

1 Fase (8 giornate)

1. Definizione del diagramma E/R e delle specifiche relative per la creazione dell'infrastruttura DB (1 giornata) – **in sede ISPRA** (già realizzata) – **21 marzo**
 2. Installazione di PostGIS, definizione di un ambiente di lavoro, creazione delle tabelle ritenute necessarie per il progetto. Test funzionale. (1/2 giornata) - **9 maggio**
 3. Installazione di Geoserver, definizione dei layer e integrazione delle tabelle. Test funzionale. (1/2 giornata) **9 maggio**
TEST ON LINE E ATTESTATO 2 GIORNATE:12 ORE
 - 9 Importazione dei dati da file Shape o altri db georeferenziati per la costruzione degli elementi verdi e blu (1 e ½ giornate) - **15 maggio, 22 maggio**
 - 10 Importazione dei dati della popolazione su raster fornito da ISPRA (1 e ½ giornate) – **22 maggio, 29 maggio**
 - 11 Installazione ambiente di lavoro usando Eclipse, definizione classi per accesso DB (2 giornate) – **12 giugno, 19 giugno**
 - 12 Confronto su sviluppo del geodatabase - (1 giornata) - in sede ISPRA – **24 giugno**

TEST ON LINE E ATTESTATO 6 GIORNATE: 36 ORE

2 fase (24 giornate)

1. Calcolo dei punti di ingresso delle aree verdi e blu (4 giornate) – **3 luglio, 10 luglio, 17 luglio, 2 settembre**
 2. Calcolo della distanza minima fra i centroidi dei poligoni rappresentanti la popolazione e i punti di ingresso delle aree verdi e blu (2 giornate) **4 settembre, 9 settembre**

TEST ON LINE E ATTESTATO 6 GIORNATE: 36 ORE



TOTJ



Administrator



vebs



Questo PC



Progetto_VeBS



Rete



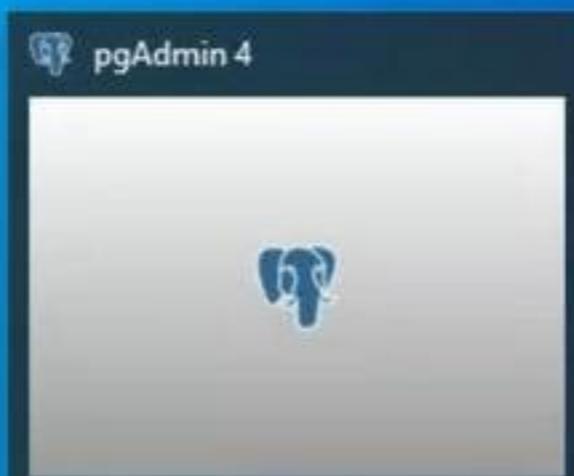
DATA



Gestino



QGIS 3.42.2



Postman

Attiva Windows

Passa a Impostazioni per attivare
Windows.

Scrivi qui il testo da cercare.

09:09
12/06/2025

1 Fase (8 giornate)

1. Definizione del diagramma E/R e delle specifiche relative per la creazione dell'infrastruttura DB (1 giornata) – **in sede ISPRA** (già realizzata) – **21 marzo**
2. Installazione di PostGIS, definizione di un ambiente di lavoro, creazione delle tabelle ritenute necessarie per il progetto. Test funzionale. (1/2 giornata) -**9 maggio**
3. Installazione di Geoserver, definizione dei layer e integrazione delle tabelle. Test funzionale. (1/2 giornata) **9 maggio**
TEST ON LINE E ATTESTATO 2 GIORNATE:12 ORE
- 9 Importazione dei dati da file Shape o altri db georeferenziati per la costruzione degli elementi verdi e blu (1 e ½ giornate) - **15 maggio, 22 maggio**
- 10 Importazione dei dati della popolazione su raster fornito da ISPRA (1 e ½ giornate) – **22 maggio, 29 maggio**
- 11 Installazione ambiente di lavoro usando Eclipse, definizione classi per accesso DB (2 giornate) – **12 giugno, 19 giugno**
- 12 Confronto su sviluppo del geodatabase - (1 giornata) - in sede ISPRA – **24 giugno**
TEST ON LINE E ATTESTATO 6 GIORNATE: 36 ORE

2 fase (24 giornate)

1. Calcolo dei punti di ingresso delle aree verdi e blu (4 giornate) – **3 luglio, 10 luglio, 17 luglio, 2 settembre**
2. Calcolo della distanza minima fra i centroidi dei poligoni rappresentanti la popolazione e i punti di ingresso delle aree verdi e blu (2 giornate) **4 settembre, 9 settembre**
TEST ON LINE E ATTESTATO 6 GIORNATE: 36 ORE

Gestione Database e Pubblicazione Geospatiale con PostGIS e GeoServer

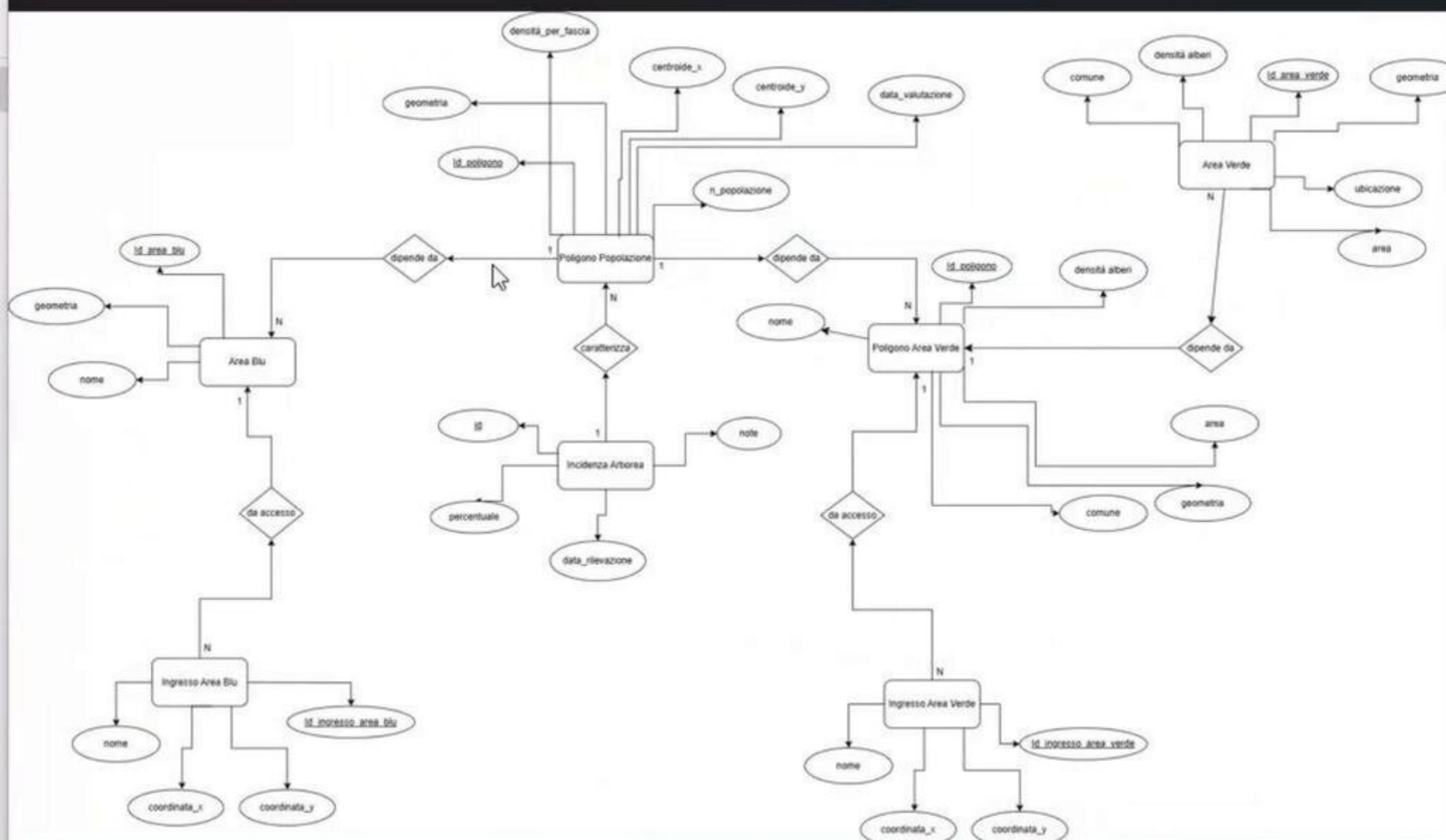
1. Definizione del Diagramma E/R e Specifiche per la Creazione dell'Infrastruttura DB

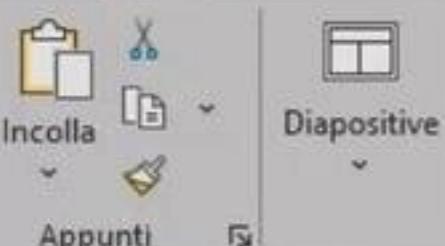
Obiettivi:

- Comprendere la struttura logica dei dati
 - Definire le entità, relazioni e attributi
 - Produrre un diagramma E/R coerente con i requisiti del progetto
 - Redigere specifiche tecniche per la realizzazione fisica del database

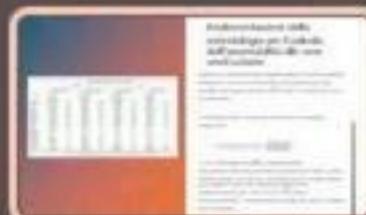
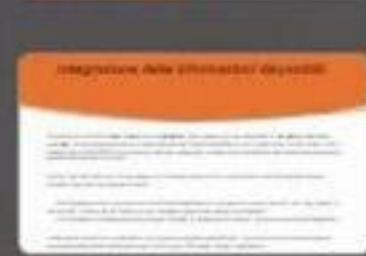
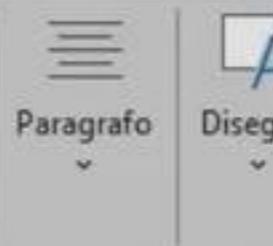
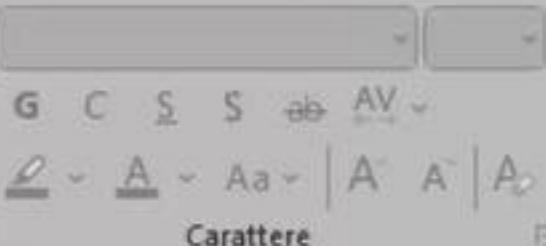
Attività previste:

1. Analisi dei requisiti informativi del progetto
 2. Identificazione delle entità principali (es. aree verdi, aree blu, poligono per popolazione, ingressi per aree)
 3. Definizione delle relazioni (es. un'area verde può avere molti ingressi)
 4. Rappresentazione E/R con strumenti visuali (es. [draw.io](#))
 5. Produzione di uno schema relazionale





Diapositive



MC	OSM				UA			
	Within 300 m		Within 600 m		Within 300 m		Within 600 m	
	Number of habit.	%						
Population That Has Access to a GLP%.								
Torino	980,440	37.9	859,799	33.9	279,268	17.9	420,980	27
Genova	124,225	14.1	192,809	29.5	103,234	15.8	159,362	24.4
Milan	1,484,951	49.4	1,876,542	66.4	891,384	29.9	1,283,204	43.3
Venice	514,030	28.3	555,279	42.3	259,444	20.6	713,489	21.3
Bologna	216,255	56.6	413,987	73.2	237,860	28.7	284,489	51.4
Florence	247,748	39.6	342,600	55.5	143,729	36.2	256,386	36.2
Rome	994,034	24.4	1,427,314	42.2	736,062	21.8	1,051,376	30.3
Naples	333,286	12.3	514,773	19	444,905	17.5	677,308	25
Bari	183,773	14.8	306,033	24.9	94,421	5.5	164,897	8.4
Bergamo	113,958	6.5	25,334	11	4809	2.7	10,101	4
Cagliari	65,953	8.9	133,860	15.4	58,718	12.5	166,365	19.2
Palermo	116,836	7	32,444	11.4	-	-	-	-
Catania	78,905	7.3	96,027	12.3	35,929	4.2	54,350	6.8
Cagliari	67,043	10.1	107,380	30.6	77,098	22	117,362	33.4
Total	4,639,889	28.2	6,374,084	40	3,222,058	19.8	6,670,153	28.3

(%) Percentage of population that has access, compared to the total population of the urban area.

Fare clic per inserire le note

Implementazione della metodologia per il calcolo dell'accessibilità alle aree verdi urbane

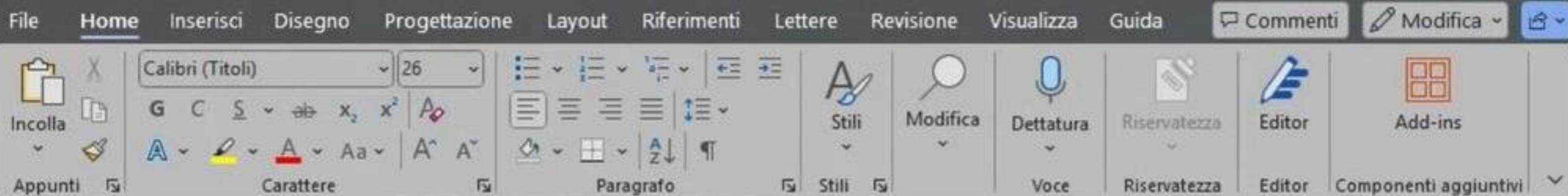
Nota la classificazione delle aree in funzione della distanza, occorre valutare la popolazione che ricade nell'area posta a 300 metri rispetto ai punti di accesso.

L'accessibilità verrebbe calcolata nel modo seguente:

$$\text{Accessibilità area verde} = \frac{\text{Popolazione}_{300}}{\text{Popolazione}_T}$$

In cui Popolazione_{300} = popolazione rappresentata dagli abitanti localizzati nelle celle della griglia che hanno accesso ad un'area verde se quest'ultima può essere raggiunta camminando per non più di 300 metri;

Popolazione_T = popolazione totale all'interno della municipalità



- Spiegazione e sviluppo delle classi Repository
- Spiegazione e sviluppo delle classi di configurazione
- Spiegazione e sviluppo delle classi Service
- Spiegazione e sviluppo della classe Application
- Spiegazione e sviluppo delle classi Controller

Spring MVC:

Spring MVC è un framework per realizzare applicazioni web basate sul modello MVC sfruttando i punti di forza offerti dal framework Spring come l'inversion of control (tramite dependency injection) e la aspect oriented programming.

Esso si occupa di mappare metodi e classi Java con determinati url, di gestire differenti tipologie di "viste" restituite al client, di realizzare applicazioni internazionalizzate e di gestire i cosiddetti temi per personalizzare al massimo l'esperienza utente.

Cos'è il pattern MVC?

I

Per comprendere al meglio il framework è necessario introdurre il pattern teorico che esso implementa ovvero il modello MVC. MVC rappresenta un acronimo per Model View

ovvero le tre componenti principali di un'applicazione web. Grazie a questo pattern i compiti vengono separati verso questi componenti:

- i **Model** si occupano di accedere ai dati necessari alla logica di business implementata nell'applicazione (nel caso di un'applicazione per una biblioteca potrebbero essere le classi Libro, Autore, Scaffale);
 - le **View** si occupano di creare l'interfaccia utilizzabile dall'utente e che espone i dati da esso richiesti (nel caso bibliotecario potrebbero essere le pagine HTML del catalogo, le **form** di ricerca oppure i PDF contenenti ricerche o appunti);
 - i **Controller** si occupano di implementare la vera logica di business dell'applicazione integrando le due componenti precedenti, ricevendo gli input dell'utente, gestendo i modelli per la ricerca dei dati e la creazione di viste da restituire all'utente (nel nostro caso potrebbero essere il motore di ricerca interno oppure il sistema di login al sito Internet della biblioteca).

Come funziona MVC in Spring?

Spring MVC implementa perfettamente questo approccio mantenendo sia i concetti che la nomenclatura del pattern. All'interno di una applicazione Spring MVC avremo quindi: i **Model** sono rappresentati dalle classi che a loro volta rappresentano gli oggetti gestiti e le **classi di accesso al database**:

le **View** sono rappresentate dai vari **file Javascript** (che vengono compilati in HTML) e da eventuali classi per l'esportazione in formati diversi da HTML (PDF, XLS, CSV...);

i **Controller** sono rappresentati da classi (chiamate appositamente **Controller**) che rimangono "in ascolto" su un determinato URL e, grazie ai **Model** e alle View, si occupano di gestire la richiesta dell'utente.

Salvataggio automatico H Introduzione a Spring + configurazione inizi... • Salvato in questo PC - □ X

File Home Inserisci Disegno Progettazione Layout Riferimenti Lettere Revisione Visualizza Guida Commenti Modifica

Calibri (Titoli) 26

G C S α x_2 x^2 A

A A Aa A^ A~

Carattere

Paragrafo

Stili Modifica Dettatura Riservatezza

Voce Riservatezza

Editor Componenti aggiuntivi

Internet della biblioteca).

Come funziona MVC in Spring?

Spring MVC implementa perfettamente questo approccio mantenendo sia i concetti che la nomenclatura del pattern. All'interno di una applicazione Spring MVC avremo quindi:

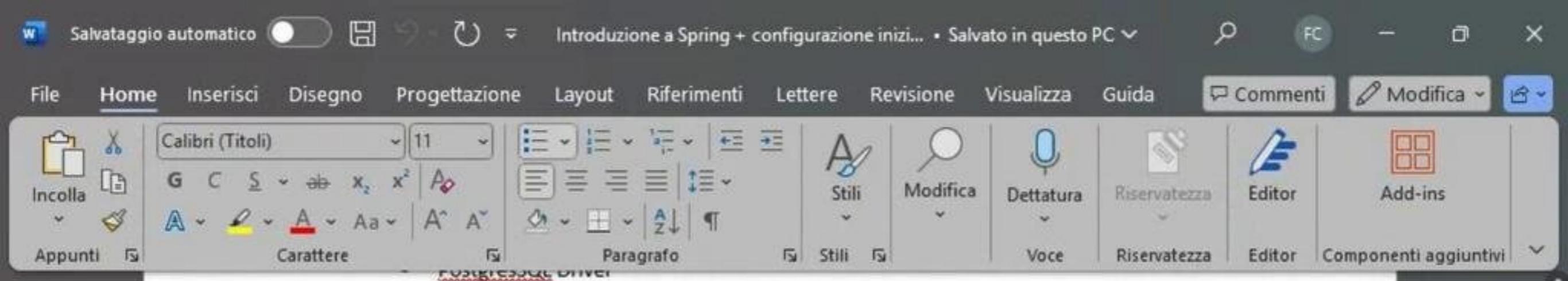
- i **Model** sono rappresentati dalle classi che a loro volta rappresentano gli oggetti gestiti e le **classi di accesso al database**;
- le **View** sono rappresentate dai vari **file Javascript** (che vengono compilati in HTML) e da eventuali classi per l'esportazione in formati diversi da HTML (PDF, XLS, CSV...);
- i **Controller** sono rappresentati da classi (chiamate appositamente **Controller**) che rimangono "in ascolto" su un determinato URL e, grazie ai Model e alle **View**, si occupano di gestire la richiesta dell'utente.

