

*Back-End: Procedure di semplificazione,
archiviazione e interrogazione dati
geografici*

Deliverable A2

Versione	Data	Autori
0,2	30/03/15	Roberto Demontis Laura Muscas

1 Introduzione e obiettivi

Il WP1 del progetto INNO prevede la progettazione e la realizzazione di un prodotto in grado di fornire servizi per il deployment, l'archiviazione, la gestione e l'interrogazione di dati Geografici e GIS tramite il meccanismo di tassellamento (tiling) vettoriale, con un approccio basato su un motore NoSQL, che garantisca scalabilità e replica.

Tale obiettivo è dettato dalla necessità di trattare efficacemente in fase di pubblicazione dati di dimensione importante tramite il meccanismo di tassellamento.

Il tiling per il trattamento dei dati GIS è frequentemente realizzato in formato raster. Tale scelta limita però l'interattività nelle interfacce utente. L'uso di tile vettoriali introduce invece il problema delle dimensioni in quanto sono notevolmente più pesanti di quelle di tipo raster.

La generazione dei tasselli può avvenire in due modi: *dinamicamente*, con le limitazioni dovute ai tempi di elaborazione di dati con un elevato numero di vertici ma la capacità di gestire dati dinamici (WMS/WFS); *staticamente*, tramite una pre-elaborazione di mosaici multilivello (WCS/WMTS) con gli ovvi problemi di gestione dei dati dinamici e delle risorse necessarie per la gestione dell'elevato numero di tasselli.

Nel progetto Inno si realizza un tassellamento vettoriale eseguito staticamente tramite una pre-elaborazione dei dati gestendo dinamicamente solo gli attributi alfanumerici degli elementi. Per poter fare questo vengono gestiti separatamente i dati vettoriali delle geometrie e quelli alfanumerici che non vengono tassellati. Per ovviare ai problemi di dimensionamento dei dati creati si è scelto un motore NoSql come gestore dell'archiviazione e pubblicazione dei tasselli evitando però operazioni di *Reduce()* e limitando anche quelle di *Map()* cioè di indicizzazione.

Come descritto nel deliverable A1 al paragrafo 5 "Il motore db NoSQL scelto e installato su Cloud", il motore NoSql scelto è Couchbase Server¹ nella versione Community. I dati saranno caricati su un unico database, chiamato nell'ambiente Couchbase *bucket*, con nome "inno".

Nel presente documento viene descritta l'applicazione web "innomanager" che offrirà gli strumenti per l'inserimento dei dati GIS nel motore NoSql e le viste sui dati per utenti non amministratori ma aventi credenziali per inserire / modificare i dati. Gli amministratori avranno a disposizione anche l'interfaccia di gestione standard di Couchbase provvista di molteplici pannelli di diagnostica (figura 1) e una serie di script shell per l'esecuzione delle stesse operazioni permesse nell'applicazione web "innomanager" in maniera più performante.

L'inserimento dei dati nel motore NoSql richiederà una pre-elaborazione degli stessi che consiste nella loro tassellazione che genererà i documenti finali, ovvero i tasselli con le geometrie. Le informazioni alfanumeriche saranno invece gestite con degli indici spaziali ed erogate separatamente.

Nel seguito: nel paragrafo 2 verrà analizzato lo stato dell'arte nella gestione dei dati GIS con motori NoSql e descritto il trattamento dei dati GIS in Inno. Nel paragrafo 3 verrà descritto lo stato dell'arte, gli standard ed i formati usati nel tassellamento vettoriale di dati geografici, quindi sarà mostrata la soluzione utilizzata nel progetto Inno. Nel paragrafo 4 verrà descritta la base dati in Inno, cioè la struttura dei documenti e gli indici spaziali o per chiave utili al loro accesso. Nel paragrafo 5 verranno mostrate le funzionalità dell'applicazione web per il deployment, l'archiviazione, la gestione e l'interrogazione dei dati. Infine nel paragrafo 6 verranno mostrate le modalità di accesso e utilizzo dell'interfaccia utente offerta dal prodotto.

¹ <http://docs.couchbase.com/couchbase-manual-2.2/>

2 Stato dell'arte nel trattamento dati Gis nei database NoSQL

Il dato GIS, anche nella sola parte geometrica, è un dato complesso con specifiche proprietà come ad esempio il sistema di riferimento per le coordinate delle geometrie. Proprio per questa sua intrinseca complessità solo recentemente sono stati sviluppati metodi di indicizzazione ad hoc per la sua gestione.

A supporto di quanto detto si può notare come nelle librerie di base dei database NoSql si trovino sempre gli strumenti per definire indici uni-dimensionali sulle geometrie più semplici, i punti, ma non quelli utili per geometrie più complesse come i poligoni. Questo ovviamente perchè i punti potevano essere gestiti bene nei metodi di indicizzazione chiave-valore all'origine dei primi motori No-Sql.

