

# **Advanced Programming Languages - Appello 18 Novembre 2021**

## **Meteo**

**Remo Valerio Labbia - matricola 1000012407**

### **Sommario**

1 - Introduzione	1
2 - Scenario	2
3 - Architettura	3
5 - Interfaccia utente (Client)	6

### **1 - Introduzione**

In questo documento è descritto il progetto svolto per l'esame di Advanced Programming Languages. L'appena citato progetto, svolto da Remo Valerio Labbia, riguarda un'applicazione per il meteo.

## 2 - Scenario

Un client utente, con apposita GUI, può richiedere al server varie informazioni meteorologiche di una città (location) con relativo codice nazione (country).

Per fare ciò viene utilizzato il server usa le API di OpenWeather, il quale quando necessario memorizza delle informazioni nel database SQLite a cui è collegato.

L'utente può anche richiedere informazioni statistiche sul meteo passato, sfruttando uno script.

### 3 - Architettura

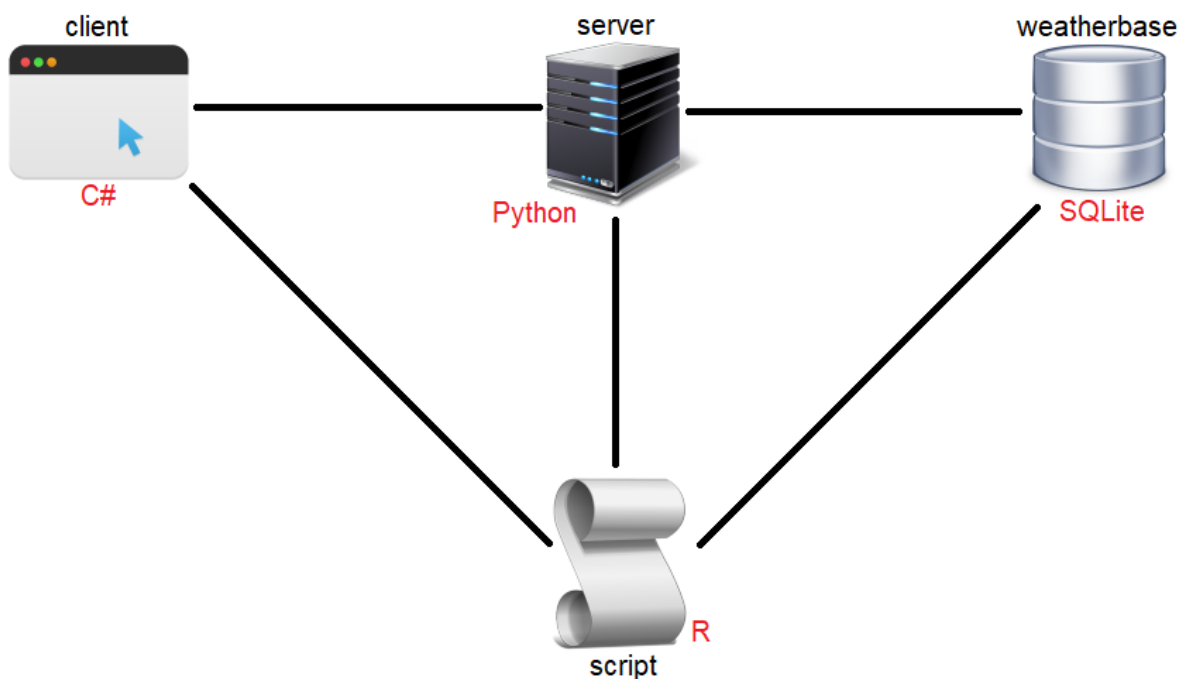
Il client è stato realizzato in C#, con Windows Forms per la GUI. Comunica tramite REST col server e utilizza lo script R.

Il server in Python, sfruttando il framework Flask per creare le varie route richiamabili dal client. Nel server avviene tutta l'elaborazione riguardo i dati richiesti dal client, facendo uso della chiave API fornita da OpenWeather per recuperare informazioni climatiche di interesse. Comunica col client e col database.

È presente un database SQLite3 in cui vengono memorizzati dati acceduti poi dal server o dallo script in R.

Lo script in R, usato dal client, permette di graficare sulla base di alcuni dati che richiede direttamente al server o al database.

