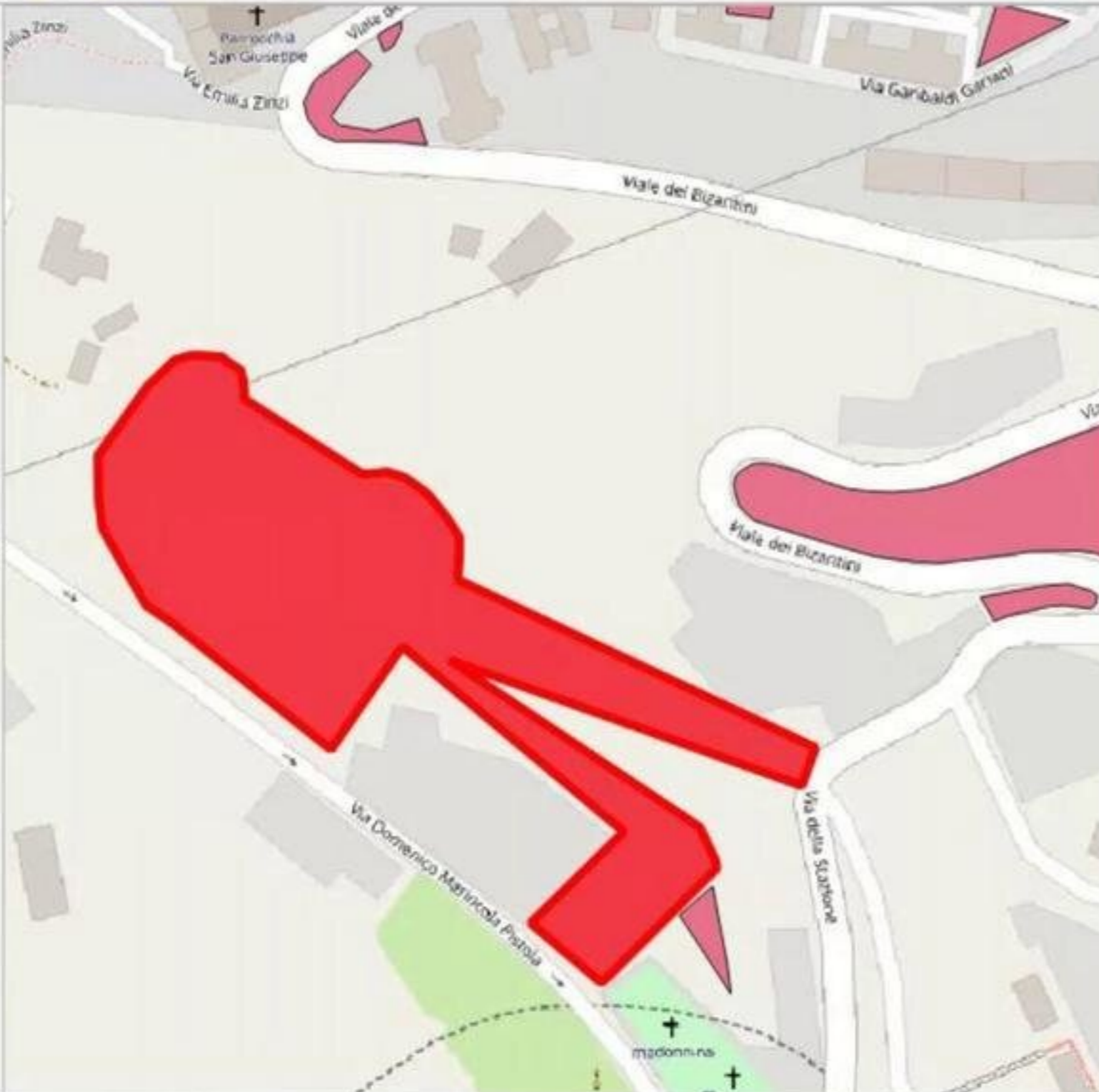
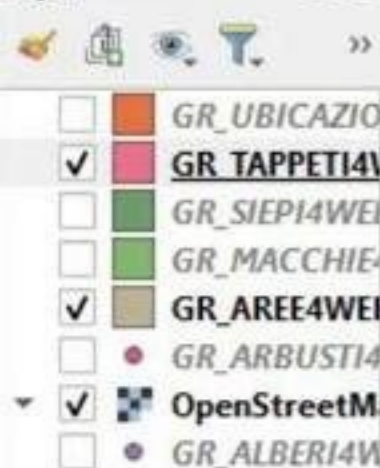




Browser



Layer



Strumenti di Processing



Search...

- Usati di recente
- Analisi raster
- Cartography
- Controlla Geometrie

Search QMS

Search string...

Filter

[Download geodata](#) for your project

Informazioni Risultati



Elemento	Valore
IDENTIF	9033
TIPOLOGIA	Prato naturale
UBICAZIONE	CATANZARO
RIFCARTO	
LUOGO	Aiuola
CIRCOSCR	Il Centro Storico, Stadio,

Modalità Layer Corrente

Vista Albero

Digita per localizzare (Ctrl+K)

