

**Pau Moriones Galve**

**SENSOR DETECTOR DE NO<sub>2</sub> AMB LECTURA DE DADES MITJANÇANT  
TECNOLOGIA NFC**

**TREBALL DE FI DE GRAU**

**dirigit pel Dr. Eduard Llobet Valero**

**Grau d'Enginyeria de Sistemes i Serveis de Telecomunicacions**



**UNIVERSITAT ROVIRA I VIRGILI**

**Tarragona**

**2023**

## **AGRAÏMENTS**

Primerament, donar les gràcies al tutor d'aquest Treball de Fi de Grau, l'Eduard Llobet Valero, per l'assessorament i guiatge rebut durant el desenvolupament del projecte.

En segon lloc, agrair a l'Èric Navarrete Gatell, integrant del grup d'investigació MINOS, per l'ajuda proporcionada en la fabricació dels sensors.

## RESUM DEL PROJECTE

**(Català)** Aquest Treball de Fi de Grau es basa en el desenvolupament d'un sensor químic resistiu per a la captació de la concentració de diòxid de nitrogen ( $\text{NO}_2$ ) en ppm que hi pot haver en un ambient interior. Per a la determinació de la concentració d'aquest gas, es realitzaran diverses proves de funcionament per a determinar la relació entre la variació de la resistència òhmica del sensor i la acumulació d'aquest gas en un ambient interior. Posteriorment es durà a terme la implementació de l'electrònica de control que permetrà la lectura de les dades captades pel sensor per al seu tractament, mitjançant *Raspberry Pi Pico* i fent ús de la programació en llenguatge *MicroPython*. Finalment, es desenvoluparan les comunicacions *Near Field Communication (NFC)* que permetran la lectura de les dades captades pel sensor a través d'una aplicació mòbil desenvolupada utilitzant el llenguatge de programació *Kotlin* i l'entorn de desenvolupament integrat *Android Studio*.

**(Español)** Este Trabajo de Fin de Grado se basa en el desarrollo de un sensor químico resistivo para la captación de concentración de dióxido de nitrógeno ( $\text{NO}_2$ ) en ppm que puede haber en ambientes interiores. Para determinar la concentración de dicho gas, se realizarán distintos ensayos para determinar la relación existente entre la variación de la resistencia óhmica del sensor y la cantidad de dicho gas en ambientes interiores. Seguidamente, se realizará la implementación de la electrónica necesaria para posibilitar el control y lectura de los datos captados por el sensor, mediante *Raspberry Pi Pico*, cuya programación se realizará mediante el lenguaje *MicroPython*. Para finalizar, se llevará a cabo el desarrollo de las comunicaciones *Near Field Communication (NFC)* que favorecerán la lectura de los datos recogidos por el sensor por una aplicación móvil desarrollada haciendo uso del lenguaje de programación *Kotlin* y el entorno de desarrollo integrado *Android Studio*.

**(English)** This Final Degree Project is based on the development of a chemo resistive sensor in order to capture the level of nitrogen dioxide ( $\text{NO}_2$ ) sensed on an indoor scenario. To determine the concentration of this gas, different tests will be carried out to establish the existing relationship between the variation of the ohmic resistance of the sensor and the amount of nitrogen dioxide in indoor environments. Next, the implementation of the electronics needed to enable the control and reading of the data collected by the sensor, using *Raspberry Pi Pico* as microcontroller, whose programming will be done using the *MicroPython* language. Finally, the development of *Near Field Communication (NFC)* communication will allow the reading of the data collected from the sensor by a mobile application developed using *Kotlin* programming language and *Android Studio* as the integrated development environment.

# ÍNDIX

|                                                                                         |           |
|-----------------------------------------------------------------------------------------|-----------|
| <b>1. INTRODUCCIÓ</b> .....                                                             | <b>1</b>  |
| <b>1.1. Planificació del projecte.</b> .....                                            | <b>1</b>  |
| <b>1.2. Objectius del projecte.</b> .....                                               | <b>1</b>  |
| <b>1.3. Estructura de la memòria.</b> .....                                             | <b>1</b>  |
| <b>2. MARC TEÒRIC</b> .....                                                             | <b>3</b>  |
| <b>2.1. Diòxid de nitrogen (NO<sub>2</sub>).</b> .....                                  | <b>3</b>  |
| 2.1.1. Efectes adversos sobre la salut. ....                                            | 3         |
| 2.1.2. Nivells recomanats i legislats de NO <sub>2</sub> . ....                         | 4         |
| <b>2.2. Conceptes bàsics dels sensors químic resistius.</b> .....                       | <b>4</b>  |
| 2.2.1. Estructura del sensor químic resistiu.....                                       | 5         |
| 2.2.2. Paràmetres que en defineixen el comportament.....                                | 5         |
| <b>2.3. Propietats del grafè.</b> .....                                                 | <b>7</b>  |
| <b>3. DESENVOLUPAMENT DEL SENSOR</b> .....                                              | <b>9</b>  |
| <b>3.1. Materials utilitzats.</b> .....                                                 | <b>9</b>  |
| <b>3.2. Metodologia.</b> .....                                                          | <b>10</b> |
| <b>4. RESPOSTA I ANÀLISI DELS RESULTATS AMB MATLAB.</b> .....                           | <b>12</b> |
| <b>4.1. Presentació dels resultats obtinguts.</b> .....                                 | <b>12</b> |
| <b>4.2. Anàlisi dels resultats.</b> .....                                               | <b>14</b> |
| <b>5. TECNOLOGIA NFC</b> .....                                                          | <b>24</b> |
| <b>5.1. Història i definició de la tecnologia Near Field Communication (NFC).</b> ..... | <b>24</b> |
| <b>5.2. Diferències entre RFID i NFC.</b> .....                                         | <b>24</b> |
| <b>5.3. Descripció dels principis físics en que es basa la comunicació NFC.</b> .....   | <b>25</b> |
| 5.3.1. Acoblament inductiu. ....                                                        | 25        |
| 5.3.2. Coeficient d'acoblament. ....                                                    | 30        |
| 5.3.3. Freqüència de ressonància.....                                                   | 31        |
| <b>5.4. Protocols i estàndards.</b> .....                                               | <b>32</b> |

|                                                                             |           |
|-----------------------------------------------------------------------------|-----------|
| <b>5.5. Funcionament.....</b>                                               | <b>36</b> |
| 5.5.1. Mode actiu.....                                                      | 37        |
| 5.5.2. Mode passiu.....                                                     | 37        |
| 5.5.3. Modes de comunicació.....                                            | 38        |
| 5.5.4. Modulació i codificació del senyal.....                              | 38        |
| <b>5.6. NDEF.....</b>                                                       | <b>44</b> |
| <b>5.7. Seguretat.....</b>                                                  | <b>46</b> |
| <b>6. PROCESSAT DE DADES AMB RASPBERRY PI PICO.....</b>                     | <b>48</b> |
| <b>6.1. Introducció a <i>Raspberry Pi Pico</i>.....</b>                     | <b>48</b> |
| 6.1.1. Descripció del hardware.....                                         | 48        |
| 6.1.2. Programació.....                                                     | 50        |
| <b>6.2. Comparació entre <i>Raspberry Pi Pico</i> i <i>Arduino</i>.....</b> | <b>52</b> |
| <b>6.3. Mòdul MIKROE-1726.....</b>                                          | <b>53</b> |
| 6.3.1 Descripció del hardware del tag.....                                  | 53        |
| <b>6.4. Comunicació entre mòdul NFC i microcontrolador.....</b>             | <b>54</b> |
| 6.4.1. I2C.....                                                             | 54        |
| <b>6.5. Adquisició de dades.....</b>                                        | <b>57</b> |
| 6.5.1. Condicionament.....                                                  | 58        |
| 6.5.2. Adaptació.....                                                       | 59        |
| <b>6.6. Desenvolupament del codi.....</b>                                   | <b>62</b> |
| 6.6.1. Variables de la classe.....                                          | 63        |
| 6.6.2. Comunicació entre microcontrolador i tag NFC.....                    | 65        |
| 6.6.4. Programa principal.....                                              | 74        |
| <b>7. APLICACIÓ MÒBIL PER A GESTIONAR I VISUALITZAR LES DADES.....</b>      | <b>77</b> |
| <b>7.1. Introducció a <i>Android Studio</i>.....</b>                        | <b>77</b> |
| <b>7.2. Consideracions prèvies.....</b>                                     | <b>78</b> |
| <b>7.3. Desenvolupament de l'aplicació.....</b>                             | <b>79</b> |
| 7.3.1. Estructura de l'aplicació i funcionament.....                        | 79        |
| 7.3.2. Descripció de la implementació.....                                  | 84        |
| <b>8. PROVES I RESULTATS.....</b>                                           | <b>97</b> |

|                                                                   |            |
|-------------------------------------------------------------------|------------|
| <b>8.1. Preparació i muntatge.</b> .....                          | <b>97</b>  |
| <b>8.2. Resultats obtinguts.</b> .....                            | <b>98</b>  |
| <b>9. CONCLUSIONS I LÍNIES FUTURES DEL PROJECTE</b> .....         | <b>102</b> |
| <b>9.1. Conclusions.</b> .....                                    | <b>102</b> |
| <b>9.2. Línies futures.</b> .....                                 | <b>103</b> |
| <b>10. REFERÈNCIES BIBLIOGRÀFIQUES</b> .....                      | <b>104</b> |
| <b>11. ANNEXOS</b> .....                                          | <b>106</b> |
| <b>11.1. Estructura WBS i diagrama de Gantt.</b> .....            | <b>106</b> |
| <b>11.2. Codis MATLAB.</b> .....                                  | <b>107</b> |
| 11.2.1. Programa funcioCalculaResposta.m .....                    | 107        |
| 11.2.2. Programa funcioRandomMatrix.m.....                        | 107        |
| 11.2.3. Programa processarDadesSensors.m.....                     | 108        |
| <b>11.3. Programes comunicació NFC i obtenció de dades.</b> ..... | <b>109</b> |
| 11.3.1. Classe tagM24SR .....                                     | 109        |
| 11.3.2. Programa principal. ....                                  | 111        |
| <b>11.4. Activitats aplicació mòbil.</b> .....                    | <b>115</b> |
| 11.4.1. MainActivity. ....                                        | 115        |
| 11.4.2. DataActivity. ....                                        | 120        |
| 11.4.3. DataAdapter. ....                                         | 121        |
| 11.4.4. LineGraphActivity. ....                                   | 122        |

## 1. INTRODUCCIÓ

En la introducció d'aquest treball es presentaran diversos aspectes en relació amb la tria del projecte, els objectius principals que es volen aconseguir amb el desenvolupament del treball i, finalment, es presentarà l'estructura de la memòria, la qual engloba els diferents punts tractats en la execució del projecte.

### 1.1. Planificació del projecte.

La planificació del projecte ens permet definir les diferents tasques a realitzar, i que engloben el conjunt del treball. És per això que s'ha decidit utilitzar l'estructura *WBS (Work Breakdown Structure)*, la qual permet la divisió del projecte en diferents activitats o tasques. La definició d'aquesta estructura, ve acompanyada del desenvolupament d'un diagrama de *Gantt*, permetent la visualització de la duració temporal de cada una de les tasques, així com del conjunt del projecte.

En l'*apartat 11.1* relatiu als *Annexos*, es pot observar l'estructura *WBS* corresponent a la distribució de les diferents tasques i activitats realitzades en el projecte, així com el respectiu diagrama de *Gantt* que identifica la seva distribució temporal.

### 1.2. Objectius del projecte.

Els principals objectius que es pretenen assolir amb la realització d'aquest treball són els següents:

- Desenvolupar un sensor químic resistiu capaç de detectar diòxid de nitrogen.
- Adquirir i processar les dades obtingudes del sensor mitjançant el microcontrolador *Raspberry Pi Pico*.
- Aprofundir i ampliar els coneixements previs sobre *Near Field Communication (NFC)*.
- Programar les comunicacions *NFC* que permetran l'intercanvi de dades entre el microcontrolador i l'aplicació mòbil.
- Desenvolupar una aplicació mòbil capaç d'obtenir les dades mitjançant *NFC* i representar-ne els resultats per a poder ser interpretats per l'usuari.

### 1.3. Estructura de la memòria.

L'estructura del projecte consta dels següents apartats:

**Introducció.** Es defineixen la planificació del projecte i els principals objectius que es pretenen assolir amb la realització del treball.

**Marc teòric.** Es descriuen els principals efectes del diòxid de nitrogen en ambients interiors. S'expliquen conceptes bàsics dels sensors químic resistius, així com les metodologies emprades per a la detecció dels gasos i els factors que hi influeixen. A continuació, s'anomenen les característiques del sensor basat en grafè.

**Desenvolupament del sensor.** S'anomenen els materials i la metodologia emprada per a la fabricació del sensor químic resistiu de grafè.

**Resposta i anàlisi dels resultats amb *MATLAB*.** Es detallen els resultats obtinguts de la mesura dels sensors sota unes determinades concentracions de NO<sub>2</sub>. Es concreta el procediment i programes utilitzats per al càlcul de la resposta, així com per a la construcció del model qualitatiu que permet diferenciar el rang de resposta en funció de la concentració de gas.

**Tecnologia NFC.** Aborda tots els conceptes teòrics relatius a la tecnologia NFC per tal d'entendre el seu principi de funcionament i el format usat per a l'intercanvi de dades, així com el conjunt de protocols i estàndards que integra.

**Processat de dades amb *Raspberry Pi Pico* i *Python*.** S'introdueix el hardware *Raspberry Pi Pico* i es compara amb Arduino. Es descriu el mòdul NFC que permet establir comunicació entre el microcontrolador i l'aplicació mòbil, així com la comunicació sèrie emprada entre el microcontrolador i el mòdul NFC. Es detalla el procés i circuit d'adaptació. Finalment, es presenta el codi desenvolupat que permet realitzar les funcions presentades anteriorment.

**Aplicació mòbil per gestionar i visualitzar les dades.** S'expliquen conceptes bàsics sobre la programació de dispositius mòbils fent ús de l'entorn de desenvolupament integrat *Android Studio*. Es presenta l'estructura de l'aplicació i com s'ha desenvolupat. S'exposen els principals mètodes utilitzats i els nous mètodes implementats.

**Proves i resultats.** Es detalla el resultat obtingut referent a l'avaluació del projecte complet, des de l'exposició del sensor a diferents concentracions de diòxid de nitrogen (NO<sub>2</sub>), fins a la visualització de les dades capturades a través de l'aplicació mòbil desenvolupada.

**Conclusions i línies futures.** S'anomenen les conclusions extretes de l'elaboració d'aquest projecte i s'exposen línies de treball futures que podrien aportar una millora del sistema complet.

**Annexes.** Es mostra l'estructura *WBS* i el diagrama de *Gantt* corresponent a la planificació del projecte, així com els codis corresponents als programes desenvolupats d'acord amb les diferents parts del projecte i que no han sigut exposats durant l'explicació de la il·lativa implementació.

## 2. MARC TEÒRIC

En aquest apartat s'exposaran els conceptes necessaris pel que fa als sensors químics resistius i l'impacte del diòxid de nitrogen davant dels diferents escenaris d'exposició. A més a més, s'explicaran els mecanismes de detecció de gasos d'aquests dispositius i els factors principals que hi intervenen.

### 2.1. Diòxid de nitrogen (NO<sub>2</sub>).

El diòxid de nitrogen, pertany a un grup de gasos coneguts com òxids de nitrogen NO<sub>x</sub> [1]. Es tracta d'un gas tòxic que produeix efectes adversos sobre la salut. Principalment, aquest gas es forma en grans centrals elèctriques degut als processos que es duen a terme a molt altes temperatures [1].

D'altra banda, també es poden trobar concentracions d'aquest gas en les ciutats o zones urbanes, així com regions on hi hagi una gran circulació de vehicles que facin ús de motors de combustió, donat que l'emissió de monòxid de carboni (CO) dels vehicles, deriva a la pol·lució de diòxid de nitrogen [2]. La formació d'aquest tipus de gas sorgeix degut a que el nitrogen, que forma part del material combustible, pot combinar-se per amb l'oxigen i generar monòxid de carboni (NO), el qual entrarà en contacte químicament amb l'oxigen de l'ambient, resultant a la formació de diòxid de nitrogen (NO<sub>2</sub>) [1][2].

Altrament, en ambients interiors, es poden donar casos d'exposició a aquest gas degut, principalment, a les calefaccions domèstiques quan no funcionen correctament i per tant, la combustió, en aquest cas de nitrogen, és incompleta. És a dir, que no hi ha suficient oxigen en l'ambient que actui com a comburent i que, per tant, reaccioni químicament amb combustible, en aquest cas, el nitrogen [1].

#### 2.1.1. *Efectes adversos sobre la salut.*

Els principals efectes adversos degut a l'exposició i inhalació de moderades concentracions de diòxid de nitrogen són tos, irritació de les vies respiratòries i els ulls, sensació de falta d'aire i fatiga. Altrament, si les exposicions són altes, es poden produir efectes més greus, com ara bé la reducció de la quantitat d'oxigen al conjunt de teixits del cos, així com una acumulació de líquids als pulmons, disminuint la funció pulmonar i l'augment de malalties respiratòries degut a una reducció de la resistència enfront aquestes afeccions [2].

Pel que fa als diferents efectes d'exposició presentats anteriorment, cal destacar que aquests es veuran incrementats en persones vulnerables, com ara població d'edat avançada i nens menors de 6 anys, així com gent que pateixi asma, al·lèrgies o malalties respiratòries cròniques, entre d'altres [2].

### 2.1.2. Nivells recomanats i legislats de NO<sub>2</sub>.

Els valors límit d'exposició a diòxid de nitrogen en ambients exteriors establerts per la OMS, són els següents, on les mesures corresponen a una temperatura normalitzada de 20 °C. [3].

| Ambients exteriors                                               | Temps d'exposició | Valor límit           | Tolerància           |
|------------------------------------------------------------------|-------------------|-----------------------|----------------------|
| Valor límit horari per garantir la protecció de la salut humana. | 1 hora            | 200 µg/m <sup>3</sup> | 10 µg/m <sup>3</sup> |
| Valor límit anual per garantir la protecció de la salut humana.  | 1 any             | 40 µg/m <sup>3</sup>  | 2 µg/m <sup>3</sup>  |

**Taula 2.1.** Nivells legislats d'exposició de NO<sub>2</sub> en ambients exteriors.

Pel que fa al valor límit horari, s'especifica que aquest valor no podrà ser superat més de 18 vegades en un any.

En ambients interiors on no es disposi d'aparells de combustió com ara estufes de gas o calefactores de gas sense ventilació, els nivells establerts corresponen aproximadament a la meitat del valor màxim corresponent a ambients exteriors. D'altra banda, si ens trobem en el cas en que en l'ambient interior es disposa d'elements calefactores com els mencionats anteriorment, els nivells d'exposició de gas podrien ser molt iguals o, fins i tot, superar als establerts en ambients exteriors [4].

| Ambients interiors                                               | Temps d'exposició | Valor límit           | Tolerància          |
|------------------------------------------------------------------|-------------------|-----------------------|---------------------|
| Valor límit horari per garantir la protecció de la salut humana. | 1 hora            | 100 µg/m <sup>3</sup> | 5 µg/m <sup>3</sup> |
| Valor límit anual per garantir la protecció de la salut humana.  | 1 any             | 20 µg/m <sup>3</sup>  | 1 µg/m <sup>3</sup> |

**Taula 2.2.** Nivells legislats d'exposició de NO<sub>2</sub> en ambients interiors.

### 2.2. Conceptes bàsics dels sensors químic resistius.

Els sensors químic resistius, son un tipus de sensors de gas formats per un material sensible al gas, que interactua amb la substància química a mesurar i generen una resposta, on la senyal de sortida és expressada com una variació en la seva resistència elèctrica. El canvi en la resistència elèctrica està directament relacionat amb el nivell de concentració del gas. Això és degut ja que la reacció química que es dona entre el material metàl·lic o semiconductor de la superfície i el gas, porten a la formació d'òxids superficials, que es tradueixen en la

generació d'un potencial elèctric, que com s'ha comentat abans, és proporcional a la concentració del reactiu [5].

### 2.2.1. Estructura del sensor químic resistiu.

L'estructura principal d'un sensor químic resistiu és la següent [5][6]:

- **Substrat.** Es tracta d'un material, com per exemple pot ser l'alumínia, que forma la base del sensor i per tant, és on s'integren tots els elements que formen el conjunt del dispositiu [5][56].
- **Elèctrodes.** Permeten traduir el nivell de concentració de gas a variacions de resistència.
- **Escalfador.** Degut a que la reacció química es duu a terme a altes temperatures, aquest element és utilitzat per a incrementar la temperatura de l'àrea on es duu a terme la reacció química. També permet reduir el temps de recuperació de la resposta [5][6].
- **Capa de material.** Es col·loca una capa del material triat per a que reaccioni amb la substància a examinar. Reacciona químicament amb el gas i varia el potencial elèctric en la superfície [5][6].

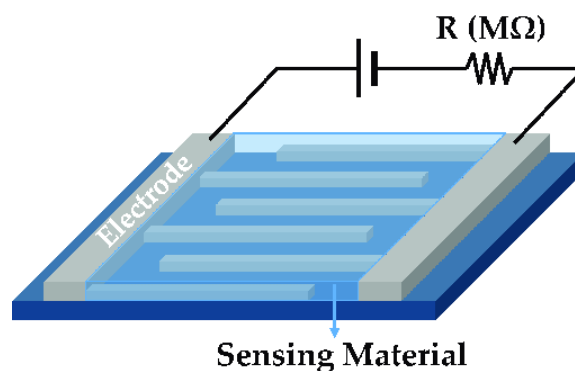


Figura 2.1.. Model d'estructura d'un sensor químic resistiu.

### 2.2.2. Paràmetres que en defineixen el comportament.

Tot i que el comportament principal d'un sensor químic resistiu és el canvi de la conductivitat elèctrica en funció de la concentració d'una determinada substància, es defineixen un conjunt de paràmetres a tindre en compte quan fem referència a l'anàlisi del comportament d'un sensor [6].

- **Resposta del sensor.** Correspon a la variació de la resistència del sensor quan aquest és exposat a un gas. Els paràmetres que defineixen l'equació varien en funció de si el material utilitzat és de tipus  $p$  o de tipus  $n$ , i de si del gas és oxidant o reductor [6].

L'equació que caracteritza la resposta del sensor, en tant per cent, és la següent [6]:

$$S(\%) = \frac{R_g - R_o}{R_o} \cdot 100 \quad (2.1)$$

On  $R_g$  és la resistència del sensor sota l'exposició del gas en un instant de mesura i  $R_o$ , correspon a la resistència del sensor abans de l'exposició al gas, així doncs,  $R_o$  és directament la resistència del sensor exposada a l'aire, o resistència de referència, sense interacció amb cap substància.

- Estabilitat. Correspon a l'habilitat del sensor per a mantenir el seu comportament de detecció davant l'exposició d'una mateixa concentració del gas objectiu al llarg del temps [5][6].
- Selectivitat. La selectivitat és la capacitat del sensor de diferenciar la detecció del gas objectiu a mesurar enfront d'altres gasos que poden interferir en la mesura [5][6].
- Temps de resposta. Correspon al temps de resposta del sensor davant de la presència d'un gas. Per a la identificació d'aquest paràmetre, es pren el temps necessari per a superar en un 90 % l'increment de resistència, sabent que l'increment de resistència o  $\Delta R$ , correspon a la diferència entre la  $R_{max}$  i la  $R_{min}$  que caracteritzen al sensor [6][7].

L'equació que expressa el valor de la resistència corresponent al límit abans de la saturació de la resposta, és la següent:

$$R_{resposta} = R_{min} + (0,9 \cdot \Delta R) \quad (2.2)$$

Finalment, podem expressar el temps de resposta del sensor com:

$$\tau = t(R_{resposta}) - t(R_{min}) \quad (2.3)$$

A partir del resultat anterior, es conclou que a menor temps de resposta ( $\tau$ ), millor comportament presenta el sensor en front la detecció de gasos [7].

- Temps de recuperació. Fa referència al temps necessari perquè el sensor retorni al valor llindar corresponent al 90 % de  $\Delta R$  un cop s'ha entrat en estat de saturació, després que aquest no estigui sotmès a l'exposició de gas. Anàlogament, si es té en compte l'expressió del valor de resistència del temps de resposta (2), es pot expressar el valor de resistència de recuperació com:

$$R_{recuperació} = R_{min} + (0,1 \cdot \Delta R) \quad (2.4)$$

Finalment, podem expressar el temps de recuperació del sensor com:

$$\tau = t(R_{recuperació}) - t(R_{max}) \quad (2.5)$$

A continuació observem una imatge que il·lustra els dos comportaments descrits anteriorment, el temps de resposta i el temps de recuperació. Així doncs, com s'ha mencionat a priori, disposem d'un estat inicial (I), un temps de resposta (II) i un temps de recuperació (III). Aquest últim correspon a l'interval de temps en que el sensor no és exposat al gas, i fins que aquest recupera l'estat inicial, o fins un valor corresponen al increment d'un 10 % del valor inicial (I).

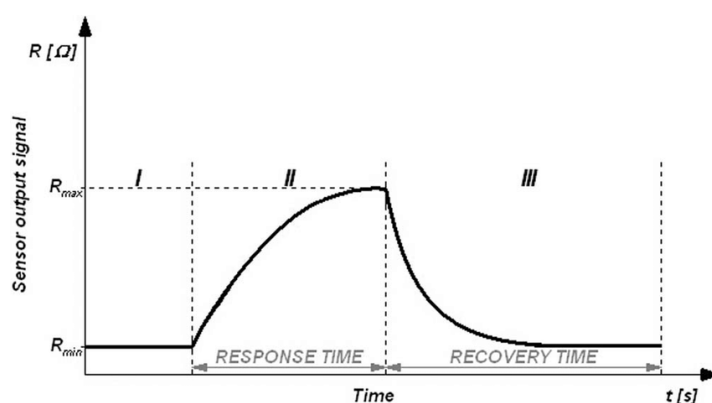


Figura 2.2. Imatge que il·lustra el temps de recuperació i de resposta d'un sensor davant l'exposició de gas.

### 2.3. Propietats del grafè.

Degut a l'estructura que presenta el grafè, aquesta li atribueix un conjunt de propietats que el fan molt versàtil per a un gran conjunt d'aplicacions. Aquest material, es comporta com un semiconductor de tipus p, i per tant, els seus portadors de càrrega estan formats bàsicament per forats. Això es tradueix en un increment en la conductància del material quan aquest és sotmès a una substància oxidant, com ho és el diòxid de nitrogen ( $\text{NO}_2$ ) [6].

A continuació, en la Taula 2.1, es detallaran les principals propietats del grafè relacionades amb el que s'ha mencionat a priori i que, per tant, el permeten ser utilitzat en aplicacions per a la detecció de gasos [7].

| Propietats del grafè per a la detecció de gasos |                                                                                                                                                                                                                                                                 |
|-------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Àrea superficial</b>                         | Una àrea superficial alta, es tradueix en un increment del nombre de molècules de gas que poden adherir-se a la seva superfície, provocant una millora en la sensibilitat del sensor.                                                                           |
| <b>Conductivitat elèctrica</b>                  | L'elevada conductivitat del grafè és deguda, principalment, a la separació entre les bandes de conducció i de valència. El nivell de Fermi defineix el comportament d'un material (conductor o semiconductor), en funció del nivell d'energia dels electrons de |

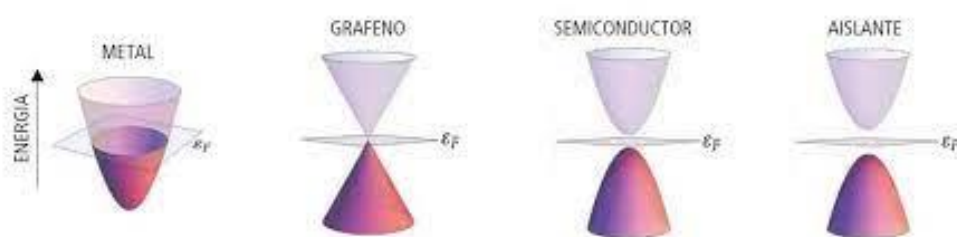
|                    |                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | l'última capa del material. En el cas del grafè, aquesta separació o banda de <i>gap</i> és nul·la, i per tant, el nivell de Fermi es troba en la intersecció entre les bandes de valència i conducció [7]. Això es tradueix en un requeriment d'energia molt baix per a la circulació d'electrons d'una banda a una altra, proporcionant una gran conductivitat elèctrica.       |
| <b>Reactivitat</b> | La capacitat de detectar gasos es deu principalment als canvis en la conductància en el procés d'adhesió de les molècules de gas [7]. D'aquesta manera, en el cas estudiat en aquest projecte, el diòxid de nitrogen (NO <sub>2</sub> ) interactua com a material oxidant o captador d'electrons, tal i com s'ha comentat abans, augmentant d'aquesta manera la conductància [7]. |

**Taula 2.1.** Propietats del grafè influents en la detecció de gasos.

En la *Taula 2.2* podem observar els valors corresponents a dos de les propietats del grafè comentades en la *Taula 2.1*, així com la comparació d'aquests amb altres materials coneguts [7].

| Propietats                     | Valors                 | Comparació                          |
|--------------------------------|------------------------|-------------------------------------|
| <b>Conductivitat elèctrica</b> | $9,6 \cdot 10^7$ S/m   | <b>Coure:</b> $5,96 \cdot 10^7$ S/m |
| <b>Àrea superficial</b>        | 2630 m <sup>2</sup> /g | <b>Carbó:</b> 500 m <sup>2</sup> /g |

**Taula 2.2.** Característiques d'algunes propietats del grafè.



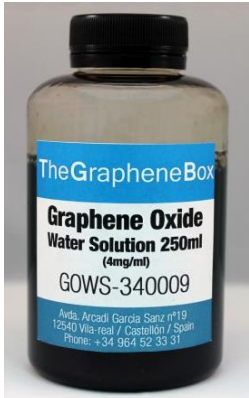

**Figura 2.3.** Comparació de les bandes d'energia entre el grafè i altres tipus de materials.


### 3. DESENVOLUPAMENT DEL SENSOR

En aquest capítol, es detallaran els materials utilitzats i la metodologia que s'ha emprat per tal de fabricar els sensors químic resistius basats en grafè.

#### 3.1. Materials utilitzats.

Abans d'explicar la metodologia utilitzada per a la implementació d'aquests sensors, es llistaran el conjunt de materials que s'han usat:

| Material                           |                                                                                                                                    |                                                                                                 |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| Solució de grafè en aigua.         |  <p><b>Figura 3.1.</b> Solució de grafè.</p>     | S'ha emprat una solució de grafè en aigua amb una proporció de 2,5 mg/5 ml.                     |
| Aerògraf.                          |  <p><b>Figura 3.2.</b> Aerògraf.</p>            | Aerògraf utilitzat per a proporcionar les capes necessàries de grafè a sobre els elèctrodes.    |
| Escalfador elèctric de laboratori. |  <p><b>Figura 3.3.</b> Escalfador elèctric.</p> | Permet evaporar l'aigua de la solució de grafè per a la seva correcta adherència a l'elèctrode. |

| Material    |                                                                                                                         |                                                                                |
|-------------|-------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| Elèctrodes. |  <p><b>Figura 3.4.</b> Elèctrodes.</p> | Mitjançant diverses capes de grafè, permetran la detecció de NO <sub>2</sub> . |

**Taula 3.1.** Descripció del conjunt de materials usats en el desenvolupament del sensor.

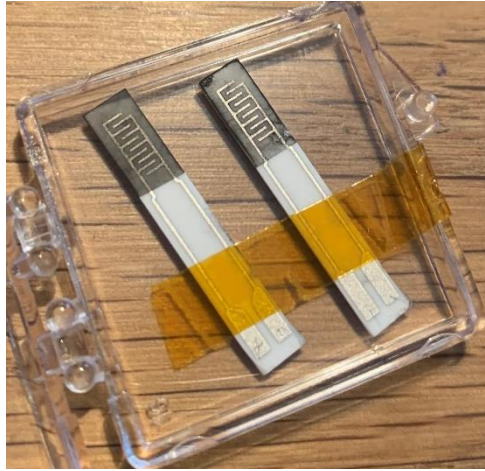
### 3.2. Metodologia.

El mètode utilitzat per al desenvolupament dels sensors químic resistius de grafè, ha estat el següent:

1. Preparació de la solució de grafè en aigua amb una proporció 2,5 mg/5 mL.
2. Agitament de la solució preparada amb un imant.
3. Selecció dels elèctrodes i col·locació d'aquests a sobre de l'escalfador elèctric de laboratori.
4. Introducció de la solució preparada al dipòsit de l'aerògraf. Aplicació de la primera capa a sobre dels elèctrodes.
5. Esperem uns minuts per a que s'evapori l'aigua de la solució, al mateix temps que permetem un bon establiment del grafè a sobre dels elèctrodes.
6. Tornem a repetir el procés d'aplicació de la solució a sobre els elèctrodes en funció del gruix de la capa de grafè desitjada, així com del valor de resistència convenient.
7. Mesura de la resistència dels elèctrodes per a comprovar i ajustar el seu rang d'operació.

Cal recalcar que el nombre de capes de grafè tindrà una relació directe amb la resistència de referència  $R_0$ . Així doncs, com més gruixuda sigui la capa de grafè, menor serà el valor d'aquesta resistència.

Finalment, després de l'execució de cada un dels passos explicats anteriorment, obtenim el següent resultat:



**Figura 3.5.** Parell d'elèctrodes amb grafè.

En la *Figura 3.5*, podem observar el parell d'elèctrodes amb la solució de grafè ja aplicada en cada un d'ells. El nombre de capes de grafè és tal que correspon a una resistència de 335 k $\Omega$  i 895 k $\Omega$  respectivament.

## 4. RESPOSTA I ANÀLISI DELS RESULTATS AMB MATLAB.

Un cop finalitzat el desenvolupament dels sensors, es va realitzar una prova a la sala de caracterització de la universitat, per tal d'avaluar i analitzar el seu comportament enfront l'exposició de NO<sub>2</sub> sota unes determinades condicions de mesura. En aquest punt, es detallaran les condicions de mesura a la que es van sotmetre els sensors i es presentaran els resultats obtinguts. Per altra banda, es detallarà l'anàlisi realitzat per tal de construir un model teòric que ens permetrà estimar la concentració de NO<sub>2</sub> que hi pot haver en un ambient anterior.

### 4.1. Presentació dels resultats obtinguts.

A continuació, podem observar en la *Taula 4.1* les condicions a les que han sigut sotmesos els sensors per tal d'observar la seva reacció davant diferents concentracions de NO<sub>2</sub>.

| Condicions de mesura              |                    |        |
|-----------------------------------|--------------------|--------|
| Temps d'exposició                 | Aire               | 30 min |
|                                   | NO <sub>2</sub>    |        |
| Concentracions de NO <sub>2</sub> | 5, 10, 15 i 20 ppm |        |
| Temperatura d'operació            | 45 °C              |        |

Taula 4.1. Condicions de mesura per a l'avaluació del comportament dels sensors.

Finalment, els resultats que es van obtenir per a cada un dels sensors els podem observar en la *Figura 4.1* i *Figura 4.2* respectivament.

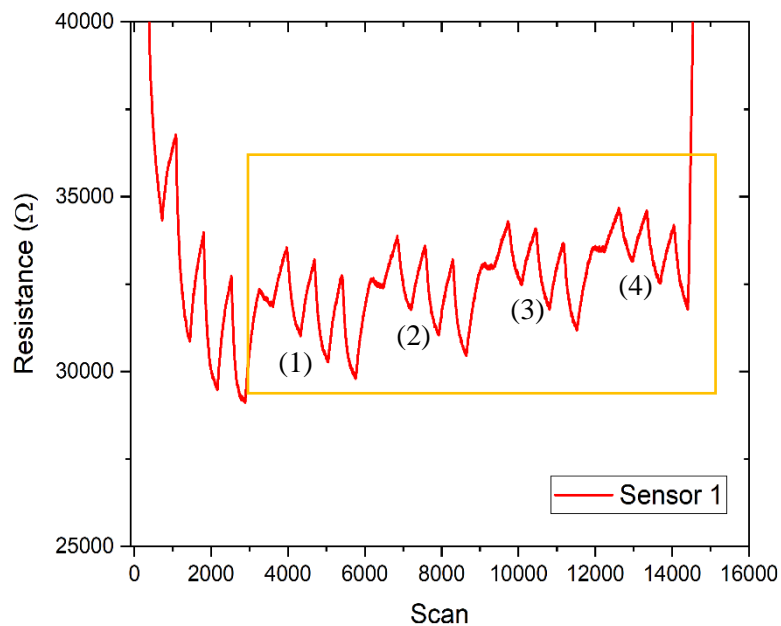
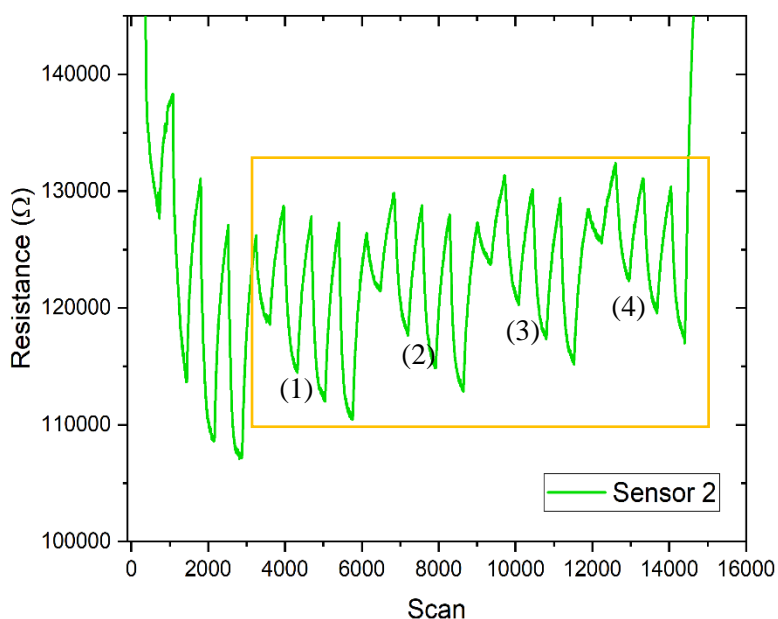
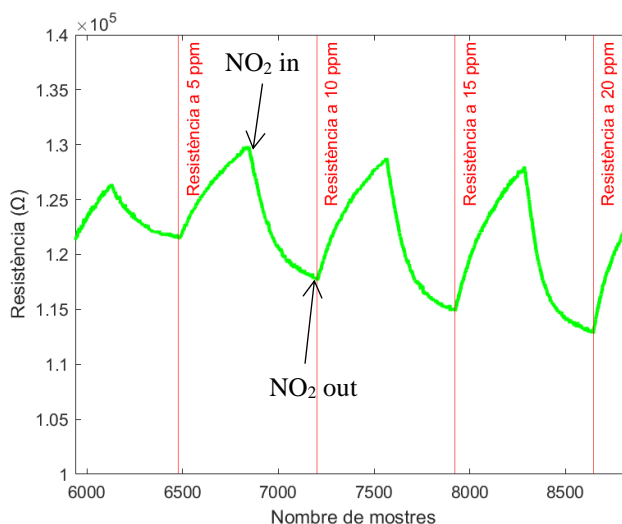


Figura 4.1. Resultats mesures sensor 1.



**Figura 4.2.** Resultats mesures sensor 2.

Dels resultats obtinguts, podem veure que per a cada un dels sensors es van programar quatre cicles o períodes de mesura. Per a cada un dels cicles, es distingeixen les diferents concentracions de  $\text{NO}_2$ , així com un exemple de l'instant on s'introdueix gas i de l'instant on s'introdueix aire, segons podem veure en la *Figura 4.3*.



**Figura 4.3.** Exemple segon cicle sensor 2.

Per altra banda, podem assenyalar com la resistència del sensor disminueix quan aquest és exposat a diferents concentracions de  $\text{NO}_2$ . Per tant, com més elevat és el factor que determina la concentració, més disminueix el valor de la resistència.

## 4.2. Anàlisi dels resultats.

Seguidament, un cop obtinguts els resultats i analitzades les característiques comentades anteriorment, es va procedir amb el seu anàlisi mitjançant *MATLAB*. Tal i com s'ha comentat en la introducció d'aquest apartat, l'objecte d'aquest anàlisi és el desenvolupament d'un model qualitatiu per a determinar, per a un rang de variació de la resistència dels sensors, la concentració de  $\text{NO}_2$  present en l'ambient. Dit això, el procediment seguit per a la generació d'aquest model el podem observar en el diagrama de la *Figura 4.4*.

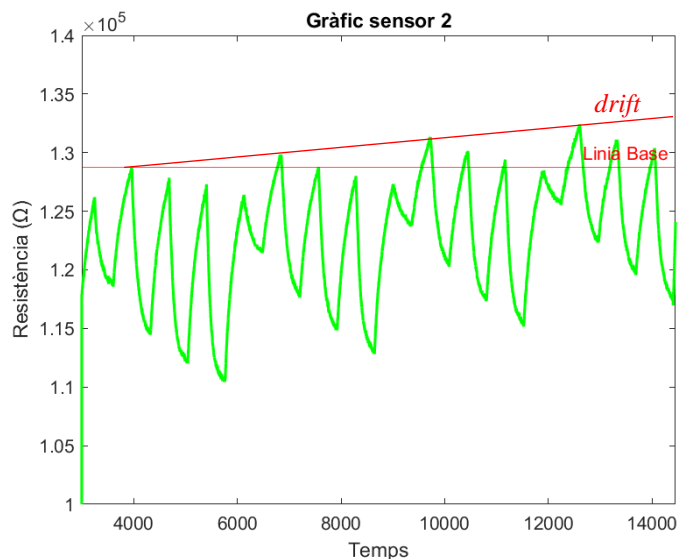


**Figura 4.4.** Procediment seguit per a la elaboració del model.

El programa utilitzat per a l'anàlisi dels resultats el podem trobar en l'*apartat 11.2* dels annexes.

