

# ESP32 GPS-TRACKER

---

*Giovanni Garofalo*

*A13002708*

*[giovanni.garofalo2@studenti.unicampania.it](mailto:giovanni.garofalo2@studenti.unicampania.it)*

*Saverio Valentino*

*A13002574*

*[saverio.valentino1@studenti.unicampania.it](mailto:saverio.valentino1@studenti.unicampania.it)*

---

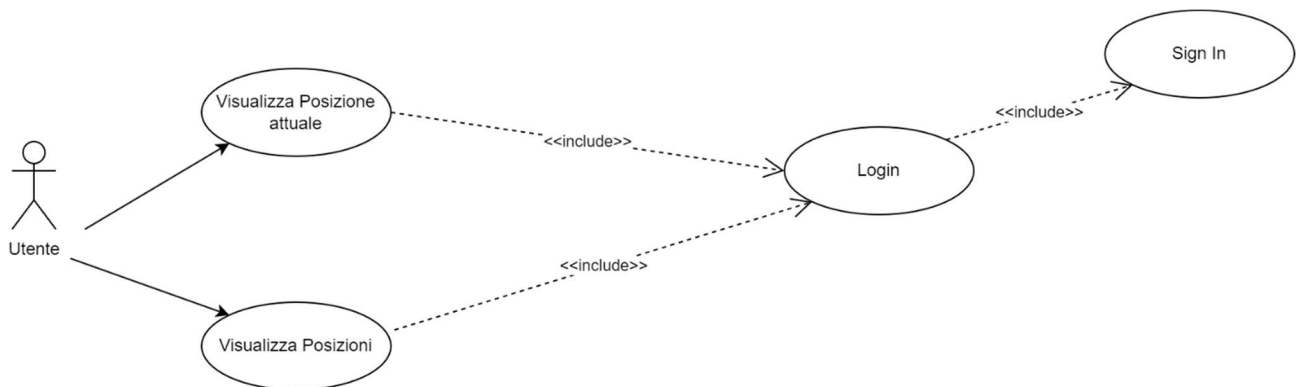
## Sommario

<b>ESP32 GPS-TRACKER .....</b>	<b>1</b>
Sommario .....	2
Introduzione .....	3
System Feature Utente .....	3
Sign In .....	3
Login .....	4
Visualizza Posizioni.....	4
Visualizza Posizione Attuale .....	5
Casi d'Uso Client-Server .....	6
System Feature Client-Server .....	6
Public Message .....	6
Stabilisci Connessione al Server.....	7
Salva Topic.....	7
Stabilisci Connessione al Database .....	8
Requisiti Hardware .....	8
Protocolli di Rete.....	9
Database .....	10
Progettazione logica.....	10
Possibili sviluppi futuri.....	11
Test dimostrativi.....	12

## Introduzione

La seguente relazione discuterà dei requisiti tecnici e le funzionalità di un GPS Tracker che legge e stampa e salva in database la posizione attuale dell'utente. L'utente può servirsi di una web application dedicata dopo essersi registrato per visualizzare le posizioni salvate in database e la propria posizione in tempo reale senza dover consultare il Serial Monitor dell'Arduino IDE.

## Casi d'Uso Utente



## System Feature Utente

### Sign In

<b>Caso d'Uso</b>	<b>Data</b>
ID	Sign In
DESCRIZIONE	Registrazione utente
ATTORI PRIMARI	Utente
PRECONDIZIONI	GUI dedicata, verifica input
EVENTI	<ol style="list-style-type: none"><li>1. L'Utente clicca su Sign In</li><li>2. L'Utente inserisce dati</li><li>3. Il Sistema verifica i dati</li><li>4. SE 3) riesce allora inserisce dati nel database</li><li>5. L'Utente è reindirizzato alla home</li></ol>
EVENTI ALTERNATIVI	<ol style="list-style-type: none"><li>4.1 L'Utente è invitato a riscrivere i dati</li></ol>

