



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

CAMPUS D'ALCOI

# UNIVERSITAT POLITÈCNICA DE VALÈNCIA

## Escola Politècnica Superior d'Alcoi

Disseny i implementació d'un sistema de sensors intel·ligent per a la supervisió de pacients en habitacions de centres geriàtrics.

Treball Fi de Grau

Grau en Enginyeria Informàtica

AUTOR/A: Capó Calabuig, Jordi

Tutor/a: Molina Picó, Antonio

CURS ACADÈMIC: 2024/2025

---

## Resum

En les darreres dècades, el nombre de persones majors ha crescut significativament a la nostra societat, fet que ha incrementat la demanda de serveis geriàtrics i d'atenció especialitzada. Amb l'envelliment, les persones són més susceptibles de patir problemes de salut, caigudes i altres incidents que poden posar en risc la seua vida. Aquestes situacions moltes voltes no es detecten a temps, sobretot durant la nit o quan hi ha menys personal disponible als centres geriàtrics, fet que augmenta el risc de complicacions greus o fins i tot de mort.

Per resoldre aquesta problemàtica, es proposa el desenvolupament d'un sistema pràctic, econòmic i aplicable a qualsevol tipus de geriàtric, que permeta la detecció immediata de situacions de risc per als pacients. La solució plantejada consisteix en la instal·lació d'un sistema de sensors basat en Arduino i l'entrenament d'alguns d'ells mitjançant Machine Learning per a diferenciar patrons que puguen suposar algún risc per al pacient.

Aquest conjunt de sensors es connectarà a un programa central destinat al personal sanitari, que comptarà amb una interfície gràfica fàcil d'utilitzar. El programa permetrà visualitzar l'estat de totes les habitacions del geriàtric en temps real, alertant el personal de manera immediata si es detecta alguna anomalia, i facilitant així una resposta ràpida i eficient davant de qualsevol situació de risc.

**Paraules clau:** habitació, sensor, Arduino, geriàtric, pacients, supervisió, Machine Learning, interfície, salut, monitorització, assistència.

---

## Abstract

In recent decades, the number of elderly people in our society has significantly increased, which has led to a growing demand for geriatric services and specialized care. As people age, they become more susceptible to health issues, falls, and other incidents that may endanger their lives. These situations often go undetected, especially at night or when there is less staff available in nursing homes, increasing the risk of severe complications or even death.

To address this issue, we propose the development of a practical, cost-effective system applicable to any type of nursing home, capable of enabling the immediate detection of risk situations for patients. The proposed solution involves the installation of a sensor system based on Arduino and the training of some of these sensors using Machine Learning to distinguish patterns that might indicate potential risks for the patient.

This set of sensors will be connected to a central program designed for healthcare personnel, featuring an easy-to-use graphical interface. The program will allow real-time visualization of the status of all rooms in the nursing home, immediately alerting staff if an anomaly is detected, thus enabling a quick and efficient response to any risky situation.

**Key Words:** room, sensor, Arduino, nursing home, patients, monitoring, Machine Learning, interface, health, supervision, assistance.

---

## Resumen

En las últimas décadas, el número de personas mayores ha crecido significativamente en nuestra sociedad, lo que ha incrementado la demanda de servicios geriátricos y de atención especializada. Con el envejecimiento, las personas son más susceptibles a sufrir problemas de salud, caídas y otros incidentes que pueden poner en riesgo su vida. Estas situaciones muchas veces no se detectan a tiempo, especialmente durante la noche o cuando hay menos personal disponible en los centros geriátricos, lo que aumenta el riesgo de complicaciones graves o incluso de muerte.

Para resolver esta problemática, se propone el desarrollo de un sistema práctico, económico y aplicable a cualquier tipo de geriátrico, que permita la detección inmediata de situaciones de riesgo para los pacientes. La solución planteada consiste en la instalación de un sistema de sensores basado en Arduino y el entrenamiento de algunos de ellos mediante Machine Learning para diferenciar patrones que puedan suponer algún riesgo para el paciente.

Este conjunto de sensores se conectará a un programa central destinado al personal sanitario, que contará con una interfaz gráfica fácil de usar. El programa permitirá visualizar el estado de todas las habitaciones del geriátrico en tiempo real, alertando al personal de manera inmediata si se detecta alguna anomalía, facilitando así una respuesta rápida y eficiente ante cualquier situación de riesgo.

**Palabras clave:** habitación, sensor, Arduino, geriátrico, pacientes, supervisión, Machine Learning, interfaz, salud, monitorización, asistencia.



# Índex general

<b>Resum</b>	<b>1</b>
<b>Abstract</b>	<b>2</b>
<b>Resumen</b>	<b>3</b>
<b>1 Introducció</b>	<b>8</b>
1.1 Motivació . . . . .	8
1.2 Estat de l'Art . . . . .	9
1.3 Objectius . . . . .	11
1.4 Estructura del Treball . . . . .	12
<b>2 Fonaments Tecnològics</b>	<b>13</b>
2.1 Hardware . . . . .	13
2.1.1 Arduino . . . . .	13
2.1.2 Raspberry Pi 4 . . . . .	16
2.2 Software . . . . .	18
2.2.1 TinyML i Edge Impulse . . . . .	18
2.2.2 Llibreries TensorFlow i Keras . . . . .	20
2.2.3 Tecnologia MQTT . . . . .	22
2.2.4 Llibreries addicionals per a la GUI i altres funcionalitats . . . . .	24
<b>3 Desenvolupament</b>	<b>25</b>
3.1 Hardware . . . . .	25
3.1.1 Components i funció . . . . .	25
3.1.2 La maqueta física . . . . .	26
3.1.3 Esquema de la disposició física . . . . .	27
3.2 Software . . . . .	28
3.2.1 Arduino per a la monitorització de la temperatura i la humitat . . . . .	28
3.2.2 Arduino per a la detecció de caigudes amb Edge Impulse . . . . .	29
3.2.3 MQTT i Raspberry Pi . . . . .	39
3.2.4 Interfície Gràfica (GUI) . . . . .	40

---

3.2.5	Integració de Components . . . . .	47
<b>4</b>	<b>Validació del sistema</b>	<b>49</b>
4.1	Validació de Sensors . . . . .	49
4.1.1	Validació del sensor de temperatura i humitat . . . . .	49
4.1.2	Validació de l'acceleròmetre per a la detecció de caigudes . . . . .	49
4.1.3	Robustesa dels Sensors . . . . .	50
4.2	Validació de la Comunicació . . . . .	50
4.2.1	Proves de Connexió entre Arduino i Raspberry Pi . . . . .	50
4.2.2	Proves de Publicació i Subscripció amb el Broker MQTT . . . . .	50
4.2.3	Validació de la Sincronització amb la GUI . . . . .	51
4.2.4	Proves d'Estabilitat i Robustesa de la Comunicació . . . . .	51
4.3	Validació de la Interfície Gràfica d'Usuari (GUI) . . . . .	52
4.4	Validació del Sistema Complet . . . . .	53
<b>5</b>	<b>Conclusions i Treball Futur</b>	<b>54</b>
5.1	Conclusions . . . . .	54
5.2	Treball Futur . . . . .	55
<b>6</b>	<b>Bibliografia</b>	<b>56</b>

## Índex de figures

1	Placa de desenvolupament anomenada Arduino UNO, una de les més utilitzades de la marca . . . . .	13
2	Placa Arduino Nano 33 BLE Sense, el model escollit per al projecte . . . . .	15
3	Raspberry Pi 4, un dels models més potents d'aquesta sèrie. . . . .	16
4	Ports USB i Ethernet . . . . .	17
5	Connectors Micro-HDMI . . . . .	17
6	Representació gràfica de com funciona TinyML (Medium, n.d.-a) . . . . .	18
7	Diagrama del flux de treball d'Edge Impulse (Edge Impulse, 2024) . . . . .	19
8	Logotips de TensorFlow i Keras, dos llibreries clau en l'àmbit del Machine Learning. . . . .	20
9	Flux de treball de TensorFlow Lite, versió específica per a TinyML (Cavagnis, 2023) . . . . .	21
10	Esquema simple del funcionament de MQTT (Medium, n.d.-b) . . . . .	22
11	Maqueta física creada per al projecte, que simula una habitació de centre geriàtric. . . . .	26
12	Esquema de la distribució dels components dins de la maqueta física. . . . .	27
13	Sensor HTS221, el qual es troba integrat a la placa Arduino. . . . .	28
14	Maqueta on s'han realitzta les mostres . . . . .	29
15	Ubicació de l'Arduino a la part inferior de la maqueta . . . . .	29
16	Mostra de la classe "passos" . . . . .	30
17	Mostra de la classe "res" . . . . .	30
18	Percentatges del train/test split . . . . .	30
19	Pantalla de creació de l'Impulse a Edge Impulse . . . . .	31
20	Esquema orientatiu de l'estructura de la xarxa neuronal . . . . .	32
21	Gràfica de l'evolució del <i>Loss</i> a l'entrenament i la validació durant l'entrenament. . . . .	34
22	Exploració de les dades del Training Set. . . . .	34
23	Distribució de les dades d'entrenament . . . . .	35
24	Exemple d'una mostra de caiguda . . . . .	35
25	Classificació de la mostra de passos . . . . .	36
26	Classificació d'anomalia de la mostra de passos . . . . .	36
27	Classificació de la mostra de caiguda . . . . .	36
28	Classificació d'anomalia de la mostra de caiguda . . . . .	36

29	Desplegament en forma de llibreria d'Arduino seleccionat en Edge Impulse . . . .	37
30	Pantalla principal del programa, on es mostren les habitacions del geriàtric . . . .	40
31	Valors anòmals de temperatura o humitat . . . . .	42
32	Possible caiguda detectada . . . . .	42
33	Finestra on es mostren les dades de temperatura i humitat en temps real . . . .	42
34	Pantalla de configuració del programa . . . . .	43
35	Distribució amb 9 habitacions . . . . .	43
36	Distribució amb 6 habitacions . . . . .	43
37	Pestanya amb les isnruccions del programa . . . . .	44
38	Notificació de caiguda . . . . .	44
39	Pestanya de caiguda atesa . . . . .	44
40	Pantalla inicial amb el logotip del programa . . . . .	45
41	Diagrama de la integració dels components del sistema. . . . .	48
42	Exemple d'una prova de missatges MQTT per terminal . . . . .	52
43	Imatge de la prova amb el sistema complet amb tots els elements connectats. . .	53

## Índex de taules

1	Resultats de l'Avaluació del Model (Int8) . . . . .	33
2	Resultats de l'Avaluació del Model (Float32) . . . . .	33
3	Resultats del Test del Model (Float32) . . . . .	33
4	Relació del treball amb els ODS . . . . .	63

# 1 Introducció

## 1.1 Motivació

En l'àmbit de la geriatria, garantir la seguretat i el benestar de les persones majors és una prioritat que sovint es veu compromesa per situacions crítiques com les caigudes. Segons dades de l'INE, l'any passat es van registrar 4.018 defuncions per caigudes accidentals a Espanya, superant per primera vegada els suïcidis com a principal causa de mort externa. Més del 80% d'aquestes morts van afectar persones majors de 65 anys, especialment dones (Institut Nacional d'Estadística (INE), 2023).

Aquestes xifres evidencien la gravetat del problema i la necessitat d'intervencions efectives per prevenir les caigudes i les seves conseqüències. Les caigudes no ateses amb un temps de resposta adequat poden derivar en complicacions greus i, fins i tot, en la mort. Així doncs, aquest projecte es planteja amb l'objectiu clar de salvar vides, millorant la capacitat de resposta davant d'aquests incidents i augmentant la seguretat dels residents en els centres geriàtrics.

A més de les caigudes, altres factors com la incontinència i les temperatures corporals anòmales també poden suposar riscos importants per a la salut de les persones majors. La incontinència urinària afecta al voltant del 20-25% de les persones majors de seixanta-cinc anys; és més freqüent en dones i s'incrementa amb l'edat. Si no és atesa de manera ràpida, pot generar complicacions com irritacions cutànies, úlceres per pressió i infeccions del tracte urinari (Conselleria de Sanitat Universal i Salut Pública, 2021). Aquest projecte pretén detectar aquests episodis d'humitat de manera immediata, permetent al personal reaccionar ràpidament.

Un altre problema potencialment perillós per a les persones majors és la presència de temperatures corporals anòmales, que poden ser un indicador de febre o d'hipotèrmia. La febre pot ser signe d'infeccions, que poden desenvolupar-se amb rapidesa i poden derivar en situacions crítiques si no es detecten i tracten a temps. D'altra banda, la hipotèrmia pot ocórrer en persones majors fins i tot en temperatures ambientals moderades. La detecció d'aquestes situacions permet una intervenció immediata, evitant riscos greus per a la salut dels pacients.

A més, aquest sistema es dissenya amb un enfocament especial en la privacitat, un valor fonamental per a la gent major, que sol sentir incomoditat davant de sistemes de vigilància que podrien fer-los sentir observats. Per això, es pretén implementar un sistema que siga discret i que funcione de forma no intrusiva, evitant mètodes com les càmeres de vigilància o altres sistemes intrusius que podrien interferir en la intimitat dels residents.

