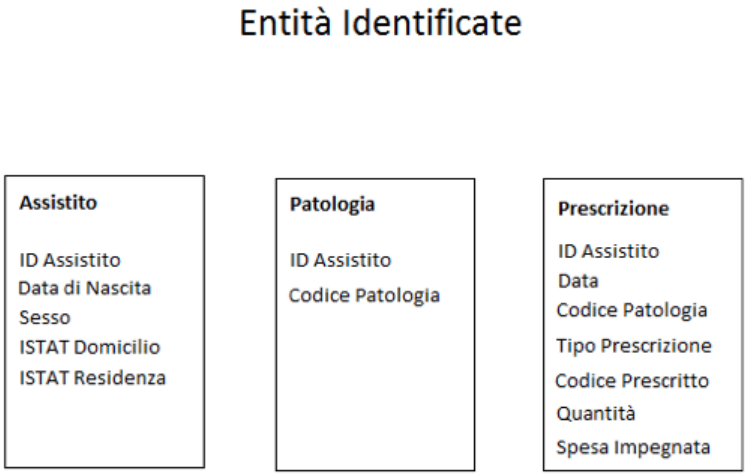


Figura 9 - Entità identificate

Le entità indicate in Figura 9 mostrano come, utilizzando l'approccio bottom-up e analizzando i dati necessari allo sviluppo di un data mart, si possano identificare altri attributi nella entità coinvolte, tali da consentire lo sviluppo di ulteriori data mart. Ad esempio, considerando l'entità *Prescrizione* (utilizzata, ricordiamo, solo per attivare l'algoritmo di identificazione dell'assistito affetto da patologia), si può osservare come sia possibile sviluppare ulteriori data mart nei quali i *fatti* potrebbero essere la spesa impegnata per una determinata patologia in un determinato periodo di tempo quanto piuttosto i farmaci prescritti per classe ATC (l'attributo Tipo Prescrizione consente di suddividere le prescrizioni tra farmaci e specialistica per cui l'attributo Codice Prodotto individua o l'AIC di un farmaco o il codice nomenclatore di una prestazione di specialistica ambulatoriale).

Le entità sopra descritte andranno caricate con le opportune procedure di ETL (Extract, Transform, Load) , che sono state implementate per il progetto sviluppato partendo dalle singole tabelle della base dati dei MMG in rete.

Tali procedure saranno descritte nel paragrafo "Data Integration".

2.3 Strumenti di BI: Pentaho Data Integration

Pentaho Data Integration (PDI) nasce con il nome di *Kettle*: "Kettle Extraction, Transport, Transformation and Loading Environment") ed è un modulo open source sviluppato in Java della suite software di PENTAHO che consente di Estrarre, Trasformare e Caricare (Extraction, Transformation and Loading - ETL) i dati da una qualsiasi fonte. PDI e' uno strumento che, con un ambiente di sviluppo grafico, viene utilizzato anche in contesti aziendali da diversi anni con ottimi risultati, ed ha portato all'ottimizzazione di svariati moduli e componenti di ETL preesistenti, oltre allo sviluppo veloce e prestazionalmente valido di nuovi componenti ETL richiesti dai clienti di diverse imprese. Grazie a questo strumento si possono incrociare i dati da più fonti, si ha la possibilità di aggiornarli in real time ed effettuare migrazione di dati tra sistemi diversi tramite la creazione di programmi (che vengono definiti job).

Pertanto PDI si configura come un potente strumento basato sui metadata che offre i seguenti vantaggi:

1. Supporto alla connessione verso la maggior parte dei database moderni;
2. Possibilità di parallelizzare e gestire il multithreading nativo;
3. Sviluppo visuale/grafico “drag and drop”;
4. Consente di trasferire i dati tra Database e Flat Files (file non Strutturati);
5. Possibilità di sviluppo di plugin;
6. Supporto al cloud computing;
7. Possibilità di salvare i lavori sia in Repository (Deposito dove vengono gestiti i Metadati attraverso tabelle relazionali , ottimo per grandi moli di dati) che Files;
8. Gestione del clustering e molte altre funzionalità avanzate.

Il software è giunto alla versione 7.1 che è possibile ottenere gratuitamente dal sito ufficiale (www.pentaho.com). Il programma si presenta con i comandi in lingua italiana e con una schermata di benvenuto che ci permette di ottenere più informazioni riguardo al software.

Pentaho Data Integration è formato da quattro componenti:

1. **Spoon** - per disegno grafico dei passi dell'ETL
2. **Pan** - per esecuzione da linea di comando delle trasformazioni
3. **Kitchen** - per esecuzione dei job
4. **Carte** - console per l'esecuzione remota

Per la realizzazione della piattaforma oggetto della presente tesi è stata utilizzata l'interfaccia grafica, eseguibile in Windows con "spoon.bat" e su Linux o Mac - OS con "spoon.sh".

Pentaho Data Integration include il supporto a tutte le funzionalità richieste ai workflow necessari alla piattaforma che è stata sviluppata come oggetto della tesi, in particolare:

- Lettura e scrittura di dati su tabelle del database;
- Supporto ai lookup (utilizzati ad esempio per i controlli anagrafici);
- Lettura da file XML (mediante libreria integrata XPath);
- Validazione XSD "semplice", con possibilità di customizzazione mediante;
- Step che permettono di iniettare codice Java;
- Supporto alle RegEx (regular expressions), utilizzate per il pattern matching;
- Sulle stringhe di errore del validatore (riconoscimento e categorizzazione degli errori);
- Supporto per le sequences su database (utilizzate per la generazione degli id delle righe);
- Supporto alla gestione di files compressi;
- Supporto per la scrittura di files CSV;
- Possibilità di iniettare codice shell (utilizzato per il controllo di esecuzione concorrente);
- Possibilità di eseguire codice Java;
- Possibilità di eseguire script R;
- Accesso a file system (copia files, spostamento, cancellazione, etc);
- Filtraggio righe e switching sulla base di condizioni;
- Manipolazione valori (ad esempio troncamento stringhe sulla base delle dimensioni dei campi specificate su db);
- Grouping, sorting e controllo univocità valori in memoria.

In PDI sono presenti principalmente due tipologie diverse di macro elementi:

1. ***Jobs***: che permettono di stabilire il workflow a livello di macro elementi, e permettono di decidere le azioni da intraprendere a seconda dell'esito dei vari elementi di cui sono composti; essi permettono l'impiego di cicli e condizioni, e sono composti da elementi atomici, inoltre qui viene gestito il parallelismo fra i macro processi (se si decide di adottarlo), seppur ogni ramo di elaborazione venga poi eseguito serialmente. Ogni job ha sempre un inizio ed una fine (quest'ultima può essere, come nell'esempio sopra riportato, sia un successo che porta ad un esito del job complessivamente positivo, sia un fallimento che porta ad un esito complessivamente negativo), e può seguire da 1 ad N rami di elaborazione.

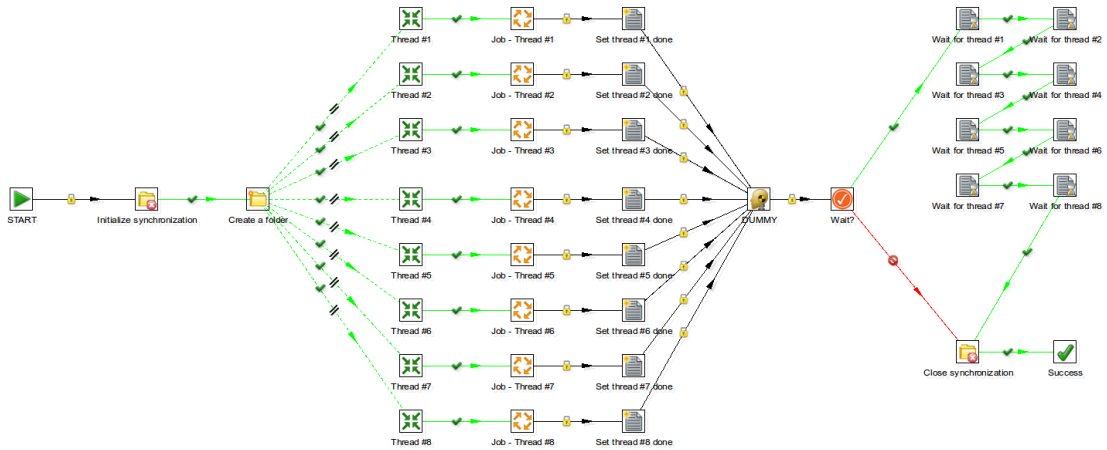


Figura 10 - Esempio di Jobs

2. ***Trasformate (Transformations)***: le trasformate servono a disegnare graficamente le micro elaborazioni che devono essere effettuate sui dati, quindi si occupano principalmente di gestire il flusso dati a livello di righe movimentate, da una o più sorgenti, verso una o più destinazioni; le trasformate sono parallele di natura, ed ogni suo elemento può esser visto come un thread che “vive di vita propria” (al contrario degli elementi che compongono i jobs), la cui esecuzione viene lanciata con l'avvio della trasformata, e termina solamente quando finisce di ricevere righe da elaborare da parte degli step precedenti.

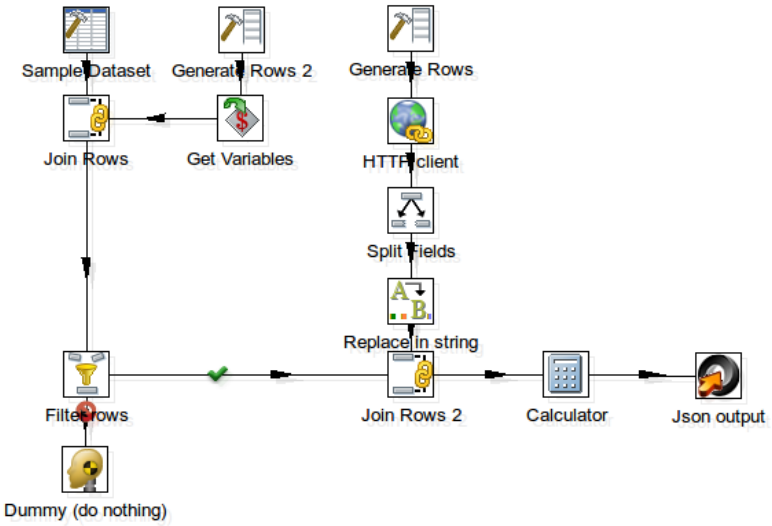


Figura 11 - Transformate example

Entrambi gli elementi introdotti sono composti da:

- Steps: sono gli elementi grafici rappresentati da un quadrato contenente un immagine rappresentativa della sua funzionalità; essi rappresentano la singola operazione che viene eseguita in un determinato punto del flusso di elaborazione, e sono configurabili mediante pannelli appositi che vengono aperti da interfaccia grafica. Nei job sono semplici operazioni "predefinite" da eseguire atomicamente (oppure trasformate sviluppate dall'utente che vengono devono essere richiamate durante l'esecuzione), mentre nelle trasformate rappresentano un operazione da eseguire per ogni singola riga di dati, e si suddividono principalmente in tre tipologie:
    - Steps che generano dati (ad esempio step di lettura da database);
    - Steps che manipolano dati (ad esempio step di elaborazione su stringhe);
    - Steps che consumano dati (ad esempio step di scrittura su file csv).
- Hops: gli hops gli elementi di connessione ("le frecce") tra i vari steps di cui sono composti i jobs oppure le trasformate; a seconda del contesto (job o trasformate), si suddividono in diverse categorie:
    - Job / hops incondizionali (vengono sempre seguiti dal flusso);

- Job / hops condizionali, a condizione vera (vengono seguiti solamente se lo step sorgente da cui parte l'hop ha esito positivo);
- Job / hops condizionali, a condizione falsa (vengono eseguiti solamente se lo step sorgente da cui parte l'hop ha esito negativo);
- Trasformate / hops normali (seguiti solitamente durante l'elaborazione dei dati);
- Trasformata / hops di error handling (seguiti solamente nel caso di errori nello step che li ha generati, per i soli step che lo supportano; permettono di "redirigere" le singole righe che hanno creato problemi per una gestione custom, ad esempio inserimento dati errati in una tabella apposita).

Pentaho Schema Workbench

Pentaho fornisce come strumento integrato nella sua suite Mondrian, un motore OLAP scritto in Java per la creazione di server Relational - OLAP usando database relazionali verso cui vengono inviate query SQL per generare un cubo multidimensionale, ed eseguire query scritte in linguaggio MDX. Essendo in pratica un server OLAP virtuale su una base di dati relazionale le prestazioni di Mondrian sono fortemente influenzate dalla capacità di aggregazione, caching e capienza del RDBMS sottostante.

Influiscono positivamente sulle prestazioni del tool l'ottimizzazione dei parametri della memoria e un buon design dello schema relazionale, che può essere a stella o a fiocco di neve. A livello della libreria Java è previsto un efficace meccanismo di caching, con la possibilità di invalidare regioni del cubo multidimensionale che andranno rigenerate dal RDBMS alla successiva query. Questa soluzione, rendendo difficile l'implementazione tecnica, in molte applicazioni Mondrian non viene utilizzata.

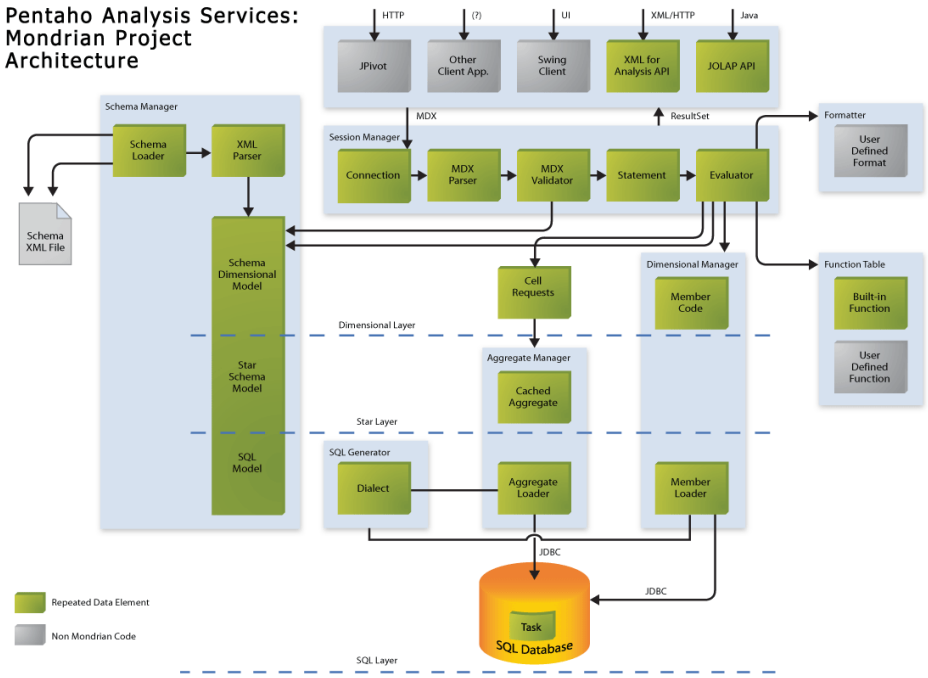


Figura 12 - Mondrian Architecture

Il modo in cui Mondrian si interfaccia con l'RDBMS per caricare il cubo multidimensionale viene dapprima definito in un file XML, contenente tutte le configurazioni dello schema dei cubi, che viene realizzato utilizzando lo strumento di sviluppo **Mondrian Schema Workbench**.

Mondrian Schema Workbench rappresenta un tool della suite software Pentaho che offre un'interfaccia grafica user friendly per la creazione di schemi che definiscono cubi OLAP. Come si evince dalla figura sottostante l'interfaccia consente in maniera rapida di costruire cubi definendo, attraverso i pulsanti della barra strumenti, la possibilità di definire:

- Una connessione (testabile) al database di appoggio;
- Una tabella dei fatti;
- Le dimensioni;
- Le misure semplici o composte utilizzando degli script.

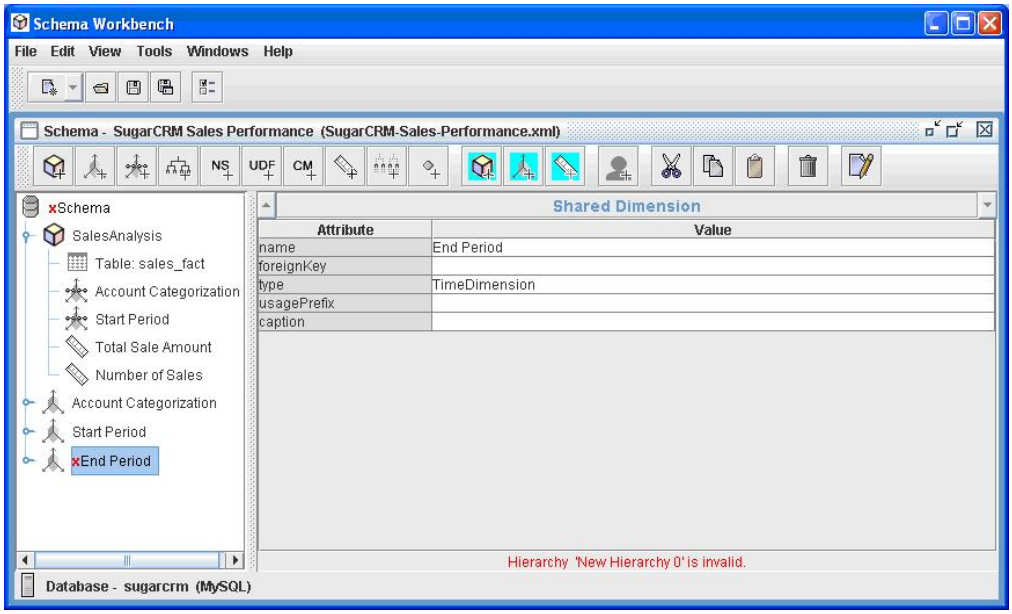


Figura 13 - Mondrian schema Workbench

Inoltre il tool consente di pubblicare lo schema realizzato direttamente nella piattaforma Pentaho Business Analytics, così da poter essere utilizzato come data source per le query MDX e realizzazione di dashboard.

Pentaho BI

Il cuore dell'architettura della suite software Pentaho (in Figura 14) è costituita dalle componenti di Business Intelligence (BI):

- BI Server;
- User console.