Di seguito è riportata un'immagine dell'architettura appena descritta:



## 4 - File e funzioni (script R e soprattutto del Server)

Sono riportate tutte le cartelle e file principali:

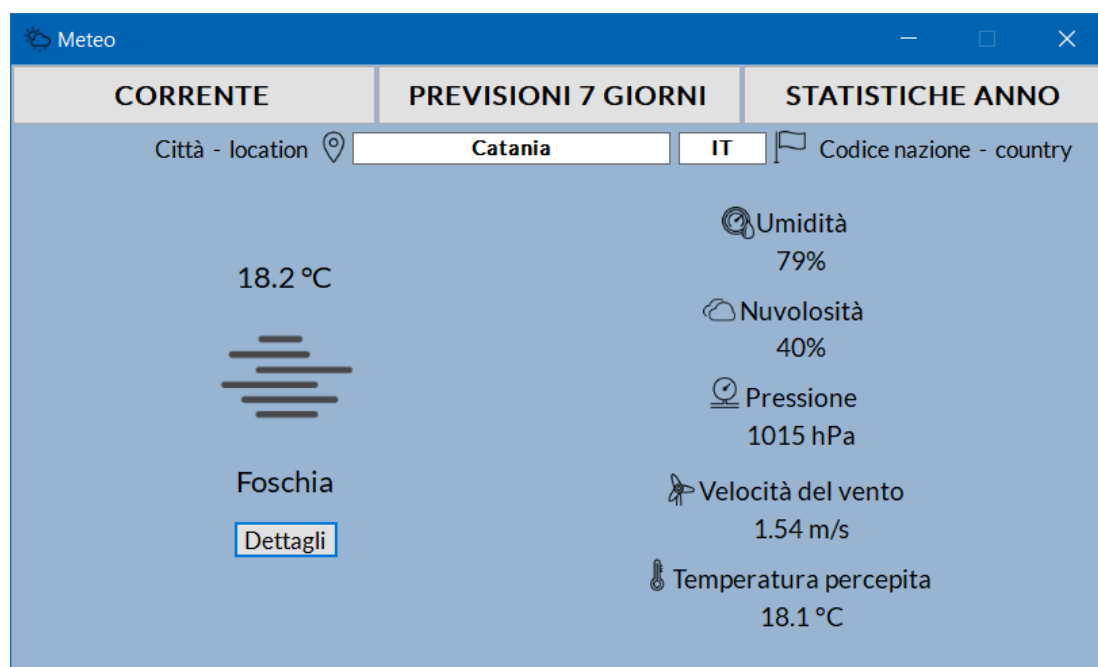
- Client\bin\Release\**Client.exe** = in questo path è contenuto il file eseguibile dal client utente per richiedere informazioni sul meteo.
- Server\.env = file che contiene informazioni di vitale importanza per l'intera architettura; quali indirizzo e porta su cui il server espone le route Flask, nome database e chiave API per OpenWeather.
- Server\server.py = per far partire il server. Qui viene configurato il tutto, impostando le varie route, con relative view, richiamate poi dal client.
- Server\weather.py = vi sono le funzioni che utilizza il server per fornire dati al client, ovvero: meteo corrente (con temperatura in °C, descrizione condizione climatica e relativa icona) e più dettagli (temperatura percepita, pressione in hPa, percentuale umidità, percentuale nuvolosità, velocità del vento in m/s) su quest'ultimo, previsioni meteo per 7 giorni, (relative a temperatura media, descrizione condizione climatica e relativa icona. Non compreso il corrente), meteo passato 1 anno (per ogni mese sono forniti temperatura media, umidità media, precipitazione media in mm).
- Server\my\_db.py = relativo a gestione e comunicazione col db. Il server salva nel db solo le informazioni riguardo le previsioni meteo per 7 giorni, preoccupandosi di salvare in quest'ultimo l'icona del meteo in stringa base64.

- Server\weather\_helper.py = modulo avente funzioni di supporto utilizzate dal server per elaborare i dati; esse comprendono check dei parametri ricevuti dal client, richiesta icona da OpenWeather e conversione in base64, arrotondamento e rimozione “.0” da temperatura.
- Server\custom\_error.py = file unico che ha le varie classi per le eccezioni custom definite.
- statistics.R = file script R che si occupa di fornire grafici al client. Il client specifica quale plot vuole elaborato da questo script; ossia grafico della temperatura prevista per i prossimi 7 giorni o dei precedenti 12 mesi, della percentuale umidità/precipitazione dei precedenti 12 mesi.
- Server\weatherbase = file database SQLite, creato la prima volta che il server viene avviato. Contiene una sola tabella per le previsioni meteo dei prossimi 7 giorni (non compreso il corrente), di una specifica città e codice nazione. Contiene come chiave primaria la coppia location/country e, appunto per ogni riga, ha le colonne riguardanti (per ognuno dei prossimi 7 giorni) la temperatura media, la descrizione della condizione climatica e l'icona codificata in base64.