## Login

<b>Caso d'Uso</b>	<b>Data</b>
ID	Login
DESCRIZIONE	Accesso alla web application
ATTORI PRIMARI	Utente
PRECONDIZIONI	GUI dedicata, verifica input
EVENTI	<ol style="list-style-type: none"><li>1. L'Utente clicca su Login</li><li>2. L'Utente inserisce dati</li><li>3. Il Sistema verifica i dati e li confronta col database</li><li>4. SE 3) riesce allora inserisce dati nel database</li><li>5. L'Utente è reindirizzato alla GPS home</li></ol>
EVENTI ALTERNATIVI	<ol style="list-style-type: none"><li>4.1 L'Utente è invitato a riscrivere i dati</li><li>4.2 Se L'Utente è già loggato ricarica la sessione</li></ol>

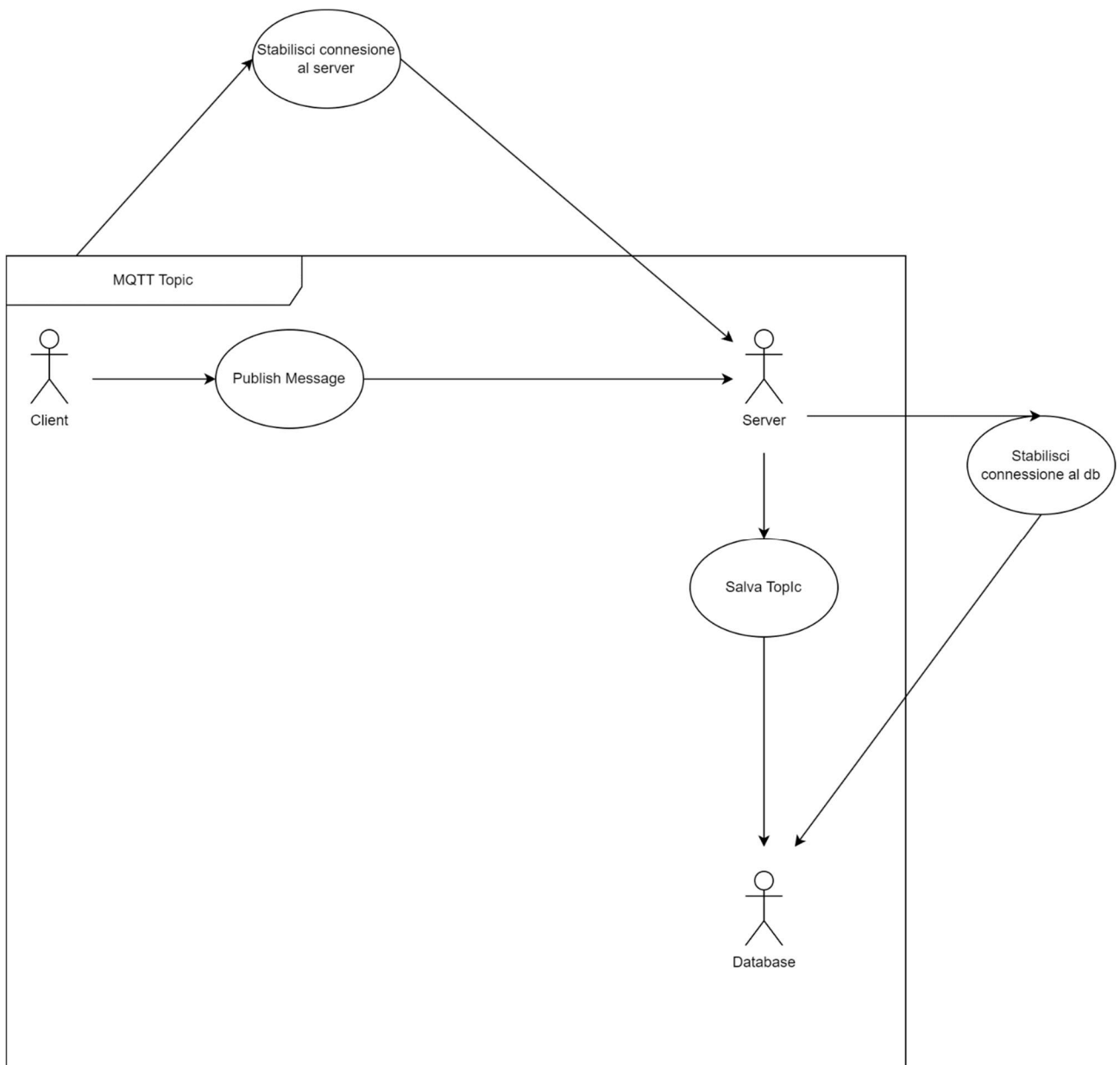
## Visualizza Posizioni

<b>Caso d'Uso</b>	<b>Data</b>
ID	Visualizza Posizioni
DESCRIZIONE	Visualizza Posizioni salvate in database
ATTORI PRIMARI	Utente
PRECONDIZIONI	GUI dedicata, connessione al database
EVENTI	<ol style="list-style-type: none"><li>1. L'Utente clicca su Visualizza Posizioni</li><li>2. Il Sistema stampa tabella dal database</li></ol>
EVENTI ALTERNATIVI	<ol style="list-style-type: none"><li>2.1 Se non è connesso al Database riporta errore</li></ol>

## Visualizza Posizione Attuale

<b>Caso d'Uso</b>	<b>Data</b>
ID	Visualizza Posizione Attuale
DESCRIZIONE	Visualizza Posizione Attuale tramite mappa Node-RED
ATTORI PRIMARI	Utente
ATTORI SECONDARI	-
PRECONDIZIONI	GUI dedicata, connessione al server
EVENTI	<ol style="list-style-type: none"><li>1. L'Utente clicca su Visualizza Posizione Attuale</li><li>2. Il Sistema mostra la mappa</li></ol>
EVENTI ALTERNATIVI	<ol style="list-style-type: none"><li>2.1 Se manca connessione al Server riporta errore</li></ol>

## Casi d'Uso Client-Server



## System Feature Client-Server

### Public Message

<b>Caso d'Uso</b>	<b>Data</b>
ID	Public Message
