

# Wave Map

## Cellular Connectivity and Noise Map

Tian Cheng Xia  
Matricola: 975129

Corso di Laboratorio di applicazioni mobili  
Alma Mater Studiorum – Università di Bologna

A.A. 2022 – 2023

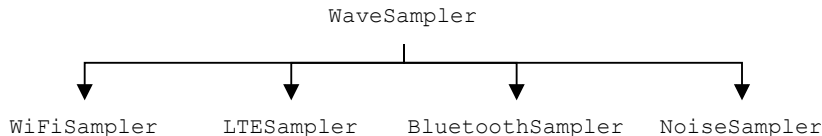
# Informazioni introduttive

- ▶ Android nativo con Kotlin.
- ▶ Room per interfacciarsi con il database locale.
- ▶ Coroutine per le operazioni asincrone.
- ▶ Conversione dei callback in funzioni `suspend` attraverso `suspendCoroutine`.
- ▶ Gestione di lingua e temi.

# Raccolta dati

La classe astratta WaveSampler rappresenta un misuratore e richiede l'implementazione dei metodi:

- ▶ `sample`: prende una nuova misurazione.
- ▶ `store`: salva una misurazione.
- ▶ `retrieve`: ricerca misurazioni.



# Raccolta dati

## WiFiSampler

- ▶ Ottiene la potenza della rete attualmente connessa attraverso il `ConnectivityManager` (o `WifiManager`).
- ▶ Misura le reti circostanti con un `BroadcastReceiver` con filtro `WifiManager.SCAN_RESULTS_AVAILABLE_ACTION` e il `WifiManager`.

## LTESampler

- ▶ Ottiene la potenza attraverso il `TelephonyManager`.

# Raccolta dati

## BluetoothSampler

- ▶ Misura la potenza dei dispositivi accoppiati attraverso il `BluetoothManager`.
- ▶ Misura i dispositivi circostanti con un `BroadcastReceiver` con filtro `BluetoothDevice.ACTION_FOUND` e il `BluetoothManager`.

## NoiseSampler

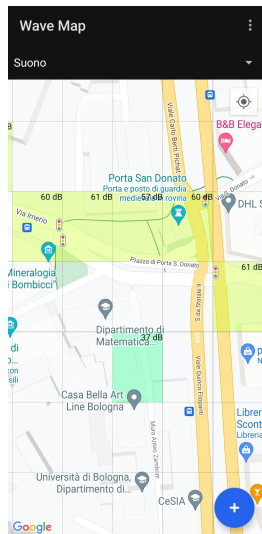
- ▶ Calcola la media di una serie di misurazioni effettuate con un `MediaRecorder`.

# Mappa

**Cella** descritta dalle coordinate nord-ovest e dalla dimensione in metri.

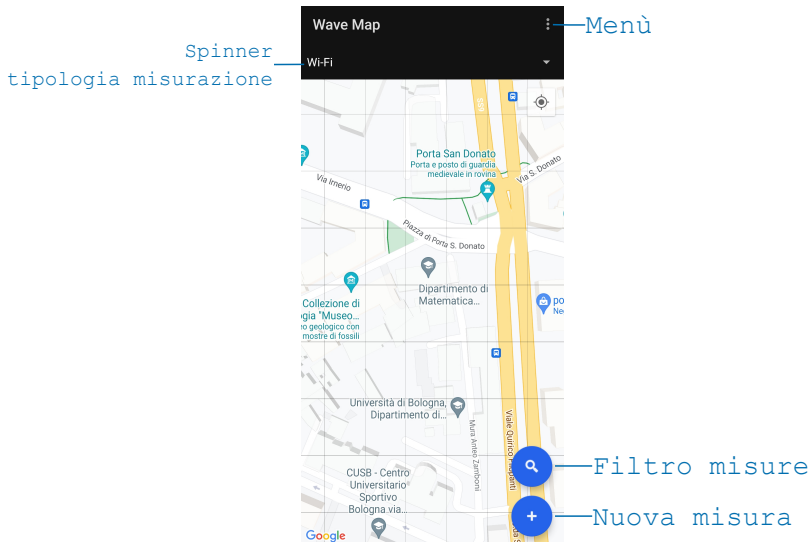
**Colore** determinato in formato HSL mappando la misurazione nell'intervallo  $[0, 150]$  (hue).