Finalment, aquest projecte es planteja com una solució assequible i fàcil d'implementar, amb la intenció que pugui ser adaptat a qualsevol centre geriàtric, independentment dels seus recursos. Això comporta la utilització de dispositius de baix cost, configuracions senzilles i una instal·lació que pugui realitzar-se sense necessitat de modificacions complexes a les instal·lacions.

## 1.2 Estat de l'Art

A continuació, es descriuen diversos sistemes destacats que aborden la problemàtica de la monitorització per a millorar la qualitat de vida de les persones majors des de diferents perspectives tecnològiques i metodològiques, així com la comparació d'aquests sistemes amb la meua proposta.

### **Xarxes de sensors inalàmbrics per a la monitorització mèdica**

Un dels enfocaments més comuns és l'ús de xarxes de sensors inalàmbrics (Wireless Sensor Networks, WSN) per a la recollida de dades fisiològiques i ambientals dels pacients. Aquests sistemes solen incloure sensors portables que mesuren variables com la freqüència cardíaca, la temperatura corporal o els nivells d'oxigen en sang. Les dades es transmeten a una plataforma centralitzada mitjançant protocols com ZigBee o Bluetooth, on són analitzades per alertar possibles anomalies.

Per exemple, en un projecte de la Universitat Oberta de Catalunya, es va dissenyar un sistema basat en sensors portables i Wi-Fi per detectar caigudes i monitoritzar variables vitals. Aquest sistema destaca pel seu enfocament en l'atenció remota, oferint un seguiment en temps real a través d'una interfície web. Tot i això, depèn de la portabilitat dels sensors, que pot ser un factor limitant per a pacients amb problemes de mobilitat o demències avançades (López Chaveinte, 2015).

### **InvisiCare: Monitorització no intrusiva mitjançant l'anàlisi de patrons d'ús**

Un altre sistema innovador és InvisiCare, desenvolupat com a una solució completament no intrusiva per a la gent gran que viu de forma independent o en residències geriàtriques. Aquest sistema no utilitza sensors físics; en lloc d'això, analitza dades d'ús de serveis com electricitat, aigua o calefacció per detectar canvis en els patrons de comportament. Per exemple, un descens sobtat en el consum d'aigua podria indicar que el pacient no s'ha aixecat del llit o que ha patit una caiguda (Fundació iSocial, 2022).

Tot i ser una solució molt atractiva per la seva discreció, InvisiCare presenta limitacions pel que fa a la precisió, ja que els patrons d'ús poden variar per múltiples raons alienes a l'estat de salut del pacient. A més, aquest tipus de monitorització pot ser insuficient en situacions d'urgència, com una caiguda que requereix assistència immediata.

### **Sistemes basats en iBeacons per a la monitorització de moviment**

Els sistemes que utilitzen iBeacons es basen en la tecnologia Bluetooth Low Energy per detectar la presència i el moviment dels pacients dins d'una àrea determinada. Les balises es col·loquen estratègicament a les habitacions i passadissos, enviant senyals a un receptor central que registra les activitats dels residents. Aquest enfocament és ideal per a monitoritzar el moviment i prevenir situacions com la desorientació o les fugues en pacients amb Alzheimer.

Un exemple destacat és un projecte desenvolupat per investigadors de la Universitat de Zaragoza, on les iBeacons es van implementar amb èxit per detectar patrons de moviment irregulars. No obstant això, aquest sistema no inclou la capacitat de mesurar altres variables fisiològiques i pot ser poc adequat per a monitoritzar pacients que passen molt temps en repòs (Universitat de Saragossa, 2019).

## Sistemes basats en la visió per computador

Un altre tipus de tecnologia emergent són els sistemes que utilitzen càmeres i algorismes de visió per computador per detectar caigudes, patrons de moviment irregulars o fins i tot expressions facials associades al dolor. Aquests sistemes, com el desenvolupat per investigadors del Centre Tecnològic Cartif; ofereixen una monitorització completa de l'entorn sense necessitat que el pacient interactuï directament amb el dispositiu (Centre Tecnològic Cartif, 2021).

Aquest enfocament és especialment útil en espais comunitaris com menjadors o sales de reunions, però pot generar preocupacions relacionades amb la privacitat i el processament d'imatges en temps real. A més, els costos associats a la instal·lació i manteniment de càmeres d'alta resolució poden ser elevats.

## Sistemes híbrids de monitorització

Molts sistemes combinen diferents tecnologies per oferir una solució més completa. Per exemple, hi ha solucions que integren sensors en matalassos per monitoritzar el moviment i la freqüència respiratòria durant el son, juntament amb càmeres o detectors de caigudes a les habitacions. Aquest tipus de sistemes són especialment adequats per a pacients amb mobilitat limitada o problemes de salut complexos.

Un exemple destacat és el desenvolupat per investigadors de la Universitat Politècnica de València, que combina sensors de pressió al llit amb micròfons ambientals per detectar anomalies en el comportament nocturn. Aquest sistema ofereix un alt nivell de precisió, però requereix una infraestructura tecnològica significativa (Universitat Politècnica de València, 2023).

## Diferenciació del sistema proposat

El sistema desenvolupat en aquest treball es diferencia de tots els anteriors per la seva filosofia basada en tres principis: no intrusivitat, assequibilitat i eficiència. A diferència de sistemes com els basats en sensors portables o càmeres, el nostre sistema evita la interacció directa amb el pacient, instal·lant els sensors de manera estratègica al llit i al terra de l'habitació. Això garanteix una experiència més còmoda per al pacient i redueix possibles problemes d'adherència al sistema.

A més, l'ús de components com Arduino Nano 33 BLE Sense i la integració amb la Raspberry Pi permet oferir una solució tecnològica avançada a un cost significativament inferior al d'altres sistemes híbrids o basats en visió per computador. Aquesta característica fa que el sistema proposat siga especialment atractiu per a la seva implementació en centres geriàtrics amb recursos limitats.

Finalment, el sistema destaca per la seva capacitat de proporcionar informació en temps real sobre diferents variables, incloent temperatura corporal, humitat del llit i detecció de caigudes mitjançant *machine learning*. Aquesta combinació única d'eficiència i abast el posiciona com una solució innovadora i pràctica dins del panorama actual de la monitorització de pacients en entorns geriàtrics.

### 1.3 Objectius

- **Objectiu general:** Dissenyar i implementar un sistema de monitorització intel·ligent, senzill i no intrusiu per a habitacions de pacients en centres geriàtrics. Aquest sistema ha de permetre detectar situacions de risc per als residents, com ara caigudes, episodis d'incontinència o temperatures corporals anòmales, amb la finalitat de millorar la seguretat dels pacients i ajudar al personal sanitari en les seues tasques de supervisió i resposta immediata davant d'emergències.
- **Objectius específics:**
  - Mesura de paràmetres d'humitat i temperatura: Desenvolupar un sistema de monitorització de temperatura corporal i humitat del llit per a detectar situacions que pogueren comprometre la salut i el benestar del pacient.
  - Detecció de caigudes: Implementar un sistema de detecció de caigudes que siga capaç de diferenciar entre moviments normals i situacions de risc, identificant patrons de moviment que indiquen una possible caiguda i alertant de forma immediata el personal sanitari.
  - Alertes en temps real i comunicació amb el personal: Desenvolupar un sistema de comunicació que connecte els dispositius de monitorització amb una interfície gràfica en temps real, enviant alertes de manera immediata quan es detecten situacions de risc.
  - No intrusivitat i privacitat: Assegurar que el sistema funcione de manera discreta, evitant tecnologies que puguen fer sentir els pacients observats, respectant així la seua intimitat mentre es manté un control eficaç de les condicions de seguretat.
  - Baix cost i facilitat d'implementació: Utilitzar components assequibles per a garantir que el sistema siga fàcil d'instal·lar en qualsevol centre geriàtric, independentment de les seues limitacions de recursos.



## 1.4 Estructura del Treball

Aquesta memòria està estructurada en diferents punts per oferir una visió clara i organitzada del projecte, des de la motivació i els fonaments fins a l'aplicació pràctica i els resultats obtinguts.

La [secció 1](#), presenta la motivació per a la realització d'aquest projecte, el context i els objectius plantejats. A més, ofereix una revisió de l'estat de l'art per situar el projecte en relació amb altres solucions existents en l'àmbit de la monitorització de pacients en geriatria.

La [secció 2](#), descriu les tecnologies de hardware i software utilitzades en el desenvolupament del sistema. Aquesta secció inclou una explicació de les eines que conformen la base tècnica del projecte, com Arduino, Raspberry Pi, el sistema de classificació TinyML, les llibreries de Machine Learning, així com les tecnologies de comunicació com MQTT. Aquest capítol proporciona una visió general sobre les característiques de cada eina.

La [secció 3](#), detalla el procés d'implementació del sistema. En aquest capítol es descriu de forma exhaustiva com s'han aplicat les tecnologies descrites en el capítol anterior per a construir el sistema de monitorització. Es presenten els diferents components de hardware, com els sensors d'Arduino i la Raspberry Pi, i es mostren les configuracions utilitzades per a la gestió de dades, la detecció d'anomalies i el sistema d'alertes en temps real.

En la [secció 4](#), s'analitzen els resultats obtinguts durant les proves de funcionament del sistema. Aquesta secció inclou una anàlisi de la precisió i l'efectivitat de la detecció d'anomalies, així com una avaluació del rendiment del sistema en termes de latència de resposta i fiabilitat.

Finalment, en la [secció 5](#), es fa una valoració global del projecte, destacant els objectius assolits, els punts forts i les possibles limitacions del sistema. A més, es plantegen diverses millores i propostes de treball futur que podrien ampliar o optimitzar el sistema desenvolupat.

## 2 Fonaments Tecnològics

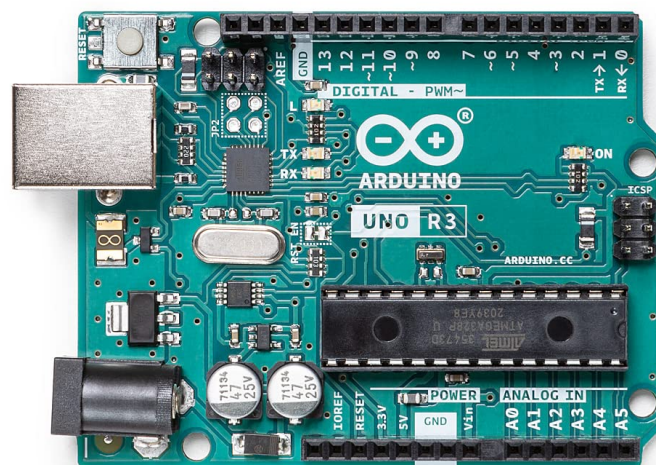
Aquest capítol explora les tecnologies fonamentals que sustenten el projecte, incloent el hardware i el software utilitzats per implementar el sistema. Es descriuen les plataformes de desenvolupament, protocols de comunicació i eines de programació seleccionades, destacant els seus avantatges i com s'alineen amb els objectius del projecte. Així mateix, es proporciona una visió detallada de les funcionalitats i característiques que fan que aquestes tecnologies siguin adequades per a una aplicació en centres geriàtrics. L'objectiu d'aquest capítol és establir la base tècnica sobre la qual s'ha construït el sistema.

### 2.1 Hardware

#### 2.1.1 Arduino

L'Arduino és una plataforma de prototipat electrònic de codi obert que inclou tant hardware com software fàcil d'utilitzar per a crear projectes interactius. La seva arquitectura i la disponibilitat de múltiples models de microcontroladors permeten que aquesta plataforma s'adapte a una gran varietat d'aplicacions, des de projectes educatius fins a sistemes avançats d'automatització industrial. L'Arduino és especialment útil en projectes que requereixen un control i monitorització de sensors, com és el cas d'aquest projecte (Arduino, 2024c).

Els microcontroladors d'Arduino han guanyat popularitat a causa de la seua robustesa, flexibilitat i la seua comunitat activa, que proporciona suport i múltiples recursos en línia. La seva capacitat per connectar-se a diferents sensors i actuadors, així com la possibilitat de programar-los de manera intuïtiva amb el llenguatge de programació basat en C/C++, fa que siguin una opció molt viable per a aquest projecte. A més, la facilitat d'integració amb altres plataformes i sistemes com Raspberry Pi, proporciona un entorn ideal per a implementar un sistema de monitorització senzill, no intrusiu i d'alt rendiment, amb una baixa latència per a la transmissió de dades.