Secondo la documentazione ufficiale Spring MVC presenta inoltre molti altri vantaggi oltre alla netta separazione tra le funzionalità:

- è adattabile, flessibile e non intrusivo grazie alla presenza di comode e chiare Java **Annotations**;
- permette di scrivere codice riusabile;
- possibilità di essere esteso tramite adattatori e validatori scritti ad hoc per le nostre esigenze;
- **url** dinamici, SEO-friendly e personalizzabili;
- gestione integrata dell'internazionalizzazione e dei temi;
- libreria JSP sviluppata ad hoc per facilitare alcune operazioni ripetitive;
- nuovi scope per i **bean** (**request** e **session**) che permettono di adattare i container base di Spring anche al mondo web.

Pagina 2 di 7 1386 parole Completamento del testo: attivato Accessibilità: conforme Focus

09:22 12/06/2025



Maven:

Maven è un progetto open source, sviluppato dalla Apache, che permette di organizzare in modo molto efficiente un progetto java.

Può essere paragonato all'altro progetto più conosciuto della Apache, Ant, ma fornisce funzionalità più avanzate. I vantaggi principali di Maven sono i seguenti:

- standardizzazione della struttura di un progetto compilazione;
- test ed esportazione automatizzate;
- gestione e download automatico delle librerie necessarie al progetto;
- creazione automatica di un semplice sito di gestione del progetto contenente informazioni.

Successivamente andremo ad importare attraverso il file pom.xml automaticamente generato le seguenti dipendenze (<https://mvnrepository.com/repos/central>):

- Jackson (per serializzare e deserializzare i dati JSON utile con dati geografici)
- Geotools (libreria per gestire dati GIS)
- JTS Core (libreria per lavorare con le geometrie)

Le classi Entity:

- **Definizione:** Classe che rappresenta una tabella nel database.
- **Annotazione principale:** @Entity
- **Responsabilità:** Mappare i dati tra il database e l'applicazione.
- **Esempio:**

Spiegazione strumenti OpenStreetMap e Overpass API e continuazione sviluppi

Obiettivi:

- Estrazione dati e continuazione sviluppi backend

Attività previste:

- Introduzione all'utilizzo di OpenStreetMap
- Approfondimento dell'uso dell'Overpass API
- Continuazione sviluppo delle classi per le API REST e dei relativi servizi

Introduzione all'utilizzo di OpenStreetMap



OpenStreetMap è un progetto collaborativo per creare una **mappa libera del mondo**, accessibile e modificabile da chiunque, un po' come Wikipedia per i dati geografici.

- **Fondato nel 2004**, OSM raccoglie dati geografici come strade, edifici, fiumi, parchi, punti di interesse, ecc.
- I dati sono **open data**, rilasciati con licenza **ODbL (Open Database License)**: cioè si possono usare, modificare, integrare anche in progetti commerciali, citando la fonte.

Salvataggio automatico

Overpass_OpenStreetMap_API_VEBS.docx -... • Salvato in questo PC

File Home Inserisci Disegno Progettazione Layout Riferimenti Lettere Revisione Visualizza Guida

Commenti

VISUALIZZAZIONE PROTETTA

Attenzione. I file provenienti da Internet possono contenere virus. A meno che non sia necessario modificare il file, è consigliabile restare in Visualizzazione protetta.

Abilita modifica

- Approfondimento dell'uso dell'Overpass API

- Continuazione sviluppo delle classi per le API REST e dei relativi servizi

Introduzione all'utilizzo di OpenStreetMap

OpenStreetMap è un progetto collaborativo per creare una **mappa libera del mondo**, accessibile e modificabile da chiunque, un po' come Wikipedia per i dati geografici.

- **Fondato nel 2004**, OSM raccoglie dati geografici come strade, edifici, fiumi, parchi, punti di interesse, ecc.
- I dati sono **open data**, rilasciati con licenza **ODbL (Open Database License)**: cioè si possono usare, modificare, integrare anche in progetti commerciali, citando la fonte.

OSM è basato su tre elementi principali:

Nodo (node) Un punto con coordinate (es. una fermata autobus, un ingresso di un parco)

Via (way) Una linea o poligono composto da nodi (es. una strada o un edificio)

Relazione (relation) Struttura per collegare oggetti (es. un itinerario bus con più vie)

Ogni oggetto può avere tag del tipo **chiave = valore**, ad esempio:

highway=primary

name=Via Roma

Pagina 1 di 7 2 di 1055 parole Completamento del testo attivato

Focus

100%

09:45

12/06/2025

OSM è basato su tre elementi principali:

Nodo (node) Un punto con coordinate (es. una fermata autobus, un ingresso di un parco)

Via (way) Una linea o poligono composto da nodi (es. una strada o un edificio)

Relazione (relation) Struttura per collegare oggetti (es. un itinerario bus con più vie)

Ogni oggetto può avere tag del tipo **chiave = valore**, ad esempio:

highway=primary

name=Via Roma

I

I dati vengono salvati in un **database globale** che chiunque può esplorare, interrogare, aggiornare o estrarre.

Cosa è possibile fare su OpenStreetMap

1. Esplorazione e visualizzazione

- Vai su openstreetmap.org
 - Puoi cercare luoghi, esplorare i dati mappa e vedere i dettagli di ogni oggetto (tag, cronologia modifiche, utenti, ecc.)

2. Modificare dati

Si può contribuire a OSM (da utente registrato) con:

- **iD editor:** editor online facile da usare, direttamente dal sito OSM
 - **JOSM:** editor avanzato desktop (Java) per modifiche massive
 - **Field Papers** o app mobile (Vespucci, StreetComplete)

3. Scaricare dati

Come ottenere dati OSM:

- In formato .osm (XML) tramite Geofabrik
 - In formato GeoJSON, KML usando [Overpass Turbo](#)
 - Con strumenti GIS come QGIS (tramite plugin QuickOSM)

4. Sviluppo software

È possibile integrare OSM in app/web usando:

VISUALIZZAZIONE PROTETTA

Attenzione. I file provenienti da Internet possono contenere virus. A meno che non sia necessario modificare il file, è consigliabile restare in Visualizzazione protetta.

Abilita modifica

- Puoi cercare luoghi, esplorare i dati mappa e vedere i dettagli di ogni oggetto (tag, cronologia modifiche, utenti, ecc.)

2. Modificare dati

Si può contribuire a OSM (da utente registrato) con:

- **iD editor:** editor online facile da usare, direttamente dal sito OSM
- **JOSM:** editor avanzato desktop (Java) per modifiche massive
- **Field Papers** o app mobile (Vespucci, StreetComplete)

3. Scaricare dati



Come ottenere dati OSM:

- In **formato .osm (XML)** tramite Geofabrik
- In **formato GeoJSON, KML** usando [Overpass Turbo](#)
- Con strumenti GIS come QGIS (tramite plugin QuickOSM)

4. Sviluppo software

È possibile integrare OSM in app/web usando:

- **Leaflet.js + Tile Server:** per visualizzare mappe
- **OpenLayers:** libreria GIS frontend potente
- **osmnx (Python):** per analisi di reti stradali
- **Nominatim API:** per geocoding/reverse geocoding

Cosa è possibile fare su OpenStreetMap

1. Esplorazione e visualizzazione

- Vai su openstreetmap.org
- Puoi cercare luoghi, esplorare i dati mappa e vedere i dettagli di ogni oggetto (tag, cronologia modifiche, utenti, ecc.)

2. Modificare dati

Si può contribuire a OSM (da utente registrato) con:

- **iD editor:** editor online facile da usare, direttamente dal sito OSM
- **JOSM:** editor avanzato desktop (Java) per modifiche massive
- **Field Papers** o app mobile (Vespucci, StreetComplete)

3. Scaricare dati

Come ottenere dati OSM:

- In **formato .osm (XML)** tramite Geofabrik
- In **formato GeoJSON, KML** usando [Overpass Turbo](#)
- Con strumenti GIS come QGIS (tramite plugin QuickOSM)

4. Sviluppo software

È possibile integrare OSM in app/web usando:

File Home Inserisci Disegno Progettazione Layout Riferimenti Lettere Revisione Visualizza Guida Commenti FC X

VISUALIZZAZIONE PROTETTA *Attenzione, i file provenienti da Internet possono contenere virus. A meno che non sia necessario modificare il file, è consigliabile restare in Visualizzazione protetta.* Abilita modifica X

- Puoi cercare luoghi, esplorare i dati mappa e vedere i dettagli di ogni oggetto (tag, cronologia modifiche, utenti, ecc.)

2. Modificare dati

Si può contribuire a OSM (da utente registrato) con:

- **iD editor:** editor online facile da usare, direttamente dal sito OSM
- **JOSM:** editor avanzato desktop (Java) per modifiche massive
- **Field Papers** o app mobile (Vespucci, StreetComplete)

3. Scaricare dati

Come ottenere dati OSM:

- In **formato .osm (XML)** tramite Geofabrik
- In **formato GeoJSON, KML** usando [Overpass Turbo](#)
- Con strumenti GIS come QGIS (tramite plugin QuickOSM)

4. Sviluppo software

È possibile integrare OSM in app/web usando:

- **Leaflet.js + Tile Server:** per visualizzare mappe
- **OpenLayers:** libreria GIS frontend potente
- **osmnx (Python):** per analisi di reti stradali
- **Nominatim API:** per geocoding/reverse geocoding

Pagina 2 di 7 2 di 1055 parole Focus 1 100% 09:50 12/06/2025



VISUALIZZAZIONE PROTETTA

Attenzione, i file provenienti da Internet possono contenere virus. A meno che non sia necessario modificare il file, è consigliabile restare in Visualizzazione protetta.

Abilita modifica

Si può contribuire a OSM (da utente registrato) con:

- **iD editor:** editor online facile da usare, direttamente dal sito OSM
- **JOSM:** editor avanzato desktop (Java) per modifiche massive
- **Field Papers** o app mobile (Vespucci, StreetComplete)

3. Scaricare dati

Come ottenere dati OSM:

- In formato **.osm (XML)** tramite [Geotools](https://overpass-turbo.eu/) https://overpass-turbo.eu/
CTRL+clic per aprire collegam.
- In formato **GeoJSON, KML** usando [Overpass Turbo](#)
- Con strumenti GIS come QGIS (tramite plugin QuickOSM)

4. Sviluppo software

È possibile integrare OSM in app/web usando:

- **Leaflet.js + Tile Server:** per visualizzare mappe
- **OpenLayers:** libreria GIS frontend potente
- **osmnx (Python):** per analisi di reti stradali
- **Nominatim API:** per geocoding/reverse geocoding
- **Overpass API:** per query personalizzate sui dati
- **PostGIS:** puoi importare i dati OSM nel tuo DB spaziale per analisi avanzate

VISUALIZZAZIONE PROTETTA

Attenzione, i file provenienti da Internet possono contenere virus. A meno che non sia necessario modificare il file, è consigliabile restare in Visualizzazione protetta.

E' possibile integrare OSM in app/web usando:

- **Leaflet.js + Tile Server**: per visualizzare mappe
- **OpenLayers**: libreria GIS frontend potente
- **osmnx (Python)**: per analisi di reti stradali
- **Nominatim API**: per geocoding/reverse geocoding
- **Overpass API**: per query personalizzate sui dati
- **PostGIS**: puoi importare i dati OSM nel tuo DB spaziale per analisi avanzate

I

Approfondimento dell'uso dell'Overpass API

Overpass API è un'API che consente di **interrogare i dati di OpenStreetMap (OSM)** in modo avanzato e personalizzato.

È progettata per estrarre *solo i dati che servono*, senza dover scaricare l'intero database OSM.

Approfondimento dell'uso dell'Overpass API

Overpass API è un'API che consente di interrogare i dati di OpenStreetMap (OSM) in modo avanzato e personalizzato.

È progettata per estrarre solo i dati che servono, senza dover scaricare l'intero database OSM.

Non è un'API REST "standard" tipo GET /endpoint, ma un motore di query geospatiali su un database OSM.

Funzionalità di Overpass Turbo:

- - Creazione ed esecuzione di query Overpass QL
 - - Visualizzazione su mappa degli oggetti ottenuti
 - - Esportazione in vari formati (GeoJSON, KML, GPX, ecc.)

Overpass API lavora su una copia aggiornata del database OSM e consente di interrogarlo tramite due linguaggi:

1. Overpass QL (query language compatto)
 2. Overpass XML (più verboso, in formato XML)

Concetti base

- Gli oggetti OSM sono: node, way, relation.
 - Ogni oggetto ha tag del tipo chiave=valore (es: amenity=school).

Salvataggio automatico: H 9:00 ▾ Overpass_OpenStreetMap_API_VEBS.docx -... • Salvato in questo PC ▾ FC - X

File Home Inserisci Disegno Progettazione Layout Riferimenti Lettere Revisione Visualizza Guida Commenti

Approfondimento dell'uso dell'Overpass API

Overpass API è un'API che consente di interrogare i dati di OpenStreetMap (OSM) in modo avanzato e personalizzato.

È progettata per estrarre *solo i dati che servono*, senza dover scaricare l'intero database OSM.

Non è un'API REST "standard" tipo GET /endpoint, ma un motore di query geospatiali su un database OSM.

Funzionalità di Overpass Turbo:

- - Creazione ed esecuzione di query Overpass QL
- - Visualizzazione su mappa degli oggetti ottenuti
- - Esportazione in vari formati (GeoJSON, KML, GPX, ecc.)

Overpass API lavora su una copia aggiornata del database OSM e consente di interrogarlo tramite due linguaggi: 

1. Overpass QL (query language compatto)
2. Overpass XML (più verboso, in formato XML)

Concetti base

- Gli oggetti OSM sono: node, way, relation.
- Ogni oggetto ha tag del tipo chiave=valore (es: amenity=school).
- Si può filtrare per:

Pagina 3 di 7 1055 parole Completamento del testo: attivato Focus

100% 09:56 12/06/2025

Windows Taskbar icons: File Explorer, Edge, Google Chrome, File, Powerpoint, Word, and a search icon.

Overpass API lavora su una copia aggiornata del database OSM e consente di interrogarlo tramite due linguaggi:

1. Overpass QL (query language compatto)
 2. Overpass XML (più verboso, in formato XML)

Concetti base

- Gli oggetti OSM sono: node, way, relation.
 - Ogni oggetto ha tag del tipo chiave=valore (es: amenity=school).
 - Si può filtrare per:
 - Tag (es: tutti i parchi)
|
 - Area geografica (box lat/lon)
 - Tipo di oggetto (nodo/poligono)
 - Tempo (modifiche recenti)

Esempio di Query in Overpass QL

```
[out:json][timeout:25];
// seleziona tutti i parchi in un'area o specificando le coordinate dell'area
```

```
//es: " (45.42,10.95,45.45,10.98)"  
(  
  node["leisure"="park"] (45.42,10.95,45.45,10.98);  
  way["leisure"="park"] (45.42,10.95,45.45,10.98);  
  relation["leisure"="park"] (45.42,10.95,45.45,10.98);  
);  
  
[out:json][timeout:25];  
//o passandogli il nome es:"Catanzaro"  
area[!name"="Catanzaro"]["boundary"="administrative"]["admin_level"="8"]->.catanzaro;  
(  
  node["leisure"="park"] (area.catanzaro);  
  way["leisure"="park"] (area.catanzaro);  
  relation["leisure"="park"] (area.catanzaro);  
)->.parchi;  
  
// trova i nodi con barrier=gate o entrance nelle aree verdi (via map_to_area)  
.parchi map_to_area -> .aree_parco;  
  
(  
  node["barrier"="gate"] (area.aree_parco);  
  node["barrier"="entrance"] (area.aree_parco);  
);  
  
out body;  
>;  
out skel qt;  
// oppure out geom;
```

Spiegazione:

- out:json: formato di output (può essere anche XML)

```
// trova i nodi con barrier=gate o entrance nelle aree verdi (via map_to_area)
.parchi map_to_area -> .aree_parco;

(
    node["barrier"="gate"](.aree_parco);
    node["barrier"="entrance"](.aree_parco);
);

out body;
>;
out skel qt;
// oppure out geom;
```

[

Spiegazione:

- out:json: formato di output (può essere anche XML)
- timeout:25: tempo massimo di esecuzione in secondi
- node, way, relation: tipi di oggetti cercati
- (lat1, lon1, lat2, lon2): bounding box
- out body: restituisce i dati principali
- >; segue le referenze (es. nodi delle way)
- out skel qt; output compatto di chiusura
- out geom; output integrale che contiene le relazioni con nodi e figli delle ways

Salvataggio automatico: H 9 - Overpass_OpenStreetMap_API_VEBS.docx -... • Salvato in questo PC ▾

File Home Inserisci Disegno Progettazione Layout Riferimenti Lettere Revisione Visualizza Guida Commenti

```
way["leisure"="park"](45.42,10.95,45.45,10.98);
relation["leisure"="park"](45.42,10.95,45.45,10.98);
);

[out:json][timeout:25];
//o passandogli il nome es:"Catanzaro"
area["name"="Catanzaro"]["boundary"="administrative"]["admin_level"="8"]->.catanzaro;
(
  node["leisure"="park"](area.catanzaro);
  way["leisure"="park"](area.catanzaro);
  relation["leisure"="park"](area.catanzaro);
) ->.parchi;

// trova i nodi con barrier=gate o entrance nelle aree verdi (via map_to_area)
.parchi map_to_area ->.aree_parco;

(
  node["barrier"="gate"](area.aree_parco);
  node["barrier"="entrance"](area.aree_parco);
);
  I
out body;
>;
out skel qt;
// oppure out geom;
```