Interni coordinat 638665,9 4306652,3

cal: 1:2220

ante d'ingrandiment 100%

otazion 0,0 °

Visualizza



Administrator



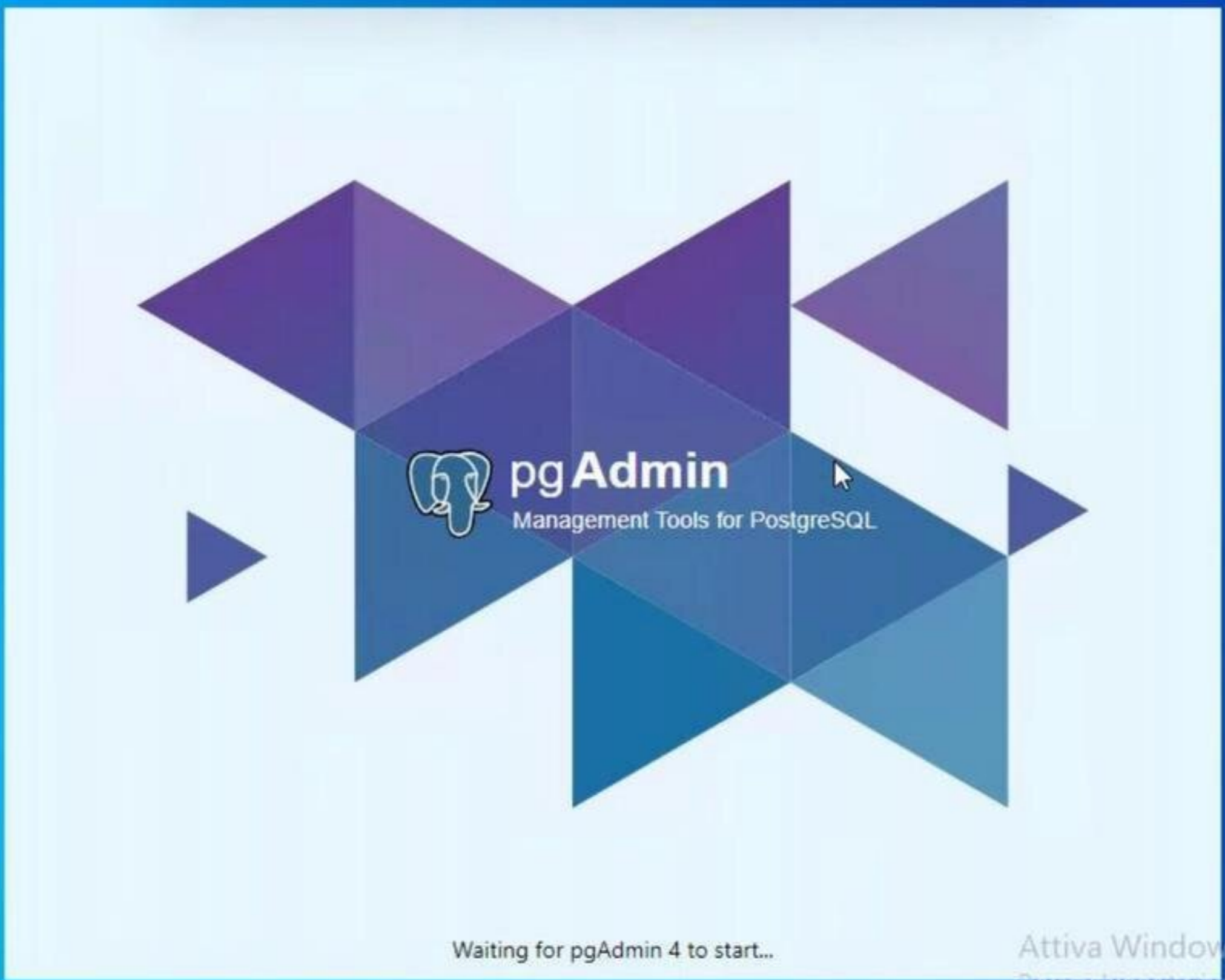
Questo PC



Rete



Cestino



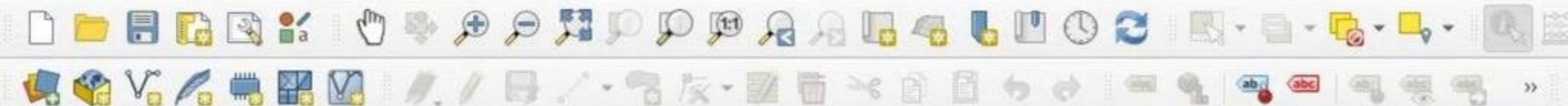
Waiting for pgAdmin 4 to start...

Attiva Windows
Passa a impostazioni per attivare Windows.

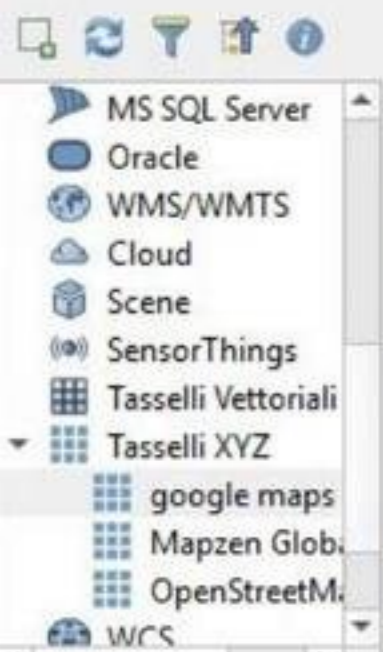


Scrivi qui il testo da cercare.