Da alcuni anni stanno via via trovando comunque spazio soluzioni più mature che usano principalmente degli indici R-Tree, come ad esempio in Neo4j², Couchbase³, Hbase e Hadoop⁴, mongoDB⁵. Tra questi particolarmente interessante la soluzione open source Spatial Hadoop e Neo4jSpatial, tuttavia non utilizzabile efficacemente per il tassellamento. Il metodo di indicizzazione più popolare rimane però ancora quello basato sulla chiave GeoHash ricavata dalle coordinate di un punto; questo metodo è preminente in diversi motori come ad esempio MongoDB o Hbase o Cassandra.

Oltre agli indici altro importante criterio di valutazione sono gli strumenti utili a caricare il dato nel motore, ovvero le procedure ETL (Extract, Transform, Load) che permettono l'importazione dei formati di scambio dei dati GIS. Esistono diverse estensioni per i maggiori motori NoSql Hbase / Hadoop⁶, MongoDB⁷.

Come si può notare nella totalità dei casi ci si basa sulle librerie GDAL derivanti dagli standard OpenGis. Standard che sono considerati, a torto, da molti sviluppatori un inutile orpello nella gestione del dato GIS nei motori No-SQL.

Allo stato dell'arte, basandoci solo sugli strumenti in dotazione ai database, i motori NoSql non sono neanche paragonabili con i più maturi DBMS Relazionali e le loro estensioni GIS quali PostGIS⁸, Oracle Spatial⁹ o SpatialLite. Se però teniamo presenti le doti di scalabilità e flessibilità nel trattamento dei Big Data non si può che constatare notevoli potenzialità e spazio per la creazione di nuovi standard per l'accesso ai dati.

Couchbase, il motore scelto per il progetto Inno, ha nella versione base la possibilità di creare indici spaziali basati su R-Tree. Tale capacità deriva da GeoCouch. Non sono invece presenti strumenti di base per l'import diretto dei dati GIS su database che comunque sarebbero inutili vista la necessità di creare delle procedure ETL ad hoc per la generazione dei tasselli.

Inoltre per la generazione dei tasselli nella procedura ETL si preferisce il database relazionale PostgreSQL con estensione PostGIS particolarmente efficace e stabile per le elaborazioni su dati di dimensioni importanti come quelli che si vogliono utilizzare.

2 <https://github.com/neo4j-contrib/spatial>

3 <https://github.com/couchbase/geocouch/wiki/Spatial-Views-API>

4 <http://spatialhadoop.cs.umn.edu/>

5 <http://www.slideshare.net/nknize/rtree-spatial-indexing-with-mongodb-mongodc>

6 http://www.slideshare.net/Hadoop_Summit/grailer-hochmuth-june27515pmroom212v3

7 <http://spatialhadoop.cs.umn.edu/publications/phdg09-eldawy.pdf>

8 <http://postgis.net/>

9 <http://www.oracle.com/technetwork/database/options/spatialandgraph/overview/index.html>

3 Tassellamento vettoriale procedura di semplificazione e stato dell'arte

Il tassellamento vettoriale dei dati GIS non è attualmente molto frequente sul web. La crescente diffusione di nuovi device mobile con elevate capacità elaborativa e la maggiore velocità della rete ha però dato un nuovo impulso a questa metodologia di trattamento dei dati GIS che permette interfacce utente più interattive. I principali problemi riguardano le dimensioni del singolo tassello e i metodi di gestione dei dati tassellati. Per ovviare al primo problema si usa la semplificazione delle geometrie, cioè la riduzione del numero dei vertici, creando versioni con maggiore o minore dettaglio a seconda dei livelli di zoom che si vogliono avere nella visualizzazione.

Oltre alla semplificazione un altro parametro importante che impatta sulle dimensioni dei tasselli è il formato usato per la trasmissione. Al momento non esiste uno standard nel formato di rappresentazione dei tasselli vettoriali, i due più usati sono il MVT (Mapnik Vector Tiles) ed il GeoJSON e la sua estensione TopoJSON. Generalmente sono privi di informazioni sul sistema di riferimento che comunque è implicitamente il WGS84 con coordinate angolari.

Il formato MVT proposto da Mapnik¹⁰ è un formato binario compresso contenente la rappresentazione WKB (Well Known Binary - http://en.wikipedia.org/wiki/Well-known_text) delle geometrie e json per gli attributi.

Il GeoJSON è una rappresentazione testuale delle geometrie e degli attributi alfanumerici. Per ovviare ai problemi dovuti alle dimensioni e per poter trattare al meglio la topologia è stata creata una sua estensione il TopoJSON (<https://github.com/mboostock/topojson>) che permette un suo utilizzo più performante.