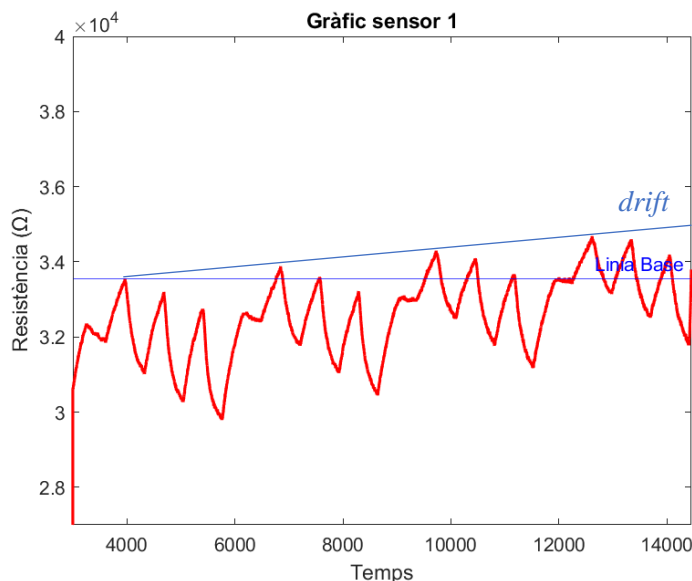
### Correcció del *drift*.

Un cop obtingudes les dades del sensor 1 i el sensor 2, corresponents a la *Figura 4.1* i *Figura 4.2* respectivament, el següent pas realitzat ha sigut la correcció del *drift*. Com bé es pot veure, els resultats obtinguts presenten un *drift* o tendència ascendent degut al desgast dels sensors al llarg del temps que és necessari corregir. En la *Figura 4.5*, podem identificar de manera més precisa el fenomen descrit per al cas de les mesures corresponents al sensor 2.



**Figura 4.5.** Representació del *drift* sensor 2.

D'igual manera, observem un comportament anàleg en les mesures obtingudes del sensor 1, identificat en la *Figura 4.6*.



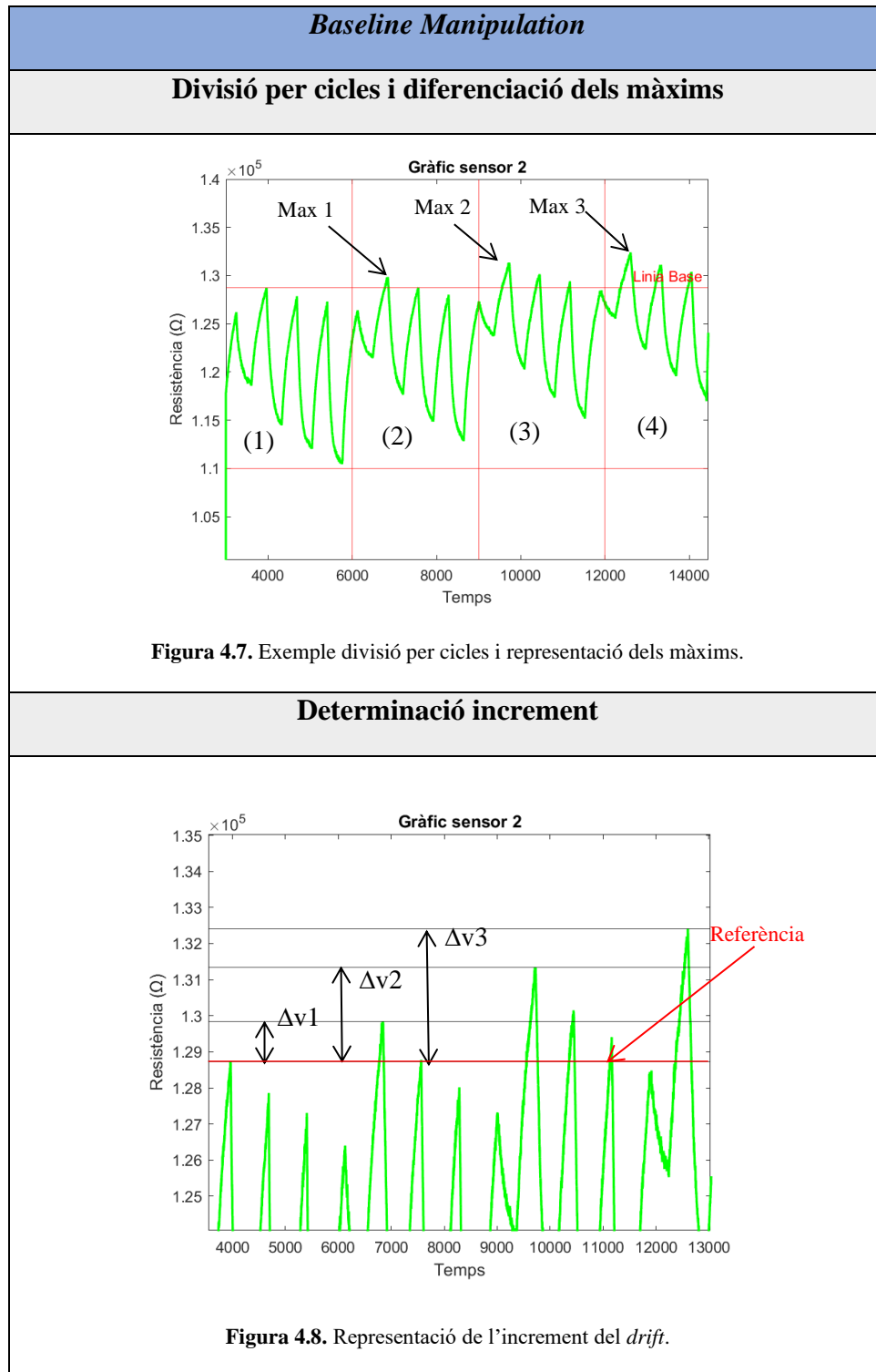
**Figura 4.6.** Representació del *drift* del sensor 1.

D'aquesta manera, s'han desenvolupat dos funcions en *MATLAB* que permeten manipular les dades obtingudes i realitzar els passos segon, tercer i quart del procediment descrit en la *Figura 4.4*.

- *funcioEliminarDrift*
- *funcioCalculResposta*

Per tal d'eliminar el *drift*, s'ha utilitzat el mètode *baseline manipulation* [8], implementat en la *funcióEliminarDrift*, el qual consisteix bàsicament en el càlcul del diferencial de la mesura en referència a la línia base, o *baseline*, i restar-lo al conjunt de la mesura. Per això, es manipula cada un dels quatre cicles de mesura per separat, determinant, per a cada un, el seu valor màxim. Un cop obtingut el valor màxim de cada un dels quatre cicles, es calcula el vector diferencial, que permet definir el valor a restar per a cada un dels quatre períodes.

En la *Taula 4.2*, es representa gràficament el procediment descrit per al cas dels resultats corresponents al sensor 2.

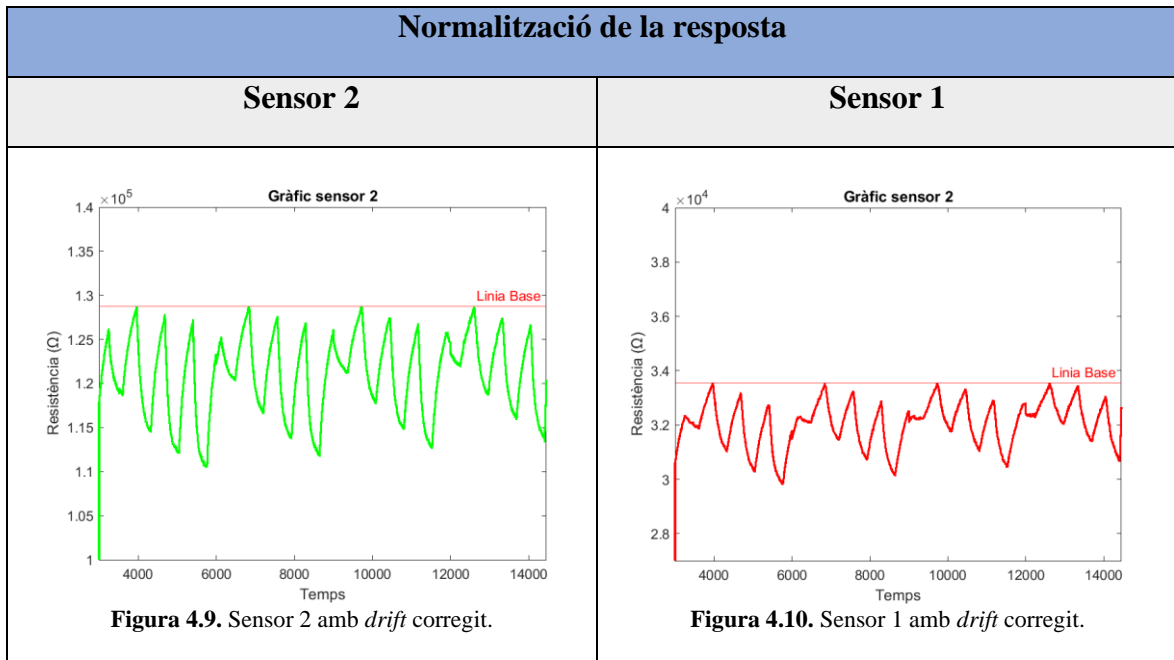


**Taula 4.2.** Representació gràfica del procés de correcció del *drift*.

On  $\Delta v_1$ ,  $\Delta v_2$ ,  $\Delta v_3$  representen l'increment dels valors màxims en referència a la línia base. Així doncs, el vector de diferències es representa segons podem observar a continuació.

$$\text{vector\_dif} = [0 \ \Delta v_1 \ \Delta v_2 \ \Delta v_3]$$

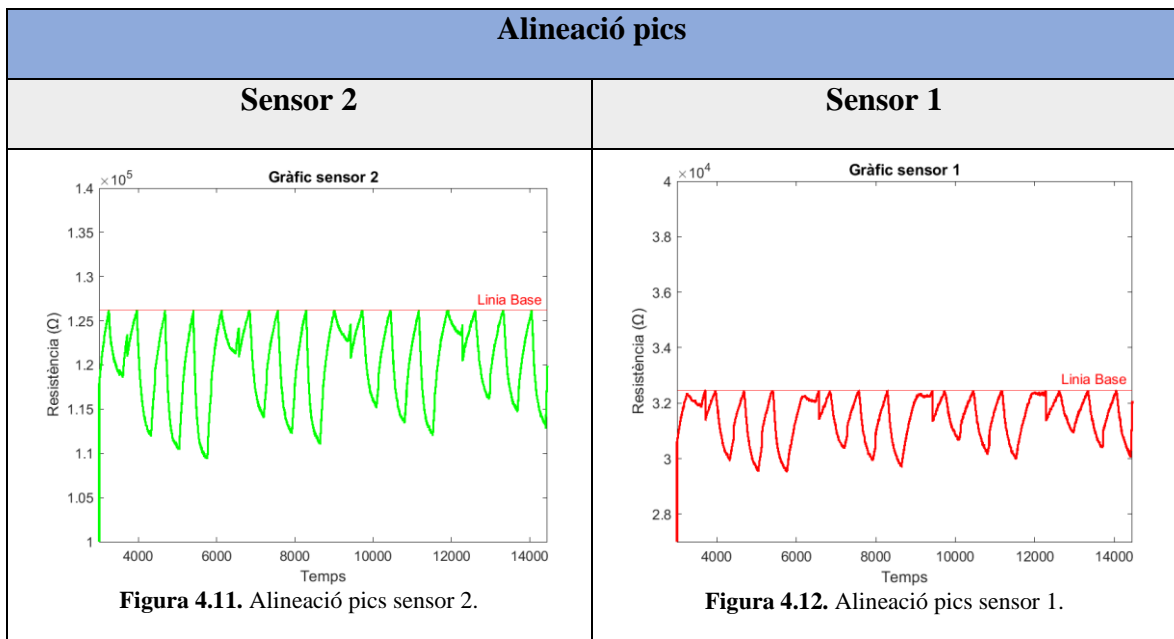
Finalment, el resultat que s'obté al restar cada un d'aquests valors per a cada *frame*, és la normalització dels resultats a la línia base, tal i com es pot observar en la *Figura 4.9*, per al cas del sensor 2, i en la *Figura 4.10* per al cas del sensor 1.



**Taula 4.3.** Resultats dels sensors amb normalització de la resposta.

### Alineació dels pics.

A continuació, es va decidir alinear cada un dels pics a la línia base mitjançant el mateix procediment però realitzant la divisió de les mesures per concentracions enlloc de per cicles. Així doncs, d'aquesta manera s'obtenen setze *frames*, en comparació amb el procés anterior, en que s'obtenien quatre. El resultat d'aquesta aproximació el podem observar en la *Taula 4.3*.



**Taula 4.4.** Exemple resultats alineació dels pics.

Seguidament, podem observar en la *Figura 4.13*, el diagrama de flux que identifica el funcionament del programa *funcioEliminarDrift*, el qual representa els procediments per a corregir el *drift* i alinear els màxims.

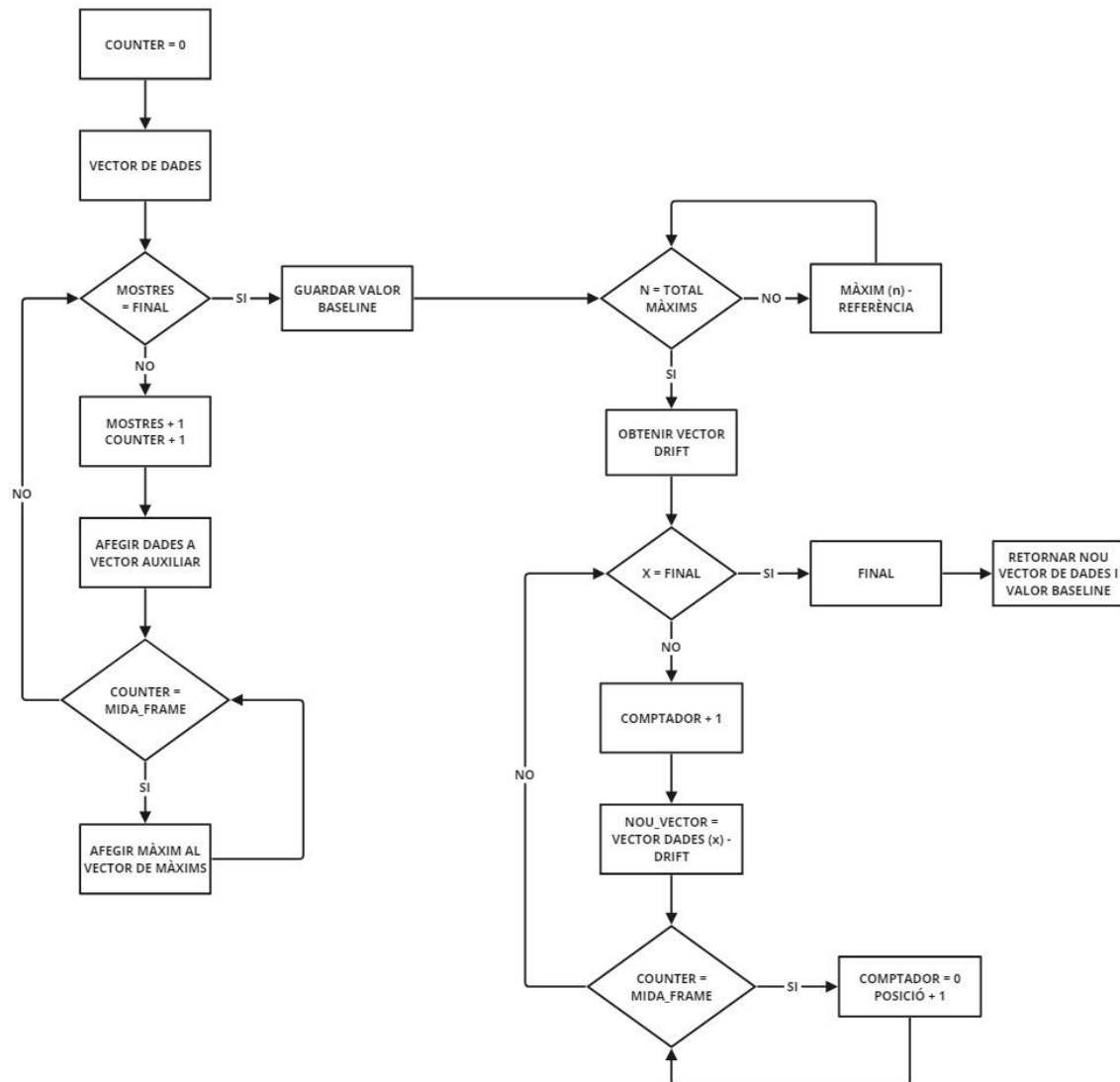


Figura 4.13. Diagrama de flux *funcioEliminarDrift*.

### Càlcul de la resposta.

Per al càlcul de la resposta, s'ha utilitzat la funció *funcioCalculResposta*, comentada anteriorment. Segons l'expressió (2.1), el càlcul de la resposta s'expressa com la relació entre la resistència del sensor en exposició a un gas i la resistència del sensor en contacte amb l'aire. En el nostre cas, l'equació (2.1) és veu lleugerament alterada, ja que els sensors realitzats experimenten un decrement del valor de resistència segons la quantitat de gas que es detecta. Així doncs, l'equació es pot reescriure com:

$$S(\%) = \frac{R_o - R_g}{R_o} \cdot 100 \quad (4.1)$$

On  $R_o$  corresponen a la presència del sensor sense exposició al gas i  $R_g$  correspon a la resistència quan es detecta gas.

En conseqüència, tenint en compte la *Figura 4.3* on s'identifiquen gràficament els valors de resistència que caracteritzen el càlcul de la resposta, la rutina *MATLAB* desenvolupada és

molt semblant a la utilitzada per a la alineació dels pics, on es divideix la gràfica en setze *frames* corresponents a cada una de les concentracions per a cada cicle o període. D'aquesta manera, al realitzar aquesta divisió, es calcula el màxim i el mínim de cada *frame* i s'aplica l'expressió (4.1). A continuació, es presenta un exemple gràfic del procediment seguit. En la *Figura 4.15* podem distingir entre la resistència  $R_o$  (línia discontinúta negra) i la resistència  $R_g$  (línia discontinúta vermella), que permeten determinar la resposta per a les concentracions esmentades.

| Exemple divisió <i>frames</i> per concentració i càlcul de la resposta                                                                          |                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Divisió per <i>frames</i> primer període de mostres                                                                                             | Exemple procediment càlcul de la resposta                                                                                                                                                                                                                                                          |
| <p>Gràfic sensor 1<br/>Primer període</p> <p>Resistència (<math>\Omega</math>) <math>\times 10^5</math></p> <p>Temps</p> <p>(1) (2) (3) (4)</p> | <p>Gràfic sensor 1</p> <p>Resistència (<math>\Omega</math>) <math>\times 10^5</math></p> <p>Temps</p> <p><math>R_o</math> 5ppm <math>R_g</math> 5ppm <math>R_o</math> 10ppm <math>R_g</math> 10ppm <math>R_o</math> 15ppm <math>R_g</math> 15ppm <math>R_o</math> 20ppm <math>R_g</math> 20ppm</p> |
| <p><b>Figura 4.14.</b> Exemple divisió per <i>frames</i> primer període.</p>                                                                    | <p><b>Figura 4.15.</b> Determinació valors de resistència per al càlcul de la resposta.</p>                                                                                                                                                                                                        |

**Taula 4.5.** Procediment per al càlcul de la resposta.

A continuació, es pot observar el diagrama de flux que representa el procediment del càlcul de la resposta mencionat anteriorment.

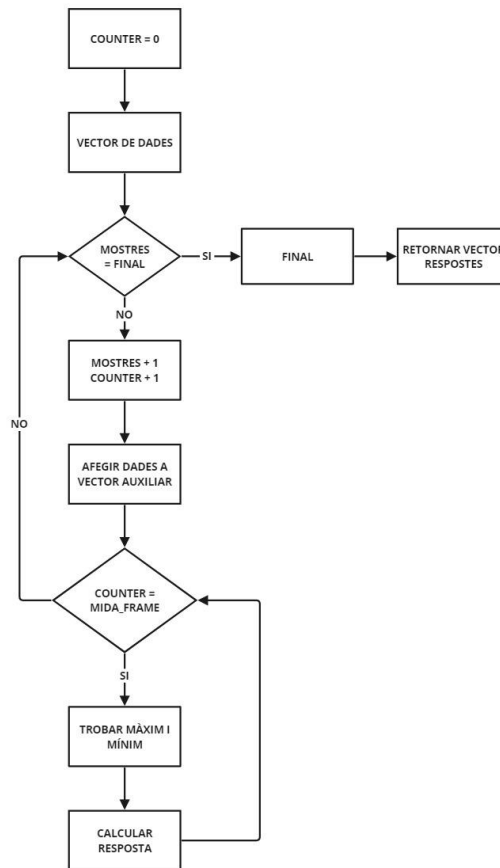


Figura 4.15. Diagrama de flux del càlcul de les respostes.

Un cop s’han calculat les respostes per a cada un dels sensors, es generen les matrius que identifiquen el valor de la resposta per a cada una de les concentracions i cicle. Tot seguit, podem observar cada una de les dos matrius. Cada una de les files correspon als quatre cicles de mesura, mentre que cada columna identifica les diferents concentracions de NO<sub>2</sub>.

| Matriu de respostes                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                    |         |         |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |        |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |         |         |         |        |        |         |         |        |        |        |         |        |        |        |         |       |        |        |        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|--------|--------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|---------|---------|---------|--------|--------|---------|---------|--------|--------|--------|---------|--------|--------|--------|---------|-------|--------|--------|--------|
| Respostes sensor 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Respostes sensor 2 |         |         |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |        |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |         |         |         |        |        |         |         |        |        |        |         |        |        |        |         |       |        |        |        |
| <p>matrixResp =</p> <table border="1"> <tr> <td>5.5665</td> <td>7.5641</td> <td>8.7379</td> <td>8.8264</td> </tr> <tr> <td>6.1908</td> <td>6.3160</td> <td>7.6287</td> <td>8.2879</td> </tr> <tr> <td>6.3128</td> <td>5.4170</td> <td>6.9508</td> <td>7.4614</td> </tr> <tr> <td>6.1529</td> <td>4.6043</td> <td>6.2309</td> <td>7.2066</td> </tr> <tr> <td>5 ppm</td> <td>10 ppm</td> <td>15 ppm</td> <td>20 ppm</td> </tr> </table> <p>Figura 4.16. Matriu respostes sensor 1.</p> | 5.5665             | 7.5641  | 8.7379  | 8.8264 | 6.1908 | 6.3160 | 7.6287 | 8.2879 | 6.3128 | 5.4170 | 6.9508 | 7.4614 | 6.1529 | 4.6043 | 6.2309 | 7.2066 | 5 ppm | 10 ppm | 15 ppm | 20 ppm | <p>matrixResp2 =</p> <table border="1"> <tr> <td>6.6013</td> <td>11.0966</td> <td>12.3945</td> <td>13.2537</td> </tr> <tr> <td>6.8103</td> <td>9.4157</td> <td>10.8210</td> <td>11.8756</td> </tr> <tr> <td>7.2267</td> <td>8.4466</td> <td>9.8607</td> <td>11.0190</td> </tr> <tr> <td>8.0220</td> <td>7.6320</td> <td>8.8167</td> <td>10.2859</td> </tr> <tr> <td>5 ppm</td> <td>10 ppm</td> <td>15 ppm</td> <td>20 ppm</td> </tr> </table> <p>Figura 4.17. Matriu respostes sensor 2.</p> | 6.6013 | 11.0966 | 12.3945 | 13.2537 | 6.8103 | 9.4157 | 10.8210 | 11.8756 | 7.2267 | 8.4466 | 9.8607 | 11.0190 | 8.0220 | 7.6320 | 8.8167 | 10.2859 | 5 ppm | 10 ppm | 15 ppm | 20 ppm |
| 5.5665                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 7.5641             | 8.7379  | 8.8264  |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |        |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |         |         |         |        |        |         |         |        |        |        |         |        |        |        |         |       |        |        |        |
| 6.1908                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 6.3160             | 7.6287  | 8.2879  |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |        |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |         |         |         |        |        |         |         |        |        |        |         |        |        |        |         |       |        |        |        |
| 6.3128                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 5.4170             | 6.9508  | 7.4614  |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |        |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |         |         |         |        |        |         |         |        |        |        |         |        |        |        |         |       |        |        |        |
| 6.1529                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 4.6043             | 6.2309  | 7.2066  |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |        |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |         |         |         |        |        |         |         |        |        |        |         |        |        |        |         |       |        |        |        |
| 5 ppm                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 10 ppm             | 15 ppm  | 20 ppm  |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |        |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |         |         |         |        |        |         |         |        |        |        |         |        |        |        |         |       |        |        |        |
| 6.6013                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 11.0966            | 12.3945 | 13.2537 |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |        |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |         |         |         |        |        |         |         |        |        |        |         |        |        |        |         |       |        |        |        |
| 6.8103                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 9.4157             | 10.8210 | 11.8756 |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |        |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |         |         |         |        |        |         |         |        |        |        |         |        |        |        |         |       |        |        |        |
| 7.2267                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 8.4466             | 9.8607  | 11.0190 |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |        |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |         |         |         |        |        |         |         |        |        |        |         |        |        |        |         |       |        |        |        |
| 8.0220                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 7.6320             | 8.8167  | 10.2859 |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |        |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |         |         |         |        |        |         |         |        |        |        |         |        |        |        |         |       |        |        |        |
| 5 ppm                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 10 ppm             | 15 ppm  | 20 ppm  |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |        |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |         |         |         |        |        |         |         |        |        |        |         |        |        |        |         |       |        |        |        |

Taula 4.6. Matriu càlcul de respostes.

**Test ANOVA.**

Per tal de poder analitzar les diferències entre les respostes per a cada una de les concentracions, s'ha realitzat un anàlisi de la variància. La consecució d'aquesta exploració, ens permetrà el desenvolupament del model que identifica el rang de respostes que correspon a cada un dels grups (concentracions) analitzats, a la vegada que permet la diferenciació entre la resta de grups.

Primerament, es va realitzar el test amb les dades de les respostes observades en la *Taula 4.4* per tal de determinar la correctesa del model. Per això, es fa especialment rellevant el resultat que s'obté al calcular la *F-distribution*, la qual permet analitzar la hipòtesi nul·la, que indica la indiferència entre les mitjanes dels diferents grups, i poder determinar amb quin percentatge de significança aquesta es pot rebutjar. Així doncs, la *F-distribution* es pot calcular com:

$$F = \frac{SS_B/(k-1)}{SS_W/(n-k)} \quad (4.2)$$

On  $n$  representa el nombre total de mostres i  $k$  el total de grups de l'experiment. D'aquesta manera,  $(k-1)$  i  $(n-k)$  identifiquen els graus de llibertat  $df_1$  i  $df_2$  que permeten determinar la distribució sota la hipòtesi nul·la. Com més gran sigui el valor de  $F$ , amb més percentatge de significança podem rebutja la hipòtesi nul·la. Per altra banda,  $SS_B$  correspon a la variació entre els diferents grups i  $SS_W$  la variació dins de cada grup. Aquestes, es representen segons les expressions (4.3) i (4.4) observades a continuació.

$$SS_B = \sum_{i=1}^k n_i (\bar{y}_i - \bar{y})^2 \quad (4.3)$$

$$SS_W = \sum_{i=1}^k \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2 \quad (4.4)$$

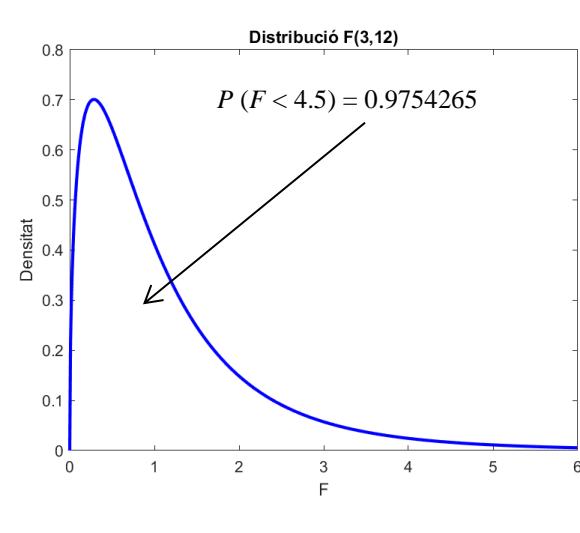
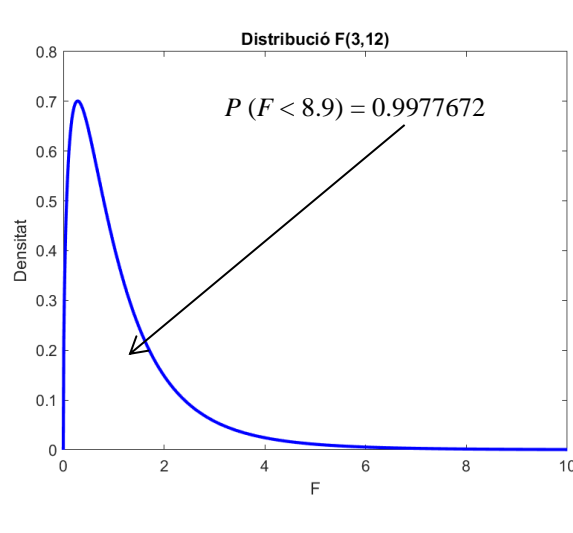
En l'equació (4.3),  $n_i$  representa el nombre de mostres de cada un dels grups,  $\bar{y}_i$  la mitjana de cada grup i  $\bar{y}$  la mitjana global. Altrament, en l'equació (4.4),  $y_{ij}$  representa cada una de les diferents mostres  $j$  analitzades dins d'un grup  $i$ .

Utilitzant la funció *anova1* en *MATLAB*, s'han obtingut els següents resultats per a les expressions (4.2), (4.3) i (4.4) expressades anteriorment.

| <b>Resultats ANOVA</b>       |                 |                 |
|------------------------------|-----------------|-----------------|
|                              | <b>Sensor 1</b> | <b>Sensor 2</b> |
| <b><i>F-distribution</i></b> | 4,5             | 8,9             |
| <b><i>SS<sub>B</sub></i></b> | 11,537          | 43,7204         |
| <b><i>SS<sub>W</sub></i></b> | 10,2484         | 19,6467         |

**Taula 4.7.** Resultats test ANOVA.

Per tant, s'han obtingut els resultats de  $F(f) = 4.5$  en el cas del sensor 1 i de  $F(f) = 8.9$  en el cas del sensor 2. Així doncs és necessari calcular el valor de  $p$  per tal de determinar si l'estudi realitzat és estadísticament significatiu. En la *Taula 4.7*, s'expressen els resultats obtinguts tenint en compte la distribució de la funció de probabilitat  $F$ .

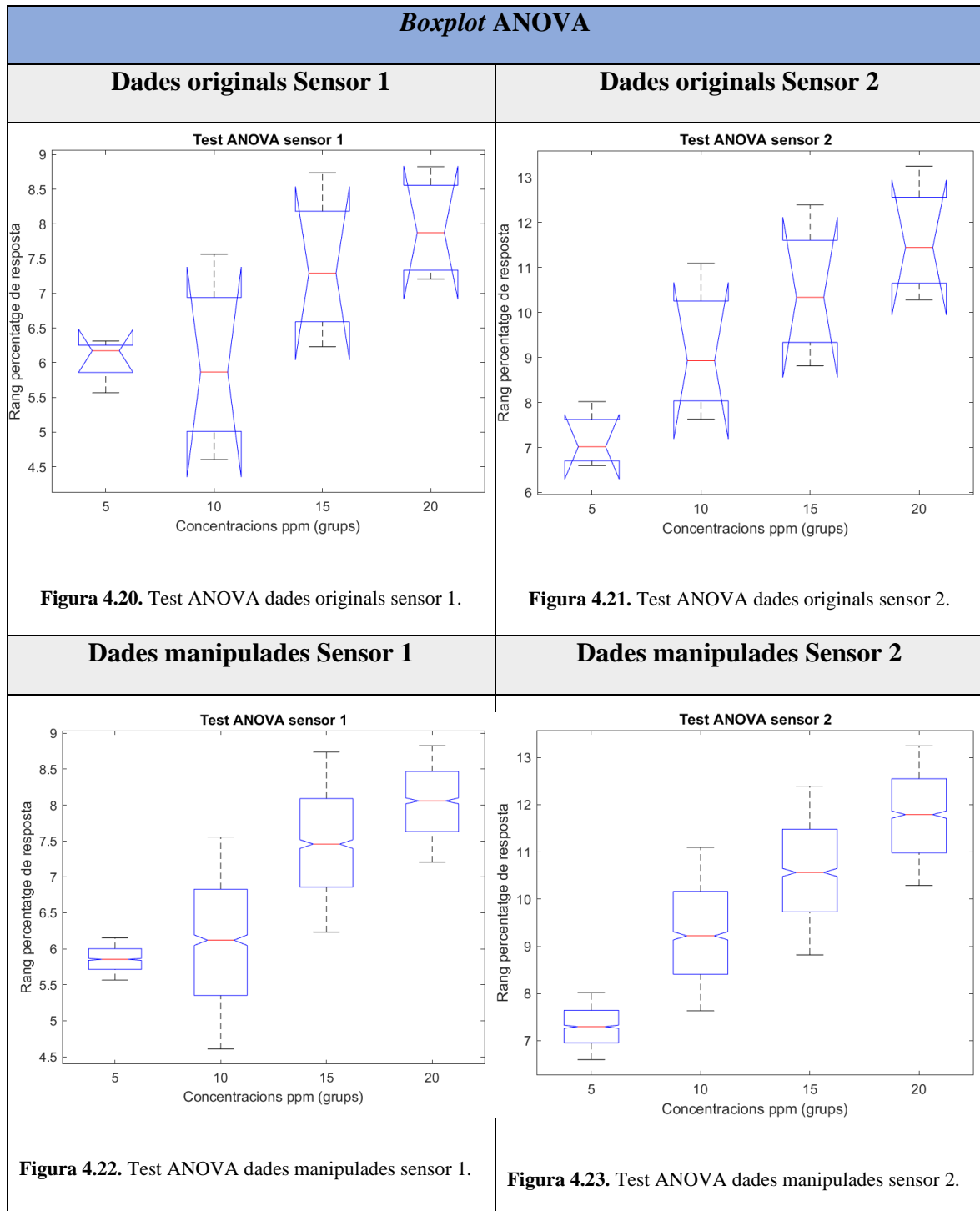
| Valor de $p$                                                                                                                                         |                                                                                                                                                       |
|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Sensor 1</b><br>$P(F \geq 4.5) = 0,0245735$                                                                                                       | <b>Sensor 2</b><br>$P(F \geq 8.9) = 0,002232781$                                                                                                      |
|  <p><b>Figura 4.18.</b> Funció de probabilitat <math>F</math>.</p> |  <p><b>Figura 4.19.</b> Funció de probabilitat <math>F</math>.</p> |

**Taula 4.8.** Valors de  $p$  per a cada sensor i àrees sota la funció de probabilitat  $F$ .

Finalment, analitzant els resultats obtinguts per als valors de  $p$  observats en la *Taula 4.8*, podem concloure que es pot rebutjar en un 97 % i en un 99 % de confiança, aproximadament, la hipòtesi que la mitjana de les respostes és igual per als quatre grups observats, i per tant, es verifica la validesa del test.

Tot seguit, s'han representat mitjançant gràficament els resultats de l'experiment realitzat per ambdós sensors. Degut a que només es tenien quatre mostres per a cada un dels grups, s'ha realitzat una funció en *MATLAB* anomenada *funcioRandomMatrix* que permet generar  $x$  valors aleatoris entre els rangs seleccionats. Així doncs, s'ha aplicat aquesta funció per a generar 1000 valors aleatoris per a cada grup de dades, compresos entre el primer i últim valor de cada grup representat per columnes segons observem en la *Figura 4.16* i *Figura 4.17*.

A posteriori, es pot observar la representació gràfica en *boxplot* dels resultats.

Taula 4.9. Resultats *boxplot* test ANOVA.

Com es pot observar, aplicant la manipulació de les dades comentada prèviament, s'obtenen millors resultats pel que fa a la distinció entre grups, tot millorant l'error. També assenyalar la diferència entre els models, sent l'experiment realitzat per al sensor 2 un millor prototip per a distingir entre un rang de respostes en funció de la concentració de NO<sub>2</sub> detectada.

## 5. TECNOLOGIA NFC

### 5.1. Història i definició de la tecnologia Near Field Communication (NFC).

*Near Field Communication* o NFC, és una tecnologia de comunicació sense fils de curt abast que permet l'intercanvi d'informació entre diferents dispositius. L'associació NFC Forum es va crear amb l'objectiu d'avançar en l'ús de la tecnologia NFC per telèfons mòbils i ordinadors i n'és l'encarregada de promoure i mantenir NFC tot assegurant la completa interoperabilitat entre diferents dispositius i serveis [14].

NFC es defineix, bàsicament, com una tecnologia dins de RFID, que fa ús de la banda freqüencial HF, amb portadora a 13,56 MHz, amb un conjunt específic i adicional de normes per a garantir la interoperabilitat entre els diferents equips habilitats per fer ús de la tecnologia NFC [14]. Mitjançant diferents estàndards, aquests permeten determinar l'entorn operatiu del sistema, el format de les dades que es transmeten, les velocitats de transmissió, que normalment giren entorn a l'ordre de kbits/s, i la modulació utilitzada [14].

Una característica important de la tecnologia NFC és el curt abast de difusió d'informació. Degut a això, els intercanvis de dades entre dispositius NFC no disposen d'alta seguretat en comparació amb altres tecnologies de transmissió de dades sense fils [14]. En el cas en que la transmissió de les dades sigui de caràcter sensible o amb certa susceptibilitat a ser interceptades, es farà ús d'un xifratge per a garantir la seguretat. En referència a la característica del curt abast comentada anteriorment, aquesta el fa ideal per a aplicacions on el requeriment energètic sigui molt baix, i és per això que aquesta tecnologia s'ha pogut integrar en els telèfons mòbils.

### 5.2. Diferències entre RFID i NFC.

Com bé s'ha comentat en el punt anterior, tot i que NFC es basa en un tipus de tecnologia RFID que opera en la banda HF (*High Frequency*) del rang de l'espectre usat per RFID, difereixen en alguns aspectes d'ús, els quals es resumeixen en la següent taula:

| RFID                                                            | NFC                                                       |
|-----------------------------------------------------------------|-----------------------------------------------------------|
| <b>Comparació característiques</b>                              |                                                           |
| Apareix l'any 1980.                                             | Sorgeix l'any 2002.                                       |
| Millora respecte els codis de barres.                           | Millora respecte els codis QR.                            |
| Emmagatzematge i transmissió codis identificadors de productes. | Emmagatzematge i transmissió de diferents tipus de dades. |
| Usat per a la prevenció de la pèrdua de productes.              | Usat àmpliament per a l'autenticació i el pagament.       |
| Rang de lectura fins a centenars de metres.                     | Rang de lectura fins a 10 cm aproximadament.              |

| Comparació aplicacions  |                             |
|-------------------------|-----------------------------|
| Entorns logístics.      | Panys electrònics.          |
| Prevenició de furt.     | Pagaments sense contacte.   |
| Seguiment d'assistents. | Autenticació d'un producte. |

Taula 5.1. Comparació de les característiques i les aplicacions RFID i NFC.

### 5.3. Descripció dels principis físics en que es basa la comunicació NFC.

#### 5.3.1. Acoblament inductiu.

La tecnologia NFC, fa ús de la inducció electromagnètica entre dues bobines de cada una de les antenes integrades en els dispositius NFC, simulant el comportament d'un transformador [9]. En el mode d'operació passiu, el camp magnètic del dispositiu que inicia la comunicació és clau per a que es produeixi l'efecte d'acoblament inductiu entre els dos dispositius, d'aquesta manera, l'esmentat ha de generar una senyal electromagnètica tal que la corrent elèctrica que es generi en el dispositiu receptor, també anomenat passiu, sigui capaç d'alimentar el circuit del mitjà mencionat, tot proporcionar la suficient energia per a generar una resposta per a transmetre les dades que aquest tingui emmagatzemades en la seva memòria [9].

Quan una espiral se situa a prop de l'espiral que emet el camp magnètic, aquesta es veu afectada pel flux magnètic generat. El flux magnètic en una espiral es pot expressar mitjançant la següent equació:

$$\Psi = N \cdot \Phi \quad (5.1)$$

On  $\Phi$ , expressa el flux magnètic d'una determinada àrea, el qual es defineix com la quantitat de línies de camp magnètic que travessen una determinada superfície  $S$ . Per a superfícies planes y camps uniformes, es pot expressar el flux magnètic que travessa una certa àrea, com el producte escalar dels dos vectors  $\vec{B}$  i  $\vec{S}$ , que corresponen al vector de la densitat de flux magnètic i al vector de la superfície, respectivament [10].

$$\Phi = \vec{B} \cdot \vec{S} = |\vec{B}| \cdot |\vec{S}| \cdot \cos(\theta) \quad (5.2)$$

On  $\theta$ , correspon a l'angle entre els dos vectors  $\vec{B}$  i  $\vec{S}$ .

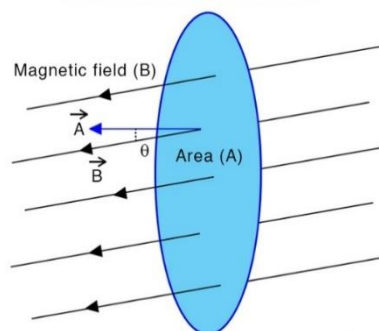


Figura 5.1. Representació del flux magnètic d'una determinada àrea.