**Figura 1:** Placa de desenvolupament anomenada Arduino UNO, una de les més utilitzades de la marca

Per a aquest projecte de monitorització de pacients geriàtrics, l'elecció d'Arduino presenta diversos avantatges. Primerament, és una plataforma de baix cost, fet que s'alineja amb l'objectiu de crear un sistema assequible i fàcilment replicable en centres geriàtrics amb recursos limitats. A més, la seua arquitectura i compatibilitat amb una àmplia gamma de sensors, tant digitals com analògics, permet implementar de manera senzilla la detecció de variables com la temperatura i la humitat del llit, així com el moviment en el sòl de l'habitació, amb l'objectiu de detectar caigudes. Arduino també proporciona una fiabilitat alta en la recollida de dades en temps real, fet essencial per a aquest projecte, ja que es tracta d'un sistema que busca prevenir riscos per a la salut dels pacients.

Un altre avantatge significatiu és la facilitat de programació. La llibreria de desenvolupament de l'Arduino IDE proporciona una gran quantitat de recursos i llibreries que permeten desenvolupar codi de forma eficient i adaptar-lo a les necessitats específiques del projecte (Arduino, 2024a). La disponibilitat d'aquestes llibreries facilita la integració de sensors i actuadors en el sistema. A més, la plataforma Arduino és capaç d'operar en condicions de baix consum energètic, cosa que resulta crucial per a sistemes que poden requerir una alimentació constant sense interrupcions.

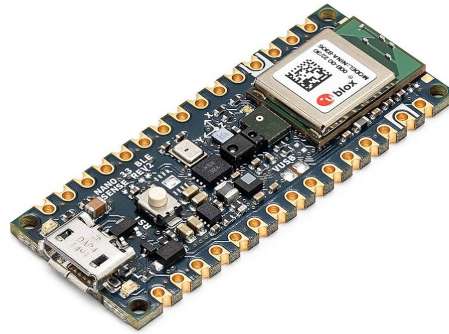
Dins de la gamma de dispositius Arduino, el model **Arduino Nano 33 BLE Sense** ha estat seleccionat per a aquest projecte per les seves característiques específiques que s'alineen perfectament amb els requeriments de la monitorització de pacients geriàtrics. Aquest model ofereix un conjunt de sensors integrats i una gran capacitat per al processament de dades, a més d'un disseny compacte, que facilita la seua integració en espais reduïts sense comprometre la comoditat del pacient (Arduino, 2024b).

Entre els sensors integrats en l'**Arduino Nano 33 BLE Sense** es troben:

**Sensor de temperatura i humitat (HTS221):** Aquest sensor permet mesurar tant la temperatura ambiental com la humitat relativa amb una alta precisió. El rang de temperatura operatiu és de  $-40\text{ }^{\circ}\text{C}$  a  $120\text{ }^{\circ}\text{C}$ , amb una resolució de  $0,01\text{ }^{\circ}\text{C}$ , mentre que la humitat relativa es mesura amb una precisió del  $\pm 3,5\%$  i un rang que va del  $0\%$  al  $100\%$ . Aquest sensor és fonamental per al projecte, ja que permet monitoritzar la humitat i la temperatura del llit del pacient, proporcionant indicadors clau per detectar incontinència o anomalies de temperatura corporal. El sensor HTS221 està connectat al controlador utilitzant el protocol de comunicació **I2C**, que garanteix una transmissió eficient de dades amb només dues línies compartides entre múltiples dispositius.

**Acceleròmetre de 3 eixos (LSM9DS1):** Aquest sensor detecta moviments i oscil·lacions en tres dimensions (x, y, z), la qual cosa el fa ideal per identificar patrons de moviment anòmals que poden indicar una caiguda. L'acceleròmetre té un rang ajustable d'operació des de  $\pm 2\text{ g}$  fins a  $\pm 16\text{ g}$  i una sensibilitat que garanteix una detecció precisa de moviments bruscos. Aquesta característica és essencial per assegurar una resposta ràpida davant d'una possible caiguda, ajudant a activar alertes immediates per al personal sanitari. En aquest cas, el sensor LSM9DS1 es comunica amb l'Arduino mitjançant el protocol **SPI**, el qual ofereix una velocitat de transmissió més alta i una latència mínima, assegurant una resposta ràpida i fiable.

La integració d'aquests sensors en l'Arduino Nano 33 BLE Sense permet no només capturar dades amb una alta fiabilitat, sinó també processar-les directament en el dispositiu gràcies a la seva capacitat per executar models optimitzats de **TinyML**. Aquesta funcionalitat és especialment útil en aplicacions on el temps de resposta és crític i la privacitat de les dades és fonamental, ja que minimitza la necessitat de transmetre dades sensibles a servidors externs.



*Figura 2: Placa Arduino Nano 33 BLE Sense, el model escollit per al projecte*

L'elecció d'aquest model presenta diversos avantatges, tant per les seves capacitats com per la seua compatibilitat amb els objectius d'aquest projecte (Arduino Team, 2023). Entre aquests avantatges es troba:

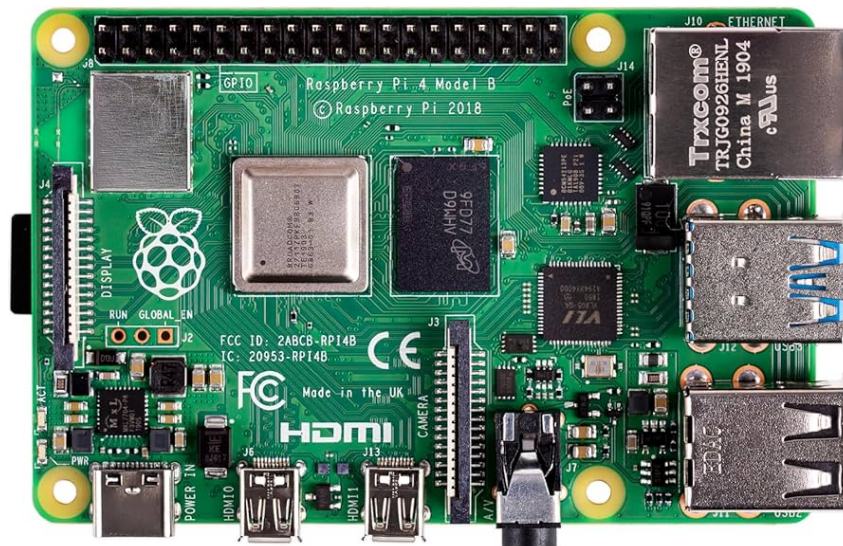
- **Processament integrat de dades:** L'Arduino Nano 33 BLE Sense pot executar models de TinyML, el que permet realitzar una primera fase d'anàlisi de dades directament en el microcontrolador, reduint la càrrega de dades a transmetre.
- **Compatibilitat amb sensors interns:** La disponibilitat de sensors integrats en el mateix microcontrolador evita la necessitat de connectar components externs addicionals, fet que simplifica el disseny físic i redueix els costos de manteniment.
- **Disseny compacte i adaptable:** El disseny de l'Arduino Nano 33 BLE Sense permet col·locar-lo en el llit o en el sòl de l'habitació del pacient de manera discreta, assegurant la no intrusivitat del sistema.
- **Cost accessible:** Tot i les seues capacitats avançades, l'Arduino Nano 33 BLE Sense es manté en una franja de preu assequible, fet que encaixa amb l'objectiu del projecte de crear una solució viable per a centres geriàtrics.

Aquesta combinació de capacitats i avantatges ha fet que el model Arduino Nano 33 BLE Sense siga el dispositiu òptim per a aquest projecte, en la mesura que permet la monitorització fiable i discreta de les condicions dels pacients, amb una capacitat de resposta ràpida davant d'anomalies.

### 2.1.2 Raspberry Pi 4

La **Raspberry Pi** és una sèrie d'ordinadors de placa reduïda desenvolupada per la Fundació Raspberry Pi amb l'objectiu inicial de promoure l'educació en ciències de la computació en centres educatius i països en vies de desenvolupament. Amb el temps, aquesta plataforma s'ha convertit en una eina fonamental per a una gran varietat de projectes gràcies a la seua versatilitat, el seu cost assequible i la seua capacitat de processament. Actualment, la Raspberry Pi s'utilitza àmpliament en camps com l'Internet de les Coses (IoT), la robòtica, l'automatització industrial i la monitorització en temps real (Raspberry Pi Foundation, 2024a).

A diferència de microcontroladors com Arduino, que estan dissenyats principalment per controlar sensors i actuadors, la Raspberry Pi és un ordinador complet capaç d'executar sistemes operatius basats en Linux i aplicar programari complex. Això permet utilitzar-la en aplicacions que requereixen un processament intensiu de dades o la gestió de xarxes de comunicació. La Raspberry Pi disposa d'una àmplia gamma de ports i connectors, com ara GPIO, USB, HDMI i Ethernet, que li atorguen una gran capacitat d'interconnexió amb altres dispositius.



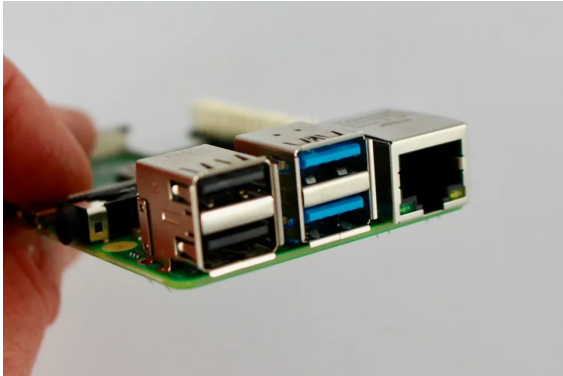
*Figura 3: Raspberry Pi 4, un dels models més potents d'aquesta sèrie.*

El model seleccionat per a aquest projecte és la **Raspberry Pi 4**, que destaca per ser una de les versions més avançades de la sèrie. Introduïda al mercat el 2019, la Raspberry Pi 4 presenta diverses millores respecte als seus predecessors. Està equipada amb un processador Broadcom BCM2711 Cortex-A72 de 64 bits amb quatre nuclis que funcionen a una velocitat de 1.5 GHz, fet que li permet gestionar múltiples tasques de manera simultània. Pel que fa a la memòria RAM, aquest model ofereix diverses opcions, que van des de 2 GB fins a 8 GB. Per a aplicacions amb requisits moderats de processament, com és aquest projecte, la versió de 2 GB és més que suficient.

Un dels aspectes més destacats de la Raspberry Pi 4 és la seva capacitat de connectivitat. Incorpora Wi-Fi 802.11ac, Bluetooth 5.0 i Ethernet Gigabit, característiques que asseguruen una comunicació ràpida i estable amb altres dispositius de la xarxa local. Així mateix, compta amb 40 pins GPIO, que li permeten connectar-se fàcilment a sensors i altres components externs, oferint un ampli marge d'expansió (Raspberry Pi Foundation, 2024b).



La compatibilitat amb protocols de comunicació com MQTT la converteix en una eina ideal per a projectes de monitorització en temps real. A més, disposa de dos ports micro-HDMI, que li permeten connectar fins a dues pantalles amb resolucions 4K, així com diversos ports USB, incloent-hi dos USB 3.0 per a transferències de dades d'alta velocitat.



*Figura 4: Ports USB i Ethernet*



*Figura 5: Connectors Micro-HDMI*

L'elecció de la **Raspberry Pi 4** es basa en les seues capacitats tècniques i la seua adaptabilitat als requisits d'aquest projecte. A continuació es detallen els motius principals que la fan ideal per a la supervisió de pacients geriàtrics:

- **Gestió de comunicacions:** La capacitat de suportar protocols com MQTT permet a la Raspberry Pi transmetre dades en temps real amb un consum mínim de recursos. Això és essencial per garantir la fiabilitat i rapidesa en les alertes.
- **Escalabilitat:** Gràcies als ports GPIO i la seua capacitat d'execució de programari complet, la Raspberry Pi pot ser ampliada fàcilment amb nous sensors o funcionalitats en el futur.
- **Cost assequible i eficiència energètica:** Tot i les seues capacitats avançades, la Raspberry Pi 4 manté un cost assequible i un consum energètic moderat, factors importants per a centres geriàtrics amb pressupostos limitats.
- **Execució autònoma:** La Raspberry Pi pot operar de manera independent, gestionant el processament de dades i les comunicacions sense dependre de servidors externs, fet que millora la robustesa i fiabilitat del sistema.

La Raspberry Pi 4 no només és un dispositiu de gran capacitat, sinó que també manté un consum energètic moderat i un cost assequible, especialment en comparació amb altres sistemes informàtics de característiques similars. Aquest equilibri entre rendiment i preu la converteix en una opció molt adequada per a sistemes de monitorització que necessiten processar dades de manera contínua sense comprometre el pressupost.