Browser



Layer



Strumenti di Processi

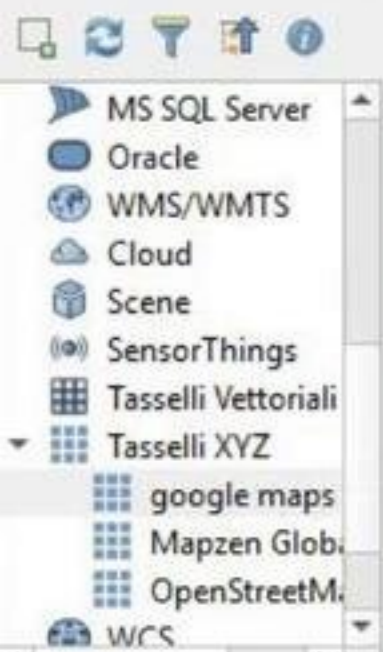


Informazioni Risultati





Browser



Layer



Strumenti di Processi

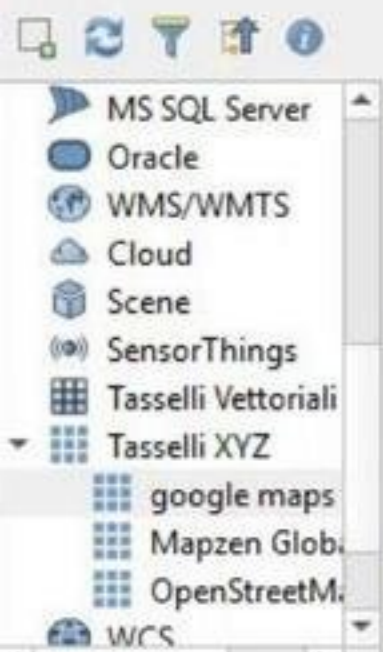


Informazioni Risultati





Browser



Layer



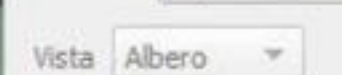
Strumenti di Processi



Informazioni Risultati



Modalità Layer Corrente



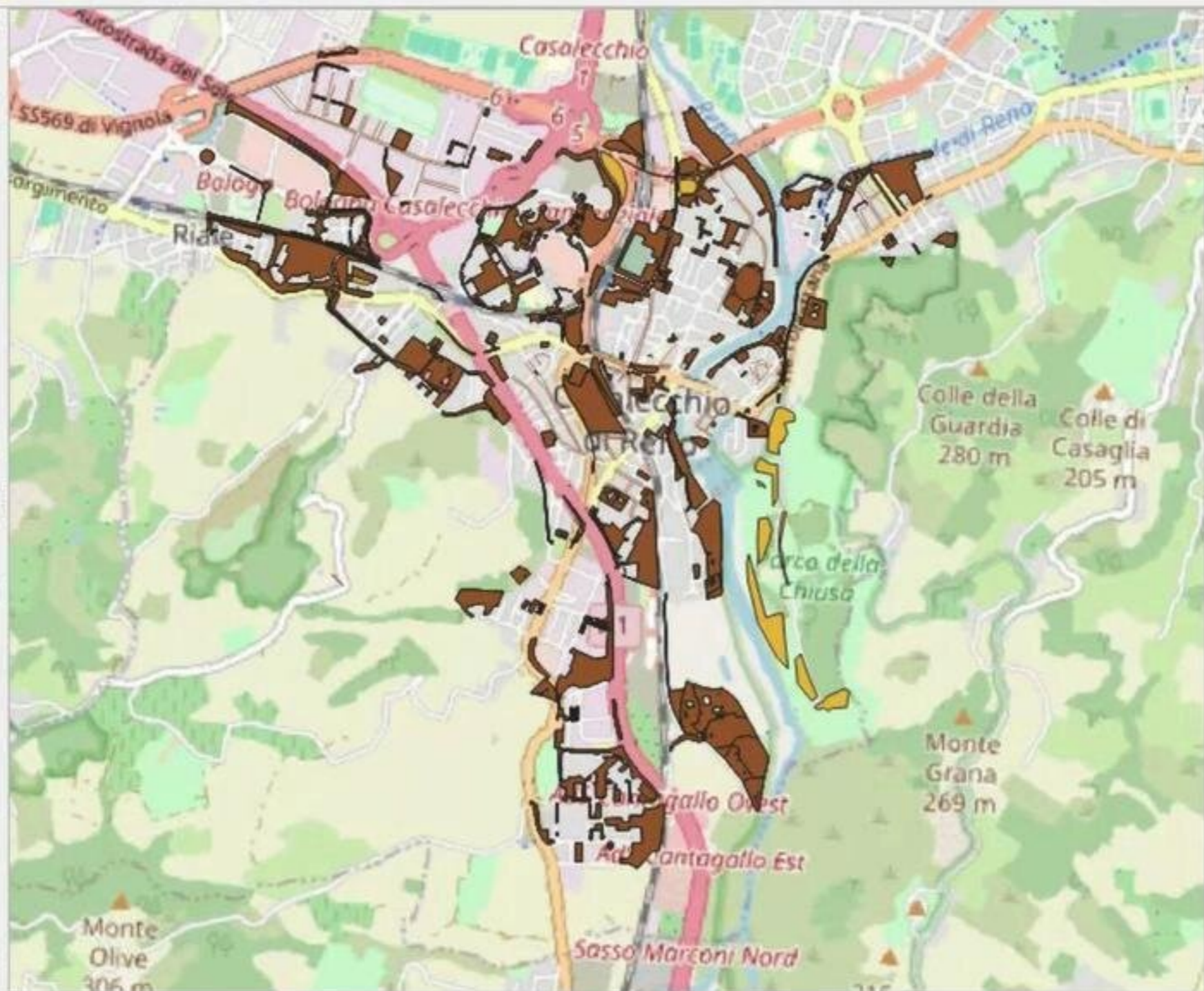


Browser

- MS SQL Server
- Oracle
- WMS/WMTS
- Cloud
- Scene
- SensorThings
- Tasselli Vettoriali
- Tasselli XYZ
- google maps
- Mapzen Global Terrain
- OpenStreetMap
- WCS

Layer

- ☒ Verde Casalecchio
- ☒ Talon
- ☒ Siepi
- ☒ Differenza_1
- ☒ Differenza
- ☒ Bordature
- ☒ atlante
- ☒ aree-verdi-piano-sp
- ☒ aree-verdi-piano-sp
- ☒ aree-verdi-piano
- ☒ aree-verdi-180605
- ☒ Aree giochi



Strumenti di Processi

- Search...
- Usati di recente
- Analisi raster
- Cartography
- Controlla Geo

Search QMS

Search string...

Download

Informazioni Risultati

Elemento

Modalità Layer Corrente

Vista Albero

Digita per localizzare (Ctrl+K)

Coordinate 200132 4930464

Scala 1:35517

ente d'ingrandimento 100%

Rotazione 0,0°

Visualizza



Browser

- MS SQL Server
- Oracle
- WMS/WMTS
- Cloud
- Scene
- SensorThings
- Tasselli Vettoriali
- Tasselli XYZ
- google maps
- Mapzen Global Terrain
- OpenStreetMap
- WCS

Layer

- aree-verdi-piano-sp
- aree-verdi-piano
- aree-verdi-180605
- Aree giochi
- Area cani
- Anomalie
- 7 Airole stradali
- ☒ 6 Parchi
- 5 Aree demaniali
- 4 Aree verdi edifici p
- 2 Aree verdi centri sp
- ☒ OpenStreetMap



Strumenti di Processi

- Search...
- Usati di recente
- Analisi raster
- Cartography
- Controlla Geo
- Search QMS
- Search string...
- Download