Tenint en compte l'expressió anterior, si assumim una orientació, on l'antena que genera el camp magnètic completament exposada al camp de visió d'aquesta, es pot expressar l'equació anterior com [10]:

$$\Phi = \vec{B} \cdot \vec{S} = |\vec{B}| \cdot |\vec{S}| \cdot \cos(90^\circ) = B \cdot S \quad (5.3)$$

On  $S$  correspon a la superfície del material,  $B = H \cdot \mu_r \cdot \mu_0$ , on  $H$  equival al camp magnètic,  $\mu_0 = 4\pi \cdot 10^{-7}$  H/m i  $\mu_r$  expressa la permeabilitat relativa, la qual depèn de les propietats magnètiques del material de l'antena.

Un cop analitzada l'expressió del flux magnètic, es pot reescriure l'equació que representa el flux magnètic en una espiral (11) com:

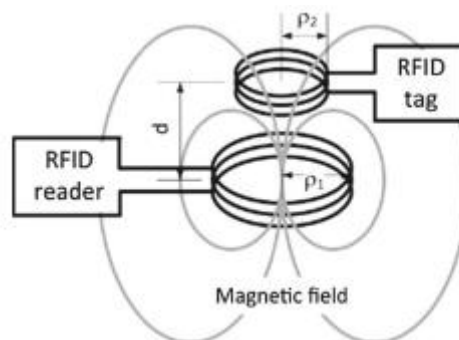
$$\Psi = N \cdot \Phi = N \cdot H \cdot \mu_r \cdot \mu_0 \cdot S \quad (5.4)$$

Per altra banda, la inductància de l'antena espiral es pot expressar com la relació entre el flux magnètic generat per l'espiral i la corrent elèctrica que circula per aquesta [10].

$$L = \frac{\Psi}{I} \quad (5.5)$$

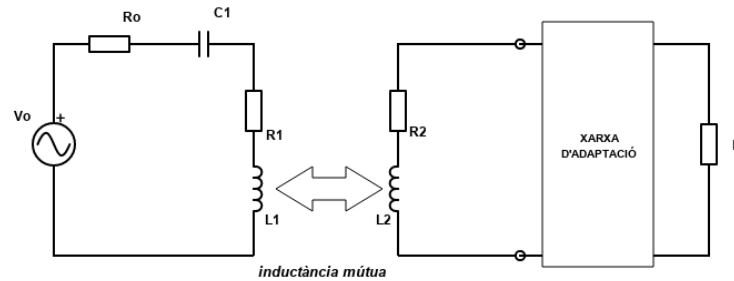
Finalment, l'expressió que permet establir la inductància mútua entre l'espiral corresponent al dispositiu emissor i el corresponent al dispositiu receptor, es pot representar mitjançant la fórmula de Neumann [10]:

$$M_{1,2} = \frac{\mu_0}{4\pi} \cdot \oint_{M1} \oint_{M2} \frac{dI_1 \cdot dI_2}{|r_2 - r_1|} \quad (5.6)$$



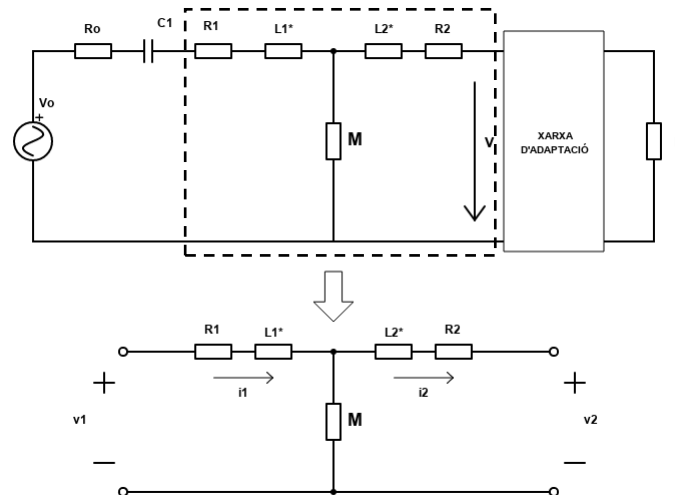
**Figura 5.2.** Representació de l'acoblament inductiu i la inductància mútua.

La determinació de la inductància mútua ens permet calcular el voltatge induït en l'antena del dispositiu secundari [10]. Així doncs, es pot fer una representació mútua dels dos circuits que intervenen en l'acoblament inductiu.



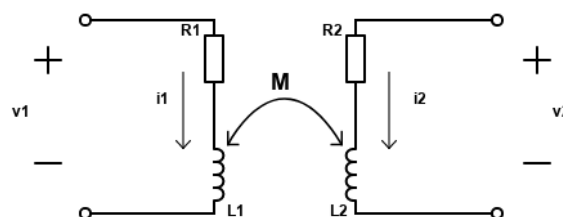
**Figura 5.3.** Representació circuital de la inductància mútua entre dos dispositius NFC.

Així doncs, donat el circuit en la *Figura 5.3*, en podem representar el seu equivalent en "T" de la següent manera:



**Figura 5.4.** Circuit equivalent en "T" de la inductància mútua.

Per a poder caracteritzar el circuit o quadripol en "T" exposat en la *Figura 5.4*, s'ha de tenir en compte un anàlisi previ, per tal de verificar la equivalència entre ambdós circuits representats de forma independent, i del circuit que permet representar la inductància mútua.



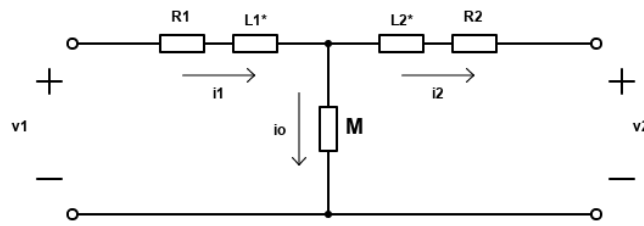
**Figura 5.5.** Representació corrents i voltatge de la inductància mútua.

Considerant el cas en què dos inductors estan mútuament acoblats, es poden representar els voltatges  $v_1$  i  $v_2$  en funció de la corrent que circula per cada un dels inductors, fent ús de les següents expressions:

$$\hat{V}_1 = R_1 \cdot \hat{I}_1 + j\omega L_1 + j\omega M \hat{I}_2 \quad (5.7)$$

$$\hat{V}_2 = R_2 \cdot \hat{I}_2 + j\omega L_2 + j\omega M \hat{I}_1 \quad (5.8)$$

Seguidament, es modela la inductància mútua entre  $L_1$  i  $L_2$  com un únic element, anomenat  $M$ , que caracteritza l'expressió que permet la representació d'aquest nou model. Aquest canvi, comportarà l'aparició de dos nous components anomenats  $L_1^*$  i  $L_2^*$ , els quals permeten conservar l'equivalència d'ambdós circuits.



**Figura 5.6.** Representació del nou model quadripol que representa la inductància mútua.

Per a provar-ne la seva equivalència, es realitzarà un anàlisi d'ambdós models representats en la *Figura 5.5* i la *Figura 5.6*. A partir de les equacions (5.7) i (5.8), podem obtenir les noves expressions per a representar  $\hat{V}_1$  i  $\hat{V}_2$  corresponents a la *Figura 5.6* [11].

$$\hat{V}_1 = R_1 \cdot \hat{I}_1 + j\omega L_1^* + j\omega M \quad (5.9)$$

$$\hat{V}_2 = R_2 \cdot \hat{I}_2 + j\omega L_2^* + j\omega M \quad (5.10)$$

En aquest cas, apareix una nova corrent,  $i_o$ , degut al component equivalent que permet representar la inductància mútua  $M$ . Així doncs, tenint en compte que [11]:

$$\hat{I}_o = \hat{I}_1 + \hat{I}_2 \quad (5.11)$$

$$L_1^* = L_1 - M \quad (5.12)$$

$$L_2^* = L_2 - M \quad (5.13)$$

Es poden reescriure les expressions (5.9) i (5.10) com [11]:

$$\hat{V}_1 = R_1 \cdot \hat{I}_1 + j\omega \hat{I}_1 (L_1 - M) + j\omega M (\hat{I}_1 + \hat{I}_2) \quad (5.14)$$

$$\hat{V}_2 = R_2 \cdot \hat{I}_2 + j\omega \hat{I}_2 (L_2 - M) + j\omega M (\hat{I}_1 + \hat{I}_2) \quad (5.15)$$

Es pot observar com els elements  $-j\omega M \hat{I}_1$  i  $j\omega M \hat{I}_1$  de l'expressió (5.14) es poden cancel·lar, obtenint com a resultat la mateixa expressió presentada en l'equació (5.7). Si es realitza el

procediment anàleg amb la fórmula (5.15), s'obté un resultat igualment equivalent amb l'equació (5.8).

Finalment, podem demostrar que els dos circuits o quadripols representats en la *Figura 5.5* i *Figura 5.6* respectivament, són equivalents.

En atenció a això, es pot representar la matriu de paràmetres  $Z$  del circuit equivalent en "T", de la següent manera, sabent que la matriu d'impedàncies ( $Z$ ) es defineix com  $\bar{V} = [Z] \cdot \bar{I}$ , on  $\bar{V}$  i  $\bar{I}$  corresponen a les tensions i corrents en cada port [12]:

$$\bar{V} = \begin{bmatrix} V_1 \\ V_2 \\ \dots \\ V_N \end{bmatrix}, \bar{I} = \begin{bmatrix} I_1 \\ I_2 \\ \dots \\ I_N \end{bmatrix} \quad (5.16)$$

On la matriu d'impedàncies ( $Z$ ) del circuit "T" equivalent es defineix de la següent manera, tenint en compte que els paràmetres  $Z_{ij}$  s'obtenen deixant en circuit obert tots els ports excepte el  $j$ , i mesurant la tensió que cau en el port  $i$  [12]:

$$Z_{ij} = \frac{V_i}{I_j}, \text{ sent el subíndex de } i \text{ diferent de } j. \quad (5.17)$$

$$[Z] = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} = \begin{bmatrix} R_1 + j\omega L_1 & j\omega M \\ j\omega M & R_2 + j\omega L_2 \end{bmatrix} \quad (5.18)$$

Estant l'anàlisi a la resta del circuit, observem que el paràmetre  $Z_{11}$  correspon a la impedància equivalent del circuit que forma el lector NFC, i per tant, la seva impedància serà la següent:

$$Z_{11} = R_1 + j\omega L_1 + \frac{1}{j\omega C_1} \quad (5.19)$$

De la mateixa manera, la impedància que correspon al circuit "tag NFC" o etiqueta NFC d'on s'obtenen les dades, tindrà com a paràmetre  $Z_{22}$ :

$$Z_{22} = R_2 + j\omega L_2 \quad (5.20)$$

D'altra banda, els paràmetres  $Z_{12}$  i  $Z_{21}$  correspondran a la inductància mútua:

$$Z_{12} = Z_{21} = j\omega M \quad (5.21)$$

Consegüentment, es pot representar novament la matriu de paràmetres  $Z$  de l'expressió (5.18), com:

$$[Z] = \begin{bmatrix} R_1 + j\omega L_1 + \frac{1}{j\omega C_1} & j\omega M \\ j\omega M & R_2 + j\omega L_2 \end{bmatrix} \quad (5.22)$$

Finalment, el voltatge  $V_2$  induït en l'antena de l'etiqueta que conté les dades, es pot adquirir mitjançant el paràmetre  $Z_{21}$ , que representa la impedància observada en el port  $V_1$  quan  $I_2 = 0$  [12]. Tenint en compte l'expressió de l'obtenció dels paràmetres  $Z_{ij}$  exposada en l'equació (17):

$$Z_{21} = j\omega M = \frac{V_2}{I_1}, \text{ sent } I_2 = 0 \quad (5.23)$$

$$V_2 = j\omega M \cdot I_1 \quad (5.24)$$

Finalment, expressant l'equació (5.24) en valor absolut, s'obté la magnitud del voltatge  $V_2$ , que representa la tensió induïda en l'antena del tag NFC [10]:

$$|V_2| = \omega \cdot M \cdot I_1 \quad (5.25)$$

### 5.3.2. Coeficient d'acoblament.

El coeficient d'acoblament, permet donar una explicació qualitativa de la inducció creada entre les dues espiras que intervenen en la inductància mútua [10]. Així doncs, aquest està directament relacionat amb la potència transferida del inductor  $L_1$  al  $L_2$ , gràcies al fenomen d'inductància mútua presentat en l'apartat anterior.

D'altra banda, aquest depèn d'una sèrie de paràmetres físics relacionats amb els conductors i el medi [10]:

- La dimensió de les espiras.
- La posició a la que estan enfrontades ambdues espiras.
- Propietats magnètiques del medi, expressades segons la constant  $\mu_0$ .

Així doncs, degut a la dependència presentada, el coeficient d'acoblament  $k$  es pot expressar únicament en funció de les característiques anteriors com [10]:

$$k = \mu_0 \cdot \frac{r^2}{2\sqrt{(r^2+x^2)^3}} \cdot \frac{A_2}{L_1 \cdot L_2} \quad (5.26)$$

On  $\mu_0 = 4\pi \cdot 10^{-7}$  H/m i correspon a les propietats magnètiques del medi,  $r$  fa referència al radi de l'antena expressat en cm,  $x$  correspon a la distància entre el lector i el dispositiu llegit, i  $A_2$  representa l'àrea del dispositiu llegit.

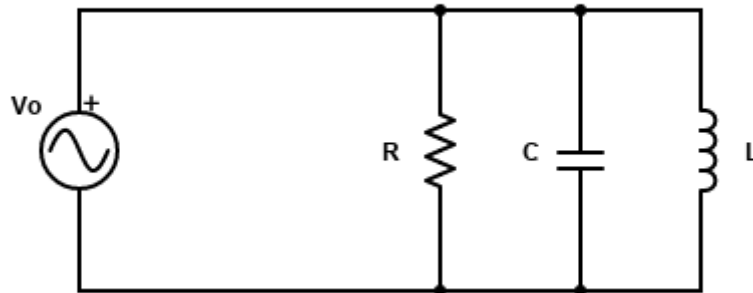
$$k = \frac{M}{\sqrt{L_1 \cdot L_2}} \quad (5.27)$$

Aquest coeficient comprèn valors  $0 < k < 1$ , permetent la discriminació del nivell d'acoblament del sistema. D'aquesta manera, valors de  $k$  iguals o propers a 0, representen que no hi ha acoblament degut a una gran separació entre els elements inductius.

D'altra banda, per a valors de  $k$  pròxims o iguals a 1, indiquen un bon acoblament, o un acoblament total d'ambdós elements inductius.

### 5.3.3. Freqüència de ressonància.

La freqüència de ressonància de l'antena, es pot obtenir mitjançant el següent procediment:



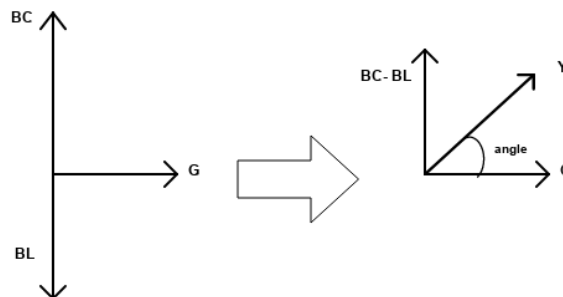
**Figura 5.7.** Exemple circuit ressonant RLC

Observant la *Figura 5.7*, que correspon a un circuit RLC, en podem obtenir la impedància equivalent  $Z_{eq}$  com:

$$\frac{1}{Z_{eq}} = \frac{1}{R + j\omega L + \frac{1}{j\omega C}} \quad (5.28)$$

$$\frac{1}{Z_{eq}} = \frac{1}{R} + j\omega C + \frac{1}{j\omega L} \quad (5.29)$$

Sabent que  $\frac{1}{Z_{eq}}$  correspon a la admitància equivalent, expressada com  $Y_{eq}$ ,  $\frac{1}{R}$  correspon a la conductància  $G$  i que el diagrama de fasors del conjunt del circuit el podem representar de la següent manera, es pot reescriure l'equació (4.2) com:



**Figura 5.8.** Diagrama de fasors de l'admitància d'un circuit RLC.

$$Y_{eq} = G + j\left(\omega C - \frac{1}{\omega L}\right) = G + j(B_C - B_L) = \sqrt{(G)^2 + (B_C - B_L)^2} \quad (5.30)$$

On  $B_C$  i  $B_L$  corresponen a les susceptàncies capacitiva i inductiva respectivament.

Observant el factor  $j(\omega C - \frac{1}{\omega L})$ , i sabent que  $\omega = 2\pi f$ , podem obtenir la freqüència d'operació del circuit sabent que la freqüència de ressonància, correspon a aquella freqüència on els efectes capacitius i inductius del circuit tenen el mateix valor absolut i, per tant, el circuit es comporta com a purament resistiu [13] i presenta una impedància d'entrada  $Z_{in} = R$ , podent expressar la relació de susceptàncies de la següent manera:

$$B_C - B_L = 0 \quad (5.31)$$

$$B_C = B_L \quad (5.32)$$

$$2\pi f C = \frac{1}{2\pi f L} \quad (5.33)$$

Finalment, obtenim l'expressió de la freqüència de ressonància  $f_R$  que caracteritzarà al circuit LC ressonant:

$$f_R = \frac{1}{2\pi\sqrt{LC}} \quad (5.34)$$

#### 5.4. Protocols i estàndards.

Els principals estàndards que implementa la tecnologia NFC són ISO/IEC 14443, ISO/IEC 15693, ISO/IEC 18092, ECMA-340 i ECMA-352 [14]. Els estàndards que corresponen a ISO/IEC 14443 i 15693, es van desenvolupar inicialment per estandarditzar la tecnologia RFID. Posteriorment, es van adoptar també per la tecnologia NFC. Així doncs, els estàndards propis de la tecnologia NFC són ISO/IEC 18092 i ECMA-340 i 352 [14].

Cada un dels estàndards mencionats inclou diferents parts en funció de les implementacions i característiques dels dispositius que integren la tecnologia NFC. A continuació, es podran observar un esquema a mode de representació del conjunt d'estàndards de NFC, tot distingint aquells desenvolupats en un principi per RFID d'aquells desenvolupats exclusivament per NFC [14]:

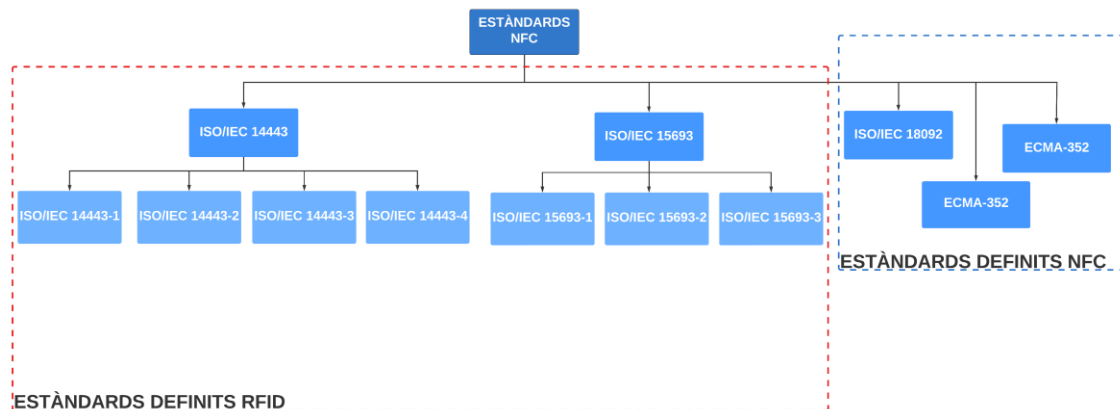


Figura 5.9. Esquema del conjunt d'estàndards NFC.

A continuació, observem una taula resum dels estàndards ISO/IEC ja existents abans de NFC [14]:

| Estàndard            | Parts           | Principals implementacions                                             |
|----------------------|-----------------|------------------------------------------------------------------------|
| <b>ISO/IEC 14443</b> | ISO/IEC 14443-1 | Descripció de les característiques físiques dels <i>tags</i> .         |
|                      | ISO/IEC 14443-2 | Potència i característiques del senyal de RF.                          |
|                      | ISO/IEC 14443-3 | Sistemes d'inicialització i prevenció de col·lisió.                    |
|                      | ISO/IEC 14443-4 | Protocol de transmissió <i>half-dúplex</i> .                           |
| <b>ISO/IEC 15693</b> | ISO/IEC 15693-1 | Característiques físiques de les <i>vicinity cards</i> .               |
|                      | ISO/IEC 15693-2 | Naturalesa i característiques de la comunicació.                       |
|                      | ISO/IEC 15693-3 | Inicialització i prevenció de col·lisió en les <i>vicinity cards</i> . |

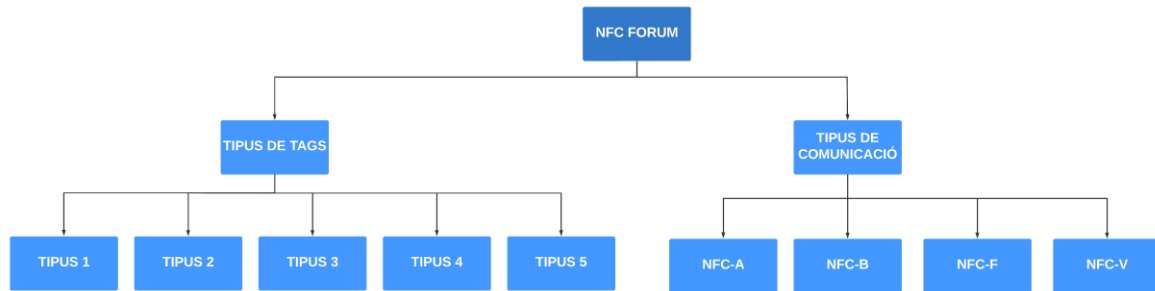
**Taula 5.2.** Principals estàndards ISO/IEC i les seves implementacions.

En referència als estàndards ISO/IEC 18092, ECMA-340 i ECMA-352, específics per a la tecnologia NFC, podem observar-ne les principals implementacions en la *Taula 5.3* [14]:

| Estàndards                                                 | Principals implementacions                                                       |
|------------------------------------------------------------|----------------------------------------------------------------------------------|
| <b>ISO/IEC 18092</b><br><b>ECMA-340</b><br><b>ECMA-352</b> | Definició dels diferents esquemes de modulació a utilitzar.                      |
|                                                            | Explicació de la codificació per a la transmissió de les dades.                  |
|                                                            | Especificació dels formats de la trama de la interfície de RF.                   |
|                                                            | Descripció de la taxa de dades usades en la transmissió.                         |
|                                                            | Caracterització de la inicialització de la comunicació i prevenció de col·lisió. |
|                                                            | Determinació dels protocols d'activació i intercanvi d'informació.               |
|                                                            | Explicació dels modes de comunicació actius i passius utilitzats.                |

**Taula 5.3.** Principals implementacions dels estàndards específics per a NFC.

NFC Forum també va desenvolupar especificacions per tal globalitzar la compatibilitat entre quatre tipus de comunicació diferents, fent ús d'una terminologia específica per a distingir-les entre elles [14]. A de més, es suma l'objectiu d'internacionalitzar i flexibilitzar l'ús de la tecnologia NFC, per a que no sigui només limitada a targetes i lectors NFC, fent possible la integració d'aquesta tecnologia en dispositius mòbils, etiquetes i altres aparells portàtils [14].



**Figura 5.10.** Especificacions desenvolupades per NFC Forum.

Tot seguit, podem assenyalar els diferents *tags* NFC especificats per NFC Forum, i que implementen els estàndards explicats anteriorment. D'aquesta manera, s'observaran les característiques estandarditzades de cada un dels dispositius passius de manera qualitativa [14].

| NFC            | Estàndard        | Memòria           | Velocitat                               | Comunicació        |
|----------------|------------------|-------------------|-----------------------------------------|--------------------|
| <b>Tipus 1</b> | ISO/IEC 14443A   | 96 bytes-2 kbytes | 106 kbits/s                             | Lectura/Escriptura |
| <b>Tipus 2</b> | ISO/IEC 14443A   | 48 bytes-2 kbytes | 106 kbits/s                             | Lectura/Escriptura |
| <b>Tipus 3</b> | ISO/IEC 18092    | 2 kbytes          | 212 kbits/s o 424 kbits/s               | Lectura/Escriptura |
| <b>Tipus 4</b> | ISO/IEC 14443A-B | 32 kbytes         | 106 kbits/s o 212 kbits/s o 424 kbits/s | Lectura/Escriptura |
| <b>Tipus 5</b> | ISO/IEC 15693    | 64 kbytes         | 26,48 kbits                             | Lectura/Escriptura |

**Taula 5.4.** Característiques dels principals *tags* NFC.

D'altra banda, es mostrarà a continuació una taula resum de les especificacions implementades per NFC Forum, tal i com s'ha comentat, per tal d'identificar els quatre tipus de comunicació diferents utilitzades per a la transmissió de dades entre el dispositiu actiu i passiu [14].

| Transmissor | Característica       | NFC-A                          | NFC-B                             | NFC-F                          | NFC-V                                                                     |
|-------------|----------------------|--------------------------------|-----------------------------------|--------------------------------|---------------------------------------------------------------------------|
| Dispositius | <b>Freqüència</b>    | 13,56 MHz                      | 13,56 MHz                         | 13,56 MHz                      | 13,56 MHz                                                                 |
|             | <b>Modulació</b>     | 100% ASK                       | 10% ASK                           | 10% ASK                        | 10% o 100% ASK                                                            |
|             | <b>Codificació</b>   | Modified Miller                | NRZ                               | Manchester                     | PPM                                                                       |
|             | <b>Velocitat</b>     | 106 kbits/s fins a 424 kbits/s | 106 kbits/s fins a 424 kbits/s    | 212 kbits/s fins a 424 kbits/s | 26,48 o 1,65 kbits/s                                                      |
| Targetes    | <b>Modulació</b>     | <i>ASK load modulation</i>     | <i>ASK load modulation o BPSK</i> | <i>ASK load modulation</i>     | <i>ASK load modulation o FSK</i>                                          |
|             | <b>Sub-portadora</b> | 848 kHz                        | 848 kHz                           | -                              | 424/484 kHz                                                               |
|             | <b>Codificació</b>   | Manchester                     | NRZ                               | Manchester                     | Manchester                                                                |
|             | <b>Velocitat</b>     | 106 kbits/s fins a 424 kbits/s | 106 kbits/s fins a 424 kbits/s    | 212 kbits/s fins a 424 kbits/s | <b>100% ASK:</b> 6,62 o 26,48 kbits/s<br><b>FSK:</b> 6,67 o 26,69 kbits/s |

Taula 5.5. Diferents tipus de comunicació especificats per NFC Forum.

Així doncs, NFC-A i NFC-B es refereixen a la comunicació tipus A i B especificada en l'estàndard ISO/IEC 14443. Pel que fa a NFC-F, correspon a la tecnologia FeliCa, la qual va ser la primera a implementar-se i estandaritzada per IOS/IEC 15408 ELA [14]. Finalment, NFC-V fa referència al conjunt d'implementacions corresponents a l'estàndard ISO/IEC 15693 [14].

En referència als protocols NFC, els principals més importants que intervenen en la comunicació són:

- LLCP o *Logical Link Protocol*.
- NDEF o *NFC Data Exchange Format*.

Seguidament, es farà una breu descripció dels dos protocols mencionats anteriorment. En l'apartat 5.6 de la memòria, s'analitzarà amb més detall el protocol usat per a l'intercanvi de dades, conegut com a NDEF.

### **LLCP o *Logical Link Protocol***

Aquest protocol permet la implementació de comunicació *peer-to-peer* entre dos dispositius NFC, essencial per a aplicacions que fan ús d'aquest tipus de comunicació i, per tant, l'intercanvi de dades es realitza *half-dúplex*, o de manera bidireccional [15].

L'especificació d'aquest protocol, defineix tres tipus de serveis per a oferir comunicació als seus usuaris, dos dels quals estan basats en serveis orientats a la connexió i serveis no orientats a la connexió, mentre que el darrer fa ús d'ambdós serveis a la vegada [15]:

- No orientat a la connexió. Permet oferir comunicació de la forma més senzilla però sense garanties pel que fa a l'establiment i la fiabilitat de la comunicació entre ambdós dispositius [15].
- Orientat a la connexió. És capaç d'oferir control de flux de dades i fiabilitat en la comunicació per tal de proporcionar un servei de confiança per a l'intercanvi de dades [15], el qual és necessari en algunes de les aplicacions que usen la funcionalitat *peer-to-peer*.
- Orientat i no orientat a la connexió. Aprofita els avantatges i els desavantatges dels dos serveis anteriors per a diferenciar, segons el tipus d'aplicació, el servei per a proporcionar comunicació segons les característiques i la sensibilitat de les dades a intercanviar [15].

### **NDEF o *NFC Data Exchange Format***

El protocol NDEF permet l'intercanvi de dades entre qualsevol dispositiu NFC o *tag*, independentment de la tecnologia NFC utilitzada o del tipus d'etiqueta. El missatge NDEF que es transmet, està format per un conjunt de NDEF *records*, que es concatenen per a formar el conjunt del missatge. La mida màxima de cada un dels NDEF *records* es limita a  $2^{32} - 1$  bytes [14].

En l'apartat 5.6, es mostra una descripció detallada de l'arquitectura del missatge NDEF, així com els paràmetres més importants.

#### **5.5. Funcionament.**

Referent al funcionament de la tecnologia NFC, podem destacar-ne els modes de d'operació, els modes de comunicació i la modulació que utilitza per a la correcta transmissió i recepció de dades. A més, també cal tenir en compte el tipus de seguretat que implementa aquesta tecnologia, ja que en funció del tipus d'aplicació, es requerirà de major o menor seguretat en la comunicació.

### 5.5.1. Mode actiu.

En el mode de comunicació actiu, ambdós dispositius participants en l'intercanvi de comunicació generen un camp electromagnètic, transmetent les dades fent ús de la modulació ASK [14]. En avantatge sobre el mode passiu, en el mode actiu les comunicacions entre dispositius es poden realitzar a més distància, tot dependrà però del protocol que aquests implementin.

Per altra banda, fent ús de la modulació PSK, el dispositiu objectiu (aquell que no inicia la comunicació), utilitza la tècnica de *load modulation* [14] per a transmetre les dades demanades per l'aparell iniciador de la comunicació. La utilització d'aquest tipus de modulació, s'arriben a velocitats de transmissió de dades de fins a 6,78 Mbit/s [14].

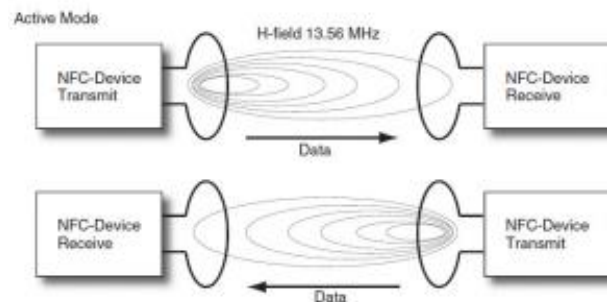


Figura 5.9. Representació de la comunicació en mode actiu.

### 5.5.2. Mode passiu.

Contràriament al mode actiu, en el mode passiu el camp electromagnètic que permet la inductància mútua i l'establiment de la connexió, només es generat per un dispositiu. Aquest, per tant, rep el nom de dispositiu actiu o iniciador, ja que serà l'únic que permetrà l'establiment de la connexió, mitjançant la seva interacció. Contràriament, l'aparell que no genera el camp electromagnètic és l'anomenat dispositiu objectiu o dispositiu a identificar [14].

El dispositiu actiu realitza la modulació de la portadora del senyal que transmet, permetent així la transmissió de dades cap al dispositiu passiu, on s'interpreta com a dades [14]. D'altra banda, aquest últim, transmet les dades cap a l'aparell iniciador tornant a modular el senyal rebut. Un cop aquest nou senyal arriba al dispositiu iniciador, aquest es detecta igualment com a dades [14].

A continuació, es pot observar una imatge que representa l'intercanvi d'informació entre els dos dispositius, tal i com s'ha explicat anteriorment.

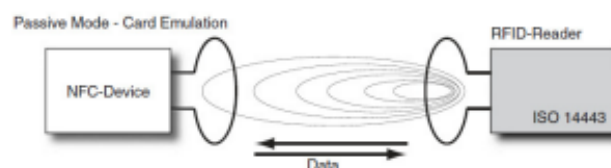


Figura 5.10. Representació de la comunicació en mode passiu.

### 5.5.3. Modes de comunicació.

A continuació, en la *Taula 5.1*, es poden observar els diferents modes de comunicació entre dispositius NFC, reconeguts per NFC Forum. Es detallen, també, les aplicacions més típiques per a cada un dels modes presentats [14].

| Mode de comunicació  | Descripció                                                                                                                    | Mode d'operació | Aplicació                                                                          |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------|-----------------|------------------------------------------------------------------------------------|
| Lectura/Escriptura.  | Permet la lectura o escriptura de dades d'un objecte NFC.                                                                     | Mode passiu.    | Obtenir informació d'un determinat objecte.                                        |
| Emulació de targeta. | El dispositiu NFC es comporta com una targeta intel·ligent.                                                                   | Mode passiu.    | Pagaments sense contacte i control d'accés.                                        |
| <i>Peer-to-Peer.</i> | Possibilita la comunicació entre dos dispositius NFC actius, permetent major velocitat de transmissió i major volum de dades. | Mode actiu.     | Intercanvi de paràmetres de configuració, targetes de contacte digitals o imatges. |

**Taula 5.6.** Resum dels modes de comunicació entre dispositius NFC.

### 5.5.4. Modulació i codificació del senyal.

El procés de modulació consisteix en la modificació dels paràmetres d'una ona sinusoidal, anomenada portadora, a través d'una altra senyal anomenada moduladora, que és la que conté la informació que es vol transmetre [16].

En el cas de l'estudi de la tecnologia NFC, aquesta utilitza una modulació digital pas banda, ja que permet el desplaçament del senyal que es vol transmetre a la banda freqüencial desitjada, en aquest cas la portadora està situada a la freqüència de 13,56 MHz.

L'expressió del senyal segueix la següent equació [16]:

$$s(t) = A(t)\cos [\omega_c t + \theta(t)] \quad (5.35)$$

A partir de l'explicació anterior, podem definir les principals modulacions utilitzades en la tecnologia NFC estudiada. La modulació utilitzada, depèn principalment del tipus de *tag* emprat. Aquestes són [17]:

- ASK o *Amplitude Shift Keying*.
- FSK o *Frequency Shift Keying*.
- PSK o *Phase Shift Keying*.

Seguidament, es detallaran les característiques bàsiques d'aquest conjunt de modulacions digitals juntament amb una representació gràfica de cada una.

Per a la representació gràfica que s'usarà com a exemple per a cada una de les modulacions i codificacions, es parteix de la següent seqüència de bits:

$$seq = [01100101] \quad (5.36)$$

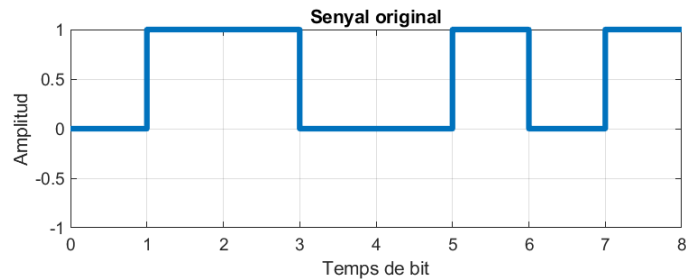


Figura 5.11. Exemple de seqüència de bits a codificar.

### ASK o Amplitude Shift Keying.

La modulació ASK, permet la representació binària del senyal en funció de les variacions d'amplitud del senyal modulad [16]. Quan el senyal binari original correspon a un valor de '0', la portadora de la senyal modulada adopta un valor ideal de 0, mentre que un valor binari de '1' correspon a un determinat valor d'amplitud de la senyal modulada. Aquesta casuística representa la forma més senzilla de modulació ASK, també anomenada *on-off keying*, ASK binari o 100% ASK.

A continuació, podem observar un exemple de modulació 100% ASK en la Figura 5.12.

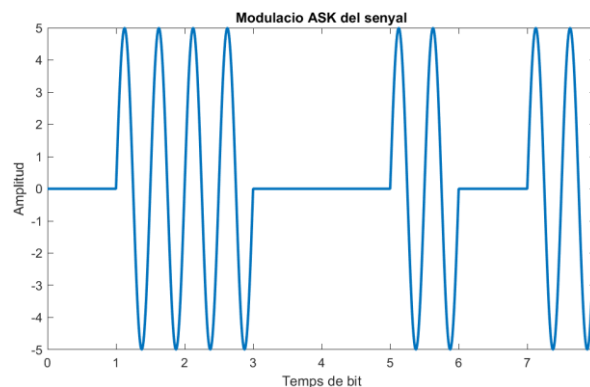


Figura 5.12. Senyal digital modulad en amplitud.

Lligant amb l'explicació anterior, la raó per la qual la modulació ASK binària és també anomenada 100% ASK, és degut al decrement que experimenta l'amplitud de la moduladora en la transmissió de '0' lògics en comparació amb el nivell d'amplitud en la transmissió de '1'. Així doncs, el percentatge és el factor que permet identificar aquest decrement. Observem doncs, que un decrement del 100% fa referència a una amplitud de valor 0 per a la transmissió lògica de '0'.

Fins ara s'ha analitzat la modulació ASK amb dos nivells de la portadora, però també existeixen modulacions ASK amb  $M$  nivells. L'expressió analítica de la modulació ASK és la següent [16]:

$$s_i(t) = \sqrt{\frac{2E_i(t)}{T}} \cos [\omega_c t + \theta] \quad (5.37)$$

On el terme que caracteritza l'amplitud pot prendre valors discrets des de 1 fins a  $M$ , i  $T$  correspon al temps de bit [16]. Quan  $M$  pren el valor  $M = 2$ , ens trobem amb el cas de ASK binari comentat anteriorment.

Les modulacions ASK amb  $M$  nivells no són gaire implementades en la pràctica, degut a que augmenta la probabilitat d'error de bit a mesura que van augmentant els diferents estats d'amplitud [17].

### **FSK o *Frequency Shift Keying***

La modulació FSK, es basa en la variació de la freqüència de la portadora en funció de la seqüència de la senyal digital [16]. La representació d'un '1' lògic es duu a terme mitjançant un augment de la freqüència de la senyal portadora, mentre que un '0' lògic és representat amb una freqüència menor respecte al nivell alt del senyal binari [16][17].

En la *Figura 5.13* podem observar un exemple de modulació FSK, a partir de la seqüència digital representada en la *Figura 5.11*.

L'expressió que permet la representació de la modulació FSK analíticament és [16]:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos [\omega_i t + \theta] \quad (5.38)$$

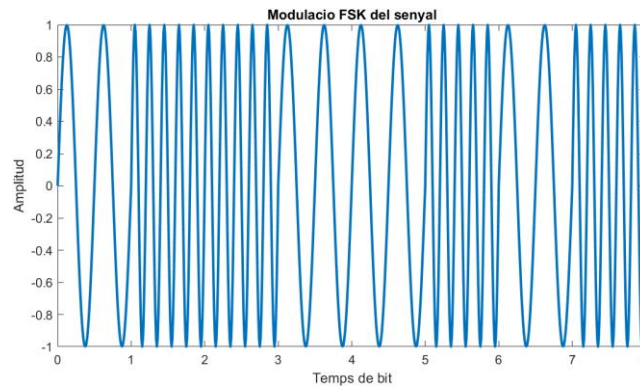
On  $i$  pot prendre valors discrets des de 1 fins a  $M$ , i  $T$  correspon al temps de bit. Quan  $M$  pren el valor  $M = 2$ , ens trobem amb el cas de BFSK [16]. En aquest cas, l'amplitud i la fase de la senyal romanen constants mentre que la freqüència  $\omega_i(t)$  podrà prendre  $M$  valors discrets.

Així doncs, el cas implementat a la pràctica i l'usat en la representació d'aquest exemple, és la modulació BFSK, és a dir, quan  $i = 1,2$  i  $M$  pren el valor  $M = 2$ .

Consegüentment, l'expressió que caracteritza la modulació BFSK és [16]:

$$S(t) = \begin{cases} \sqrt{\frac{2E}{T}} \cos[\omega_1 t + \theta] & \text{quan 1 binari} \\ \sqrt{\frac{2E}{T}} \cos[\omega_2 t + \theta] & \text{quan 0 binari} \end{cases}, \text{ on } \omega_1 > \omega_2 \quad (5.39)$$

Podem veure una representació gràfica de la modulació presentada en la *Figura 5.13*, segons l'expressió analítica descrita en (5.50).

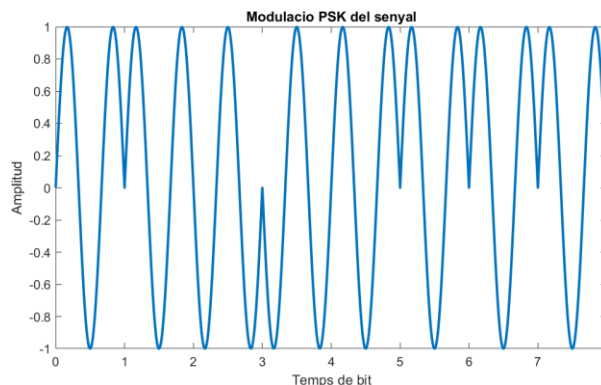


**Figura 5.13.** Senyal digital modulats en freqüència.

### PSK o Phase Shift Keying

La modulació PSK, es basa en la variació de la fase de la portadora amb la finalitat de representar cada símbol de la senyal moduladora, o senyal binari original que conté la informació a transmetre, mitjançant un valor angular que permetrà distingir la transmissió de '0' o '1' [16][17].

En la *Figura 5.14* podem observar un exemple de modulació PSK a partir de la senyal moduladora de la *Figura 5.11*, que correspon a la senyal que conté la informació que volem transmetre.