DESCRIZIONE	Public Message al Server
ATTORI PRIMARI	Client
ATTORI SECONDARI	Server

PRECONDIZIONI	Connessione al server
EVENTI	<ol style="list-style-type: none"> <li>1. Il Client riceve segnale GPS dal Sensore</li> <li>2. Il Client tenta il Publishing del Topic al Server</li> </ol>
EVENTI ALTERNATIVI	<ol style="list-style-type: none"> <li>2.1 Se manca connessione al Server riporta errore</li> </ol>

### Stabilisci Connessione al Server

<b>Caso d'Uso</b>	<b>Data</b>
ID	Stabilisci Connessione al Server
DESCRIZIONE	Stabilisci Connessione al Server
ATTORI PRIMARI	Client
ATTORI SECONDARI	Server
PRECONDIZIONI	-
EVENTI	<ol style="list-style-type: none"> <li>1. Il Client tenta la connessione al Server</li> <li>2. Il Client si connette al Server</li> </ol>
EVENTI ALTERNATIVI	<ol style="list-style-type: none"> <li>2.1 Se manca connessione al Server riporta errore</li> </ol>

### Salva Topic

<b>Caso d'Uso</b>	<b>Data</b>
ID	Salva Topic
DESCRIZIONE	Ricevi e salva Topic
ATTORI PRIMARI	Server
ATTORI SECONDARI	Database
PRECONDIZIONI	Connessione al Database
EVENTI	<ol style="list-style-type: none"> <li>1. Il Server legge il Topic dal Client</li> <li>2. Il Server tenta la query al Database</li> <li>3. Se 2) riesce allora salva in Database</li> </ol>

EVENTI ALTERNATIVI	3.1 Se manca connessione al Database riporta errore 3.2 Se la query è illecita riporta errore
--------------------	--

### Stabilisci Connessione al Database

<b>Caso d'Uso</b>	<b>Data</b>
ID	Stabilisci Connessione al Database
DESCRIZIONE	Stabilisci Connessione al Database
ATTORI PRIMARI	Server
ATTORI SECONDARI	Database
PRECONDIZIONI	-
EVENTI	1. Il Server tenta connessione al Database 2. Se 1) riesce allora si connette
EVENTI ALTERNATIVI	2.1 Se manca connessione al Database riporta errore