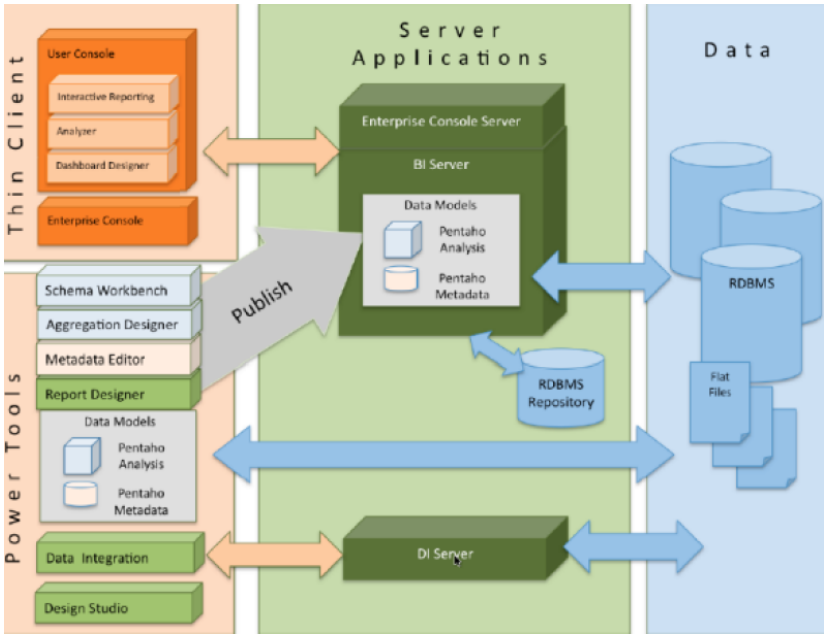


Figura 14 - Architetture Suite Pentaho

Il server di BI costituisce il vero motore delle suite, processa il contenuto dei report, delle analisi e delle dashboard create dagli utenti. Inoltre, concentra al suo interno le funzionalità di pianificazione dei lavori (ricompilazione delle soluzioni) e di validazione.

Si tratta di un'applicazione web interamente scritta nel linguaggio JAVA, basata sul framework JEE (Java Platform, Enterprise Edition). Questa caratteristica rende Pentaho un software scalabile, integrabile e portabile in ogni ambiente. L'applicazione necessita di uno dei seguenti application server per poter funzionare:

- Tomcat 6.0.39;
- JBoss 7.2.x(EAP 6.1.x).

Al suo interno, ospita il repository di Business Analytics, utilizzato per condividere in maniera sicura le varie soluzioni create. La configurazione di base di Pentaho, imposta un repository interno usufruendo di un database HyperSQL. È possibile però configurare l'utilizzo di un database esterno di proprio gradimento, utilizzando ad esempio

- MySQL;
- Oracle;
- PostgreSQL .

Il server viene gestito attraverso la propria interfaccia web, denominata *Console Utente*. Attraverso quest'ultima è possibile esplorare le soluzioni contenute nel repository, pubblicare nuovi contenuti e pianificare i lavori nel tempo. L'accesso è gestito da un modulo di sicurezza ed è possibile configurarlo per demandare la validazione delle credenziali ad un sistema esterno, utilizzando il protocollo LDAP, e definire utenti con diversi livelli di accessibilità alle informazioni.

Una volta eseguito l'accesso con il proprio utente (si veda Figura 15) l'interfaccia da la possibilità di:

- Esplorare i file presenti nel repository
- Creare nuovi elementi quali dashboard
- Gestire i Data Source (connessioni a Data warehouse, cubi multidimensionali, Kettle transformation etc.)

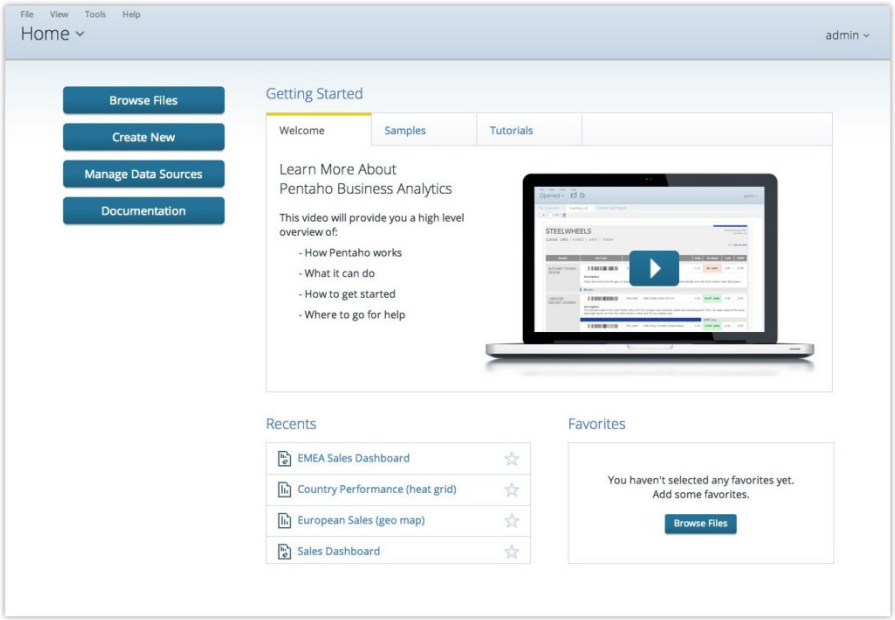


Figura 15 - Interfaccia Utente

La Console Utente di Pentaho inoltre mette a disposizione un servizio di gestione dei plug-in, chiamato Marketplace. Dalla pagina del Market, è possibile monitorare le versioni dei moduli attualmente installati nel server ed installarne di nuovi. Un esempio di plug in di notevole importanza, che può essere installato sul server è *Saiku*, uno strumento grafico per

creare visualizzazioni personalizzate, a partire da cubi multidimensionali realizzati con Mondrian Schema Workbench settati come data source.

2.4 Data integration

In questo paragrafo verranno analizzate le principali trasformazioni realizzate, utilizzando il componente *Pentaho Data Integration (PDI o Kettle)* della suite Pentaho, il quale è responsabile della gestione ed implementazione dei processi di ETL (Extract, Transform and Load), per l'implementazione del componente *Data Integration* riportato in Figura 16 - **Fase B** - dell'architettura complessiva.

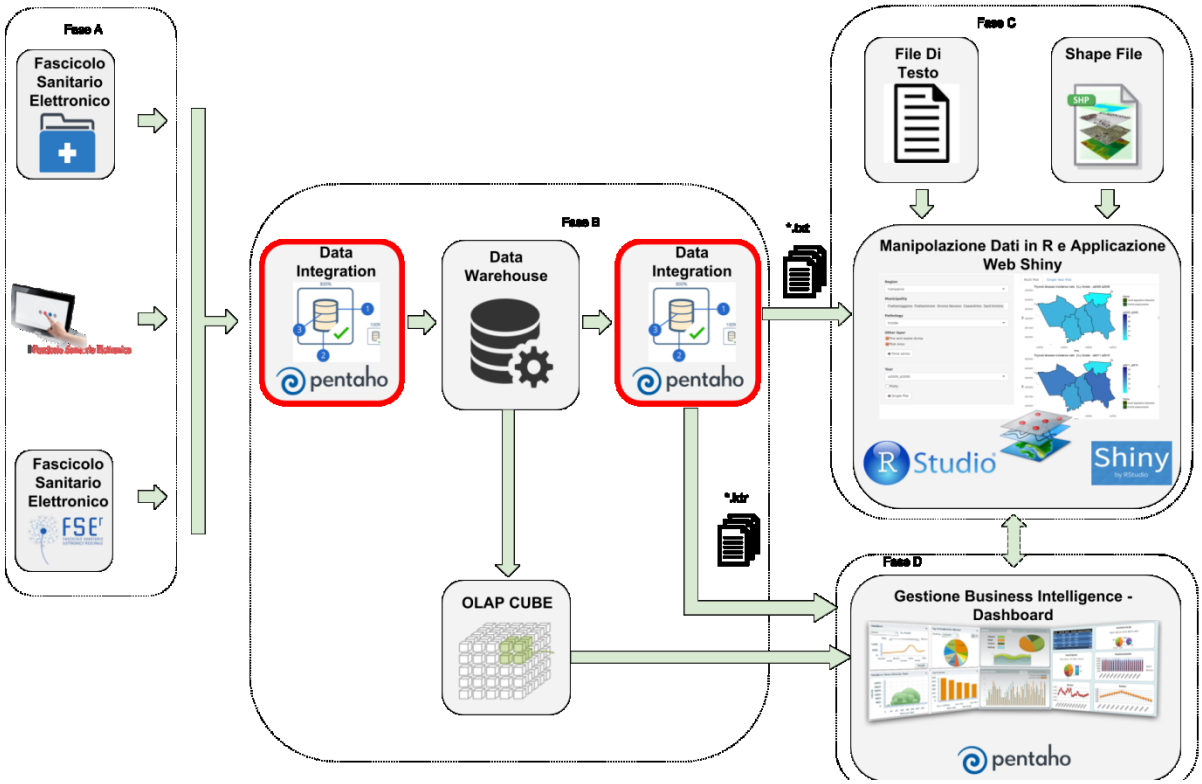


Figura 16 - Architettura Complessiva

In particolare vengono qui descritte le due trasformazioni principali realizzate per la manipolazione dei dati.

Trasformazione ETL dati Data warehouse

Sono state realizzate diverse trasformazioni per l'estrazione delle informazioni dai Database MMG, la loro manipolazione e il caricamento dei dati risultanti nel Data warehouse.

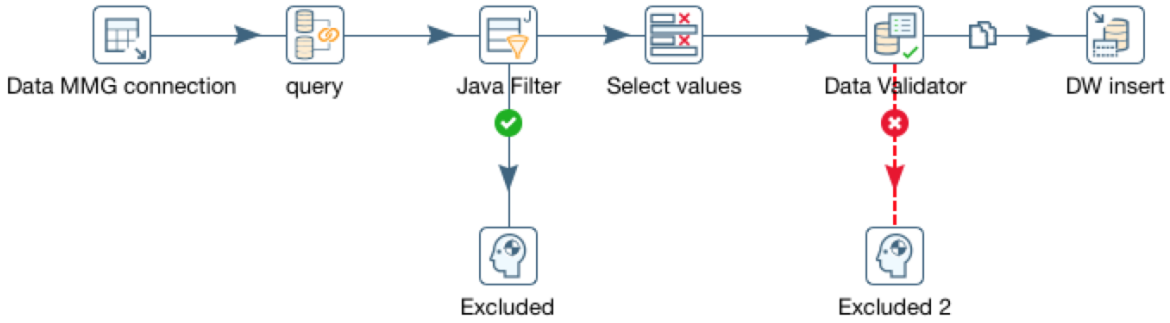


Figura 17 - Schema ETL

Tali trasformazioni seguono lo schema in Figura 17 che prevede i seguenti step PDI:

- **Get Variables**: Intercetta i valori dei parametri in ingresso (comune di residenza, prescrizione, patologia etc) che vengono resi disponibili durante la trasformazione come variabili globali;
- **Data MMG Connection**: blocco che effettua la connessione a determinate tabelle dei Database MMG;
- **Query**: blocchi che effettuano delle query opportunamente costruite sulla base dei parametri di ingresso per estrarre le informazioni dai database MMG (come riportato in Figura 3 dove viene mostrata una query per l'estrazione dei pazienti per patologia e paese di residenza);
- **Merge**: effettua il Merge delle informazioni estratte dalle query;
- **Java Filter**: Blocco progettato per operare mediante l'impiego di espressioni regolari su collezioni di dati (ossia dati trattati con il java collection framework) al fine di manipolare le informazioni estratte dalle query e renderle compatibili col formato richiesto dai successivi blocchi della trasformazione;
- **Excluded**: In questo blocco vengono convogliati i dati che non superano la selezione

- **Select Values**: In questo blocco vengono convogliati i dati che superano la selezione rispetto ai parametri in ingresso e possono essere forniti in ingresso al blocco successivo;
- **Data Validator**: blocco che consente di validare i dati estratti;
- **DW insert**: Questo blocco consente di stabilire una connessione al Data Warehouse e effettuare delle insert dei risultati all'interno delle tabelle del DW impostate.

Query per totalizzare i pazienti malati per patologia (per singolo ISTAT di Assistenza)

```
select codice_icdix, descrizione_icdix, sum(totale_2008) as tot_2008, sum(totale_2009) as tot_2009, sum(totale_2010) as tot_2010, sum(totale_2011) as tot_2011,
sum(totale_2012) as tot_2012, sum(totale_2013) as tot_2013, sum(totale_2014) as tot_2014, sum(totale_2015) as tot_2015, sum(totale_2016) as tot_2016
from (select codice_icdix, descrizione_icdix, count(*) as totale_2008, 0 as totale_2009, 0 as totale_2010, 0 as totale_2011,
0 as totale_2012, 0 as totale_2013, 0 as totale_2014, 0 as totale_2015, 0 as totale_2016 from an2008_061049c group by codice_icdix, descrizione_icdix
union all
select codice_icdix, descrizione_icdix, 0, count(*) as totale_2009, 0, 0, 0, 0, 0, 0 from an2009_061049c group by codice_icdix, descrizione_icdix
union all
select codice_icdix, descrizione_icdix, 0, 0, count(*) as totale_2010, 0, 0, 0, 0, 0, 0 from an2010_061049c group by codice_icdix, descrizione_icdix
union all
select codice_icdix, descrizione_icdix, 0, 0, 0, count(*) as totale_2011, 0, 0, 0, 0, 0 from an2011_061049c group by codice_icdix, descrizione_icdix
union all
select codice_icdix, descrizione_icdix, 0, 0, 0, 0, count(*) as totale_2012, 0, 0, 0, 0 from an2012_061049c group by codice_icdix, descrizione_icdix
union all
select codice_icdix, descrizione_icdix, 0, 0, 0, 0, 0, count(*) as totale_2013, 0, 0, 0 from an2013_061049c group by codice_icdix, descrizione_icdix
union all
select codice_icdix, descrizione_icdix, 0, 0, 0, 0, 0, 0, count(*) as totale_2014, 0, 0 from an2014_061049c group by codice_icdix, descrizione_icdix
union all
select codice_icdix, descrizione_icdix, 0, 0, 0, 0, 0, 0, 0, count(*) as totale_2015, 0 from an2015_061049c group by codice_icdix, descrizione_icdix
union all
select codice_icdix, descrizione_icdix, 0, 0, 0, 0, 0, 0, 0, 0, count(*) as totale_2016 from an2016_061049c group by codice_icdix, descrizione_icdix
) tabella
group by codice_icdix, descrizione_icdix;
```

Query per estrarre i dati di un singolo anno della query dei totali (per singolo ISTAT di Assistenza)

```
select codice_fiscale_paziente, codice_icdix, descrizione_icdix from prdi061049c where data_prescrizione between '2016-01-01' and '2016-12-31'
group by codice_fiscale_paziente, codice_icdix, descrizione_icdix
having count(*) > 1;
```

Figura 18 - Query ETL

Trasformazione per l'estrazione dati

La trasformazione principale è responsabile di tutte le operazioni che consentono di accedere al data warehouse per effettuare l'estrazione, manipolazione e l'esportazione dei dati al fine di renderli disponibili alla piattaforma R-based ed alla piattaforma di Business Intelligence.

Lo schema PDI di tale trasformazione è rappresentato in Figura 19.

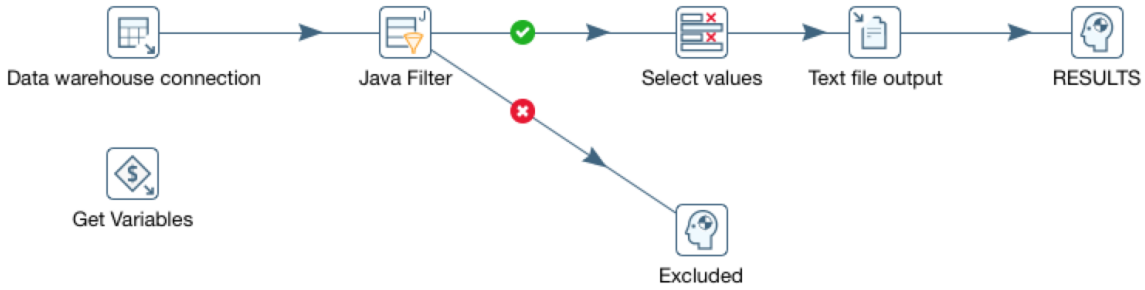


Figura 19 - Schema Data_extraction

In tale tale trasformazione vengono impiegati i seguenti step PDI:

- **Get Variables**: Intercetta i valori dei parametri in ingresso (comune, patologia, anno, etc) che vengono resi disponibili durante la trasformazione come variabili globali;
- **Data warehouse Connection**: Effettua la connessione a determinate tabelle del Data warehouse;
- **Java Filter**: blocco progettato per operare mediante l'impiego di espressioni regolari su collezioni di dati (ossia dati trattati con il java collection framework) al fine di manipolare le informazioni estratte dal data warehouse per renderle compatibili con il formato richiesto dai successivi blocchi della trasformazione e di conseguenza con le piattaforme che li utilizzeranno;
- **Excluded**: In questo blocco vengono convogliati i dati che non superano la selezione
- **Select Values**: In questo blocco vengono convogliati i dati che superano la selezione del blocco precedente e vengono selezionati solo i valori definiti e che possono anche essere settati come parametro di ingresso;
- **Text file output**: Questo blocco salva i risultati dell'elaborazione in un file di testo (si veda Figura 20). In questo file sono presenti le seguenti informazioni:
    - *id_wirgilio - id del paziente*
    - *codice_icdix - codice della patologia*
    - *descrizione_icdix - descrizione patologia*
    - *anno - anno di rilevamento*
    - *comune - comune di residenza*
- **Result**: Blocco che consente di visualizzare i risultati anche nella piattaforma di Business Intelligence, fungendo da data source.

id_wirgilio; codice_icdix; descrizione_icdix; anno; comune

6318; 724.2; Lombalgia; 2016; Frattaminore

128923; 786.2; Tosse; 2016; Frattaminore

130833; 285.9; Anemia. non specificata; 2016; Frattaminore

194076; 272.4; Altre e non specificate iperlipidemie; 2016; Frattaminore

10132; 733; Osteoporosi; 2016; Frattaminore

10927; 250;Diabete mellito. tipo II (non insulinodipendente) (diabete dell'adulto) o non specificato. non definito se scompensato. senza menzione di complicanze; 2016; Frattaminore

10020; 696.1; Psoriasi; 2016; Frattaminore

57006; 556.9; Colite ulcerosa. non specificata; 2016; Frattaminore

119665; 245.2; Tiroidite linfocitaria cronica; 2016; Frattaminore

193071; 456.4; Varicocele; 2016; Frattaminore

90301; V22.0; Controllo di prima gravidanza normale; 2016; Frattaminore

Figura 30 - Estratto file di output per piattaforma R

Trasformazione calcolo Incidenza

La trasformazione per il calcolo dell'incidenza è responsabile dell'estrazione ed elaborazione dei dati presenti in un file di testo (es. out trasformazione precedente) per il calcolo dell'incidenza dell'incidenza fornendo in ingresso paese e patologia.