**Figura 5.14.** Senyal digital modulats en fase.

A continuació, es presenta l'expressió analítica que caracteritza a la modulació PSK [16]:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos [\omega_c t + \theta(t)] \quad (5.40)$$

On  $i$  pot prendre valors discrets des de 1 fins a  $M$ , i  $T$  correspon al temps de bit. Quan  $M$  pren el valor  $M = 2$ , ens trobem amb el cas de BPSK, utilitzada en la tecnologia NFC.

Observem que a diferència de la modulació ASK, on l'amplitud és funció del temps  $i$ , per tant, adopta diferents valors segons aquest, mentre la fase és constant, en la modulació PSK

l'argument  $\theta(t)$  és el que varia en funció del temps i l'amplitud és el terme que es manté constant.

Així doncs, l'argument  $\theta(t)$  pot prendre diferents valors discrets  $M$  múltiples de  $2\pi$  [16]:

$$\theta(t) = \frac{2\pi(i-1)}{M} \quad (5.41)$$

El tipus de modulació PSK que s'utilitza en la tecnologia NFC és la modulació binària PSK, també coneguda com a modulació BPSK. Aquesta, disposa de dues fases de sortida [16], i per tant, una representa un '1' i l'altre representa un '0'. A partir de l'equació (5.49), podem determinar la variació entre les dues fases, sabent que  $M = 2$  i  $i = 1,2$ :

$$\theta_1(t) = \frac{2\pi(1-1)}{2} = 0^\circ \quad (5.42)$$

$$\theta_2(t) = \frac{2\pi(2-1)}{2} = \pi \text{ rad} = 180^\circ \quad (5.43)$$

Així doncs, els dos angles que caracteritzen el '0' i el '1' estan desfasats  $180^\circ$ , com es pot apreciar en la representació en *MATLAB* de la *Figura 5.14*.

La codificació permet la traducció dels valors de tensió, per tal de donar lloc a senyals digitals que podran ser transmeses a través del medi fent ús de la modulació [16].

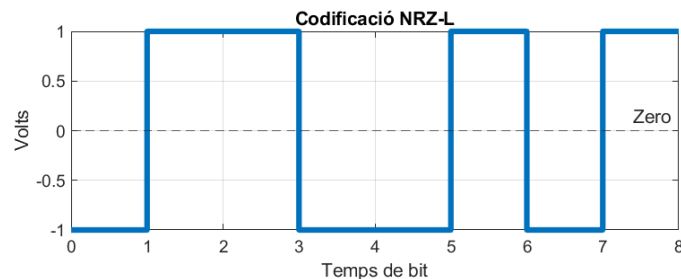
Les principals codificacions que fa servir la tecnologia NFC són [17]:

- NRZ-L
- *Modified Miller*
- *Manchester Coding*

Seguidament, es detallaran les característiques principals de les codificacions de senyal utilitzades en NFC, així com un exemple per a cada cas.

### NRZ-L

En la codificació NRZ-L, un nivell de voltatge positiu s'interpreta com un '1' lògic, mentre que un nivell de voltatge negatiu s'interpreta com un '0' lògic [17]. Aquest tipus de codificació es fa servir per a la posterior modulació en freqüència, FSK, o en fase, PSK [17].

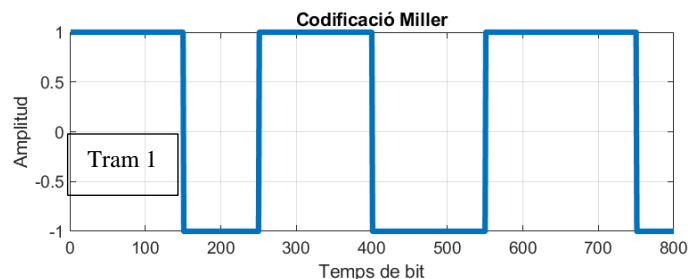


**Figura 5.15.** Representació de la codificació NRZ-L en *MATLAB*.

En la *Figura 5.15*, podem observar una representació gràfica de la codificació NRZ-L, on el valor d'amplitud 1, correspondria a un voltatge positiu  $V$ , i el valor d'amplitud -1 a un valor de voltatge negatiu  $-V$ , tal i com s'ha comentat.

### **Modified Miller**

En la codificació *Miller*, un '1' lògic es codifica com un nivell baix de tensió en la segona meitat del temps de bit. Contràriament, un '0' lògic es codifica com un nivell baix de tensió en la primera meitat del temps de bit [17]. Ara bé, si en la trama de bits a codificar, ens trobem la següent seqüència ...01..., és a dir, un '0' i tot seguit un '1', es codifica com a nivell alt de tensió, i per tant no hi ha cap transició [17].

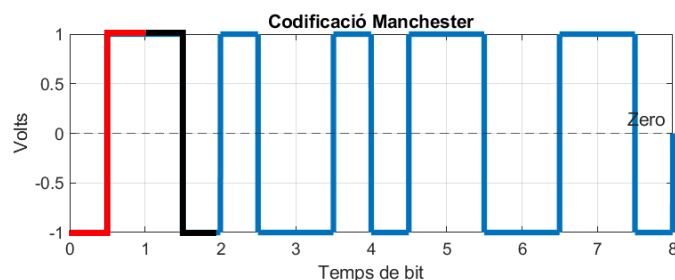


**Figura 5.16.** Representació de la codificació *Miller* en *MATLAB*.

En la *Figura 5.16*, podem observar un exemple de l'estat de codificació mencionat anteriorment, referent a la seqüència ...01... Així doncs, donat el tram de bits (5.58), observem com el *Tram 1* està codificat tal i com s'ha descrit.

### **Manchester Coding**

En la codificació *Manchester*, el temps corresponent a la transmissió d'un bit es divideix en dues parts [17]. En la transmissió d'un valor lògic de '1', la primera meitat del temps de bit correspon a un nivell alt de tensió, mentre que l'altre meitat correspon a un valor baix de tensió. D'altra banda, un '0' lògic es correspondrà a un valor baix de tensió en la primera meitat del temps de bit, seguit d'un valor alt de tensió en la segona meitat del temps [17].



**Figura 5.17.** Representació de la codificació *Manchester* en *MATLAB*.

En la *Figura 5.17*, podem parlar esment al exemple de codificació per tal de poder comprendre el procediment fent ús d'un exemple pràctic.

Tenint en compte la seqüència de bits inicial a codificar (5.58), observem com el bit més significatiu correspon a un 0, seguit d'un 1. Segons l'explicació anterior, s'aprecia com el

temps de bit inicial és dividit en dos trams. Així doncs, el temps de bit comprès entre el 0 i l'1, correspon a la codificació d'un '0' lògic, i tal com s'ha comentat anteriorment, es pot veure com el primer tram de temps de bit correspon a un valor baix de tensió, mentre que el segon tram correspon a un valor alt de tensió (tram subratllat en vermell). Observem el procediment invers per a la codificació d'un '1' lògic (tram subratllat en negre).

## 5.6. NDEF.

Tal i com s'ha comentat en l'apartat 5.4, referent als estàndards i protocols usats en la tecnologia NFC, el protocol NDEF permet l'intercanvi d'informació entre dispositius NFC compatibles o *tags*. L'estructura i format d'aquest protocol és el següent [14]:

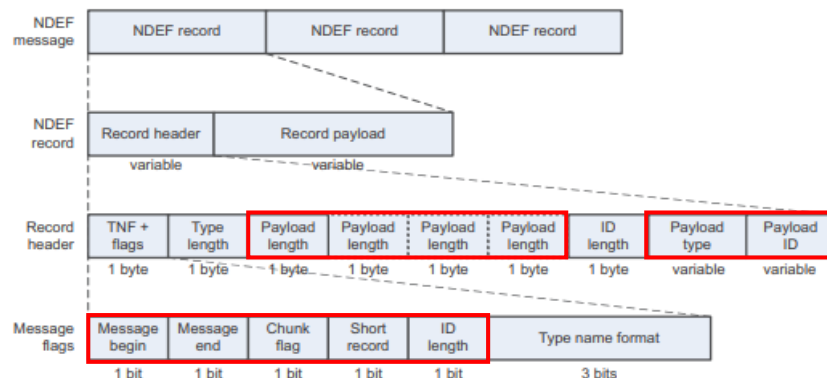


Figura 5.18. Estructura del registre NDEF.

Les principals dades que es poden transmetre en els NDEF *records* són [14]:

- Text
- Identificador URI
- Signatura digital
- Informació de targetes electròniques

L'estructura que segueix el NDEF *record* està formada per un *Record Payload*, que conté el conjunt de dades a transmetre, i un *Record Header*, que descriu com s'interpreten aquestes dades. Aquest utilitza, principalment, els següent camps per a descriure la informació que es transmet [10,14,15]:

- TNF (Type Name Format) + flags. Aquest camp, format per 1 byte d'informació, conté 8 bits que permeten la identificació i la interpretació de les dades que conté el camp *payload type*. Els *flags* ocuparan 5 bits del registre, mentre que el TNF ocuparà 3 bits, obtenint un total d'1 byte d'ocupació [10,14].

Els diferents *flags* continguts en el primer camp del registre NDEF són els següents [10,14]:

| <b>Flags del primer camp de la capçalera NDEF</b> |                                                                     |
|---------------------------------------------------|---------------------------------------------------------------------|
| <i>Message Begin</i> (MB)                         | Permet indicar l'inici d'un missatge NDEF (MB = 1).                 |
| <i>Message End</i> (ME)                           | Indica el final d'un missatge NDEF (ME = 1).                        |
| <i>Chunk Flag</i> (CF)                            | Indica que només hi ha un <i>payload</i> en el missatge (CF = 0).   |
| <i>Short Record</i> (SR)                          | Anuncia que el <i>payload length</i> només ocuparà 1 byte (SR = 1). |
| <i>ID length</i> (IL)                             | Valor booleà que permet expressar si el camp és present o no.       |

Taula 5.2. *Flags* de la capçalera NDEF.

Pel que fa al valor de 3 bits TNF, aquest permetrà identificar l'estructura del *payload type*. Amb 3 bits, podem representar fins a 8 possibles identificadors, que es mostren descrits en la Taula 5.3 [10].

| <b>TNF</b>                       | <b>Identificador</b> |
|----------------------------------|----------------------|
| No usat                          | 0x00                 |
| NFC RTD                          | 0x01                 |
| MIME (Contingut Media)           | 0x02                 |
| Identificador URI                | 0x03                 |
| Extern, definit per l'usuari     | 0x04                 |
| Desconegut                       | 0x05                 |
| Identifica varis <i>payloads</i> | 0x06                 |
| Reservat                         | 0x07                 |

Taula 5.3. Valors de TNF.

- *Type length*. Permet especificar la longitud del camp *payload type*, en octets. Per a valors 0x05 i 0x06 del camp TNF, el *type length* és zero [10].
- *Payload length*. Consta de 4 camps d'1 byte cadascun, que permeten indicar el nombre de bytes que ocupa la informació a transmetre, continguda en el *payload*. La mida d'aquest està condicionat per el valor del *flag* SR [10].
- *Payload type*. Identifica el tipus de missatge contingut en el camp *payload*. El tipus de missatge a transmetre, dependrà de les necessitats de cada aplicació. La

informació que es pot emmagatzemar en el camp *payload*, ha sigut descrita prèviament [10].

- *Payload Identifier*. Aquest identificador és opcional, i permet, a les aplicacions, identificar el tipus de *payload* contingut en el camp de dades del missatge NDEF en la transmissió de diferents dades [10].

### 5.7. Seguretat.

Un dels aspectes importants de les comunicacions a distància és la seguretat. Aquesta, permet evitar la interferència d'algun usuari o dispositiu durant la transmissió de dades, ja que aquestes poden ser de caràcter sensible.

Pel que fa a la seguretat en NFC, degut a que el funcionament d'aquesta tecnologia es limita a pocs centímetres, dificulta la intercepció de la comunicació entre lectors o *tags* NFC. D'altra banda, el tipus de comunicació *peer-to-peer*, implementa un mecanisme per al xifratge de les dades per a evitar-ne l'accés enfront dispositius atacants [18]. Tot i així, existeixen certs tipus d'amenaces i atacs que es poden dur a terme per tal de capturar les dades transmeses [17]:

- *Eavesdropping*. Consisteix en l'escolta, per part de l'usuari atacant, de les dades intercanviades entre els dispositius NFC. Existeix una gran dependència entre el mode d'operació NFC i l'execució d'aquest mètode, degut a les diferències de modulació i codificació utilitzades per a l'intercanvi de dades [18]. D'aquesta manera, quan el dispositiu actua en mode actiu, es pot interceptar la comunicació des d'una distància d'uns quants metres, tot depenent de la qualitat del receptor que utilitzi l'atacant, així com d'altres característiques referents al guany de l'antena i la seva geometria [18]. D'altra banda, quan el dispositiu funciona en el mode passiu, la potència del senyal és menor, i per tant, la distància que permet la seva intercepció és bastant més petita. Tot i així, es recomana l'ús de claus d'autenticació per a evitar aquest tipus d'atacs [18].
- *Data corruption*. En aquest cas, l'atacant té com a objectiu la destrucció o modificació de les dades transmeses per tal d'impossibilitar el seu ús un cop són rebudes pel dispositiu corresponent [18]. Així doncs, s'emeten senyals que interfereixen en la comunicació i provoquen col·lisió durant la transmissió, obtenint com a resultat una denegació en el servei [18].
- *Data modification*. La modificació de les dades és duu a terme mitjançant l'alteració de l'amplitud de la senyal modulada ASK [18]. D'aquesta manera, modificant l'amplitud, el valor descodificat pel receptor tindrà un valor diferent i, per tant, resultant en la modificació de les dades enviades [18].

- *Data insertion.* Aquest tipus d'atac, l'usuari assaltant de la comunicació, procura la introducció de dades en el procés de transferència d'informació entre els dispositius per tal d'alterar-ne el seu procés d'intercanvi normal [18].

Un cop anomenats el conjunt d'atacs i intercepció de les dades que es poden produir durant la comunicació NFC entre dos dispositius, s'analitzaran, seguidament algunes de les solucions aportades per tal de garantir, en tot moment i independentment de l'aplicació, la seguretat de la comunicació [18].

Així doncs, el principal mètode per tal de garantir la protecció de les dades, és l'establiment d'un canal segur entre els dos dispositius que interactuen en la comunicació NFC [18]. Aquest mètode, garantirà una comunicació segura davant d'atacs que intervenen durant la comunicació, com ara bé els que tenen a veure amb la modificació de les dades. Un camp d'aplicació de l'establiment d'un canal segur, és en les transaccions o pagaments mitjançant NFC. En aquest tipus d'aplicacions, les dades intercanviades entre el telèfon mòbil i el lector NFC han d'estar segures davant d'atacs que puguin interferir, corrompent la informació o bé extraient dades bancàries de l'usuari [18]. Així doncs, s'utilitza un servei de confiança, anomenat TSM, per tal de realitzar aquest tipus d'operacions [18].

## 6. PROCESSAT DE DADES AMB *RASPBERRY PI PICO*

En aquest apartat es detallen el conjunt d'aspectes referents al processament de les dades que s'han obtingut dels sensors químics resistius, així com la descripció del principal hardware utilitzat en el projecte i el software implementat per tal de la correcta interpretació i transmissió de la informació captada.

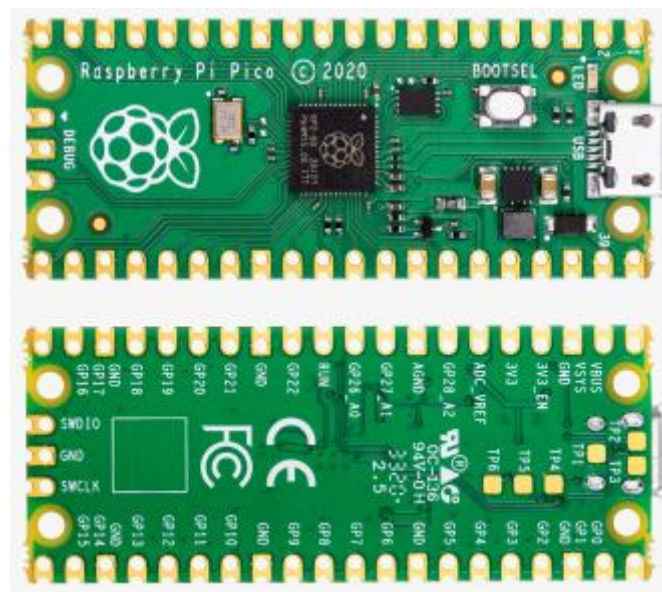
### 6.1. Introducció a *Raspberry Pi Pico*

*Raspberry Pi Pico* és el primer microcontrolador dissenyat per l'associació *Raspberry Pi* Foundation. A diferència dels altres models de *Raspberry Pi*, aquest no permet la instal·lació de cap sistema operatiu degut a que no correspon a l'àmbit dels microprocessadors [19]. D'aquesta manera, podem definir la *Raspberry Pi Pico* com un microcontrolador alternatiu a Arduino, més econòmic i amb algunes millores pel que fa a l'eficiència i el seu rendiment.

Així doncs, pot ser àmpliament utilitzat per a projectes d'electrònica i robòtica. S'ha decidit usar aquest microcontrolador en el desenvolupament del present projecte, per tal d'estudiar possibles alternatives a les tecnologies ja conegudes, així com posar en pràctica els coneixements adquirits en diferents productes del mercat actual.

#### 6.1.1. Descripció del hardware

*Raspberry Pi Pico* és un microcontrolador basat en el chip desenvolupat per la fundació *Raspberry Pi* RP2040 [19]. Aquest, ha estat dissenyat amb el principal objectiu de crear una plataforma flexible i de baix cost que integra el RP2040 [19].



**Figura 6.1.** Placa *Raspberry Pi Pico*.

Les principals característiques tècniques del microcontrolador *Raspberry Pi Pico* es resumeixen en la *Taula 6.1* [19].

| Característiques tècniques <i>Raspberry Pi Pico</i> |                                  |                             |
|-----------------------------------------------------|----------------------------------|-----------------------------|
| <b>MCU</b>                                          | RP2040 amb 2 MB de memòria Flash |                             |
| <b>Alimentació i dades</b>                          | Micro-USB tipus B                |                             |
| <b>Descripció dels pins</b>                         | <b>40 pins en total</b>          | 26 x GPIO                   |
|                                                     |                                  | 8 x GND                     |
|                                                     |                                  | 3 x alimentació             |
|                                                     |                                  | 2 x control del sistema     |
| <b>Perifèrics</b>                                   | <b>Comunicació sèrie</b>         | 2 x UART                    |
|                                                     |                                  | 2 x I <sup>2</sup> C        |
|                                                     |                                  | 2 x SPI                     |
|                                                     | <b>Control</b>                   | 16 x canals PWM             |
|                                                     | <b>Timers</b>                    | 1 x comptador amb 4 alarmes |
|                                                     |                                  | 1 x comptador real          |

Taula 6.1. Resum de les característiques tècniques del microcontrolador *Raspberry Pi Pico*.

En la *Figura 6.2* podem observar la distribució de les entrades i sortides dels 40 pins que integra la placa *Rev3* del microcontrolador *Raspberry Pi Pico*, així com el conjunt de perifèrics [19].

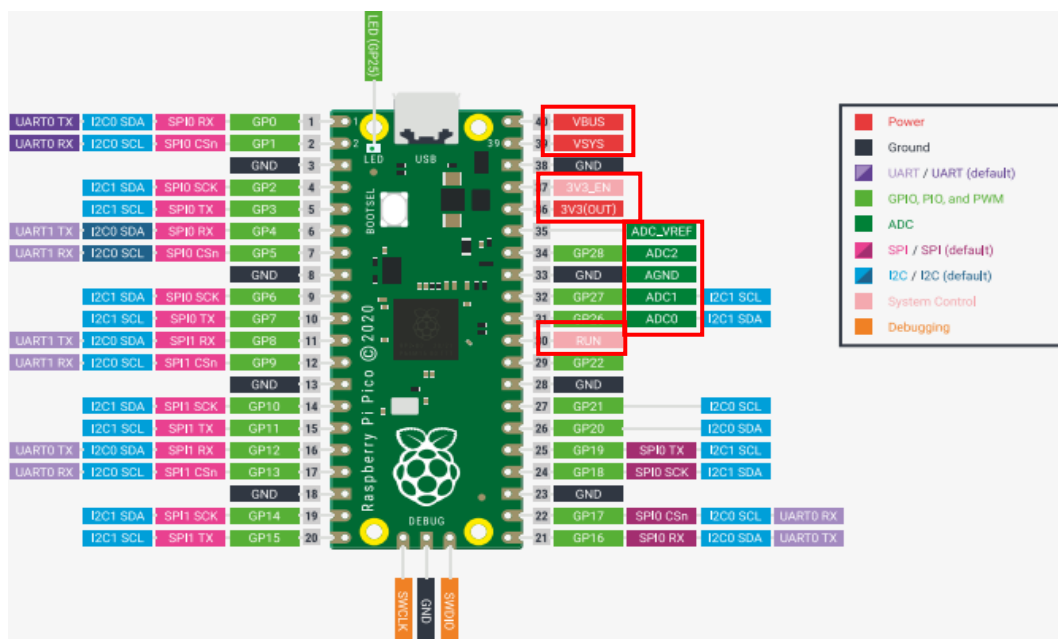


Figura 6.2. Distribució dels pins de la placa *Rev3*.

A continuació identificarem i descriurem els set pins més característics del microcontrolador per a comprendre la seva funcionalitat. Aquests, contempen els tres pins d'alimentació i els dos pins de control del sistema, comentats anteriorment en la *Taula 6.1*, així com dos pins addicionals referents al ADC i al GND [19].

- **VBUS (PIN\_40)**. Correspon al voltatge de 5 V d'alimentació de la placa a través del Micro-USB tipus B. Per tant, aquest pin subministra 5 V per a l'alimentació de perifèrics que ho requereixin.
- **VSYS (PIN\_39)**. Aquest fa referència al voltatge subministrat d'operació del conjunt del sistema, el qual pot variar entre el valor mínim de 1,8 V i el valor màxim de 5,5 V. Una vegada alimentat el sistema, el convertidor elevador-reductor (*buck-boost*) SMPS permet generar els 3,3 V requerits per al funcionament del xip RP2040 i els GPIOs.
- **SMPS**. Es tracta d'un convertidor elevador-reductor (*buck-boost*) integrat, el qual permet incrementar el voltatge funcionant com a elevador, en el cas en que el subministrament correspongui a 1,8 V, o disminuir el voltatge, actuant com a reductor, si l'alimentació externa correspon al valor màxim de 5,5 V.
- **3V3\_EN (PIN\_37)**. Permet tallar el subministrament d'energia cap al RP2040 quan aquest és curtcircuitat.
- **3V3 (PIN\_36)**. Correspon al voltatge de 3,3 V que permet alimentar el xip RP2040 així com el conjunt de GPIOs que aquest implementa. Es pot utilitzar per a alimentar elements externs a la placa, sempre i quan es respecti el límit de 300 mA de subministrament màxim.
- **ADC\_VREF (PIN\_35)**. Fa referència al voltatge de referència dels ADC que implementa la placa. Aquest permet el subministrament a partir de voltatges de referència externs, en cas que fos necessari.
- **AGND (PIN\_33)**. És el GND usat per als GPIOs corresponents al ADC0, ADC1, ADC2 i ADC\_VREF.
- **RUN (PIN\_30)**. Permet fer un *reset* del RP2040 quan aquest és curtcircuitat, connectant, per exemple, un polsador.

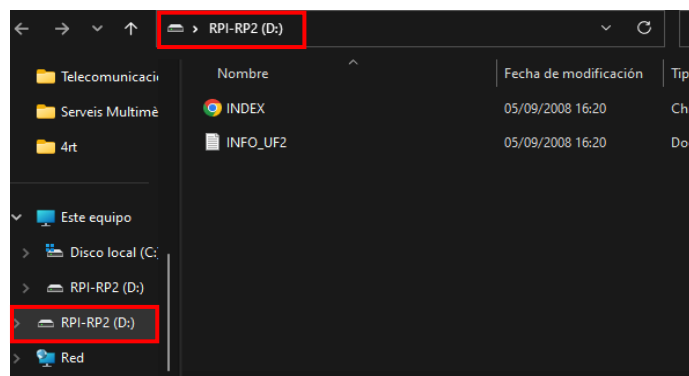
### 6.1.2. Programació

La *Raspberry Pi Pico* permet l'execució de programes dissenyats amb llenguatges com el *MicroPython* i d'altres de coneguts com el *C* o *C++*. Pel que fa al *MicroPython*, es tracta d'una implementació del llenguatge *Python 3*, el qual inclou un conjunt de llibreries estandarditzades referents al llenguatge de programació *Python*, i aquest ha estat dissenyat i optimitzat per tal de poder ser implementat i executat en l'àmbit dels microcontroladors.

Així doncs, l'entorn de desenvolupament integrat utilitzat per a la programació en *MicroPython* ha estat el *Thonny IDE*. A continuació, es mostrarà l'aspecte d'aquest entorn de desenvolupament, així com els passos seguits per a la configuració de la *Raspberry Pi Pico* i la seva preparació per a la programació.

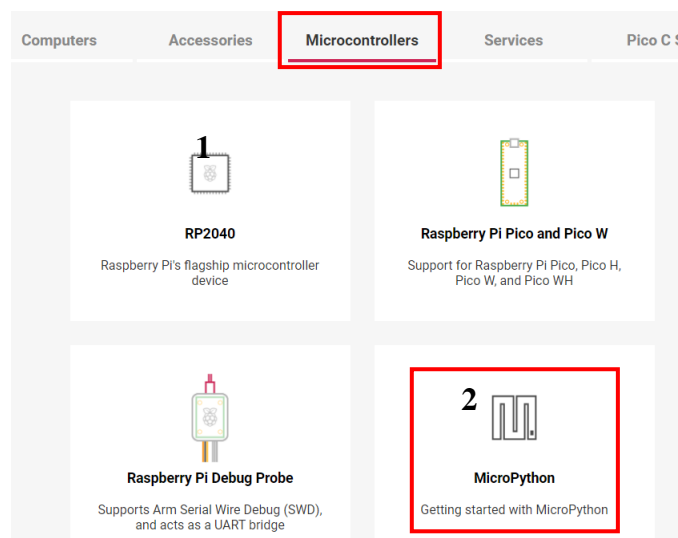
Per tal de configurar el microcontrolador Raspberry Pi Pico i habilitar-lo per a la seva programació, s'han realitzat els següents passos.

1. Connectem el dispositiu a l'ordinador a través del port USB, mentre mantenim polsat el botó *BOOTSEL* de la placa. Aquest procediment, permet la programació *firmware* del microcontrolador a través del port USB. Durant l'execució d'aquest procés, el nostre maquinari reconeixerà la *Raspberry Pi Pico* com un dispositiu extern.



**Figura 6.3.** Reconeixement de la *Raspberry Pi Pico* com a dispositiu extern.

2. Seguidament, entrem a la pàgina oficial de *Raspberry Pi* i cerquem l'apartat de *microcontrollers* i seleccionem l'apartat *MicroPython*. Un cop escollida aquesta opció, podrem descarregar el fitxer UF2 corresponent al *firmware* de *MicroPython*.



**Figura 6.4.** Primers passos seguits per la descàrrega del *firmware* de *MicroPython*.

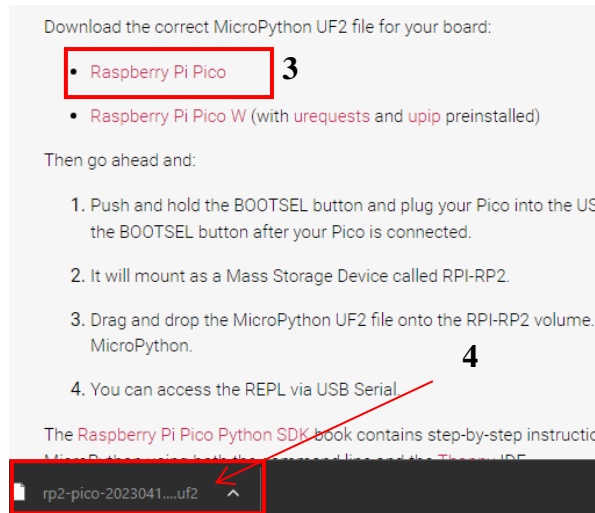


Figura 6.5. Descàrrega del firmware de MicroPython.

3. Un cop descarregat el fitxer, l'arrosseguem fins al *drive* RPI-RP2 corresponent al pas 1. El fitxer UF2, s'instal·larà automàticament a la memòria *flash* del microcontrolador i aquest es reiniciarà. Un cop arribats en aquest punt, la *Raspberry Pi Pico* ja estarà llesta per a la programació i l'execució de programes.

## 6.2. Comparació entre *Raspberry Pi Pico* i *Arduino*

En aquest apartat, es mostrarà una taula amb les principals diferències entre ambdós microcontroladors.

| Característiques     | Arduino Uno | <i>Raspberry Pi Pico</i> |
|----------------------|-------------|--------------------------|
| Preu oficial         | 24,00 €     | 4,75 €                   |
| Programació          | C/C++       | C/C++ i MicroPython      |
| Microcontrolador     | Atmega 328P | RP2040                   |
| Arquitectura         | 8-bit RISC  | 32-bit ARM               |
| Nuclis               | 1           | 2                        |
| Relotge CPU          | 16 MHz      | 48 MHz fins a 133 MHz    |
| Memòria RAM          | 2 kByte     | 264 kByte                |
| Memòria FLASH        | 32 kByte    | 2 MByte                  |
| Memòria EEPROM       | 1 kByte     | -                        |
| Voltatge alimentació | 5 VDC       | 3,3 VDC                  |
| GPIOs                | 20          | 26                       |

|                       |            |            |
|-----------------------|------------|------------|
| <b>ADC</b>            | 6 x 10-bit | 3 x 12-bit |
| <b>UART</b>           | 1          | 2          |
| <b>I<sup>2</sup>C</b> | 1          | 2          |
| <b>SPI</b>            | 1          | 2          |
| <b>PWM</b>            | 6          | 16         |

Taula 6.2. Diferències entre *Raspberry Pi Pico* i Arduino.

### 6.3. Mòdul MIKROE-1726

Per al desenvolupament de la comunicació NFC i, per tant, poder intercanviar les dades captades per la *Raspberry Pi Pico* i obtenir-les en la nostra aplicació mòbil, s'ha utilitzat el mòdul MIKROE-1726. Aquest, és un *tag* NFC que té capacitat per a realitzar lectura i escriptura de dades, ja que disposa del circuit integrat M24SR64-Y amb 64-kbits de memòria EEPROM, el qual permet realitzar l'intercanvi de dades a través de comunicació sèrie amb el microcontrolador i amb un dispositiu NFC a través de radiofreqüència a 13,56 MHz.

Tot i que aquest mòdul està dissenyat per a ser utilitzat fent ús del *socket* de mikroBUS<sup>TM</sup> i de les llibreries per a compiladors mikroC<sup>TM</sup>, mikroBasic<sup>TM</sup> i mikroPascal<sup>TM</sup>, ha sigut adaptat per tal de poder connectar-se amb la *Raspberry Pi Pico*.



Figura 6.6. Mòdul MIKROE-1726

#### 6.3.1 Descripció del hardware del tag

A continuació, es detallaran algunes de les característiques del conjunt del *tag*, així com del circuit integrat M24SR64-Y [24].

Característiques del tag:

- Permet únicament l'intercanvi de dades amb el microcontrolador a través de la interfície I2C.
- És compatible amb els estàndards ISO/IEC 14443 Tipus A i amb el NFC Forum Tipus 4.

- Suporta l'intercanvi de dades fent ús del protocol NDEF, amb 7 bytes d'identificador únic i contrasenya de 128 bits per a la protecció de les dades emmagatzemades.
- Abast de l'antena d'aproximadament 3 cm.
- Integra resistències de *pull-up* de 4,7 k $\Omega$  per als canals SDA i SCL necessàries en la comunicació I2C.
- Voltatge d'alimentació de 3,3 V.

#### Característiques del circuit integrat M24SR64-Y:

- Freqüència d'operació de 100 MHz per a la interfície I2C. Aquest paràmetre s'ha de tenir en compte per a la posterior programació i transmissió de dades des de la *Raspberry Pi Pico*.
- Velocitat de transmissió de dades de 106 kbps.
- Memòria EEPROM de 8-kbytes.
- Capacitat per a lectura i escriptura de dades fins a 246 bytes en una sola comanda NDEF.
- Integra pins GPO i RF per a la configuració de la senyal de sortida i l'activació o desactivació de les comandes de comunicació per radiofreqüència, respectivament.

### **6.4. Comunicació entre mòdul NFC i microcontrolador.**

En aquest apartat, es descriurà la principal interfície I2C de comunicació que permet l'intercanvi de dades entre el microcontrolador i el *tag* NFC.

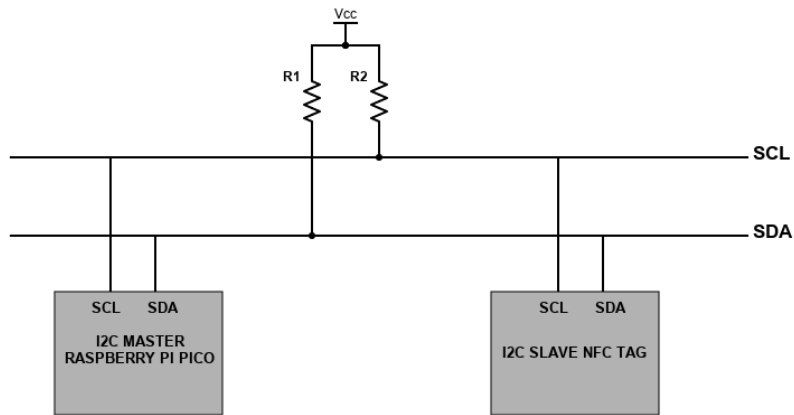
#### **6.4.1. I2C**

I2C és un bus de comunicació sèrie que permet la transferència de dades entre un microcontrolador i els seus perifèrics. Els dispositius que es connecten al bus, s'identifiquen com a dispositius *masters* i *slaves*. Típicament, el dispositiu *master* permet iniciar la comunicació, mentre que el dispositiu, o dispositius *slaves*, poden intervenir en la comunicació quan el *master* ho sol·licita. Per tal d'intercanviar les dades, el bus I2C disposa de dues línies, SDA i SCL, juntament amb la línia d'alimentació i *ground*.

La línia SDA permet identificar el canal per on es transmet la informació, i la línia SCL, correspon al canal de referència que determina la freqüència de validació dels bits de la línia de dades. A més a més, les dues línies SDA i SCL necessiten dues resistències, anomenades resistències de *pull-up* que es connecten cap a l'alimentació, degut a que totes dues línies funcionen des del punt de vista de la configuració *open-collector* [20].

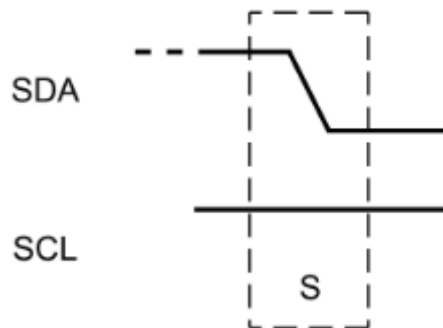
Aquest parell de resistències són necessàries per tal d'aconseguir un nivell de tensió proper al de la línia d'alimentació, i que representaria un nivell alt, o '1' lògic, en la transmissió de dades [20]. D'altra banda, si no es disposés d'aquestes resistències, la tensió de la línia seria molt propera a 0 V, interpretant sempre un nivell baix de tensió en ambdues línies [20].

Així doncs, un cop anomenades les característiques bàsiques de la interfície I2C, podem observar en la *Figura 6.7*, una representació de l'esquema de connexió entre un microcontrolador i el seu perifèric.



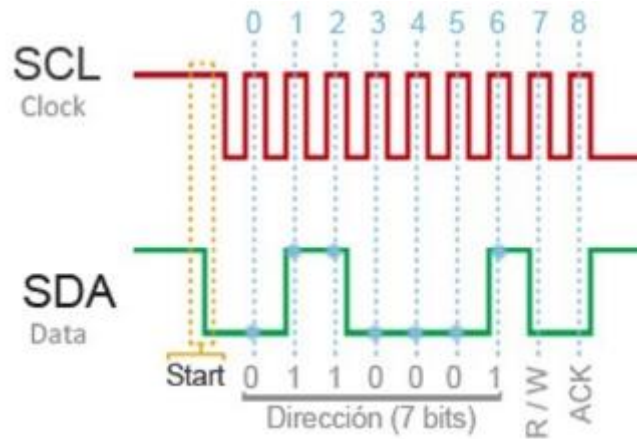
**Figura 6.7.** Exemple esquema de connexió I2C

Referent al protocol de comunicació, en condicions normals, el bus indica que està lliure mitjançant la representació d'ambdues línies SDA i SCL en estat lògic alt [20]. Per tal de poder iniciar la comunicació, abans de transmetre les dades, s'envia una condició de *start* que s'identifica com un canvi d'estat de la línia de dades SDA, passant d'un estat alt a un estat baix, mentre que la línia de senyal de rellotge manté el seu estat en alt.



**Figura 6.8.** Condició d'inici I2C.

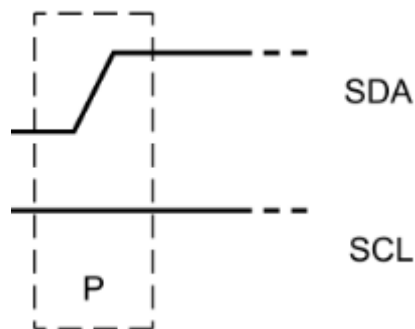
Seguidament, un cop transmesa la condició d'inici de la comunicació, els vuit bits següents que es transmetran per la línia de dades SDA correspondran a l'adreça del dispositiu amb el que es vol establir comunicació. D'aquests vuit bits, sis fan referència a l'adreça i l'últim identifica si el tipus d'operació que es vol realitzar és de lectura o escriptura [20]. L'enviament d'un '0' identificarà el procés com a escriptura, mentre que si s'envia un '1', el procés és de lectura. Finalment, si l'adreça que ha enviat el *master* identifica algun dispositiu dins de la interfície I2C, aquest respon amb un bit de confirmació (ACK) a nivell baix [20].



**Figura 6.9.** Exemple condició inici i transmissió d'adreça.

En el mode d'escriptura, a mesura que el dispositiu *master* envia dades al dispositiu *slave*, aquest respon al *master* enviant seqüències de confirmació per cada byte enviat, indicant que la informació s'ha rebut correctament [20]. Finalment, un cop s'ha acabat la transmissió de les dades, el *master* envia una petició per a tancar la comunicació. Aquesta petició es codifica amb un canvi de nivell baix a nivell alt de la línia de dades, mentre que la senyal de rellotge es manté en un estat alt [20].

En la *Figura 6.10* podem observar un exemple de la condició de finalització de la comunicació.



**Figura 6.10.** Condició per a finalitzar la comunicació I2C.

En resum, podem il·lustrar els conjunt de processos indicats anteriorment per a establir comunicació fent ús de la interfície I2C entre dos dispositius, mitjançant el següent diagrama de la *Figura 6.11*. El procés explicat correspon a l'utilitzat en el present projecte, on el dispositiu *master* realitza operacions d'escriptura al dispositiu *slave*.

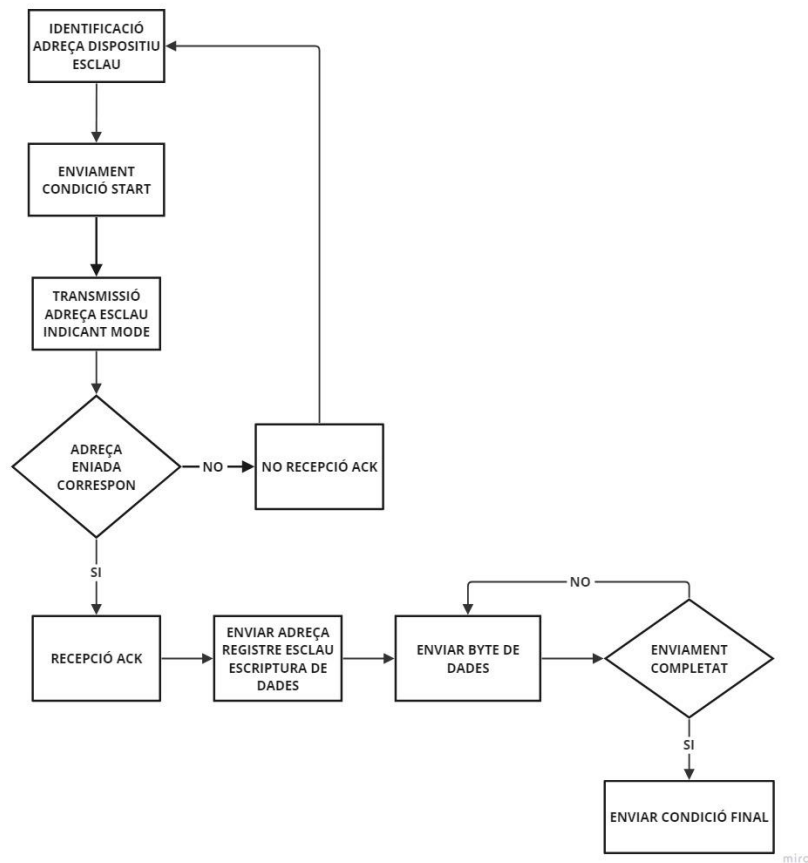


Figura 6.11. Il·lustració procés d'escriptura dispositiu *slave*.

## 6.5. Adquisició de dades.

Abans de realitzar el processament de les dades obtingudes, és necessari elaborar el disseny del sistema d'adquisició. En la *Figura 6.12*, es pot observar un esquema que representa de forma resumida, els passos necessaris per a l'acompliment d'aquesta etapa.