Nel progetto Inno viene utilizzato il JSON, la rappresentazione delle geometrie segue un formato proprio. I tasselli seguono le specifiche *Slippy Map TileNames*¹¹, oramai uno standard, con tasselli di dimensione di 256 pixel logici in larghezza ed altezza. Dopo una trasformazione in Simple Mercatore dei vertici le coordinate vengono riscritte relativamente all'origine del tassello con coordinate intere positive e minori di 256. Per la rappresentazione di un vertice sono quindi sufficienti 4 byte in valori esadecimali con la possibilità futura di rappresentarlo tramite 2 byte in forma binaria. Tale procedimento implementa anche una semplificazione sulle geometrie di tipo linea o poligono diminuendone i vertici. Alcuni esempi di documenti con tasselli vettoriali si trovano in appendice.

La rappresentazione finale della geometria è simile a quella WKT¹², inizia con il codice del tipo e le coordinate sono inserite tra parentesi tonde:

Nome	Codice	Struttura
POINT	1	1(#####)
LINEA	2	2(#####...#####)
POLIGONI	3	3((#####...#####),(#####...#####))

Tabella 1: tipi di geometrie e loro rappresentazione

¹⁰ <http://wiki.openstreetmap.org/wiki/Mapnik>

¹¹ http://wiki.openstreetmap.org/wiki/Slippy_map_tilenames

¹² http://en.wikipedia.org/wiki/Well-known_text

Nel tassello è comunque sempre presente la rappresentazione in GeoJSON della sua area in latitudine e longitudine WGS84 e l'identificatore, ciò permette l'indicizzazione e la trasformazione delle coordinate in coordinate angolari WGS84 o planari Simple Mercatore sul client nel FrontEnd.

I tasselli con le informazioni alfanumeriche sono invece di due tipi e vengono creati dinamicamente tramite un indice spaziale sugli elementi del dato GIS. Il primo tipo consiste nei valori di un singolo attributo mentre il secondo nei valori di tutti gli attributi. Alcuni esempi di documenti con i tasselli delle informazioni alfanumeriche si trovano in appendice;

Nel progetto Inno si è deciso di imporre dei vincoli molto restrittivi sulle dimensioni delle tile che non possono superare i 25 kb. Per ovviare alla perdita di informazione conseguente a tale limite è previsto un meccanismo di impaginazione che permette di costruire le pagine di uno specifico tassello che saranno poi caricate a discrezione del client sul Front End.

Per definire la priorità degli elementi nella paginazione, e quindi gli elementi da restituire per primi al client, si è scelto di usare la proprietà geometrica delle geometrie: nello specifico l'area per i poligoni, la lunghezza per le linee e una semplice clusterizzazione nel caso dei punti.

Inoltre, visto l'alto numero di tasselli da generare, si è deciso di ridurre il numero almeno per il dato GIS di tipo poligonale. Sono quindi state realizzate le Macro-Tile ovvero insiemi di tasselli con un'area complessiva rettangolare contenenti la stessa informazione. I Macro Tasselli sono insiemi di tasselli completamente contenuti in un poligono e gestiti con indici spaziali.

La generazione dei tasselli viene eseguita su PostgreSQL PostGIS in quanto questo permette di applicare funzioni GIS evolute e riesce a gestire efficacemente anche geometrie di grandi dimensioni.

4 La base di dati NoSql definita in INNO

I tasselli e le informazioni sui dati GIS sono caricati su di un unico database o bucket di Couchbase. Il formato utilizzato è il documento JSON. Si possono individuare 3 tipologie di documenti: il documento con le informazioni di un dato GIS o strato informativo che chiameremo “Strato” (tabelle 1 e 2); il documento con il tassello geometrico che chiameremo “Tile”; il documento con le informazioni alfanumeriche di un elemento di uno strato che chiameremo “Infos”.

I documenti “Strato” hanno come chiavi di indirizzamento diretto il campo chiamato “_inno_name_”. I documenti di tipo “Tile” hanno come chiavi di indirizzamento diretto un campo testuale derivato dalla chiave dello strato informativo e dalle coordinate Slippy Map Tile names del tassello. I documenti di tipo “Infos” hanno come chiavi di indirizzamento diretto un campo testuale derivato dalla chiave dello strato informativo e dall'identificativo dell'elemento che è la sua priorità in base all'ordinamento eseguito in fase di pre-trattamento del dato, prima dell'inserimento nel database. Un esempio dei documenti si trova in appendice.

