

# Cellular Connectivity and Noise Map

Relazione

**Xia · Tian Cheng**

Matricola: 0000975129

Email: [tiancheng.xia@studio.unibo.it](mailto:tiancheng.xia@studio.unibo.it)

Anno accademico

2022 — 2023

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

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Scelte progettuali</b>	<b>1</b>
2.1	Informazioni generali . . . . .	1
2.2	Mappa . . . . .	1
2.3	Raccolta dei dati . . . . .	1
2.3.1	Struttura e memorizzazione delle misurazioni . . . . .	1
2.3.2	Sampler . . . . .	1

# 1 Introduzione

## 2 Scelte progettuali

### 2.1 Informazioni generali

### 2.2 Mappa

### 2.3 Raccolta dei dati

#### 2.3.1 Struttura e memorizzazione delle misurazioni

Una misurazione è descritta dall'interfaccia `WaveMeasure` e contiene il valore della misurazione, un timestamp e la posizione. In aggiunta, è presente un campo per informazioni aggiuntive utile per distinguere alcune tipologie di misurazioni (es. per Wi-Fi e Bluetooth viene salvato il BSSID).

L'interfaccia `WaveMeasure` viene quindi utilizzata per implementare la classe `MeasureTable` che descrive la tabella del database dedicata per memorizzare le misurazioni. Tutte le misurazioni sono salvate nella stessa tabella e sono differenziate da un campo `type`.

#### 2.3.2 Sampler

Per la raccolta dei dati è stato introdotto il concetto di *sampler* per gestire in maniera modulare le misurazioni. Nello specifico, un *sampler* è descritto dalla classe astratta `WaveSampler` e richiede l'implementazione dei seguenti metodi:

- `sample` per prendere una nuova misurazione
- `store` per il salvataggio dei dati nel database
- `retrieve` per la ricerca dei dati note le coordinate dei vertici di una cella della mappa

Inoltre, sono esposte le seguenti funzioni ausiliarie:

- `average` richiama `retrieve` e restituisce la media dei valori
- `sampleAndStore` richiama in sequenza `sample` e `store`

Per maggiore flessibilità, le misure vengono sempre intese come liste di `WaveMeasure`. Ciò permette di gestire misurazioni che per loro natura non generano un'unica misurazione (es. Wi-Fi e Bluetooth).

A partire da `WaveSampler` sono quindi implementati i *sampler* per Wi-Fi, Bluetooth, LTE e suono.