

## **FITA 2**

### **MÈTODE D'ENTRADA BÀSIC**

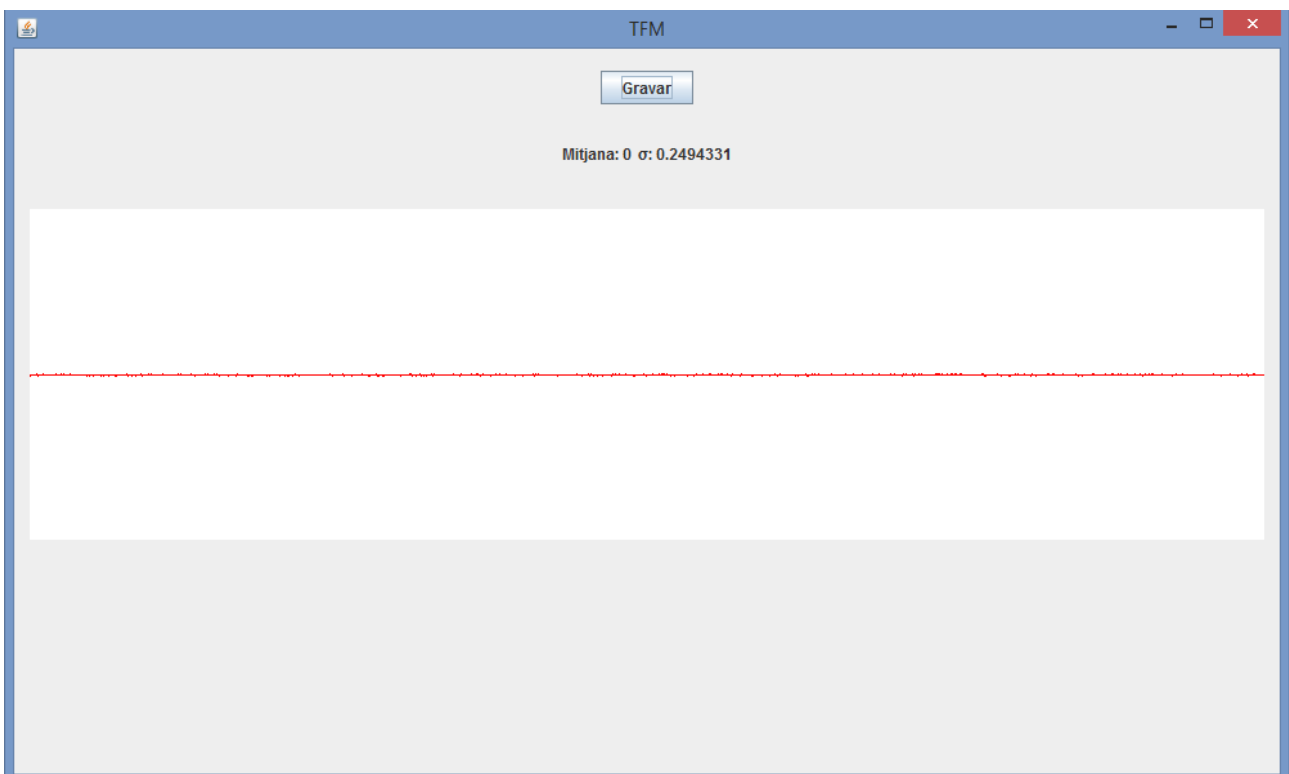
**MÀSTER INTERUNIVERSITARI EN SEGURETAT DE  
LES TECNOLOGIES DE LA INFORMACIÓ I COMUNICACIÓ**

Cristian Requena Barreda

## **DESENVOLUPAMENT**

Seguint amb la previsió, aquesta fita inclou l'habilitació d'un mètode d'entrada per l'usuari, que li permet interactuar amb el programari per indicar quan ha de començar i acabar el tractament.

L'entrada d'àudio es realitza mitjançant 44100 mostres per segon amb una resolució de 8 bits. Tot i això, el tractament de les dades es duu a terme en subconjunts de 4410 mostres, ja que això permet una visualització de dades quasi en temps real. Aquest temps de resposta àgil ha permès poder comprovar com era el soroll de fons, ja que revisant el gràfic de l'espectre de so es veuen petites variacions:



**Captura 1: Soroll de fons.**

D'altra banda, s'han realitzat dos càlculs senzills que es computen cada vegada que s'actualitza l'espectre de so, això és, 10 vegades per segon, i empen les 4410 mostres comentades anteriorment. Aquests són la mitjana del valor de les mostres i la desviació estàndard de les mateixes.