Spiegazione:

- `out:json`: formato di output (può essere anche XML)
- `timeout:25`: tempo massimo di esecuzione in secondi
- `node, way, relation`: tipi di oggetti cercati

Pagina 4 di 7 1 di 1055 parole Completamento del testo: attivato Focus 10:11 12/06/2025

```
// trova i nodi con barrier=gate o entrance nelle aree verdi (via map_to_area)
.parchi map_to_area -> .aree_parco;

(
    node["barrier"="gate"](.aree_parco);
    node["barrier"="entrance"](.aree_parco);
);
I
out body;
>;
out skel qt;
// oppure out geom;
```

Spiegazione:

- out:json: formato di output (può essere anche XML)
- timeout:25: tempo massimo di esecuzione in secondi
- node, way, relation: tipi di oggetti cercati
- (lat1, lon1, lat2, lon2): bounding box
- out body: restituisce i dati principali
- >; segue le referenze (es. nodi delle way)
- out skel qt; output compatto di chiusura
- out geom; output integrale che contiene le relazioni con nodi e figli delle ways

```

//es:" (45.42,10.95,45.45,10.98)"
(
  node["leisure"="park"](45.42,10.95,45.45,10.98);
  way["leisure"="park"](45.42,10.95,45.45,10.98);
  relation["leisure"="park"](45.42,10.95,45.45,10.98);
);

[out:json][timeout:25];
//o passandogli il nome es:"Catanzaro"
area["name"="Catanzaro"]["boundary"="administrative"]["admin_level"="8"]->.catanzaro;
(
  node["leisure"="park"](.catanzaro);
  way["leisure"="park"](.catanzaro);
  relation["leisure"="park"](.catanzaro);
) ->.parchi;           I

// trova i nodi con barrier=gate o entrance nelle aree verdi (via map_to_area)
.parchi map_to_area -> .aree_parco;

(
  node["barrier"="gate"](.aree_parco);
  node["barrier"="entrance"](.aree_parco);
);

out body;
>;
out skel qt;
// oppure out geom;

```

Spiegazione:

```
//es: "(45.42,10.95,45.45,10.98)"  
(  
  node["leisure"]="park"](45.42,10.95,45.45,10.98);  
  way["leisure"]="park"](45.42,10.95,45.45,10.98);  
  relation["leisure"]="park"](45.42,10.95,45.45,10.98);  
);  
  
[out:json][timeout:25];  
//o passandogli il nome es:"Catanzaro"  
area["name"="Catanzaro"]["boundary"="administrative"]["admin_level"="8"]->.catanzaro;  
(  
  node["leisure"]="park"](area.catanzaro);  
  way["leisure"]="park"](area.catanzaro);  
  relation["leisure"]="park"](area.catanzaro);  
)->.parchi;  
  
// trova i nodi con barrier=gate o entrance nelle aree verdi (via map_to_area)  
.parchi map_to_area -> .aree_parco;  
  
(  
  node["barrier"]="gate"(area.aree_parco);  
  node["barrier"]="entrance"(area.aree_parco);  
);  
  
out body;  
>>  
out skel qt;  
// oppure out geom;
```

Spiegazione:

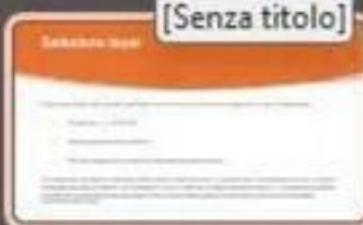
- out:json: formato di output (può essere anche XML)

Salvataggio automatico H aree verdi_AC_10_04_2025 + Filippo_re... • Salvato in questo PC FC - □ X

File Home Inserisci Disegno Progettazione Transizioni Animazioni Presentazione Registra Revisione Visualizza Guida

Incolla Diapositive

7 

8 

9 

10 

11 

Selezione layer

Dopo aver effettuato l'analisi dei layer disponibili si introducono i seguenti criteri di selezione:

- Superficie (> 5.000 m²)
- Destinazione d'uso pubblica
- Buona presenza di copertura arborea (se disponibile)

La selezione dei layer dipenderà dalla disponibilità dei dati: si potrebbero considerare solo gli accessi associati all'area di sfalcio, considerando usufruibile solo quella parte del parco. L'informazione della qualità sarà associata solo se disponibile il dato della copertura arborea o attributi di carattere qualitativo dell'area.

Fare clic per inserire le note

Diapositiva 8 di 11 61% 10:12 12/06/2025

Windows Taskbar:

MapStruct 1.6.3 Reference | TOTJ | overpass-turbo.eu | Riavvia per aggiornare

Esegui Condividi Esporta Wizard Salva Carica Impostazioni Aiuto overpass turbo

```
1[out:json][timeout:25];
2//o passandogli il nome es:"Catanzaro"
3area["name"="Catanzaro"]["boundary"="administrative"]["admin_level"="8"]->.catanzaro;
4(
5  node["leisure"="park"](.catanzaro);
6  way["leisure"="park"](.catanzaro);
7  relation["leisure"="park"](.catanzaro);
8);
9
10
11out body;
12>;
13out skel qt;
14
```

Map showing the area around Rome, Italy, with major roads and natural reserves. The map includes labels for the Vatican City, Civitas Vaticana - Città del Vaticano, Riserva Naturale Statale del Litorale Romano, Parco Naturale Regionale Appia Antica, and surrounding towns like Frascati, Monte Compatri, and Grottaferrata. Roads shown include A1, A12, A24, A91, SS1, SS7, and SR6. A search bar and zoom controls are visible on the map interface.

Scrivi qui il testo da cercare.

10:14 12/06/2025

Commenti

Consente di visualizzare e rispondere ai commenti inclusi nel documento.

```
//es:" (45.42,10.95,45.45,10.98)"
(
    node["leisure"="park"](45.42,10.95,45.45,10.98);
    way["leisure"="park"](45.42,10.95,45.45,10.98);
    relation["leisure"="park"](45.42,10.95,45.45,10.98);
);

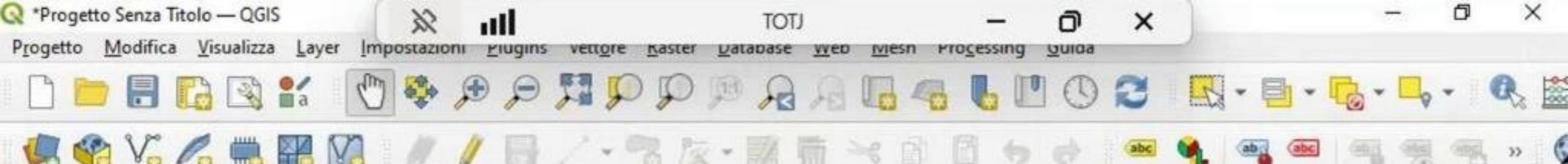
[out:json][timeout:25];
//o passandogli il nome es:"Catanzaro"
area["name"="Catanzaro"]["boundary"="administrative"]["admin_level"="8"]->.catanzaro;
(
    node["leisure"="park"](.catanzaro);
    way["leisure"="park"](.catanzaro);
    relation["leisure"="park"](.catanzaro);
) ->.parchi;

// trova i nodi con barrier=gate o entrance nelle aree verdi (via map_to_area)
.parchi map_to_area -> .aree_parco;

(
    node["barrier"="gate"](.aree_parco);
    node["barrier"="entrance"](.aree_parco);
);

out body;
>;
out skel qt;
// oppure out geom;
```

Spiegazione:

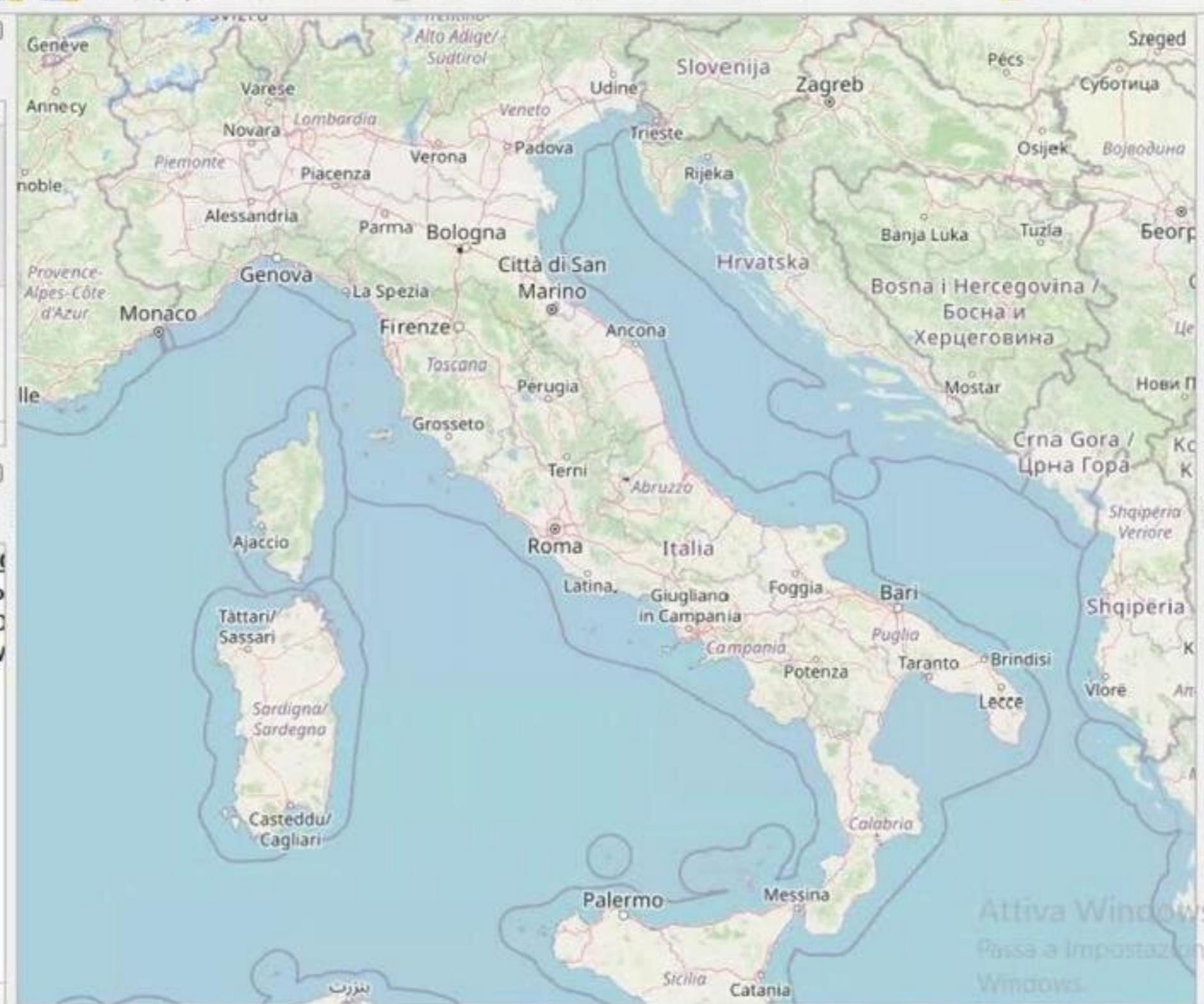


Browser

- Cloud
- Scenes
- SensorThings
- Vector Tiles
- XYZ Tiles
- google earth
- Mapzen Global Terrain
- OpenStreetMap
- WCS
- WFS / OGC API - Features
- ArcGIS REST Servers

Layer

- Aree Verdi Catanzaro (Aree Verdi Catanzaro)
- Aree_Verdi_Casalecchio
- Aree_Verdi_Terracina_C
- Aree_Verdi_Aquila_OSN
- OpenStreetMap



Digita per localizzare (Ctrl+K)

Coordinate: 639942 5007607

Scal: 1:9196496

Ente d'ingrandimenti: 100%

Rotazione: 0,0 °

10:52

12/06/2025

Strumenti di Processing

- rip
- Usati di recente
- Generale vettore
- Geometria vettore
- GDAL

Informazioni Risultati

- Elemento

Modalità Layer Corrente

Vista Albero



Scrivi qui il testo da cercare.



*Progetto Senza Titolo — QGIS

Progetto Modifica Visualizza Layer Impostazioni Plugins vettore Raster Database vveo mesi Processing Guida

TOTI

Digitare per localizzare (Ctrl+K) 1474879 5055519 1:14121 d'ingrandim 100% 0,0 ° Visualizza EPSG:3857

Scrivi qui il testo da cercare.

Strumenti di Processing

rip

- Usati di recente
- Generale vettore
- Geometria vettore
- GDAL

Informazioni Risultati

Elemento	Valore
addr:houseNumber	NULL
addr:postcode	NULL
addr:street	NULL
description	NULL
image	NULL
layer	1
leisure	park
name	Parco della B
operator	NULL
type	multipolygon
website	NULL
wikidata	Q3895518

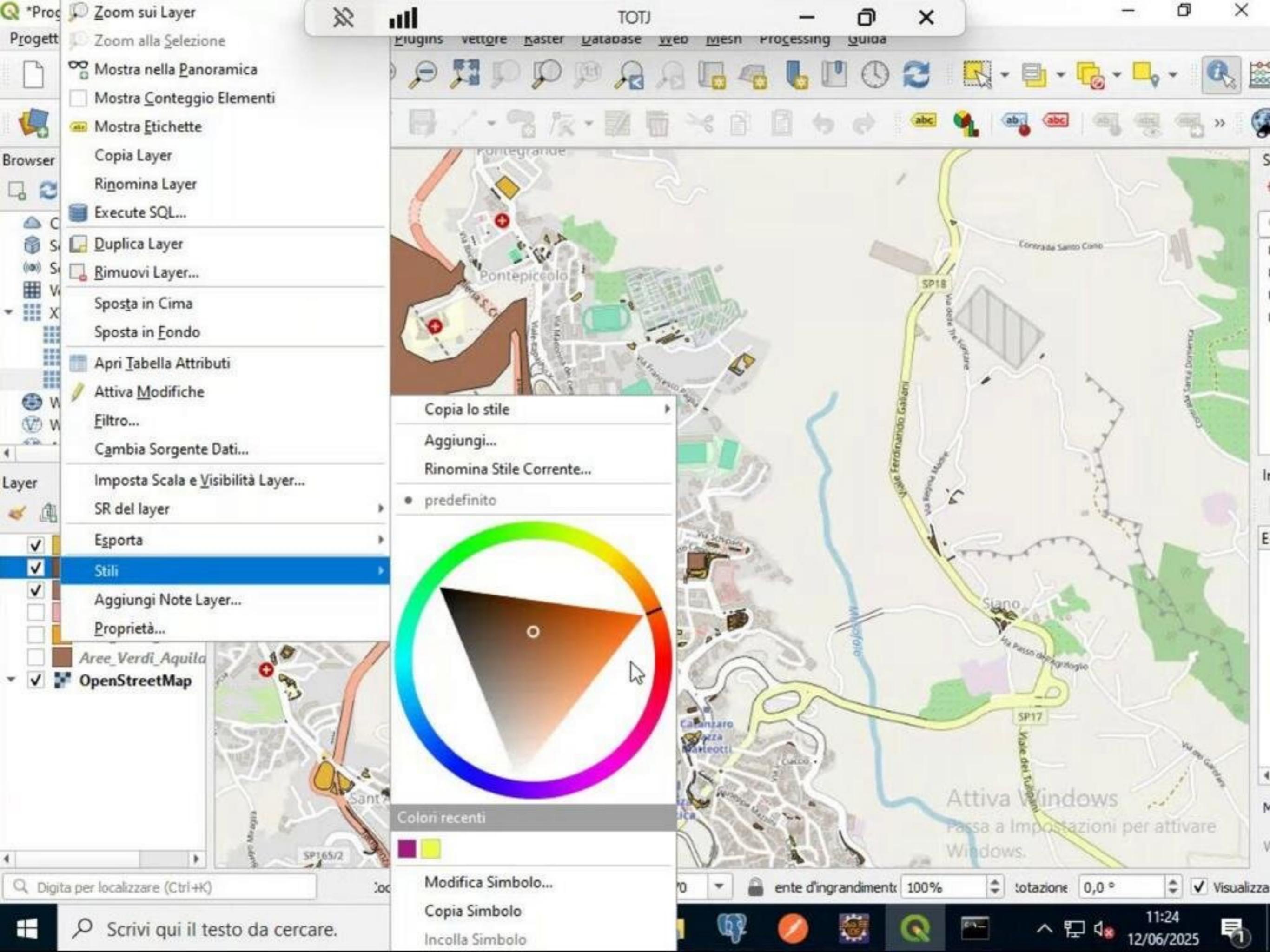
Attiva Windows

Modalità Layer Corrente

Passa a Impostazioni per attivare

Vista Albero

11:21 12/06/2025





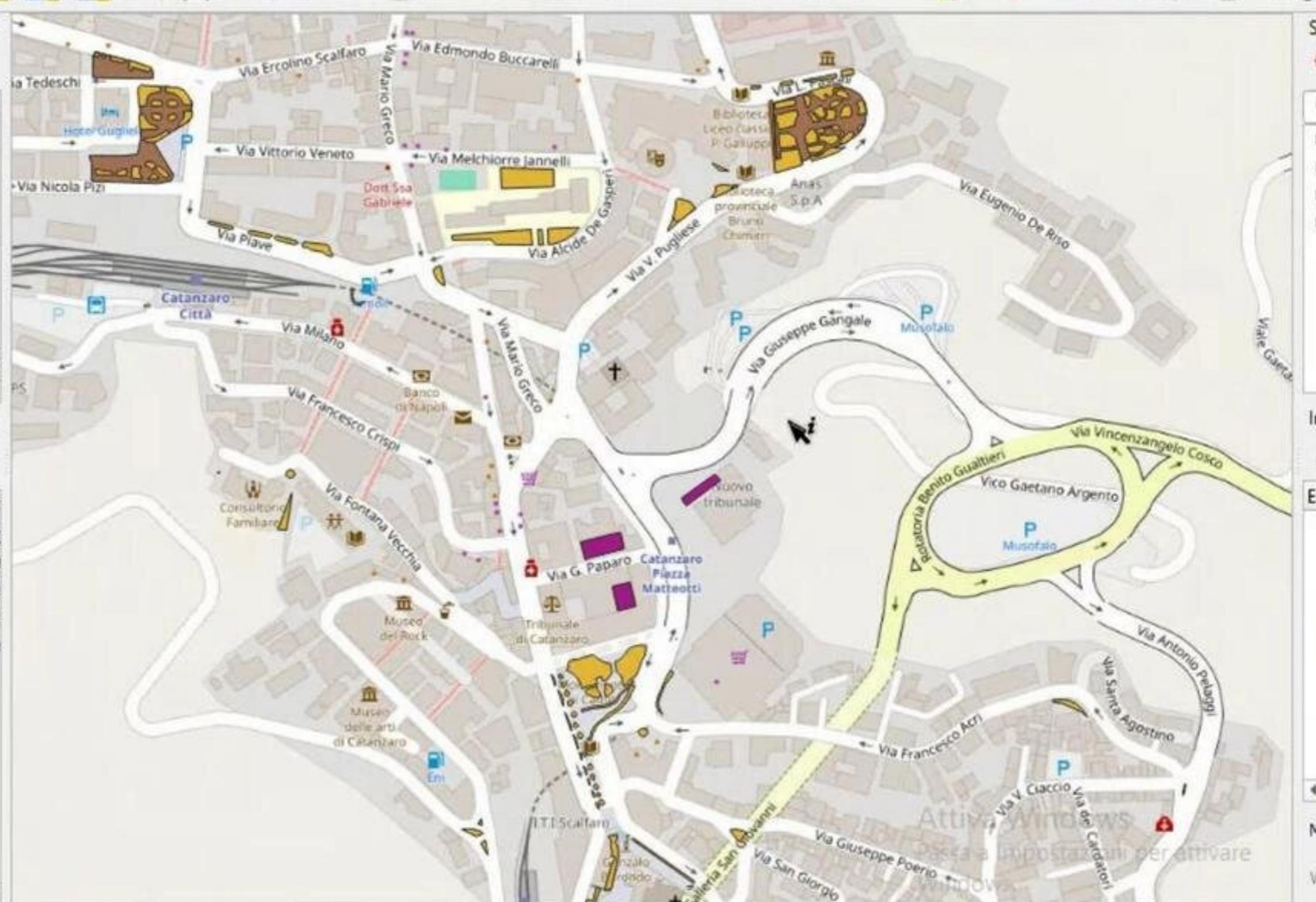
Browser

- Cloud
- Scenes
- SensorThings
- Vector Tiles
- XYZ Tiles
- google earth
- Mapzen Global Tiles
- OpenStreetMap
- WCS
- WFS / OGC API - Features