I valori calcolati sono poi resi disponibili alla piattaforma di Business Intelligence usando il file .ktr della trasformazione.

Lo schema PDI di tale trasformazione è rappresentato in Figura 21.

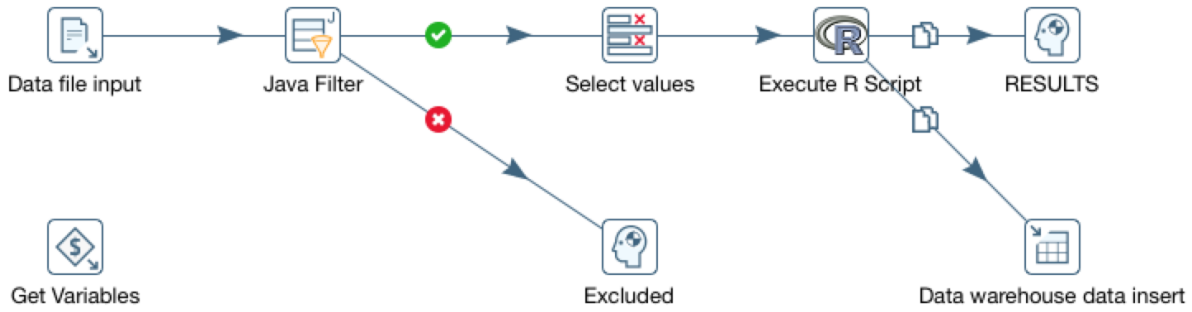


Figura 21 - Schema Incidence_evaluator

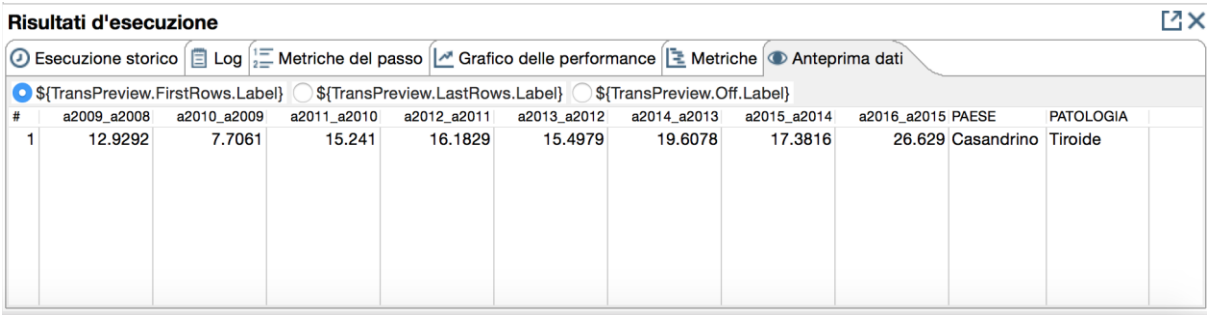
In tale tale trasformazione vengono impiegati i seguenti step PDI:

- **Get Variables**: Intercetta i valori dei parametri in ingresso (comune, patologia, anno, etc) che vengono resi disponibili durante la trasformazione come variabili globali;
- **Data file input**: Questo blocco acquisisce i dati da un file di testo opportunamente formattato (es. l'output della trasformazione precedente);
- **Java Filter**: Blocco progettato per operare mediante l'impiego di espressioni regolari su collezioni di dati (ossia dati trattati con il java collection framework) al fine di manipolare i le informazioni estratte dal file di testo per renderle compatibili con il formato richiesto dai successivi blocchi della trasformazione e al blocco che le utilizzerà nell'elaborazione dello script R;
- **Excluded**: In questo blocco vengono convogliati i dati che non superano la selezione
- **Select Values**: In questo blocco vengono convogliati i dati che superano la selezione del blocco precedente e vengono selezionati solo i valori definiti e che possono anche essere settati come parametro di ingresso;
- **Execute R Script**: Questo estrae dal flusso dati in ingresso i parametri di input ed esegue uno script R su tali dati per effettuare il calcolo dell'incidenza in funzione dei parametri settati in ingresso alla trasformazione e acquisiti dal blocco "Get Variables";

- **Result**: Blocco che consente di visualizzare i risultati anche nella piattaforma di Business Intelligence, fungendo da data source;
- **Data warehouse data insert**: Questo blocco consente di stabilire una connessione al Data Warehouse e effettuare delle insert dei risultati all'interno delle tabelle del DW impostate.

La Figura 22 mostra un particolare dell'interfaccia di Pentaho Data Integration per la trasformazione descritta. La sezione Anteprima dati mostra l'anteprima dei risultati dell'elaborazione per il caso specifico:

- Paese: Casandrino
- Patologia: Tiroide



#	a2009_a2008	a2010_a2009	a2011_a2010	a2012_a2011	a2013_a2012	a2014_a2013	a2015_a2014	a2016_a2015	PAESE	PATOLOGIA
1	12.9292	7.7061	15.241	16.1829	15.4979	19.6078	17.3816	26.629	Casandrino	Tiroide

Figura 22 - Scheda Risultati d'esecuzione - Anteprima dati output

2.5 Origine dei dati

I dati oggetto dello studio afferiscono alla serie storica 2000-2016 e fanno riferimento a dati sanitari di pazienti (over 14 anni) presenti nelle cartelle cliniche di circa 600 Medici di Medicina Generale (MMG). Questi MMG che svolgono la loro attività principalmente alle province di Napoli, Caserta e Salerno, utilizzano lo stesso sistema di Cartella Clinica Elettronica (CCE) e sono coinvolti nella medicina di rete. Le CCE oltre a contenere dati di carattere sanitario, includono una serie di informazioni accessorie, quali per esempio quelle anagrafiche (e.g., domicilio, residenza, data di nascita, etc.) utili per l'analisi dati e l'estrazione di nuove informazioni tramite l'impiego di opportune relazioni, regole, assiomi e vincoli specifici. In particolare in questa tesi, l'attenzione è stata rivolta alle informazioni anagrafiche riguardanti la residenza e/o il domicilio dell'assistito, e alle informazioni sanitarie, ovvero il codice di diagnosi ICD-IX e la relativa descrizione.

2.5.1 Il codice ICD-IX

Il codice ICD-IX e' un sistema internazionale di classificazione che organizza le malattie ed i traumatismi in gruppi sulla base di criteri definiti.

La classificazione ICD-IX descrive in codici numerici o alfa-numerici i termini medici in cui sono espressi le diagnosi di malattia o di traumatismo, gli altri problemi di salute, le cause di traumatismo e le procedure diagnostiche e terapeutiche. I caratteri fondamentali della ICD-IX sono i seguenti:

- l'esaustività: tutte le entità trovano una loro collocazione, più o meno specifica, entro i raggruppamenti finali della classificazione;
- la mutua esclusività: ciascuna entità è classificabile soltanto in uno dei raggruppamenti finali della classificazione;
- il numero di raggruppamenti: circa 16.000 codici consentono la classificazione delle diagnosi, dei problemi di salute e delle principali procedure diagnostiche e terapeutiche;
- la specificità dei raggruppamenti in ragione della rilevanza delle entità nosologiche dal punto di vista della sanità pubblica: le entità nosologiche di particolare importanza per la sanità pubblica o che si verificano con maggiore frequenza sono individuate da una specifica categoria; tutte le altre entità nosologiche sono raggruppate in categorie non strettamente specifiche, che comprendono condizioni differenti, benchè tra loro correlate.

Il sistema ICD-IX contiene due classificazioni, una per le malattie ed una per le procedure, ciascuna delle quali è costituita da un indice alfabetico e da un elenco sistematico; si configurano così le seguenti quattro parti:

- indice alfabetico delle malattie e dei traumatismi;
- elenco sistematico delle malattie e dei traumatismi;
- indice alfabetico degli interventi chirurgici e delle procedure diagnostiche e terapeutiche;
- elenco sistematico degli interventi chirurgici e delle procedure diagnostiche e terapeutiche.

Inoltre sono presenti due classificazioni supplementari:

- la classificazione supplementare dei fattori che influenzano lo stato di salute ed il ricorso alle strutture sanitarie (codici V);
- la classificazione supplementare delle cause esterne di traumatismo e avvelenamento (codici E).

L'indice alfabetico e l'elenco sistematico delle due classificazioni sono concepiti per integrarsi a vicenda: i singoli termini clinici, di patologia o procedura, si ricercano negli indici alfabetici e la correttezza dei codici attribuiti viene quindi verificata con tutte le indicazioni accessorie riportate nei relativi elenchi sistematici.

2.5.2 Il codice ICPC

Al fine di operare una classificazione delle patologie in base alle "comuni" caratterizzazioni, e al fine di rendere il dato maggiormente fruibile, si è pensato di raggruppare le patologie utilizzando una differente codifica, facendo ricorso alla *International Classification of Primary Care* (codifica ICPC), adatta alla codifica delle cure primarie da parte del medico di base (Figura 16 - Fase C).

Infatti, il codice ICD-IX, pur rappresentando uno standard internazionale di classificazione, è poco funzionale a studi descrittivi e/o epidemiologici in quanto in quanto le entità diagnostiche sono classificate con la stessa logica in più di un capitolo, (per esempio, l'influenza viene considerata sia nel capitolo delle infezioni che in quello dell'apparato respiratorio che in entrambi).

Nel 1987 venne pubblicata la classificazione ICPC a cura del comitato internazionale chiamato *World International Classification Committee* (**WICC**) che è a sua volta espressione della *World Organization of Family Doctor*s (**WONCA**). E' ovvio che l'uso internazionale di questa classificazione permette un interscambio di informazioni riguardanti la medicina di primo livello, non solo fra i vari Medici, ma fra i vari Sistemi Sanitari Regionali, Nazionali e sovranazionali che adottano questa metodologia di registrazione comune. Per la prima volta la classificazione ICPC e i MMG furono in grado di identificare, attraverso una sola classificazione, tre importanti elementi dell'incontro sanitario: i motivi dell' "incontro" (MDI), le diagnosi o i problemi, le procedure di cura. Il collegamento dei vari elementi ha permesso una categorizzazione completa dell'incontro (dall'inizio alla fine).

Le due classificazioni non sono in competizione, ma semplicemente si adattano a contesti e scopi diversi; inoltre, già la prima edizione dell'ICPC conteneva un elenco di codici di conversione all'ICD-IX. Il punto di forza della ICPC è la registrazione del lavoro del medico in ambulatorio diviso per **"Episodi di Cura".** Vale a dire che questa classificazione offre una serie molto semplice di codici che permette il raggruppamento di tutti gli atti e le procedure che sono effettuate dal medico e dai suoi intermediari per rispondere al meglio ad ogni singola richiesta o problema di salute presentati dai suoi assistiti. Dalla struttura dell'Episodio di Cura si possono estrarre i "Motivi dell'incontro" (ciò che porta il paziente dal medico, vera espressione del bisogno sanitario dell'assistito, come da lui stesso interpretato), le "Diagnosi o le Condizioni" (ciò che il medico interpreta come bisogno sanitario per quel paziente), e le "Procedure", (ovvero tutto ciò che il sanitario mette in campo per confermare o trattare quel bisogno sanitario). L'Episodio di Cura si costruisce attraverso un 'Incontro" (o gli incontri quando per definirlo ne servono più di uno) fra il Paziente ed il Medico.

2.5.3 Stima della rappresentatività dei dati

I dati a nostra disposizione rappresentavano una visione parziale del territorio in quanto, i MMG sono consorziati in diverse cooperative che usano sistemi software forniti da aziende diverse di cui non avevamo la disponibilità dei dati. Per questo motivo prima di procedere al processamento dei dati si è reso necessario stimare la percentuale di copertura dei comuni al fine di mostrare dei dati che avessero un minimo di rappresentatività della situazione reale.

A tale scopo, il numero di "assistiti certi" è stato stimato tenendo conto delle persone che presentavano almeno una prescrizione medica durante l'anno solare. Sebbene lo scopo del lavoro non è una analisi epidemiologica, risultava comunque di primaria importanza, al fine di dimostrare le capacità della piattaforma, utilizzare dei dati che avessero una copertura maggiore o uguale al 50% dei residenti. I dati riguardanti il numero di persone residenti sono stati estratti dal data warehouse dell'Istituto Nazionale di Statistica (ISTAT). I comuni che mostravano una copertura maggiore o uguale al 50% sono i seguenti (Tab. 4).

Comune	Numero di abitanti
Bellona	5045
Casal di Principe	17706
Parete	9628
Pietramelara	4031
Pontelatone	1469
San Cipriano d'Aversa	11502
San Potito Sannitico	1646
Villa Literno	9986
Casandrino	11679
Frattamaggiore	25826
Frattaminore	13514
Grumo Nevano	15388

Sant'Antimo	27878
Baronissi	14585
Castiglione del Genovesi	1157

Tabella 4. Elenco dei comuni con copertura > 50%

In totale sono state processate le CCE relative a circa 100.000 persone residenti e/o domiciliate nei comuni riportati in tabella 4.

2.5.4 Scelta delle patologie da trattare per testare il prototipo

Dopo aver definito i comuni da indagare la nostra attenzione è stata rivolta all'indagine delle malattie croniche.

Alla base delle principali patologie croniche ci sono fattori di rischio comuni e modificabili, come alimentazione poco sana, consumo di tabacco, abuso di alcol, mancanza di attività fisica (Fig. 23). Queste cause possono generare quelli che vengono definiti fattori di rischio intermedi, ovvero l'ipertensione, la glicemia elevata, l'eccesso di colesterolo e l'obesità. Ci sono poi fattori di rischio che non si possono modificare, come l'età o la predisposizione genetica (Fig. 23). Nel loro insieme questi fattori di rischio sono responsabili della maggior parte dei decessi per malattie croniche in tutto il mondo e in entrambi i sessi.

Figura 23 Cause principali delle malattie croniche.

Per testare il prototipo si è scelto di considerare le malattie croniche a carico della tiroide. A questo punto per definire un assistito come affetto da una patologia cronica tiroidea abbiamo filtrato i dati in base alle prescrizioni, ovvero, abbiamo considerato una persona affetta da una patologia tiroidea se, durante l'anno solare, fossero state effettuate almeno due prescrizioni mediche con la stessa diagnosi indicata dal codice ICD-IX (Figura 16 - Fase B).

Nel caso delle patologie croniche a carico della tiroide sono stati considerati i codici ICD-IX compresi tra 240 e 246.9. Una volta filtrati i dati questi sono stati aggregati e pre-processati per calcolare l'incidenza.

2.5.6 Incidenza: definizione e applicazione ai nostri dati

L'incidenza misura la frequenza statistica di una patologia, vale a dire quanti nuovi casi di una data malattia compaiono in un determinato lasso di tempo (quest'ultimo, ad esempio, può essere rapportato a un mese o a un anno). Essa si calcola mettendo al numeratore il numero di nuovi casi di malattia registrati durante il periodo di osservazione e al denominatore il numero di persone a rischio di ammalarsi all'inizio del periodo di osservazione.

Il nostro tempo zero a cui riferire la comparsa dei nuovi casi è stato l'anno 2008. Infatti i dati forniti dai MMG sono cumulativi, ovvero i dati di ogni anno rappresentano quelli dell'anno in esame sommati a quelli degli anni precedenti. Pertanto sottraendo i casi del 2008 a quelli del 2009 otterremo i nuovi casi riferiti esclusivamente all'anno 2009.

Dopo aver calcolato l'incidenza, queste informazioni sono state georeferenziate grazie a pacchetti e librerie *ad hoc* descritti nel paragrafo 2.6.

2.6 Progettazione Dashboard Business Analytics

In questo paragrafo verranno descritte le principali dashboard realizzate ed integrate all'interno di *Pentaho Business Analityics* per l'implementazione del componente identificato con il nome *Gestione Business Intelligence - Dashboard* riportato in Figura 24 - **Fase D** - dell'architettura complessiva.

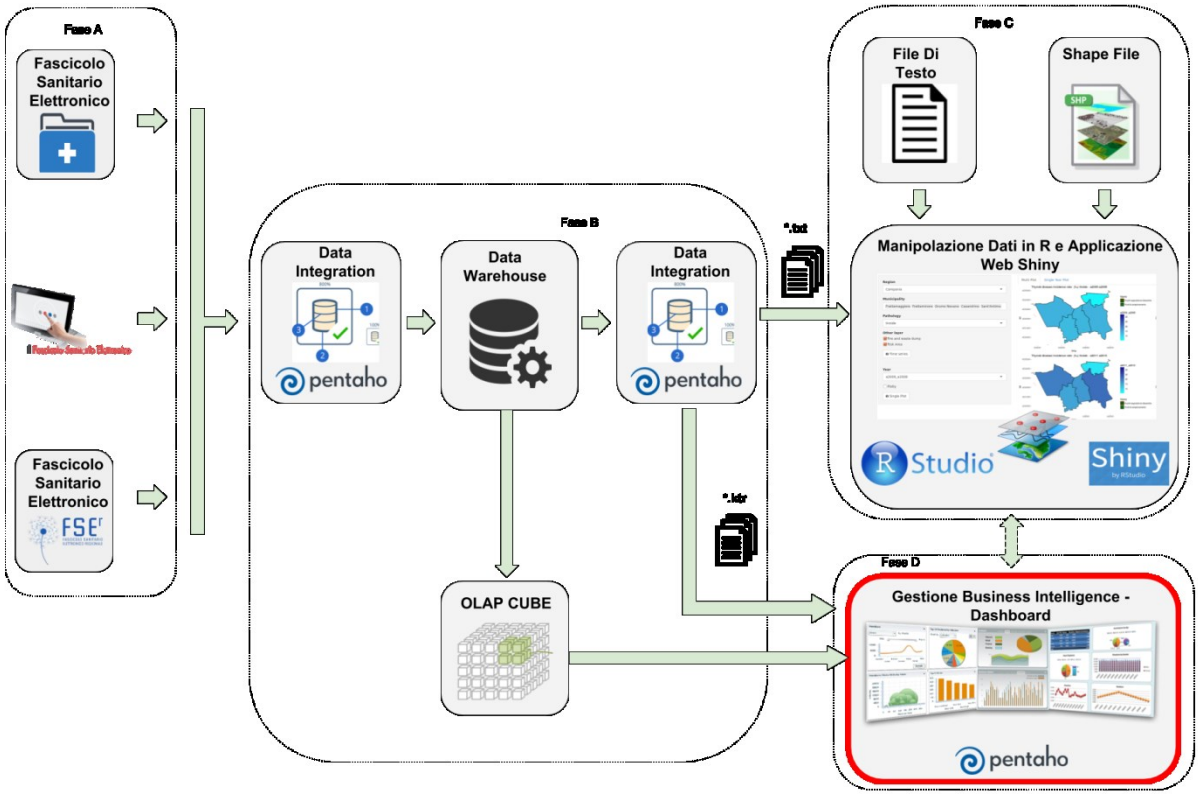


Figura 24- Architettura piattaforma

Processo di Massima per la Realizzazione di una Dashboard in Pentaho

Le dashboard che vengono messe a disposizione dell'utente della piattaforma di Business Intelligence realizzata, sono state implementate usando *Community Dashboard Editor* (CDE), un editor messo a disposizione dal BI server sella suite software di Pentaho.