Informazioni Risultati

- Elemento
- Modalità Layer Corrente
- Vista Albero

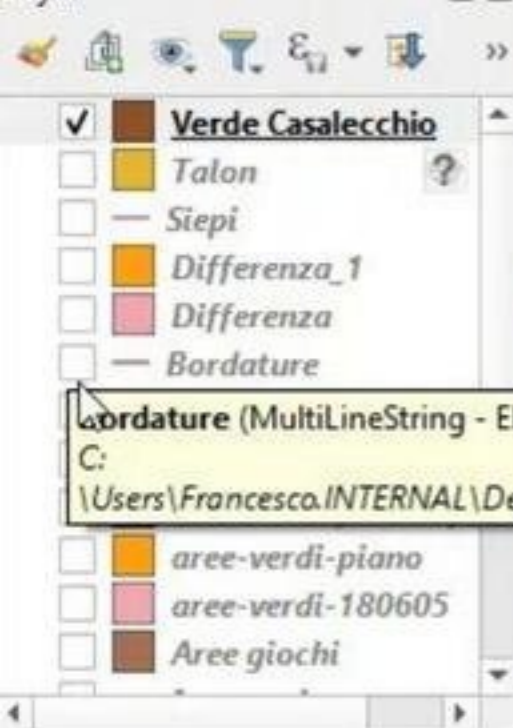
Coordinate: 201926,3 4932215,5 Scale: 1:1110 Ante d'ingrandimento: 100% Rotazione: 0,0 ° Visualizza



Browser



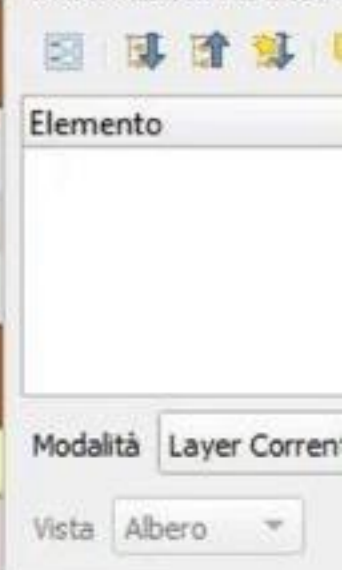
Layer



Strumenti di Processing



Informazioni Risultati





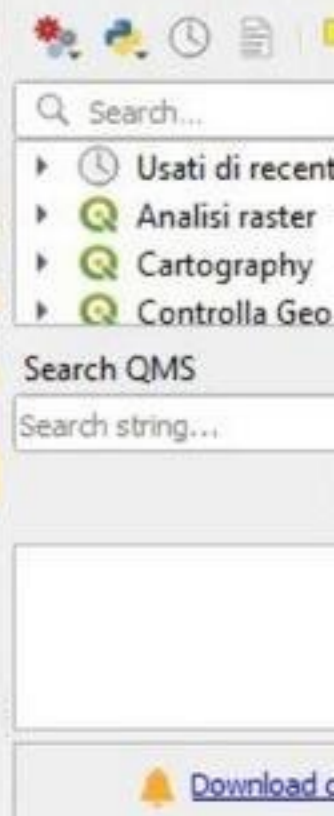
Browser



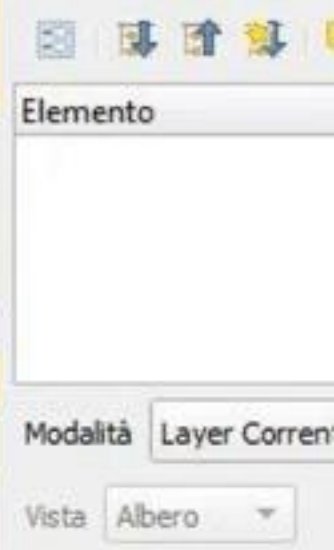
Layer



Strumenti di Processing



Informazioni Risultati



Q Digita per localizzare (Ctrl+K)