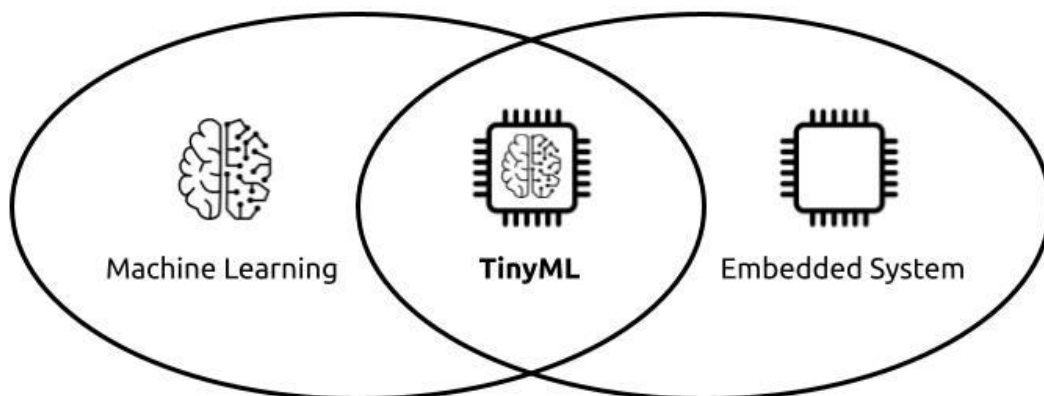
La compatibilitat amb múltiples llenguatges de programació i la seua àmplia comunitat de desenvolupadors són altres factors que fan que la Raspberry Pi 4 siga una de les opcions preferides per a projectes tecnològics. Les seues capacitats permeten abordar tant aplicacions senzilles com sistemes més complexos que requereixen una alta escalabilitat. Aquest model ofereix, per tant, la robustesa necessària per integrar-se com una peça fonamental en sistemes de supervisió de dades com el que es desenvolupa en aquest projecte.

## 2.2 Software

### 2.2.1 TinyML i Edge Impulse

La tecnologia **TinyML** (Tiny Machine Learning) és una branca de l'aprenentatge automàtic que se centra en l'execució de models de Machine Learning en dispositius de baix consum i capacitat de càlcul limitada, com microcontroladors i sensors integrats. L'objectiu principal de TinyML és permetre l'anàlisi de dades i la detecció d'anomalies en temps real sense necessitat d'enviar les dades a un servidor extern per al processament. Aquesta capacitat de processament local és especialment valuosa en aplicacions d'Internet de les Coses (IoT), on es requereix una resposta immediata i sovint s'opera amb limitacions d'energia o de connectivitat (DataCamp, 2024).

Per a aquest projecte, TinyML ofereix una solució òptima per integrar un sistema de detecció d'anomalies directament en el microcontrolador **Arduino Nano 33 BLE Sense**. Aquesta capacitat permet una resposta ràpida davant de situacions de risc sense dependre completament d'un servidor central, contribuint a un sistema de monitorització més robust i autònom, alhora que es redueix la latència i es millora la fiabilitat de la detecció de situacions perilloses per als pacients.

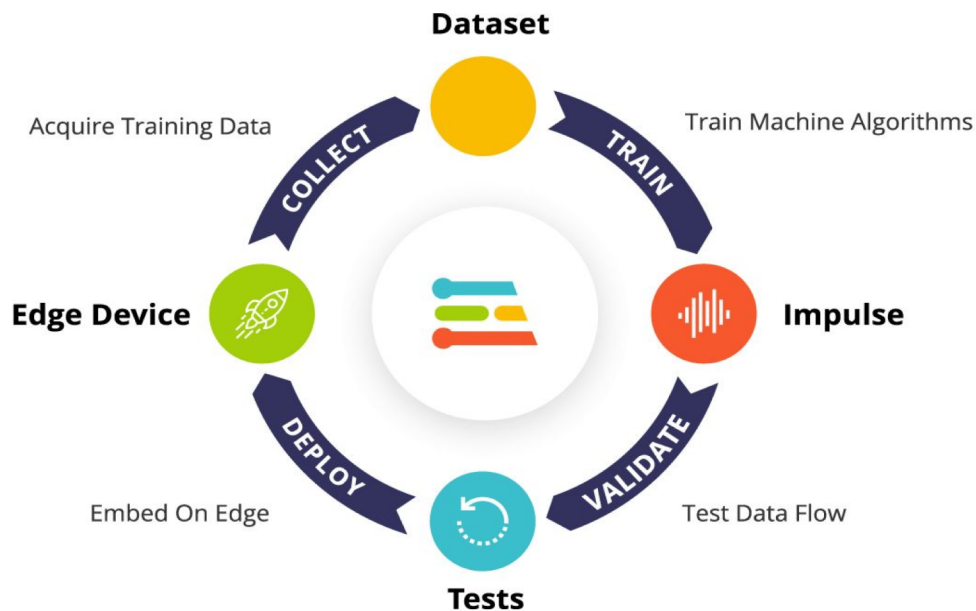


*Figura 6: Representació gràfica de com funciona TinyML (Medium, n.d.-a)*

TinyML està integrat amb llibreries com **TensorFlow Lite for Microcontrollers** i **Keras**, de les quals es parla en la [subsecció 2.2.2](#), que faciliten el desenvolupament i desplegament de models optimitzats per a microcontroladors. Aquestes eines permeten implementar algorismes avançats de detecció de patrons i anomalies amb un consum mínim de recursos, garantint alhora una interfície de programació intuïtiva per als desenvolupadors.

En aquest projecte, la tecnologia TinyML es complementa amb **Edge Impulse**, una plataforma de desenvolupament de Machine Learning dissenyada específicament per a dispositius amb capacitats limitades. Edge Impulse facilita tot el procés de creació de models TinyML, des de la recopilació de dades fins a la generació de codi optimitzat per a microcontroladors (Edge Impulse, 2024). Aquesta eina ha estat fonamental per entrenar el model de detecció de caigudes basat en dades de moviment, així com per optimitzar els algorismes de detecció d'anomalies, com les oscil·lacions brusques que poden indicar un episodi de risc per al pacient.

El procés de treball amb Edge Impulse es divideix en diverses etapes clau: recopilació i etiquetatge de dades, processament previ, entrenament del model i generació de codi per al desplegament. La plataforma permet enregistrar dades directament des de l'Arduino Nano 33 BLE Sense, etiquetar-les segons els diferents patrons detectats, com moviments normals o caigudes, i entrenar un model adaptat a les necessitats específiques del projecte. A més, genera automàticament un codi optimitzat per al microcontrolador, que inclou el model entrenat i les configuracions necessàries per a la seva execució.



*Figura 7: Diagrama del flux de treball d'Edge Impulse (Edge Impulse, 2024)*

Les principals característiques de TinyML i Edge Impulse que les fan idònies per a aquest projecte són:

- **Eficiència energètica:** Els models TinyML estan optimitzats per consumir el mínim d'energia possible, fet que permet una operació prolongada en dispositius amb alimentació limitada.
- **Temps de resposta curt:** En processar les dades directament en el microcontrolador, es garanteix una detecció ràpida d'anomalies, fonamental per a la monitorització de caigudes i altres riscos.
- **Privacitat de les dades:** Processar la informació al dispositiu evita la necessitat d'enviar dades a servidors externs, augmentant la privacitat dels pacients.
- **Facilitat d'implementació:** Edge Impulse proporciona eines senzilles per a tot el procés de desenvolupament, des de la recopilació de dades fins a la implementació del model en el dispositiu.

Gràcies a l'ús de TinyML i Edge Impulse, aquest projecte aconsegueix integrar un sistema de detecció d'anomalies eficient, fiable i energèticament sostenible.



### 2.2.2 Llibreries TensorFlow i Keras

**TensorFlow** és una llibreria de codi obert desenvolupada per Google destinada a la implementació de models d'aprenentatge automàtic i xarxes neuronals. Aquesta eina permet la creació, entrenament i desplegament de models d'aprenentatge profund (*Deep Learning*) amb un alt rendiment, tant en entorns locals com al núvol. La seva flexibilitat i robustesa han convertit TensorFlow en un estàndard de la indústria per a aplicacions d'intel·ligència artificial (IA) en àmbits com la visió per ordinador, el processament del llenguatge natural i l'Internet de les Coses (IoT) (TensorFlow, 2024b).

**Keras**, per altra banda, és una API d'alt nivell integrada dins de TensorFlow que simplifica el desenvolupament de models d'aprenentatge automàtic, oferint una interfície intuïtiva i fàcil d'usar. Aquesta capa d'abstracció permet dissenyar xarxes neuronals complexes sense necessitat de gestionar directament les operacions matemàtiques de baix nivell, proporcionant alhora flexibilitat per personalitzar els models segons les necessitats del projecte (Keras, 2024).

En el context d'aquest projecte, TensorFlow i Keras són eines fonamentals per al desenvolupament i implementació de models de detecció d'anomalies optimitzats per a executar-se en dispositius de baix consum com l'**Arduino Nano 33 BLE Sense**. Gràcies a aquestes llibreries, és possible integrar models preentrenats o personalitzats dins del sistema, assegurant un rendiment òptim amb recursos limitats.

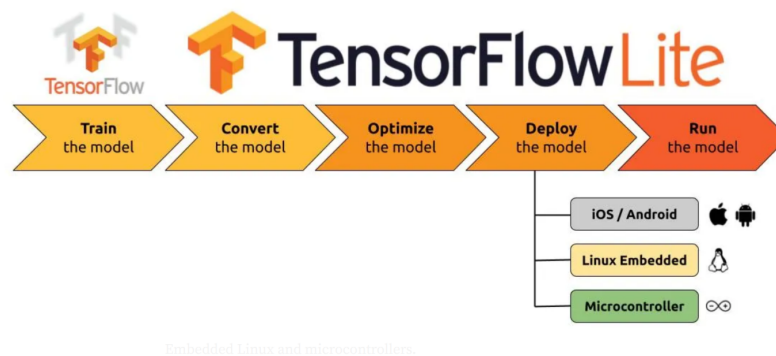


*Figura 8: Logotips de TensorFlow i Keras, dos llibreries clau en l'àmbit del Machine Learning.*

Les característiques principals de TensorFlow i Keras son les següents:

- **Ecosistema complet:** TensorFlow ofereix eines per al desenvolupament, entrenament i desplegament de models en diverses plataformes, incloent microcontroladors, dispositius mòbils i entorns al núvol.
- **Compatibilitat amb microcontroladors:** Mitjançant **TensorFlow Lite for Microcontrollers**, els models desenvolupats poden ser optimitzats per a dispositius amb recursos limitats, com els utilitzats en aquest projecte.
- **Facilitat d'ús:** Keras proporciona una interfície amigable per al desenvolupament de xarxes neuronals, permetent una implementació ràpida i senzilla.
- **Suport comunitari i documentació:** La popularitat de TensorFlow i Keras garanteix una extensa comunitat d'usuaris i una documentació detallada, que facilita la resolució de problemes i l'accés a exemples pràctics.

Aquesta combinació de característiques fa que TensorFlow i Keras siguin llibreries ideals per a aquest projecte, especialment per al desenvolupament de models de detecció d'anomalies que s'executen directament en el maquinari del sistema.



*Figura 9: Flux de treball de TensorFlow Lite, versió específica per a TinyML (Cavagnis, 2023)*

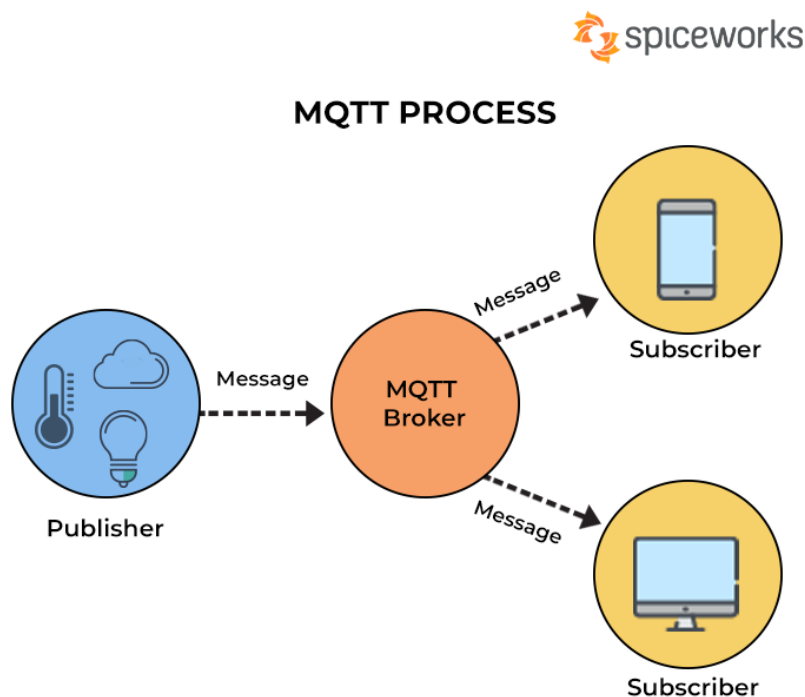
**TensorFlow Lite** és una variant de TensorFlow dissenyada específicament per a executar models d'aprenentatge automàtic en dispositius amb capacitats limitades de memòria i processament, com microcontroladors, sensors intel·ligents i dispositius mòbils. Aquesta llibreria optimitza els models d'aprenentatge automàtic per reduir-ne la mida i el consum de recursos, permetent-ne l'execució de manera eficient fins i tot en entorns amb restriccions d'energia o connectivitat. A més, ofereix funcionalitats com la quantificació de models, que disminueix els requeriments de precisió dels càlculs matemàtics sense comprometre significativament l'eficàcia dels models. Gràcies a la seua compatibilitat amb **TensorFlow Lite for Microcontrollers**, és possible desplegar models optimitzats en dispositius com l'**Arduino Nano 33 BLE Sense**, assegurant una detecció d'anomalies en temps real i amb un mínim impacte en el consum energètic. Aquesta capacitat de desplegament directe és especialment rellevant per a sistemes IoT, com el desenvolupat en aquest projecte, on l'eficiència i l'autonomia són essencials (TensorFlow, 2024a).