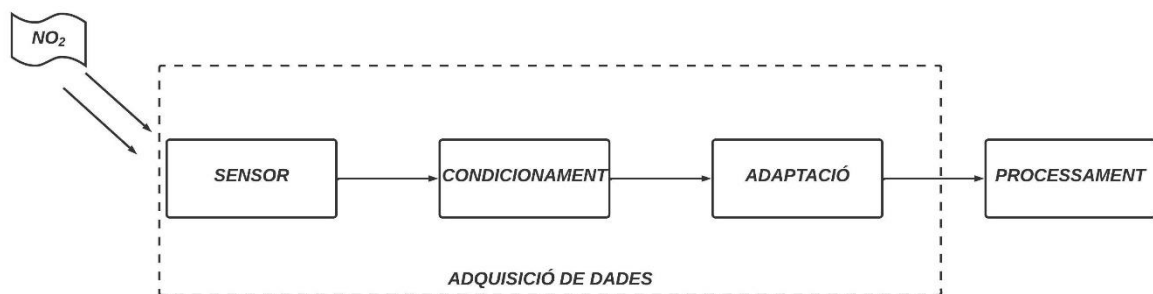


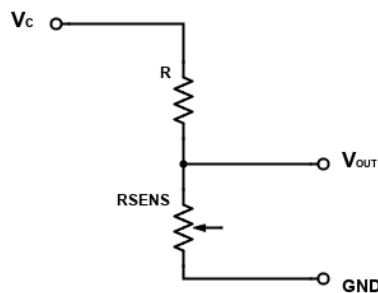
Figura 6.12. Representació del procés d'adquisició de dades.

Tal i com es pot considerar, el procediment d'adquisició de les dades està format per la sensorització, el condicionament i l'adaptació. Degut a que en el present projecte ja s'han explicat els aspectes referents a la sensorització, en aquest apartat es donarà èmfasi als processos corresponents al condicionament i l'adaptació.

### 6.5.1. Condicionament.

Segons s'ha comentat, els sensors desenvolupats corresponen a un comportament resistiu. D'aquesta manera, en funció de la concentració de gas, s'experimentarà una variació en la resistència del sensor. Així doncs, en el present projecte, l'objectiu de l'apartat de condicionament és la captació d'aquesta variació de resistència per a que posteriorment, pugui ser adaptada i processada de forma correcte. Per tant, s'han desenvolupat dos divisors de tensió (un per a cada sensor) per a traduir les variacions de la resistència del sensor, en variacions del voltatge [21], el qual podrà ser processat per el microcontrolador.

En la *Figura 6.13* podem observar un exemple del circuit de condicionament descrit anteriorment.



**Figura 6.13.** Esquema del circuit de condicionament.

D'aquesta manera, sent  $R$  el valor d'una resistència coneguda i  $V_C$  la tensió de referència de la *Raspberry Pi Pico*, podem determinar la variació del valor de  $R_{SENS}$  segons l'equació (6.1) característica del divisor de tensió.

$$V_{OUT} = \frac{R_{SENS} \cdot V_C}{R_{SENS} + R} \quad (6.1)$$

Per a determinar el voltatge de sortida ideal  $V_{OUT}$ , així com el valor de la resistència  $R$ , inicialment es mesura la resistència dels sensors  $R_{SENS}$  quan estan en contacte amb l'aire. D'aquesta manera, es pot determinar un valor de tensió òptim de lectura entre 0 i 3,3 V, sent aquest el rang conversor A/D del propi microcontrolador. Consegüentment, en la *Taula 6.1*, es poden observar els valors de resistència  $R_{SENS}$  i  $R$  determinats per a cada un dels sensors, així com el valor de la tensió de sortida  $V_{OUT}$  en cada cas.

| Elements del circuit de condicionament |            |        |                                                                                                                        |
|----------------------------------------|------------|--------|------------------------------------------------------------------------------------------------------------------------|
|                                        | $R_{SENS}$ | $R$    | $V_{OUT}$                                                                                                              |
| <b>Sensor 1</b>                        | 895 kΩ     | 1 MΩ   | $V_{OUT} = \frac{890 \text{ k}\Omega \cdot 3,3 \text{ V}}{890 \text{ k}\Omega + 1 \text{ M}\Omega} = 1,55 \text{ V}$   |
| <b>Sensor 2</b>                        | 335 kΩ     | 820 kΩ | $V_{OUT} = \frac{335 \text{ k}\Omega \cdot 3,3 \text{ V}}{335 \text{ k}\Omega + 820 \text{ k}\Omega} = 0,96 \text{ V}$ |

**Taula 6.3.** Elements del circuit de condicionament.

### 6.5.2. Adaptació.

Seguidament, pel que fa al procés d'adaptació del senyal de sortida del circuit de condicionament, aquest es duu a terme mitjançant l'ús del conversor A/D del microcontrolador.

El conversor que utilitza la *Raspberry Pi Pico* és de 16-bits amb un rang de voltatge d'entrada de 0 a 3,3 V. D'aquesta manera, podem definir la resolució d'aquest com:

$$Resolució = \frac{V_{ADC}}{2^{n-1}} = \frac{3,3V}{2^{16-1}} = 0,05035 \text{ mV} \quad (6.2)$$

On  $V_{ADC}$  és el valor del voltatge corresponent als 3,3 V i  $n$  correspon al nombre de bits del conversor, en aquest cas, 16-bits. Així doncs, la resolució del conversor és de 0,05035 mV tal i com es pot observar en l'expressió (6.2).

Per altra banda, per tal d'adquirir les dades a través del conversor A/D de la *Raspberry Pi Pico*, s'utilitza la funció ja implementada `.read_u16()` sobre la instància `ADC` creada en la seva inicialització. D'aquesta manera, s'obté un valor en el rang 0 i  $2^{16-1}$  que pot ser interpretat com a tensió seguint la següent expressió.

$$V_{ADC\_IN} = adc.read_u16() \cdot \left(\frac{V_{ADC}}{2^{16-1}}\right) \quad (6.3)$$

Seguidament, degut a que en el present cas la variable d'interès és la resistència, s'altera lleugerament l'expressió (6.1) del divisor de tensió per tal d'obtenir les variacions de resistència corresponents a les alteracions en la tensió.

$$R_{SENS} = \frac{V_{ADC\_IN} \cdot R}{(V_{ADC} - V_{ADC\_IN})} \quad (6.4)$$

Altrament, degut a la introducció d'una component de soroll a causa de la commutació del valor de referència de tensió de la pròpia *Raspberry Pi Pico*, s'ha hagut d'aplicar un filtre per a millorar el valor de les lectures analògiques [22][23]. D'aquesta manera, el filtre aplicat correspon a l'expressió (6.5).

$$V_F[n] = \alpha \cdot V[n] + (1 - \alpha) \cdot V_F[n - 1] \quad (6.5)$$

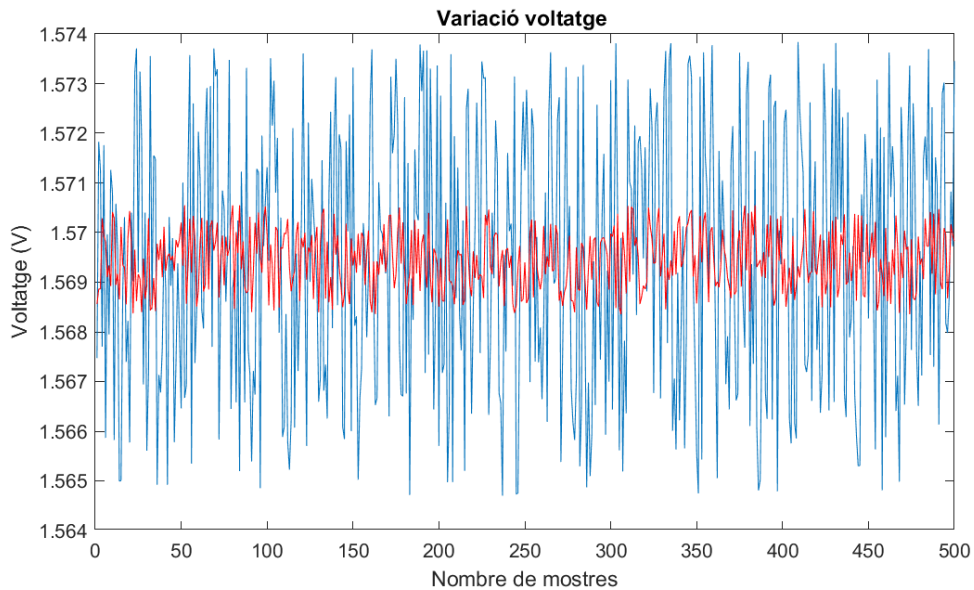
On  $V_F[n]$  fa referència al valor de voltatge filtrat,  $V[n]$  equival al valor de voltatge obtingut de la lectura analògica,  $V_F[n - 1]$  equival a la mostra anterior del valor de voltatge filtrat i  $\alpha$  correspon al coeficient (entre 0 i 1) que permet distingir el grau de filtratge que es vol aconseguir. Com més pròxim a 0 sigui aquest valor, millor resultat obtindrem.

Així doncs, aquest procediment s'ha implementat tant pel que fa al valor analògic de tensió que es llegeix a través del conversor A/D, com pel valor de resistència obtingut aplicant l'expressió (6.4).

A continuació, es poden observar els valors obtinguts sense la implementació del filtre i amb la correcció del filtre per al cas del sensor 2, ja que anàleg per al sensor 1.

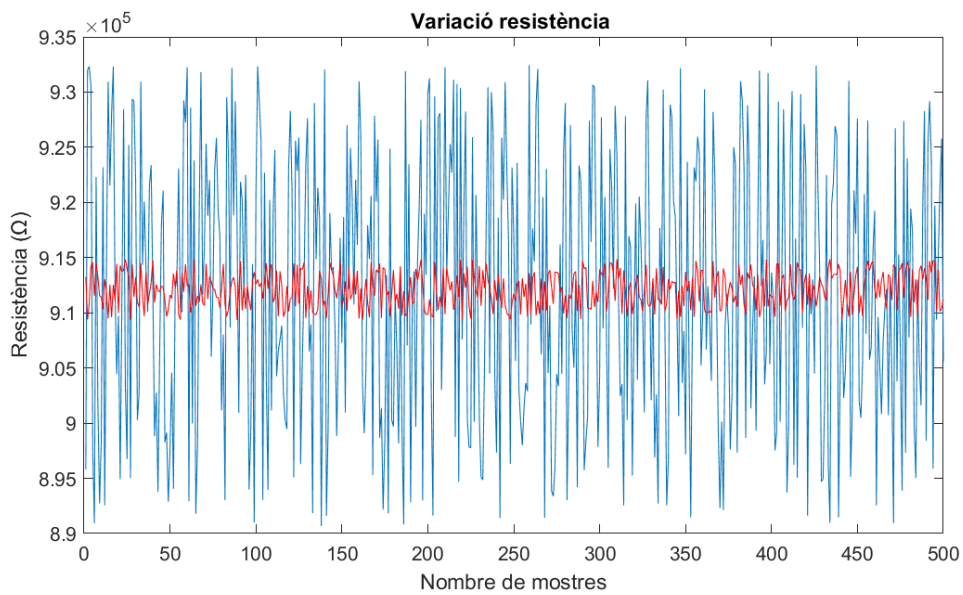
### Diferències aplicació i no aplicació del filtre

#### Valors de tensió amb i sense filtre



**Figura 6.14.** Representació de la variació de tensió amb filtre i sense filtre.

#### Valors de resistència amb i sense filtre



**Figura 6.15.** Representació de la variació de resistència amb filtre i sense filtre

**Taula 6.3.** Exemple de representació dels valors obtinguts filtrats.

La gràfica de color vermell correspon, en ambdós casos, a el valor d'interès amb el filtre aplicat. En el cas de la tensió mesurada a través del convertidor A/D (*Figura 6.14*), el valor de  $\alpha$  correspon a  $\alpha = 0,5$ , mentre que per al valor de resistència (*Figura 6.15*), es correspon a  $\alpha = 0,3$ .

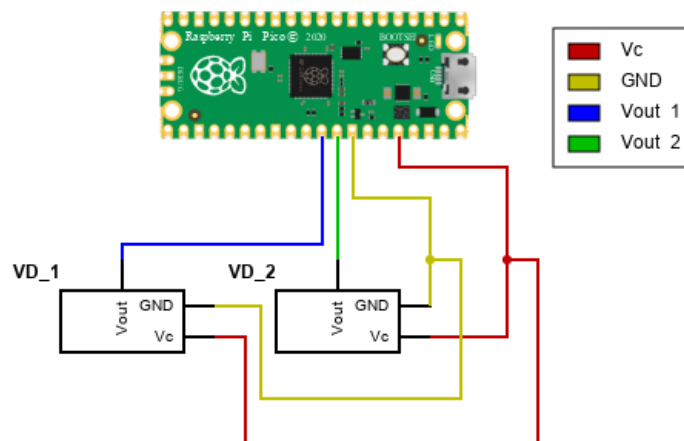
El valor de mostres capturades no correspon amb els valors experimentals obtinguts, sinó que s'han generat a partir d'una mostra referent a uns quants valors reals obtinguts en l'experiment, els quals es poden observar en la *Figura 6.16*.

|          |          |          |          |           |           |          |          |
|----------|----------|----------|----------|-----------|-----------|----------|----------|
| 1.56417  | 1.560147 | 896712.4 | 901108.1 | 0.9766811 | 0.9693272 | 341038.1 | 344713.1 |
| 1.573838 | 1.561516 | 898206.0 | 911756.1 | 0.9710414 | 0.9694986 | 341123.5 | 341892.7 |
| 1.566981 | 1.562346 | 899112.0 | 907305.0 | 0.9750698 | 0.9700557 | 341401.2 | 343905.9 |
| 1.551279 | 1.561239 | 897903.2 | 887093.9 | 0.9678186 | 0.9698319 | 341289.6 | 340287.2 |
| 1.548057 | 1.559921 | 896465.3 | 883622.7 | 0.9686243 | 0.9697112 | 341229.4 | 340688.2 |
| 1.551279 | 1.559057 | 895523.9 | 887093.9 | 0.9637903 | 0.9691191 | 340534.4 | 338286.4 |
| 1.549668 | 1.550118 | 894502.3 | 885356.8 | 0.9654016 | 0.9687473 | 340749.3 | 339085.9 |
| 1.556113 | 1.557917 | 894284.3 | 892325.0 | 0.9629847 | 0.968171  | 340462.5 | 337897.1 |
| 1.555308 | 1.557856 | 894000.5 | 891451.1 | 0.9589562 | 0.9672495 | 340004.1 | 335894.7 |
| 1.56417  | 1.558308 | 894708.9 | 901108.1 | 0.9637903 | 0.9669036 | 339832.0 | 338286.4 |
| 1.564576 | 1.558974 | 895434.5 | 901991.0 | 0.9589562 | 0.9661089 | 339437.1 | 335894.7 |
| 1.568199 | 1.559897 | 896439.1 | 905530.4 | 0.967013  | 0.9661993 | 339482.0 | 339886.5 |

**Figura 6.16.** Mostra d'alguns dels valors obtinguts durant l'experiment.

Els valors emmarcats per el requadre vermell, corresponen a les dades del sensor 2, mentre que les dades emmarcades pel requadre verd, fan referència a les dades del sensor 1. De la mateixa manera, les dues primeres columnes de cada requadre identifiquen els valors de tensió real i filtrada, respectivament, de cada un dels sensors. Anàlogament, les dues últimes columnes corresponen al valor de resistència filtrada i real, respectivament.

Finalment, en la *Figura 6.17*, es pot observar les connexions de l'etapa d'adquisició amb el microcontrolador.



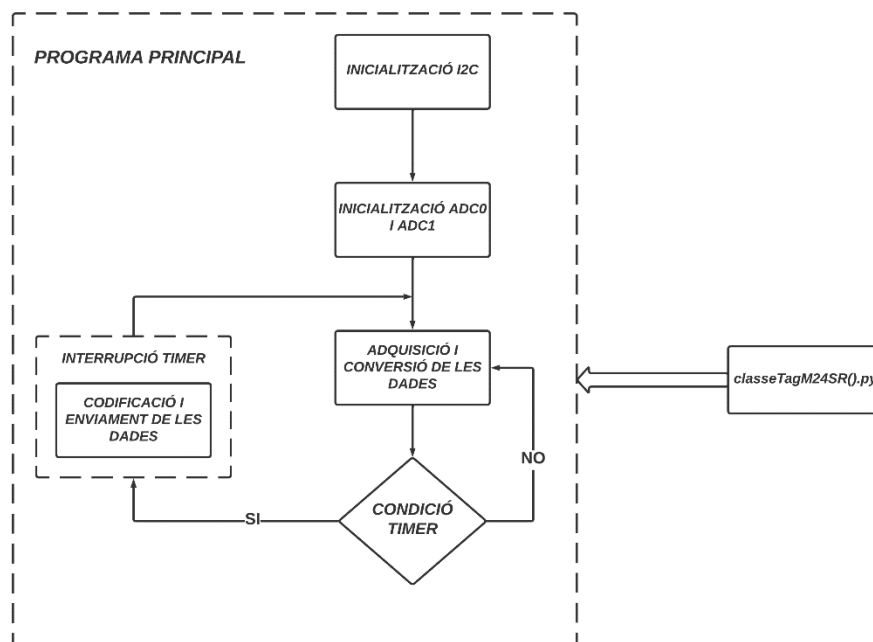
**Figura 6.17.** Representació de les connexions de l'etapa d'adquisició.

Els blocs *VD\_1* i *VD\_2* representen els divisors de tensió implementats.

## 6.6. Desenvolupament del codi.

Un cop exposats els passos necessaris per a configurar el microcontrolador *Raspberry Pi Pico*, així com el procés requerit per a la instal·lació del entorn de desenvolupament integrat, en aquest apartat es procedirà a analitzar el codi desenvolupat per tal d'aconseguir una correcta comunicació entre el microcontrolador i el *tag*, i per tant, poder escriure les dades captades pels sensor a la memòria de l'etiqueta NFC.

Principalment, podem identificar l'estructura del programa com dos fitxers *.py*, els quals identifiquen el programa principal *main* i una classe auxiliar, respectivament. S'ha decidit utilitzar aquesta jerarquia ja que facilita la organització en la programació, la detecció d'errors, així com la lectura i revisió del conjunt del programa. En la *Figura 6.13*, podem identificar de forma visual, l'estructura comentada prèviament, així com les parts més rellevants del seu contingut, el qual s'explicitarà a continuació.



**Figura 6.18.** Estructura del programa.

En el programa principal, s'ha importat la classe i s'ha creat una instància d'aquesta per tal de fer-ne les crides a través de l'objecte *classeTagM24SR()*, evitant la crida als mètodes implementats per mitjà del nom complet de la classe. Per altra banda, s'han implementat els mètodes necessaris per a la lectura de les dades dels sensors, així com el seu tractament abans de ser enviades.

Referent a la classe auxiliar desenvolupada, aquesta s'ha anomenat *classeTagM24SR()*, la qual consta d'un conjunt de mètodes que permetran la comunicació sèrie amb l'etiqueta NFC i la codificació del missatge a enviar, fent ús del format NDEF. A continuació, es pot observar en la *Figura 6.19*, un esquema permet identificar, de forma visual, l'estructura dels principals mètodes de la classe.

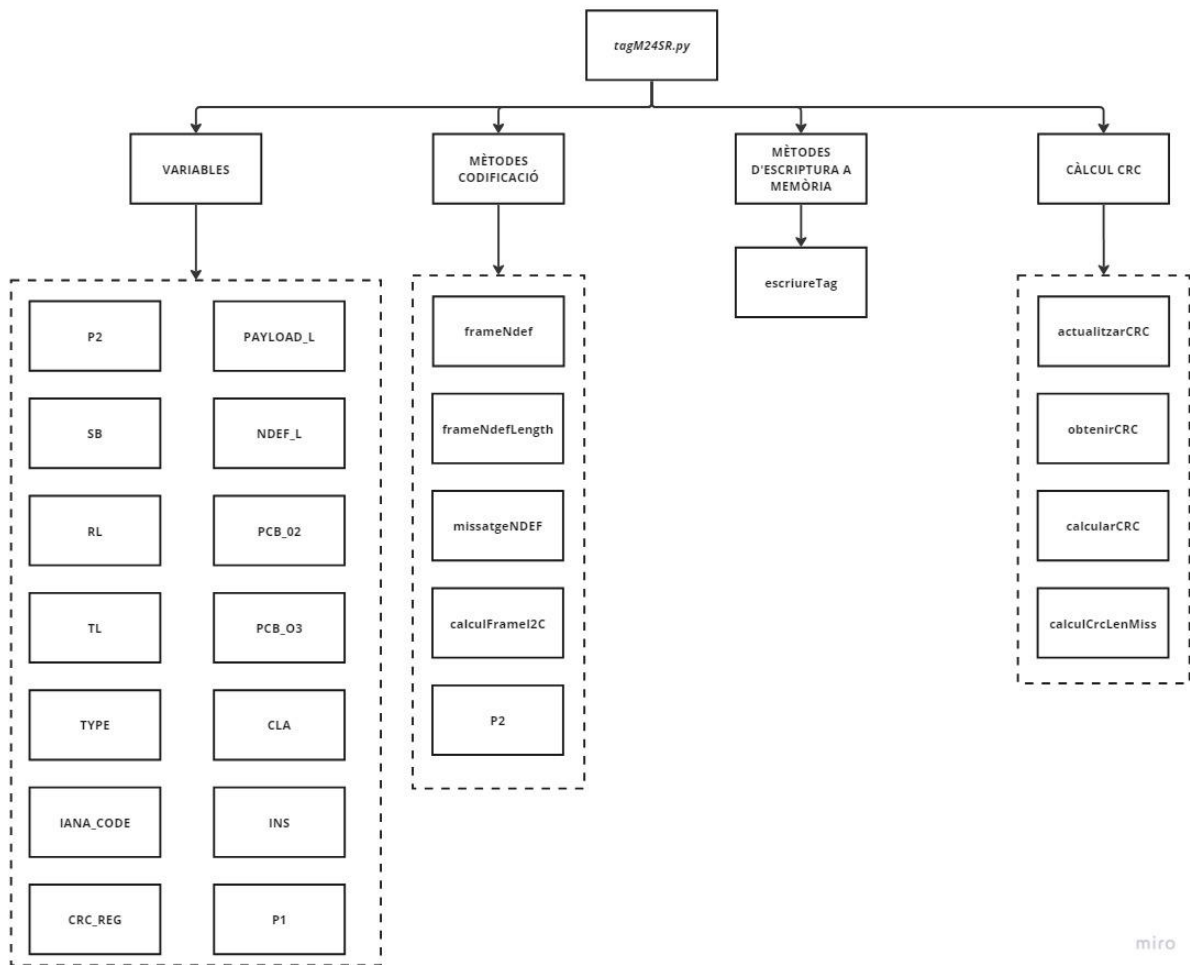


Figura 6.19. Exemple estructura de la classe TagM24SR().

Seguidament, en els apartats subsegüents, s’explicarà de forma explícita el funcionament de cada un dels mètodes, així com la presentació del conjunt de paràmetres d’entrada usats.

### 6.6.1. Variables de la classe.

A continuació, es detallarà el valor de cada una de les variables definides, així com la seva aplicació dins del programa. L’ordre seguit en l’explicació dels paràmetres es correspon d’esquerra a dreta, començant per el valor *SB* i acabant per el *CONS\_PAYLOAD\_L*, ja que els valors *P1* i *P2* s’exposaran junts.

***SB***. Correspon al byte que indica que la codificació de caràcters és UTF-8. El seu valor és 0x02.

***RL***. Fa referència al valor del *record layout*, el qual està format per el conjunt de *message flags* que formen la capçalera del missatge NDEF. Per al present projecte, s’ha simplificat el valor d’aquest valor degut a que el contingut del missatge NDEF consta d’un sol *record* en format text que representa la concentració de NO<sub>2</sub> detectada pel sensor. Per tant, per a determinar-ne el valor, s’ha construït el byte de la següent manera:

| Byte <i>RL</i> |    |    |    |    |     |   |   |
|----------------|----|----|----|----|-----|---|---|
| MB             | ME | CF | SR | IL | TNF |   |   |
| 1              | 1  | 0  | 1  | 0  | 0   | 0 | 1 |

Taula 6.4. Exemple càlcul del byte *RL*.

El significat de les sigles que formen el conjunt del byte *RL* està exposat en l'apartat 5.6 del projecte. Per tant, el byte *RL* té el valor en hexadecimal de 0xD1.

*TL*. El valor del byte *type length* en hexadecimal és 0x01, ja que correspon a la longitud del camp TNF.

*TYPE*. Pren el valor 0x54 en hexadecimal que permet representar que el contingut del *payload* correspon a un missatge de text.

*IANA\_CODE*. Aquest valor són dos bytes, 0x65 i 0x6E, que representen la codificació del llenguatge del missatge. En el present cas, els valors 0x65 i 0x6E corresponen a 'en', representant l'anglès.

*CRC\_REG*. Representa el contingut inicial que ha de tenir el registre per al càlcul del CRC. Aquest valor està definit en el *datasheet* com a dos bytes hexadecimals 0x6363 [24].

*P1/P2*. El seu valor varia en funció de la trama que es vol transmetre. En les trames que no venen definides, utilitzem els valors 0x01 i 0x02.

*INS*. Aquest camp pren un valor hexadecimal diferent en funció de si s'està realitzant una instrucció de lectura a memòria, escriptura a memòria o seleccionant un tipus d'instrucció. En el cas d'escriptura, el valor correspon a 0xD6 [24]. En el cas en que es realitzi l'enviament o selecció d'una instrucció, aquest valor és 0xA4 [24].

*CLA*. Correspon al *class byte* [24]. Per al programa realitzat sempre pren el valor de 0x00.

*PCB-03/PCB-02*. Representa el *protocol control byte* i el seu valor es va alternant entre 0x02 i 0x03. S'utilitzen en la capçalera de la trama per a enviar dades o comandes.

*CONS\_NDEF\_L*. Correspon al valor constant que s'ha de tenir en compte per al càlcul de la llargada del missatge NDEF (*payload* + capçalera). La llargada del missatge NDEF engloba la mida de la capçalera i la mida del missatge. La mida de la capçalera està formada per la mida dels bytes *RL*, *TL*, *TYPE* i el *payload length*, els quals ocupen 1 byte cada un. Per tant, el valor definit és 4. Aquest valor serà sumat a la llargada total del *payload*. El valor del *payload length* variarà en el cas en què el missatge ocupés més d'1 byte.

*CONS\_PAYLOAD\_L*. Identifica la llargada de la capçalera del *payload*, la qual descriu la codificació del missatge i l'*status byte*. Per tant, aquest correspondrà a la mida dels bytes que identifiquen el *IANA\_CODE* (2 bytes) i el *SB* (1 byte).

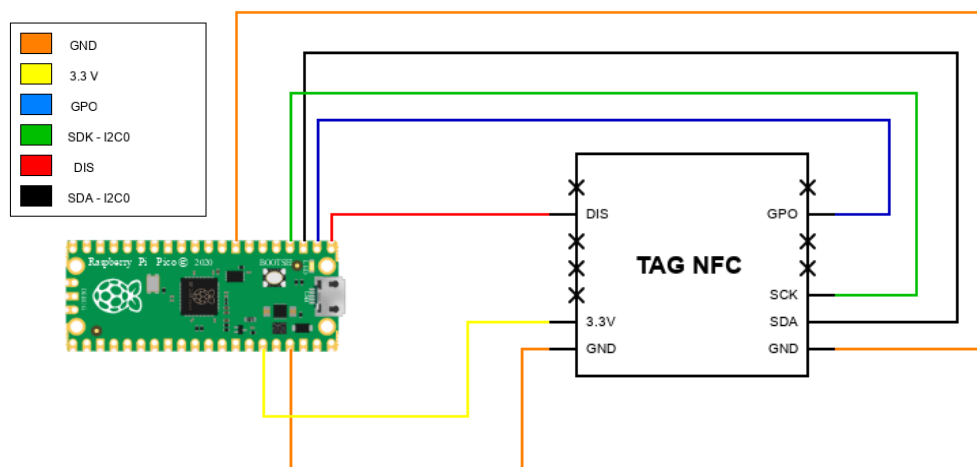
Finalment, en la *Figura 6.15* podem observar la declaració del atributs constants anteriorment mencionats en el nostre programa.

```
class tagM245R():
    I2C_ADDRESS = 0x56
    PCB_02 = 0x02
    PCB_03 = 0x03
    CLA = 0x00
    INS = 0x06
    P1 = 0x00
    P2 = 0x02
    RL = 0xB1
    TL = 0x01
    TYPE = 0x54
    SB = 0x02
    IANA_CODE = [0x65,0x6E]
    CRC_REG = 0x6363
    NDEF_L = 4
    PAYLOAD_L = 3
```

**Figura 6.20.** Variables definides en el programa.

### 6.6.2. Comunicació entre microcontrolador i tag NFC.

La comunicació entre l'etiqueta NFC i el microcontrolador, s'ha realitzat a través de la interfície sèrie I2C. Tal i com s'ha vist en l'*apartat 6.1.1*, la *Raspberry Pi Pico* disposa de dos interfícies I2C, la I2C0 i la I2C1. A continuació, es presentarà un esquema de les connexions realitzades.



**Figura 6.21.** Esquema de connexions *Raspberry Pi Pico* amb *tag NFC*.

En el cas present, s'ha utilitzat la interfície I2C1 per a la comunicació entre el *tag* i el microcontrolador. Aquest paràmetre, s'ha de tenir en compte per a la inicialització de la comunicació I2C al començament del desenvolupament del programa.

Un cop realitzades les connexions entre el *tag* i el microcontrolador, es va comprovar l'adreça del *tag NFC*. En la *Figura 6.17* podem observar com s'executa la funció *i2c.scan()* per tal d'observar les adreces disponibles de la interfície I2C.

```

i2c = I2C(1, sda=machine.Pin(6), scl=machine.Pin(7),freq=100000)

print('Escanejant I2C...')
dispositius = (i2c.scan())
print(dispositius)
if(len(dispositius) == 0):
    print("No s'han trobat dispositius i2c")
else:
    print("Adreces i2c trobades:", len(dispositius))

for i in dispositius:
    print("Decimal: ",i," | Hexadecimal: ", hex(i))

```

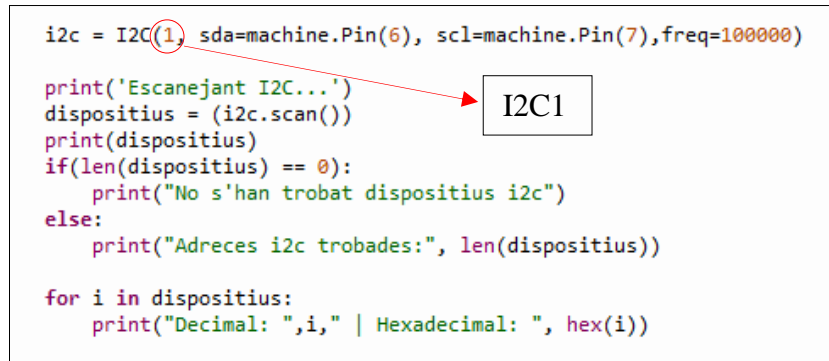


Figura 6.22. Escaneig i obtenció de l'adreça I2C.

Finalment, el resultat que obtenim és el següent:

```

Escanejant I2C...
[45, 86]
Adreces i2c trobades: 2
Decimal: 45 | Hexadecimal: 0x2d
Decimal: 86 | Hexadecimal: 0x56

```

Figura 6.23. Exemple resultat de les adreces disponibles en la interfície I2C.

Comprovant ambdós valors amb els del *datasheet*, observem que en la documentació l'adreça que caracteritza l'etiqueta és 0xAC. Examinant el resultat obtingut, podem observar que desplaçant un bit cap a l'esquerra l'adreça 0x56, obtenim com a resultat 0xAC. Per tant, afegim un nou atribut a la classe que identificarà l'adreça I2C de l'etiqueta i permetrà l'intercanvi de dades.

Seguidament, abans de començar amb la programació referent a l'escriptura de dades al *tag* NFC, es va fer un estudi de les característiques del circuit integrat M24SR64-Y del qual disposa l'etiqueta NFC. D'aquesta manera, es van poder determinar el processos necessaris, que engloben una sèrie de comandes, per tal de poder realitzar la correcta escriptura a memòria. Degut a que l'accés a memòria del M24SR64-Y no pot ser de forma directe, a continuació es detallarà l'estructura de la memòria del circuit integrat M24SR64-Y, juntament amb els passos que s'han de seguir per tal de poder escriure o llegir informació de la memòria EEPROM del circuit integrat.

L'estructura de la memòria EEPROM de l'etiqueta NFC és la següent:

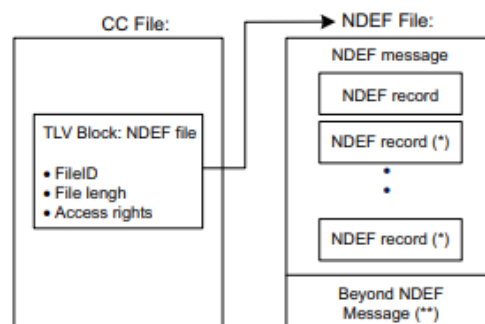
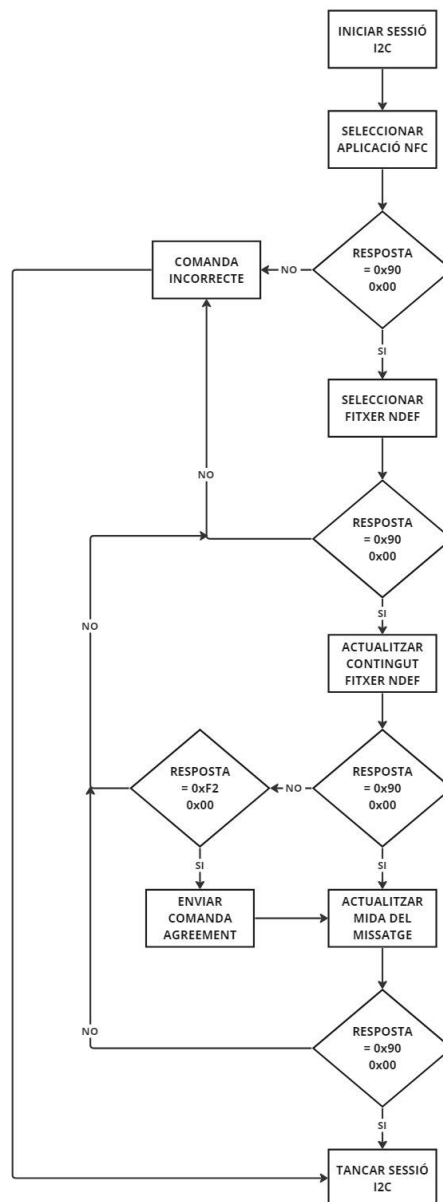


Figura 6.24. Estructura de la memòria EEPROM del circuit integrat M24SR64-Y.

Podem observar en la *Figura 6.19* que l'estructura de la memòria està formada per dos fitxers principals, el *CC File* i el *NDEF File*. El *CC File* ens permet obtenir informació sobre el *NDEF File*, on s'emmagatzemen les dades que volem escriure, i del propi circuit integrat M24SR64-Y. Consegüentment, per a realitzar operacions de lectura o escriptura al *NDEF File*, primer es pot sol·licitar accés al *CC File* per tal d'obtenir característiques generals del contingut del missatge.

En el present projecte, s'ha decidit ometre aquest pas i realitzar la operació d'escriptura directament al *NDEF File*, sense seleccionar el *CC File*. En atenció a això, el programa realitzat compren les següents comandes i comprovacions, mostrades en el següent diagrama de flux.



**Figura 6.25.** Passos per a l'escriptura de contingut al fitxer NDEF de la memòria.

Cal recalcar que el diagrama observat correspon al programa inicialment utilitzat per a la prova de comunicació. El programa final no compren el conjunt de comprovacions realitzades i per tant, les comandes i processos d'escriptura es realitzen directament, agilitzant així la comunicació.

En primer lloc, és necessari iniciar una sessió I2C per tal de poder realitzar la comunicació. La instrucció que permet la inicialització es realitza mitjançant l'escriptura del valor 0x26 a l'adreça I2C de l'etiqueta, fent ús de la funció *i2c.writeto(addr,buf)*, definida en la classe I2C. Un cop iniciada la sessió, ja es pot prosseguir amb l'enviament de les següents comandes. Mentre la sessió I2C està iniciada, no es pot accedir a través de RF al contingut de la memòria de l'etiqueta.

La trama de bytes que forma les comandes *Seleccionar Aplicació NFC<sup>1</sup>*, *Seleccionar Fitxer NDEF<sup>2</sup>* i *Tancar Sessió I2C*, venen definides en el *datasheet* del M24SR64-Y. Les comandes referents a *Actualitzar Mida del Fitxer amb la Mida del Missatge* i *Actualitzar el Contingut* corresponen a la codificació de les dades per a l'enviament de la trama de bytes que caracteritzen la longitud del missatge NDEF i el propi missatge, respectivament.

La trama de bytes està formada per els següents camps, alguns d'ells vistos en l'apartat referent a la declaració de variables [24][25].

| Trama I2C |     |     |    |    |    |       |    |                  |                  |
|-----------|-----|-----|----|----|----|-------|----|------------------|------------------|
| PCB       | CLA | INS | P1 | P2 | Lc | Dades | Le | CRC <sub>1</sub> | CRC <sub>2</sub> |

Taula 6.5. Trama missatge I2C que identifica la transmissió de les comandes<sup>1,2</sup>

Així doncs, el contingut de les trames per a seleccionar l'aplicació NFC [24], i seleccionar el fitxer NDEF es pot observar a continuació [24][25]:

| Trama I2C Seleccionar Aplicació NFC |      |      |      |      |      |                                       |      |                  |                  |
|-------------------------------------|------|------|------|------|------|---------------------------------------|------|------------------|------------------|
| PCB                                 | CLA  | INS  | P1   | P2   | Lc   | Dades                                 | Le   | CRC <sub>1</sub> | CRC <sub>2</sub> |
| 0x02                                | 0x00 | 0xA4 | 0x04 | 0x00 | 0x07 | 0xD2 0x76 0x00 0x00<br>0x85 0x01 0x01 | 0x00 | 0x35             | 0xC0             |

Taula 6.6. Trama I2C Seleccionar Aplicació NFC.

| Trama I2C Seleccionar Fitxer NDEF |      |      |      |      |      |           |    |                  |                  |
|-----------------------------------|------|------|------|------|------|-----------|----|------------------|------------------|
| PCB                               | CLA  | INS  | P1   | P2   | Lc   | Dades     | Le | CRC <sub>1</sub> | CRC <sub>2</sub> |
| 0x02                              | 0x00 | 0xA4 | 0x00 | 0x0C | 0x02 | 0x00 0x01 | -  | 0x3E             | 0xFD             |

Taula 6.7. Trama I2C Seleccionar Fitxer NDEF.

Utilitzant la mateixa funció *i2c.writeto(addr,buf)*, podem enviar cada una de les comandes a través de I2C. En aquest cas, el paràmetre *buf* correspon a tota la trama I2C, expressada com una cadena de bytes.

Pel que fa a les trames *Actualitzar Mida del Fitxer* i *Actualitzar Contingut del Fitxer*, el format general de la trama és el mateix, però el contingut del camp de dades i els valors del CRC variaran en funció del missatge que es vulgui escriure. Tot i així, alguns paràmetres del camp de dades que identifiquen la codificació del missatge en el format NDEF,

romandran constants per a qualsevol missatge, tal i com s'ha vist en l'apartat 6.6.1. A continuació, es presentarà un exemple del format general d'ambdues trames.

| <b>Trama I2C Actualitzar Mida del Fitxer</b> |            |            |           |           |           |              |           |                        |                        |
|----------------------------------------------|------------|------------|-----------|-----------|-----------|--------------|-----------|------------------------|------------------------|
| <i>PCB</i>                                   | <i>CLA</i> | <i>INS</i> | <i>P1</i> | <i>P2</i> | <i>Lc</i> | <i>Dades</i> | <i>Le</i> | <i>CRC<sub>1</sub></i> | <i>CRC<sub>2</sub></i> |
| 0x03                                         | 0x00       | 0xD6       | 0x00      | 0x00      | 0x02      | 0xXX         | -         | 0xXX                   | 0xXX                   |

**Taula 6.8.** Trama I2C Actualitzar Mida del Fitxer.

En aquest cas, el camp de dades contindrà la llargada del missatge NDEF.

| <b>Trama I2C Actualitzar Contingut del Fitxer</b> |            |            |           |           |           |                                                   |           |                        |                        |
|---------------------------------------------------|------------|------------|-----------|-----------|-----------|---------------------------------------------------|-----------|------------------------|------------------------|
| <i>PCB</i>                                        | <i>CLA</i> | <i>INS</i> | <i>P1</i> | <i>P2</i> | <i>Lc</i> | <i>Dades</i>                                      | <i>Le</i> | <i>CRC<sub>1</sub></i> | <i>CRC<sub>2</sub></i> |
| 0x03                                              | 0x00       | 0xD6       | 0x00      | 0x00      | 0x02      | 0xXX 0xD1 0x01 0x54<br>0x02 0x65 0x6E 0xXX<br>... | -         | 0xXX                   | 0xXX                   |

**Taula 6.9.** Trama I2C Actualitzar Contingut del Fitxer.

D'aquesta manera, els paràmetres ja escrits en el camp de dades corresponen als paràmetres constants declarats en la *classeTagM24SR()* del programa realitzat. Per altra banda, el primer valor no determinat que podem observar en el camp de dades de la trama, fa referència a la longitud del missatge NDEF complet, tenint en compte el conjunt de paràmetres de la capçalera. Pel que fa al segon terme no definit, aquest correspon a tota la trama que representa el missatge que es vol enviar.

Altrament, per a cada comanda enviada de les anteriors, el *tag* envia una resposta cap al dispositiu *master* de la comunicació I2C amb informació referent a la correcta recepció del missatge, o informació sobre algun error en la comunicació, principalment degut a la mala formació de la trama que s'ha transmès. És per això que en el programa realitzat, cada vegada que s'envia una comanda cap a l'etiqueta NFC, es llegeix la resposta que envia el circuit integrat M24SR64-Y del *tag* a través de la interfície I2C. Per a això, s'ha utilitzat la funció *i2c.readfrom(addr,nbytes)*, que llegeix un número *n* de bytes de l'adreça que identifica l'etiqueta NFC.