- WPS
- Geoprocessing
- Processing

Layer

- GR_TAPPETI4WEB
- GR_AREE4WEBPol
- Aree_Verdi_Catanz
- Aree_Verdi_Casale
- Aree_Verdi_Terrac
- Aree_Verdi_Aquila
- OpenStreetMap



Digita per localizzare (Ctrl+K)

Coordinate: 1846733 4709126

Scale: 1:5636

Ente d'ingrandimento: 100%

Rotazione: 0,0 °

Visualizza



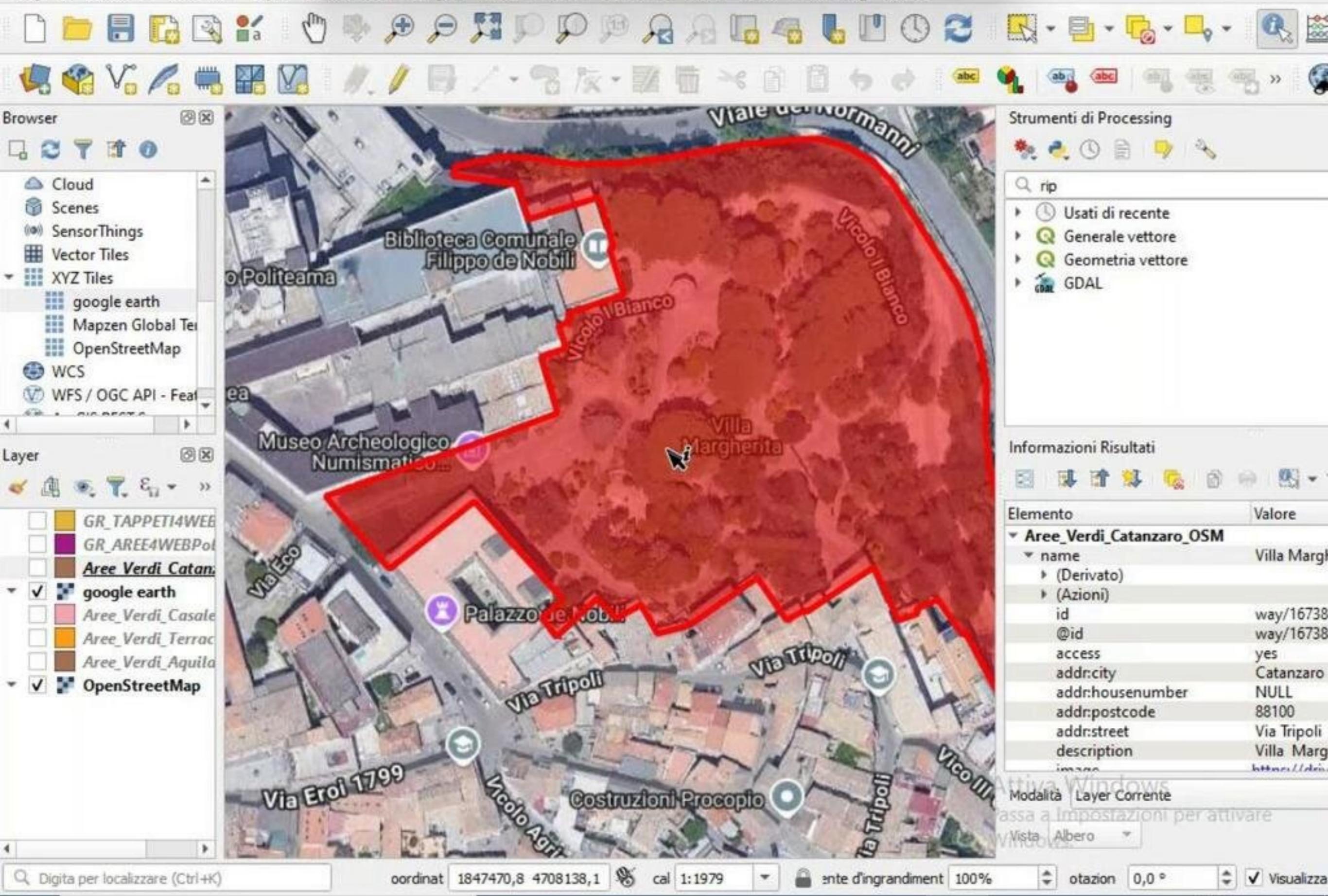
Scrivi qui il testo da cercare.



11:25

12/06/2025







Browser

- Cloud
- Scenes
- SensorThings
- Vector Tiles
- XYZ Tiles
- google earth
- Mapzen Global Tiles
- OpenStreetMap
- WCS
- WFS / OGC API - Features

Layer

- GR_TAPPETI4WEB
- GR_AREE4WEBPol
- Aree Verdi Catanzaro**
- google earth
- Aree_Verdi_Casale
- Aree_Verdi_Terracina
- Aree_Verdi_Aquila
- OpenStreetMap



Strumenti di Processing



rip

- Usati di recente
- Generale vettore
- Geometria vettore
- GDAL

Informazioni Risultati

Elemento	Valore
Aree_Verdi_Catanzaro_OSM	
name	Villa Margherita
(Derivato)	
(Azioni)	
id	way/16738
@id	way/16738
access	yes
addr:city	Catanzaro
addr:housenumber	NULL
addr:postcode	88100
addr:street	Via Tripoli
description	Villa Margherita
image	https://.../16738

Modalità Layer Corrente

Passa a Impostazioni per attivare

Vista Albero

Digita per localizzare (Ctrl+K)

Spost: coordinate 1847545,7 4708138,1

cal 1:1979

ente d'ingrandimento 100%

otazioni 0,0 °

Visualizza



Scrivi qui il testo da cercare.



11:29

12/06/2025





Browser

- Cloud
- Scenes
- SensorThings
- Vector Tiles
- XYZ Tiles
- google earth
- Mapzen Global Tiles
- OpenStreetMap

- WCS
- WFS / OGC API - Feat

Layer

- GR_TAPPETI4WEB
- GR_AREEVERDI
- GR_TAPPETI4WEBPolygon (MultiPolygon -)
C:\Users\Administrator\Desktop\DATA\AREE_VERDI\Catanzaro\GR_TAPPETI4WEBPolygon.shp
- Aree_verdi_Casate
- Aree_Verdi_Terrac
- Aree_Verdi_Aquila
- OpenStreetMap



Digita per localizzare (Ctrl+K)

coordinat

1847303,8 4708068,9

cal

1:1979

ente d'ingrandimento

100%

otazion

0,0 °

11:30

12/06/2025

Scrivi qui il testo da cercare.



Strumenti di Processing



rip

- Usati di recente
- Generale vettore
- Geometria vettore
- GDAL

Informazioni Risultati



Elemento

Elemento	Valore
AREE_VERDI_OSM	Villa Marg...
name	(Derivato)
id	way/16738
access	yes
addr:city	Catanzaro
addr:housenumber	NULL
addr:postcode	88100
addr:street	Via Tripoli
description	Villa Marg...
image	https://...

Attiva Windows

Modalità Layer Corrente

Passa a Impostazioni per attivare

Vista Albero



Browser

- Cloud
- Scenes
- SensorThings
- Vector Tiles
- XYZ Tiles
- google earth
- Mapzen Global Tiles
- OpenStreetMap
- WCS
- WFS / OGC API - Features

Layer

- GR_TAPPETI4WEB
- GR_AREE4WEBPol
- Aree Verdi Catan...
- google earth
- Aree_Verdi_Casale
- Aree_Verdi_Terrac...
- Aree_Verdi_Aquila
- OpenStreetMap



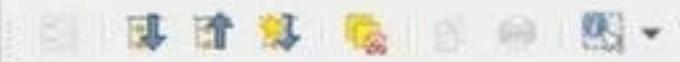
Strumenti di Processing



rip

- Usati di recente
- Generale vettore
- Geometria vettore
- GDAL

Informazioni Risultati



Elemento

Valore

Attiva Windows

Modalità Layer Corrente

Passa a Impostazioni per attivare

Vista Albero

Digita per localizzare (Ctrl+K)

Spost: coordinate 1847596,0 4708050,6

cal 1:1979

Ente d'ingrandimento 100%

otazioni 0,0 °

11:30

12/06/2025



Scrivi qui il testo da cercare.





Browser



- Cloud
- Scenes
- SensorThings
- Vector Tiles
- XYZ Tiles
- google earth
- Mapzen Global Tiles
- OpenStreetMap
- WCS
- WFS / OGC API - Features

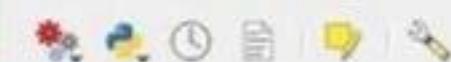
Layer



- GR_TAPPETI4WEB
- GR_AREE4WEBPol
- Aree_Verdi_Catanz
- google earth
- Aree_Verdi_Casale
- Aree_Verdi_Terrac
- Aree_Verdi_Aquila
- OpenStreetMap



Strumenti di Processing



rip

- Usati di recente
- Generale vettore
- Geometria vettore
- GDAL

Informazioni Risultati



Elemento

Valore

Attiva Windows
Modalità Layer Corrente
Passa a Impostazioni per attivare
Vista Albero

Digita per localizzare (Ctrl+K)

Coordinate: 1847150 4708085

Scale: 1:3958

Ente d'ingrandimento: 100%

Rotazione: 0,0 °

11:36

12/06/2025

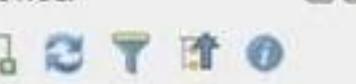


Scrivi qui il testo da cercare.





Browser



-  Cloud
 -  Scenes
 -  SensorThings
 -  Vector Tiles
 -  XYZ Tiles
 -  google earth
 -  Mapzen Global
 -  OpenStreetMap
 -  WCS
 -  WFS / OGC API - Fe

Layer



-  Aree Verdi Casa
 -  aree-verdi-pianc
 -  GR_TAPPETI4WI
 -  GR_AREE4WEBP
 -  Aree_Verdi_Cata
 -  google earth
 -  Aree_Verdi_Terre
 -  Aree_Verdi_Aqui
 -  OpenStreetMap



Strumenti di Processing



9 rig

- ▶  Usati di recente
 - ▶  Generale vettore
 - ▶  Geometria vettore
 - ▶  GDAL

Informazioni Risultati



Elemento	Valore
----------	--------

卷之三

Valore

Modalità Layer Corrente

 Digita per localizzare (Ctrl+F)

Spost: Coordinate: 1256346 5538668

calc 1:10753

ente d'ingrandimenti 100%

Rotazione 0,0 °



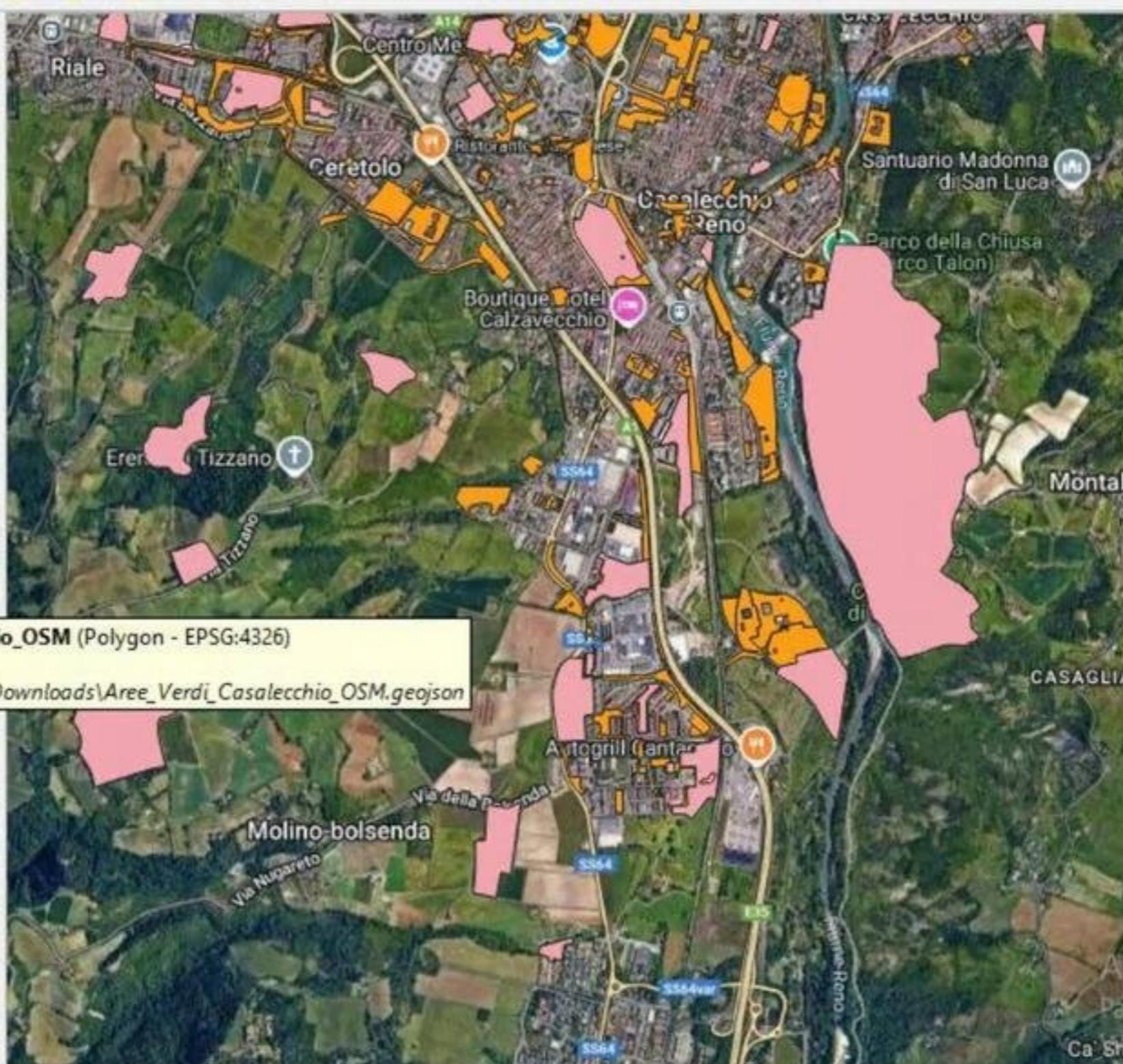


Browser

- Cloud
- Scenes
- SensorThings
- Vector Tiles
- XYZ Tiles
- google earth
- Mapzen Global
- OpenStreetMap
- WCS
- WFS / OGC API - Fe

Layer

- Aree Verdi Casa
- Aree_Verdi_Casalecchio_OSM (Polygon - EPSG:4326)
 - C: \\Users\\Administrator\\Downloads\\Aree_Verdi_Casalecchio_OSM.geojson
- Aree_Verdi_Cata
- google earth
- Aree_Verdi_Terre
- Aree_Verdi_Aqui
- OpenStreetMap



Strumenti di Processing

- rip
- Usati di recente
- Generale vettore
- Geometria vettore
- GDAL

Informazioni Risultati



Elemento	Valore

Attiva Windows
Modalità Layer Corrente
Passa a Impostazioni per attivare

Vista Albero

Scrivi qui il testo da cercare.

Digit per localizzare (Ctrl+K) coordinate: 1251253 5537322 icale: 1:43013 ente d'ingrandimenti: 100%

Rotazione: 0,0 ° Visualizza

Progetto Modifica Visualizza Layer Impostazioni Plugins vettore Raster Database vwo mesn Processing Guida

TOTJ

Digitare per localizzare (Ctri+K)

coordinate 1846788,3 4709418,4

cal 1:1140

Ente d'ingrandimento 100%

otazion 0,0 °

Visualizza

Scrivi qui il testo da cercare.