**Griglia** generata rispetto ad una celle di riferimento eletta in fase di avvio.



# App principale

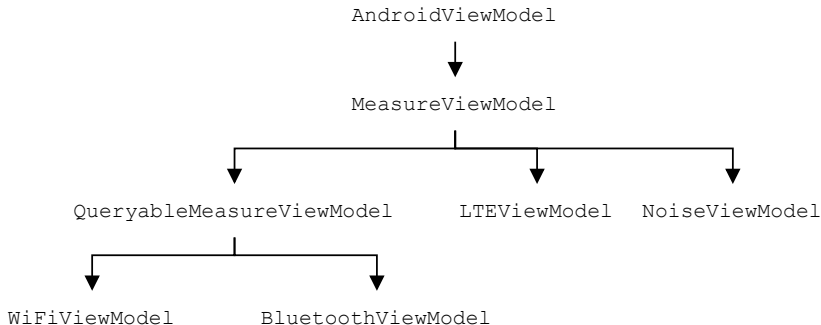
## Overview



# App principale

## ViewModel

Ciascun misuratore ha un proprio ViewModel dedicato.



**MeasureViewModel** descrive le proprietà base di un misuratore.

**QueryableMeasureViewModel** rappresenta un misuratore a cui è applicabile un filtro.



# App principale

## ViewModel

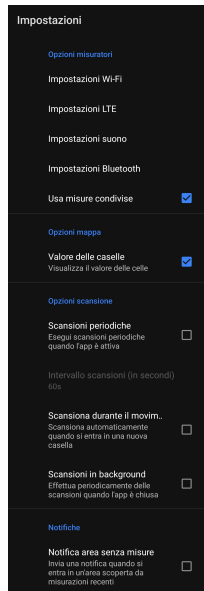
Il ViewModel principale è `MainViewModel`.

- ▶ Contiene un'istanza di ciascun ViewModel dei misuratori.
- ▶ Gestisce il misuratore attualmente attivo.
- ▶ Garantisce mutua esclusione per ciascun misuratore.
- ▶ LiveData per notificare la view dei cambiamenti.

# Impostazioni

Creata utilizzando la libreria Preference.

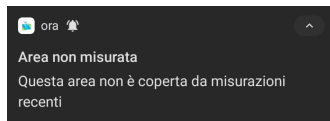
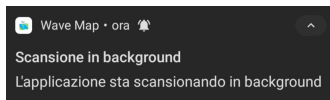
Ciascun misuratore ha una pagina dedicata generata dinamicamente.



# Servizio in background

- ▶ Scansioni durante il movimento.
- ▶ Notifica di aree prive di misurazioni recenti.

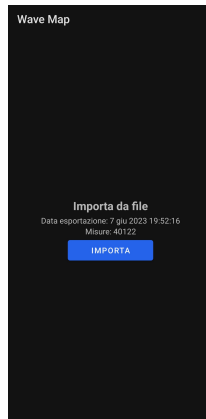
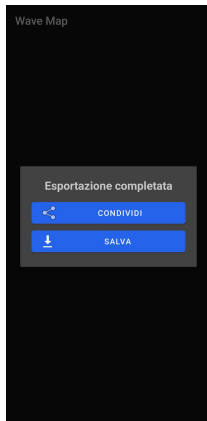
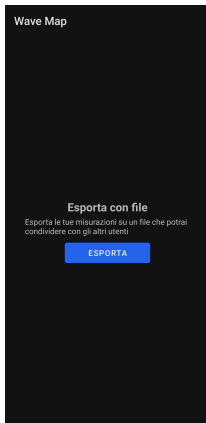
Il servizio viene avviato come *Foreground Service* se è necessario effettuare scansioni.



# Condivisione dati

Esportazione su file in formato JSON.

Importazione unificando le misurazioni temporalmente vicine.



# Problemi noti

- ▶ Dall'API 26 sono state introdotte restrizioni al numero di scansioni Wi-Fi che un'applicazione può fare.
- ▶ Alcuni produttori hanno politiche di ottimizzazione della batteria che tendono a limitare i servizi in background.
- ▶ L'`intent-filter` per l'importazione ammette input che non necessariamente riguardano l'app.