Documenti “Strato”

Nome	Tipo	Descrizione
_inno_name_	Testo	Identificativo univoco dello strato informativo
bbox	GeoJSON	Il bounding Box in Wgs84 che contiene l'intero strato informativo
description	Testo	Descrizione testuale dello strato informativo
vertices	Intero	Il numero totale di vertici nello strato informativo
count	Intero	Il numero di elementi nello strato informativo
type	Testo	Il tipo di geometrie contenuto in tutti gli elementi. Può essere: “MULTIPOINT”, “POINT”, “MULTILINESTRING”, “LINESTRING”, “MULTIPOLYGON”, “POLYGON”
levels	Array<Level>	Le informazioni sulle tile prodotte
attributes	Array<Attribute>	Le caratteristiche delle proprietà alfanumeriche dello strato informativo

Tabella 2: campi del documento “Strato”

Nome	Tipo	Descrizione
name	Testo	Il nome della proprietà così come nello shapefile
type	Testo	Il tipo della proprietà
unique	Integer	Il numero di valori distinti

Tabella 3: campi dell'oggetto Attribute nel documento “Strato”

Nome	Tipo	Descrizione
zoom	Integer	Il livello di zoom

tiles	Array<Integer>	i valori minimi e massimi delle coordinate Slippy Map Tilenames per le coordinate x ed y: [xmin,ymin,xmax,ymax]
-------	----------------	---

Tabella 4: campi dell'oggetto Level nel documento "Strato"

Documenti "Tile"

Nome	Tipo	Descrizione
Id	Testo	Identificatore univoco del tassello {layername}:{x}:{y}:{zoom}(:{page}) page è aggiunto solo se page è > 1
{layername}macrobbox{zoom}	GeoJSON	Presente solo nel caso sia una macro tile contiene l'area ricoperta dal tassello
bbox	GeoJSON	Presente solo nel caso sia un tassello semplice contiene l'area ricoperta dal tassello
page	integer	Numero di pagina rispetto all'insieme delle pagine del tassello
pages	integer	Numero pagine con i dati delle geometrie nel tassello
objs	Array[Element]	Contiene gli elementi o dati del layer nel tassello

Tabella 5: campi json nel documento "Tile"

Nome	Tipo	Descrizione
id	Testo	Identificatore univoco dell'elemento dello strato privato del prefisso con il nome dello strato
g	Testo	La rappresentazione della geometria così come descritto in tabella 1

Tabella 6: campi dell'oggetto Element nel documento "Tile"

Documenti "Infos"

Nome	Tipo	Descrizione
id	Testo	Identificatore univoco dell'elemento dello strato informativo con il prefisso del il nome dello strato
{layername}bbox	GeoJSON	contiene l'area ricoperta dall'elemento
attributes	Array[Attribute]	Contiene gli elementi o dati del layer nel tassello

Tabella 7: campi json nel documento "Infos"

Nome	Tipo	Descrizione
id	Testo	Identificatore univoco dell'elemento dello strato privato del prefisso con il

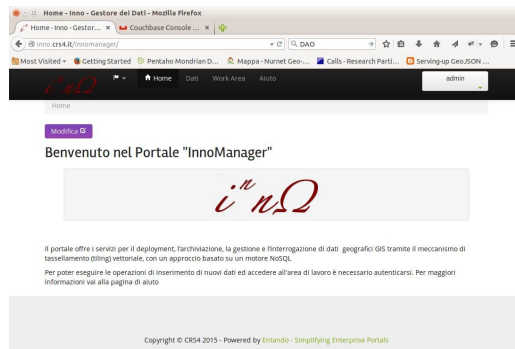


Figura 2: pagina iniziale del portale “innomanager”

che permette un accesso controllato ai dati gestiti nel bucket couchbase Inno e ne regola le modifiche e gli inserimenti oltre che le fasi di pre-trattamento sul database di lavoro.

L'applicazione web si chiama “Innomanager” (figura 2) ed assolve a tutti i requisiti indicati nel WP1 ma può essere facilmente estesa a funzionalità ulteriori attraverso la realizzazione di API per l'erogazione diretta dei tasselli o tramite la gestione delle chiamate alle API di couchbase, cioè sarebbe possibile aggiungere un pannello di monitoraggio delle operazioni sul bucket.

Il linguaggio usato per l'implementazione è il Java. L'applicazione è stata installata sul web container Tomcat nel server inno all'indirizzo <http://inno.crs4.it/innomanager>

L'applicazione web è un portale che segue la logica MVC (Model View Controll) propria di Entando. Il portale permette ad utenti di un gruppo particolare, chiamato “innouser”, di accedere ad un'area di lavoro che, tramite la connessione al database PostgreSQL di lavoro, elabora i documenti che poi saranno caricati, cioè pubblicati, sul database di Inno in Couchbase.

La gestione dei dati avviene tramite due Data Access Object (DAO), uno dei quali è collegato con il DBMS postgresQL in locale ed utilizzato per la gestione dei dati di lavoro, mentre l'altro è collegato con i nodi del cluster couchbase per la gestione dei dati pubblicati sul database Inno.