Strumenti di Processing

rip

- Usati di recente
- Generale vettore
- Geometria vettore
- GDAL

Informazioni Risultati

Elemento Valore

Attiva Windows

Modalità Layer Corrente

Passa a Impostazioni per attivare

Vista Albero

11:54

12/06/2025

1

Cloud

Scenes

SensorThings

Vector Tiles

XYZ Tiles

google earth

Mapzen Global

OpenStreetMap

WCS

WFS / OGC API - Fe

Layer

Aree_Verdi_Casa

aree-verdi-piani

GR_ALBERI4WEB

GR_TAPPETI4WEB

GR_AREE4WEBP

Aree_Verdi_Cata

google earth

Aree_Verdi_Terre

Aree_Verdi_Aqui

OpenStreetMap

Caserma Pepe

Museo del Risorgimento

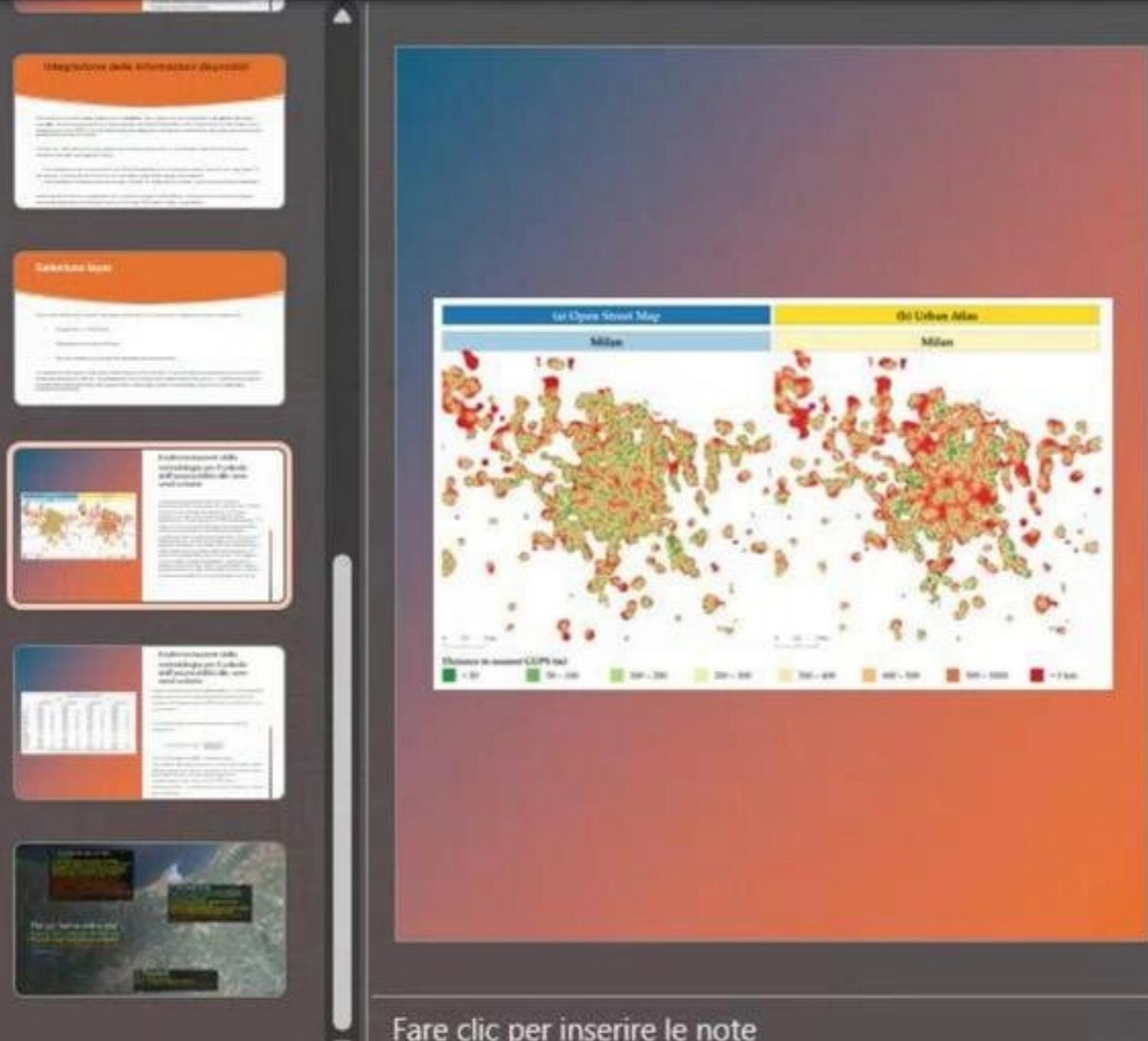
Via L. Pascali

Biblioteca Liceo classico P. Galluppi

Biblioteca provinciale Bruno Chimirri

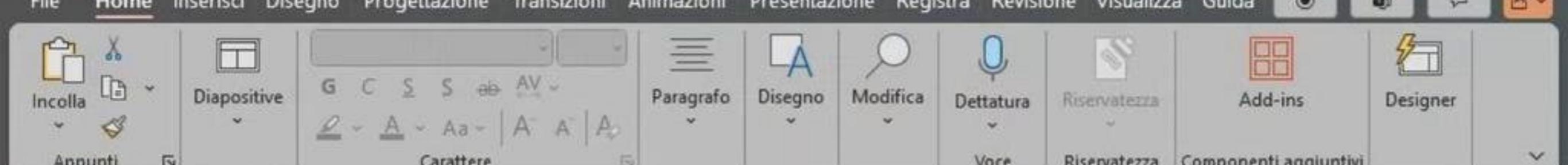
Anas S.p.A

Via Eugenio De Riso



Implementazione della metodologia per il calcolo dell'accessibilità alle aree verdi urbane

La metodologia selezionata per il calcolo dell'accessibilità viene descritta nell'articolo: "Green Urban Public Spaces Accessibility: A Spatial Analysis for the Urban Area of the 14 Italian Metropolitan Cities Based on SDG Methodology". Si basa sull'applicazione dell'algoritmo della minima distanza e permette di classificare le aree in prossimità degli accessi alle aree verdi in funzione della distanza. L'obiettivo è quello di individuare il migliore percorso (il minimo) tra il centroide di ogni cella (inerente al grigliato della popolazione) e il punto di ingresso del parco più vicino. Per ciascun punto iniziale, viene considerato il percorso più breve e quindi ad ogni cella che contiene il valore della popolazione viene associato anche un valore di distanza rispetto al punto di accesso più vicino.



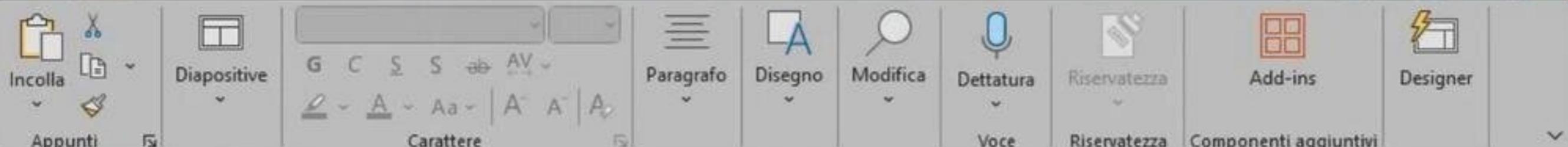
Selezione layer

Dopo aver effettuato l'analisi dei layer disponibili si introducono i seguenti criteri di selezione:

- Superficie (> 5.000 m²)
- Destinazione d'uso pubblica
- Buona presenza di copertura arborea (se disponibile)

La selezione dei layer dipenderà dalla disponibilità dei dati: si potrebbero considerare solo gli accessi associati all'area di sfalcio, considerando usufruibile solo quella parte del parco. L'informazione della qualità sarà associata solo se disponibile il dato della copertura arborea o attributi di carattere qualitativo dell'area.

Fare clic per inserire le note



Integrazione delle informazioni disponibili

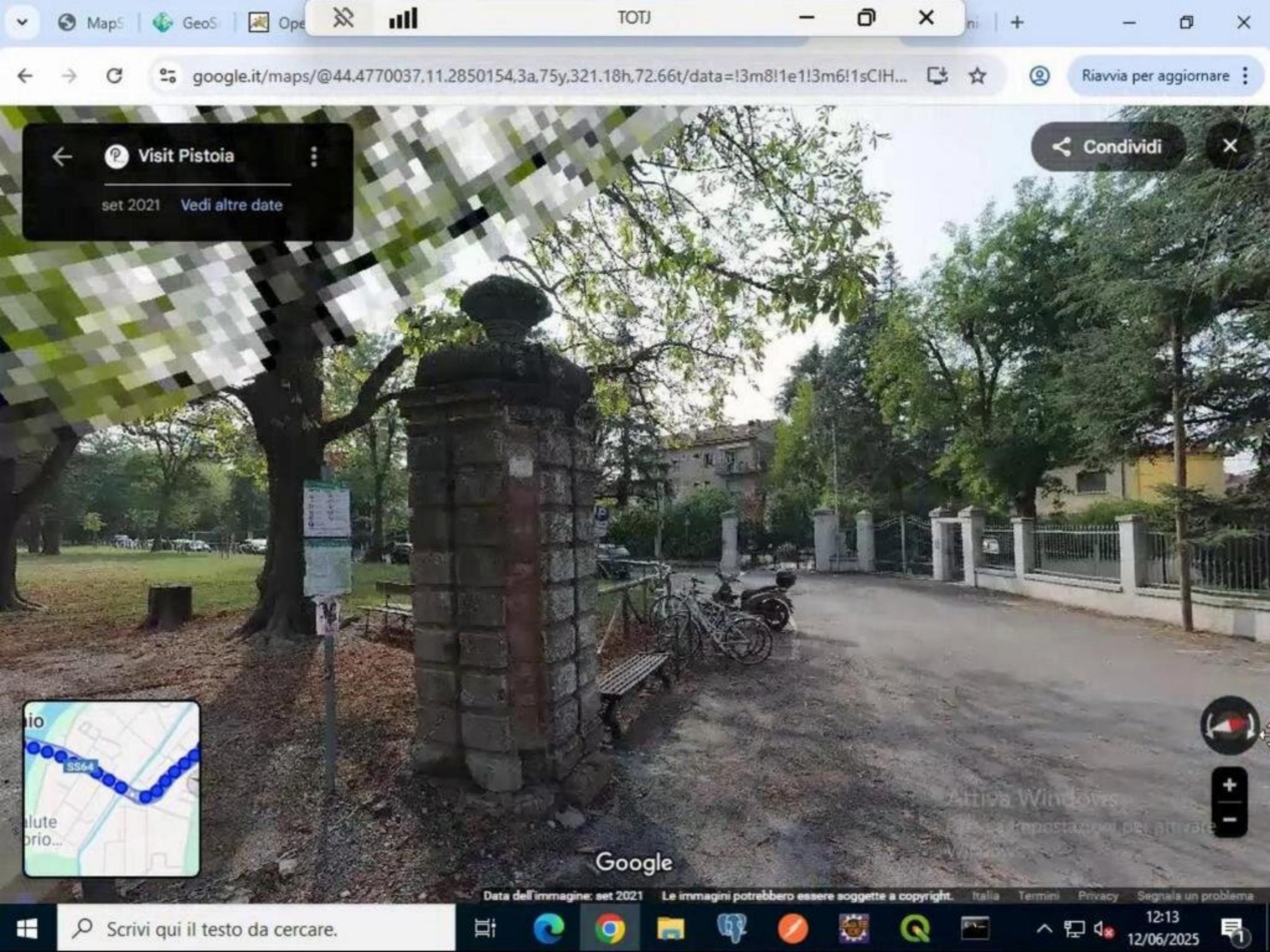
Nonostante l'obiettivo sia quello di procedere con un approccio **bottom-up partendo da fonti ufficiali**, la sovrapposizione con layer estratti da OpenStreetMap e da Copernicus Urban Atlas Land Cover/Land Use 2018 (UA) diventa utile per integrare il contenuto informativo dei layer resi disponibili dalle Amministrazioni locali.

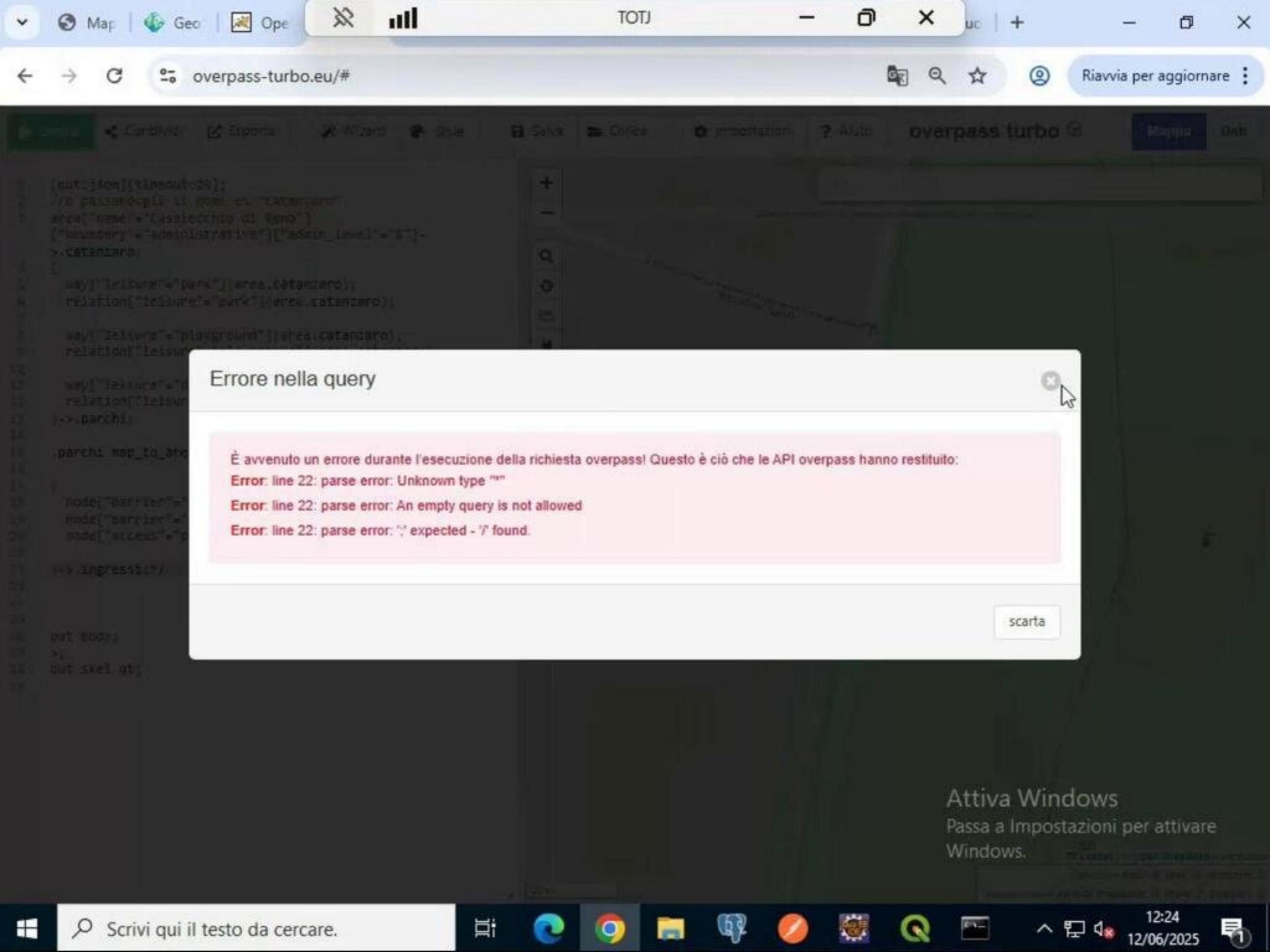
Inoltre, se i dati dei punti di accesso non fossero disponibili, si potrebbero identificare attraverso l'analisi spaziale nei seguenti modi:

- Considerare solo i nodi estratti da OpenStreetMap con la parola chiave "barrier" ed i tag "gate" e "entrance", localizzati all'interno o sul perimetro dell'area verde considerata
- Considerare l'intersezione tra il layer "strade" e "aree verdi urbane", scaricati da OpenStreetMap.

Laddove le condizioni precedenti non possono essere soddisfatte, una soluzione potrebbe essere rappresentata dalla individuazione di punti ogni 100 metri lungo il perimetro.

Fare clic per inserire le note





Map | Geo | Ope | TOTJ

openstreetmap.org/#map=9/39.931/16.219

Riavvia per aggiornare

OpenStreetMap

Modifica

Cronologia Esporta Tracciati GPS Diari degli utenti Comunità Copyright Altro

Accedi Registrati

Cerca

Dove si trova?

+

-

1

+

i

?