L'editor CDE consente di realizzare delle dashboard interattive, definendole rispetto a tre prospettive, ad ognuna delle quali corrisponde uno specifico pannello. Nel seguito verranno descritte le funzionalità di ogni singolo pannello.

Layout Panel

In questo pannello è possibile definire il layout della dashboard definendo gli elementi che la compongono organizzandoli per righe e per colonne. Ad ogni cella saranno poi associati degli specifici componenti che vengono definiti nel pannello successivo. In questo pannello è inoltre possibile applicare stili, inserire elementi HTML, testo e immagini connotando la definizione del layout grafico. In Figura 25 è possibile vedere come per ogni elemento siano definibili un insieme di parametri che ne caratterizzano la grafica (dimensione, stile, posizione, background, color ecc.).

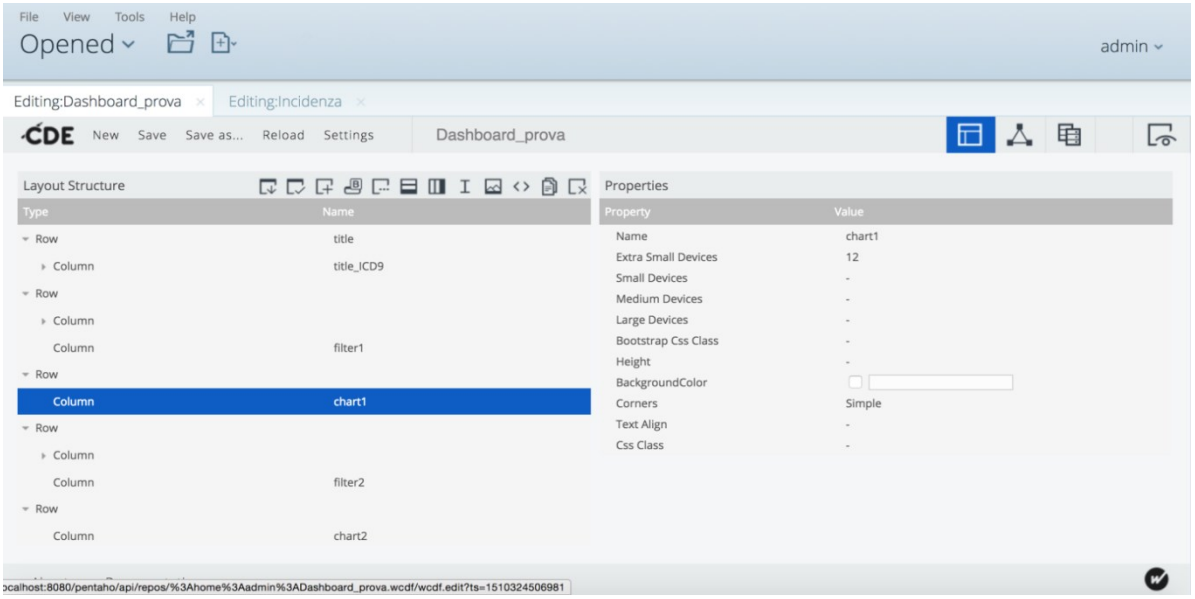


Figura 25 - Layout panel

Nel pannello Components (Fig. 26) vengono definiti tutti i componenti della dashboard, che rappresentano il cuore della struttura. In questo pannello è infatti possibile effettuare la connessione tra gli elementi del layout ed i datasource.

Sono disponibili tre tipologie di componenti:

- **Componenti visuali e dati** che includono text box, tabelle, grafici, selettori, viste OLAP, report;
- **Parametri** che rappresentano i valori che possono essere condivisa dai componenti. Questi risultano fondamentali per l'interazione tra i componenti e l'interattività della dashboard;
- **Script** che rappresentano blocchi di codice JavaScript che danno la possibilità di customizzare l'aspetto e il comportamento di altri componenti.

Alcuni componenti reagiscono ad azioni esterna (eseguite dall'utente) come ad esempio i selettori, altri reagiscono alle variazioni dei parametri definiti, come ad esempio i grafici.

Ogni componente viene configurato attraverso la definizione di tutti i suoi parametri caratteristici, come si evince dalla Figura 26.

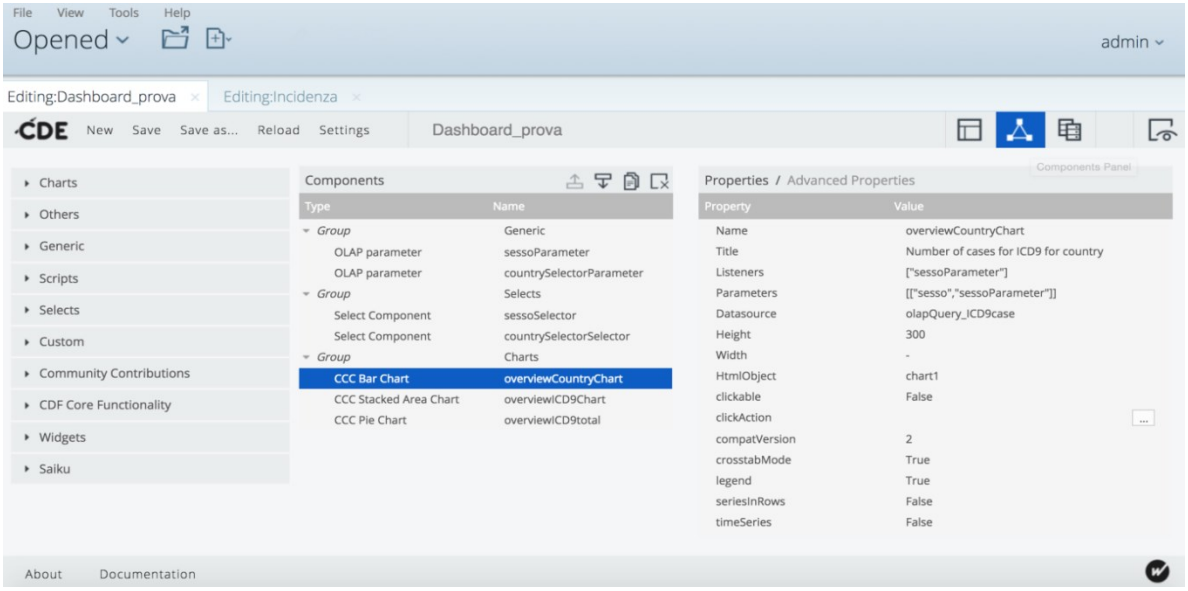


Figura 26 - Components Panel

Datasources Panel

Nel pannello Datasources è possibile definire le sorgenti di dati che saranno poi richiamate dagli elementi della dashboard.

Sono supportati diverse tipologie di datasource quali:

- PDI/Kettle transformations;
- OLAP MDX queries;
- SQL queries;
- Xaction result sets etc.

Come si evince dalla Figura 27, ogni tipologia di datasource, una volta selezionato, deve essere configurato definendone le proprietà caratteristiche, settando ad esempio le informazioni sulla connessione, le query, i parametri, la configurazione delle colonne, le colonne di output ecc.

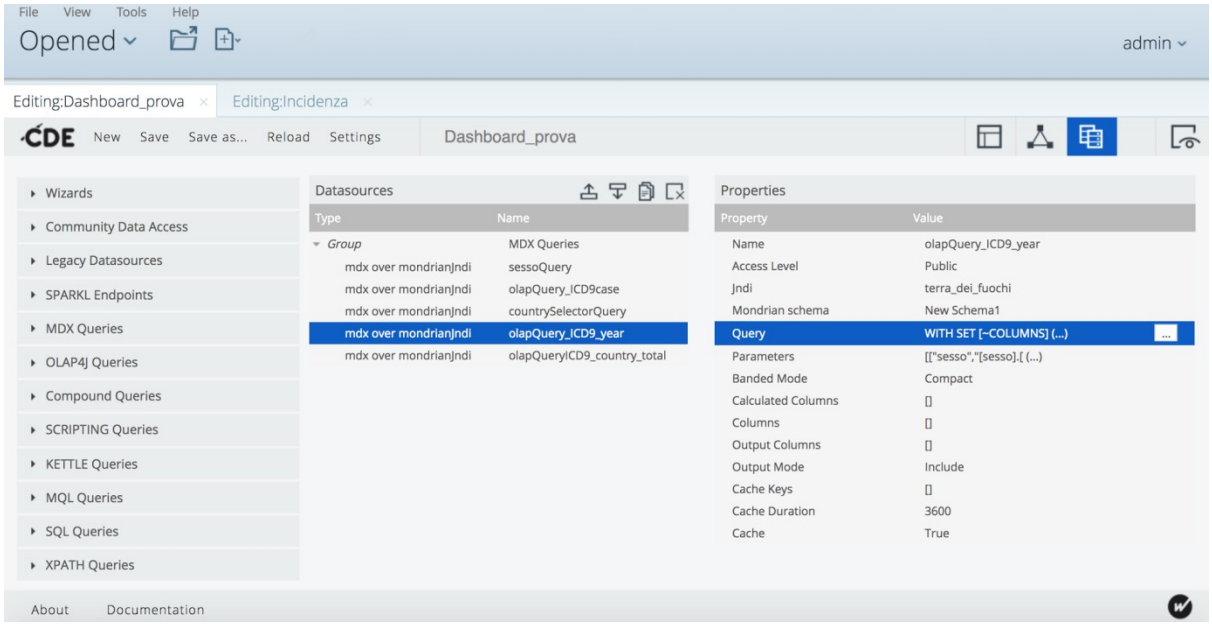


Figura 27 - DataSources Panel

In Figura 28 è rappresentata una query MDX su un cubo OLAP, utilizzata nell'implementazione delle dashboard per la determinazione del numero di casi di assistiti affetti dalle diverse patologie, divise per codice ICDIX.

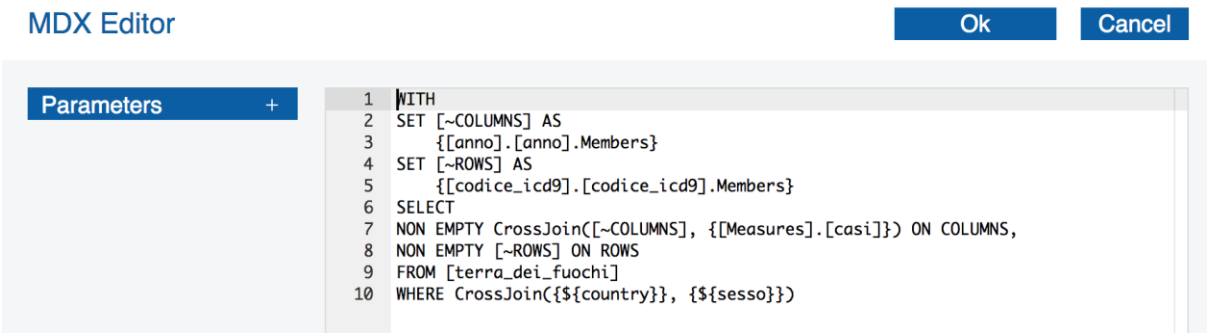


Figura 28 - Query MDX casi

Nel seguito verranno descritte due dashboard implementate usando CDE.

2.6.1 Dettagli sui componenti utilizzati per la creazione della Dashboard casi ICDIX

La Dashboard che verrà mostrata come caso studio nel presente lavoro di tesi, ha come obiettivo il monitoraggio del numero di casi di pazienti affetti dalle diverse patologie divise per codice ICDIX attraverso grafici che possono essere aggiornati dinamicamente sulla base della selezione del sesso e della città di appartenenza, grazie ad appositi menu a tendina. I risultati ottenuti e le dashboard verranno discussi in una sezione apposita del presente lavoro dedicata ai risultati. Di seguito verranno forniti alcuni dettagli sui pannelli realizzati e sulla loro configurazione.

Per tale dashboard nel pannello di layout gli elementi sono stati divisi su sei righe.

Nel pannello components, come si evince dalla Figura 29, sono stati definiti i seguenti componenti: *i)* 2 parametri OLAP (sessoParameter countrySelectorParameter) usati come parametri di scambio e come variabili nelle query MDX; *ii)* 2 Select Component connessi ai parametri definiti; *iii)* 1 Bar chart; *iv)* 1 Stacked Area Chart; *v)* 1 Pie Chart.

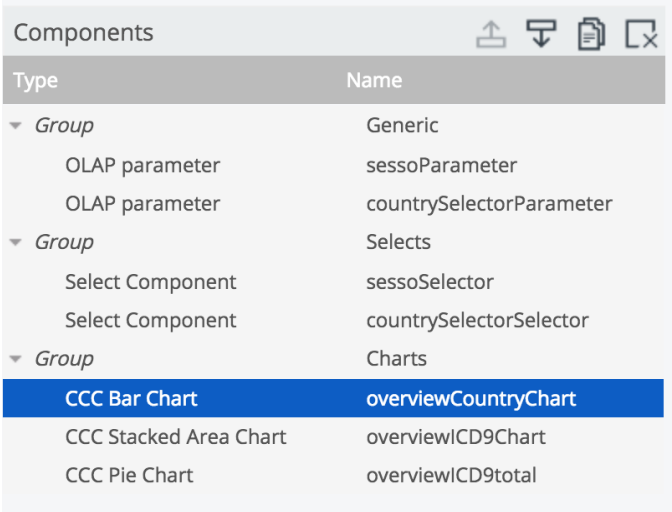


Figura 29. Componenti Dashboard casi ICDIX

Nel pannello Datasources sono state effettuate 6 query MDX definite su cubi mondrian uploadati sulla piattaforma di Pentaho.

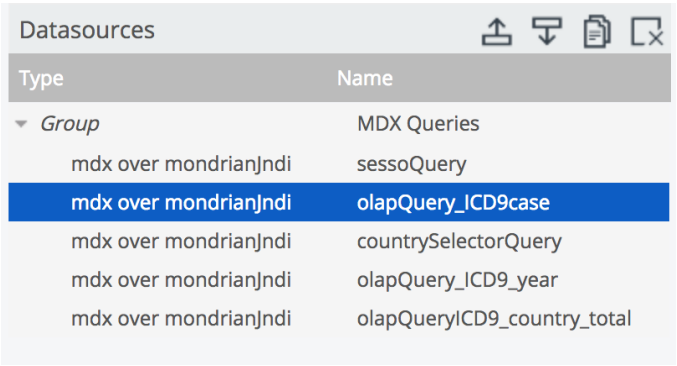


Figura 30. Datasources Dashboard casi ICDIX

Dettagli sui componenti utilizzati per la creazione della Dashboard Incidenza

La Dashboard che verrà mostrata come caso studio nel presente lavoro di tesi, ha come obiettivo il monitoraggio dei valori di incidenza attraverso grafici e tabelle che possono essere aggiornati dinamicamente sulla base della selezione della città di appartenenza e patologia. Per tale dashboard nel pannello di layout gli elementi sono stati divisi su cinque righe.

Nel pannello components, come si evince dalla Figura 30, sono stati definiti i seguenti componenti: i) 2 Simple Parameter (selected_country, selected_patologia) usati come

parametri di scambio con i diversi componenti; ii)2 Select Component(country_selector, patologia_selector) connessi ai parametri; iii)2 Table Component; iii)1 Bar Chart.

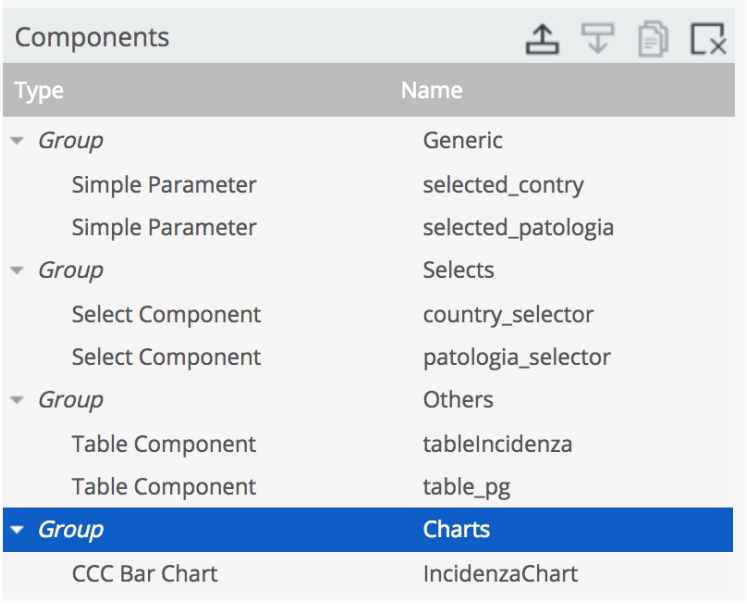


Figura 31- *Componenti Dashboard Incidenza*

Nel pannello Datasources sono state definite:

- 2 KETTLE query eseguite sulle trasformazioni (file .ktr) realizzate in *Pentaho Data Integration* e uploadate sulla piattaforma di Pentaho BI;
- 2 Query SQL eseguite utilizzando una connessione JNDI al Data Warehouse opportunamente configurata.