El codi desenvolupat ha estat principalment al bucle principal de l'aplicatiu, on s'ha realitzat, tal com es mostra a continuació, el filtratge del soroll de fons, els càlculs bàsics i la persistència de dades:

```
while (!stopped) {
    numBytesRead = line.read(data, 0, data.length);
    if (isRecording) {
        // Anul·lar el soroll de fons.
        for (int i=0; i<numBytesRead; i++) {
            if (Math.abs(data[i]) < 5) {
                data[i] = 0;
            }
        }

        // Actualitzar controls.
        // Calcular la mitjana.
        Long calcul = 0L;
        int mitjana = 0;
        for (int i=0; i<numBytesRead; i++) {
            calcul = calcul + data[i];
        }
        mitjana = (int) (calcul / numBytesRead);
        lblMitjana.setText("Mitjana: " + mitjana);

        // Calcular la desviació estàndard.
        double calculDouble = 0F;
        Float desviacio = 0F;
        for (int i=0; i<numBytesRead; i++) {
            calculDouble = calculDouble + Math.pow(data[i] - mitjana, 2);
        }
        desviacio = (float) (calculDouble / numBytesRead);
        lblDesviacio.setText("σ: " + desviacio);

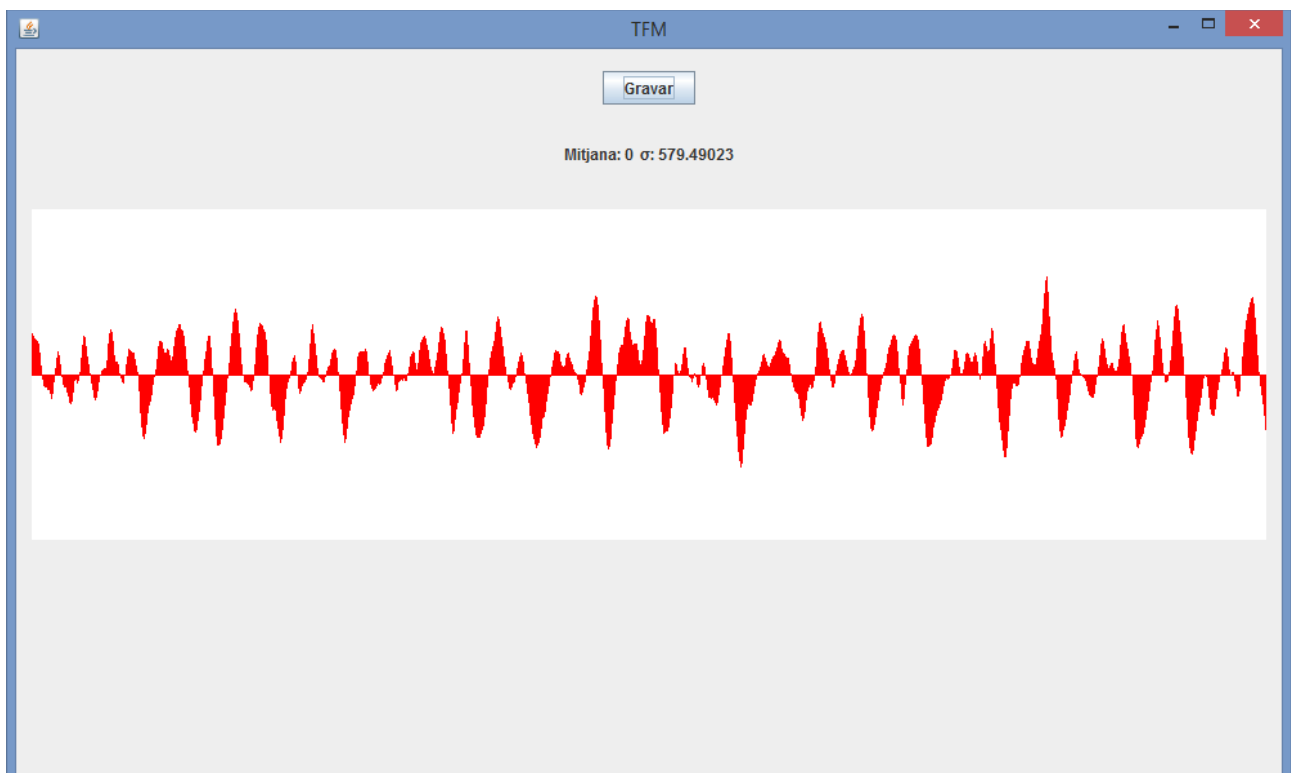
        // Guardar fitxer.
        Files.write(Paths.get("recording.kav"), data, StandardOpenOption.APPEND);

        // Repintar espectre.
        myGrafic.repaint();
    }
}
```

La gravació es realitza mitjançant un fitxer binari anomenat *recording.kav*, on cada byte correspon a una mostra de so, aprofitant que aquestes tenen una resolució de 8 bits.

## CONCLUSIONS

Durant el desenvolupament d'aquesta fita s'ha pogut constatar que, degut a l'espectre de so té una forma sinusoïdal, no té sentit calcular la mitjana aritmètica de les mostres, ja que sempre tendirà a ser 0. D'altra banda, el valor de la desviació estàndard sí que és rellevant, perquè podrà ser emprat, per exemple, per detectar quan es comença a enregistrar un so, quan s'acaba o estimar dinàmicament quin és el soroll de fons en un moment determinat.



Captura 2: Espectre de so d'una cançó.