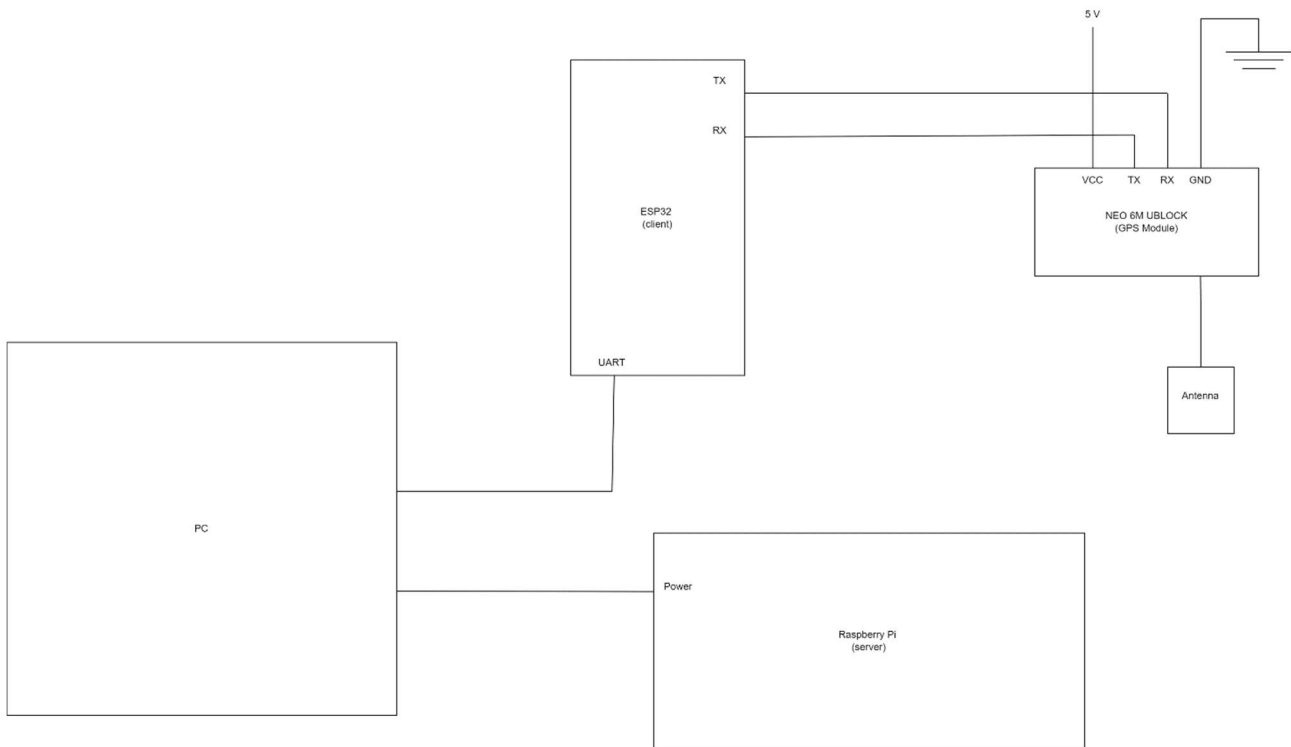
### Requisiti Hardware

Per la realizzazione circuitale del GPS tracker si è fatto uso dei seguenti componenti:

- Microcontrollore ESP32 Wrover B (client)
- Modulo GPS Neo 6m UBLOX (sensor)
- SBC Raspberry Pi 3B+ con Display IPS (broker)
- 4 jumper M-M



## Schema Circuitale



## Requisiti Software

Per l'utilizzo dell'app basta una stabile connessione ad Internet e la consultazione del file README.txt allegato alla Relazione Progettuale.