Gràcies a l'ús combinat de TensorFlow, Keras i TensorFlow Lite for Microcontrollers, aquest projecte pot implementar models d'aprenentatge automàtic avançats en dispositius de baix consum, assegurant una detecció precisa i ràpida d'anomalies en temps real.

### 2.2.3 Tecnologia MQTT

**MQTT** (Message Queuing Telemetry Transport) és un protocol de comunicació basat en un model de publicació i subscripció que ha estat dissenyat específicament per a dispositius amb recursos limitats i xarxes amb una amplada de banda reduïda. Aquest protocol, desenvolupat inicialment per IBM en els anys 90, és àmpliament utilitzat en aplicacions de l'Internet de les Coses (IoT) gràcies a la seua simplicitat i eficiència. MQTT es caracteritza per ser extremadament lleuger, fet que el fa ideal per a dispositius que operen amb limitacions d'energia o connectivitat intermitent, com ara sensors, microcontroladors o petits ordinadors de placa única (MQTT.org, 2024).

El funcionament de MQTT es basa en la figura del *broker*, un servidor intermediari que actua com a nucli del sistema. Els dispositius clients poden publicar missatges en unes bústies específiques o subscriure's a la informació que arriba a aquestes bústies per a rebre-la en temps real. Aquest model de publicació/subscripció ofereix diversos avantatges respecte als protocols tradicionals de comunicació punt a punt. D'una banda, elimina la necessitat que els dispositius clients estiguen connectats directament entre si, simplificant la gestió de les comunicacions en sistemes amb múltiples components. D'altra banda, proporciona una escalabilitat significativa, ja que permet afegir nous dispositius o funcionalitats sense afectar la resta del sistema.



**Figura 10:** Esquema simple del funcionament de MQTT (Medium, n.d.-b)

Un dels aspectes més destacats de MQTT és la seua flexibilitat en la gestió de missatges, gràcies a l'ús de temes jeràrquics que permeten organitzar la informació de manera lògica i estructurada. Aquesta característica facilita la implementació d'aplicacions complexes on es gestionen múltiples fluxos de dades de forma simultània. Per exemple, un sistema pot utilitzar temes separats per a diferents tipus de dades, com ara lectures de sensors, alertes de seguretat o estadístiques de rendiment. Aquesta organització millora la claredat i eficiència del sistema global, especialment en entorns distribuïts.

Un altre avantatge clau de MQTT és el seu suport per a diferents nivells de qualitat de servei (*Quality of Service, QoS*), que garanteixen la fiabilitat en la transmissió de missatges segons les necessitats de l'aplicació. Aquestes opcions inclouen des de lliuraments simples (*at most once*) fins a confirmacions robustes que assegurin la recepció dels missatges (*exactly once*). Aquesta capacitat permet adaptar el comportament del sistema a la criticitat de les dades transmeses, assegurant que la informació essencial arribi al seu destí sense errors.

En aquest projecte s'ha utilitzat el broker **Mosquitto**, una implementació lleugera i d'alt rendiment de MQTT que és compatible amb diverses plataformes, incloent Windows. Mosquitto destaca per la seua facilitat d'instal·lació i configuració, així com per la seua compatibilitat amb protocols de seguretat com SSL/TLS que s'encarreguen de xifrar la informació per assegurar la condifencialitat i integritat de les dades. Aquest broker és àmpliament utilitzat en aplicacions IoT per la seua robustesa i flexibilitat, permetent gestionar múltiples connexions simultànies amb un consum mínim de recursos (Eclipse Foundation, 2024).

MQTT és també altament compatible amb una àmplia gamma de llenguatges de programació i plataformes de maquinari, incloent dispositius petits com microcontroladors, així com sistemes més avançats com ordinadors de placa única. Aquesta compatibilitat facilita la integració en sistemes heterogenis on es combinen diferents tecnologies i dispositius. Les característiques lleugeres i escalables de MQTT han portat a la seua adopció massiva en aplicacions IoT modernes, incloent monitorització remota, automatització industrial i sistemes domòtics. La seua simplicitat de configuració i ús, juntament amb la capacitat de suportar un gran volum de dispositius connectats, fan d'aquest protocol una opció destacada per a sistemes distribuïts que requereixen comunicacions eficients i fiables.

En resum, MQTT és un protocol dissenyat per maximitzar l'eficiència en entorns amb recursos limitats. Les seues capacitats de publicació/subscripció, flexibilitat en la gestió de dades, suport per a nivells de qualitat de servei i compatibilitat amb diverses plataformes el converteixen en una eina indispensable per a aplicacions IoT que requereixen comunicacions segures, fiables i escalables.

### 2.2.4 Llibreries addicionals per a la GUI i altres funcionalitats

El desenvolupament d'aquest projecte ha requerit l'ús de diverses llibreries que ofereixen suport per a les funcionalitats clau del sistema. Aquest apartat descriu les llibreries principals utilitzades, destacant les seues característiques i avantatges.

- **Tkinter** és una llibreria estàndard inclosa en Python que permet el desenvolupament d'interfícies gràfiques d'usuari (GUI) de manera senzilla i eficient. Aquesta tecnologia ofereix un conjunt ampli de widgets, com botons, etiquetes, quadres de text i menús, que poden ser utilitzats per crear interfícies visuals intuïtives i accessibles.

Una de les raons principals per triar Tkinter és la seua integració directa amb Python, que elimina la necessitat d'instal·lar components externs, garantint compatibilitat i facilitat d'ús. A més, el seu caràcter multiplataforma permet que les aplicacions desenvolupades amb aquesta llibreria funcionen tant en Windows com en Linux i macOS, assegurant flexibilitat per a diferents entorns d'ús.

Tkinter també destaca per la seua estabilitat i maduresa, sent una de les llibreries més utilitzades per a projectes que requereixen funcionalitats bàsiques però efectives en el desenvolupament d'interfícies gràfiques. Això la converteix en una elecció ideal per a aplicacions que necessiten una implementació ràpida i senzilla sense sacrificar funcionalitat.

- **Paho-MQTT** és una biblioteca desenvolupada per Eclipse que implementa el protocol MQTT (Message Queuing Telemetry Transport). Aquesta tecnologia està dissenyada específicament per a aplicacions IoT (Internet of Things) i altres sistemes que requereixen una comunicació lleugera i eficient.

El protocol MQTT, tal i com s'ha explicat anteriorment, funciona amb un model de publicació-subscripció, que facilita la transmissió de dades en temps real entre múltiples dispositius. Paho-MQTT és altament apreciat per la seua fiabilitat i la seua capacitat d'operar amb un baix consum de recursos, cosa que és especialment important en sistemes amb limitacions de maquinari o en entorns amb xarxes inestables.

La biblioteca Paho-MQTT ofereix una implementació senzilla i fàcil d'integrar amb altres llenguatges de programació i sistemes, la qual cosa la converteix en una solució versàtil per a projectes que necessiten transmetre informació de manera eficient. A més, suporta mecanismes de seguretat com ara l'ús de credencials o xifrat TLS/SSL, garantint la privacitat i la integritat de les dades en entorns crítics.

- Les llibreries estàndard **OS** i **Sys** també tenen un paper fonamental en molts projectes Python. **OS** proporciona eines per interactuar amb el sistema operatiu, incloent-hi la gestió de fitxers i directoris, la configuració de variables d'entorn i la manipulació de processos. Per altra banda, **Sys** és útil per a accedir a informació del sistema, com ara els arguments passats a l'script, així com per gestionar la sortida i entrada estàndard del programa.

Aquest conjunt de llibreries ofereix un marc complet i robust per desenvolupar sistemes que combinen interfícies gràfiques, comunicació en xarxa i interacció amb el sistema operatiu.

## 3 Desenvolupament

Aquest capítol descriu el desenvolupament del sistema, incloent el hardware i software necessaris per al seu funcionament. Es detallen les funcionalitats principals dels components, la seua configuració i interconnexió, així com el disseny de la interfície gràfica d'usuari (GUI), que permet al personal sanitari monitoritzar i gestionar alertes en temps real.

### 3.1 Hardware

#### 3.1.1 Components i funció

El sistema de monitorització es basa en una combinació de microcontroladors amb sensors integrats i una unitat de processament per a la gestió i comunicació de dades. Els elements de hardware inclosos són els següents:

- **Arduino Nano 33 BLE Sense:** Aquest dispositiu actua com a node principal de monitorització. Es disposa en dues ubicacions clau:
  - Col·locat al **llit del pacient**, utilitzant el sensor HTS221 integrat per a monitoritzar la temperatura corporal i detectar humitat.
  - Situat al **terra de l'habitació**, utilitzant l'acceleròmetre de 3 eixos per identificar patrons de moviment que indiquen possibles caigudes.
- **Raspberry Pi 4:** Aquesta unitat s'utilitza com a central de processament i comunicació. Reuneix les dades enviades per l'Arduino i les transmet a l'ordinador principal mitjançant el protocol MQTT. També permet executar l'script de gestió i assegurar una comunicació fluida amb la GUI.
- **Font d'alimentació:** Cada Arduino es connecta a la Raspberry Pi via USB, mentre que aquesta es deu connectar amb un transformador de tensió amb eixida de 5V i que proporcione almenys 3A.
- **Equip per a executar la GUI:** A més dels dispositius descrits anteriorment, el sistema requereix un equip informàtic per a executar la interfície gràfica d'usuari (GUI). Aquest ordinador actua com a punt central de monitorització per al personal sanitari i rep les dades dels sensors mitjançant el protocol MQTT.

Les funcions principals d'aquest equip són:

- Mostrar en temps real els estats de les habitacions monitoritzades, incloent alertes per a situacions de risc com caigudes, anomalies de temperatura o humitat al llit.
- Permetre la configuració del sistema, com ara la gestió de paràmetres personalitzats per habitació.
- Garantir una interfície fàcil d'usar per al personal sanitari, amb colors i notifikacions per a una identificació ràpida de les anomalies.

Aquest ordinador ha de comptar amb els requisits mínims següents:

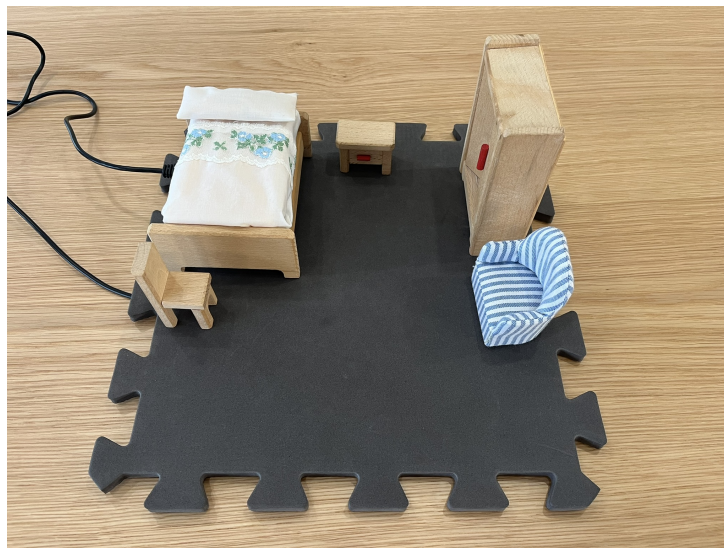
- Sistema operatiu compatible amb Python (Windows, Linux o macOS).
- Capacitat de connexió a xarxes Wi-Fi o Ethernet per comunicar-se amb la Raspberry Pi.
- Resolució mínima de pantalla de 1280x720 per mostrar correctament la GUI.

### 3.1.2 La maqueta física

Per reforçar la comprensió de la disposició i interconnexió dels elements del sistema, s'ha creat una maqueta física que simula una habitació típica d'un centre geriàtric. Aquesta maqueta inclou elements com el llit del pacient, un armari, una tauleta de nit, un silló i una cadira, amb l'objectiu de reproduir fidelment l'entorn d'ús.

Els components del sistema s'han col·locat de manera que es representen les funcions següents:

- El **llit del pacient** actua com a punt de monitorització de temperatura corporal i humitat. L'Arduino Nano 33 BLE Sense situat al llit està connectat de manera discreta, evitant interferències visuals o molestes per al pacient.
- El **terra de l'habitació** és la zona de detecció de moviments bruscos o caigudes. L'Arduino per a caigudes es troba fixat en un punt estratègic per garantir la cobertura de tota l'àrea pròxima al llit.
- La **Raspberry Pi 4**, encarregada de la gestió de dades i comunicacions, està situada prop del mobiliari de l'habitació per facilitar l'accés al sistema i mantenir una connexió estable amb els microcontroladors.