20 km

10 mi

Basilicata

Roccadaspide

Sala Consilina

Agropoli

Vallo della Lucania

Sapri

Lauria

Parco Nazionale del Pollino

Scalea

Castrovilliari

Bonifati

Cetraro

La Gàrdia / Guardia Piemontese

Bisignano

Acri

Rossano Scalo

Montalto Uffugo

Paola

Taranto

Talsano

Lizzano

Fontanelli

Atten Windows

Passa a impostazioni per attivare Windows

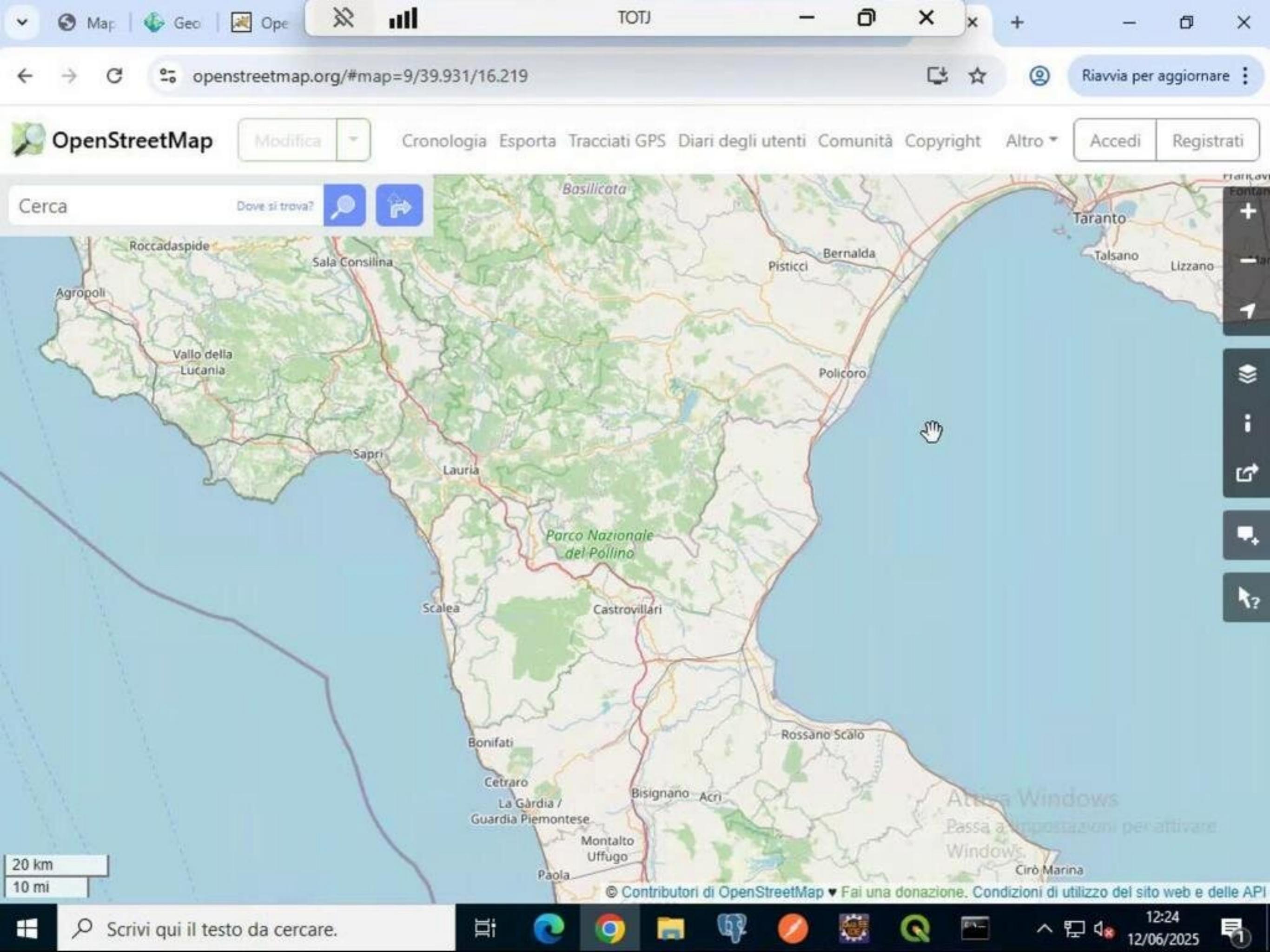
Cirò Marina

© Contributori di OpenStreetMap ▾ Fai una donazione. Condizioni di utilizzo del sito web e delle API

Scrivi qui il testo da cercare.

12:24

12/06/2025





Browser

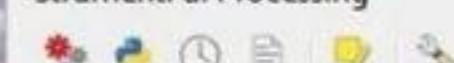
- Cloud
- Scenes
- SensorThings
- Vector Tiles
- XYZ Tiles
 - google earth
 - Mapzen Global
 - OpenStreetMap
- WCS
- WFS / OGC API - Fe

Layer

- Ingressi_Aree_Ve
- Ingressi_Aree_Ve
- Ingressi_Aree_Ve
- Ingressi_Aree_Ve
- Aree_Verdi_Casa
- Aree_Verdi_Cata
- Aree_Verdi_Terre
- Aree_Verdi_Aqui
- google earth
- OpenStreetMap



Strumenti di Processing



rip

- Usati di recente
- Generale vettore
- Geometria vettore
- GDAL

Informazioni Risultati



Elemento	Valore
Arete_Verdi_Casalecchio_OSM	
name	Parco Fresu
(Derivato)	
(Azioni)	
id	way/681394275
@id	way/681394275
access	NULL
fixme	NULL
leisure	park
name	Parco Fresu
openin...	NULL
ref:cori...	NULL
ref:cori...	NULL

Attiva Windows

Passa a Impostazioni per attivare

Vista Albero

Windows

Digita per localizzare (Ctrl+K)

Interni coordinate

1252119,0 5540459,5



cal

1:719



ente d'ingrandimento

100%



otazion

0,0 °



Visualizza



Scrivi qui il testo da cercare.



12:45

12/06/2025



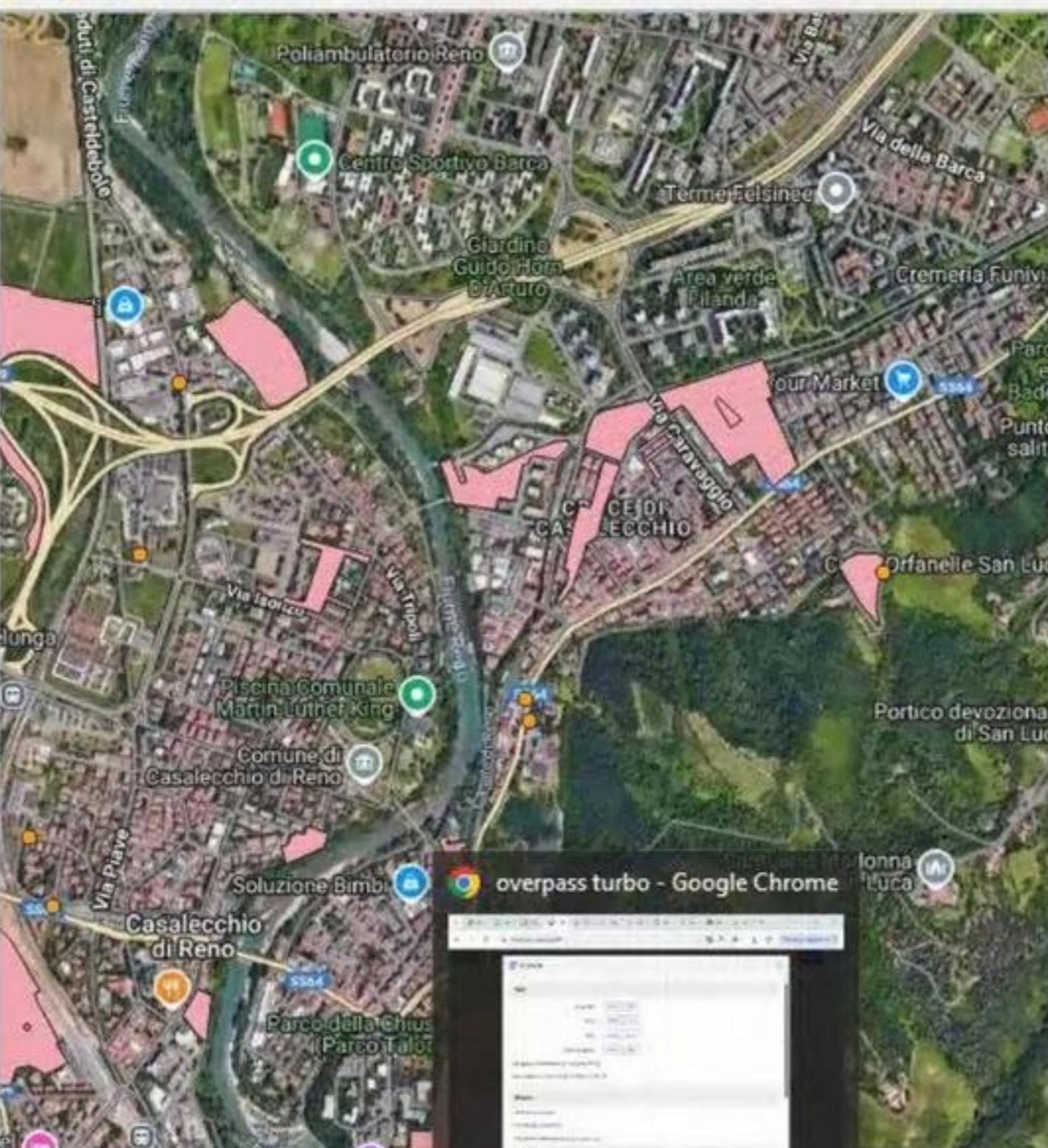


Browser

- Cloud
- Scenes
- SensorThings
- Vector Tiles
- XYZ Tiles
 - google earth
 - Mapzen Global
 - OpenStreetMap
- WCS
- WFS / OGC API - Fe

Layer

- Ingressi_Aree_Ve**
- Ingressi_Aree_Ve
- Ingressi_Aree_Ve
- Ingressi_Aree_Ve
- Areare_Casa**
- Areare_Cata
- Areare_Terre
- Areare_Aqui
- google earth
- OpenStreetMap



Digitare per localizzare (Ctrl+K)

Coordinate: 125637

Ente d'ingrandimento: 100%

Rotazione: 0,0 °

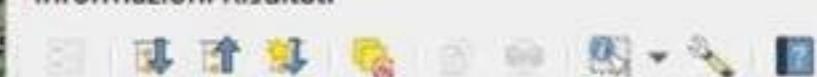
Visualizza

Strumenti di Processing

rip

- Usati di recente
- Generale vettore
- Geometria vettore
- GDAL

Informazioni Risultati



Elemento

Valore

Modalità: Layer Corrente

Passa a Impostazioni per attivare

Vista

Albero

Windows

12:47

12/06/2025



Scrivi qui il testo da cercare.



Salvataggio automatico H ⌂ ⌂ Overpass_OpenStreetMap_API_VEBS.docx -... • Salvato in questo PC FC

File Home Inserisci Disegno Progettazione Layout Riferimenti Lettere Revisione Visualizza Guida Commenti

Continuazione sviluppo delle classi per le API REST e dei relativi servizi

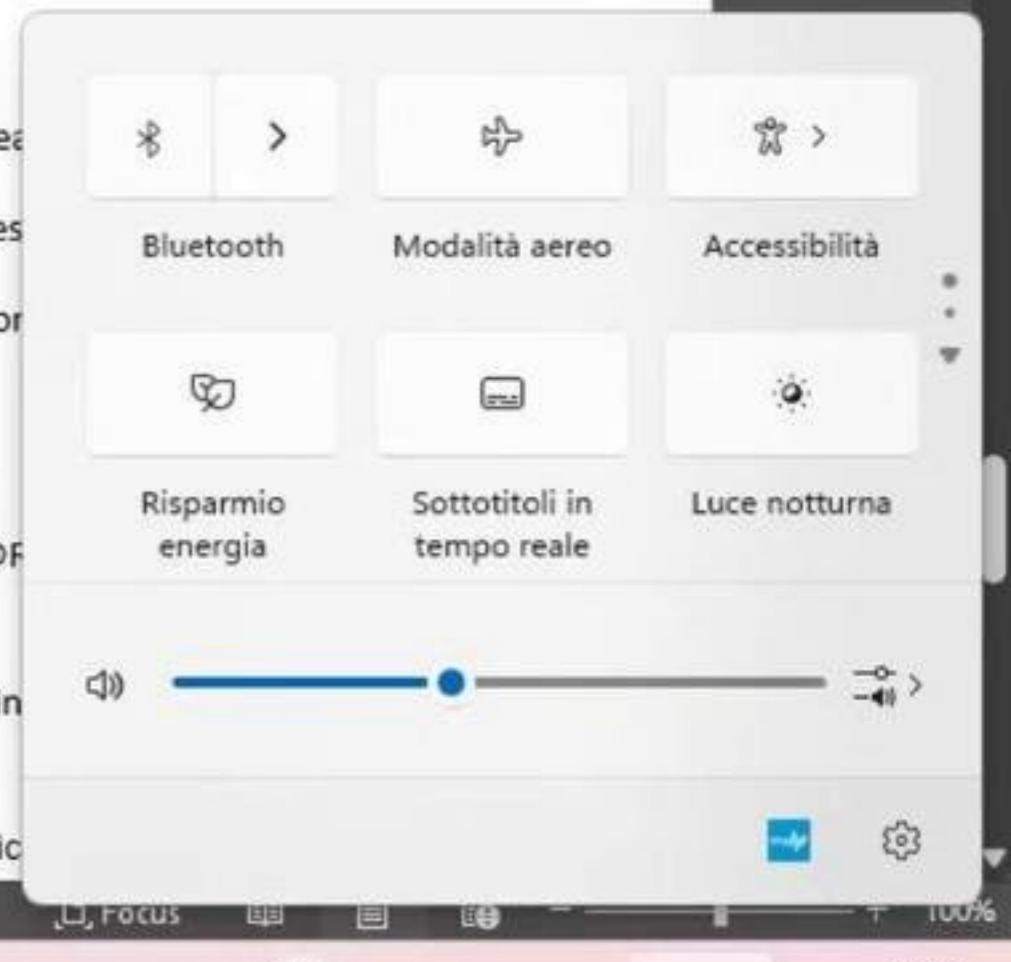
In questa fase del training si proseguirà lo sviluppo dell'architettura a strati del progetto, con particolare attenzione alla costruzione delle API REST e dei corrispondenti servizi di business logic nel backend Java/Spring.

Obiettivi principali:

- Consolidare e completare le operazioni CRUD (Create, Read, Update, Delete) per le principali entità del progetto, in particolare quelle legate alla gestione delle aree verdi, dei poligoni associati e di altri elementi territoriali.
- Sviluppare nuove API REST personalizzate per supportare operazioni specifiche richieste dal flusso applicativo, ad esempio:
 - Inserimento di poligoni selezionati su mappa
 - Aggiornamento di metadati ambientali associati a un'area
 - Recupero filtrato di aree sulla base di criteri spaziali o des
- Implementare la comunicazione tra controller, service e repository secondo il principio della separazione delle responsabilità.

Attività tecniche previste:

- Progettazione e stesura di nuovi controller REST, annotati con `@RequestMapping`.
- Definizione di endpoint REST con annotazioni come `@GetMapping`, `@PutMapping` e `@DeleteMapping`.
- Implementazione dei metodi nei service, che conterranno la logic



Pagina 6 di 7 4 di 1055 parole Completamento del testo attivato 100% 31° 14:06 12/06/2025

Windows Start File Explorer Mail Internet Explorer Microsoft Edge Word PowerPoint Google

Continuazione sviluppo delle classi per le API REST e dei relativi servizi

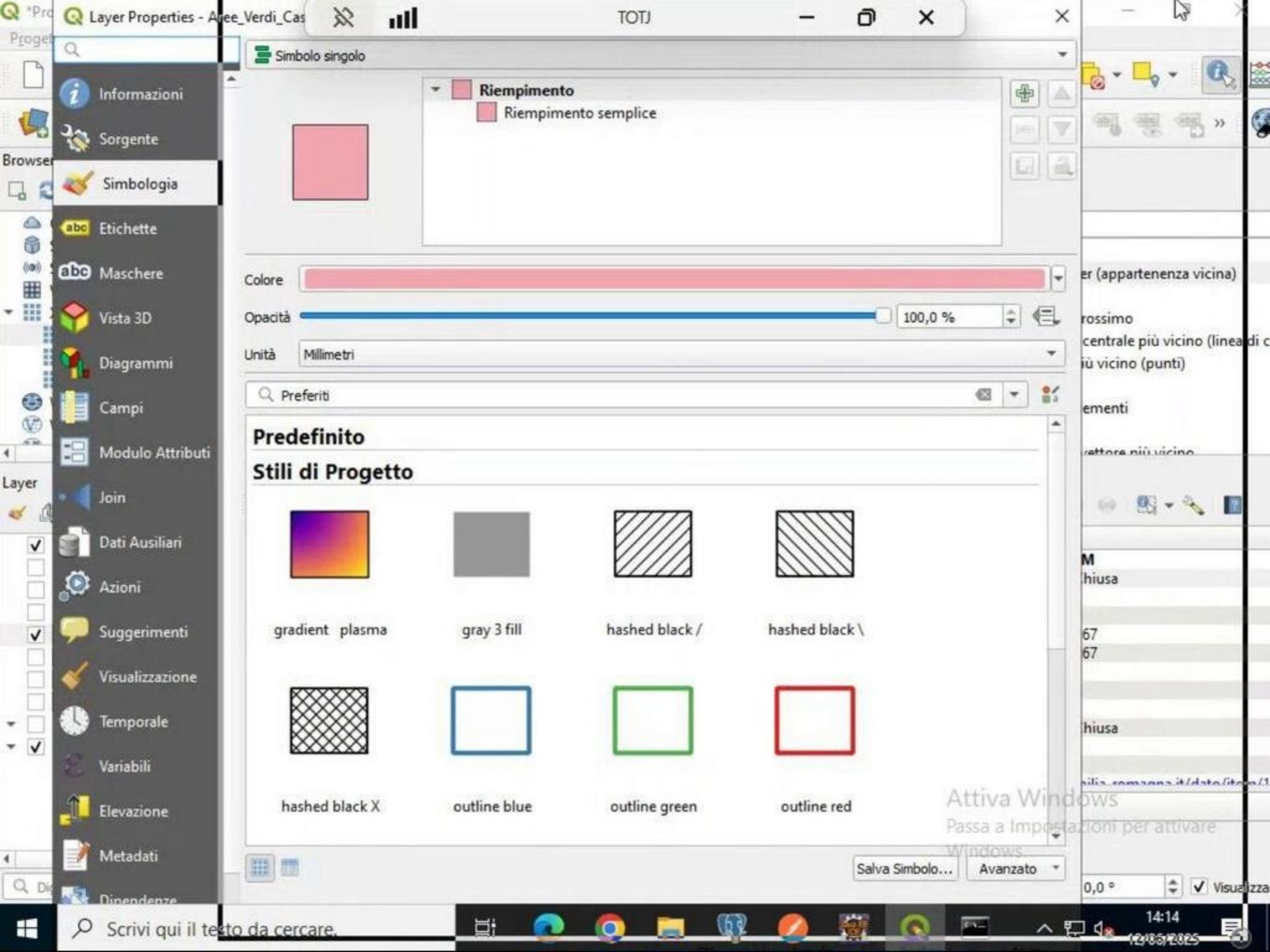
In questa fase del training si proseguirà lo sviluppo dell'architettura a strati del progetto, con particolare attenzione alla costruzione delle API REST e dei corrispondenti servizi di business logic nel backend Java/Spring.