## Protocolli di Rete

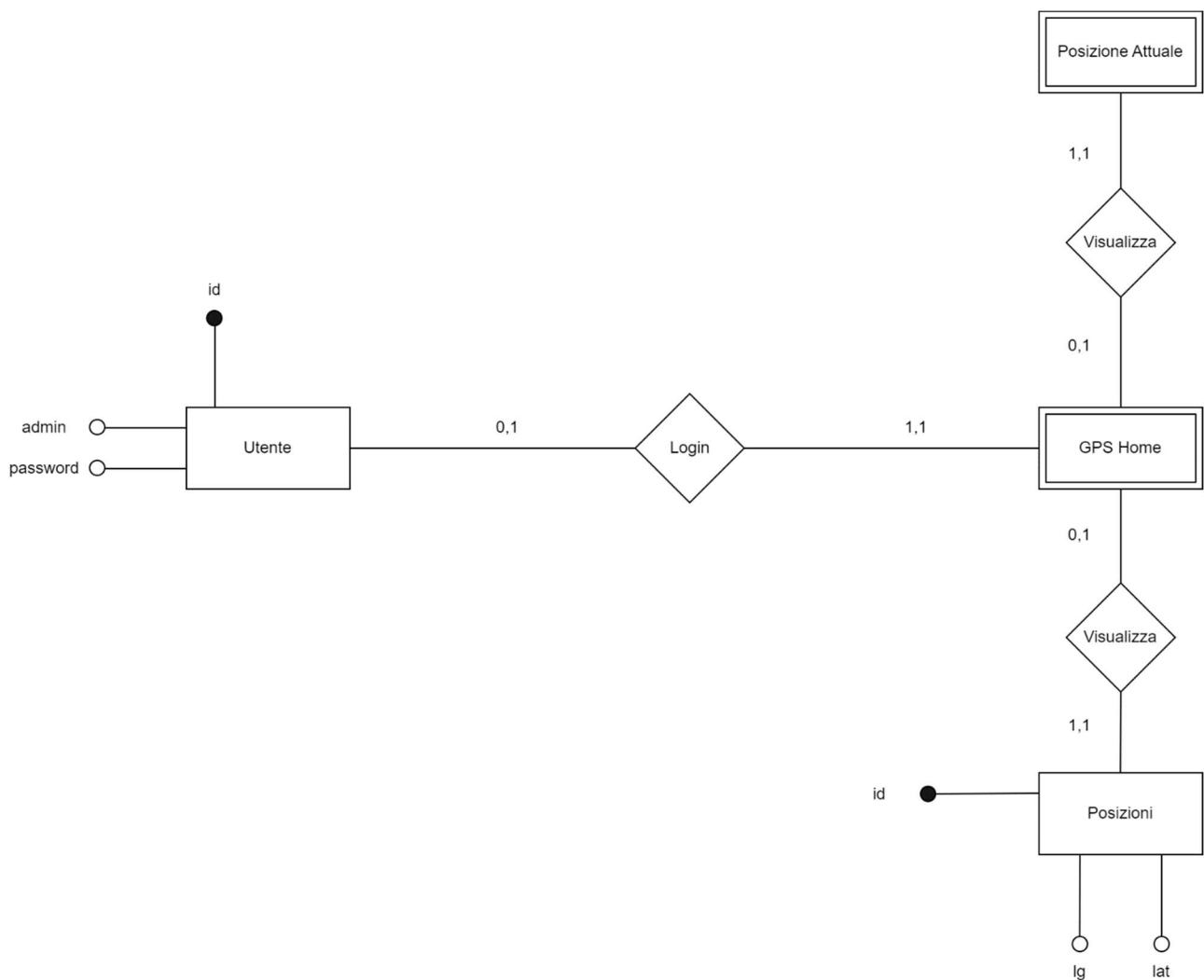
Il progetto sfrutta il protocollo ISO standard MQTT per la comunicazione tra la Raspberry Pi e la ESP32. MQTT è un protocollo di messaggistica di tipo publish-subscribe appartenente alla categoria dei TCP/IP. Ottimo per progetti a banda limitata e a basso consumo energetico è stato scelto proprio per la sua semplicità d'uso e per simulare tecnicamente quello che avviene in sistemi più complessi. Infatti, in questo progetto è stato utilizzato un solo GPS tracker per ovvie ragioni tecniche, ma potenzialmente si potrebbero sviluppare una moltitudine di dispositivi che fanno da client in grado di lavorare contemporaneamente grazie all'opera di "smistamento" dei messaggi ricevuti dal broker. Inoltre, si serve del protocollo HTTP per la connessione ad Internet tipica delle applicazioni client-server.

## Database

Il progetto sfrutta come gestore del database il software XAMPP che permette l'installazione di un server Apache HTTP, di un database server MariaDB e del linguaggio di programmazione PHP. Di qui si tratterà nello specifico della programmazione del suddetto database.