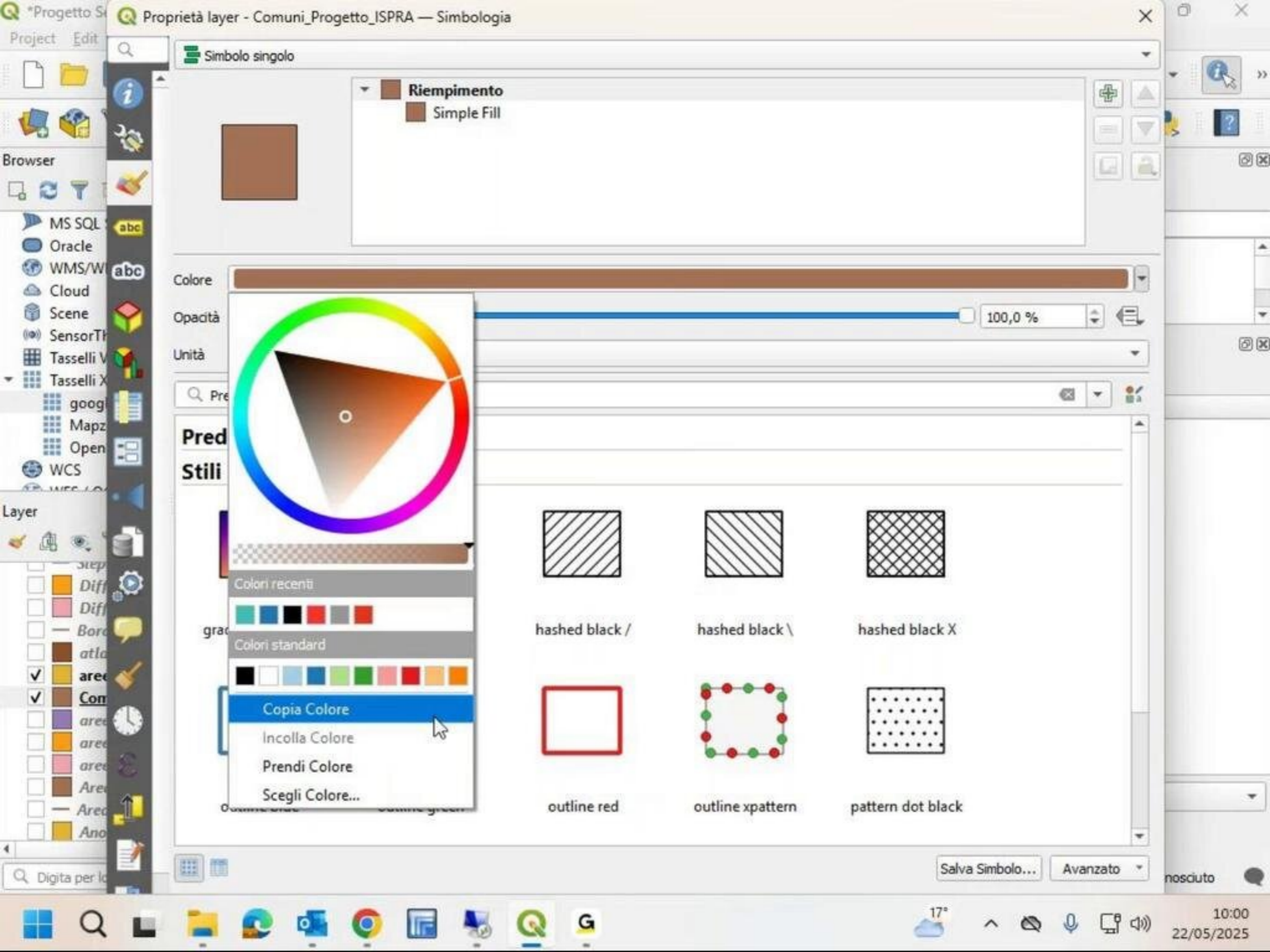
Coordinate: 201800,9 4932295,7

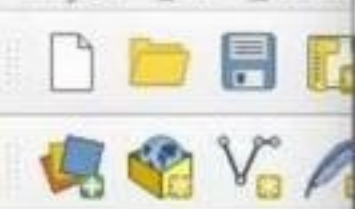
Scala: 1:2220

Ente d'ingrandimento: 100%

Rotazione: 0,0°

Visualizza





Browser



- MS SQL Server
- Oracle
- WMS/WMTS
- Cloud
- Scene
- SensorThings
- Tasselli Vettoriali
- Tasselli XYZ
 - google maps
 - Mapzen Global
 - OpenStreetMap
- WCS
- WFS LOGICAL

Layer

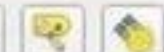
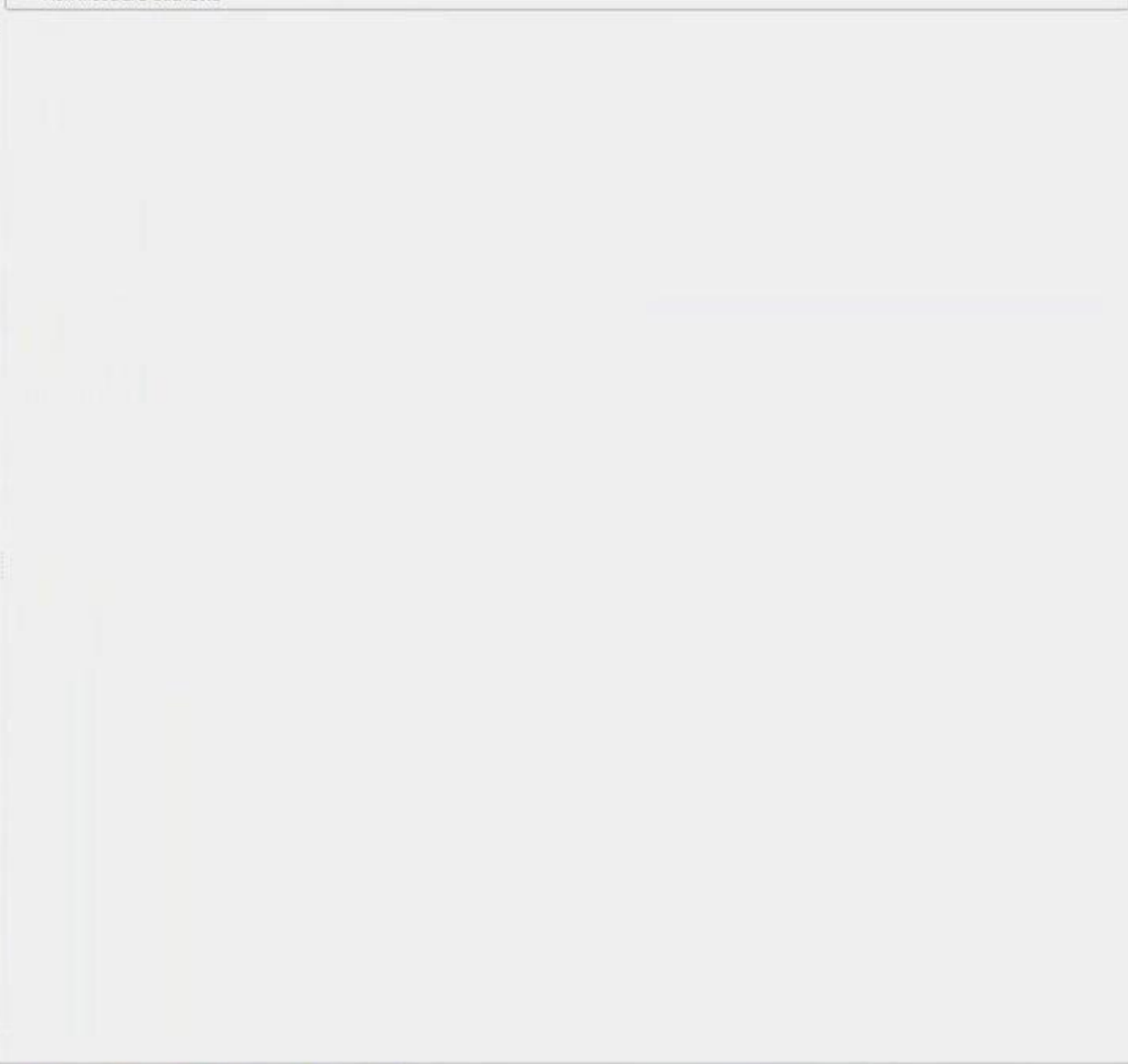


- ☐ Bordature
- ☐ atlante
- ☒ aree-verdi-p
- ☒ Comuni Pro
- ☐ aree-verdi-p
- ☐ aree-verdi-p
- ☐ aree-verdi-1
- ☐ Aree giochi
- ☐ Area cani
- ☐ Anomalie
- ☐ 7 Aiutole stro
- ☐ 6 Parchi

4

Q Digita per localizzare

Non mostrare etichette



Training on the Job - Introduzione a Spring + configurazione iniziale progetto e creazione delle classi per accesso a DB, per il trasferimento dati e altre...