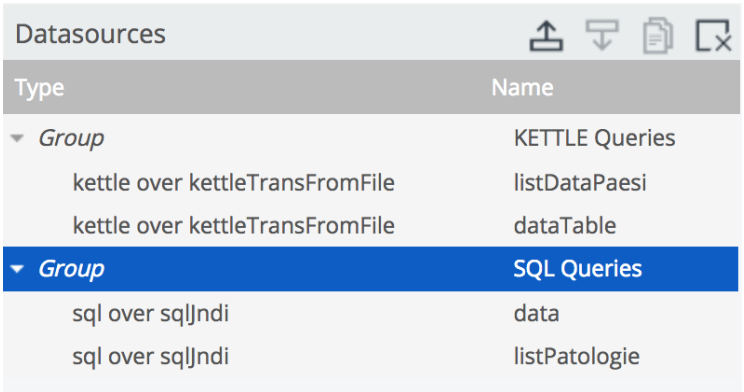


Figura 32 - *Datasources Dashboard Incidenza*

2.7 Sviluppo in R e rappresentazione georeferenziata in Shiny

2.7.1 Filtro e aggregazione dati

I dati in input per il software di georeferenziazione provengono dalle elaborazioni fatte in R data integration (Fi. 16- Fase B). Come descritto nei paragrafi precedenti i dati in output dalla fase B, che coincidono con quelli di input del software di georeferenziazione (Fig. 16 - Fase C) sono dei file di testo strutturati nel seguente modo:

id_wirgilio	codice_icdix	descrizione_icdix
753822	272.1	Ipergliceridemia pura
506636	785.0	Tachicardia non specificata
734312	300.0	Stati di ansia
736488	528.0	Stomatite
745394	280.9	Anemia da carenza di ferro, non specificata
754559	281.9	Anemia da carenza non specificata
505814	346.0	Emicrania classica
745785	427.3	Fibrillazione e flutter atriali
734197	250	Diabete mellito
505118	709.9	Alterazioni non specificate della cute e del tessuto sottocutaneo
753697	462	Faringite acuta
754506	466	Bronchite e bronchiolite acuta
734346	710.0	Lupus eritematoso sistemico
733150	462	Faringite acuta
742057	401	Ipertensione essenziale
752695	434.0	Trombosi cerebrale
506837	412	Infarto miocardico pregresso
741336	462	Faringite acuta
742896	530.1	Esofagite
754214	535.0	Gastrite acuta
734531	296.2	Depressione maggiore, episodio singolo
741865	240.9	Gozzo non specificato
753094	117.9	Altre e non specificate micosi
505377	799.9	Altre cause sconosciute e non specificate di morbosit,Ä¶ o mortalit,Ä¶
741504	401	Ipertensione essenziale

Tabella 5. Esempio della struttura dati.

Il codice id_wirgilio è l'identificativo interno del paziente al software di CCE; questa informazione è utile per valutare i nuovi casi e, quindi, calcolare l'incidenza. Le altre due colonne fanno riferimento al codice ICD-IX e alla relativa descrizione. Come spiegato nei paragrafi precedenti, il codice ICD-IX è molto dispersivo, pertanto, per rendere il dato

maggiormente sintetico e fruibile, le patologie possono essere raggruppate secondo il codice ICPC che meglio si adatta alle cure primarie dei MMG.

Per testare il software sviluppato sono stati utilizzati i dati sanitari contenuti nelle CCE di alcune cooperative di MMG operanti nelle province di Napoli, Caserta e Salerno (Tabella 4). La nostra attenzione è stata dedicata alle malattie croniche tiroidee. Tali patologie sono comprese tra i codici ICD-IX 240 e 246.9. Allo scopo di rappresentare un dato sintetico delle malattie croniche tiroidee, queste sono state raggruppate e aggregate; in questo modo si perde l'informazione specifica della patologia, ma si ottiene una visione complessiva della numerosità delle patologie croniche a carico della tiroide. In particolare, per ottenere un sottogruppo con le sole patologie di interesse, è stata usata la funzione "subset" del pacchetto base di R. Pertanto, il nuovo dataframe contiene esclusivamente i pazienti che hanno avuto almeno due prescrizioni mediche nell'anno solare (criterio usato nella fase B per filtrare i dati) con i codici ICD-IX compresi tra 240 e 246.9.

A questo punto, prima di procedere all'aggregazione dei dati, sono stati calcolati i nuovi casi per ogni anno rispetto al precedente. Sfruttando il codice id paziente contenuto nei dati, è stata usata la funzione "setdiff", presente nel pacchetto base di R "*sets*", per creare un nuovo vettore con gli id esclusivi dell'anno di interesse, eliminando pertanto quelli comuni a entrambi. Di seguito è riportata una riga di codice a titolo di esempio:

```
ss <-setdiff(anno_2009$id, anno_2008$id))
```

Nell'esempio mostrato sopra è stato considerato l'anno 2009 rispetto al 2008; in particolare, è stato creato un nuovo vettore (ss) che contiene solo gli id paziente presenti in maniera esclusiva nel solo anno 2009.

Per contare i casi presenti nel 2009 è bastato, quindi, misurare la lunghezza del vettore. Tale lunghezza, infatti, corrisponde al numero dei nuovi casi dell'anno 2009 che riportano un codice ICD-IX compreso tra 240 e 246.9.

2.7.2 Calcolo dell'incidenza

I dati ottenuti e aggregati come descritto nel paragrafo precedente sono stati usati per il calcolo dell'incidenza. Questa operazione tiene conto dei nuovi casi osservati durante l'anno di riferimento e della popolazione recettiva inizialmente priva della malattia. Tuttavia, dopo aver calcolato i nuovi casi li abbiamo rapportati alla popolazione residente (in quanto tutti gli individui sono potenzialmente soggetti a questo tipo di patologie) con età maggiore di 14 anni (in quanto i dati degli MMG si riferiscono alle persone di questa fascia di età). Sebbene la definizione di incidenza riportata nel paragrafo 2.3 preveda di rapportare i nuovi casi alla popolazione ammalabile, nel nostro studio i nuovi casi ogni anno erano nell'ordine dell'unità, quindi sottrarli al totale della popolazione residente non avrebbe cambiato in modo rilevante il valore dell'incidenza. Il calcolo dell'incidenza avviene tramite le funzioni base di R e prevede il rapporto tra i nuovi casi dell'anno considerato e della popolazione residente. Dato che il range dei risultati ottenuti era molto limitato, si è optato per esprimere tale valore per mille, ciò anche allo scopo di evidenziare le differenze in scala di colore nei plot georeferenziati.

```
incidenza <- (ss*1000) / abitanti.comune
```

2.7.3 Georeferenziazione dei dati

I risultati ottenuti dell'incidenza consistevano in singoli file di testo divisi per comune e si presentavano in questo modo (Tab. 6).

Comune	2009 _2008	2010 _2009	2011 _2010	2012 _2011	2013 _2012	2014 _2013	2015 _2014	2016 _2015
esempio	3.1	2.8	2.9	3.1	3.2	3.3	2.8	2.9

Tabella 6. Esempio struttura dati dopo filtraggio e aggregazione dei dati.

Per georeferenziare i dati presenti in questa tabella è necessario associarla a un file contenente le informazioni spaziali, ovvero a uno shape file.

Gli shape file relativi ai confini amministrativi regionali, provinciali e comunali sono stati scaricati dal sito dell'ISTAT e importati in R mediante la funzione readOGR inclusa nel pacchetto rgdal.

```
shape <- readOGR(dsn = folder.path.shapes, layer = shape.name)
```

Gli argomenti di questa funzione sono rappresentati dal "dsn", ovvero dal percorso dove si trova lo shape file, e da "layer", ovvero dal nome dello shape file.

Gli shape file relativi ai fuochi e alle discariche segnalate dalla popolazione sul sito www.terradeifuochi.it sono stati scaricati in formato klm, convertiti in shape file e importati in R con la stessa funzione readOGR, così come avvenuto per gli shape file relative alle campagne di monitoraggio condotte dall'ARPAC.

Allo scopo rendere omogenee i sistemi di riferimento e di proiezione, i file importati sono stati convertiti nel sistema di proiezione Universal Transverse Mercator (UTM), e nel sistema di riferimento World Geodetic System (WGS84) grazie alla seguente funzione:

```
shape.transformed<-spTransform(shape, CRS(" +proj=utm +zone=33n +ellps=WGS84 +datum=WGS84 +units=m +no_defs +towgs84=0, 0, 0"))
```

Gli argomenti di questa funzione sono: il nome dello shape file, e il "CRS", ovvero la stringa contenente le informazione relative al sistema di proiezione e di riferimento.

In questo caso la georeferenziazione è stata eseguita per i dati relativi alle incidenze a livello di comune. Dal punto di vista tecnico la georeferenziazione consiste nella costruzione di un **Polygon Dataframe** che avviene tramite la funzione *merge spatial dataframe* compresa nel pacchetto "sp".

```
merged.data.frame <- merge( shape.list[[ind.comune.shape.list]], patologia.comuni.dataframe[comune.ind,],by=intersect(names(shape.list[[ind.comune.shape.list]]), names(patologia.comuni.dataframe[comune.ind,])), duplicateGeoms=TRUE)
```

Gli argomenti di questa funzione sono: lo shape file, il dataframe e la chiave univoca presente sia nello shape file che nel dataframe; questa è necessaria per associare i due file. Inoltre, dato che si tratta di un poligono, per quanto descritto nel paragrafo 2.3, è necessario

duplicare i dati (attributi) per tutte le n righe rappresentanti le coordinate delle linee spezzate che costituiscono il poligono (duplicateGeoms=TRUE).

Questa funzione costruisce un oggetto di classe S4. Un oggetto di classe S4 è una struttura dati contenente diversi attributi (denominati slot) di tipo predeterminato. Dopo aver creato l'oggetto di classe S4 Spatial dataframe, questo, così come gli altri shape file importati in precedenza, è stato convertito in un dataframe che include le coordinate spaziali presenti nello slot coords e gli attributi presenti nello slot data dell'oggetto precedente. Questa operazione permette ulteriori manipolazioni e visualizzazioni in altri pacchetti come dplyr, reshape2, ggplot2 e ggvis. I dati così pre-processati sono stati archiviati in un file RData che sarà poi richiamato nelle successive funzioni.

Le funzioni utili al rendering grafico dei dataframe georeferenziati sono state sviluppate implementando quelle presenti nel pacchetto ggplot2, e in particolare la funzione ggplot. Questa funzione permette di generare grafici con oggetti georeferenziati con la possibilità di sovrapporre anche più strati informativi. Infatti, oltre a graficare il dataframe (strato informativo) dei comuni a cui sono associati i dati dell'incidenza, sono stati processati e graficati anche gli shape file dei confini amministrativi provinciali, il dataset riguardanti i fuochi e le discariche segnalate dalla popolazioni sul sito internet www.terradeifuochi.it, e le le aree interessate dalle campagne di monitoraggio dell'Agenzia Regionale per la Protezione Ambientale.

Un esempio di funzione usata in questo lavoro è riportata di seguito:

```
zp16 <- zp16 + geom_polygon(aes_string(x="long", y="lat", fill=stringa.anno, group="group", name="Name", Comune=NULL), data = patologia.dataset, colour="black")+scale_fill_gradient(low="#00FFFF",high="#00008B", limits=c(min(patologia.dataset[,stringa.anno]), max(patologia.dataset[,stringa.anno])))
```

La funzione ggplot() inizializza l'oggetto ggplot. Questa funzione è sempre seguita da "+" per aggiungere componenti al grafico. In questo caso è stata aggiunta la geometria poligono e, grazie alla funzione aes_string disponibile in ggplot2, sono state descritte quali variabili considerare per mappare le geometrie. In particolare "x" e "y" rappresentano rispettivamente la longitudine e la latitudine, "group" rappresenta la variabile secondo cui mappare le diverse righe afferenti allo stesso gruppo (identificato da un valore differente),

mentre “name” e “Comune” sono altre variabili che ggplot non prende in considerazione ma che sono utili nello sviluppo di plot interattivi. A seguire della funzione aestetics è presente l’argomento “data”, che corrisponde al dataframe da plottare, e il colore delle linee che formano il poligono (colour=”black”). E’ possibile, quindi, specificare con “+ scale_fill_gradient” in che modo colorare i poligoni. In questo caso è stata scelta una scala di colore settata sul valore minimo e massimo di tutti i comuni nei diversi anni indagati. Questa mappa è stata realizzata con la funzione scale_fill_gradient dove sono indicati i valori minimi e massimi di colore con il codice RGB, e il riferimento numerico minimo e massimo del dataframe a cui far corrispondere i colori. Al grafico zp16 è possibile aggiungere altri strati informativi come mostrato di seguito:

zp16 <- zp16+ geom_point(data = fuochi.camp.dataset, aes(x = long, y = lat, colour= Name, name=Name, Comune=Comune), size=0.4)

In questo caso è stata aggiunta la geometria punti (geom_point) relativa al dataframe fuochi.cap.dataset. Gli argomenti della funzione “aes” sono, ovviamente, la longitudine (“x”), la latitudine (“y”), il colore (“colour”), che in questo caso è legato alla variabile “nome”, e “Comune” che è ignorato da ggplot ma risulterà utile in seguito. Nel caso dei punti è possibile anche definire la misura del punto graficato (“size=0.4”).

I grafici costruiti grazie al pacchetto ggplot2 possono essere resi dinamici e interattivi grazie ad alcune funzioni presenti nel pacchetto plotly.

Plotly è un servizio per la creazione e la condivisione di “data visualization” che offre anche uno strumento per svolgere analisi statistiche. Inoltre offre interfaccia di programmazione di un'applicazione (API), la possibilità di disegnare funzioni personalizzate e una shell Python integrata. Tra le API, c’è anche quella per R: le visualizzazioni interattive offerte da plotly possono essere create direttamente in R.

In particolare è stata prevista l’integrazione con ggplot, pertanto è possibile creare grafici interattivi in plotly usando la sintassi di ggplot. Plotly per ggplot2 è una libreria di grafici interattiva basata su browser, costruita sulla libreria di grafica javascript open source di plotly, plotly.js. Funziona interamente a livello locale, attraverso il framework di widget HTML. La funzione che realizza il plot è la seguente:

```
p16 <- ggplotly(zp16)
```

In questo modo l'oggetto creato in ggplot (zp16) è reso interattivo grazie alla funzione ggplotly il cui argomento è proprio l'oggetto creato in ggplot2; di default saranno mostrate tutte le estetiche inserite nell'oggetto ggplot. Nel caso in cui si voglia specificare quale estetiche mostrare e in che ordine è possibile usare l'argomento tooltip. Ad esempio, utilizzare tooltip = c ("y", "x", "colour") se si desidera che y prima, x secondo e colore per ultimo.

2.7.4 Sviluppo shiny app

Per sviluppare un software *user-friendly* con interfaccia grafica che ne rendesse facile l'utilizzo anche a chi non ha dimestichezza con R e i comandi da terminale, è stata sfruttata la potenzialità offerta dal pacchetto Shiny.

Lo sviluppo di un software in shiny tiene conto di tre elementi:

- **interfaccia utente**, o UI, costruita come una pagina web e che può essere personalizzata con titoli, apposite sezioni per inserire i comandi di input ed altre per osservare i risultati in output sotto forma, ad esempio, di grafici;

- **server**, che contiene i comandi che permettono di definire gli output attraverso cui interpretare le analisi svolte, i quali rispondono tempestivamente alle nuove istruzioni impostate dall'utente dall'interfaccia;

- **funzione di lancio del software**, che prende il nome di shinyApp(), che riconosce il nome assegnato alle variabili "interfaccia" e "server" e permette quindi il corretto funzionamento del programma.

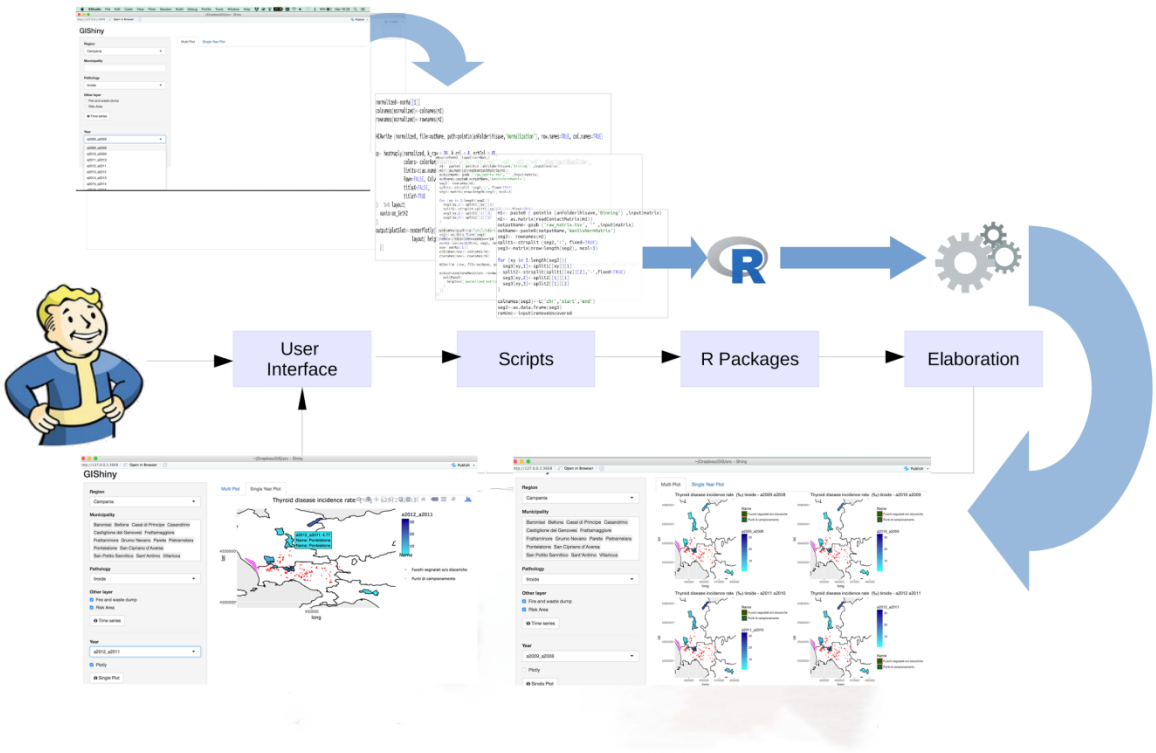


Fig. 33. Schema di funzionamento del software di georeferenziazione.