## Progettazione concettuale

### Diagramma ER



## Progettazione logica

user(**ID**, admin, password).

position(**ID**, lat, lg).

## **Possibili sviluppi futuri**

Il nostro GPS-TRACKER è stato studiato per utilizzarlo insieme alla consultazione dell'apposita web application o tramite il monitor del Raspberry Pi, in alternativa può essere implementato al circuito un qualsiasi monitor OLED per la visualizzazione della mappa e/o dei dati satellitari. Inoltre, è studiato e collaudato per funzionare collegato ad un PC, è quindi consigliato sviluppare in futuro una connessione wireless alla UART del ESP32 per permetterne la mobilità e l'indipendenza dal PC. Si può pensare anche di inscatolare il GPS per renderne la fruibilità migliore.

## Test dimostrativi

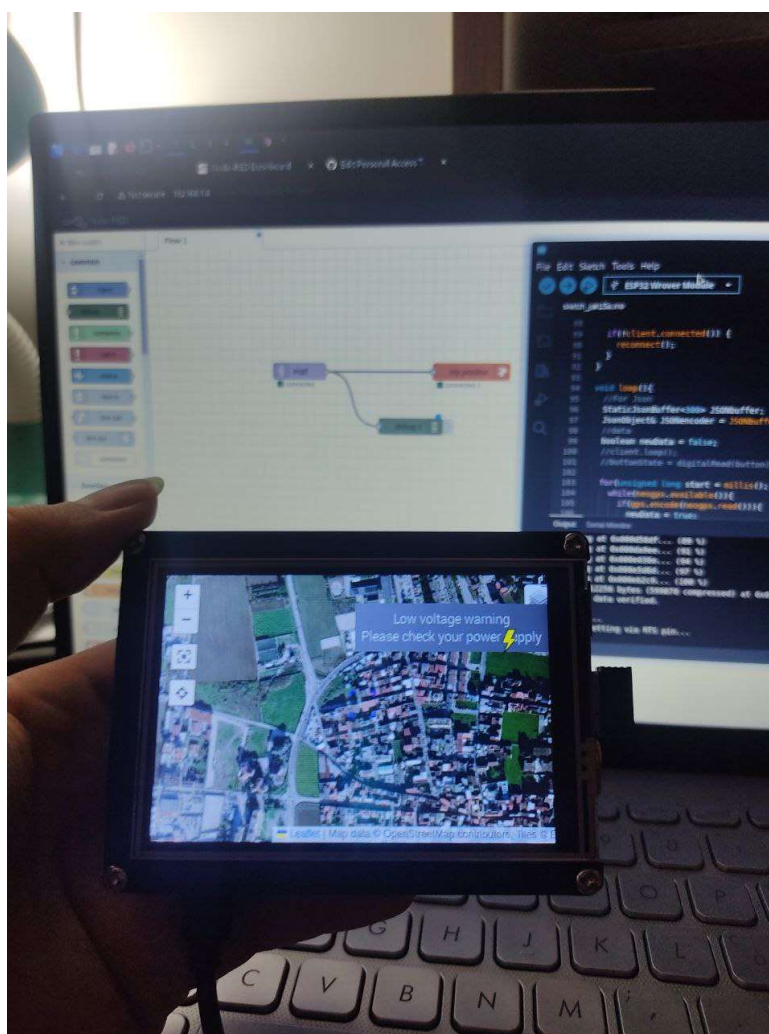


Figura 1 L'utente visualizza la sua posizione momentanea

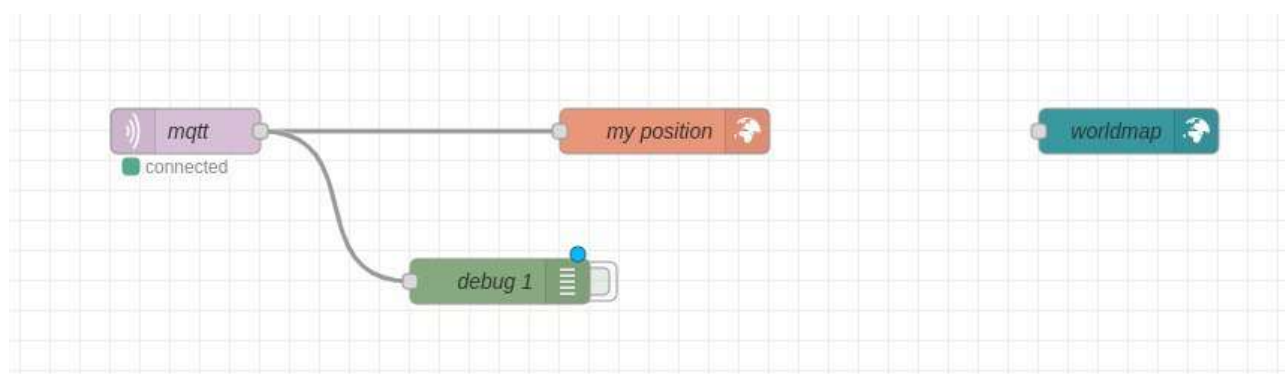


Figura 2 Flow per la visualizzazione della mappa

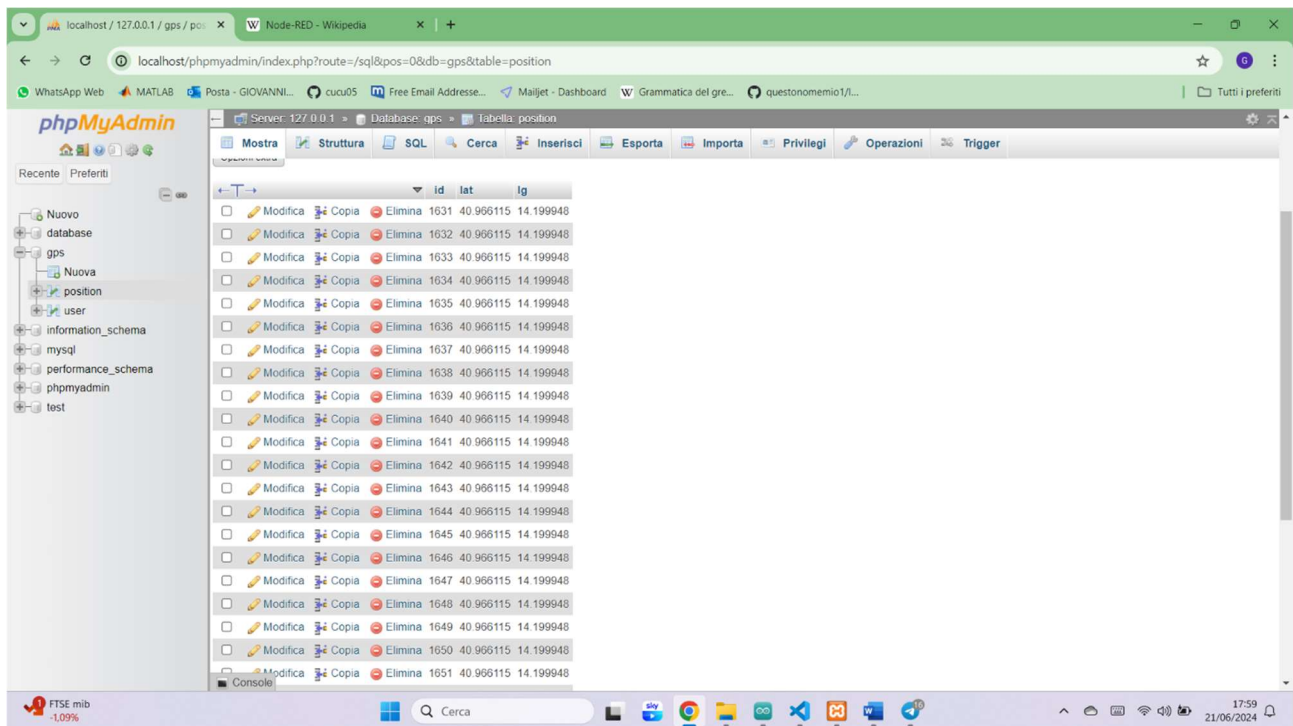


Figura 3 Database delle posizioni