Il dato GIS da caricare su Inno è gestito tramite un oggetto rappresentante lo Strato Informativo di lavoro chiamato *WorkLayer* di proprietà di un unico utente (figura 3). Solo l'utente proprietario o l'amministratore del portale possono eseguire operazioni su di esso. Il Work Layer ha uno stato di elaborazione che è determinato dalla fase di pre-trattamento. Un utente del gruppo innouser può: caricare un WorkLayer nel portale tramite un operazione di upload; eseguire su di esso le fasi di pre-trattamento; visualizzare il log relativo alle elaborazioni eseguite o in corso; eseguire l'inserimento dei tasselli sul database Inno, cioè pubblicarli se il pre-trattamento ha dato esito positivo.

L'operazione di upload è particolarmente delicata. E' possibile scaricare solo shapefiles zippati, comprensivi dei file con estensione .shp, .dbf, .shx. E' possibile scaricare più di un file compresso per un singolo WorkLayer, questo per ovviare ai problemi di gestione upload dei file tramite http che non è performante quanto quello su sftp. Il limite consigliato sulle dimensioni di ciascun file scaricato è di 100 Mb ma il portale è stato testato con successo anche per file di dimensioni di circa 150 Mb. Ovviamente sarà a carico dell'utente suddividere i dati GIS con dimensioni superiori ai 100 MB in shapefiles distinti che una volta compressi siano di dimensioni inferiori ai 100 Mb.

Una volta nel sistema i file vengono caricati tramite l'utilità di sistema shp2pgsql e quindi caricati nel database PostgreSQL di lavoro. Di seguito l'utente può eseguire le altre fasi di

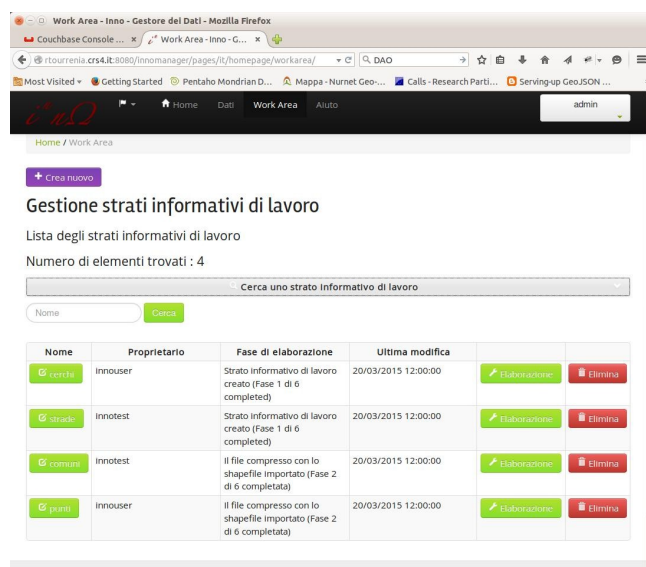


Figura 3: Area di lavoro

pre-trattamento. Ogni fase richiede un certo tempo che è relativamente breve per tutte le operazioni eccetto che per il tassellamento ai livelli di zoom 17 e 18. Per questo motivo è previsto un log ed una stima di elaborazione della fase di pre-trattamento che viene visualizzata ogni qual volta l'utente la richiede.

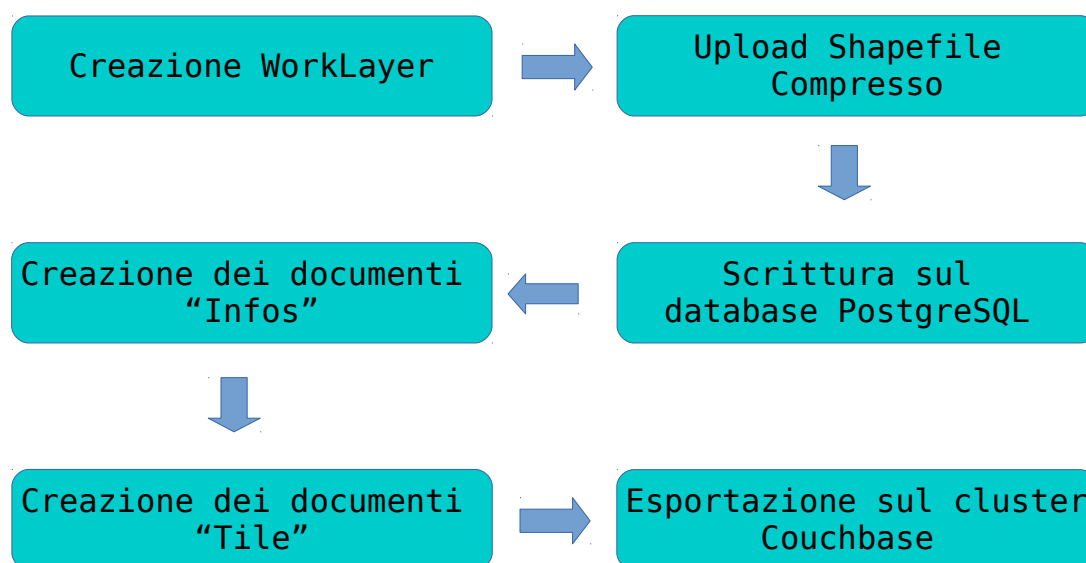


Figura 4: Fasi del pre-trattamento dei dati Gis

Finita la fase di pre-trattamento l'utente avrà un report delle operazioni effettuate e potrà effettuare l'inserimento dei dati sul database Inno. Tale operazione viene eseguita tramite il tool di Couchbase *cbdocloader*.