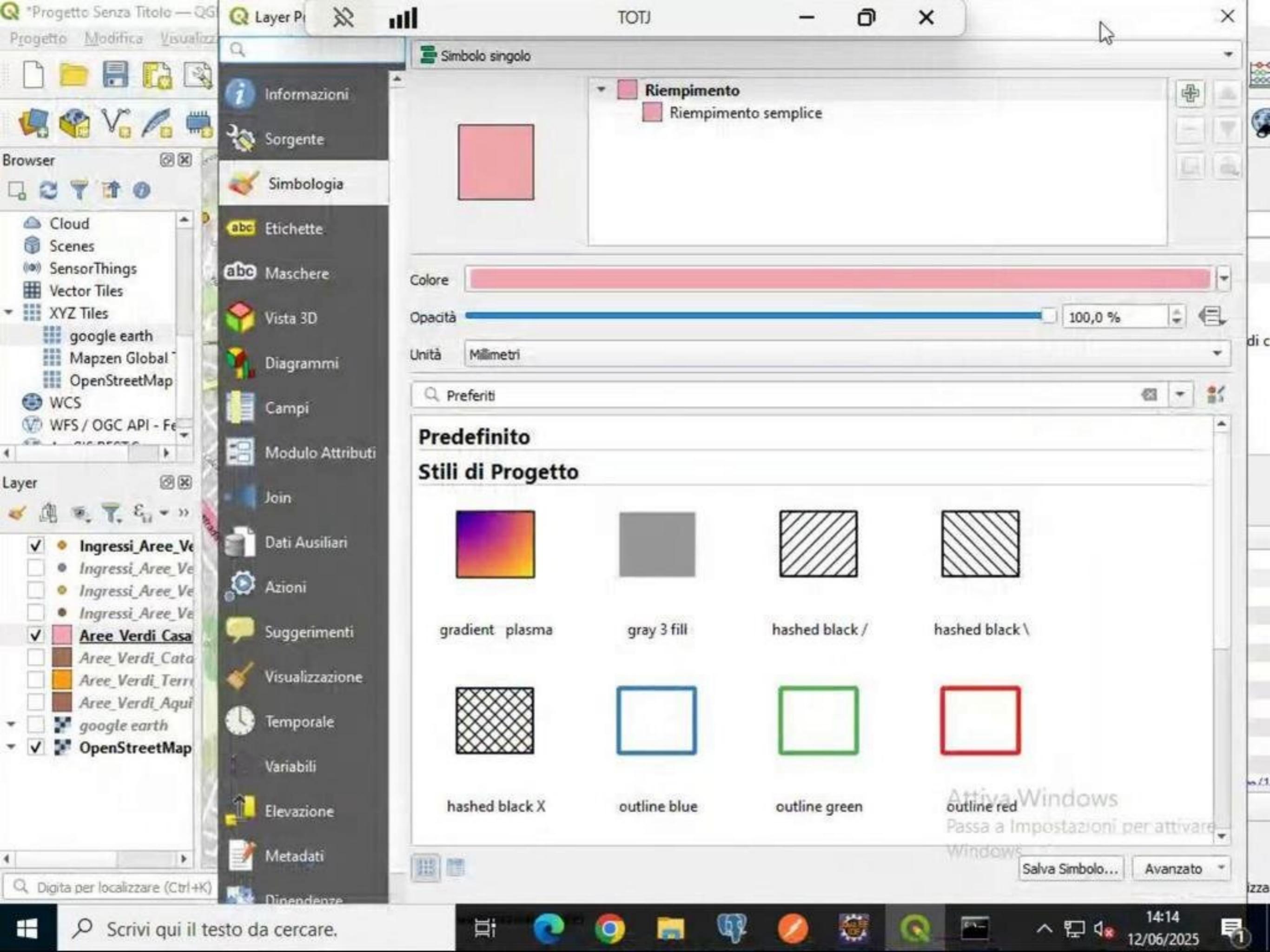
Obiettivi principali:

- Consolidare e completare le operazioni CRUD (Create, Read, Update, Delete) per le principali entità del progetto, in particolare quelle legate alla gestione delle aree verdi, dei poligoni associati e di altri elementi territoriali.
- Sviluppare nuove API REST personalizzate per supportare operazioni specifiche richieste dal flusso applicativo, ad esempio:
 - Inserimento di poligoni selezionati su mappa
 - Aggiornamento di metadati ambientali associati a un'area
 - Recupero filtrato di aree sulla base di criteri spaziali o descrittivi
- Implementare la comunicazione tra controller, service e repository, secondo i principi della separazione delle responsabilità.

Attività tecniche previste:

- Progettazione e stesura di nuovi controller REST, annotati con `@RestController` e `@RequestMapping`.
- Definizione di endpoint REST con annotazioni come `@GetMapping`, `@PostMapping`, `@PutMapping` e `@DeleteMapping`.
- Implementazione dei metodi nei service, che conterranno la logica di validazione e





Progetto Sen *Zoom sui Layer

Progetto **Mod**

Zoom alla Selezione

Mostra nella Panoramica

Mostra Conteggio Elementi

Mostra Etichette

Copia Layer

Rinomina Layer

Execute SQL...

Duplica Layer

Rimuovi Layer...

Sposta in Fondo

Apri Tabella Attributi

Attiva Modifiche

Filtro...

Cambia Sorgente Dati...

Imposta Scala e Visibilità Layer...

SR del layer

Esporta

Stili

Aggiungi Note Layer...

Proprietà...

Ingre

Ingre

Ingre

Ingressi_zucca_z

Aree_Verdi_Casa

Aree_Verdi_Cata

Aree_Verdi_Terre

Aree_Verdi_Aqui

google earth

OpenStreetMap

vettore **Raster** **Database** **Web** **mesn** **Processing** **Guida**

TOTJ

Strumenti di Processing

vicin

- Analisi raster**
 - Rendi fuzzy un raster (appartenenza vicina)
- Analisi sui vettori**
 - Analisi vicino più prossimo
 - Distanza dal punto centrale più vicino (linea di c)
 - Distanza dall'hub più vicino (punti)
- Cartografia**
 - Allinea punti agli elementi
- Generale vettore**
 - Unisci attributi dal vettore più vicino

Informazioni Risultati

Elemento	Valore
Aree_Verdi_Casalecchio_OSM	
name	Parco della Chiusa
(Derivato)	
(Azioni)	
id	way/260065267
@id	way/260065267
access	NULL
fixme	NULL
leisure	park
name	Parco della Chiusa
openin...	NULL
ref:cori...	1411

Modalità Layer Corrente

Passa a Impostazioni per attivare Windows.

Vista Albero Windows.

Digit per localizzare (Ctrl+K)

Attiva coordinate: 1254338 5539360

Scrittura

1:22996

Ente d'ingrandimenti: 100%

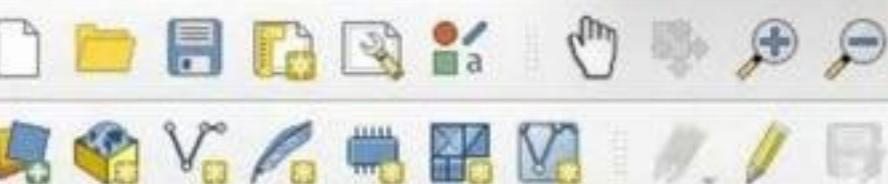
Orientazione: 0,0 °

Visualizza

14:15

12/06/2025

Scrivi qui il testo da cercare.



Browser

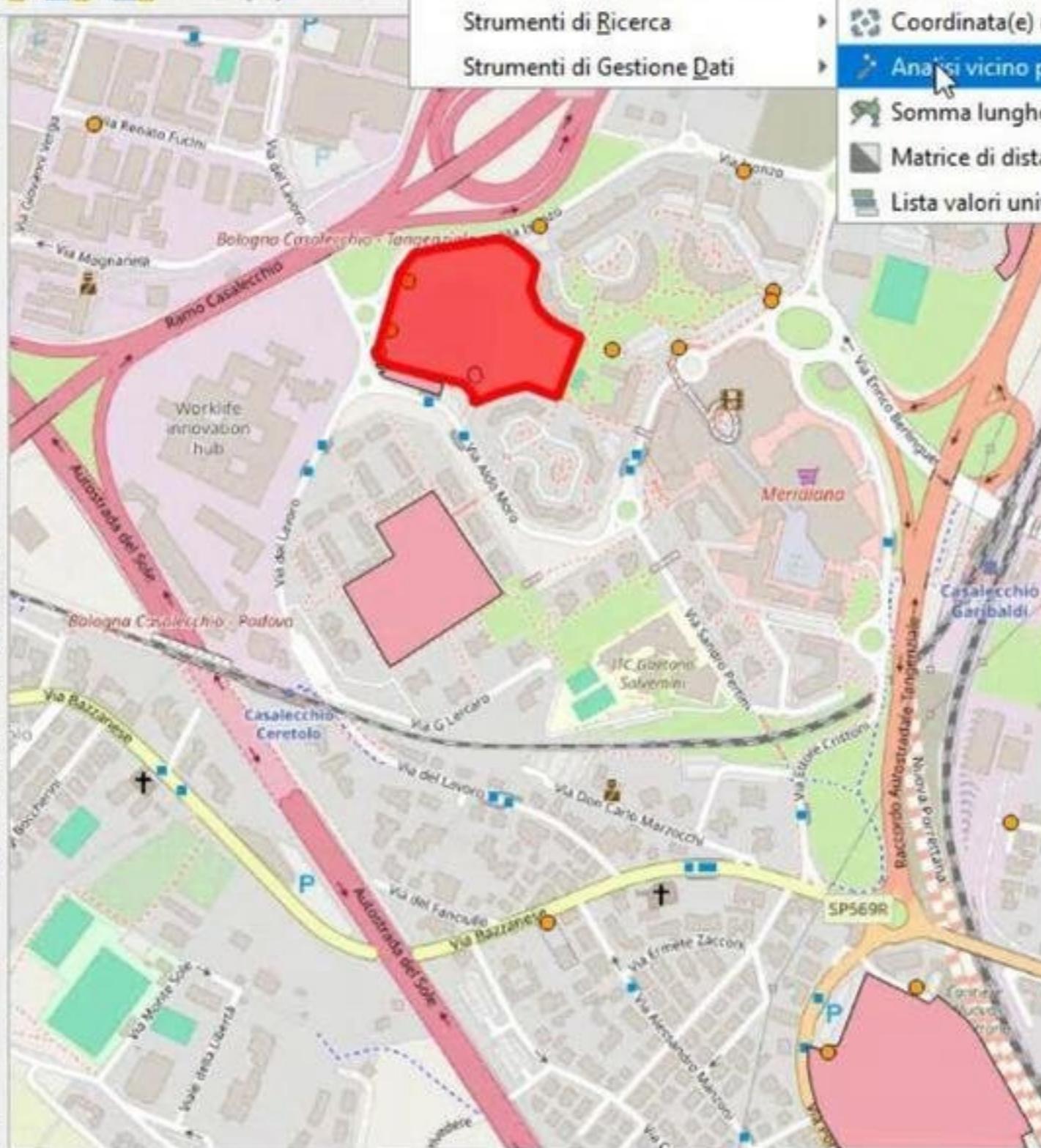


-  Cloud
 -  Scenes
 -  SensorThings
 -  Vector Tiles
 -  XYZ Tiles
 -  google earth
 -  Mapzen Globetile
 -  OpenStreetMap
 -  WCS
 -  WFS / OGC API

Layer



-  **Ingressi_Aree_Ve**
 -  *Ingressi_Aree_Ve*
 -  *Ingressi_Aree_Ve*
 -  *Ingressi_Aree_Ve*
 -  **Area Verdi Casa**
 -  *Aree_Verdi_Cata*
 -  *Aree_Verdi_Terre*
 -  *Aree_Verdi_Aqui*
 -  *google earth*
 -  **OpenStreetMap**



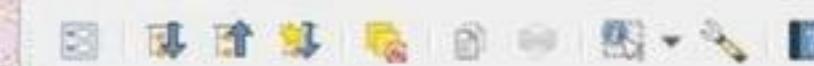
- Strumenti di Analisi
 - Strumenti di Geoprocessing
 - Strumenti di Geometria
 - Strumenti di Ricerca
 - Strumenti di Gestione Dati

- ▶  Basic statistics for fields...
 - ▶  Conta i punti nel poligono...
 - ▶  Intersezione di linee...
 - ▶  Coordinata(e) media...
 - ▶  **Analisi vicino più prossimo**
 - ▶  Somma lunghezze linea...
 - ▶  Matrice di distanze...
 - ▶  Lista valori univoci...

un raster (appartenenza vicina)

- ▼  Analisi sui vettori
 -  Analisi vicino più prossimo
 -  Distanza dal punto centrale più vicino (linea di c
 -  Distanza dall'hub più vicino (punti)
 - ▼  Cartografia
 -  Allinea punti agli elementi
 - ▼  Generale vettore
 -  N.B. In questo menu non sono presenti i comandi

Informazioni Risultati



Elemento	Valore
▼ Aree_Verdi_Casalecchio_OSM	
▼ name	NULL
► (Derivato)	
► (Azioni)	
id	way/260065272
@id	way/260065272
access	NULL
fixme	NULL
leisure	park
name	NULL
openin...	NULL
ref:cori...	1411
source	http://dati.emilia-romagna.it/dato/item/1

Modalità Layer Corrente

Passa a Impostazioni per attivare

Vista Albero 

www.english-test.net

dimenzi 100% rotazione 0,0 ° Visualizza

Digit per localizzare (Ctrl+K)

Internal coordinates: 1253907 5541241

icak 1:11498

ente d'ingrandimento 100%

Rotazione 0,0 °

Visualizza



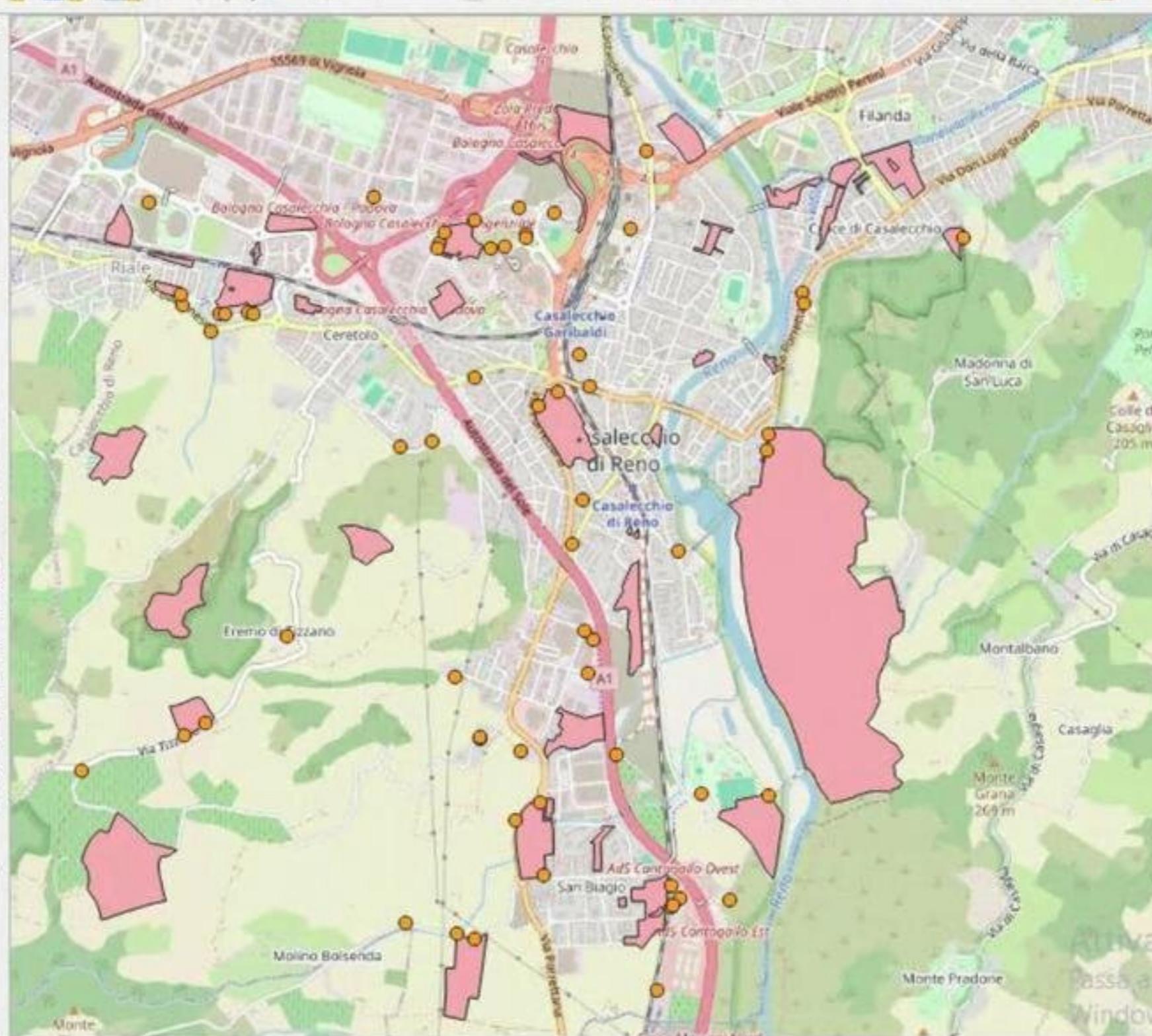


Browser

- Cloud
- Scenes
- SensorThings
- Vector Tiles
- XYZ Tiles
- google earth
- Mapzen Global
- OpenStreetMap
- WCS
- WFS / OGC API - Fe

Layer

- Ingressi_Aree_Ve
- Ingressi_Aree_Ve
- Ingressi_Aree_Ve
- Ingressi_Aree_Ve
- Aree_Verde_Casa
- Aree_Verde_Casa
- Aree_Verde_Terre
- Aree_Verde_Aqui
- google earth
- OpenStreetMap



Strumenti di Processing

- vicin
- Analisi raster
 - Rendi fuzzy un raster (app)
 - Analisi sui vettori
 - Analisi vicino più prossim
 - Distanza dal punto central
 - Distanza dall'hub più vicin
 - Cartografia
 - Allinea punti agli element
 - Generale vettore
 - Unisci attributi dal vettor

Informazioni Risultati

Elemento	Valore

Modalità Layer Corrente

Passa a impostazioni per attivare

Vista Albero

Digita per localizzare (Ctrl+K)

Coordinate: 1251017 5537710

Scale: 1:45992

Ente d'ingrandimento: 100%

Rotazione: 0,0 °

Visualizza



Scrivi qui il testo da cercare.



14:21

12/06/2025



*Progetto Senza Titolo — QGIS

Progetto Modifica Visualizza Layer Impostazioni Plugins vettore Raster Database vveo mesn Processing Guida

Digitare per localizzare (Ctrl+K)

Coordinate: 1488009 5212162

Scalare: 1:41787

Ente d'ingrandimento: 100%

Orientazione: 0,0 °

Visualizza

Scrivi qui il testo da cercare.