- **Interfaccia utente**

Per la costruzione di un'applicazione Shiny, il primo passo consiste nel definire un'interfaccia utente adeguata al tipo di indagine che sarà possibile svolgere con l'ausilio del software.

Tutti i comandi per la creazione di tale interfaccia, compresi quelli relativi ad intestazioni, titoli, posizione di bottoni di azione e di menù, devono essere inseriti all'interno della funzione fluidPage(), attraverso la quale è possibile visualizzare il layout definitivo del software, che è reso in formato HTML e quindi visualizzabile attraverso un qualsiasi browser per internet.

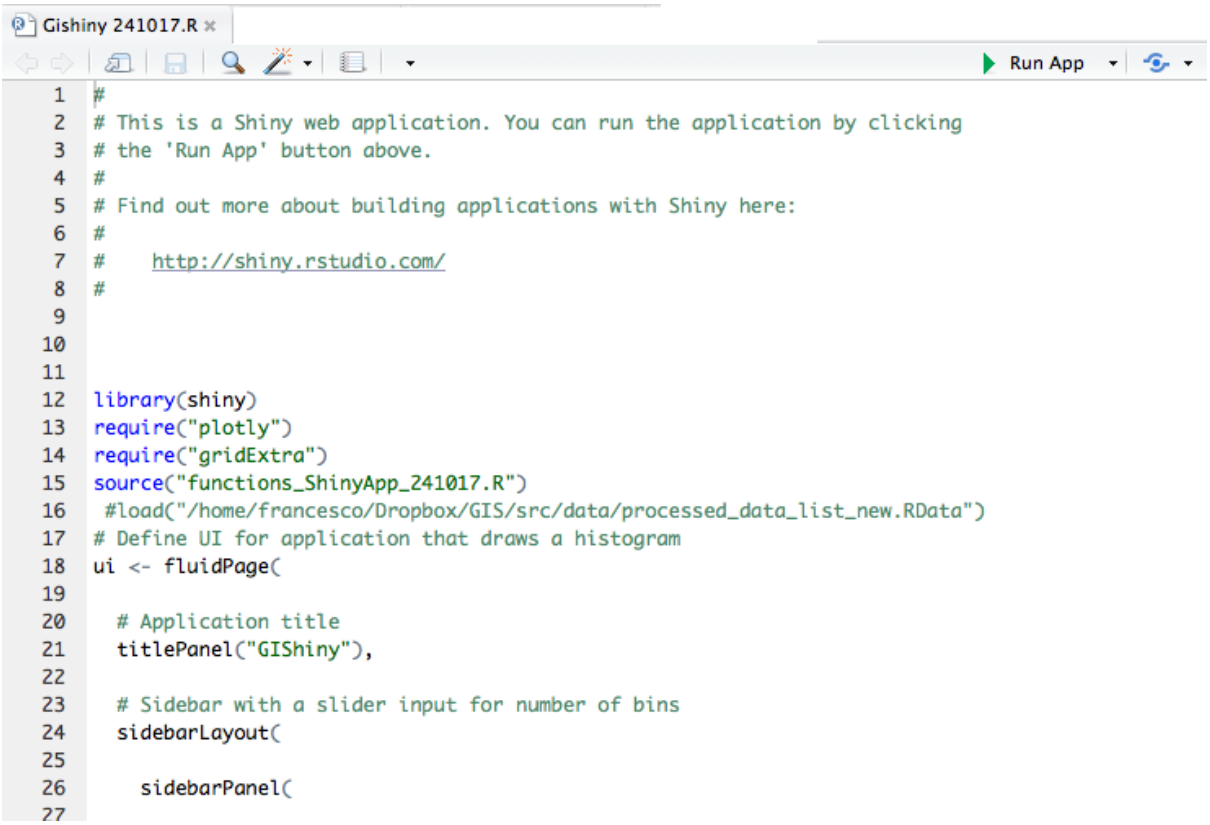


Fig. 34. Script esemplificativi per lo sviluppo GUI.

Lo scheletro dell'interfaccia è costituito da specifiche aree in cui devono essere visualizzati tutti gli elementi dell'applicativo. A tal proposito gli sviluppatori di Shiny hanno previsto diverse funzioni che permettono di aggiungere dei template predefiniti all'interno dell'interfaccia utente, in cui qualsiasi programmatore può decidere di inserire comandi o mostrare determinati tipi di risultati. È il caso della funzione sidebarLayout(), grazie alla quale, con l'ausilio di altre sub – funzioni, è possibile creare delle colonne laterali e dei pannelli centrali, così da avere a disposizione un'interfaccia in cui al lato possono essere inseriti determinati comandi, mentre nella zona centrale si possono mostrare gli output.

Gli oggetti da inserire all'interno del layout dell'interfaccia possono essere sia statici, come immagini o testo usato per titoli di determinate sezioni e veri e propri paragrafi, oppure dinamici, nel senso che l'utente può interagire con essi o possono essi stessi mutare a seconda del comando fornito all'interno del server; a questo secondo gruppo appartengono gli oggetti di input e output, fondamentali per la costruzione di un software in cui è l'utilizzatore finale a poter scegliere che tipo di risultato vuole osservare.

- **Server**

Nonostante l'interfaccia possa riservare uno spazio per l'output scelto, questo ancora non compare quando lanciata l'applicazione, poiché le funzioni di qualsiasi tipo di output sono regolate dal server di Shiny, il quale deve essere compilato con i comandi attinenti lo studio che si vuole condurre.

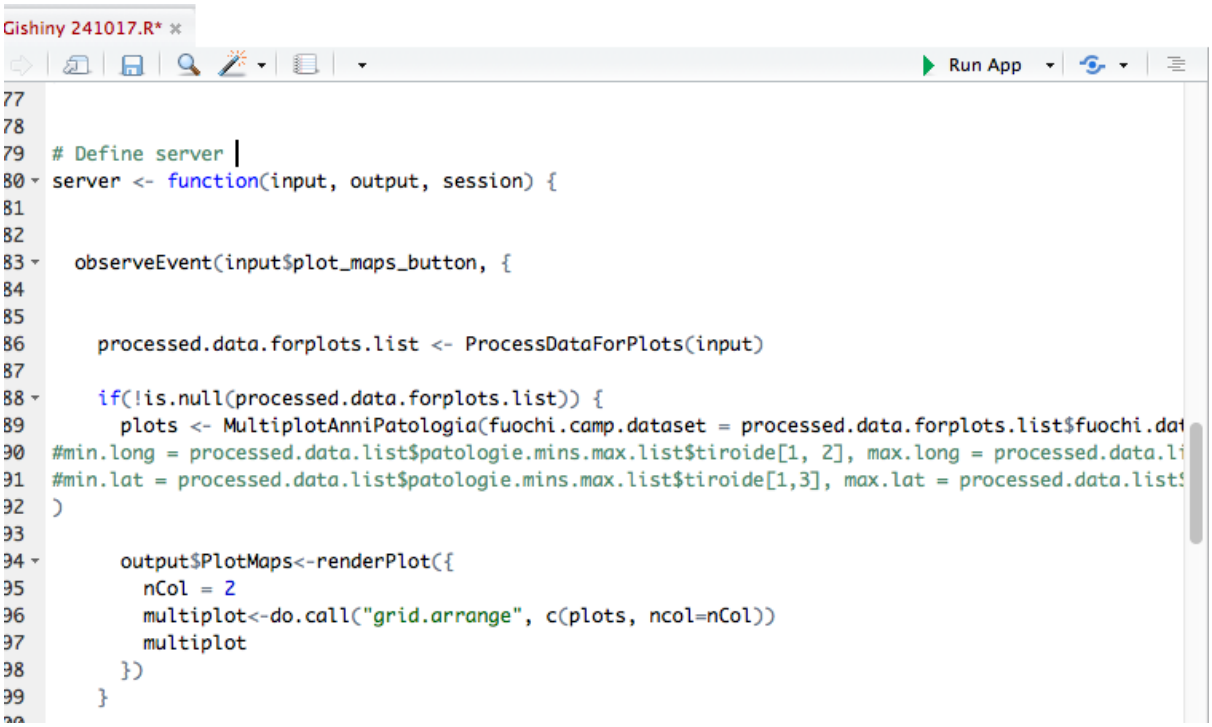


Fig. 35. Script esemplificativi per lo sviluppo GUI.

Affinché il server riconosca quelli che sono gli input e gli output definiti nell'UI, ogni tipo di istruzione che gli viene impartita per costruire un output dovrà essere inserita all'interno di una funzione iniziale i cui argomenti sono proprio gli input e gli output precedentemente impostati nell'interfaccia.

- **La funzione shinyApp()**

La funzione di shiny chiamata shinyApp() permette di far partire l'applicazione all'interno di un browser HTML. Questa funzione è costituita da due argomenti, ossia "ui" e "server", i quali devono essere uguali rispettivamente all'oggetto con cui è stata identificata l'interfaccia utente, e a quello con cui è stato identificato il server.

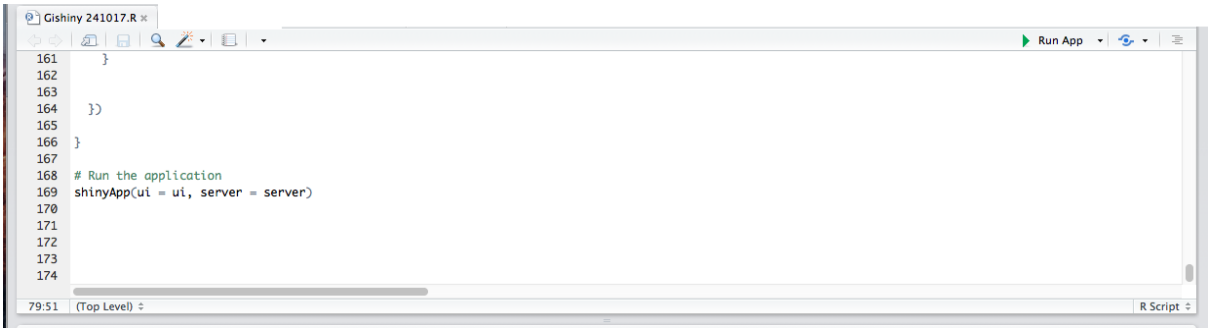


Fig. 36. Script esemplificativi per lo sviluppo GUI.

Quando questo comando completo di argomenti è presente all'interno di un R-script, nell'interfaccia di RStudio appare il pulsante "RunApp", che permette di eseguire la lettura di tutto lo script ed aprire una finestra del browser predefinito per lanciare l'applicazione. In generale, se il dataset da cui attingere informazioni è presente in R esso può essere richiamato semplicemente attraverso il comando library().

3 Dettaglio di funzionamento del prototipo sviluppato

3.1 Flusso dati

Analizzando lo schema della piattaforma, si evince che tre sono i possibili flussi di dati che, partendo dalla sorgente (cartelle MMG), vengono convogliati verso la piattaforma R-based e la piattaforma di Business Intelligence.

In tutte le condizioni i dati vengono estratti dalle sorgenti informative manipolati in maniera opportuna e memorizzati nel DW, grazie a trasformazioni ETL sviluppate in *Pentaho Data Integration (vedi paragrafo 2.3)*.

Flusso dati 1

Nel primo caso i dati, vengono estratti secondo le regole e i parametri definiti in ingresso dalle trasformazioni implementate in Data Integration *(vedi paragrafo 2.3)* che li manipola, li filtra, li formatta e li salva in un file di testo, che andrà in ingresso alla piattaforma R-based.

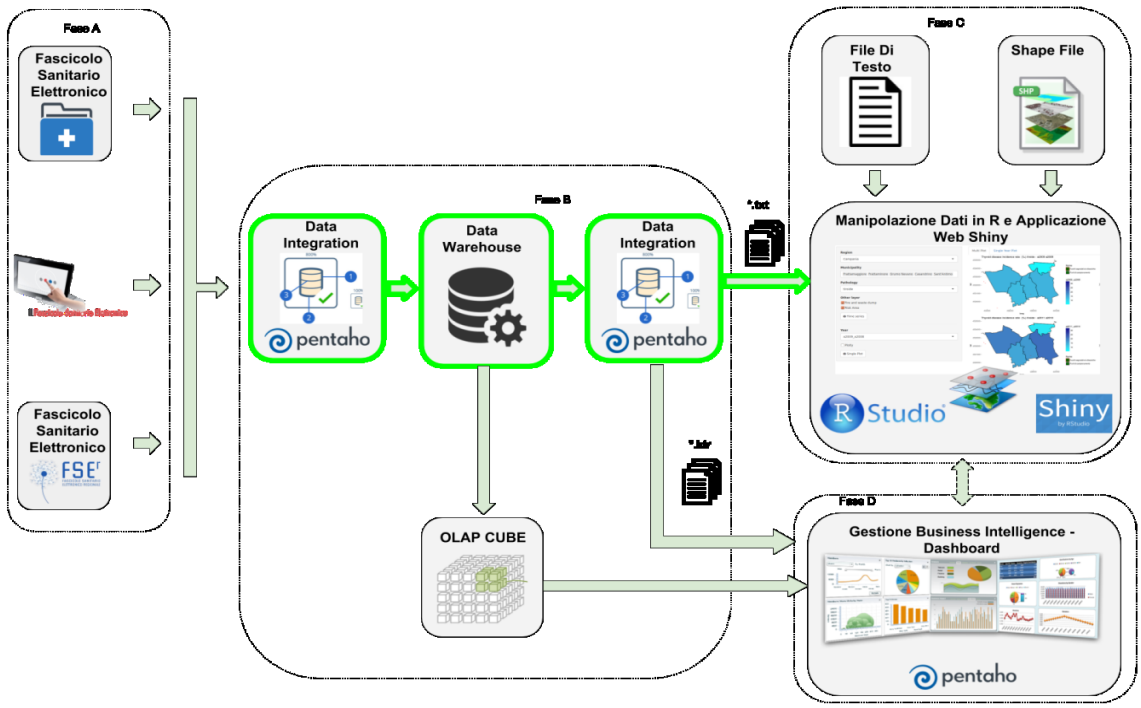


Figura 37 - Flusso dati 1

Flusso dati 2

Nel secondo caso i dati , vengono estratti secondo le regole e i parametri definiti in ingresso dalle trasformazioni implementate in Data Integration *(vedi paragrafo 2.3)* che li manipola, li filtra, li formatta e li salva la trasformazione in un file di trasformazione KETTLE (.ktr) che è uploadato nella piattaforma di business Intelligence per essere utilizzato come datasource.

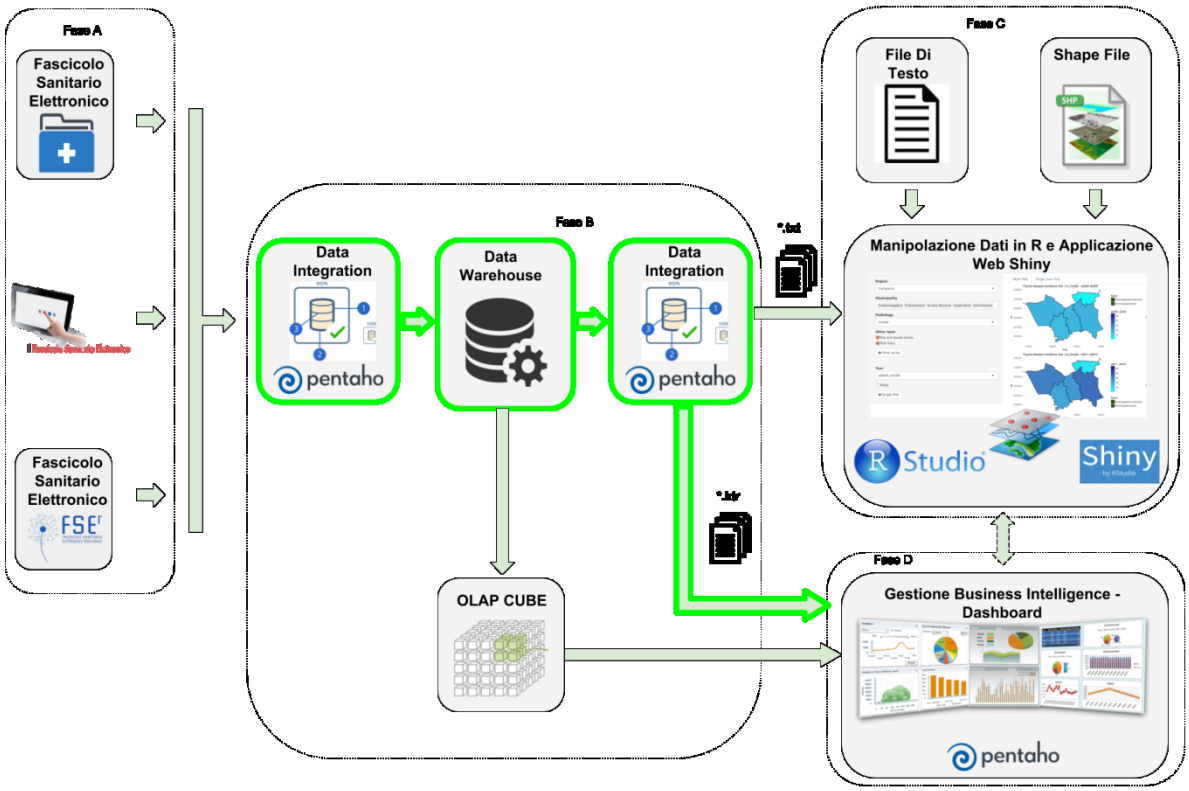


Figura 38 - Flusso dati 3

Flusso dati 3

Nel terzo caso le informazioni presenti nel DW vengono lette dalla piattaforma di BI attraverso cubi OLAP, appositamente definiti, e che vengono settati come datasource della piattaforma.