La trama I2C que conté la resposta del *tag* està formada, principalment, per els següents camps [24].

| <b>Resposta M24SR64-Y</b> |            |            |                        |                        |
|---------------------------|------------|------------|------------------------|------------------------|
| <i>PCB</i>                | <i>SW1</i> | <i>SW2</i> | <i>CRC<sub>1</sub></i> | <i>CRC<sub>2</sub></i> |
|                           |            |            |                        |                        |

**Taula 6.10.** Resposta M24SR64-Y.

Els camps més importants de la trama I2C de la resposta són el *SW1* i el *SW2*. La comanda s'ha transmès correctament quan el valor d'aquests dos bytes és 0x90 i 0x00, respectivament.

Pel que fa a la transmissió l'escritura del missatge NDEF, s'ha vist que la resposta pot ser diferent de 0x90 0x00. Així doncs com es pot veure en la *Figura 6.20*, s'ha establert una condició en el programa per tal de analitzar la trama rebuda. Aquest valor, correspon al *Waiting Frame Extension*, el qual ens indica que el M24SR64-Y necessita més de 9,6 ms per a l'execució de la comanda o resposta. Per tal de completar l'execució de la comanda o de la resposta, s'ha d'enviar al *tag* la mateixa trama *Waiting Frame Extension* rebuda. En cas contrari, no s'acomplirà la petició.

Finalment, un cop s'han realitzat aquest conjunt de processos, les dades han quedat emmagatzemades en el fitxer NDEF de la memòria EEPROM i es tanca la sessió I2C per tal de poder llegir les dades amb el dispositiu mòbil.

A continuació, es podrà observar un exemple del conjunt de comandes i respostes enviades detallades amb antelació.

```

Escanejant I2C...
[45, 86]
Adreces i2c trobades: 2
Decimal: 45 | Hexadecimal: 0x2d

Decimal: 86 | Hexadecimal: 0x56

Sessió I2C iniciada

Resposta Seleccionar App NFC: b'\x02\x90\x00\xff1\t'

Comanda enviada correctament

Resposta seleccionar Fitxer NDEF: b'\x03\x90\x00-S'

Comanda enviada correctament

Resposta actualitzar contingut del fitxer NDEF b'\xf2\x04<\x17'

Necessites la comanda agreement

Agreement enviat correctament

Resposta agreement: b'\x02\x90\x00\xff1\t'

Missatge enviat correctament

Resposta actualitzar mida del missatge NDEF: b'\x03\x90\x00-S'

Missatge enviat correctament

Resposta tancar sessió NFC: b'\xc2\xe0\xb4'

Sessió I2C finalitzada

```

**Figura 6.27.** Exemple de comunicació I2C amb el *tag* NFC.

### 6.6.3. Descripció dels mètodes de la classe.

Un cop explicat de forma general els passos i processos realitzats per a establir comunicació amb l'etiqueta NFC i escriure-hi un missatge, en aquest apartat es detallaran els mètodes principals que s'han implementat en la classe per tal de codificar el missatge i enviar-lo a través de I2C. Pel que fa al càlcul del CRC, s'ha fet ús de la següent guia [25] per a la implementació del codi. Anàlogament, s'ha usat la mateixa referència per a la definició de les trames que identifiquen cada una de les comandes a enviar [25].

| Mètodes de la classe   |                                                         |                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------|---------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nom                    | Paràmetres entrada                                      | Descripció                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>__init__</code>  | <i>i2c: I2C</i>                                         | És el constructor de la classe que permet la inicialització de l'objecte <i>I2C</i> del programa principal per tal d'implementar els mètodes de lectura i escriptura dins la classe.                                                                                                                                                                                                             |
| <i>frameNdef</i>       | <i>ndef_length: bytes</i><br><i>ndef_payload: bytes</i> | Permet el càlcul de la capçalera de la trama NDEF. Es reben els valors de la mida de la trama NDEF i de la mida del missatge per paràmetre. Finalment es concatenen aquests dos paràmetres amb els altres definits com a variables de la classe. Es retorna la capçalera de la trama que correspon al missatge en concret que es vol enviar.                                                     |
| <i>frameNdefLength</i> | <i>ndef_length: bytes</i>                               | Aquest mètode calcula la trama que permet identificar la mida del missatge NDEF. Es rep per paràmetre la dimensió de tota la trama NDEF, inclosa la capçalera. Es retorna la trama que s'envia a través de I2C per actualitzar la mida del missatge NDEF.                                                                                                                                        |
| <i>missatgeNdef</i>    | <i>dades: string</i>                                    | Permet codificar les dades captades per els sensors i l'hora de detecció, generant la trama hexadecimal equivalent. També calcula cadascuna de les dues llargades referents al missatge i a la trama NDEF. Es reben per paràmetre el valor de la concentració i l'hora de la mesura. Es retorna la mida del missatge, la mida de la trama NDEF i el missatge codificat com una cadena de bytes.  |
| <i>calculFrameI2C</i>  | <i>data: bytes [3]</i>                                  | Permet generar tota la trama NDEF corresponen al missatge que s'envia. Es rep per paràmetre un vector de 3 posicions corresponen als valors retornats per la funció <i>missatgeNdef</i> . Es separen els tres valors, referents a la mida del missatge, la mida de la trama NDEF i el missatge codificat. Es concatenen cada un dels valors per generar la trama i es calcula el CRC. Finalment, |

|                                  |                                                            |                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------------|------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                  |                                                            | es retorna tota la trama I2C que s'envia per actualitzar el contingut del fitxer NDEF.                                                                                                                                                                                                                                                                |
| <b><i>calculCrCLenMiss</i></b>   | <i>Llargada:</i><br><i>bytes</i>                           | Calcula el CRC corresponent a la trama de bytes que es transmet per I2C i que permet actualitzar la mida del missatge NDEF. Es rep per paràmetre la llargada, es calcula tota que s'ha d'enviar i finalment es calcula el CRC. Retorna tota la trama com una cadena de bytes.                                                                         |
| <b><i>comprovarResposta</i></b>  | <i>missatge:</i><br><i>bytes</i>                           | Aquest mètode ens serveix per a comprovar la resposta del <i>tag</i> , després de l'enviament del missatge o comanda corresponent. Es rep per paràmetre la trama de bytes corresponent a la resposta i es comprova que aquesta contingui els dos bytes 0x90 0x00 que indiquen la correcta recepció. Retorna un booleà.                                |
| <b><i>comprovarAgreement</i></b> | <i>agr: bytes</i>                                          | Si la resposta rebuda no correspon a 0x90 0x00, amb aquest mètode es comprova que es necessita la verificació del <i>master</i> per a seguir amb l'execució de la comanda. Es comprova que la trama rebuda té un byte igual a 0xF2. Es rep per paràmetre la resposta i es retornà un booleà.                                                          |
| <b><i>escriureTag</i></b>        | <i>missatge:</i><br><i>bytes</i><br><br><i>length: int</i> | Permet escriure el missatge codificat o comanda a través de la interfície I2C i a partir de l'adreça del dispositiu <i>slave</i> . S'obté la resposta que transmet el <i>tag</i> un cop s'ha finalitzat l'operació d'escriptura. Es rep per paràmetre el nombre de bytes de la resposta i es retorna una cadena de bytes corresponent al valor rebut. |
| <b><i>actualitzarCRC</i></b>     | <i>data:bytes</i>                                          | Permet calcular el valor del CRC de la trama de dades rebuda per paràmetre. Es retorna el valor del CRC en format 16-bits.                                                                                                                                                                                                                            |
| <b><i>obtenirCRC</i></b>         | -                                                          | Es converteix el CRC format 16-bits en el format 8-bits, per tal de representar-lo mitjançant dos bytes hexadecimals. Es desplaça el byte menys significatiu 8 posicions a la dreta per a canviar-ne l'ordre, tal i com s'indica en el <i>datasheet</i> .                                                                                             |

|                    |                                  |                                                                                                                                                                                                                           |
|--------------------|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>calcularCRC</i> | <i>i2cframe:</i><br><i>bytes</i> | Finalment, per a cada byte de la trama I2C, s'invoca a la funció <i>actualitzarCRC</i> per tal d'anar restablint i calculant el valor. Finalment, es retorna el valor dels dos bytes que formen el CRC en format decimal. |
|--------------------|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Taula 6.11. Resum explicació mètodes de la classe *tagM24SR()*.

Els mètodes descrits es poden trobar en l'apartat 11.3.1 del annexes.

Un cop detallats els mètodes de la classe, es pot observar a continuació, un diagrama de flux que mostra el funcionament dels mètodes descrits anteriorment, des de que s'obtenen les dades dels dos sensors, fins que s'envien a través de la interfície I2C.

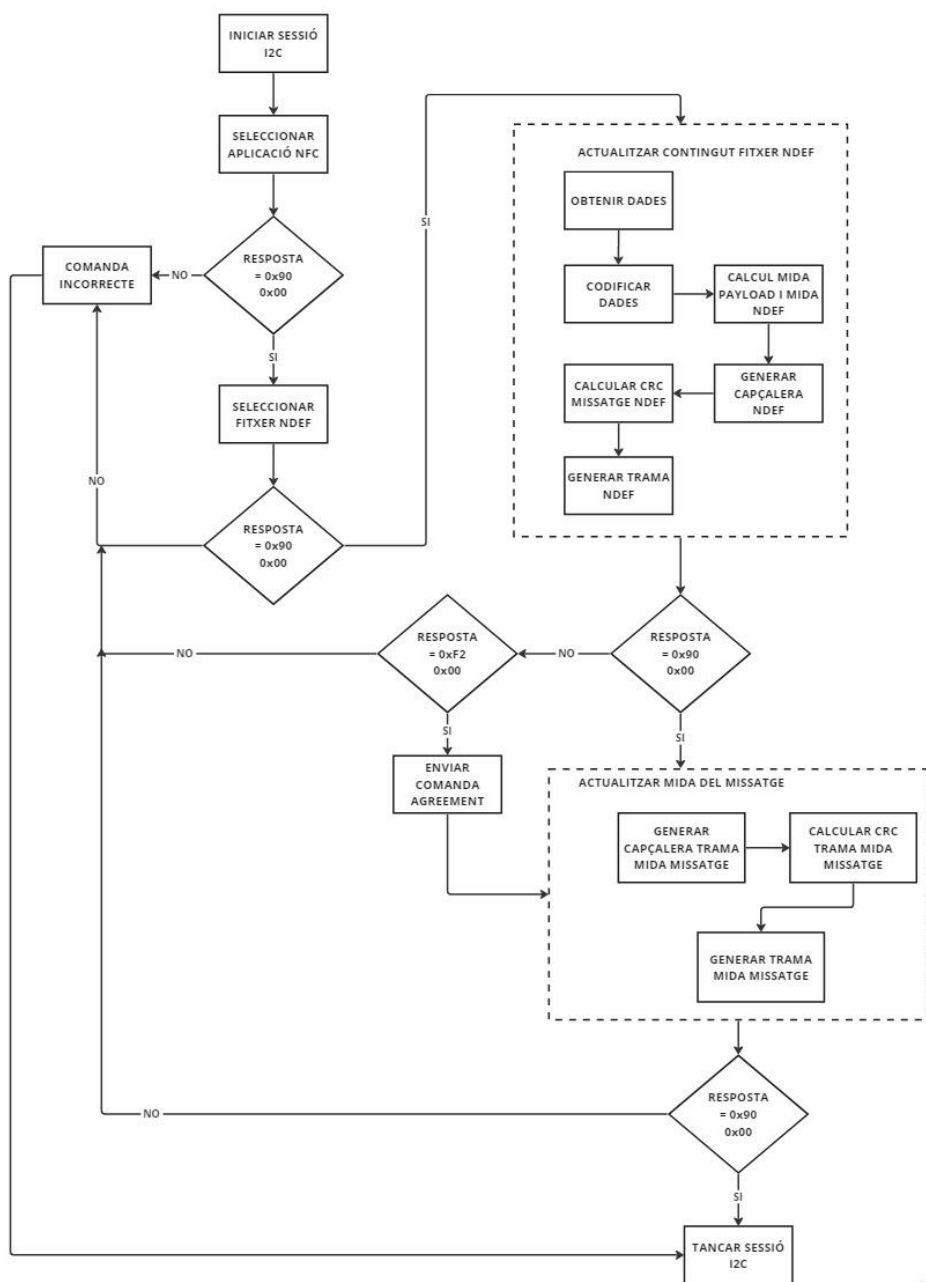
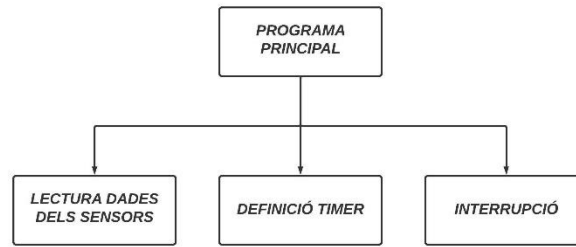


Figura 6.28. Diagrama de flux del programa.

### 6.6.4. Programa principal.

Finalment, el programa principal està format per les següents parts, definides en la *Figura 6.23*.



**Figura 6.29.** Identificació de les parts del programa principal.

Per una banda, es realitza contínuament la lectura de les dades dels sensors, s’aplica el filtre comentat en l’apartat 6.5.2 per tal de millorar els efectes de soroll, i s’actualitzen els valors màxim i mínim que permetran el posterior càlcul de la resposta. A més a més, es defineixen alguns mètodes que permeten convertir el valor de voltatge a resistència i calcular la resposta dels sensors en funció de la variació de la resistència, per tal de poder determinar la quantitat de concentració. Així mateix, també es defineixen mètodes per a obtenir l’hora del RTC i concatenar les dades a enviar. En la *Taula 6.12* es pot observar un resum d’aquests.

| Mètodes auxiliars lectura de dades |                                                            |                                                                                                                                                                                          |
|------------------------------------|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nom                                | Paràmetres d’entrada                                       | Descripció                                                                                                                                                                               |
| <i>voltatgeAresis</i>              | <i>V1: float</i><br><i>V2: float</i>                       | Es rep per paràmetre els valors de tensió de sortida del circuit de condicionament i es calcula el valor de resistència corresponent.                                                    |
| <i>comprovarConcentracions</i>     | <i>Resp: float</i>                                         | Es rep per paràmetre el valor de la resposta, calculat en la rutina d’interrupció, i es comprova a quin rang de concentració de NO <sub>2</sub> correspon el valor de resposta calculat. |
| <i>comprovarRang</i>               | <i>Resp: float</i><br><i>v1: float</i><br><i>v2: float</i> | Permet comprovar si el valor de resposta obtingut correspon a algun dels rangs establerts per aquest projecte i exposats en el l’apartat 4.                                              |
| <i>convertirTemps</i>              | <i>temps: RTC[6]</i>                                       | S’obtenen els camps corresponents a l’hora, minuts i segons de la variable <i>temps</i> corresponen al vector de dades obtingut del mòdul RTC.                                           |
| <i>convertirDades</i>              | <i>dada1: int</i>                                          | Es reben per paràmetre els valors corresponents a la concentració ( <i>dada1</i> ) i l’hora de mesura ( <i>dada3</i> ). Es concatenen i                                                  |

|  |                   |                                                                                  |
|--|-------------------|----------------------------------------------------------------------------------|
|  | <i>dada3: int</i> | es retorna el resultat, el qual representa la dada a escriure al <i>tag</i> NFC. |
|--|-------------------|----------------------------------------------------------------------------------|

Taula 6.12. Mètodes auxiliars per a la lectura de dades.

Per altra banda, degut a que la resposta dels sensors no és instantània, s'ha decidit utilitzar un *Timer* que permet executar la rutina d'escriptura de les dades al *tag* NFC cada  $x$  temps.

En relació a la definició del *Timer*, s'ha utilitzat la classe ja definida per a poder programar una interrupció periòdica cada  $x$  mil·lisegons. A continuació, en la *Figura 6.24*, es pot observar un exemple de la seva implementació.

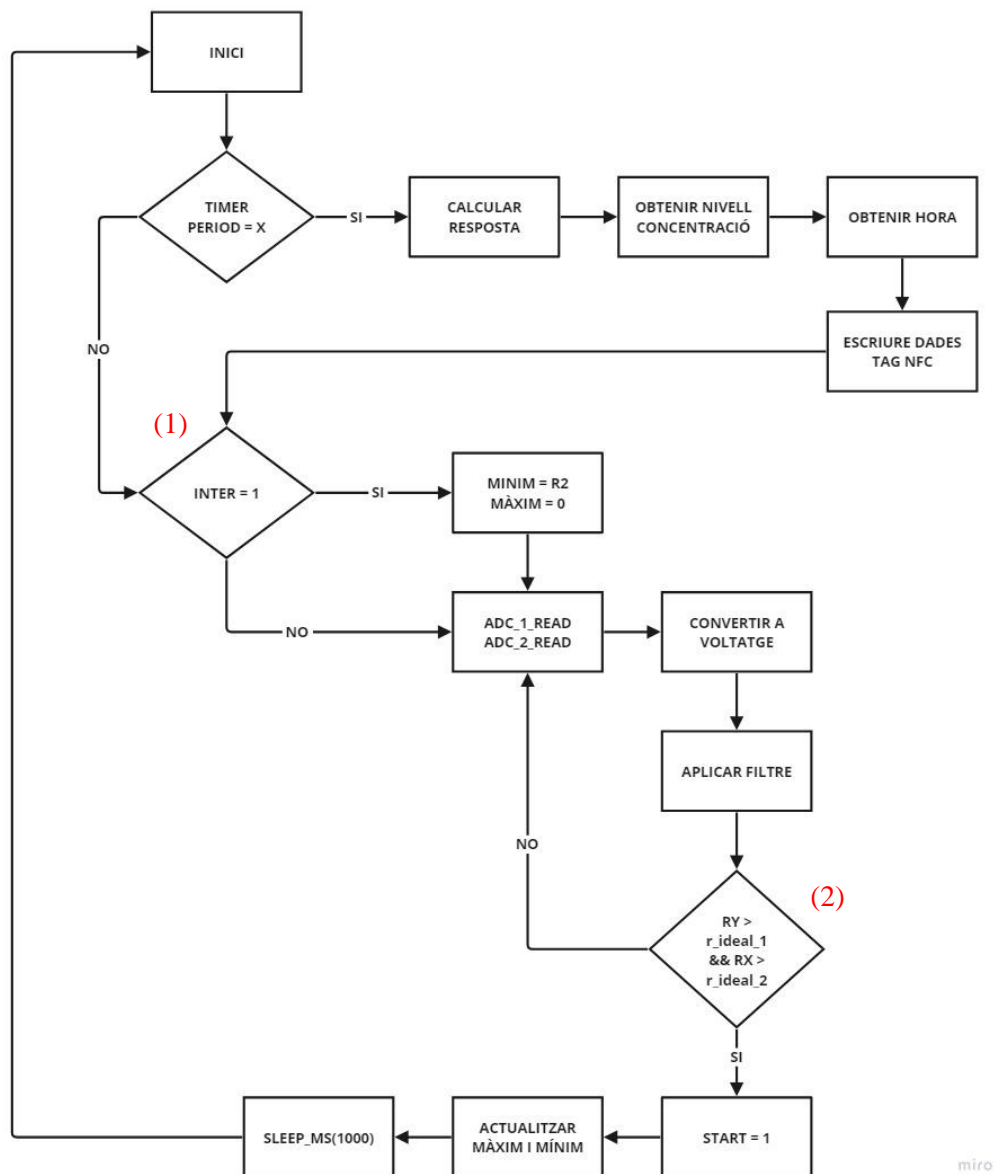
```
timer = Timer(period=300000,mode = Timer.PERIODIC,callback = interrupcio)
```

Figura 6.30. Exemple de la definició del *Timer*.

Tal i com es pot observar, es disposen de tres paràmetres en la instància del *Timer*. En el paràmetre *period*, es defineix el període que identifica cada quant s'executarà la funció d'interrupció. El paràmetre *mode*, permet determinar el tipus de *Timer*. En el cas del present projecte, s'ha configurat de forma que es vagi executant de forma periòdica. Finalment, en el paràmetre *callback*, es defineix la funció que s'executarà cada vegada que s'arriba al valor definit en el camp *period*.

Altrament, en la rutina d'interrupció que gestiona l'escriptura de les dades a l'etiqueta NFC, es realitza el càlcul de la resposta i es determina el nivell de concentració segons el valor de resposta obtingut. Aquest procediment es realitza fent ús de les funcions exposades en la *Taula 6.12* i, concretament, mitjançant les funcions *comprovarRang* i *comprovarConcentracions*. El valor que retorna aquesta última funció, correspon al nivell de concentració, i per tant, serà un dels valors a escriure al *tag* NFC. L'altre valor que s'escriu a l'etiqueta NFC, correspon a l'hora, els minuts i els segons en què s'ha realitzat la mesura. Per això, s'utilitza el mòdul *RTC (Real Time Clock)*, integrat en la *Raspberry Pi Pico*. En última instància, es concatenen aquestes dues dades i es passen per paràmetre a la funció *missatgeNDEF*, que s'encarregarà de codificar la informació per a la seva posterior transmissió.

Finalment, podem observar en la *Figura 6.25*, el diagrama de flux que representa el funcionament del programa principal comentat anteriorment.



**Figura 6.31.** Diagrama de flux del programa.

Les condicions (1) i (2) anotades en el diagrama de flux de la *Figura 6.25*, corresponen a la variable de control per a reiniciar els valors màxim i mínim utilitzats per al càlcul de la resposta (1), i el temps d'espera per a obtenir valors de resistència correctes un cop s'aplica el filtre (2).

Finalment, el codi corresponent al programa principal descrit anteriorment es pot observar en l'*apartat 11.3.2* dels annexes.

## 7. APLICACIÓ MÒBIL PER A GESTIONAR I VISUALITZAR LES DADES

En aquest apartat, es presentarà breument l'entorn de desenvolupament *Android Studio* utilitzat per al desenvolupament de l'aplicació. A més, s'exposarà sumàriament, els passos i modificacions prèvies a tenir en compte en el desenvolupament de l'aplicació. Per altra banda, es detallarà l'aplicació realitzada per a poder observar i emmagatzemar les dades dels sensors, llegides del *tag* NFC mitjançant el dispositiu mòbil.

### 7.1. Introducció a *Android Studio*.

*Android Studio* és un entorn de desenvolupament integrat que s'utilitza per a implementar aplicacions per a dispositius *Android*. Al crear un nou projecte, es crea un mòdul anomenat *app*, el qual conté diferents carpetes que formen la base de l'aplicació [26].

| Mòdul <i>App</i> |                                                                                                                                                                                                                           |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Carpetes         | Descripció                                                                                                                                                                                                                |
| <i>Manifest</i>  | Especifica la informació essencial de l'aplicació, el qual conté les eines i elements principals de creació d' <i>Android</i> [26].                                                                                       |
| <i>Java</i>      | Conté els principals arxius de codi font de l'aplicació. Si l'aplicació es desenvolupa en <i>Kotlin</i> , els arxius <i>.kt</i> també s'incorporaran en aquesta carpeta [26].                                             |
| <i>Res</i>       | Engloba tots aquells arxius que no corresponen al codi de l'aplicació [26]. Així doncs, en aquesta carpeta es poden emmagatzemar recursos gràfics i els arxius <i>.xml</i> corresponents al disseny dels <i>layouts</i> . |

Taula 7.1. Components mòdul *app* de l'aplicació.

Pel que fa al format de l'aplicació, aquesta està constituïda bàsicament per *activities*, les quals identifiquen les diferents pantalles de l'aplicació. Una *activity* està formada per dos components principals. Per una banda, es disposa d'una classe que permet l'escriptura del codi que identifica les diferents accions a realitzar per l'aplicació. Per altra banda, es proporciona un arxiu *.xml* corresponent al *layout* o disseny de l'*activity*.

Un altre paràmetre important a tenir en compte a l'hora d'entendre l'estructura i el funcionament d'una *activity*, és el seu cicle de vida. Aquest, comprèn un conjunt de mètodes que descriuen l'estat de l'aplicació i es van executant en funció d'aquest. En la Taula 7.2, es pot observar la característica descrita.

| Cicle de vida d'una <i>activity</i> |                                                                                                                                                                                                       |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nom                                 | Descripció                                                                                                                                                                                            |
| <i>onCreate</i>                     | És el primer mètode que s'executa a l'hora de crear-se l' <i>activity</i> . En aquest mètode, és on s'associen els diferents paràmetres corresponents al <i>layout</i> i disseny de l'aplicació [27]. |
| <i>onStart</i>                      | Correspon al procés d'inicialització de l'aplicació un cop s'han creat les vistes. És l'estat transitori entre la inicialització i l'execució de l'aplicació en primer pla [27].                      |
| <i>onResume</i>                     | Un cop l' <i>activity</i> ja s'ha creat i inicialitzat, aquest mètode és el que permet interactuar amb l'usuari. Qualsevol acció sobre l' <i>activity</i> es realitzarà en aquest estat [27].         |
| <i>onPause</i>                      | Descriu les accions realitzades quan s'interromp l'execució en primer pla de l' <i>activity</i> [27].                                                                                                 |
| <i>onStop</i>                       | Quan es navega cap a una altra pantalla, l' <i>activity</i> anterior passa a executar aquest mètode per tal d'estalviar recursos [27].                                                                |
| <i>onDestroy</i>                    | Un cop finalitzada la interacció amb l' <i>activity</i> , s'invoca aquest mètode que permet la destrucció i alliberació dels recursos usats per aquella <i>activity</i> [27].                         |

Taula 7.2. Descripció del cicle de vida d'una *activity*.

Un cop estudiats els principals components que permeten comprendre els conceptes bàsics de l'estructura d'*Android Studio* i la programació de dispositius mòbils, en el següent punt es tractaran les modificacions aplicades a l'arxiu font de l'aplicació per tal de poder activar les comunicacions NFC i el seguiment de la ubicació en el nostre dispositiu.

## 7.2. Consideracions prèvies.

En aquest apartat, s'exposaran les consideracions i modificacions prèvies necessàries per proporcionar a l'aplicació, les opcions de lectura de dades a través de NFC, així com l'obtenció de la ubicació del dispositiu mòbil. Totes les modificacions exposades a continuació, es realitzen sobre el fitxer *AndroidManifest.xml*, contingut en la carpeta *manifest* del programa.

En la *Figura 7.1*, s'observa la sentència necessària per tal de poder habilitar l'ús del hardware NFC per part de l'aplicació [28].

```
<uses-permission android:name="android.permission.NFC" />
```

Figura 7.1. Sol·licitud de permisos per fer ús de NFC.

Un cop autoritzat l'accés per a l'ús de NFC, s'ha d'especificar el tipus d'etiqueta utilitzada i el format de la informació que es pretén intercanviar. En el cas present, s'indica únicament

que es vol rebre informació de *tags* que usin el protocol NDEF per a l'intercanvi de missatges de text [28]. En la *Figura 7.2* es pot observar l'especificació comentada.

```
<intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <action android:name="android.nfc.action.NDEF_DISCOVERED" />

  <data android:mimeType="text/plain" />
</intent-filter>
```

**Figura 7.2.** Especificació del format dels missatges intercanviats i la tecnologia empleada.

Adicionalment, per tal d'habilitar les funcionalitats per a obtenir la ubicació del dispositiu, s'afegeixen les següents sentències observades en la *Figura 7.3* [29].

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

**Figura 7.3.** Permisos per a obtenir la ubicació del dispositiu.

El permís *ACCES\_FINE\_LOCATION* permet adquirir la ubicació de manera precisa [29]. Contràriament, el *ACCES\_COARSE\_LOCATION* obté la ubicació aproximada del dispositiu [29]. S'implementen les dues opcions per tal de poder mostrar una ubicació aproximada en cas que aquesta no es pugui mostrar de forma precisa.

Un cop modificats els elements comentats anteriorment, l'aplicació ja permet la interacció amb altres dispositius NFC, així com la obtenció de la ubicació del dispositiu.

### 7.3. Desenvolupament de l'aplicació.

A continuació, es presentarà el desenvolupament de l'aplicació on s'identificarà la seva estructura principal i el seu funcionament, així com les *Activities* desenvolupades, juntament amb els mètodes implementats per a la correcta lectura de les dades a través de NFC i la seva visualització.

#### 7.3.1. Estructura de l'aplicació i funcionament.

L'estructura bàsica de l'aplicació la podem representar segons el diagrama observat en la *Figura 7.1*.



**Figura 7.4.** Estructura de l'aplicació.

Per una banda, l'aplicació ens permet obtenir les dades de la concentració de NO<sub>2</sub> que ha sigut emmagatzemada en l'etiqueta NFC. A més del valor de concentració, també es mostra informació sobre la mesura, com ara el dia, el mes i l'hora exacta en que s'ha realitzat la detecció, així com la ubicació on ens trobem en el moment en que obtenim les dades.

D'altra banda, s'ha afegit una funcionalitat que permet la persistència o emmagatzematge de la informació obtinguda mitjançant la implementació d'una base de dades en la pròpia aplicació, fent ús de la llibreria *Room* [30]. D'aquesta manera, les dades poden persistir al programari encara que l'usuari surti o tanqui l'aplicació.

Finalment, de la informació que l'usuari ha emmagatzemat a la base de dades, s'ha afegit una vista que permet obtenir les dades referents a la concentració de NO<sub>2</sub> de cada un dels sensors i representar-ne els resultats en un gràfic. D'aquesta manera, es poden analitzar de forma visual, els diferents valors de concentració obtinguts.

Seguidament, podem observar en la *Figura 7.5*, un diagrama de flux que representa el funcionament ideal del programari desenvolupat. A continuació d'aquest, es detallaran les funcionalitats referents a l'estructura de l'aplicació, exposades en la *Figura 7.4*.

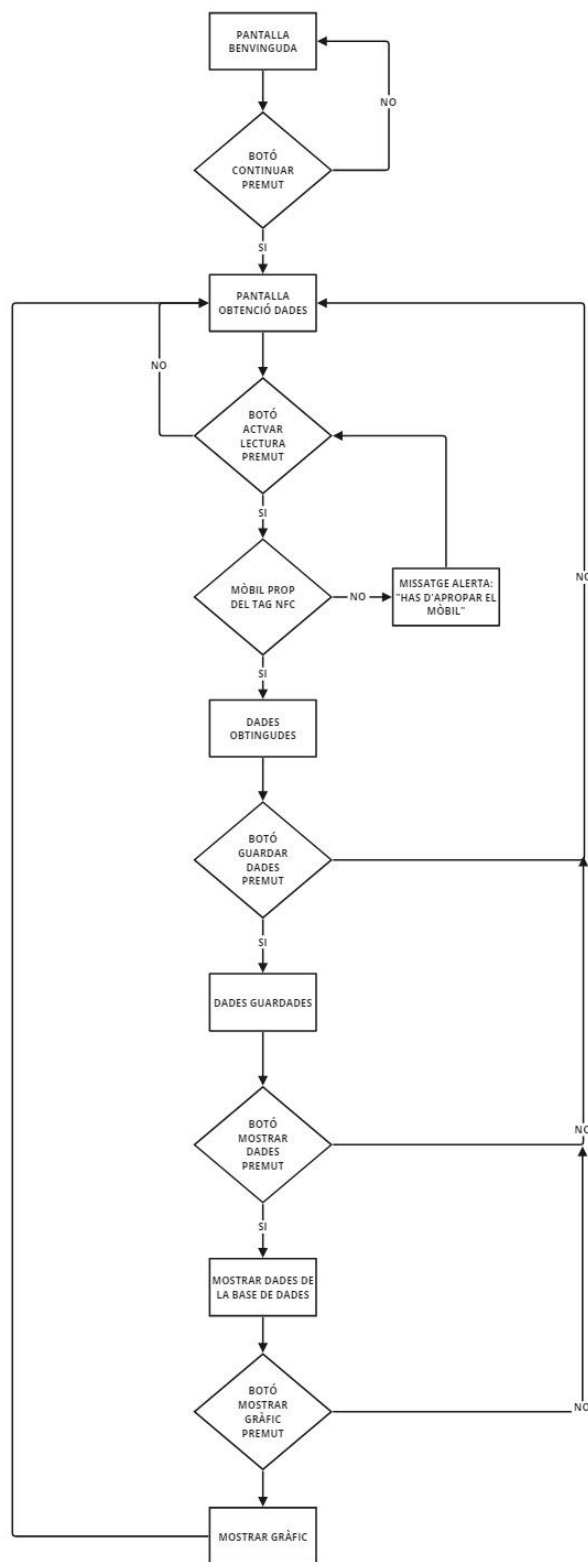

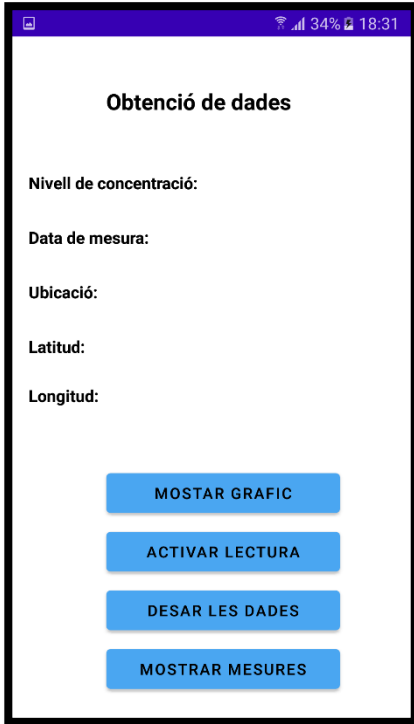


Figura 7.5. Diagrama de flux representant l'execució ideal de l'aplicació.

Al iniciar l'aplicació, es mostra una pantalla simple de benvinguda on s'indica el títol de l'aplicació, una imatge representativa de la tecnologia NFC i un botó que permet a l'usuari navegar cap a la següent pantalla.

Un cop s'ha premut el botó continuar, es mostra la pantalla principal de l'aplicació, on es poden realitzar les diferents funcionalitats introduïdes anteriorment.

A continuació, en la *Taula 7.2* es poden observar les vistes anteriorment exposades.

| Pantalla inicial benvinguda                                                                                                                                                | Pantalla principal amb funcionalitats                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p data-bbox="320 1240 724 1267"><b>Figura 7.6.</b> Pantalla inicial de l'aplicació</p> |  <p data-bbox="895 1240 1299 1267"><b>Figura 7.7.</b> Pantalla principal de l'aplicació</p> |

**Taula 7.2.** Vistes inicial i principal de l'aplicació.

El botó *Mostrar Gràfic* permetrà a l'usuari visualitzar la dada corresponent al camp *Nivell de concentració* de la *Figura 7.7*, la qual fa referència a la concentració de NO<sub>2</sub> captada pels sensors. Per a poder visualitzar les dades, prèviament s'han d'haver emmagatzemat a la base de dades. D'altra banda, si la base de dades està buida, no s'observarà cap valor, i la pantalla de visualització del gràfic contindrà un missatge anunciant que no es disposa de dades per a representar. Aquesta funció es pot consultar sense necessitat de realitzar cap lectura de dades prèvia.

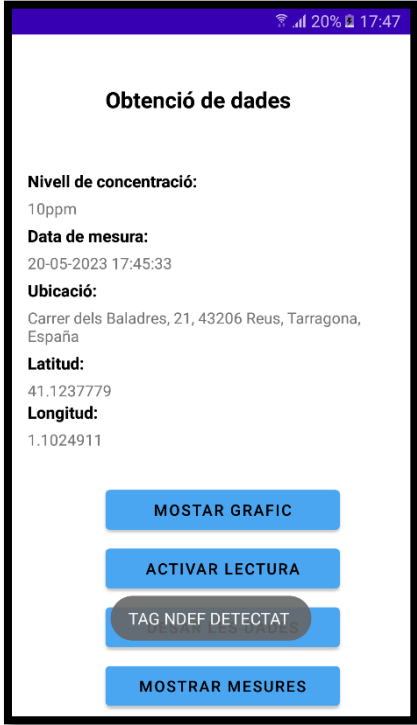
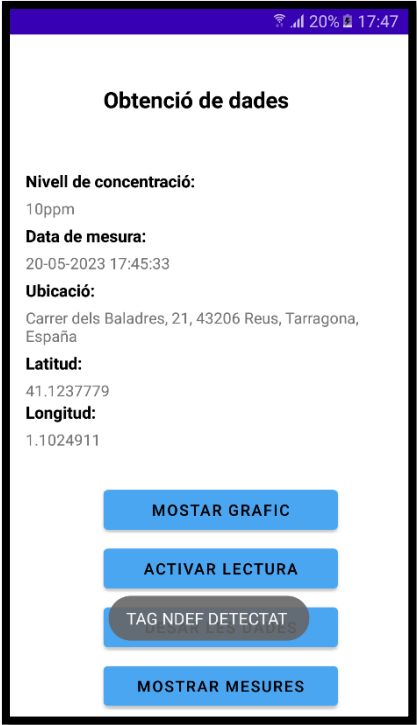
Seguidament, prement el botó *Activar Lectura*, s'habilitarà la funcionalitat NFC del dispositiu mòbil, permetent la lectura de les dades apropant aquest a una distància d'uns 3 cm del *tag*. Al llegir les dades, automàticament s'ompliran els camps referents a la *Data de mesura*, *Ubicació*, *Latitud* i *Longitud*. El valor *Data de mesura* correspon al dia i hora en que el sensor ha detectat la concentració de gas, mentre que la resta de camps corresponen al moment en que s'ha realitzat la lectura amb el dispositiu mòbil. Mentre aquest botó no

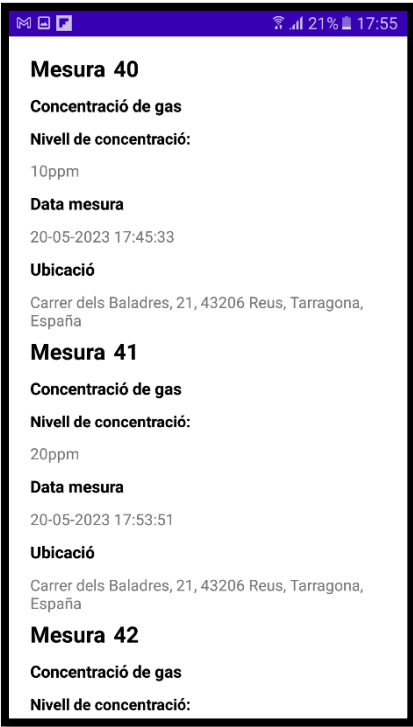
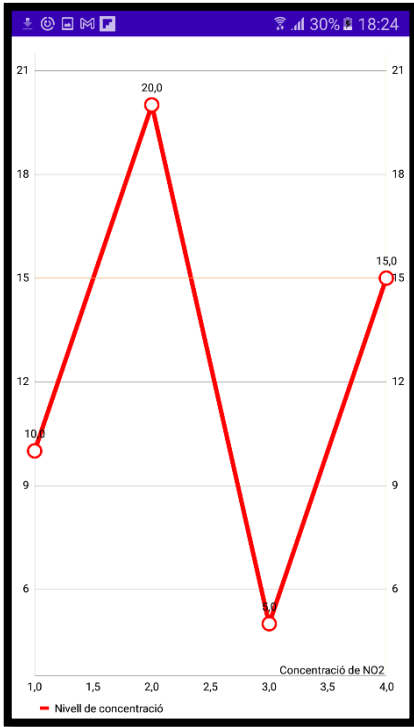
estigui polsat, no es podran llegir les dades de l'etiqueta NFC, mantenint també la resta de camps buits.

El botó *Desar les Dades* permetrà emmagatzemar les dades, incloent-hi la data i la ubicació, a la base de dades. Degut a que es pot donar el cas en què l'usuari premi el botó *Desar les Dades* sense haver iniciat la lectura, aquesta funcionalitat només és operativa quan prèviament s'ha premut el botó *Activar Lectura* i els camps obtinguts no estan buits. D'aquesta manera evitem l'emmagatzematge de valors erronis.

Finalment, el botó *Mostrar Mesures* permet obrir una nova vista i mostrar el conjunt de les dades emmagatzemades en una llista. Aquesta funció sí que està disponible tot i no haver activat la funció per llegir les dades mitjançant NFC, així l'usuari pot consultar les dades que han sigut prèviament captades i emmagatzemades. A més a més, aquesta llista inclou una opció per a esborrar les dades que l'usuari desitja.

Tot seguit, es pot observar en la *Taula 7.3*, un conjunt d'imatges que exposen de forma visual, les explicacions anteriors. Els resultats representats no són reals, són proves del funcionament del sistema de comunicació en general i, concretament, de l'aplicació mòbil. En l'*apartat 8.2* es detallen els resultats obtinguts amb dades reals.

| Exemple <i>Activar Lectura</i> i obtenir dades                                                                                               | Exemple <i>Desar les Dades</i>                                                                                                                      |
|----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p><b>Figura 7.8.</b> Exemple obtenció de les dades.</p> |  <p><b>Figura 7.9.</b> Exemple emmagatzematge de les dades.</p> |

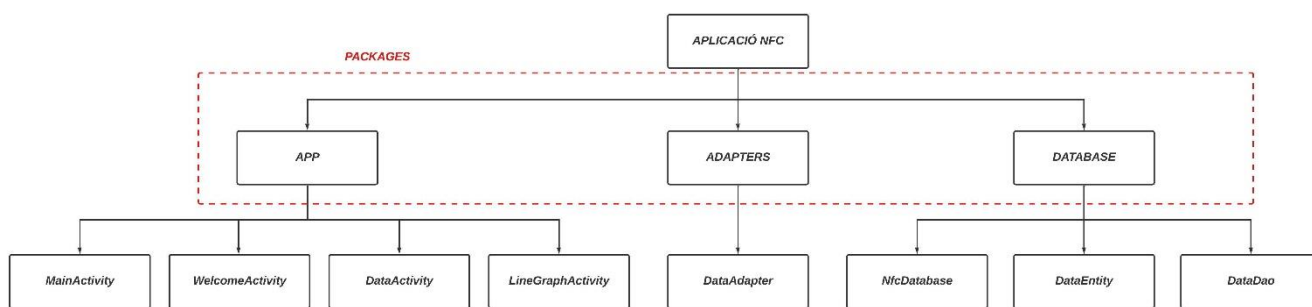
| Exemple <i>Mostrar Mesures</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Exemple <i>Mostrar Gràfic</i>                                                                                                                         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p><b>Mesura 40</b><br/>                 Concentració de gas<br/>                 Nivell de concentració:<br/>                 10ppm<br/>                 Data mesura<br/>                 20-05-2023 17:45:33<br/>                 Ubicació<br/>                 Carrer dels Baladres, 21, 43206 Reus, Tarragona, España</p> <p><b>Mesura 41</b><br/>                 Concentració de gas<br/>                 Nivell de concentració:<br/>                 20ppm<br/>                 Data mesura<br/>                 20-05-2023 17:53:51<br/>                 Ubicació<br/>                 Carrer dels Baladres, 21, 43206 Reus, Tarragona, España</p> <p><b>Mesura 42</b><br/>                 Concentració de gas<br/>                 Nivell de concentració:</p> |  <p>Figura 7.11. Exemple gràfic de les mesures emmagatzemades.</p> |

Taula 7.3. Exemples de les diferents funcionalitats de l'aplicació.