1. Sviluppo classi mappatura e connessione database (Entity, DTO e Application)

Obiettivi:

- Sviluppo delle classi backend in Java

Attività previste:

- Introduzione al framework Spring MVC
- Configurazione iniziale del progetto con Maven anziché importazione diretta
- Spiegazione e sviluppo delle classi Entity
- Spiegazione e sviluppo delle classi DTO
- 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

- 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.

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;

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.

Dopo l'aver installato **JAVA** e l'IDE di sviluppo nel nostro caso **Eclipse**, la prima cosa **cosa** da fare in ogni progetto **Spring**, è fare visita al sito **Spring Initializr** (<https://start.spring.io/>) per scaricare un template di progetto pronto all'uso.



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 gestire la richiesta dell'utente.
- base di Spring anche al mondo web.

Dopo l'aver installato JAVA e l'IDE di sviluppo nel nostro caso, da fare in ogni progetto Spring, è fare visita al sito Spring Initializr (<https://start.spring.io/>) per scaricare un template di progetto pronto all'uso.

Le impostazioni del template saranno:

- Java -> 24

Le dipendenze del progetto che andremo ad aggiungere inizialmente dal sito saranno:

- Spring Data JPA
- Spring Web
- PostgreSQL Driver

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:

- o standardizzazione della struttura di un progetto compilazione;
- o 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



TOTJ



Administrator



Questo PC



Rete



Cestino



vebs.zi



DATA



QGIS 3.42.2



ERD_ISPRA.xml

Apri

Estrai tutto...

Aggiungi a Start

Condivisione

Apri con...

Dare accesso a

Ripristina versioni precedenti

Invia a

Taglia

Copia

Crea collegamento

Elimina

Rinomina

Proprietà

Attiva Windows

Passa a Impostazioni per attivare Windows.



Scrivi qui il testo da cercare.

10:29
22/05/2025



Administrator



Questo PC



vebs.zip



Rete



DATA



Cestino



QGIS 3.42.2



ERD_ISPRA.xml

Attiva Windows
Passa a Impostazioni per attivare Windows.



Qgis_Conversioni

← → ↑ ↺ ↻ ⌂ > ... Qgis_Conversioni

Cerca in Qgis_Ci 🔍

⊕ Nuovo ✂ 📄 📁 📄 📄 🗑

↕ Ordina ▾ ≡ Visualizza ▾ ... 📄 Dettagli

MONASTERIUM.

Münster.

QGIS 3.42

Münster

Recupero plugin caricati

Musica

Video

Training_on_the_job

LEZIONE_3

LEZIONE_2

11 elementi

he.qir

Reticolo_Intersezione.gpkg

Statistiche_Esagoni.gpkg

Ultima modifica

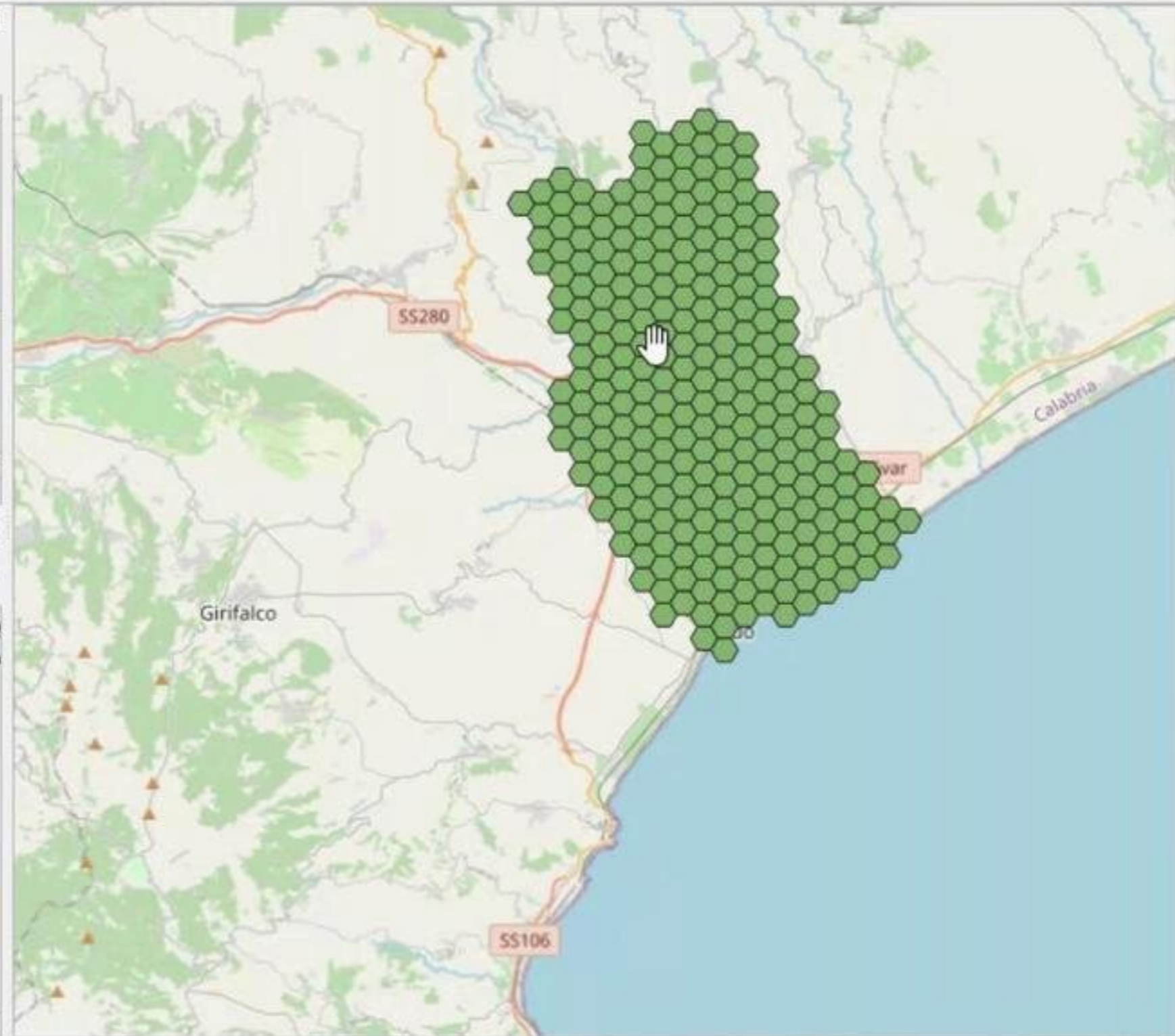
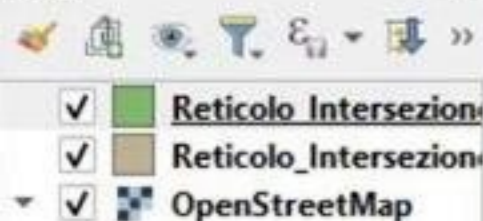
19/05/2025 17:16
19/05/2025 15:52
20/05/2025 08:40
20/05/2025 11:54
20/05/2025 10:52
14/05/2025 16:05
14/05/2025 16:05
20/05/2025 12:21
20/05/2025 08:43
21/05/2025 17:23
20/05/2025 09:08