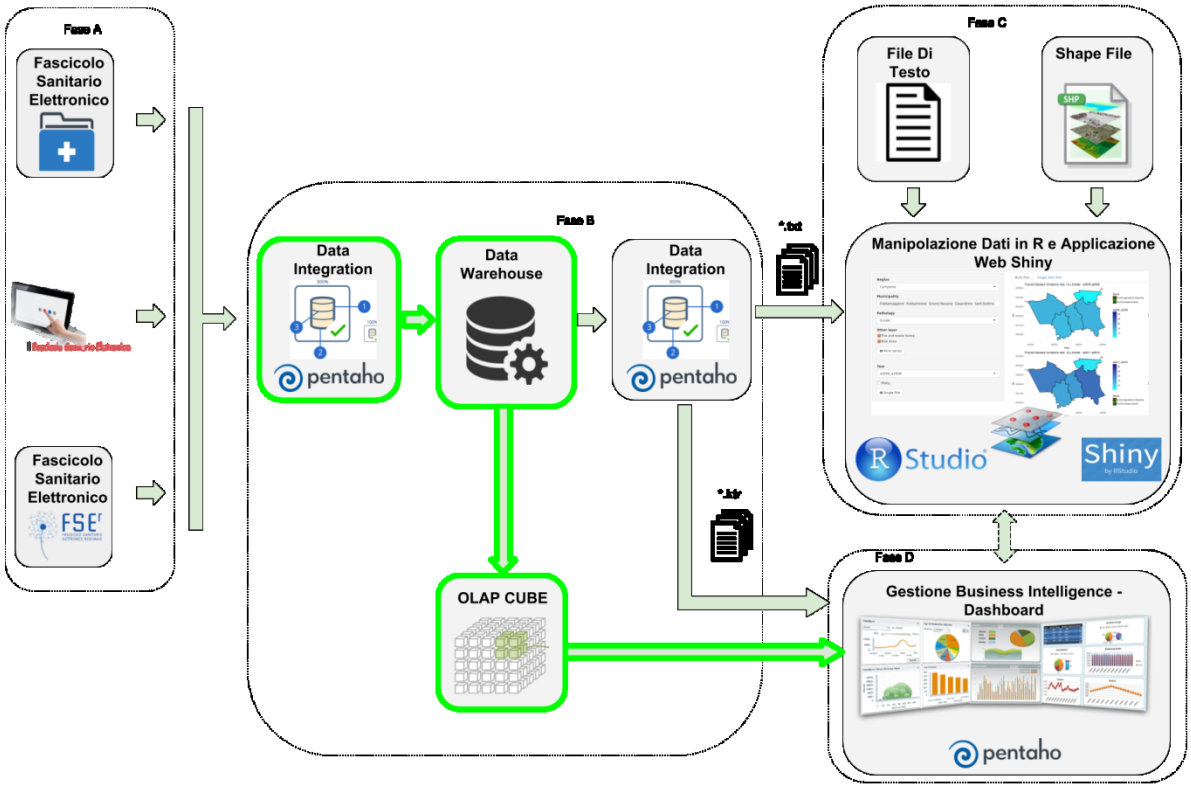


Figura 39 - Flusso dati 3

La Figura 39 mostra un esempio di cubo OLAP implementato grazie a Pentaho Schema Workbench. Tale cubo consente di visualizzare le informazioni relativi ai casi di pazienti affetti dalle diverse patologie (misura) analizzando le informazioni per anno, sesso, età, comune, codice ICDIX. Lo schema, salvato in XML contiene i metadati formattati secondo la specifica struttura (modello), usata dal Mondrian Engine presente in Pentaho BI e può dunque essere uploadato sulla piattaforma di Business Intelligence.

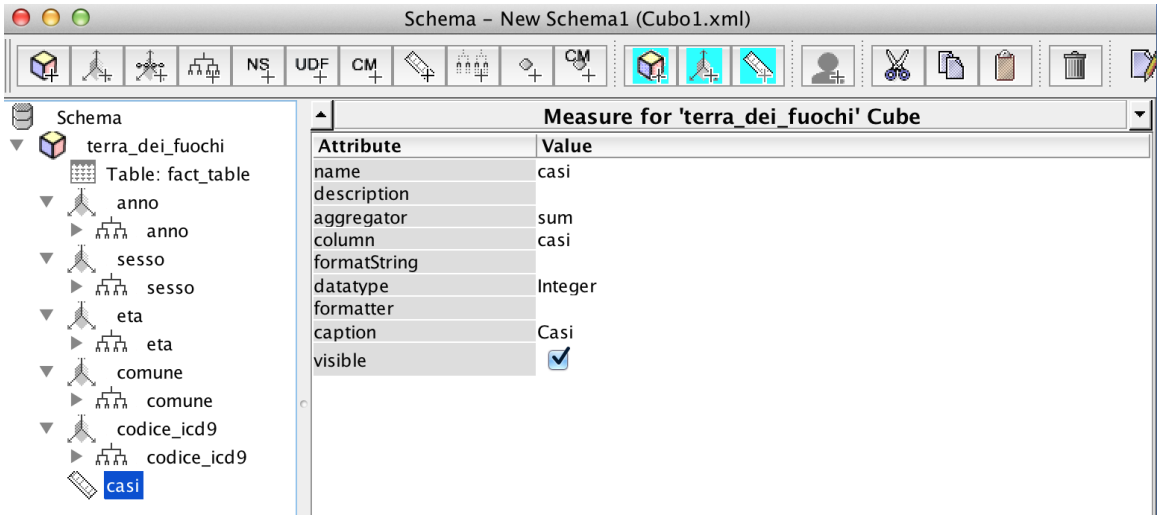


Figura 40 - Cubo OLAP

3.2 GiShiny: descrizione del software sviluppato

Gli obiettivi di questa tesi sono i) lo sviluppo di uno strumento informatico prototipale in ambiente *open source* capace di gestire i big data di tipo sanitario mediante una piattaforma di BI e di estrarre informazione nascosta dai dati, e parallelamente ii) lo sviluppo di un software in R con interfaccia grafica *user friendly* (GUI) integrabile in Pentaho Business Analytics, capace di processare e rappresentare in maniera georeferenziata i dati presenti in un data warehouse sanitario.

Questi risultati sono stati raggiunti, infatti, la piattaforma di BI è stata progettata per ospitare e gestire i dati delle CCE contenuti nei diversi server. Tra i file di output della piattaforma ci sono:

1. Dei file di trasformazione (.ktr) utilizzati come datasource per il componente di Business Intelligence di Pentaho dedicato alla produzione delle dashboard interattive.
2. Uno o più file di testo che a loro volta coincidono con l'input della piattaforma per l'analisi georeferenziata realizzata in R;

Questi ultimi sono strutturati in modo da essere geo-processati nel software prodotto in RStudio. Nell'esempio d'uso sono stati estratti, a solo scopo descrittivo del software, i dati relativi all'identificativo anonimizzato del paziente (id_wirgilio), al codice e alla descrizione

del ICD-IX della prescrizione fatta dal medico. Questi dati hanno di fatto popolano le cartelle di input del software di georeferenziazione. I dati importati e processati in R sono quindi mostrati su mappa nella Shiny app a cui è stato dato il nome di GiShiny.

GiShiny è basato sul linguaggio di programmazione *R* convertito in HTML per funzionare su pagine web accessibili da qualsiasi tipo di browser (Internet Explorer, Safari, etc.). Nella sua versione attuale, può lavorare correttamente anche offline ogni qualvolta viene lanciato direttamente da *RStudio*, ma ovviamente, per l'aggiornamento in tempo reale dei dati è necessario connettersi al server che ospita la piattaforma di BI.

Per quanto concerne i requisiti di sistema, il software funziona su calcolatori con qualsiasi tipo di sistema operativo a 32 o 64 bit (Windows, OS, Linux) e su cui sia installato *R* ed *RStudio*. I pacchetti di funzioni presenti nel software e di seguito elencati devono essere preventivamente installati in *RStudio* per il corretto funzionamento di tutti i comandi dell'applicazione:

- *shiny*;
- *ggplot2*;
- broom;
- *ggfortify* (Tang 2016);
- *maptools* (Bivand and Lewin-Koh 2017);
- *rgdal* (Bivand et al 2017);
- *plotly*;
- *gridExtra* (Auguie 2016);
- *ggdendro* (de Vries and Ripley 2016);
- *adegenet*;
- *ade4* (Chessel et al. 2004).

Per avviare il programma, una volta selezionato lo script di *R* in cui sono presenti tutte le istruzioni da impartire al server, e per generare l'interfaccia utente, è sufficiente cliccare sul tasto "Run App" di *RStudio* per aprire automaticamente una pagina HTML nel browser

predefinito per osservare il template iniziale del software, mostrato nell'immagine di seguito (Fig. 41):

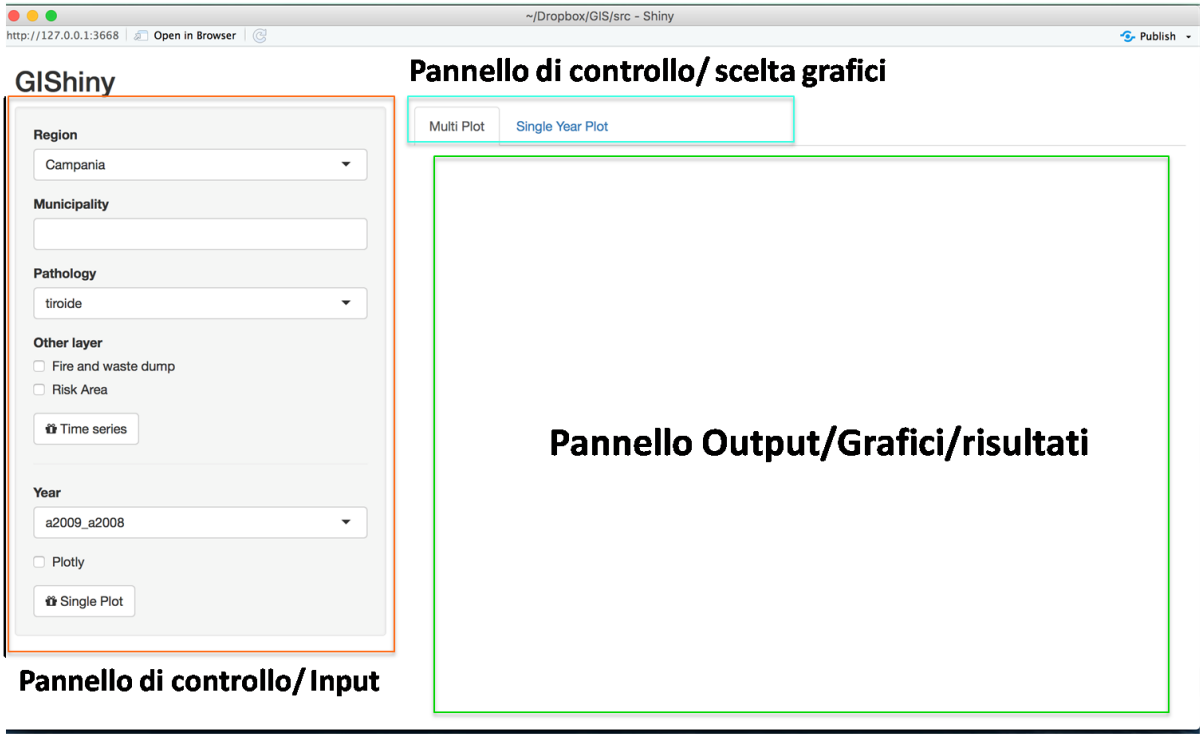


Figura 41. Schema layout GIShyni.

Quello che si può subito notare è che, al di sotto del titolo, la schermata principale è suddivisa in due colonne principali (attraverso l'utilizzo della *fluidRow*):

- la colonna a sinistra è occupata dalla sezione per la scelta dei dati da georeferenziare e mappare;
- la colonna di destra, che occupa tutto lo spazio del pannello centrale dell'interfaccia, è quella in cui verranno mostrati i contenuti, e quindi tutti i risultati delle azioni svolte nell'applicazione.

Nella colonna di sinistra (rettangolo arancione), sotto al nome dell'applicazione, sono presenti una serie di menù per la definizione di cosa vogliamo vedere e in quale area.

Partendo dall'alto verso il basso, nel primo menù a tendina è possibile selezionare la/le regione/i (Region) di interesse. In questo caso d'uso è presente solo la Campania. Dopo aver selezionato la regione, nel menù "Municipality" saranno caricati tutti i comuni presenti nella regione; l'utente potrà selezionare quelli di suo interesse. A questo punto l'operatore potrà scegliere quale raggruppamento di patologie (aggregate secondo il codice ICPC) visualizzare. Insieme allo strato informativo dei comuni, colorati con una scala di colore definita in funzione del valore di incidenza dell'anno selezionato nel menù a tendina "Year", è possibile visualizzare come strati informativi sovrapposti altri layer, ovvero i punti dei fuochi e delle discariche abusive segnalate dalla popolazione (di colore rosso) e dei punti di campionamenti degli enti di controllo (colorati in viola), e quello delle aree a rischio analizzate dall'ARPAC (poligoni colorati in verde). Per visualizzare questi layer è sufficiente spuntare le caselle "Fire and waste dump", e "Risk area", rispettivamente.

Dopo aver selezionato le informazioni da visualizzare, le aree e l'anno di interesse, l'utente può scegliere se visualizzare un grafico statico, o interattivo e dinamico.

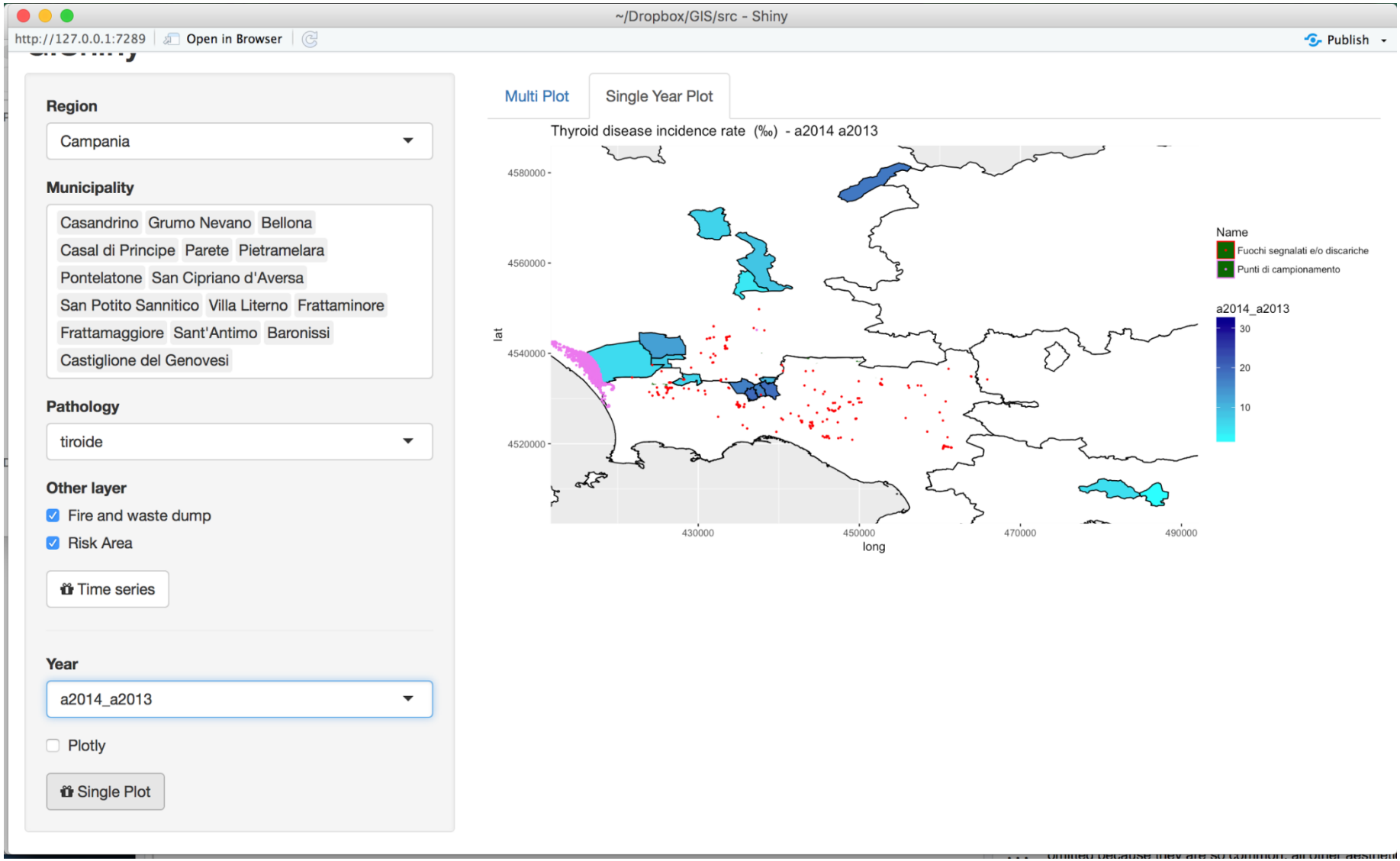


Fig. 42. Esempio output GIShyni.

Per visualizzare un grafico statico è sufficiente premere il bottone “Single plot” nella colonna sinistra di input, e selezionare, nella barra superiore dedicata alla scelta del grafico da visualizzare (evidenziata da un rettangolo celeste), la finestra “Single Year Plot”. A questo punto il risultato sarà visualizzato nel pannello centrale, evidenziato da un rettangolo verde. In questo caso il software procederà con uno zoom automatico sulle aree selezionate; i colori di riempimento dei poligoni rappresentanti i comuni sono scalati sul minimo e massimo dei valori presenti nel dataframe, ovvero rispetto a tutti i comuni e rispetto a tutti gli anni.

Nel caso in cui l’utente voglia produrre un grafico interattivo può spuntare l’opzione “Plotly” e cliccare su “Single plot”; in questo caso il grafico, che sarà sempre visualizzabile nella finestra selezionata in alto “Single year plot”, mostrerà l’intera regione, con i confini provinciali, i punti di interesse eventualmente selezionati (fuochi e discariche abusive e/o aree a rischio), e i comuni colorati secondo la stessa scala descritta in precedenza. In questo caso il grafico *plolty* non solo rende interattivo un generico grafico, ma permette anche di visualizzarne meglio tutte le informazioni ad esso associate dato che è possibile utilizzare funzioni di zoom e di pan per muoversi all’interno di esso, o visualizzare i dati associati agli strati informativi semplicemente passando con il puntatore sul punto o sul poligono di interesse.