Per l'import dei dati si è preferito utilizzare l'utilità *shp2pgsql* e non la libreria Geotools, che può eseguire la stessa operazione, in tal modo viene velocizzato il caricamento e limitato l'utilizzo di risorse del portale. Lo stesso dicasi per il tool *cbdocloader* che potrebbe essere sostituito dal modulo DAO di connessione a Couchbase del portale.

I dati pubblici, cioè gli strati informativi provvisti di tasselli e presenti sul database Inno, sono altresì gestiti sul portale "innomanager". Un utente può visualizzare una preview dei dati appena inseriti (fig), verificandone i tasselli generati e le informazioni alfanumeriche restituite.

Al momento non sono previste operazioni CRUD sui documenti pubblicati da parte degli utenti ma solo da parte dell'amministratore del portale. E' possibile inoltre eseguire una operazione di cancellamento di tutti i documenti di un singolo strato tramite un'operazione eseguibile solo da parte dell'amministratore del portale. Una versione successiva potrebbe permettere la modifica delle informazioni alfanumeriche di singoli elementi di uno strato ma non delle geometrie.

6 Linee guida di utilizzo del prodotto sviluppato

Il portale è provvisto di pagine on line di aiuto consultabili ogni qual volta lo si ritenga necessario. Saranno comunque disponibili manuali in formato PDF scaricabili. Al momento la documentazione è solo in lingua italiana.

La parte più delicata, strettamente legata ai tempi di elaborazione, è la qualità del dato GIS che si vuole importare nel database Inno di couchbase.

Sono stati definite perciò delle linee guide relative alla creazione dello shapefile che poi sarà tassellato. Qualora il dato non le rispettasse nel caso peggiore l'elaborazione potrebbe essere sospesa oppure potrà escludere o modificare alcuni elementi. Le linee guida sono in realtà delle limitazioni e sono elencate di seguito assieme alla descrizione delle azioni intraprese dal sistema qualora non fossero rispettate.

- a) *Lo shapefile compresso di cui si esegue l'upload deve avere dimensione inferiore ai 150MB ma è consigliato come limite i 100MB***

In caso contrario il sistema non caricherà lo shapefile.

- b) *Il file con estensione dbf dello shapefile deve avere come codifica dei caratteri utf8***

In caso contrario il sistema non caricherà lo shapefile.

- c) *Le geometrie di uno strato informativo devono avere lo stesso tipo di geometria e il tipo di geometria può essere uno tra PUNTO, LINEA, POLIGONO, MULTI PUNTO, MULTI LINEA, MULTI POLIGONO***

In caso contrario il sistema non caricherà i dati

- d) *Le geometrie devono essere valide o semplici secondo le specifiche OGC (specifiche: <http://www.opengeospatial.org/standards/sfa>)***

Per i poligoni andare a pagina 25 e poi 31 del documento, per le linee a pagina 22 e 24.

In caso contrario l'elemento non valido sarà scartato.

- e) *Nel caso di geometrie poligonali non vi devono essere sovrapposizione, cioè non devono esistere due elementi aventi intersezione.***

In caso contrario l'intersezione genererà un errore nella rappresentazione e l'area intersecata sarà assegnata ad uno dei due poligoni

- f) *I nomi di attributi che cominciano con '_' non sono ammessi***

In caso contrario al nome sarà aggiunto un '_' al suo inizio

- g) *I nomi di attributi contenenti il carattere ':' non sono ammessi***

In caso contrario ogni occorrenza del carattere ':' sarà rimpiazzato da '_' o da '___'

- h) *Il nome di attributo "gid" non è ammesso***

In caso sia presente verrà sostituito con "_gid"