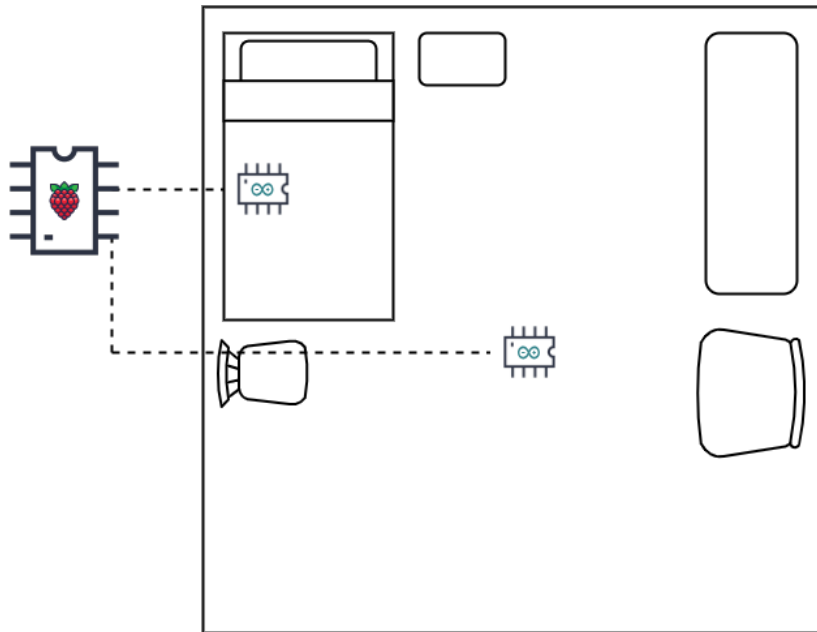


*Figura 11: Maqueta física creada per al projecte, que simula una habitació de centre geriàtric.*

La maqueta no només permet visualitzar l'entorn on operarà el sistema, sinó que també facilita la realització de proves i demostracions en condicions controlades. S'han utilitzat materials lleugers per a la construcció, assegurant que la maqueta siga fàcilment transportable i adaptable per a futures modificacions o demostracions.

### 3.1.3 Esquema de la disposició física

Per complementar la maqueta, es presenta un esquema gràfic de la distribució dels components dins de l'habitació simulada. Aquest esquema permet observar com s'han col·locat els dispositius respecte als punts clau de monitorització.



*Figura 12: Esquema de la distribució dels components dins de la maqueta física.*



## 3.2 Software

### 3.2.1 Arduino per a la monitorització de la temperatura i la humitat

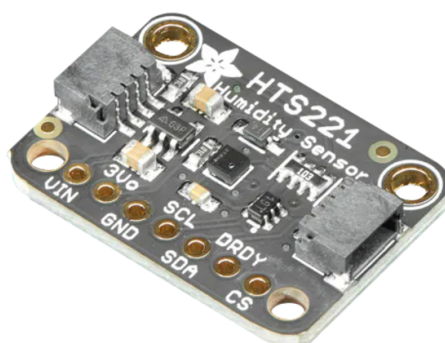
Per a la monitorització de la temperatura i la humitat del llit del pacient, s'ha utilitzat l'**Arduino Nano 33 BLE Sense**, que integra el sensor **HTS221**. Aquest sensor és capaç de mesurar la temperatura i la humitat relativa amb un alt nivell de precisió, fent-lo ideal per detectar anomalies que podrien comprometre la salut del pacient.

En primer lloc, es va preparar l'entorn de desenvolupament configurant l'Arduino Nano 33 BLE Sense amb l'ordinador mitjançant un cable USB, que permet tant la programació com la transmissió de dades en temps real. Per accedir a les funcionalitats del sensor HTS221, va ser necessari descarregar la llibreria oficial `Arduino_HTS221` a través del gestor de llibreries de l'IDE d'Arduino. Aquesta llibreria proporciona les funcions necessàries per inicialitzar el sensor i obtenir lectures fiables de temperatura i humitat.

Amb l'entorn configurat, es va desenvolupar un codi que permet al sistema realitzar lectures contínues cada cinc segons. El codi inclou la inicialització del sensor amb la funció `HTS.begin()`, que comprova si el dispositiu s'ha activat correctament. En cas d'error, el sistema genera un missatge per sèrie i es deté, assegurant així que no es transmeten dades incorrectes. Una vegada inicialitzat, el sensor llegeix la temperatura i la humitat utilitzant les funcions `HTS.readTemperature()` i `HTS.readHumidity()`, respectivament. Les dades obtingudes es transmeten mitjançant una comunicació sèrie virtualitzada a través del port USB en un format que posteriorment pot ser processat fàcilment per la Raspberry Pi.

El funcionament del codi es basa en lectures periòdiques, que es realitzen amb un retard configurat a cinc segons. Aquest interval permet obtenir dades suficientment detallades per al monitoratge sense sobrecarregar el sistema amb lectures excessives. El format de transmissió utilitza etiquetes descriptives per diferenciar clarament els valors de temperatura i humitat, cosa que facilita la interpretació de les dades pel sistema central. Aquest sistema és capaç de proporcionar informació en temps real sobre les condicions del llit del pacient, incloent episodis d'incontinència o anomalies de temperatura corporal.

Per a més detalls, el codi complet utilitzat per aquesta implementació es troba a l'annex a la [secció 5.2](#), on es poden observar totes les funcions i configuracions utilitzades.



*Figura 13: Sensor HTS221, el qual es troba integrat a la placa Arduino.*

### 3.2.2 Arduino per a la detecció de caigudes amb Edge Impulse

#### Connexió de l'Arduino Nano 33 BLE Sense amb Edge Impulse

Per al desenvolupament del sistema de detecció d'anomalies i classificació de moviments, s'ha utilitzat la plataforma web **Edge Impulse**, que ofereix eines específiques per a integrar models de *Tiny Machine Learning (TinyML)* en dispositius de baix consum com l'**Arduino Nano 33 BLE Sense**.

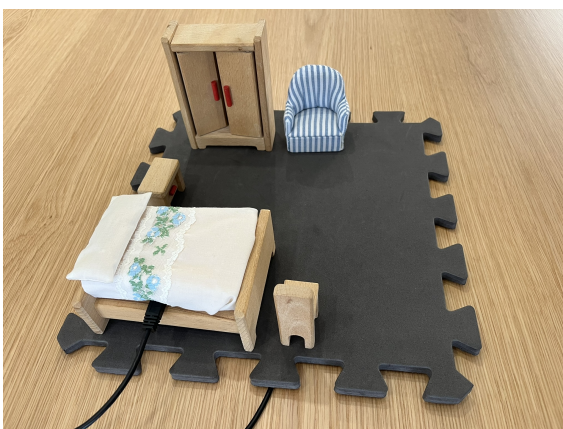
El procés ha començat amb la instal·lació del *firmware* d'Edge Impulse, disponible oficialment per a l'Arduino Nano 33 BLE Sense. Aquest *firmware* proporciona compatibilitat directa entre el dispositiu i la plataforma, permetent la captura i transmissió de dades en temps real des dels sensors integrats, com ara l'acceleròmetre. La configuració inicial s'ha dut a terme seguint els passos següents: primerament, s'ha connectat l'Arduino a l'ordinador mitjançant un cable USB per carregar-hi el *firmware* oficial. A continuació, mitjançant el terminal de la plataforma Edge Impulse, s'ha configurat el dispositiu utilitzant `edge-impulse-daemon` en un terminal CMD. Aquesta utilitat reconeix automàticament el dispositiu connectat i estableix una connexió segura amb el projecte creat prèviament a Edge Impulse.

#### Procés de Recopilació de Dades (Data Acquisition)

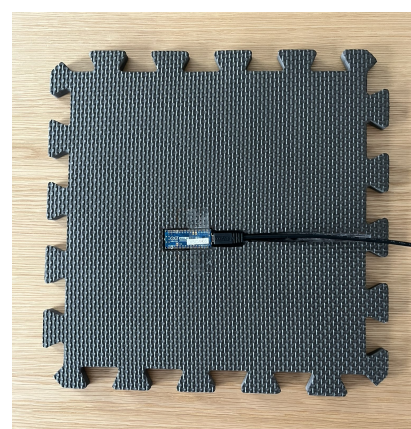
La recopilació de dades és una de les fases més importants en el desenvolupament d'un model de *Tiny Machine Learning*, ja que la qualitat i la diversitat de les dades recollides influeixen directament en l'eficàcia del model entrenat. Per a aquest projecte, s'han utilitzat les dades capturades per l'acceleròmetre de tres eixos (*AccX*, *AccY*, *AccZ*) integrat en l'Arduino Nano 33 BLE Sense, mitjançant la connexió prèviament establerta amb **Edge Impulse**.

El procés de recopilació s'ha dissenyat per obtenir un conjunt de dades representatiu que inclou dues classes principals: "passos" i "res". Aquestes classes són fonamentals per identificar patrons associats al moviment o la manca de moviment dins de l'habitació del pacient.

Totes les mostres s'han obtingut utilitzant la maqueta a escala dissenyada per a aquest projecte (figura 14 i figura 15). En aquesta maqueta, es va simular la presència d'un pacient fent ús d'un objecte que representava el cos d'un ancià, amb dimensions i pes adequats. Aquest enfocament permet recrear condicions de moviment i quietud de manera controlada, assegurant que els patrons registrats per l'acceleròmetre són comparables als que es produïrien en un escenari real.



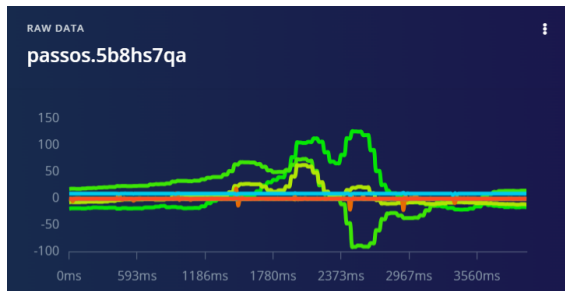
**Figura 14:** Maqueta on s'han realitzta les mostres



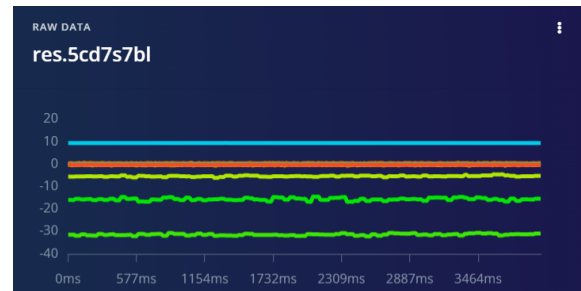
**Figura 15:** Ubicació de l'Arduino a la part inferior de la maqueta

Per a la classe "passos", s'han recollit un total de 50 mostres, cadascuna amb una durada de 4 segons. Aquesta durada ha estat seleccionada per garantir que es capturen patrons de moviment complets que representen el moviment dels passos en diverses direccions. Les mostres es van recollir amb una taxa de mostreig de 100 Hz, la qual cosa assegura una resolució temporal suficient per detectar canvis subtils en els patrons de moviment.

En el cas de la classe "res", que representa l'absència de moviment significatiu, s'han capturat 32 mostres, també amb una durada de 4 segons cadascuna. Aquest conjunt de dades ha estat recollit en condicions estàtiques per assegurar que els valors mesurats reflecteixen amb precisió situacions d'immobilitat.

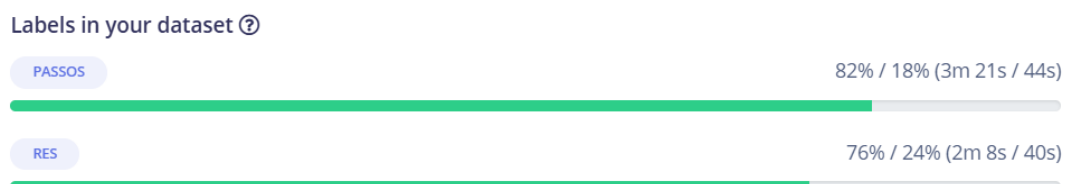


*Figura 16: Mostra de la classe "passos"*



*Figura 17: Mostra de la classe "res"*

Una vegada recollides les dades, s'ha aplicat una divisió en **train/test split** per preparar el conjunt de dades per a l'entrenament i la validació del model. Per a la classe "passos", el conjunt de dades s'ha dividit en un 82% per a l'entrenament i un 18% per al test. Per a la classe "res", la divisió ha estat del 76% per a l'entrenament i del 24% per al test. Aquesta distribució garanteix que el model té prou dades per aprendre patrons mentre es reserva un percentatge adequat per avaluar el seu rendiment en dades no vistes durant l'entrenament.