Strumenti di Processing

riproiett

Usati di recente

Riproietta layer

Generale vettore

Definisci la proiezione della

Riproietta layer

Informazioni Risultati

Elemento Valore

Modalità Layer Corrente

Vista: Albero

14:29

12/06/2025

Progetto Modifica Visualizza Layer Impostazioni Plugins vettore Raster Database vveo vmesn Processing Guida

Digitare per localizzare (Ctrl+K)

Scrivi qui il testo da cercare.

Coordinate: 1255572 5538834

Scalare: 1:15894

Ente d'ingrandimento: 100%

Orientazione: 0,0 °

Visualizza

Strumenti di Processing

riproiett

- Usati di recente
 - Riproietta layer
- Generale vettore
 - Definisci la proiezione dello Shapefile
 - Riproietta layer

Informazioni Risultati

Elemento	Valore
Areे Verdi Casalecchio_OSM	
name	Parco della Chiusa
(Derivato)	
(Azioni)	
id	way/260065267
@id	way/260065267
access	NULL
fixme	NULL
leisure	park
name	Parco della Chiusa
opening_hours	NULL
ref:corine_land...	1411

Modalità Layer Corrente

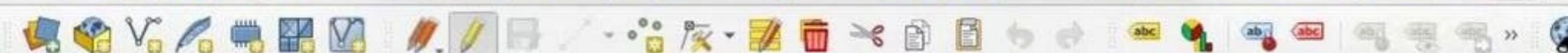
Attiva Windows

Passa a Impostazioni per attivare Windows.

Vista Albero

14:46

12/06/2025



Browser

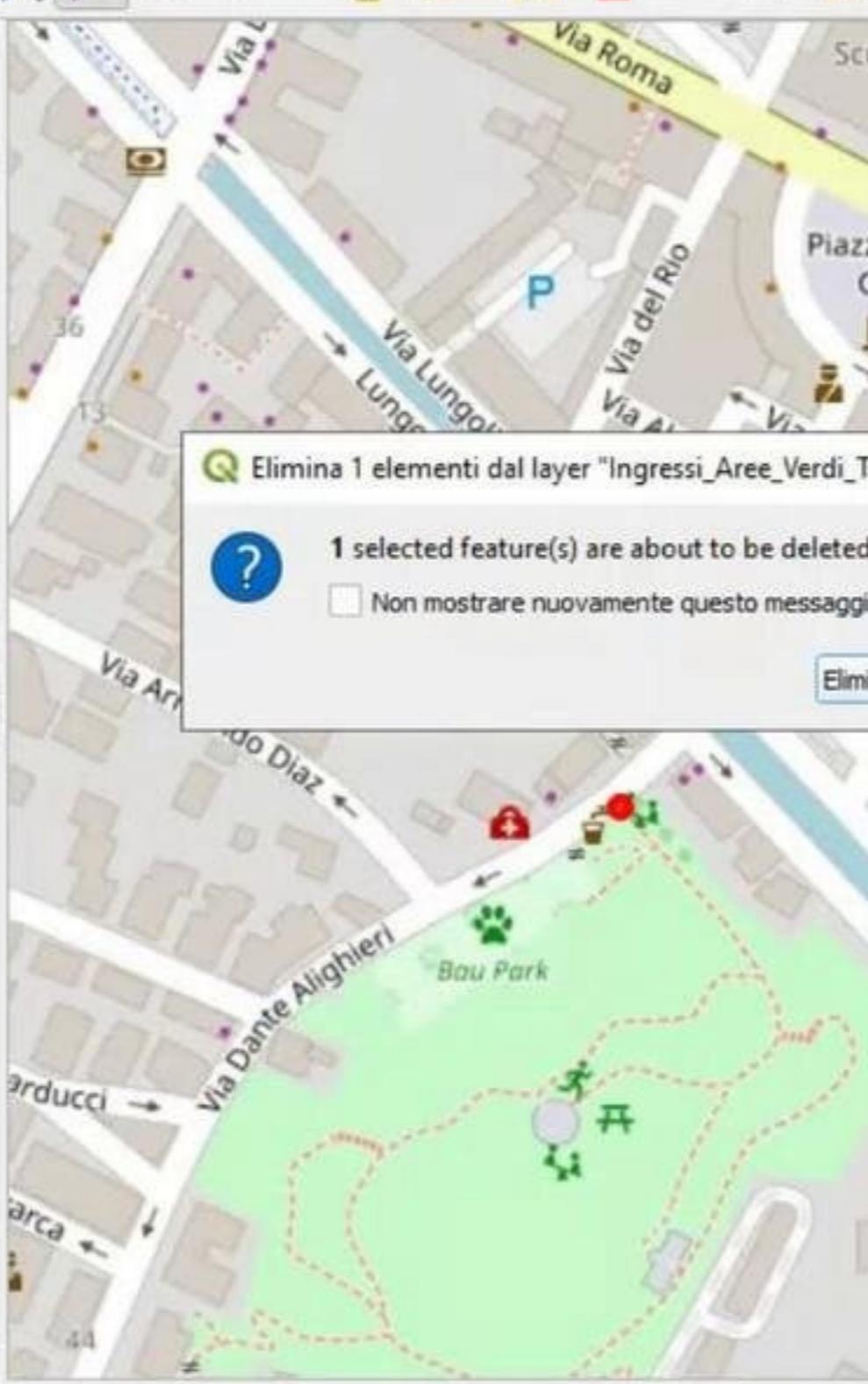
Browser

- C:\
- GeoPackage
- Spatialite
- PostgreSQL
- Progetto_ISPRA
 - public
 - Alberi Catanzaro
 - area_blu
 - area_verde
 - Aree Complesse Catanzaro

Layer

Layer

- Totale_Ingressi_Casalecchio
- Totale_Ingressi_Catanzaro
- Totale_Ingressi_Terracina
- Totale_Ingressi_Aquila
- Ingressi_Aree_Verdi_Casalecchio
- Ingressi_Aree_Verdi_Catanzaro
- Geometrie riparate
- Geometrie riparate
- **Ingressi_Aree_Verdi_Terracina**
- Ingressi_Aree_Verdi_Aquila
- Aree_Verdi_Casalecchio_OSM
- Aree_Verdi_Catanzaro_OSM
- Aree_Verdi_Terracina_OSM
- Aree_Verdi_Aquila_OSM
- google earth



Strumenti di Processing

Strumenti di Processing

- ripar
- Usati di recente
- Generale vettore
 - Ripara Shapefile
- Geometria vettore

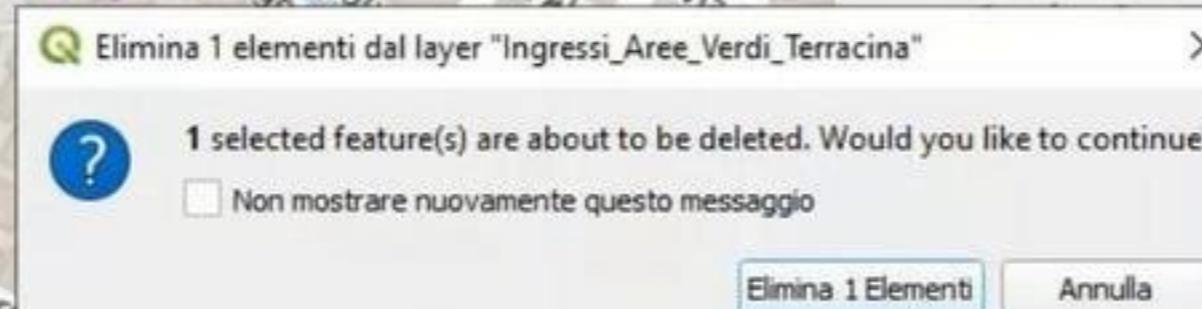
Informazioni Risultati

Risultato

ULL

code/5808537445

code/5808537445



access	yes
addr:city	NULL
addr:country	NULL
addr:housename	NULL
addr:housenumber	NULL
addr:postcode	NULL
addr:street	NULL
alt_name	NULL
barrier	gate
bicycle	NULL
foot	NULL
historic	NULL
horse	NULL
inland	NULL

Attiva Windows

Passa a Impostazioni per attivare Windows.

Vista Albero

Continuazione sviluppo delle classi per le API REST e dei relativi servizi

In questa fase del training si proseguirà lo sviluppo dell'architettura a strati del progetto, con particolare attenzione alla costruzione delle API REST e dei corrispondenti servizi di business logic nel backend Java/Spring.

Obiettivi principali:

- Consolidare e completare le operazioni CRUD (Create, Read, Update, Delete) per le principali entità del progetto, in particolare quelle legate alla gestione delle aree verdi, dei poligoni associati e di altri elementi territoriali.
- Sviluppare nuove API REST personalizzate per supportare operazioni specifiche richieste dal flusso applicativo, ad esempio:
 - Inserimento di poligoni selezionati su mappa
 - Aggiornamento di metadati ambientali associati a un'area
 - Recupero filtrato di aree sulla base di criteri spaziali o descrittivi
- Implementare la comunicazione tra controller, service e repository, secondo i principi della separazione delle responsabilità.

Attività tecniche previste:

- Progettazione e stesura di nuovi controller REST, annotati con `@RestController` e `@RequestMapping`.
- Definizione di endpoint REST con annotazioni come `@GetMapping`, `@PostMapping`, `@PutMapping` e `@DeleteMapping`.
- Implementazione dei metodi nei service, che conterranno la logica di validazione e

Salvataggio automatico H Overpass_OpenStreetMap_API_VEBS.docx -... • Salvato in questo PC

File Home Inserisci Disegno Progettazione Layout Riferimenti Lettere Revisione Visualizza Guida Commenti

cronologia modifiche, utenti, ecc.)

2. Modificare dati

Si può contribuire a OSM (da utente registrato) con:

- iD editor: editor online facile da usare, direttamente dal sito OSM
- JOSM: editor avanzato desktop (Java) per modifiche massive
- Field Papers o app mobile (Vespucci, StreetComplete)

3. Scaricare dati

Come ottenere dati OSM:

- In formato .osm (XML) tramite Geofabrik
- In formato GeoJSON, KML usando [Overpass Turbo](#)
- Con strumenti GIS come QGIS (tramite plugin QuickOSM)

4. Sviluppo software

È possibile integrare OSM in app/web usando:

- Leaflet.js + Tile Server: per visualizzare mappe
- OpenLayers: libreria GIS frontend potente
- osmnx (Python): per analisi di reti stradali
- Nominatim API: per geocoding/reverse geocoding
- Overpass API: per query personalizzate sui dati
- PostGIS: puoi importare i dati OSM nel tuo DB spaziale per analisi avanzate

Pagina 2 di 7 4 di 1055 parole Completamento del testo: attivato Focus

15:46 12/06/2025

Salvataggio automatico H 9:00 Overpass_OpenStreetMap_API_VEBS.docx -... • Salvato in questo PC FC - □ X

File Home Inserisci Disegno Progettazione Layout Riferimenti Lettere Revisione Visualizza Guida Commenti

Approfondimento dell'uso dell'Overpass API

Overpass API è un'API che consente di interrogare i dati di OpenStreetMap (OSM) in modo avanzato e personalizzato.

È progettata per estrarre *solo i dati che servono*, senza dover scaricare l'intero database OSM.

Non è un'API REST "standard" tipo GET /endpoint, ma un motore di query geospatiali su un database OSM.

Funzionalità di Overpass Turbo:

- - Creazione ed esecuzione di query Overpass QL
- - Visualizzazione su mappa degli oggetti ottenuti
- - Esportazione in vari formati (GeoJSON, KML, GPX, ecc.)

Overpass API lavora su una copia aggiornata del database OSM e consente di interrogarlo tramite due linguaggi:

1. Overpass QL (query language compatto)
2. Overpass XML (più verboso, in formato XML)

Concetti base

- Gli oggetti OSM sono: node, way, relation.
- Ogni oggetto ha tag del tipo chiave=valore (es: amenity=school).
- Si può filtrare per:
 - Tag (es: tutti i parchi)
 - Area geografica (box lat/lon)
 - Tipo di oggetto (nodo/poligono)
 - Tempo (modifiche recenti)

Pagina 3 di 7 4 di 1055 parole Completamento del testo attivato Focus 80% 15:46 12/06/2025

Windows Start File Explorer Mail Internet Explorer Microsoft Edge Word PowerPoint Google Chrome

Salvataggio automatico

Overpass_OpenStreetMap_API_VEBS.docx -... • Salvato in questo PC

File Home Inserisci Disegno Progettazione Layout Riferimenti Lettere Revisione Visualizza Guida

Commenti

Overpass API lavora su una copia aggiornata del database OSM e consente di interrogarlo tramite due linguaggi:

1. Overpass QL (query language compatto)
2. Overpass XML (più verboso, in formato XML)

Concetti base

- Gli oggetti OSM sono: node, way, relation.
- Ogni oggetto ha tag del tipo chiave=valore (es: amenity=school).
- Si può filtrare per:
 - Tag (es: tutti i parchi)
 - Area geografica (box lat/lon)
 - Tipo di oggetto (nodo/poligono)
 - Tempo (modifiche recenti)

Esempio di Query in Overpass QL

```
[out:json][timeout:25];
// seleziona tutti i parchi in un'area o specificando le coordinate dell'area
```

Pagina 3 di 7 1055 parole

Focus

15:47 12/06/2025

Progetto Modifica Visualizza Layer Impostazioni Plugins vettore Raster Database Web Mosaico Processing Guida

Digitare per localizzare (Ctrl+K) Coordinati: 1492773 5214422 Icalz: 1:21869 Ente d'ingrandimenti: 100% Rotazione: 0,0 ° Visualizza

Scrivi qui il testo da cercare.

Strumenti di Processing

ripar

- Usati di recente
- Generale vettore
- Ripara Shapefile
- Geometria vettore

Informazioni Risultati

Elemento Valore

Ingressi_Aree_Verdi_Aquila

- alt_name NULL

Modalità Layer Corrente

Vista Albero

15:49 12/06/2025

The screenshot shows the QGIS application interface with the following details:

- Toolbar:** Standard QGIS tools for selection, zoom, pan, and data manipulation.
- Browser:** Shows the project structure with layers from C:\, GeoPackage, SpatiaLite, and PostgreSQL, including the "Progetto_ISPRA" project with its sub-layers: public, Alberi Catanzaro, area_blu, area_verde, and Aree Complesse Catanzaro.
- Layer List:** A list of layers currently loaded in the map canvas, including Totale_Ingressi_Catanzaro, Totale_Ingressi_Terracina, Totale_Ingressi_Aquila, Ingressi_Aree_Verdi_Casalecchio, Ingressi_Aree_Verdi_Catanzaro, Geometrie riparate, Ingressi_Aree_Verdi_Terracina, Ingressi_Aree_Verdi_Aquila (selected), Aree_Verdi_Casalecchio_OSM, Aree_Verdi_Catanzaro_OSM, Aree_Verdi_Terracina_OSM, Aree_Verdi_Aquila_OSM, google earth, and OpenStreetMap.
- Map Canvas:** Displays a detailed map of the Aquila area, Italy, showing streets, buildings, and green spaces. A red dot marks a specific location near the "Ex Caserma Francesco Rossi" and the "Cimitero monumentale dell'Aquila".
- Processing Panel:** Shows the "Strumenti di Processing" (Processing Tools) panel with a search bar for "ripar" and a list of recent tools: Usati di recente, Generale vettore, Ripara Shapefile, and Geometria vettore.
- Information Panel:** Displays information about the selected layer, "Ingressi_Aree_Verdi_Aquila", specifically the "alt_name" field which is NULL.
- Bottom Status Bar:** Shows the coordinate (1492773, 5214422), scale (1:21869), and rotation (0,0 °) of the map.
- System Icons:** Standard Windows system icons for file, folder, and search.

```

//es: " (45.42,10.95,45.45,10.98)"
(
node["leisure"="park"](45.42,10.95,45.45,10.98);
way["leisure"="park"](45.42,10.95,45.45,10.98);
relation["leisure"="park"](45.42,10.95,45.45,10.98);
);

[out:json][timeout:25];
//o passandogli il nome es: "Catanzaro"
area["name"="Catanzaro"]["boundary"="administrative"]["admin_level"="8"]->.catanzaro;
(
node["leisure"="park"](.catanzaro);
way["leisure"="park"](.catanzaro);
relation["leisure"="park"](.catanzaro);
)->.parchi;

// trova i nodi con barrier=gate o entrance nelle aree verdi (via map_to_area)
.parchi map_to_area -> .aree_parco;

(
node["barrier"="gate"](.aree_parco);
node["barrier"="entrance"](.aree_parco);
);

out body;
>;
out skel qt;
// oppure out geom;

```

Spiegazione:

- `out:json`: formato di output (può essere anche XML)
 - `timeout:25`: tempo massimo di esecuzione in secondi

Salvataggio automatico FC

aree verdi_AC_10_04_2025 + Filippo_re... • Salvato in questo PC

File Home Inserisci Disegno Progettazione Transizioni Animazioni Presentazione Registra Revisione Visualizza Guida

Incolla Diapositive Appunti

Paragrafo

Modifica

Dettatura

Riservatezza

Add-ins

Designer

Integrazione delle informazioni disponibili

Nonostante l'obiettivo sia quello di procedere con un approccio bottom-up partendo da fonti ufficiali, la sovrapposizione con layer estratti da OpenStreetMap e da Copernicus Urban Atlas Land Cover/Land Use 2018 (UA) diventa utile per integrare il contenuto informativo dei layer resi disponibili dalle Amministrazioni locali.

Inoltre, se i dati dei punti di accesso non fossero disponibili, si potrebbero identificare attraverso l'analisi spaziale nei seguenti modi:

- Considerare solo i nodi estratti da OpenStreetMap con la parola chiave "barrier" ed i tag "gate" e "entrance", localizzati all'interno o sul perimetro dell'area verde considerata
- Considerare l'intersezione tra il layer "strade" e "aree verdi urbane", scaricati da OpenStreetMap.

Laddove le condizioni precedenti non possono essere soddisfatte, una soluzione potrebbe essere rappresentata dalla individuazione di punti ogni 100 metri lungo il perimetro.

Fare clic per inserire le note

Diapositiva 7 di 11 Accessibilità: verifica Note 15:51 12/06/2025