Appendici

Esempi di documenti JSON restituiti da inno

Lista dei strati informativi pubblici

```
{
  "layers": [
    {
      "_innoname_": "comuni",
      "_bbox_": {
        "type": "Polygon",
        "coordinates": [
          [
            [
              [8.130769799999999, 38.859124],
              [8.130769799999999, 41.3132405],
              [9.8270429, 41.3132405],
              [9.8270429, 38.859124],
              [8.130769799999999, 38.859124]
            ]
          ]
        ]
      },
      "vertices": 314610,
      "count": 377,
      "type": "MULTIPOLYGON",
      "attributes": [
        {
          "name": "gid",
          "type": "integer"
        },
        {
          "name": "istat",
          "type": "double precision"
        },
        {
          "name": "nome",
          "type": "character varying"
        },
        {
          "name": "regione",
          "type": "character varying"
        },
        {
          "name": "subregione",
          "type": "character varying"
        }
      ],
      "levels": [
        {
          "zoom": 10,
          "tiles": [535, 382, 539, 391]
        },
        {
          "zoom": 11,
          "tiles": [1070, 765, 1079, 783]
        },
        {
          "zoom": 12,
          "tiles": [2140, 1530, 2159, 1567]
        },
        {
          "zoom": 13,
          "tiles": [4281, 3061, 4319, 3134]
        },
        {
          "zoom": 14,
          "tiles": [8562, 6123, 8639, 6269]
        },
        {
          "zoom": 15,
          "tiles": [17124, 12247, 17278, 12539]
        },
        {
          "zoom": 16,
          "tiles": [34248, 24495, 34556, 25079]
        },
        {
          "zoom": 17,
          "tiles": [68496, 48990, 69113, 50158]
        }
      ]
    }
  ]
}
```

Tassello livello 10 coordinate 535, 391 strato "comuni"

```
[{"_bbox_": {"type": "Polygon", "coordinates": [[[8.08594, 38.82259], [8.08594, 39.09596], [8.4375, 39.09596], [8.4375, 38.82259], [8.08594, 38.82259]]]}}, {"objs": [{"id": "085", "g": "3((ff78ff79ff79ff79ff7aff7bfff7cff7cfe7cfd7cfd7dfc7efb7efb7ffa7ffa7ef97ef97df87df87ef77ef67ef67ff57ff47ff47ef37ef37df37ef27ef17ef17ff07ff17ff07ff080f180f080f081f082f083ef83ee83ee82ee83ee82ed82ec82ec81eb81ea81ea80ea7fe97fe97ee87ee87de77de77ce77be77ae779e679e779e778e777e778e877e777e776e876e875e775e774e874e773e772e872e772e771e772e671e570e470e471e470e370e36fe26fe26ee26de26ce16be16ae169e069e169e069e068e067e066e166e066e166e066e166e065e165e065e064e164e064e063e163e063e062df62de62de61de60dd60dd61dc61dc60dc5fdc5edc5ddc5edb5ddb5cdb5ddb5cdb5bda5bda5adb5bdb5ada5ad95ad959d958da58da57db56da56d957d956d957d956d856d956d955d855d854d954d953d853d854d853d753d853d953da53d953d952da52da51d951d851d751d651d650d750d74fd64fd74fd74ed64ed64fd54fd44fd44ed34ed44ed44dd34dd34cd34bd24bd24ad14ad149d148d048d047cf47d047d046d045d046d045d145d144d244d243d143d243d242d342d442d441d440d43fd33fd33ed43ed43dd53dd53cd63cd63bd73bd73ad739d738d638d637d636d736d834d833d933d932d931da2fdb2edc2cdd2cde29df28e027e026e125e124e123e222e31fe41de51ae617e616e615e811e90fe90ee90dea0aeb09ee09ee08ef06f004f101f100fc00fd00fe00ff00ff00ff78))"}], {"id": "158", "g": "3((9a009a00909009a00))", {"id": "158", "g": "3((9800980098019701960195009500940093009800))"}, {"id": "158", "g": "3((9800980098019701960195009500940093009800))"}, {"id": "158", "g": "3((8d008d008c008c018b018b008a008a01890189008800890088008d00))"}, {"id": "427", "g": "3((ecd9edd9eddaeedaeeedbeedceddcedcdceddbdddbdceadbdbdbdbdaebdb9ecd9))"}], {"id": "comuni:535:391:10", "page": 1, "pages": 1}
```

Valori di un attributo, livello 10 coordinate tassello 535, 391 strato
"comuni"

```
{
  "subregione": [
    {
      "id": "comuni:158",
      "v": "Sulcis-Iglesiente"
    },
    {
      "id": "comuni:249",
      "v": "Sulcis-Iglesiente"
    },
    {
      "id": "comuni:085",
      "v": "Sulcis-Iglesiente"
    },
    {
      "id": "comuni:427",
      "v": "Sulcis-Iglesiente"
    },
    {
      "id": "comuni:112",
      "v": "Sulcis-Iglesiente"
    }
  ],
  "id": "comuni:535:391:10"
}
```