*Figura 18: Percentatges del train/test split*

Durant la recopilació, es va fer ús de la interfície de Edge Impulse, que permet etiquetar cada mostra amb la classe corresponent, assegurant que les dades s'organitzen correctament dins del conjunt de dades del projecte. A més, l'ús d'aquesta eina ha facilitat la visualització de les dades en temps real, cosa que ha permès verificar que les lectures capturades reflecteixen les condicions desitjades.

El procés de recopilació ha estat dissenyat per cobrir una varietat de condicions. Això inclou moviments de passos en diverses direccions i velocitats i també períodes de calma sense moviments. Aquesta diversitat en les dades augmenta la robustesa del model, ja que li permet identificar correctament els patrons de moviment i distingir-los de situacions de quietud.

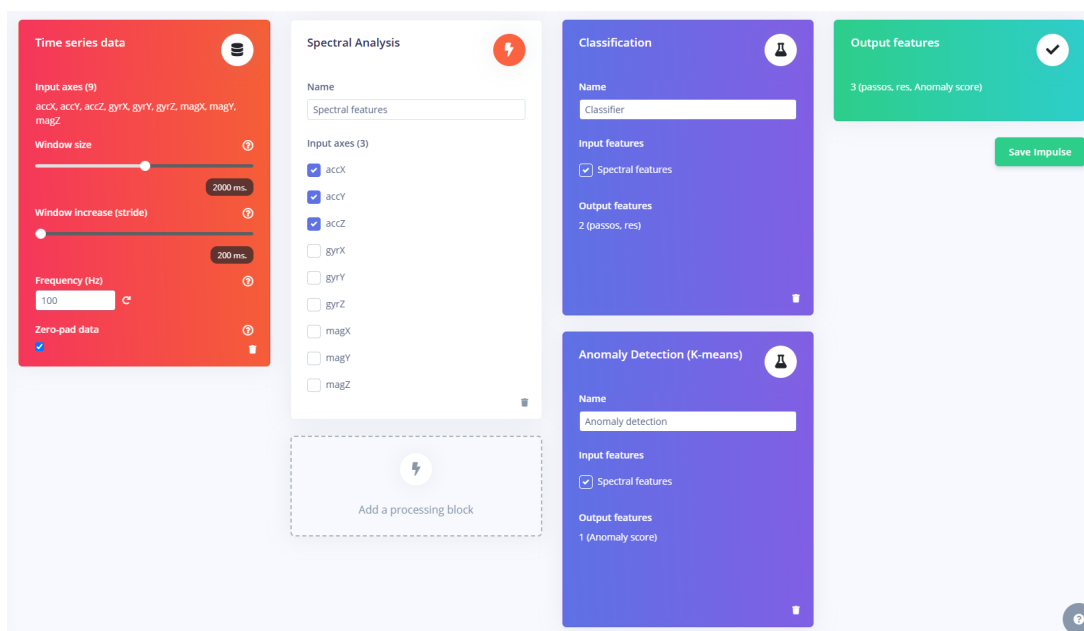
## Creació de l'Impulse

La creació de l'*Impulse* és un pas clau en el desenvolupament d'un model *TinyML*, ja que defineix com es processen i classifiquen les dades capturades pels sensors. En aquest projecte, s'ha configurat un *Impulse* específicament adaptat a les necessitats del sistema de detecció de passos i moviment. El procés de creació de l'*Impulse* a la plataforma **Edge Impulse** inclou diverses fases: la configuració de la finestra temporal (*Window Size*) per analitzar les dades, la selecció de les característiques rellevants (*features*), l'aplicació d'anàlisis espectrals i la configuració del model de classificació. A continuació es descriuen els detalls de cada fase.

Per analitzar les dades capturades per l'acceleròmetre, s'ha configurat una finestra temporal (*window size*) de 2000 ms, amb un increment de finestra (*stride frequency*) de 100 Hz. Aquesta configuració permet dividir les dades de moviment en segments uniformes, garantint que cada finestra conté informació suficient per identificar patrons rellevants. Les característiques seleccionades per al model són les tres dimensions de l'acceleròmetre: *AccX*, *AccY* i *AccZ*. Aquestes característiques representen les variacions del moviment en els tres eixos espacials de l'acceleròmetre, proporcionant informació completa sobre el comportament del sistema.

Per extreure característiques significatives dels senyals de moviment, s'ha aplicat una anàlisi de Fourier ràpida (*FFT*) amb una longitud de 16 mostres. Aquest procés converteix els senyals temporals en el domini de freqüència, permetent al model identificar components freqüencials clau. No s'ha aplicat cap filtre addicional, ja que les dades recollides ja eren prou netes per al seu ús directe.

L'*Impulse* s'ha dissenyat amb dos components principals: un *classifier* i una detecció d'anomalies. El *classifier* genera dues sortides: "*passos*" i "*res*". Paral·lelament, s'ha implementat una detecció d'anomalies per a la detecció de caigudes basada en el mètode de clustering anomenat *k-means*, que afegeix una tercera sortida "*anomaly*" per identificar patrons inesperats en els senyals. Aquesta detecció d'anomalies es la que permet la detecció de caigudes, ja que crear una classe específica per a les caigudes era molt difícil, degut a la gran varietat de senyals que poden provocar. S'ha de tindre en compte que una caiguda pot ser de moltíssimes maneres i entrenar amb caigudes específiques podria donar lloc a errors en la detecció.



**Figura 19:** Pantalla de creació de l'Impulse a Edge Impulse

## Entrenament del Model

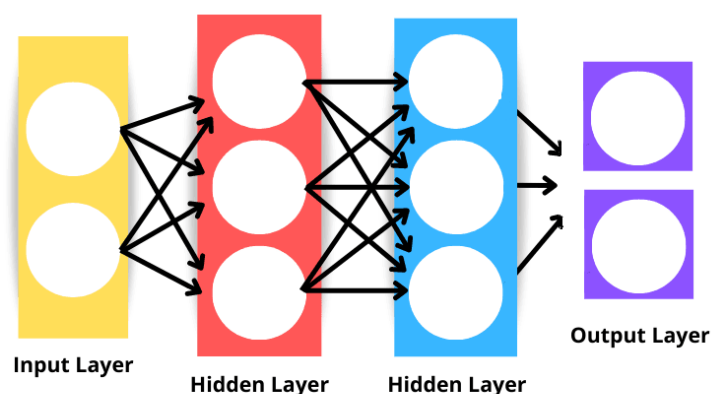
De l'entrenament del model depèn la capacitat del model per detectar correctament els patrons de moviment i identificar anomalies. Per assegurar un entrenament òptim, s'han definit uns hiperparàmetres i una arquitectura de xarxa neuronal adequats per a les dades i els requeriments computacionals del projecte.

El model s'ha entrenat durant **50 cicles d'entrenament**, amb una taxa d'aprenentatge (*learning rate*) de **0.0005**. Aquesta configuració s'ha seleccionat empíricament per garantir un equilibri entre una convergència ràpida i una pèrdua mínima. Els 50 cicles han estat suficients per aconseguir una precisió elevada, mentre que una taxa d'aprenentatge moderada ha evitat problemes de sobreajustament (*overfitting*) o subajustament (*underfitting*).

La xarxa neuronal implementada té una estructura compacta, però suficientment poderosa per manejar les característiques extremes. L'arquitectura inclou:

- **Capa d'entrada:** Rep un total de **39 característiques**, extretes a partir de les dades de moviment processades. Aquestes característiques s'obtenen mitjançant una anàlisi espectral basada en FFT (Fast Fourier Transform) aplicada sobre els senyals d'acceleració capturats en els tres eixos espacials (*AccX*, *AccY* i *AccZ*). Amb una longitud de finestra configurada a 2000 ms i una FFT de 16 punts, es divideixen les dades en segments temporals uniformes. Per a cada eix, es generen 13 característiques espectrals (Edge Impulse, 2023). En total, la combinació dels tres eixos genera les 39 característiques (13 per eix  $\times$  3 eixos).
- **Capes intermèdies:** Dues capes consecutives amb **20 neurones cadascuna**, que utilitzen la funció d'activació *ReLU*. Aquesta funció d'activació, abreviatura de *Rectified Linear Unit*, transforma els valors negatius en zero i manté els positius, la qual cosa permet modelar relacions no lineals i accelera la convergència de la xarxa neuronal (Wikipedia, 2023b).
- **Capa de sortida:** Una capa amb **2 classes** ("*passos*" i "*res*"), amb funció d'activació *softmax* per proporcionar probabilitats normalitzades.

Aquesta arquitectura, senzilla però robusta, s'ha seleccionat per la seua capacitat d'adaptar-se als requeriments del projecte, optimitzant tant el rendiment com l'eficiència computacional.



*Figura 20: Esquema orientatiu de l'estructura de la xarxa neuronal*

Els resultats de l'entrenament han estat altament satisfactoris, assolint una **precisió del 98.9%** en les dades d'entrenament, amb una pèrdua (*loss*) mínima de **0.04**. El paràmetre *loss* és una mètrica que permet mesurar quantitativament com de bona és la xarxa neuronal en executar el model, indicant el grau d'ajustament de les prediccions respecte a les dades d'entrenament (Wikipedia, 2023a). Una pèrdua tan baixa suggereix que el model és capaç de classificar amb gran precisió els patrons de moviment i la seua absència, mantenint una robustesa adequada per evitar errors en situacions reals.

**Taula 1:** Resultats de l'Avaluació del Model (*Int8*)

Classe	Precisió	Record	F1-Score	Suport
Passos	1.000	0.981	0.990	103
Res	0.975	1.000	0.987	78
Mitjana Macro	0.988	0.990	0.989	181
Mitjana Ponderada	0.989	0.989	0.989	181

**Taula 2:** Resultats de l'Avaluació del Model (*Float32*)

Classe	Precisió	Record	F1-Score	Suport
Passos	1.000	0.981	0.990	103
Res	0.975	1.000	0.987	78
Mitjana Macro	0.988	0.990	0.989	181
Mitjana Ponderada	0.989	0.989	0.989	181

**Taula 3:** Resultats del Test del Model (*Float32*)

Classe	Precisió	Record	F1-Score	Suport
Passos	1.000	0.983	0.992	121
Res	0.982	1.000	0.991	110
Mitjana Macro	0.991	0.992	0.991	231
Mitjana Ponderada	0.991	0.991	0.991	231

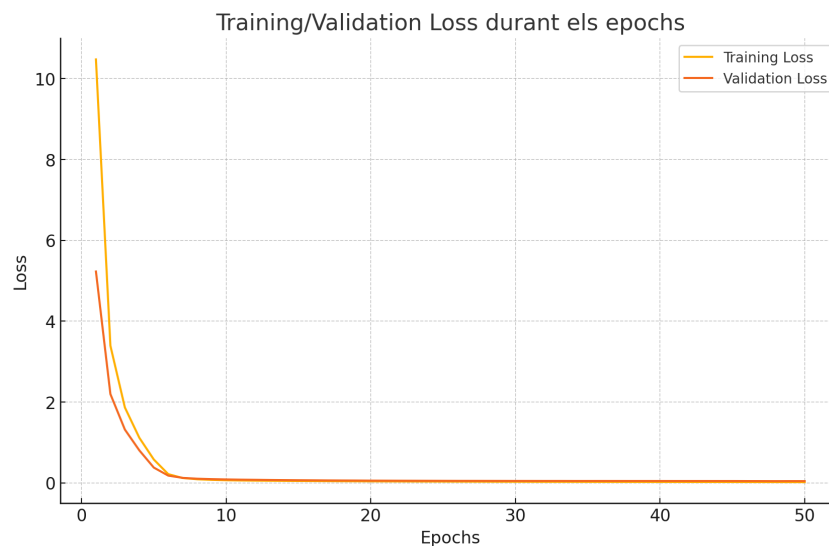
Les dades presentades en les taules 1, 2 i 3 mostren molt bons resultats pel que fa al rendiment del model entrenat. La precisió i el record són consistentment alts en totes les configuracions, superant el 98.5% en les classes "passos" i "res". Aquestes mètriques indiquen que el model té una gran capacitat per distingir amb precisió els patrons de moviment dels casos de quietud, fins i tot en situacions de test no vistes durant l'entrenament.

A més, la mitjana macro i ponderada es mantenen uniformes, la qual cosa reflecteix que el model funciona de manera equilibrada entre les diferents classes. Les lleugeres diferències entre les configuracions *int8* i *float32* evidencien que la quantització no ha tingut un impacte negatiu significatiu sobre el rendiment del model, assegurant la viabilitat del seu ús en microcontroladors com la placa que s'utilitza en aquest projecte.