### 7.3.2. Descripció de la implementació.

Per a la elaboració de l'aplicació, s'han desenvolupat les següents classes i *activities* observades en la Figura 7.12, les quals permeten la implementació de les diferents funcionalitats exposades anteriorment. Així mateix, aquestes s'han organitzat en diferents *packages* o carpetes, per tal d'establir una diferenciació d'usos, tot mantenint un ordre en la programació. Dit això, els tres *packages* principals són:

- *Package App*. Conté totes les quatre *activities* principals de l'aplicació, les quals permeten identificar cada una de les vistes observades en el punt anterior.
- *Package Adapters*. Disposa d'una única classe, la qual identifica la vista observada en la Figura 7.10, permetent la visualització de les dades en una llista.
- *Package Database*. Engloba aquells fitxers que permeten la construcció i interacció amb la base de dades. En aquest cas, podem identificar dues classes i una interfície.



**Figura 7.12.** Representació de l'organització de l'aplicació.

Tot seguit, es presentaran les variables i mètodes definits en cada una de les diferents *activities* i classes introduïdes al començament d'aquest apartat, organitzades segons els respectius *packages*.

### **Package App**

a) *MainActivity*. Caracteritza la pantalla principal de l'aplicació, segons podem observar en la *Figura 7.7*. En la *Taula 7.3*, es representen el conjunt de variables que conté, juntament amb una descripció de la seva principal funció o utilitat.

| <b>Variables <i>MainActivity</i></b> |                                    |                                                                                                                                                                                                                                          |
|--------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Nom</b>                           | <b>Tipus</b>                       | <b>Descripció</b>                                                                                                                                                                                                                        |
| <i>fusedLocationProviderClient</i>   | <i>FusedLocationProviderClient</i> | Identifica l'objecte <i>FusedLocationProviderClient</i> . Permet obtenir diferents mètodes per a identificar la ubicació del dispositiu [29].                                                                                            |
| <i>latitud</i>                       | <i>TextView</i>                    | Corresponen al conjunt de components de tipus <i>TextView</i> que permeten mostrar en la pantalla de l'aplicació, les informacions de latitud, longitud, el nivell de concentració mesurat, la data de mesura i la ubicació de l'usuari. |
| <i>longitud</i>                      |                                    |                                                                                                                                                                                                                                          |
| <i>sensor</i>                        |                                    |                                                                                                                                                                                                                                          |
| <i>data</i>                          |                                    |                                                                                                                                                                                                                                          |
| <i>ubicacio</i>                      |                                    |                                                                                                                                                                                                                                          |
| <i>locationRequest</i>               | <i>LocationRequest</i>             | Variable que identifica la classe <i>LocationRequest</i> i que permet obtenir diferents                                                                                                                                                  |

|                       |                    |                                                                                                                                                                                                                           |
|-----------------------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       |                    | característiques sobre la ubicació de l'usuari [29].                                                                                                                                                                      |
| <i>desarDades</i>     | <i>Button</i>      | Identifiquen els diferents botons de la pantalla principal de l'aplicació, tal i com es mostra en la <i>Figura 7.7</i> . Cada un d'ells, disposarà d'un element <i>listener</i> que permetrà realitzar diferents accions. |
| <i>activarLectura</i> |                    |                                                                                                                                                                                                                           |
| <i>mostrarDades</i>   |                    |                                                                                                                                                                                                                           |
| <i>mostrarGrafic</i>  |                    |                                                                                                                                                                                                                           |
| <i>activat</i>        | <i>Boolean</i>     | Variable que permet controlar que no s'insereixin valors buits a la base de dades, en el cas en que l'usuari premi el botó <i>desar dades</i> sense haver obtingut cap valor.                                             |
| <i>db</i>             | <i>NfcDatabase</i> | Identifica la classe <i>NfcDatabase</i> . Aquesta variable s'utilitza per a instanciar la base de dades i inserir-hi els valors obtinguts i que es volen fer persistents.                                                 |
| <i>nfcAdapter</i>     | <i>NfcAdapter</i>  | Distingeix la classe <i>NfcAdapter</i> que permet proporcionar accés a la funcionalitat NFC del dispositiu mòbil. Permet la lectura de missatges NDEF emmagatzemats en <i>tags</i> NFC [28].                              |

**Taula 7.3.** Descripció de les variables del *MainActivity*.

Un cop identificades les variables principals del programa, es prosseguirà explicant els principals mètodes implementats. Principalment, podem dividir l'explicació en tres blocs principals. Els mètodes referents a l'obtenció de les dades a través de NFC, els que identifiquen l'estructura i cicle de vida d'una *activity*, i els associats a l'obtenció de la ubicació. Per a la implementació d'aquests últims mencionats, s'ha fet ús de la referència [26].

Pel que fa als mètodes referents a l'obtenció de dades NFC, es resumirà el nom i els paràmetres d'entrada i sortida en una taula. Posteriorment, es realitzarà una descripció més detallada de cada un d'ells.

Respecte als mètodes que identifiquen el cicle de vida d'una *activity*, es resumirà, en una taula, les principals crides i inicialitzacions en cada una d'ells. D'altra banda, fent referència als mètodes que permeten obtenir la ubicació, es resumirà el seu funcionament en una taula. El codi corresponent a l'*activity* completa es pot observar en l'*apartat 11.4.1* dels annexos.

Dit això, en la *Taula 7.4*, la *Taula 7.5* i la *Taula 7.6*, podem identificar cada una de les petites descripcions dels tres blocs mencionats anteriorment.

| Mètodes NFC                  |                       |                          |
|------------------------------|-----------------------|--------------------------|
| Nom                          | Paràmetres d'entrada  | Paràmetres de sortida    |
| <i>processarMissatgeNdef</i> | <i>intent: Intent</i> | -                        |
| <i>separarRecord</i>         | <i>record: String</i> | <i>contingut: String</i> |
| <i>convertirHexaAscii</i>    | <i>sep: String</i>    | <i>missatge: String</i>  |

Taula 7.4. Mètodes per a obtenir les dades dels sensors mitjançant NFC.

A continuació, s'exposaran els principals mètodes descrits en la *Taula 7.4*.

El mètode *processarMissatgeNdef* permet obtenir i processar les dades llegides del *tag* NFC. Primerament, es comprova que al apropar el mòbil a l'etiqueta, aquesta correspon al tipus NDEF. En cas contrari, si l'etiqueta no és de tipus NDEF, es mostra un missatge emergent per pantalla indicant que l'aplicació no pot llegir-ne el contingut. En cas afirmatiu, s'emmagatzema el missatge en un *array* de tipus *NdefMessage*. Seguidament, degut a que només s'envia un *record* o missatge, s'obté la dada emmagatzemada i es converteix a *string*. A posteriori, aquesta dada serà processada pels mètodes *convertirHexaAscii* i *separarRecord* per tal de visualitzar de forma entenedora el contingut del missatge.

```
private fun processarMissatgeNdef(intent: Intent) {
    val parcelabled = intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES)

    if (parcelabled != null) {
        //Indiquem que s'ha detectat una etiqueta NDEF
        Toast.makeText(
            applicationContext, text: "TAG NDEF DETECTAT ",
            Toast.LENGTH_LONG
        ).show()
        //Creem un array per emmagatzemar els missatges NDEF que obtenim.
        val ndefmsg = arrayOfNulls<NdefMessage>(parcelabled.size)

        ndefmsg[0] = parcelabled[0] as NdefMessage
        // Obtenir les dades del primer record NDEF.
        Log.i( tag: "NDEF Intent process", ndefmsg[0]!!.records[0].toString())
        val mis = ndefmsg[0]!!.records[0].toString()
        val sep: String = separarRecord(mis)
        val miss: String = convertirHexaAscii(sep)
        val valor = miss.split( ...delimiters: " ")
        Log.i( tag: "NDEF MISSATGE", sep)
        //missatge.setText(ndefmsg[0].getRecords()[i].getPayload().toString());
        val simpleDateFormat = SimpleDateFormat( pattern: "dd-MM-yyyy " )
        val dades_data = simpleDateFormat.format(Date())

        sensor1.text = valor[0]
        data.text = dades_data + valor[1]
    }
}
```

Figura 7.13. Exemple mètode *processarMissatgeNdef*.

Degut a que el missatge rebut per l'aplicació disposa del format observat en la *Figura 7.11*, és necessari separar el camp *payload*, que és el que conté les dades, de la resta de camps que únicament ens donen informació del tipus de missatge enviat.

```
I/NDEF Intent process: NdefRecord tnf=1 type=54 payload=02656E313570706D2031373A35353A3232
```

**Figura 7.11.** Exemple missatge NDEF rebut.

Així doncs, amb el mètode *separarRecord* podem obtenir i diferenciar la part que correspon al *payload* o missatge NDEF, de la resta de camps. D'aquesta manera, es separa la trama per tal de processar únicament el camp *payload* que conté les dades enviades. Per tant, es converteix el missatge rebut a *string* i es separa mitjançant el mètode *split*. Un cop obtinguda la trama, aquesta es retorna per tal de poder ser processada pel mètode *convertirHexAscii*.

```
private fun separarRecord(record: String):String{
    var missatge = ""
    var contingut = ""

    val separar = record.split(" ".toRegex()).toTypedArray()

    for (i in separar.indices) {
        missatge = if (separar[i].contains( other: "payload")) {
            separar[i]
        } else {
            " "
        }
    }
    if (missatge != " ") {
        val getPayload = missatge.split("=".toRegex()).toTypedArray()
        for (j in getPayload.indices) {
            if (getPayload[j] != "payload") {
                contingut = getPayload[j]
            }
        }
    }

    return contingut
}
```

**Figura 7.14.** Exemple mètode *separarRecord*.

Finalment, el mètode *convertirHexAscii* permet descodificar la trama hexadecimal que correspon al missatge enviat, obtinguda pel mètode exposat anteriorment.

Degut a que els tres primers bytes corresponen a la capçalera del *payload*, es comença a descodificar a partir de la sisena posició de la trama de dades, corresponent al tercer byte. Mitjançant el mètode *toInt(16)*, podem obtenir el valor decimal que identifica el nombre hexadecimal contingut en la trama. Seguidament, el nombre obtingut es converteix a caràcter i s'emmagatzema en un *array* de tipus *string*. A mesura que es van descodificant i afegint caràcters al *array*, es va formant el missatge. Finalment, es retorna aquest valor.

```
private fun convertirHexaAscii(sep: String): String {
    var res: String
    val mess = arrayOfNulls<String>( size: sep.length / 2)
    var num: Int
    var missatge = ""
    var n = 6

    var i = 6
    while (i < sep.length) {
        res = "" + sep.get(i) + sep.get(i + 1)
        num = res.toInt( radix: 16)
        if (i - n > 0) {
            n++
            mess[i - n] = num.toChar().toString()
        } else {
            n = 6
            mess[i - n] = num.toChar().toString()
        }
        missatge += mess[i - n]
        i += 2
    }
    return missatge
}
```

Figura 7.15. Exemple mètode *convertirHexaAscii*.

A continuació, s'exposen els mètodes necessaris per a obtenir la ubicació del dispositiu mòbil. Aquests mètodes han estat desenvolupats fent ús de la referència [32].

| Mètodes obtenció ubicació |                          |                                                                                                                                                                                                                                |
|---------------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nom                       | Paràmetres de sortida    | Descripció                                                                                                                                                                                                                     |
| <i>ubicacioActual</i>     | -                        | Permet obtenir la ubicació actualitzada del dispositiu. Es comprova que l'usuari hagi acceptat el permís per a fer ús de la ubicació i que aquesta estigui activa. En cas contrari, es fa una petició per a la seva activació. |
| <i>comprovarPermisos</i>  | <i>resultat: Boolean</i> | Comprova els permisos d'ubicació que es demanen a l'usuari quan entra per primera vegada a l'aplicació. Retorna un booleà indicant el resultat de l'operació.                                                                  |
| <i>demanarPermisos</i>    | -                        | És el mètode que s'encarrega de la demanda dels permisos necessàries per a l'activació de la ubicació i que són controlats per el mètode <i>comprovarPermisos</i> .                                                            |
| <i>ubicacioActiva</i>     | <i>resultat: Boolean</i> | Permet validar si l'usuari té activada la ubicació en el seu dispositiu. Retorna un                                                                                                                                            |

|                       |                       |                                                                                                                                                                                                                                |
|-----------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       |                       | booleà indicant-ne el resultat. En cas negatiu, es mostra un missatge sol·licitant-la.                                                                                                                                         |
| <i>obtenirNovaUbi</i> | -                     | Permet actualitzar la ubicació del dispositiu. Es comproven els permisos i es fa la crida al mètode <i>requestLocationUpdates</i> de la classe <i>FusedLocationProviderClient</i> .                                            |
| <i>ubicacioAdreça</i> | <i>adress: String</i> | A partir de la latitud i la longitud obtingudes, es permet fer una aproximació de la ubicació de l'adreça que hi correspon. Es reben per paràmetre les dades de latitud i longitud i es retorna un <i>String</i> amb l'adreça. |

**Taula 7.5.** Mètodes per a obtenir la ubicació del dispositiu mòbil.

Seguidament, es pot observar la implementació dels mètodes d'obtenció de la ubicació i de processament del missatge obtingut a través de NFC, en els procediments propis del cicle de vida *d'android*.

| Implementacions en el cicle de vida |                                                                                                                                                                                                                                                                                                     |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nom                                 | Descripció                                                                                                                                                                                                                                                                                          |
| <i>onCreate</i>                     | Inicialització dels principals elements de disseny de l'aplicació, així com dels components <i>nfcAdapter</i> i <i>fusedLocationProviderClient</i> , creació de la base de dades i implementació dels botons que permeten navegar a les <i>activities DataActivity</i> i <i>LineGraphActivity</i> . |
| <i>onNewIntent</i>                  | Es fa la crida als mètodes per a l'obtenció de dades a través de NFC, així com els necessaris per adquirir la informació de la ubicació i del dia i hora de mesura.                                                                                                                                 |
| <i>onResume</i>                     | S'implementa la interacció dels botons per a desar les dades obtingudes a la base de dades i per a l'activació de la lectura NFC.                                                                                                                                                                   |
| <i>onPause</i>                      | Es desactiva la opció per a llegir dades amb NFC. D'aquesta manera, no s'obtenen les dades quan no s'està fent ús de l'aplicació.                                                                                                                                                                   |

**Taula 7.6.** Crides i implementacions dels mètodes anteriors en el cicle de vida d'una *activity*.

b) *WelcomeActivity*. Representa l'*activity* més simple de l'aplicació, la qual identifica la pantalla inicial segons podem observar en la *Figura 7.6* de l'*apartat 7.3.1*. Disposa d'una única variable la qual identifica el botó per a navegar cap a la següent pantalla. L'únic mètode que conté és el *listener* del botó, que permet iniciar el programa.

c) *DataActivity*. És l'*activity* usada per a l'obtenció de les dades que han sigut emmagatzemades en la base de dades. La informació és obtinguda en el moment de la creació de l'*activity*. Es pot observar un exemple visual en la *Figura 7.10* de l'*apartat 7.3.1*.

Principalment, disposa d'una única variable i un únic mètode, detallats en la *Taula 7.7*. Per al codi complet de *l'activity*, es pot consultar *l'apartat 11.4.2* dels annexos.

| <b>Variables <i>DataActivity</i></b> |                                                                                                                                                                                   |                                                                                                                  |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| <b>Nom</b>                           | <b>Tipus</b>                                                                                                                                                                      | <b>Descripció</b>                                                                                                |
| <i>recyclerView</i>                  | <i>RecyclerView</i>                                                                                                                                                               | Identifica la classe <i>RecyclerView</i> , que ens permet visualitzar les dades en una àmplia llista d'elements. |
| <b>Mètodes <i>DataActivity</i></b>   |                                                                                                                                                                                   |                                                                                                                  |
| <b>Nom</b>                           | <b>Descripció</b>                                                                                                                                                                 |                                                                                                                  |
| <i>obtenirDades</i>                  | Permet obtenir la informació emmagatzemada en la base de dades, a través de la interfície <i>DataDao</i> , i visualitzar-les en forma de llista fent ús del <i>RecyclerView</i> . |                                                                                                                  |

**Taula 7.7.** Variables i mètodes *DataActivity*.

d) *LineGraphActivity*. Ens permet visualitzar els valors de concentració que han sigut desats en la base de dades en forma de gràfic, visualitzant-ne la tendència. Si no hi ha dades emmagatzemades, es mostra un missatge al centre de *l'activity* indicant la manca de dades a representar.

En la *Figura 7.11* de *l'apartat 7.3.1* es pot observar un exemple visual de *l'activity*. La caracterització del gràfic s'ha realitzat fent ús de la llibreria ja existent, *MPAndroid*, detallada en la referència [32].

Tot seguit, podem veure l'explicació de les variables i mètodes principals en la *Taula 7.8* i *Taula 7.9*, respectivament.

| <b>Variables <i>LineGraphActivity</i></b> |                               |                                                                                                                                                                                                                              |
|-------------------------------------------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Nom</b>                                | <b>Tipus</b>                  | <b>Descripció</b>                                                                                                                                                                                                            |
| <i>arrayDadesSensor1</i>                  | <i>ArrayList&lt;Entry&gt;</i> | Per a cada un dels sensors, s'emmagatzemen les seves dades en un <i>array</i> de tipus <i>Entry</i> , que identifica els valors <i>x</i> i <i>y</i> , que es refereixen al número de dada i valor de mesura, respectivament. |
| <i>arrayDadesSensor2</i>                  |                               |                                                                                                                                                                                                                              |
| <i>arrayDadesDb</i>                       | <i>List&lt;DataEntity&gt;</i> | Llista on s'emmagatzemen tots els valors disponibles en la base de dades.                                                                                                                                                    |

|                           |                                     |                                                                                                                                                                                              |
|---------------------------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>lineChart</i>          | <i>LineChart</i>                    | Variable que identifica la classe <i>LineChart</i> . Permet l'accés a un conjunt de mètodes i variables definits en aquesta classe.                                                          |
| <i>numElements</i>        | <i>Int</i>                          | Identifica el nombre d'elements de la llista <i>arrayDadesDb</i> . S'utilitza per a controlar que aquesta no estigui buida.                                                                  |
| <i>lineDataSetSensor1</i> | <i>LineDataSet</i>                  | Correspon a la representació del valor del valor de concentració detectat pels sensors.                                                                                                      |
| <i>descriptionGrafic</i>  | <i>Description</i>                  | Permet afegir una descripció al gràfic.                                                                                                                                                      |
| <i>dataSetSensors</i>     | <i>ArrayList&lt;LineDataSet&gt;</i> | Emmagatzema <i>lineDataSet</i> que permeten la representació de diferents grups en un mateix gràfic. En el nostre cas, l' <i>array</i> està format únicament pel <i>lineDataSetSensor1</i> . |
| <i>llegenda</i>           | <i>Legend</i>                       | Permet afegir una llegenda al gràfic representat.                                                                                                                                            |

 Taula 7.8. Variables *LineGraphActivity*.

| <b>Mètodes <i>LineGraphActivity</i></b> |                             |                               |
|-----------------------------------------|-----------------------------|-------------------------------|
| <b>Nom</b>                              | <b>Paràmetres d'entrada</b> | <b>Paràmetres de sortida</b>  |
| <i>obtenirDadesGrafic</i>               | -                           | -                             |
| <i>separarDades</i>                     | <i>dada: String</i>         | <i>ArrayList&lt;Float&gt;</i> |

 Taula 7.9. Mètodes principals *LineGraphActivity*.

El mètode *obtenirDadesGrafic*, permet obtenir la llista de dades disponible emmagatzemada en la base de dades. Inicialment, s'obtenen els valors i s'emmagatzemen en la variable *arrayDadesDb*. Es comprova que la mida del *array* sigui diferent de zero, indicant que es disposa de dades. En cas contrari, es mostra un missatge informant que no hi ha valors emmagatzemats i finalitza l'execució del mètode. Per a cada entitat, s'obté el camp que identifica la mesura dels sensors i es separa del identificador *ppm*. Tot seguit, s'afegeix cada valor en el seu *arrayDadesSensor* corresponent i es formen els *lineDataSet* corresponent a les dada captada pels sensors. Finalment, les dades s'afegeixen al gràfic, així com una petita

llegenda i descripció, i es visualitza el resultat. A continuació, en la *Figura 7.14* i la *Figura 7.15*, es pot observar la implementació del mètode anteriorment descrit.

```
private fun obtenirDadesGrafic() {
    arrayDadesSensor1 = ArrayList<Entry>()
    val arrayX = ArrayList<String>()
    arrayDadesDb = NfcDatabase.getDataBase(context: this).dataDao.getValues()
    numElements = arrayDadesDb.size
    if(numElements != 0) {
        for (i in 0..numElements - 1) {
            Log.i(tag: "DATA FROM DATABASE:", arrayDadesDb[i].gasdata.toString())
            arrayX.add((i + 1).toString())
            val res = separarDades(arrayDadesDb[i].gasdata.toString())
            Log.i(tag: "DATA SEPARADA", res[0].toString())
            arrayDadesSensor1.add(Entry((i.toFloat() + 1), res[0]))
        }
    }
}
```

**Figura 7.16.** Exemple per obtenir els valors emmagatzemats en la base de dades.

```
lineDataSetSensor1 = LineDataSet(arrayDadesSensor1, label: "Nivell de concentració")
dataSetsSensors = ArrayList<LineDataSet>()
//Iniciolització de les variables que permeten personalitzar
//La descripció i la llegenda del gràfic
descriptionGrafic = Description()
llegenda = Legend()
//Custom Linedataset de les dades
lineDataSetSensor1.lineWidth = 4F
lineDataSetSensor1.circleRadius = 7F
lineDataSetSensor1.circleHoleRadius = 5F
lineDataSetSensor1.setCircleColor(Color.RED)
lineDataSetSensor1.valueTextSize = 10F
lineDataSetSensor1.color = Color.RED

//Descripció
descriptionGrafic.text = "Concentració de NO2"
descriptionGrafic.textSize = 10F

//Llegenda
llegenda = lineChart.legend
llegenda.form = Legend.LegendForm.LINE

lineChart.description = descriptionGrafic

dataSetsSensors.add(lineDataSetSensor1)
val dades = LineData(dataSetsSensors as List<ILineDataSet>)
lineChart.data = dades
//lineChart.invalidate()
val x: XAxis = lineChart.xAxis
val y: YAxis = lineChart.yAxis
y.setDrawGridLines(false)
y.setDrawAxisLine(false)
//x.valueFormatter = IAxisValueFormatter{_,_ -> arrayX[h]}
x.setDrawGridLines(false)

x.setPosition(XAxis.XAxisPosition.BOTTOM)
} else {
    lineChart.setNoDataText("No hi han dades")
}
```

**Figura 7.17.** Exemple construcció i representació del gràfic.

El mètode *separarDades* permet separar el valor numèric que representa la concentració de gas, de la seva unitat de mesura *ppm*. Es rep per paràmetre un *string*, format per el valor i les seves unitats. Es realitza la operació de separació i es converteix el valor a *float*, degut a que els valors de les entrades *x* i *y* en el moment de construir el gràfic així ho requereixen. Finalment, s'afegeix cada un dels valors en un *arrayList* de tipus *float* i es retorna el resultat. El resultat és processat per el mètode *obtenirDadesGràfic* anteriorment explicat.

```
private fun separarDades(dada:String):ArrayList<Float>{
    val resultat:ArrayList<Float> = ArrayList()
    val aux = dada.split( ...delimiters: " ")
    var identifica = false
    var chArr: CharArray
    var valor = " "
    for(i in aux.indices){
        try{
            chArr = aux[i].toCharArray()
            for(j in chArr.indices){
                if(chArr[j].isDigit()) {
                    valor += chArr[j]
                }
            }
            identifica = true
        }catch (e:NumberFormatException) {
            identifica = false
        }
        if(identifica){
            resultat.add(valor.toFloat())
            valor = " "
            //resultat.add(aux[i].toFloat())
        }
    }
    return resultat
}
```

Figura 7.18. Exemple mètode *convertirHexaAscii*.

### ***Package Adapter***

a) *DataAdapter*. Principalment, *l'adapter* permet vincular les dades que han sigut definides en la seva classe amb el *RecyclerView*, que es defineix com la manera eficient de mostrar grans conjunts de dades en format de llista. Així doncs, *l'adapter* implementat en l'aplicació corresponent al present projecte, permet obtenir la informació que ha sigut emmagatzemada en la base de dades i mostrar-la segons el format mencionat. El codi es pot observar en l'apartat 11.4.3 referent als annexos.

### ***Package Database***

a) *NfcDatabase*. És la classe de tipus *abstract* que identifica la base de dades *RoomDatabase* [30] de la nostra aplicació. Degut a que s'utilitza la llibreria *Room* per a la seva implementació, en la definició de la classe s'indica que hereta de *RoomDatabase()*. Conté una única variable de tipus *Dao*, per tal d'accedir a les dades, i un únic mètode anomenat *getDatabase* que permet crear o instanciar la base de dades, en cas que aquesta ja existeixi.

```

@Database(entities = [DataEntity::class], version = 1)
abstract class NfcDatabase:RoomDatabase() {
    abstract val dataDao:DataDao

    companion object{
        @Volatile private var INSTANCE: NfcDatabase? = null
        //si la base de dades ja ha sigut creada, es retorna la seva instància
        //en cas contrari, es crea.
        fun getDataBase(context: Context): NfcDatabase {
            val tempInstance = INSTANCE
            if(tempInstance != null){
                return tempInstance
            }
            synchronized( lock: this){
                val instance = Room.databaseBuilder(
                    context.applicationContext,NfcDatabase::class.java, name: "NFC_DATABASE"
                ).build()
                INSTANCE = instance
                return instance
            }
        }
    }
}
    
```

Figura 7.19. Classe NfcDatabase.

b) *DataEntity*. Identifica la classe que permet definir i representar els objectes que es volen emmagatzemar, per tal de concretar l'esquema de base de dades de l'aplicació. En el projecte realitzat, aquesta classe té els camps observats en la Figura 7.10.

```

@Entity(tableName = "gas_sensing_table")
data class DataEntity(
    @PrimaryKey(autoGenerate = true) val uuid: Int,
    @ColumnInfo(name = "gas_data") val gasdata: String?,
    @ColumnInfo(name = "measure_time") val timemeasure: String?,
    @ColumnInfo(name = "location_measure") val locationmeasure: String?
)
    
```

Figura 7.20. Classe DataEntity.

Cada un dels següents camps identifica els objectes contigus en l'aplicació, segons podem observar en la Taula 7.10.

| Classe DataEntity      |                                                                    |
|------------------------|--------------------------------------------------------------------|
| Nom                    | Descripció                                                         |
| <i>uuid</i>            | Correspon a l'identificador numèric de cada entitat emmagatzemada. |
| <i>gasdata</i>         | Fa referència al valor de concentració mesurat pels sensors.       |
| <i>timemeasure</i>     | Identifica l'hora i dia en que s'ha obtingut la mesura.            |
| <i>locationmeasure</i> | Determina la ubicació en format adreça.                            |

Taula 7.10. Descripció dels camps de la classe DataEntity.

c) *DataDao*. Els DAO, es defineixen com els objectes d'accés a dades, els quals implementen un conjunt de mètodes per tal d'accedir a les dades emmagatzemades en la base de dades [30]. S'usen mètodes amb sentències SQL per tal d'accedir-hi.

Un cop exposat el concepte d'utilitat d'aquesta classe, es pot observar la seva implementació en el present treball, segons es mostra en la *Figura 7.21*.

```

@Dao
interface DataDao {

    @Query( value: "SELECT * from gas_sensing_table ORDER BY uuid")
    fun getAll(): List<DataEntity>

    @Query( value: "SELECT * FROM gas_sensing_table")
    fun getValues():List<DataEntity>

    @Delete
    fun deleteData(data: DataEntity)

    @Query( value: "DELETE FROM gas_sensing_table")
    fun deleteAll()

    @Insert
    fun insertAll(data: DataEntity)

    @Query( value: "SELECT count(*)")
    fun nMeasures(): Int
}
    
```

**Figura 7.21.** Classe *DataDao*.

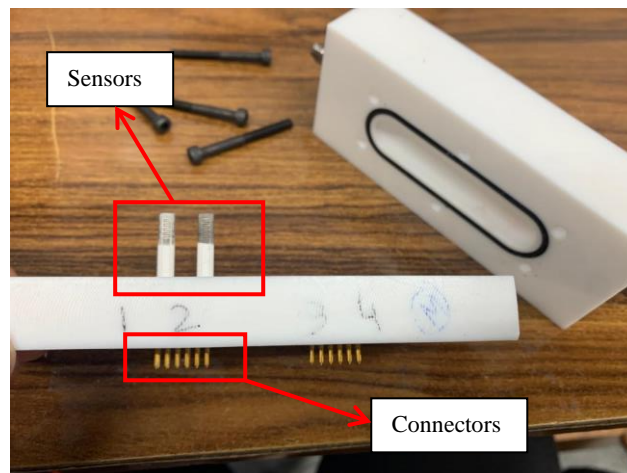
## 8. PROVES I RESULTATS

Arribats a aquest punt, es detallarà la prova realitzada per tal d'avaluar el conjunt del sistema. Degut a la necessitat d'una infraestructura habilitada per a l'alliberament controlat de NO<sub>2</sub>, només s'ha realitzat una única prova per tal de comprovar el funcionament del projecte. D'aquesta manera, la prova es va realitzar el dia 25 de maig del 2023 a la sala de caracterització de les plantes pilot de la universitat. En els següents apartats, es detallaran el muntatge complet del sistema i els resultats de la prova realitzada.

### 8.1. Preparació i muntatge.

Inicialment, es comentarà la preparació del material i la infraestructura per tal de duu a terme l'assaig.

En primera instància, s'ha usat una cambra on es col·loquen els sensors per al seu aïllament en el moment de l'alliberament del gas. En la *Figura 8.1* es pot observar un exemple de la cambra utilitzada.



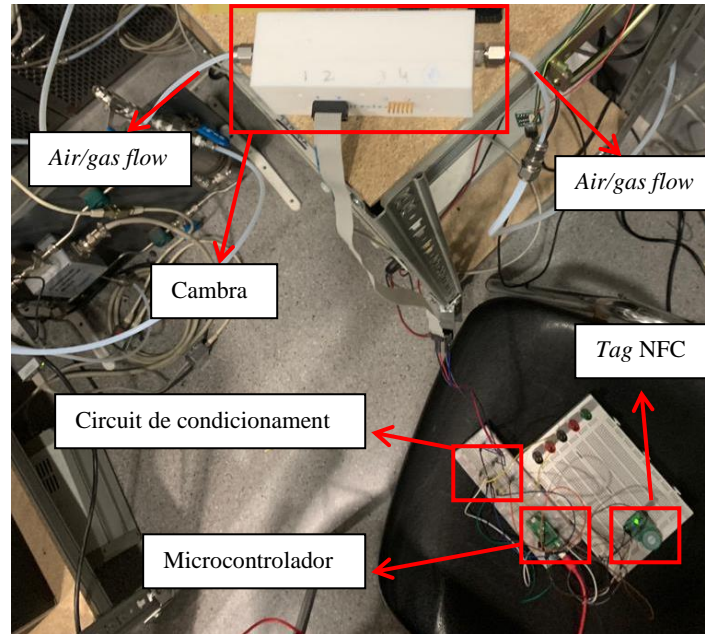
**Figura 8.1.** Cambra utilitzada per a la prova de concepte.

El connector utilitzat en la cambra per a la subjecció i mesura de la resistència dels sensors es pot observar en la *Figura 8.2*.



**Figura 8.2.** Connectors utilitzats per als sensors.

Seguidament, un cop col·locats els sensors, el següent pas ha sigut la muntura del sistema complet. En la *Figura 8.3* es pot identificar el muntatge i preparació final.



**Figura 8.3.** Muntatge i preparació final.

## 8.2. Resultats obtinguts.

Un cop muntat el conjunt del sistema, per tal de realitzar la prova de funcionament, s'han programat quatre cicles de mesura. El primer cicle i el tercer corresponen a aire, mentre que el segon correspon a una concentració de NO<sub>2</sub> de 20 ppm. L'hora d'inici de la prova es refereix a les 11:12 aproximadament. En la *Taula 8.1* es pot observar la programació dels diferents cicles, juntament amb l'hora corresponent i els resultats teòrics que s'haurien d'obtenir.

| Representació dels cicles en funció de l'hora |              |                 |                              |
|-----------------------------------------------|--------------|-----------------|------------------------------|
| Hora                                          | Cicle        | Element         | Concentració NO <sub>2</sub> |
| 11:12                                         | Primer cicle | Aire            | 0 ppm                        |
| 11:27                                         | Segon cicle  | NO <sub>2</sub> | 20 ppm                       |
| 11:42                                         | Tercer cicle | Aire            | 0 ppm                        |

**Taula 8.1.** Representació dels diferents cicles en funció de l'hora.

Pel que fa al processament de les dades, s'ha programat el *Timer* del microcontrolador per a realitzar l'escriptura de la concentració mesurada al *tag* NFC cada 10 minuts.

Tot seguit, es poden observar els resultats obtinguts en la *Taula 8.2*, d'acord amb les característiques exposades prèviament en la *Taula 8.1*.

**Resultats obtinguts**

**Mesura primer cycle**



Figura 8.4. Mesura corresponent al primer cycle.

**Mesura segon cycle**

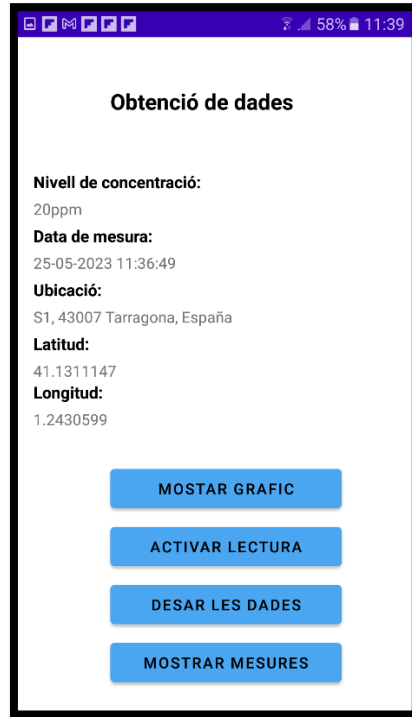


Figura 8.5. Mesura corresponent al segon cycle.

**Mesura tercer cycle**



Figura 8.6. Mesura corresponent al tercer cycle.

Taula 8.2. Representació dels resultats obtinguts.

Seguidament, en la *Taula 8.3* podem observar la *Figura 8.7*, *Figura 8.8* i *Figura 8.9* que representen l'emmagatzematge i la representació de les dades obtingudes.

**Emmagatzematge i visualització dels resultats obtinguts**



Figura 8.7. Emmagatzematge de les dades.

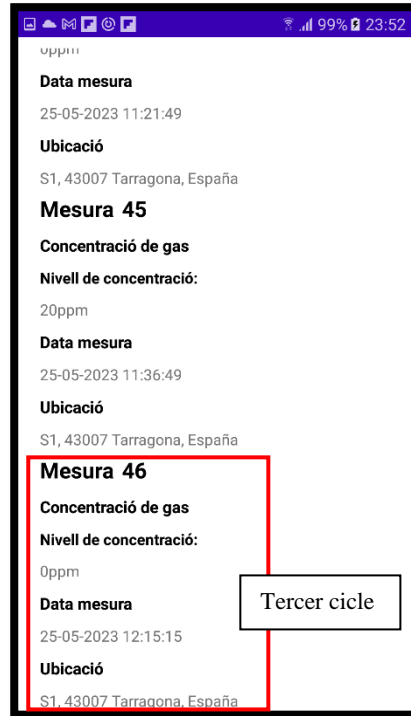


Figura 8.8. Emmagatzematge de les dades.

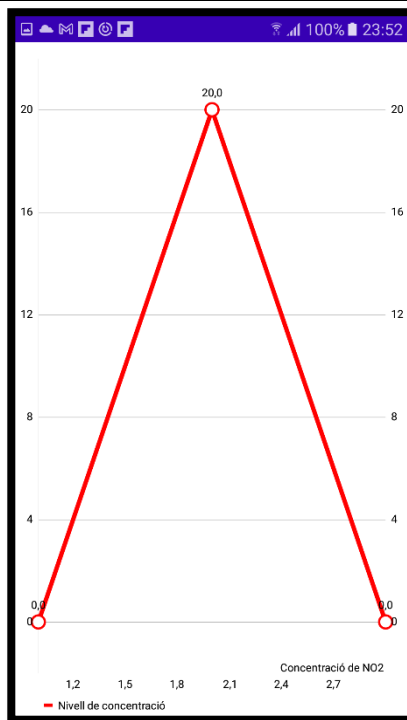


Figura 8.9. Representació de les dades obtingudes.

Taula 8.3. Resultats de l'emmagatzematge i la representació de les dades.

Finalment, per a concloure amb l'apartat del procediment i resultats obtinguts, en la referència [33] es pot veure un vídeo, a mode demostració i en temps real, on s'observen els resultats aconseguits corresponents al primer i segon cicle.

## 9. CONCLUSIONS I LÍNIES FUTURES DEL PROJECTE

Per finalitzar, en aquest últim capítol de la memòria, es revisaran els principals objectius i es detallaran les conclusions extretes derivades de la feina realitzada en el projecte. A més a més, es formalitzaran les possibles millores i línies d'investigació futures a partir de les activitats plasmades en aquest Projecte Final de Carrera.

### 9.1. Conclusions.

Els objectius principals plantejats per al desenvolupament d'aquest projecte han sigut el desenvolupament d'un sensor de tipus químic resistiu que permet detectar una determinada concentració de NO<sub>2</sub> en ambients interiors, juntament amb l'adquisició de dades i les comunicacions necessàries per a possibilitar la lectura del valor obtingut a través de NFC i mitjançant un dispositiu mòbil.

Tal i com s'ha observat, el sistema funciona correctament i executa les principals funcions descrites, tot i que és millorable. Per una banda, referent al desenvolupament del model qualitatiu per a diferenciar la concentració de NO<sub>2</sub> detectada en funció d'un rang de respostes, s'ha decidit la implementació d'un ANOVA per a la diferenciació entre grups, i per tant, entre concentracions. La tècnica desenvolupada no és errònia però tampoc perfecte, atès que els elements i factors que influeixen en la determinació de la resposta dels sensors contemplen altres variables com són la temperatura i la humitat. L'ideal, per tant, seria la realització d'una anàlisi multivariant, com ara un PCA o un MANOVA, que tingués en compte els aspectes comentats anteriorment per tal de confeccionar un model més precís.

Per altra banda, referent a l'etapa d'adquisició de les dades, s'ha utilitzat el voltatge de referència del propi microcontrolador per a l'alimentació del circuit de condicionament. S'ha observat com s'experimentava una mica de soroll degut a la commutació del voltatge que proporciona la *Raspberry Pi Pico*, tot i que no ha afectat el rendiment general del sistema, ja que s'ha pogut minimitzar amb l'aplicació d'un filtre. Una millora notable d'aquesta praxis, seria la utilització d'un voltatge de referència extern que permeti obtenir una major estabilitat.

Respecte a la comunicació entre el microcontrolador i l'etiqueta NFC, s'ha desenvolupat un programa que permet l'escriptura d'un missatge de text seguint el format NDEF, i suficient per als requisits funcionals del sistema. Així mateix, l'aplicació s'ha dissenyat d'acord al format del contingut escrit en el *tag* NFC. Per tant, en aquest aspecte, les conclusions extretes únicament contemplen la possible millora o optimització d'alguns dels mètodes de comunicació entre el microcontrolador i l'etiqueta NFC, d'acord amb l'aprenentatge de les funcionalitats no estudiades del llenguatge de programació *MicroPython*.

Pel que fa a la prova realitzada i els resultats obtinguts, s'ha verificat el funcionament global del sistema, podent diferenciar entre la detecció d'una certa concentració de NO<sub>2</sub> enfront la no detecció. Degut a la dependència d'instal·lacions requerides per a la realització de l'assaig, així com de personal per a la supervisió i preparació de la infraestructura, ha sigut difícil tant la preparació de l'etapa d'adquisició de dades com la de processament, atès que no es disposava de cap referència empírica del comportament i sensibilitat dels sensors en

les condicions de la prova i prèvies a l'assaig realitzat. Així doncs, un aspecte millorable seria la realització de diferents proves per analitzar el comportament de la resposta dels sensors i realitzar els ajustaments necessaris en les etapes d'adquisició i processament.

Finalment, per a culminar amb l'apartat de conclusions, es pot remarcar l'acompliment de bona part dels objectius inicialment plantejats tot i les circumstàncies donades. També constatar que la realització d'aquest projecte ha permès l'exploració i tractament amb sensors no estudiats en el grau, així com el desenvolupament i aplicació de la tecnologia NFC en altres àmbits.