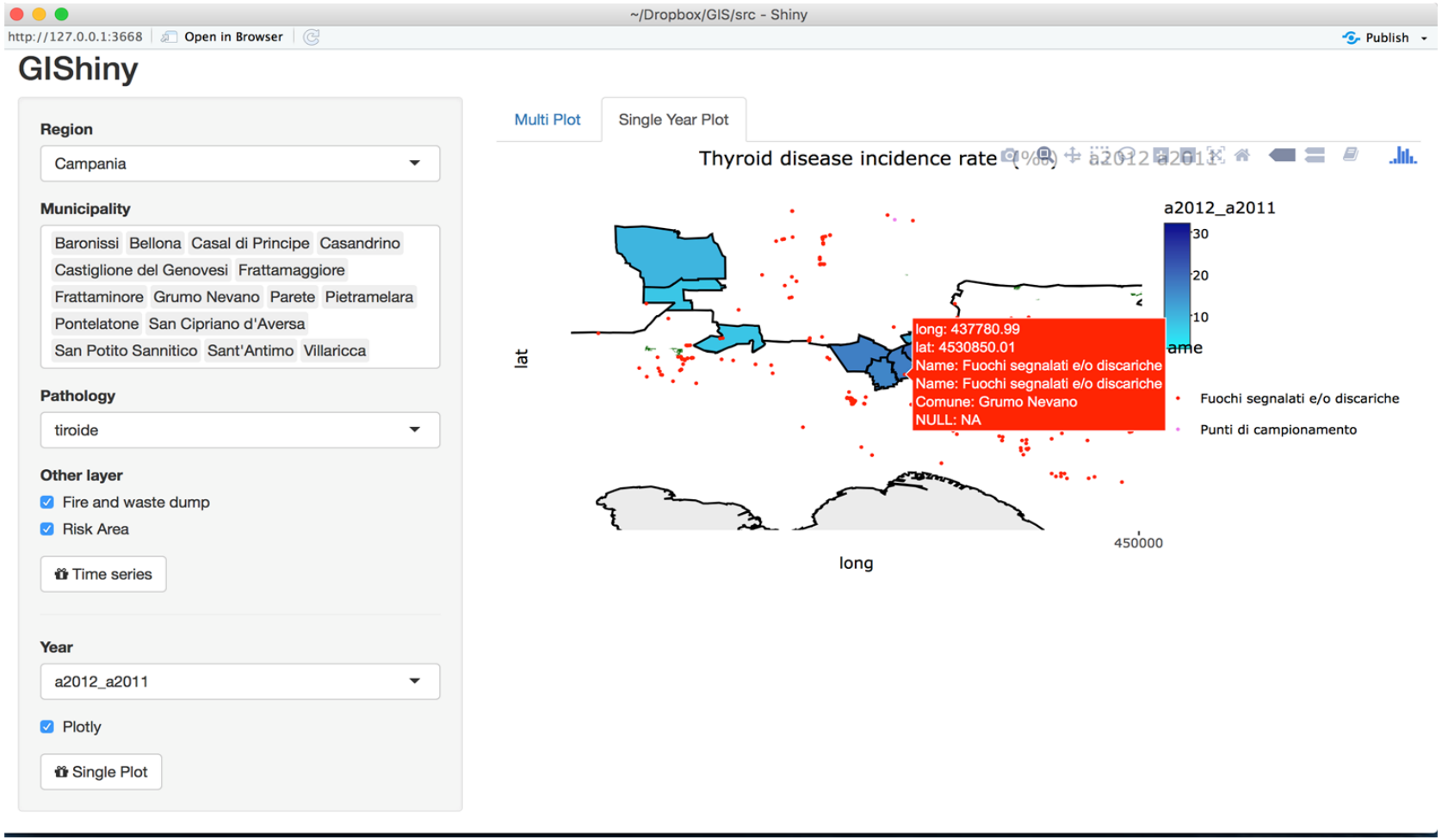


Fig. 43. Esempio output sofware GIShiny.

Nel caso in cui l'utente desideri una visione storica dei dati può selezionare come in precedenza la regione, i comuni di interesse, la patologia, se visualizzare o meno altri strati informativi come i fuochi e le discariche abusive o le aree a rischio, ma in questo caso non è necessario selezionare l'anno di interesse, in quanto cliccando sul bottone "Time series", e selezionando in alto la finestra "Multiplot", si produce il grafico statico relativo ai dati delle patologie di interesse, rappresentate con la stessa scala di colori descritta in precedenza, per tutti gli anni presenti nel dataframe. Anche in questo caso la mappa sarà ingrandita in automatico in base ai comuni selezionati.

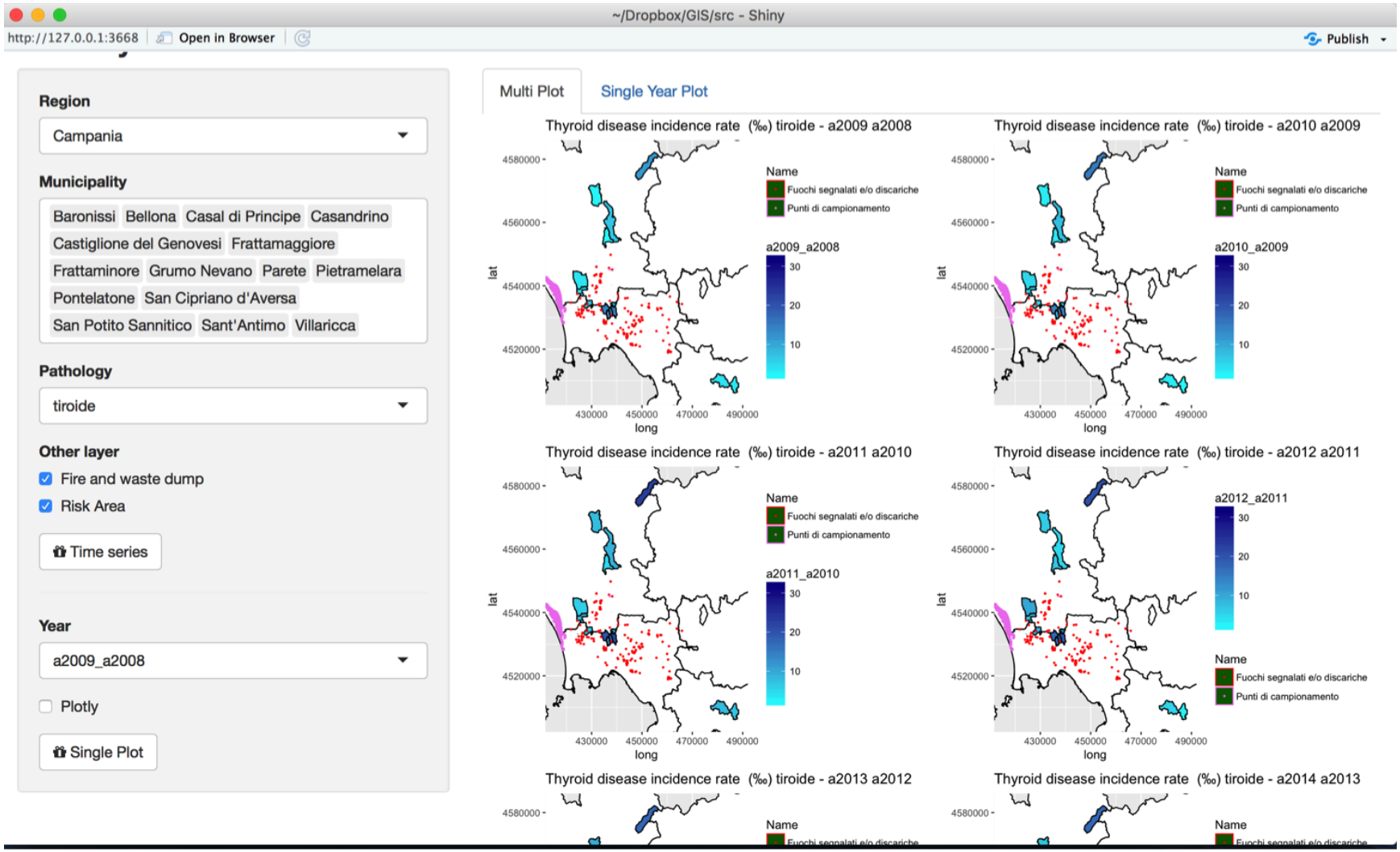


Figura 44. Esempio output software GISHiny.

3.3 Fase D - Dashboard

In questo paragrafo vengono descritte due Dashboard implementate.

Dashboard casi per ICDIX

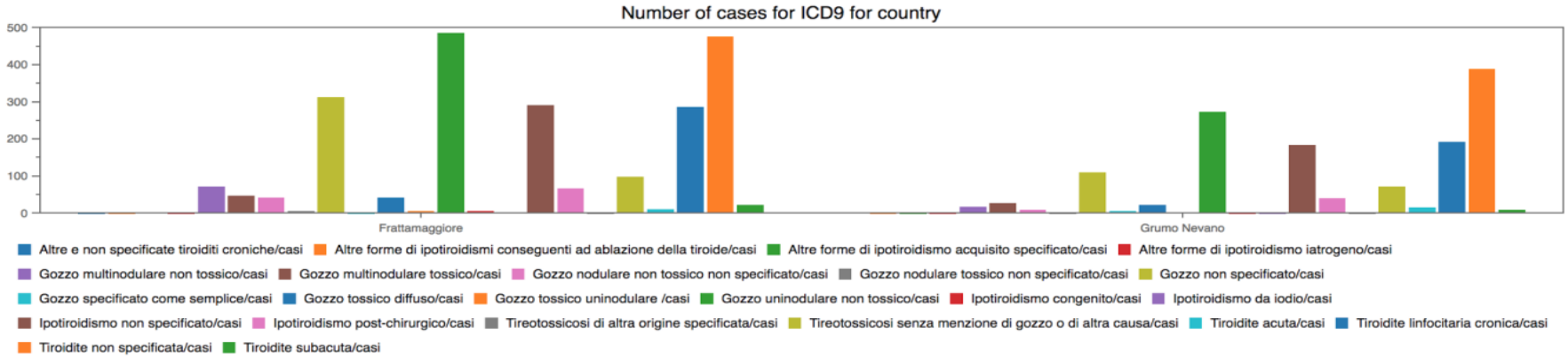
Questa Dashboard consente di monitorare il numero di casi di assistiti affetti dalle patologie divise per ICDIX, rielaborando i dati in maniera dinamica sulla base dei parametri “Gender” e “Country” selezionati attraverso appositi menu a tendina.

Come è possibile notare osservando la figura 45, i dati sono plottati su tre grafici:

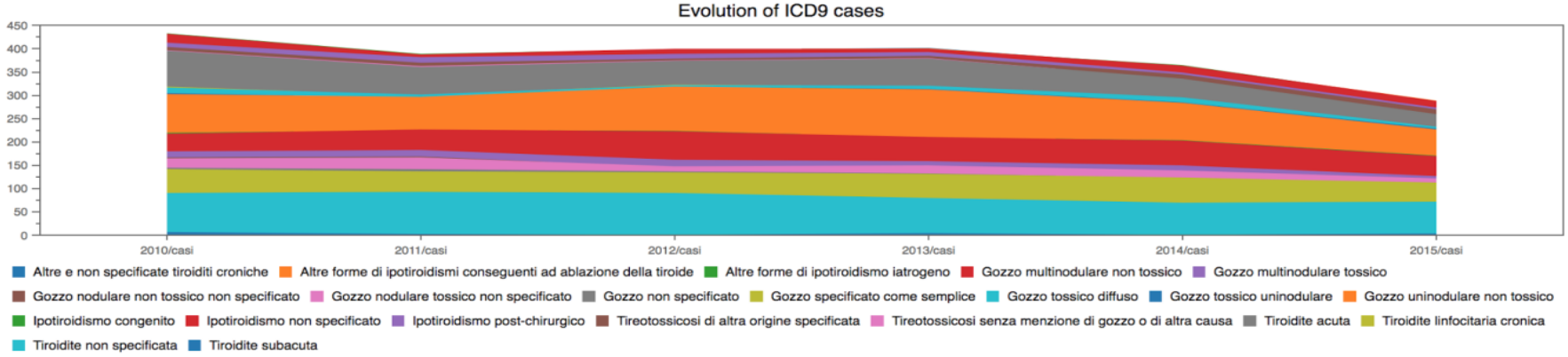
1. grafico a barre
2. stacked Area
3. grafico a torta

ICD9 Pathology

Select Gender: Femmina



Select Country: Frattamaggiore



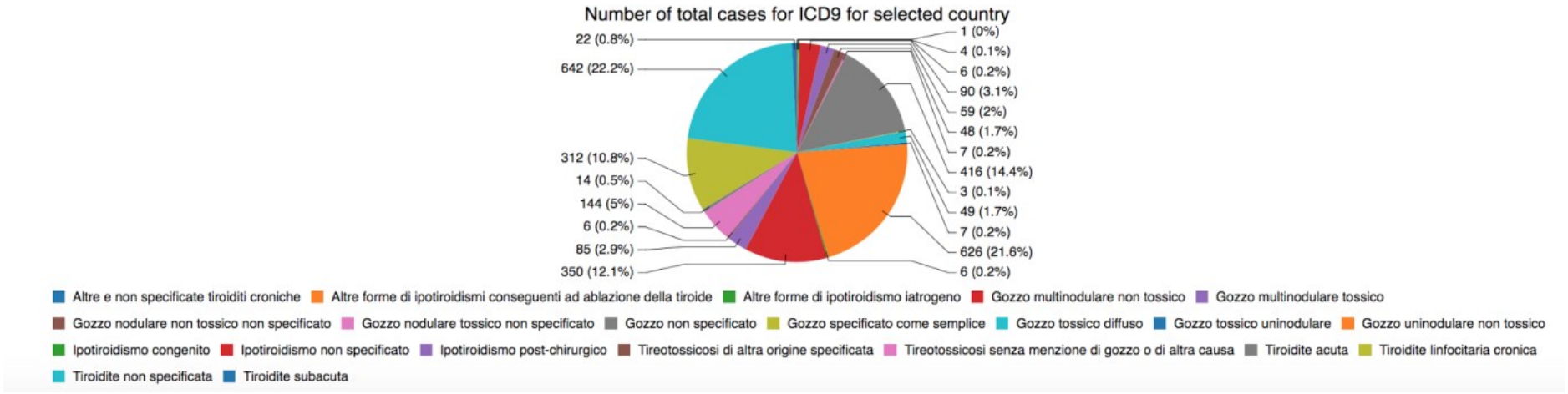


Figura 45 - Dashboard casi per ICDIX

Dashboard Incidenza

Questa Dashboard consente di monitorare l'evoluzione dei valori di incidenza delle diverse patologie, rielaborando i dati in maniera dinamica sulla base dei parametri "Country" e "Pathology" selezionati attraverso appositi menu a tendina.

Come si evince dalla Figura 46, le informazioni vengono rappresentate in forma tabellare e plottate su un grafico a barre, consentendo di mostrare l'evoluzione dell'incidenza al passare degli anni.

Figura 46- Dashboard Incidenza

Bibliografia

Bivand, R. K., T.; Rowlingson, B. *rgdal: Bindings for the Geospatial Data Abstraction Library*. R package version 1.2-5. (2016).

Buchanan, Leigh; O Connell, Andrew. *A brief history of decision making*. Harvard business review, 84.1: 32. (2006.

Chaudhuri, Surajit; Dayal, Umeshwar; Narasayya, Vivek. *An overview of business intelligence technology.* Communications of the ACM, 54.8: 88-98. (2011).

Chang W, Cheng J, Allaire J, Xie Y, McPherson J. *shiny: Web Application Framework for R.* R package version 0.14.1, URL https://CRAN.R-project.org/package=shiny. (2016)

Chen, Hsinchun; Chiang, Roger HL; Storey, Veda C. *Business intelligence and analytics: From big data to big impact*. MIS quarterly, 36.4. (2012)

DY, Barry; CY, Lynne Doran. *Data warehouse: from architecture to implementation*. Addison-Wesley Longman Publishing Co., Inc., (1996).

Ervural, Beyzanur Cayir; Ervural, Bilal. *Overview of Cyber Security in the Industry 4.0 Era.* In: Industry 4.0: Managing The Digital Transformation. Springer, Cham,. p. 267-284. (2017)

Franklin, C. *An introduction to geographic information systems: linking maps to databases.* (1992).

Gartner, Inc., 19 March 2015 report. *Transitioning to Value-Based Healthcare: Building Blocks for Effective Analytics*. ID: G00273137. Analyst(s): Laura Craft. (2016)

Gartner, Inc., 2014 report. *Top Actions for Healthcare Delivery Organization CIOs, 2014: Avoid 25 Years of Mistakes*. Enterprise Data Warehousing. (2015)

Häyrinen, Kristiina; Saranto, Kaija; Nykänen, Pirkko. *Definition, structure, content, use and impacts of electronic health records: a review of the research literature.* International journal of medical informatics, 77.5: 291-304. (2008)

Inmon, William H. *Building the data warehouse*. John wiley & sons. (2005).

Mettler, Tobias; Vimarlund, Vivian. *Understanding business intelligence in the context of healthcare.* Health informatics journal, 15.3: 254-264. (2009).

Pebesma E. J. R. S. B. *Classes and methods for spatial data in R.* (2005).

Rivest, S. et al.. *SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data.* (2005).

RStudio: Integrated development environment for R (RStudio, Boston, MA, 2012).

Robinson, D.. **broom: Convert Statistical Analysis Objects into Tidy Data Frames**. R package version 0.4.2. (2017).

Sievert, C. P., C.; Hocking,T.; Chamberlain,S.; Ram, K.; Corvellec, M.; Despouy, P.. *plotly: Create Interactive Web Graphics via 'plotly.js'.* R package version 4.5.6. (2016).

Tang, Paul C., et al. *Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption.* Journal of the American Medical Informatics Association, 13.2: 121-126. (2006).

Turban, Efraim; Sharda, Ramesh; Delen, Dursun. *Decision support and business intelligence systems.* Pearson Education India. (2008)

Weed, Lawrence L. Medical records, patient care, and medical education. Irish Journal of Medical Science (1926-1967), 39.6: 271-282. (1964)

Wickham, H. F., R. dplyr: *A Grammar of Data Manipulation.* R package version 0.5.0. (2016).

Wickham, H.. ggplot2*: Elegant Graphics for Data Analysis*. R package (2009).

https://www.jaspersoft.com

https://eclipse.org/birt

www.terradeifuochi.it