Esempi dei documenti JSON gestiti su couchbase

Documento di tipo “strato”

```
{
  "_innoname_": "comuni",
  "_bbox_": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          8.1307698,
          38.859124
        ],
        [
          8.1307698,
          41.3132405
        ],
        [
          9.8270429,
          41.3132405
        ],
        [
          9.8270429,
          38.859124
        ],
        [
          8.1307698,
          38.859124
        ]
      ]
    ]
  },
  "vertices": 314610,
  "count": 377,
  "type": "MULTIPOLYGON",
  "attributes": [
    {
      "name": "gid",
      "type": "integer"
    },
    {
      "name": "istat",
      "type": "double precision"
    },
    {
      "name": "nome",
      "type": "character varying"
    },
    {
      "name": "regione",
      "type": "character varying"
    },
    {
      "name": "subregione",
      "type": "character varying"
    }
  ]
}
```

```

"levels": [
  {
    "zoom": 10, "tiles": [535,382,539,391]
  },
  {
    "zoom": 11, "tiles": [1070,765,1079,783]
  },
  {
    "zoom": 12, "tiles": [2140,1530,2159,1567]
  },
  {
    "zoom": 13, "tiles": [4281,3061,4319,3134]
  },
  {
    "zoom": 14, "tiles": [8562,6123,8639,6269]
  },
  {
    "zoom": 15, "tiles": [17124,12247,17278,12539]
  },
  {
    "zoom": 16, "tiles": [34248,24495,34556,25079]
  },
  {
    "zoom": 17, "tiles": [68496,48990,69113,50158]
  }
]
}

```

Documento di tipo “tile”

```

{
  "id": "comuni:535:391:10",
  "page": 1,
  "pages": 1,
  "_bbox_": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          8.08594,
          38.82259
        ],
        [
          8.08594,
          39.09596
        ],
        [
          8.4375
          39.09596
        ],
        [
          8.4375
          38.82259
        ],
        [
          8.08594,
          38.82259
        ]
      ]
    ]
  },
  "objs":
  [{
    "id": "085", "g": "3((ff78ff79ff79ff79ff79ff7aff7bfff7cfff7cfe7cfd7cfd7dfc7efb7efb7ffa7ffa7ef97ef97df87df87ef77ef67ef67ff57ff47ff47ef37ef37df37ef27ef17ef17ff07ff17ff07ff080f180f080f081f082f083ef83ee83ee82ee83ee82ed82ec82ec81eb81ea81ea80ea7fe97fe97ee87ee87de77de77ce77be77ae779e679e779e778e777e778e877e777e776e876e875e775e774e874e773e772e872e772e771e772e671e570e470e471e470e370e36fe26fe26ee26de26ce16be16ae169e069e169e069e068e067e066e166e066e166e066e166e065e165e065e064e164e064e063e163e063e062df62de62de61de60dd60dd61dc61dc60dc5fdc5edc5ddc5edb5ddb5cdb5ddb5bdb5bda5bdb5adb5bdb5ada5ad95ad959d958da58da57db56da56d957d956d957d956d856d956d955d855d854d954d953d853d854d853d753d853d953da53d953d952da52da51d951d851d751d651d650d750d74fd64fd74fd74ed64ed64fd54fd44fd44ed34ed44ed44dd34dd34cd34bd24bd24ad14ad149d148d048d047cf47d047d046d045d046d045d145d144d244d243d143d243d242d342d442d441d440d43fd33fd33ed43ed43dd53dd53cd63cd63bd73bd73ad739d738d638d637d636d736d834d833d933d932d931da2fdb2edc2cdd2cde29df28e027e026e125e124e123e222e31fe41de51ae617e616e615e811e90fe90ee90dea0aeb09ee09ee08ef06f004f101f100fc00fd00fe00ff00ff00ff78)))", {"id": "158",

```

```

"gid": "3((9a009a0099009a00))", {"id": "158",
"gid": "3((9800980098019701960195009500940093009800))", {"id": "158",
"gid": "3((8d008d008c008c018b018b008a008a01890189008800890088008d00))",
{"id": "427",
"gid": "3((ecd9edd9eddaeedaeedbeedceddcecdcecddebddebdceadbdbdbdaebd9ecd9))"}],
"id": "comuni:535:391:10", "page": 1, "pages": 1}

```

Documento di tipo “tile” con una macro-tile

```

{
  "id": "comuni:4282+0:3078+1:13",
  "page": 1,
  "pages": 1,
  "_comunimacro13bbox_": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          8.17383,
          40.71396
        ],
        [
          8.17383,
          40.78054
        ],
        [
          8.21777,
          40.78054
        ],
        [
          8.21777,
          40.71396
        ],
        [
          8.17383,
          40.71396
        ]
      ]
    ]
  },
  "objs": [{ "id": "001", "gid": "3((ff00ffff00ff0000ff00))" }]
}

```

Documento di tipo “infos”

```

{
  "_comunibbox_": {
    "type": "Polygon",
    "coordinates": [
      [[8.13077,40.65175],[8.13077,40.8693],[8.6436,40.8693],
      [8.6436,40.65175],[8.13077,40.65175]]]
    ],
  "gid": 128,
  "istat": 90064,
  "nome": "Sassari",
  "regione": "Sardegna",
  "subregione": "Sassarese",
  "id": "001"}
}

```