## **9.2. Línies futures.**

Com a línies futures del projecte, es tenen en compte tres possibles escenaris de millora i treball futur.

Per una banda es contempla la realització d'un prototip més avançat i adaptat per a aplicacions reals, mitjançant el disseny i construcció d'una PCB per a la integració dels sensors, l'etapa de condicionament i el microcontrolador.

Així mateix, també es considera la realització i plasmació d'un model més precís que tingui en compte el conjunt de variables que intervenen en el càlcul de la resposta dels sensors, tal i com s'ha esmentat en les conclusions.

Finalment, la optimització de recursos físics i energètics és un aspecte a millorar. D'aquesta manera, utilitzant una font d'alimentació externa i fent ús del mode de baix consum per a la lectura de les dades es pot millorar aquesta qüestió.

## 10. REFERÈNCIES BIBLIOGRÀFIQUES

- [1] *United States Environmental Protection Agency, "Basic Information About NO<sub>2</sub>".* Enllaç web: <https://www.epa.gov/no2-pollution/basic-information-about-no2#What%20is%20NO2>
- [2] Instituto Nacional de Seguridad y Salud en el Trabajo. Documentación toxicológica para la actualización del límite de exposición profesional del dióxido de nitrógeno.
- [3] Real Decreto 1073/2002, de 18 d'octubre sobre l'avaluació i la gestió de la qualitat de l'aire. Enllaç web: <https://www.boe.es/boe/dias/2002/10/30/pdfs/A38020-38033.pdf>
- [4] *United States Environmental Protection Agency, "The impact of nitrogen dioxide on indoor air quality".* Enllaç web: <https://espanol.epa.gov/cai/el-impacto-del-dioxido-de-nitrogeno-en-la-calidad-del-aire-interior#levels>
- [5] Science Direct: Chemoresistors. Enllaç web: <https://www.sciencedirect.com/topics/engineering/chemiresistors>
- [6] Wenchao Tian, Xiaohan Liu and Wenbo Yu. "*Research Progress of Gas Sensor Based on Graphene and Its Derivatives: A Review*".
- [7] Irene Sanz Montero. "*Preparación de sensores de gases basados en grafeno para la detección de contaminantes atmosféricos*". Treball Final de Grau. 2019.
- [8] S. Di Carlo and M. Falasconi .Department of Control and Computer Engineering Politecnico di Torino, Torino. "*Drift Correction Methods for Gas Chemical Sensors in Artificial Olfaction Systems: Techniques and Challenges*"
- [9] María Victoria Bueno Delgado, Pablo Pavón Mariño y Alfonso de Gea García Departamento de Tecnologías de la Información y las Comunicaciones. Universidad Politécnica de Cartagena, "*La tecnología NFC y sus aplicaciones en un entorno universitario*".
- [10] Antonio Lázaro. "*Near Field Communication Technology*", Teoria sobre l'assignatura de Laboratori de Sensors i Technologies Mòbils.
- [11] Boris Evstatiev. "*Lecture notes in theory of electrical engineering. Mutually coupled inductors. Coupling coefficient. Power and energy of mutually coupled inductors. Analysis of circuits with mutually coupled inductor*".
- [12] Antonio Lázaro. Tecnologia de Radiofreqüència. URV 2009.
- [13] D.M.Pozar. "*Microwave Engineering*".
- [14] "*ST25 NFC guide*". Technical note.
- [15] Sony. "*About NFC*". Enllaç web: <https://www.sony.net/Products/felica/NFC/forum.html>
- [15] NFC Forum. "*Logical Link Control Protocol*". Technical Specification.
- [16] David Girbau. "*Tema 4. Transmissió Digital Pas Banda*". Teoria sobre l'assignatura de Comunicacions Digitals.
- [17] Naser Hossein Motlagh. University of Vaasa. 2012, "*Near Field Communication (NFC), a technical overview*".
- [18] Mohammed Riyazuddin. Information Technology Center. "*NFC A review of the technology, applications and security*".
- [19] *Raspberry Pi Ltd. 2020-2023. "Raspberry Pi Pico Datasheet"*.
- [20] Vicente Madero. Treball Final de Màster. "*Apéndice H. Bus de comunicación I<sup>2</sup>C*".
- [21] Ernesto González , Juan Casanova-Chafer, Alfonso Romero, Xavier Vilanova, Jan Mitrovics and Eduard Llobet, "*LoRa Sensor Network Development for Air Quality Monitoring or Detecting Gas Leakage Events*"

- [22] Filtre EMA. Enllaç web: <https://tttpa.github.io/Pages/Mathematics/Systems-and-Control-Theory/Digital-filters/Exponential%20Moving%20Average/Exponential-Moving-Average.html>
- [23] AN5250 Application Note, Freescale Semiconductor, Inc. *“How to Increase the Analog-to-Digital Converter Accuracy in an Application”*
- [24] *“Dynamic NFC/RFID tag IC with 64-Kbit EEPROM, NFC Forum Type 4 Tag and I<sup>2</sup>C interface”, Datasheet.*
- [25] AN4420 Application note, *“M24SR Programmer’s Application Note (based on M24SRDISCOVERY Firmware)”*
- [26] Android Developers, *“Introducción a Android Studio”*
- [27] Android Developers, *“Cómo interpretar el ciclo de vida de una actividad”*. Enllaç web: <https://developer.android.com/guide/components/activities/activity-lifecycle?hl=es-419>
- [28] Android Developers, *“Conceptos básicos de NFC”*. Enllaç web: <https://developer.android.com/guide/topics/connectivity/nfc/nfc?hl=es-419>
- [29] Android Developers, *“Cómo conocer la ubicación más reciente”*. Enllaç web: <https://developer.android.com/training/location/retrieve-current?hl=es-419>
- [30] Pere Millán. *“BD Room Corrutinas”*. Teoria sobre l’assignatura d’Aplicacions i Serveis Mòbils, 2021-2022.
- [31] Doctor Code. *“Android Studio and Kotlin Tutorial get Current Location”*. 13 de Maig del 2020. Youtube. Enllaç web: <https://www.youtube.com/watch?v=vard0CUTLbA&t=1476s>
- [32] MPAndroidChart. Github. Enllaç web: <https://github.com/PhilJay/MPAndroidChart>
- [33] Pau Moriones Galve. 7 de Juny del 2023. Vídeo demostració del funcionament del sistema. Enllaç web: <https://youtu.be/-hU42Cta11E>

## 11. ANNEXOS

L'apartat d'annexes consta, en primer lloc, del WBS i el diagrama de *Gantt* referent a la planificació del projecte. Seguidament, es presenten els diferents codis utilitzats per al desenvolupament del projecte i que no han sigut presentats en els apartats destinats a la seva descripció.

### 11.1. Estructura WBS i diagrama de *Gantt*.

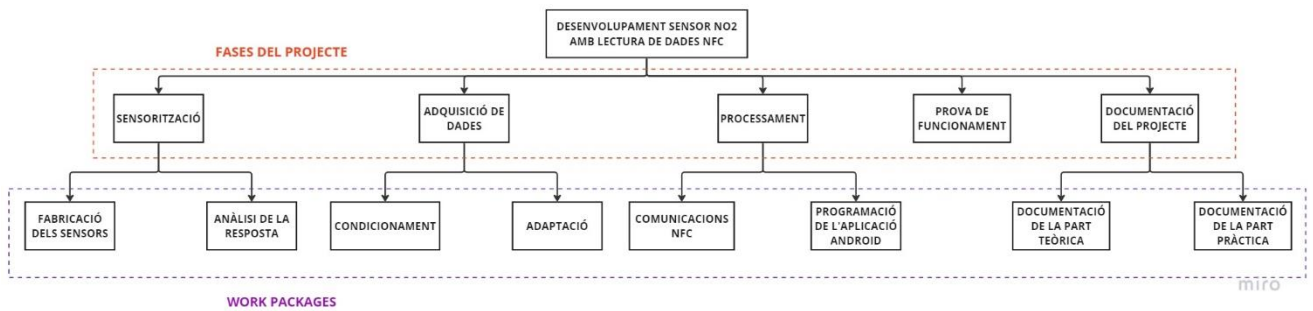


Figura 11.1. Estructura WBS del projecte.

Setmana 1: 13 a 17 de març de 2023.

Setmana 11: 22 a 26 de maig del 2023.

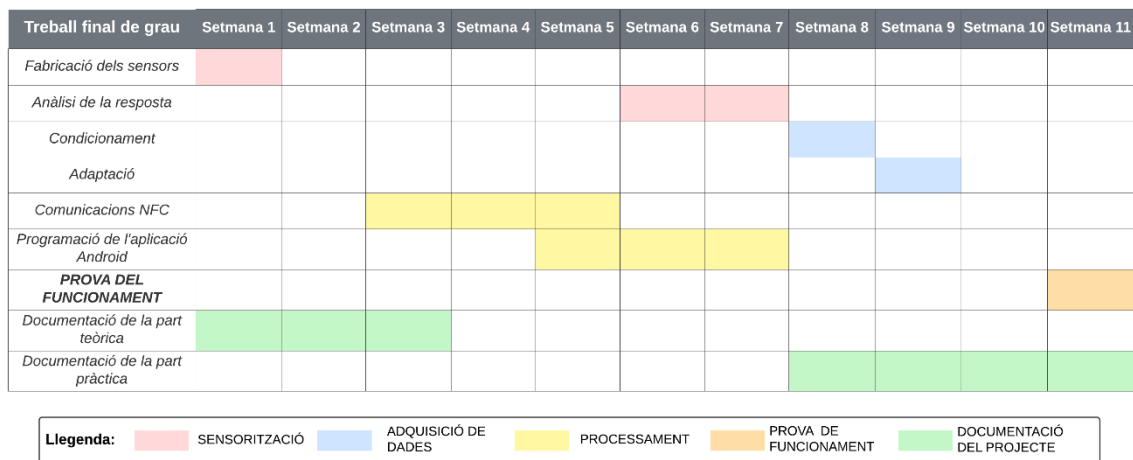


Figura 11.2. Diagrama de *Gantt* de la planificació del projecte.

## 11.2. Codis *MATLAB*.

### 11.2.1. Programa *funcioCalculaResposta.m*

```
function [resposta] = funcioCalculaResposta(dades,frames,inici,final)
%Funció Calcula Resposta.
% Permet calcular la resposta de les dades dels sensors.
% Dades - identifica el vector corresponent a les dades dels sensors.
% Frames - correspon als frames en que es vol dividir la senyal per a
% realitzar el càlcul de la resposta. En aquest projecte s'ha utilitzat
% com a valor, frames = 16.
% Inici - correspon a la mostra d'inici del vector de dades. El valor
% típic usat en aquest projecte és inici = 3000.
% Final - identifica el valor de mostra final del vector de dades. El
% valor típic usat en aquest projecte és final = 14450.

index = 1;
j = 0;
k = 1;
q = 1;
for i = inici:final
    j = j+1;
    aux(j) = dades(i);
    if(j == frames)
        j = 0;
        r_aire(k) = max(aux);
        r_gas(k) = min(aux);
        resposta(q) = (r_aire(q)-r_gas(q))/r_aire(q)*100;
        k = k+1;
        q = q+1;
    end
end
```

### 11.2.2. Programa *funcioRandomMatrix.m*

```
function [m,range] = funcioRandomMatrix(matrix,values)
%RANDOM MATRIX - Funció Random Matrix
% Permet calcular n nombres aleatoris compresos entre dos rangs.
% Matrix - identifica la matriu de dades sobre la qual es volen generar
% nombres aleatoris.
% Values - correspon al nombre de valors aleatoris que es volen
% generar.
% En aquest projecte, els dos rangs corresponen al primer i últim valor
% de cada columna de la matriu matrix.

for i = 1:length(matrix)
    matrixRandInferior = matrix(1,i);
    matrixRandSuperior = matrix(4,i);
    if(i == 1)
        random = matrixRandSuperior-(matrixRandSuperior -
        matrixRandInferior).*rand(values,1);
    else
        random = matrixRandInferior-(matrixRandInferior -
        matrixRandSuperior).*rand(values,1);
    end

    for j = 1:length(random)
        m(j,i) = random(j,1);
    end
    range = [min(m) max(m)];
end
```

### 11.2.3. Programa processarDadesSensors.m

```

dadesSensors = xlsread("Raw Data Mesures NO2 Amb Grafè");

%Variables
numberSamples = 14450;
min_axis_1 = 27000;
max_axis_1 = 40000;
min_axis_2 = 100000;
max_axis_2 = 140000;
inici = 3000;
mostresConc = 711;

%Carreguem les dades del fitxer Excel
for i=inici:numberSamples
    sensor1(i,1) = dadesSensors(i,2);
    sensor2(i,1) = dadesSensors(i,3);
end

%Correcció drift dades sensor 1
[newMatriu,referencia] =
funcioEliminarDrift(sensor1,mostresConc,inici,numberSamples);
resultatMatriu_1 = newMatriu;
llindar_1 = referencia;

%Correcció drift dades sensor 2
[newMatriu,referencia] =
funcioEliminarDrift(sensor2,mostresConc,inici,numberSamples);
resultatMatriu_2 = newMatriu;
llindar_2 = referencia;

%Representació de les dades corregides del sensor 1
figure(1);
plot(1:numberSamples,resultatMatriu_1)
title('Sensor 1 drift corregit')
yline(llindar_1,'--','Línia Base','Color','r')
xlabel('Temps(s)')
ylabel('Resistència ( $\Omega$ )')
axis([inici numberSamples min_axis_1 max_axis_1])

%Representació de les dades corregides del sensor 2
figure(2);
plot(1:numberSamples,resultatMatriu_2)
title('Sensor 2 drift corregit')
yline(llindar_2,'--','Línia Base','Color','r')
xlabel('Temps(s)')
ylabel('Resistència ( $\Omega$ )')
axis([inici numberSamples min_axis_2 max_axis_2])

%Càlcul de les respostes del sensor 1
[resposta] =
funcioCalculaResposta(resultatMatriu_1,mostresConc,inici,numberSamples);
vector_respostes1 = resposta

%Càlcul de les respostes del sensor 2
[resposta] =
funcioCalculaResposta(resultatMatriu_2,mostresConc,inici,numberSamples);
vector_respostes2 = resposta

%Matriu de respostes
[matriuRespostes] = generarMatriuRespostes(4,4,vector_respostes1);

```

```

matriuRespostes_1 = matriuRespostes;

[matriuRespostes] = generarMatriuRespostes(4,4,vector_respostes2);
matriuRespostes_2 = matriuRespostes;

%Afegim 1000 valors a les matrius de respostes amb la funcioRandomMatrix
[m,range] = funcioRandomMatrix(matriuRespostes_1,1000);
nova_matriu_resp1 = m;
rang_1 = range;
[m,range] = funcioRandomMatrix(matriuRespostes_2,1000);
nova_matriu_resp2 = m;
rang_2 = range;

%Realitzem el test ANOVA per a cada un dels sensors
groups = [5 10 15 20];
test1 = anoval(nova_matriu_resp1,groups)
test2 = anoval(nova_matriu_resp2,groups)
%Representació de la funció de probabilitat F
df1 = 3;
df2 = 12;
x = 0:0.01:10;
y = fpdf(x,df1,df2);
plot(x,y,'b','LineWidth',2)
title('Distribució F(3,12)')
xlabel('F')
ylabel('Densitat')

```

### 11.3. Programes comunicació NFC i obtenció de dades.

#### 11.3.1. Classe tagM24SR

```

class tagM24SR():

    I2C_ADDRESS = 0x56
    PCB_02 = 0x02
    PCB_03 = 0x03
    CLA = 0x00
    INS = 0xD6
    P1 = 0x00
    P2 = 0x02
    RL = 0xB1
    TL = 0x01
    TYPE = 0x54
    SB = 0x02
    IANA_CODE = [0x65,0x6E]
    CRC_REG = 0x6363
    NDEF_L = 4
    PAYLOAD_L = 3

    def __init__(self,i2c):
        self.i2c = i2c
        self.addr = self.I2C_ADDRESS
        self.crc = self.CRC_REG

    def actualitzarCRC(self, data):
        data = data ^ ((self.crc) & 0x00FF)
        data = data ^ ((data << 4) & 0x00FF)
        self.crc = (self.crc >> 8) \
            ^ (data << 8) \
            ^ (data << 3) \
            ^ (data >> 4)

```

```

self.crc = self.crc & 0xFFFF
return self.crc

def obtenirCRC(self):
    return (self.crc & 0xFF), ((self.crc & 0xFF00)>>8)

def calcularCRC(self,i2cframe):
    self.crc = self.CRC_REG
    for i in range(len(i2cframe)):
        self.actualitzarCRC(i2cframe[i])
    crc0, crc1 = self.obtenirCRC()
    return crc0, crc1

def frameNdef(self,ndef_length,payload_length):
    header=[bytes([self.PCB_02]),bytes([self.CLA]),bytes([self.INS]),
    bytes([self.P1]),bytes([self.P2]),ndef_length,bytes([self.RL]),
    bytes([self.TL]),payload_length,bytes([self.TYPE]),bytes([self.SB],
    bytes(self.IANA_CODE)]

    missatge = b''.join(header)
    return missatge

def frameNdefLength(self,ndef_length):
    header=[bytes([self.PCB_03]),bytes([self.CLA]),bytes([self.INS]),
    bytes([self.P1]),bytes([self.CLA]),bytes([self.P2]),
    bytes([self.P1]),ndef_length]

    length = b''.join(header)
    return length

def missatgeNDEF(data):
    suma = 0
    llista = list()
    x = data.split(" ")

    for i in range(len(x)):
        suma += len(x[i])

    blancs = len(x) - 1
    llista.append(bytes(data,'ascii'))
    v_int = b''.join(llista)
    length_tot = bytes([suma+blancs+self.PAYLOAD_L])
    length_NDEF = length_tot + bytes([self.NDEF_L])
    return length_tot,length_NDEF,v_int

def calculFrameI2C(self,data):
    payload_length = b''
    s = b''
    NDEF_length = b''
    i = 0
    payload_length = data[i]
    NDEF_length = data[i+1]
    s = data[i+2]
    cap_ndef = self.frameNdef(NDEF_length,payload_length)
    i2c_frame = cap_ndef + s
    crc = self.calcularCRC(i2c_frame)
    i2c_frame += bytes(crc)

```

```

    return i2c_frame

def calculCrClenMiss(self, llargada):
    length = self.frameNdefLength(llargada)
    crc = self.calcularCRC(length)
    length += bytes(crc)
    return length

def comprovarAgreement(self, agr):
    agreement = b''
    status = self.comprovarResposta(agr)
    if(status == 1):
        print("No necessites la comanda agreement\n")
    else:
        agreement = agr
        print("Necessites la comanda agreement\n")
    return agreement

def comprovarResposta(missatge):
    failed = 0
    if(missatge.count(b'\x90') == 1 and missatge.count(b'\x00') ==1):
        failed = 1
    return failed

def escriureTag(missatge, length):
    com = bytes(missatge)
    self.i2c.writeto(self.addr, com)
    utime.sleep_ms(5)
    resultat = self.i2c.readfrom(self.addr, length)
    return resultat

```

### ***11.3.2. Programa principal.***

```

resposta1 = 0.0
resposta2 = 0.0

concentracions = ['0', '5', '10', '15', '20']
umbral_respostes = [7.0, 8.4, 10.5, 11.7, 12.5]

def convertirDades(dada1, dada3):
    dades = dada1+"ppm"+" "+str(dada3)
    return dades

def convertirTemps(temps):
    count = 0
    time = ""
    for i in range(4, 7):
        time = time+ str(temps[i])
        if(count < 2):
            time = time+":"
            count+=1
    return time

def comprovarRang(resp, v1, v2):
    resultat = False
    if(resp > v1 and resp < v2):
        resultat = True
    return resultat

```

```

def comprovarConcentracions(resp):
    concentracio = concentracions[0]
    if(comprovarRang(resp,umbral_respostes[0],umbral_respostes[1])):
        concentracio = concentracions[1]

    if(comprovarRang(resp,umbral_respostes[1],umbral_respostes[2])):
        concentracio = concentracions[2]

    if(comprovarRang(resp,umbral_respostes[2],umbral_respostes[3])):
        concentracio = concentracions[3]

    if(comprovarRang(resp,umbral_respostes[3],umbral_respostes[4]) or
resp > umbral_respostes[4]):
        concentracio = concentracions[4]

    return concentracio

def voltatgeAresis(V1,V2):
    R1 = 820000
    R2 = 1000000
    Rsens1 = (V1*R1)/(Vc-V1)
    Rsens2 = (V2*R2)/(Vc-V2)

    return Rsens1,Rsens2

#Inicialitzacions I2C, real time clock i classe tagM24SR
i2c = I2C(1, sda=machine.Pin(6), scl=machine.Pin(7),freq=100000)
rtc = machine.RTC()
tag = tagM24SR(i2c)

#variables
Vdt = 0.0
Vdt2 = 0.0
R1 = 820000
R2 = 1000000
maximm = 0.0
minimm = R2
array_temps = []

#Trames per a la comunicació amb el tag

#trama i2c de NDEF Tag Application
NDEFAppTagCommand =
[0x02,0x00,0xA4,0x04,0x00,0x07,0xD2,0x76,0x00,0x00,0x85,0x01,0x01,0x00,0x
35,0xC0]

#trama i2c SelectCCFile
SelectCCFile = [0x03,0x00,0xA4,0x00,0x0C,0x02,0xE1,0x03,0xD2,0xAF]

#trama Read CC_File_length
Read_CC_file_length = [0x02,0x00,0xB0,0x00,0x00,0x02,0x6B,0x7D]

#trama Read CC_File
Read_CC_file = [0x03,0x00,0xB0,0x00,0x00,0x0F,0xA5,0xA2]

#trama select NDEF file
Select_NDEF_file = [0x02,0x00,0xA4,0x00,0x0C,0x02,0x00,0x01,0x3E,0xFD]

#trama llegir missatge NDEF

```

```

Read_NDEF_message_length = [0x03,0x00,0xB0,0x00,0x00,0x02,0x40,0x79]

#trama erase NDEF length
Erase_NDEF_length = [0x03,0x00,0xD6,0x00,0x00,0x02,0x00,0x00,0x6B,0x37]

#trama deselect
deselect_command = [0xC2,0xE0,0xB4]
#Rutina d'interrupció del timer

def escriureDades(missatge):
    #processar missatge i codificar missatge NDEF i longitud
    a = tag.missatgeNDEF(missatge)
    ndef_frame = tag.calculFrameI2C(a)
    length_frame = tag.calculCrCLenMiss(a[1])

    #actualitzar missatge NDEF
    update_bin_NDEF = ndef_frame
    #actualitzar mida missatge NDEF
    update_bin = length_frame

    #iniciar sessió I2C
    result = tag.escriureTag(GET_I2C_SESSION,4)

    #seleccionar aplicació NFC
    NFC_App = tag.escriureTag(NDEFAppTagCommand,4)
    print("Select NFC App:",NFC_App)

    #seleccionar fitxer NDEF
    NDEF_file = tag.escriureTag(Select_NDEF_file,4)
    print("Select NDEF file:",NDEF_file)

    #esborrar longitud missatge NDEF anterior
    erase_len_NDEF = tag.escriureTag(Erase_NDEF_length,4)
    print("Erase NDEF length:",erase_len_NDEF)

    #actualitzar missatge NDEF
    update_binary_NDEF = tag.escriureTag(update_bin_NDEF,4)
    print("Update binary NDEF file content:",update_binary_NDEF)

    #comprovar correcte enviament del missatge
    agreement = tag.comprovarAgreement(update_binary_NDEF)
    print(agreement)

    #escriptura agreement en cas que sigui necessari
    agree = tag.escriureTag(agreement,4)
    print("Agreement:",agree)

    #actualitzar mida del missatge NDEF
    update_binary_length = tag.escriureTag(update_bin,4)
    print("Update NDEF content:",update_binary_length)

    #tancar sessió I2C i activar sessió RF
    deselct = tag.escriureTag(deselect_command,4)
    print("Deselect:",deselct)

def interrupcio(Source):

    #calcul màxim, mínim iu resposta
    print("MAXIM 2",maximm,"MINIM 2",minimm)
    dif = maximm - minimm
    resposta = (dif/maximm)*100

```

```

print("RESPOSTA", resposta)

#obtenir dades de concentració i temps
temps = rtc.datetime()
concentracio = comprovarConcentracions(resposta)
hora = convertirTemps(temps)
dades = convertirDades(concentracio, hora)
escriureDades(dades)
#reiniciar valors màxim i mínim
inter = 1

#Inicialització timer i variable de control del temps
#d'adaptació del filtre
start = 0
timer = Timer(period=3000, mode = Timer.PERIODIC, callback = interrupcio)

#Inicialització adc
adc_sensor1 = ADC(Pin(26, mode = Pin.IN))
adc_sensor2 = ADC(Pin(27, mode = Pin.IN))

#Voltatge de referència i nivells adc 16-bits
Vc = 3.3
quant_levels = 65535

#Inicialització valors filtre de voltatge
Y = Vdt
Y2 = Vdt2

#Inicialització valors de control
#dels màxims i els mínims
global inter
inter = 0

#Valors alfa filtre de voltatge (alfa1) i resistència (alfa2)
alfa1 = 0.5
alfa2 = 0.3

#resistències ideals per al
#càlcul del temps d'adaptació del filtre
r_ideal_1 = 895000
r_ideal_2 = 335000

while(True):

    #reiniciar els màxims i mínims després de cada execució de la
    interrupció
    if(inter == 1):
        minimm = R2
        maximm = 0
        inter = 0

    #lectura adc
    sensor1 = adc_sensor1.read_u16()
    sensor2 = adc_sensor2.read_u16()

    #conversió lectura adc a tensió
    Vdt = sensor1*(Vc/quant_levels) #Voltatge ADC
    Vdt2 = sensor2*(Vc/quant_levels) #Voltatge ADC

```

```

#càlcul dels filtres de tensió
Y = (alfa1*Vdt) + ((1-alfa1)*Y)
Y2 = (alfa1*Vdt2) + ((1-alfa1)*Y2)

#conversió a resistència
R = voltatgeAresis(Y2,Y)
Rsens = voltatgeAresis(Vdt2,Vdt)

#càlcul filtres de resistència
RY = R[1]
RX = R[0]
RY = (alfa2*R[1]) + ((1-alfa2)*RY)
RX = (alfa2*R[0]) + ((1-alfa2)*RX)

#temps d'espera per a l'adaptació dels filtres
if(RY > r_ideal_1*0.99 and RX > r_ideal_2*0.99 and start == 0):
    start = 1
    print (Vdt,Y,RY,Rsens[1],Vdt2,Y2,RX)

if(start == 1):
    print (Vdt,Y,RY,Rsens[1],Vdt2,Y2,RX,Rsens[0])
    if(RY > maximm):
        maximm = RY
    if(RY < minimm):
        minimm = RY
    init = 1
    time.sleep_ms(1000)

```

## 11.4. Activitats aplicació mòbil.

### 11.4.1. MainActivity.

```

class MainActivity : AppCompatActivity() {

    private lateinit var fusedLocationProviderClient:
    FusedLocationProviderClient
    private lateinit var latitud:TextView
    private lateinit var longitud:TextView
    private lateinit var ubicacio:TextView
    private lateinit var locationRequest: LocationRequest
    private lateinit var desarDades: Button
    private lateinit var activarLectura:Button
    private lateinit var mostrarDades:Button
    private lateinit var mostrarGrafic:Button
    private lateinit var sensor1:TextView
    private lateinit var data:TextView
    private var activat:Boolean = false
    private var nfcAdapter:NfcAdapter? = null
    private lateinit var db:NfcDatabase
    var PERMIS_ID = 500

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        supportActionBar?.hide()

        fusedLocationProviderClient =
        LocationServices.getFusedLocationProviderClient(this)

        latitud = findViewById(R.id.contingut_latitud)

```

```

longitud = findViewById(R.id.contingut_longitud)
ubicacio = findViewById(R.id.contingut_ubicacio)
desarDades = findViewById(R.id.boto_desar)
activarLectura = findViewById(R.id.boto_mostrar)
mostrarDades = findViewById(R.id.boto_mostrar_mesures)
mostrarGrafic = findViewById(R.id.boto_mostrar_grafic)
sensor1 = findViewById(R.id.contingut_nfcl)
data = findViewById(R.id.contingut_data)

//Inicialitzar base de dades
db = NfcDatabase.getDataBase(this)

nfcAdapter = NfcAdapter.getDefaultAdapter(this)

val intent: Intent = intent
val action: String? = intent.action

if(NfcAdapter.ACTION_NDEF_DISCOVERED.equals(action)){
    processarMissatgeNdef(intent)
}

mostrarDades.setOnClickListener {

    val intent = Intent(this, DataActivity::class.java)
    startActivity(intent)
}

mostrarGrafic.setOnClickListener {
    val intent = Intent(this, LineGraphActivity::class.java)
    startActivity(intent)
}

}

private fun processarMissatgeNdef(intent: Intent) {
    val parcelabled =
intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES)

    if (parcelabled != null) {
        //Indiquem que s'ha detectat una etiqueta NDEF
        Toast.makeText(
            applicationContext, "TAG NDEF DETECTAT ",
            Toast.LENGTH_LONG
        ).show()
        //Creem un array per emmagatzemar els missatges NDEF que
obtenim.
        val ndefmsg = arrayOfNulls<NdefMessage>(parcelabled.size)

        ndefmsg[0] = parcelabled[0] as NdefMessage
        // Obtenir les dades del primer record NDEF.
        Log.i("NDEF Intent process",
ndefmsg[0]!!.records[0].toString())
        val mis = ndefmsg[0]!!.records[0].toString()
        val sep: String = separarRecord(mis)
        val miss: String = convertirHexaAscii(sep)
        val valor = miss.split(" ")
        Log.i("NDEF MISSATGE", sep)

        //missatge.setText(ndefmsg[0].getRecords()[i].getPayload().toString());
        val simpleDateFormat = SimpleDateFormat("dd-MM-yyyy ")

```

```

        val dates_data = simpleDateFormat.format(Date())

        sensor1.text = valor[0]
        data.text = dates_data + valor[1]
    }

}

private fun convertirHexaAscii(sep: String): String {
    var res: String
    val mess = arrayOfNulls<String>(sep.length / 2)
    var num: Int
    var missatge = ""
    var n = 6

    var i = 6
    while (i < sep.length) {
        res = "" + sep.get(i) + sep.get(i + 1)
        num = res.toInt(16)
        if (i - n > 0) {
            n++
            mess[i - n] = num.toChar().toString()
        } else {
            n = 6
            mess[i - n] = num.toChar().toString()
        }
        missatge += mess[i - n]
        i += 2
    }
    return missatge
}

private fun separarRecord(record: String):String{

    var missatge = ""
    var contingut = ""

    val separar = record.split("█".toRegex()).toTypedArray()

    for (i in separar.indices) {
        missatge = if (separar[i].contains("payload")) {
            separar[i]
        } else {
            " "
        }
    }
    if (missatge != " ") {
        val getPayload = missatge.split("█".toRegex()).toTypedArray()
        for (j in getPayload.indices) {
            if (getPayload[j] != "payload") {
                contingut = getPayload[j]
            }
        }
    }

    return contingut

}

private fun ubicacioActual(){
    if(comprovarPermissos()){

```

```

        if(ubicacioActiva()){
fusedLocationProviderClient.lastLocation.addOnCompleteListener { task->
    val location = task.result
    if(location==null){
        obtenirNovaUbi()
    }else{
        latitud.text = location.latitude.toString()
        longitud.text = location.longitude.toString()
        ubicacio.text =
ubicacioAdreca(location.latitude,location.longitude)
    }
    }else{
        Toast.makeText(this,"Has d'activar la teva
ubicacio",Toast.LENGTH_LONG).show()
    }
    }else{
        demanarPermissos()
    }
}

//Comprovar permissos d'ubicació inclossos a l'AndroidManifest.xml
private fun comprovarPermissos():Boolean{
    var resultat = false

if(ActivityCompat.checkSelfPermission(this,Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,Manifest.permission.ACCESS_COARSE_LOCATION) == PackageManager.PERMISSION_GRANTED ){
    resultat = true
}
return resultat
}

//Demanar els permissos d'ubicació inclossos a l'AndroidManifest.xml
private fun demanarPermissos(){
    ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.ACCESS_FINE_LOCATION,Manifest.permission.ACCESS_COARSE_LOCATION),PERMIS_ID)
}

private fun ubicacioActiva():Boolean{
    var resultat = false
    val locationManager:LocationManager =
getSystemService(Context.LOCATION_SERVICE) as LocationManager
    val ubi_activa =
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)
    val ubi_xarxa =
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER)

    if(ubi_activa || ubi_xarxa){
        resultat = true
    }
    return resultat
}

private fun obtenirNovaUbi(){

```

```

        locationRequest = LocationRequest()
        locationRequest.priority = LocationRequest.PRIORITY_HIGH_ACCURACY
        locationRequest.interval = 0
        locationRequest.fastestInterval = 0
        locationRequest.numUpdates = 2

fusedLocationProviderClient.requestLocationUpdates(locationRequest, locati
onCallback,
            Looper.myLooper())
    }

    private val locationCallback = object: LocationCallback(){
        override fun onLocationResult(p0: LocationResult) {
            super.onLocationResult(p0)

            val lastLocation:Location = p0.lastLocation
            latitud.text = lastLocation.latitude.toString()
            longitud.text = lastLocation.longitude.toString()
            ubicacio.text =
ubicacioAdreca(lastLocation.latitude,lastLocation.longitude)
        }
    }

    private fun ubicacioAdreca(latitud:Double,longitud:Double):String{
        val address:String
        val ad:String

        val geocoder = Geocoder(this, Locale.getDefault())
        val ubicacio = geocoder.getFromLocation(latitud,longitud,1)
        ad = ubicacio!![0].getAddressLine(0)
        address = ad

        return address
    }

    override fun onResume() {
        super.onResume()
        activarLectura.setOnClickListener {

            val intent = PendingIntent.getActivity(
                this, 0, Intent(
                    this,
                    javaClass
                ).addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP), 0
            )
            NfcAdapter.getDefaultAdapter(this).enableForegroundDispatch(
                this, intent, null,
                null
            )

            activat = true

        }

        desarDades.setOnClickListener {
            if(activat){
                val mesural:String = sensor1.text.toString()
                val data: String = data.text.toString()
            }
        }
    }

```

```

        val ubi:String = ubicacio.text.toString()
        val mesures:String = mesural
        if((mesural.isNotEmpty()) && data.isNotEmpty() &&
ubi.isNotEmpty()){

            val dataEntity = DataEntity(0,mesures,data,ubi)
            //    DevUtils.addData(dataDao,dataEntity)

            GlobalScope.launch(Dispatchers.IO) {
                db.dataDao.insertAll(dataEntity)
                //db.dataDao.deleteAll()
            }

        }
        Toast.makeText(this, "Dades emmagatzemades
correctament",Toast.LENGTH_LONG).show()
    }else{
        Toast.makeText(this, "Per emmagatzemar les dades has
d'habilitar la lectura i llegir un tag NFC",Toast.LENGTH_LONG).show()
    }
    activat = false
}

}

override fun onPause() {
    super.onPause()
    if (NfcAdapter.getDefaultAdapter(this) != null) {
NfcAdapter.getDefaultAdapter(this).disableForegroundDispatch(this)
    }
}

override fun onNewIntent(intent: Intent) {
    super.onNewIntent(intent)
    ubicacioActual()
    if (intent.action == NfcAdapter.ACTION_NDEF_DISCOVERED) {
        processarMissatgeNdef(intent)
    }
}
}
}

```

### 11.4.2. DataActivity.

```

class DataActivity : AppCompatActivity() {

    private lateinit var recyclerView: RecyclerView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_data)

        supportActionBar?.hide()

        recyclerView = findViewById(R.id.recyclerview)
    }
}

```

```

        GlobalScope.launch(Dispatchers.IO) {
            obtenerDades()
        }

    }

    private fun obtenerDades(){
        recyclerView.adapter =
        DataAdapter(this,NfcDatabase.getDataBase(this).dataDao.getAll())
    }
}

```

### 11.4.3. DataAdapter.

```

class
DataAdapter(list:List<DataEntity>):RecyclerView.Adapter<DataAdapter.ViewHolder>() {

    private var dataList: List<DataEntity> = list
    private lateinit var dialog:AlertDialog.Builder
    private lateinit var dadesEntity: DataEntity
    private lateinit var separarDadesSensors: List<String>
    class ViewHolder(view: View):RecyclerView.ViewHolder(view){

        val dadaGasSensor1: TextView
        val dadaData:TextView
        val dadaUbi:TextView
        val uuid:TextView
        val root:View

        init {
            dadaGasSensor1 = view.findViewById(R.id.dada_rv_sensor1)
            dadaData = view.findViewById(R.id.data_rv)
            dadaUbi = view.findViewById(R.id.ubicacio_rv)
            uuid = view.findViewById(R.id.dada_uuid_rv)
            root = view
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
    ViewHolder {
        val view = LayoutInflater.from(parent.context)
            .inflate(R.layout.recyclerview_layout, parent, false)

        return ViewHolder(view)
    }

    @OptIn(DelicateCoroutinesApi::class)
    override fun onBindViewHolder(holder: ViewHolder, position: Int) {

        val currentItem = dataList[position]
        holder.dadaGasSensor1.text = currentItem.gasdata.toString()
        holder.dadaData.text = currentItem.timemeasure.toString()
        holder.dadaUbi.text = currentItem.locationmeasure.toString()
        holder.uuid.text = currentItem.uuid.toString()

        holder.root.setOnClickListener {
            Toast.makeText(holder.root.context,"Mesura %d

```

```

seleccionada".format(position+1), Toast.LENGTH_LONG).show()
        dadesEntity =
DataEntity(currentItem.uuid, currentItem.gasdata, currentItem.timemeasure, c
urrentItem.locationmeasure)
        dialeg = AlertDialog.Builder(holder.root.context)
        dialeg.setTitle("Mesura %d seleccionada".format(position+1))
        val d:Array<CharSequence> = arrayOf("Esborrar")
        dialeg.setItems(d, DialogInterface.OnClickListener {
dialogInterface, i ->
            if( i == 0){
                GlobalScope.launch(Dispatchers.IO) {

NfcDatabase.getDataBase(holder.root.context).dataDao.deleteData(dadesEnti
ty)
                    }
                }
            })
            dialeg.create().show()
        }

    }

    override fun getItemCount(): Int {
        return dataList.size
    }
}

```

#### **11.4.4. LineGraphActivity.**

```

class LineGraphActivity : AppCompatActivity() {

    private lateinit var arrayDadesSensor1: ArrayList<Entry>
    private lateinit var arrayDadesDb: List<DataEntity>
    private lateinit var lineChart: LineChart
    private var numElements = 0
    private lateinit var lineDataSetSensor1: LineDataSet
    private lateinit var descriptionGrafic: Description
    private lateinit var dataSetsSensors: ArrayList<LineDataSet>
    private lateinit var llegendas: Legend

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_line_graph)
        supportActionBar?.hide()
        lineChart = findViewById(R.id.lineGraph)

        GlobalScope.launch(Dispatchers.IO) {
            obtenirDadesGrafic()
        }
    }

    private fun obtenirDadesGrafic() {
        arrayDadesSensor1 = ArrayList<Entry>()
        val arrayX = ArrayList<String>()
        arrayDadesDb = NfcDatabase.getDataBase(this).dataDao.getValues()
        numElements = arrayDadesDb.size
        if(numElements != 0) {
            for (i in 0..numElements - 1) {
                Log.i("DATA FROM DATABASE:",
arrayDadesDb[i].gasdata.toString())
                arrayX.add((i + 1).toString())
            }
        }
    }
}

```

```

        val res =
separarDades(arrayDadesDb[i].gasdata.toString())
        Log.i("DATA SEPARADA", res[0].toString())
        arrayDadesSensor1.add(Entry((i.toFloat() + 1), res[0]))
    }

    lineDataSetSensor1 = LineDataSet(arrayDadesSensor1, "Nivell
de concentració")
    dataSetsSensors = ArrayList<LineDataSet>()
    //Inicialització de les variables que permeten personalitzar
    //la descripció i la llegenda del gràfic
    descriptionGrafic = Description()
    llegenda = Legend()
    //Custom linedataset de les dades
    lineDataSetSensor1.lineWidth = 4F
    lineDataSetSensor1.circleRadius = 7F
    lineDataSetSensor1.circleHoleRadius = 5F
    lineDataSetSensor1.setCircleColor(Color.RED)
    lineDataSetSensor1.valueTextSize = 10F
    lineDataSetSensor1.color = Color.RED
    //Descripció
    descriptionGrafic.text = "Concentració de NO2"
    descriptionGrafic.textSize = 10F
    //Llegenda
    llegenda = lineChart.legend
    llegenda.form = Legend.LegendForm.LINE
    lineChart.description = descriptionGrafic
    dataSetsSensors.add(lineDataSetSensor1)
    val dades = LineData(dataSetsSensors as List<ILineDataSet>?)
    lineChart.data = dades
    //lineChart.invalidate()
    val x: XAxis = lineChart.xAxis
    val y: YAxis = lineChart.axisRight
    y.setDrawGridLines(false)
    y.setDrawAxisLine(false)
    //x.valueFormatter = IAxisValueFormatter{_,_ -> arrayX[h]}
    x.setDrawGridLines(false)
    x.setPosition(XAxis.XAxisPosition.BOTTOM)

}else{
    lineChart.setNoDataText("No hi han dades")
}

}

private fun separarDades(dada:String):ArrayList<Float>{
    val resultat:ArrayList<Float> = ArrayList()
    val aux = dada.split(" ")
    var identifica = false
    var chArr: CharArray
    var valor = " "
    for(i in aux.indices){
        chArr = aux[i].toCharArray()
        for(j in chArr.indices){
            if(chArr[j].isDigit()) {
                valor += chArr[j]
            }
        }
        identifica = true
    }
    if(identifica){

```

```
        resultat.add(valor.toFloat())
        valor = " "
        //resultat.add(aux[i].toFloat())
    }
}
return resultat
}
}
```