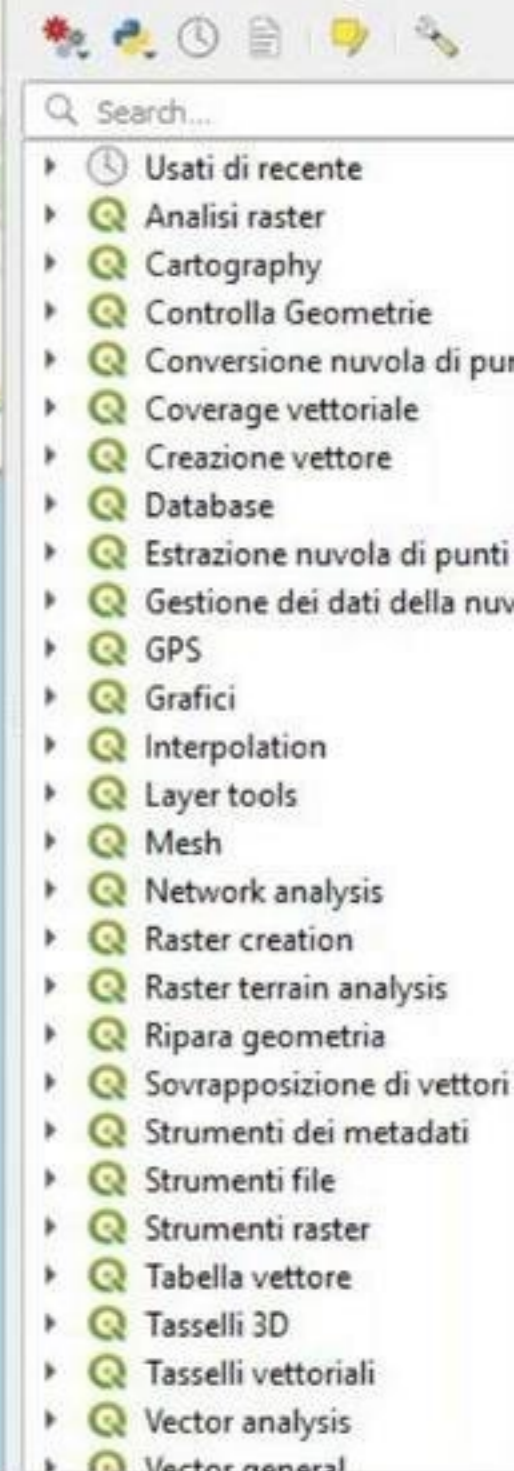
Browser



Layer



Strumenti di Processing

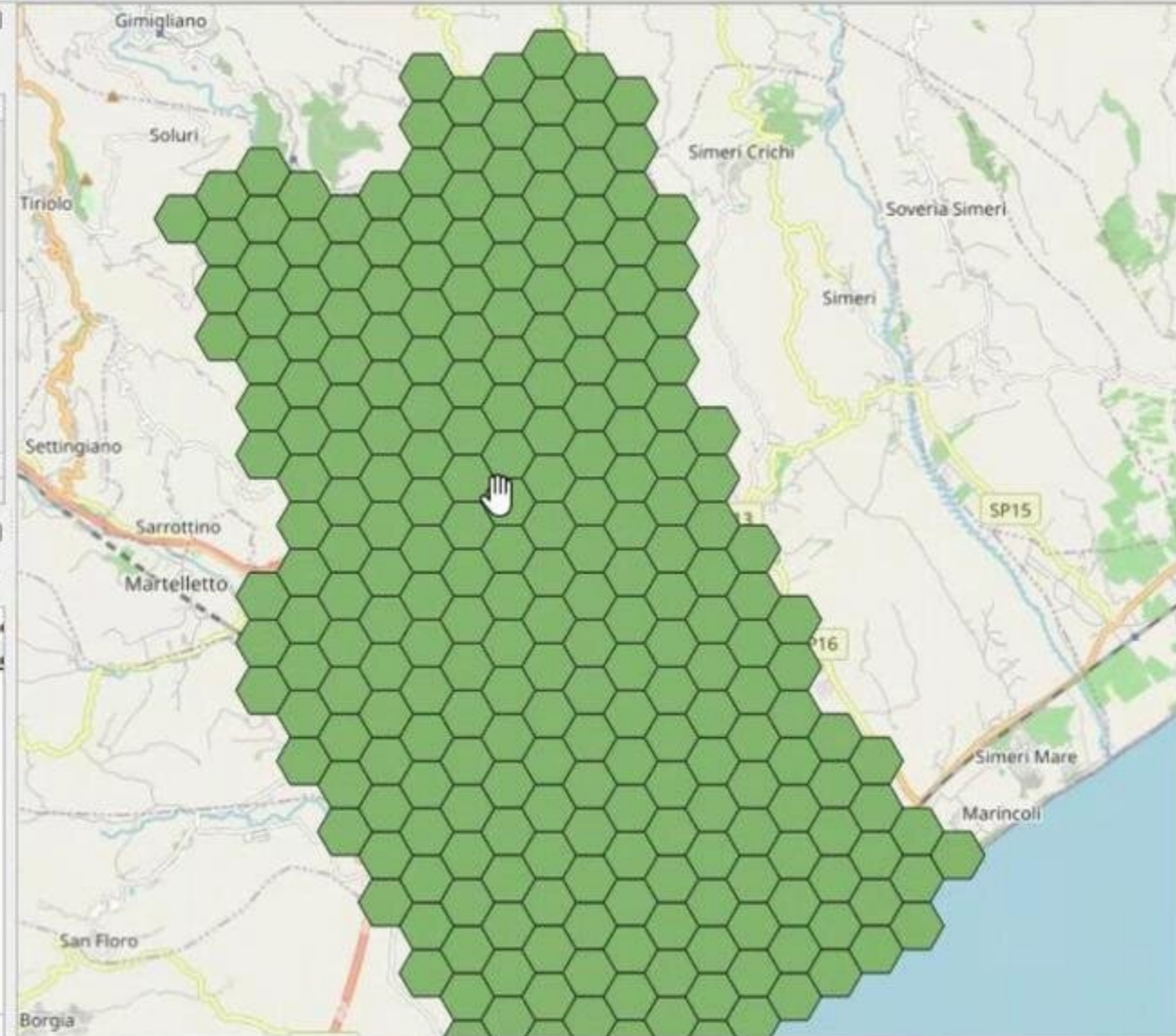




Browser



Layer



Strumenti di Processing



Q Digita per localizzare (Ctrl+K)

Coordinate: 1847311 4707934