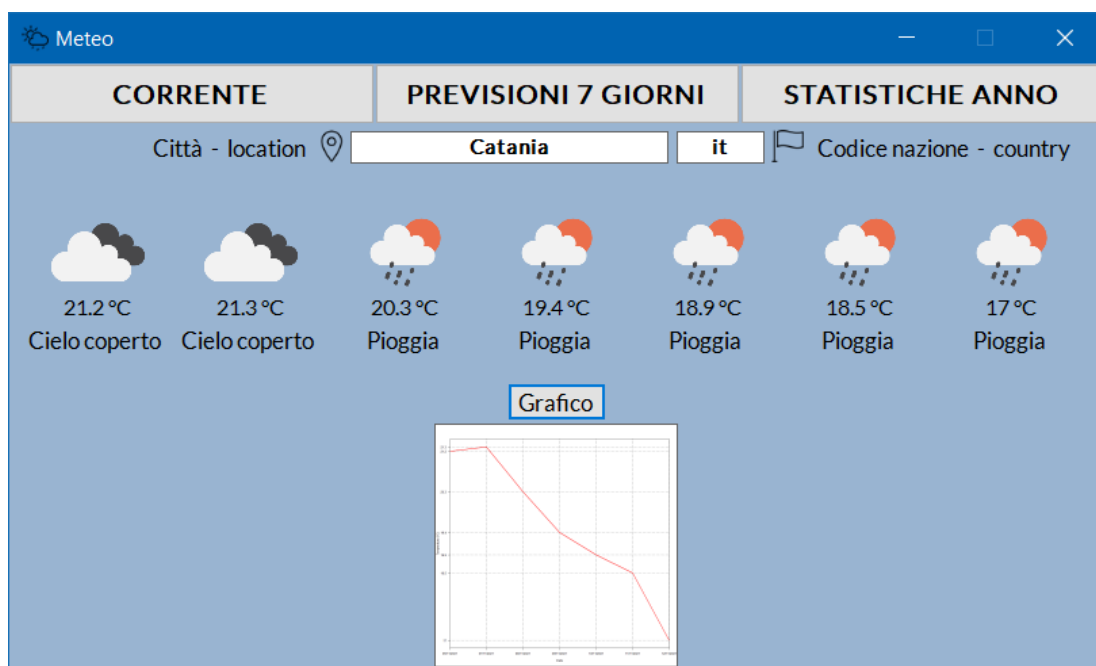
## 5 - Interfaccia utente (Client)

Il form principale, WeatherForm, è composto da 3 pagine, costruite sulla base di 3 UserControl, accedute rispettivamente dai 3 appositi bottoni che alla pressione si collegano al server. Oltre alle funzioni associate alla pressione dei bottoni, sono presenti funzioni richiamate anche dalle pagine per estrarre l'immagine dalla stringa base64 ricevuta dal server (con overload del metodo), eseguire lo script R e per il recupero del relativo plot.

Inizialmente, nella prima pagina, è possibile avere inizialmente solo informazioni sul meteo corrente e successivamente, se si preme il bottone “Dettagli”, anche i dettagli meteorologici di seguito riportati. I risultati testuali ottenuti dal server vengono impostati su appositi label.



Nella seconda pagina sono presenti le informazioni meteorologiche citate sui prossimi 7 giorni. Premendo il bottone è possibile avere il relativo grafico, elaborato dallo script R, di queste previsioni.



Nella terza pagina si possono avere, in seguito alla pressione dell'apposito bottone, i grafici di temperatura, umidità e precipitazione dei precedenti 12 mesi. Ognuna delle funzioni presenti in questa pagina passerà allo script R una stringa indicativa del grafico desiderato, ed eliminerà il grafico dal sistema per non lasciare tracce. Tutto ciò avviene anche nella seconda pagina, come anche il fatto che alla pressione in uno dei grafici verrà aperta una finestra relativa al form "RealZoom" per vedere il plot più grande.