Scala: 1:145716

Percentuale d'ingrandimento: 100%

Rotazione: 0,0 °

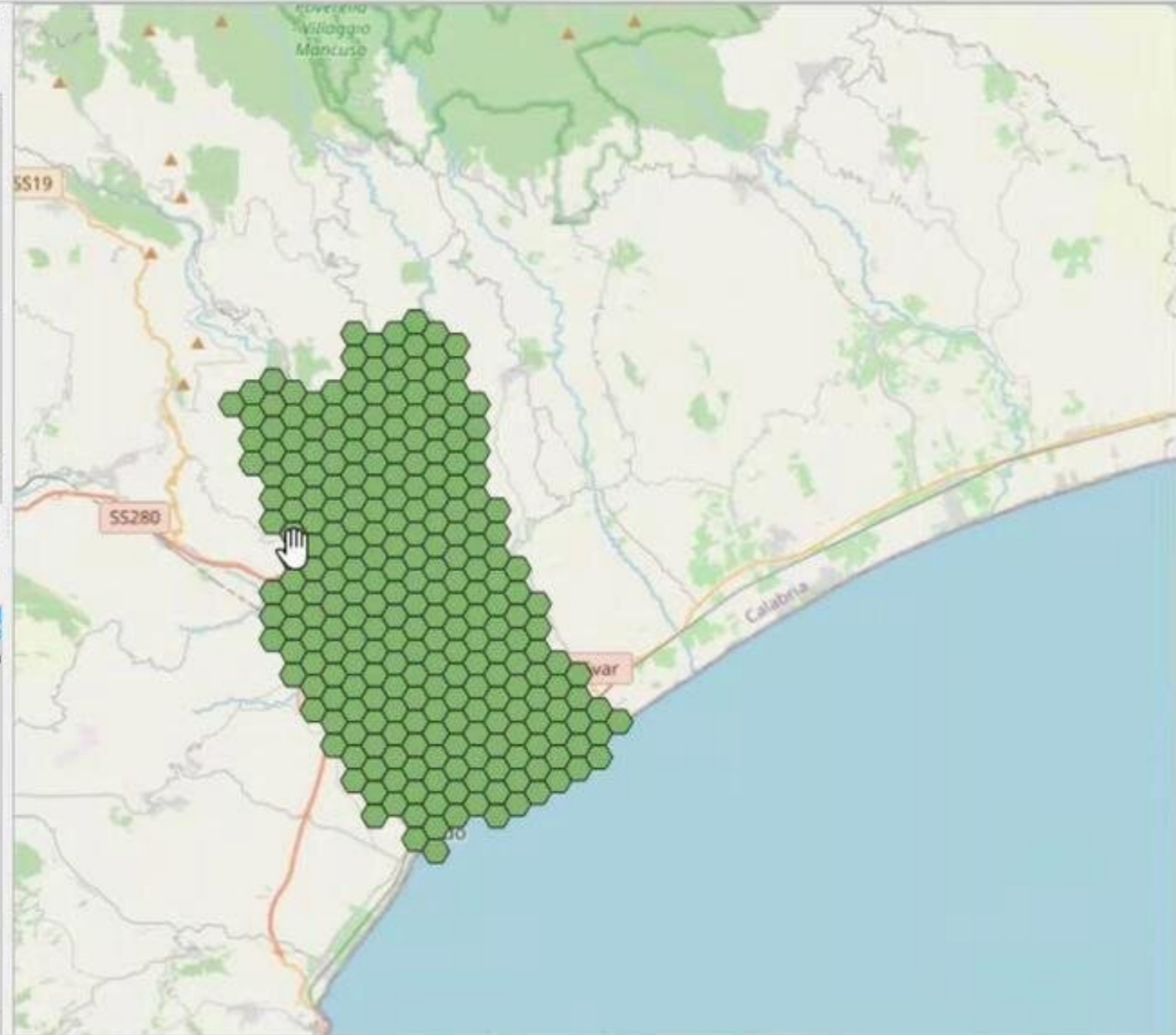
Visualizza



Browser



Layer



Strumenti di Processing



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:**