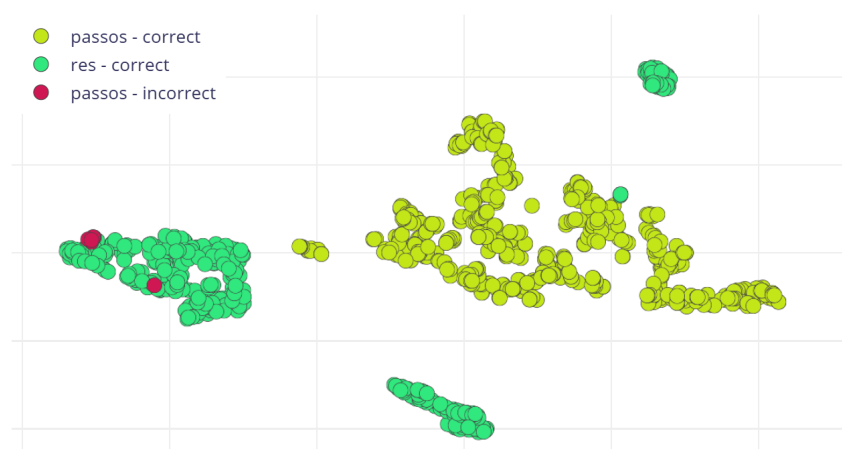
Durant el procés d'entrenament del model, es van obtenir resultats destacables que evidencien la capacitat del sistema per classificar correctament les dades i identificar patrons específics.

La [figura 21](#) mostra l'evolució de la pèrdua (*loss*) tant en les dades d'entrenament com en les de validació durant els **50 cicles d'entrenament**. Es pot observar que el model ha après adequadament les característiques necessàries. La reducció significativa de la pèrdua en les primeres iteracions evidencia un procés d'aprenentatge ràpid, mentre que les fases posteriors mostren una estabilització, confirmant que el model ha arribat a una òptima generalització sense caure en (*overfitting*).



**Figura 21:** Gràfica de l'evolució del Loss a l'entrenament i la validació durant l'entrenament.

Per tal d'avaluar el rendiment del model sobre el conjunt d'entrenament, la [figura 22](#) il·lustra com el model neural classifica totes les dades del *training set*. Els punts representats en verd corresponen a dades classificades correctament, mentre que els punts en roig són dades classificades incorrectament. Com es pot observar, la gran majoria de dades estan classificades de manera encertada, la qual cosa reflecteix que el model és capaç d'identificar patrons clars per a les dues categories definides: "passos" i "res".

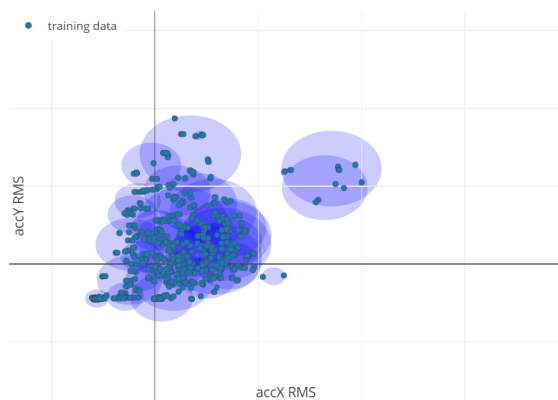


**Figura 22:** Exploració de les dades del Training Set.

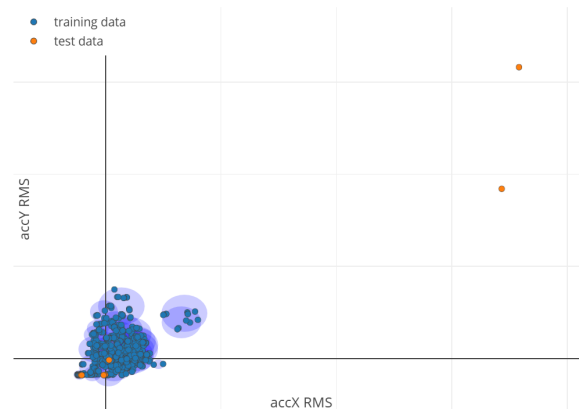
## Detecció d'Anomalies

La detecció d'anomalies és un bloc essencial del sistema, ja que permet identificar esdeveniments inesperats o irregulars en els senyals capturats per l'acceleròmetre. En aquest projecte, aquesta funcionalitat ha estat dissenyada amb l'objectiu de detectar situacions no classificables en les categories definides ("passos" i "res"), és a dir, les caigudes. Per implementar aquesta detecció, s'ha utilitzat l'algorisme de **k-means**, que agrupa les dades en funció de la seva similitud en un espai multidimensional. En concret, s'han generat *clusters* basats en les característiques RMS (*Root Mean Square*) dels senyals capturats pels 3 eixos de l'acceleròmetre. Aquestes característiques permeten analitzar la variabilitat dels senyals i identificar patrons que s'allunyen dels grups establerts durant l'entrenament.

La [figura 23](#) mostra la distribució de les dades d'entrenament en funció de les característiques seleccionades. Els punts representen mostres etiquetades durant la fase d'entrenament, mentre que les zones envoltades indiquen els límits dels *clusters* definits. Els punts que es troben fora d'aquestes zones es consideren anomalies i es classifiquen com a tals pel model.



**Figura 23:** Distribució de les dades d'entrenament



**Figura 24:** Exemple d'una mostra de caiguda

Com es pot observar en la [figura 24](#), la majoria de les mostres es concentren al voltant de dos grups principals, corresponents a les categories "passos" i "res". No obstant això, les mostres de caigudes es troben en zones que no pertanyen a cap *cluster*, marcant-se com anomalies. Aquesta capacitat és essencial per identificar les caigudes o altres situacions atípiques que no es poden preveure durant l'entrenament.

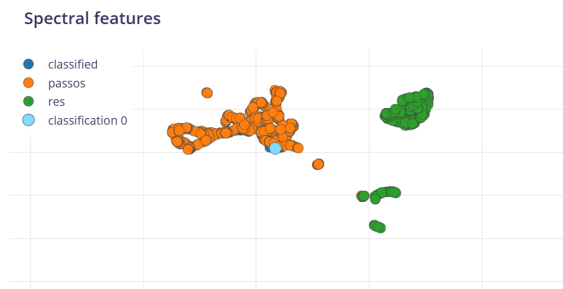


## Proves en Viu (Live Classification)

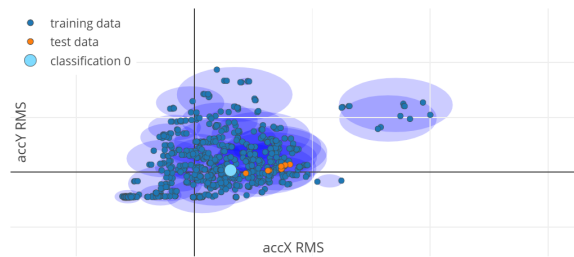
Abans de procedir al desplegament final del model en el maquinari de l'Arduino Nano 33 BLE Sense, es van realitzar proves en viu utilitzant la funcionalitat de classificació en temps real (*Live Classification*) proporcionada per la plataforma **Edge Impulse**. Aquestes proves són essencials per validar el comportament del model amb dades noves que no han estat utilitzades durant l'entrenament ni la validació.

Durant les proves, es van simular tres situacions representatives: "res", "passos" i "caigudes". Cada prova es va realitzar en condicions controlades utilitzant la maqueta a escala desenvolupada per al projecte. En les simulacions de "res", es va assegurar que no hi havia cap moviment significatiu en l'àrea monitoritzada, mentre que en les simulacions de "passos" es van reproduir patrons de moviment constants i repetitius amb l'objecte que representa el pacient. Per simular les caigudes, es van generar moviments bruscos amb canvis sobtats en la direcció i la intensitat de les oscil·lacions, reproduint el que es podria esperar en un escenari real.

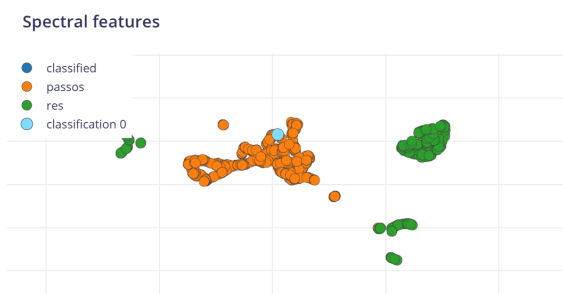
Per il·lustrar alguns resultats obtinguts durant aquestes proves en viu per a comprovar el correcte funcionament del sistema, es presenten dos exemples representatius de classificacions realitzades. Les figures 25 i 26 mostren el comportament del model durant una simulació de "passos", on es poden observar els patrons consistents de moviment identificats amb alta confiança per la classificació i on no es mostra cap signe d'anomalia. Les figures 27 i 28 corresponen a una simulació de "caiguda", on el sistema, encara que la predicció s'acosta als passos, detecta valors molt elevats d'anomalia, activant correctament la categoria "anomaly".



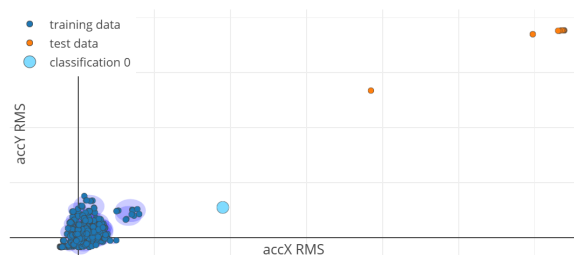
**Figura 25:** Classificació de la mostra de passos



**Figura 26:** Classificació d'anomalia de la mostra de passos



**Figura 27:** Classificació de la mostra de caiguda

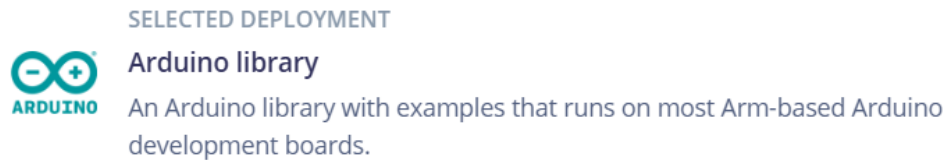


**Figura 28:** Classificació d'anomalia de la mostra de caiguda

Els resultats obtinguts en les proves en viu van confirmar la viabilitat del model per al seu ús pràctic i van establir una base sòlida per procedir amb el seu desplegament.

## Exportació i Desplegament del Model a l'Arduino

Un cop validat el model mitjançant les proves en viu, es va procedir a l'exportació del model per integrar-lo al maquinari del projecte. La plataforma **Edge Impulse** facilita aquest procés gràcies a la possibilitat d'exportar el model en forma de llibreria per a Arduino optimitzada per a dispositius de baix consum, com és el cas de l'**Arduino Nano 33 BLE Sense**, utilitzant TensorFlow Lite. Aquest format d'exportació inclou el model entrenat, les configuracions de preprocessament i els codis necessaris per a la seva execució en temps real.



**Figura 29:** Desplegament en forma de llibreria d'Arduino seleccionat en Edge Impulse

La llibreria generada per Edge Impulse es va descarregar en un arxiu comprimit (*ZIP*), que es va importar directament a l'entorn de desenvolupament Arduino IDE. La integració d'aquesta llibreria dins del projecte va permetre configurar el sistema perquè el model entrenat poguera utilitzar les dades capturades per l'acceleròmetre com a entrada per a la inferència. A més, es van ajustar les configuracions del projecte a l'IDE, assegurant que les configuracions del port sèrie i el tipus de placa seleccionada coincidien amb les de l'Arduino Nano 33 BLE Sense.

Un dels aspectes més destacats d'aquesta llibreria és que està optimitzada específicament per a executar models de *TinyML*, el que garanteix un ús eficient dels recursos limitats del microcontrolador. Això permet realitzar inferències amb baixa latència, mantenint el consum energètic dins dels nivells requerits pel projecte. A més, el codi generat inclou una implementació completa del model i del preprocessament necessari, podent així realitzar proves per a comprovar el seu correcte funcionament.

Després de carregar el codi final al microcontrolador, es van dur a terme les primeres proves amb els codis d'exemple per assegurar que el sistema era capaç de processar les dades capturades en temps real. Aquestes proves van confirmar que el model podia realitzar classificacions correctes per a les classes definides ("*passos*" i "*res*") i també detectar anomalies amb alta precisió per a detectar les "*caigudes*", complint així amb els objectius inicials del projecte.

No obstant això, encara que els resultats eren correctes, es mostrava un missatge d'error molest que dificultava enviar les dades per port serie a la raspberry ("*Error INFO:HW RFFT failed, falling back to SW*"). Aquest error va resultar ser solament una informació del log que passa sovint als usuaris d'Edge Impulse (Edge Impulse Forum, 2024), que es va poder eliminar dels fitxers de la llibreria per a generar una sortida més neta.

## Implementació del Model en l'Arduino

Després de l'exportació del model optimitzat des d'**Edge Impulse**, el pas següent va ser integrar-lo al maquinari de l'Arduino Nano 33 BLE Sense. Per a aquesta tasca, es va partir del codi d'exemple generat automàticament per la plataforma d'Edge Impulse, el qual proporciona una base funcional per executar el model d'inferència en el dispositiu. Aquest codi inclou totes les funcionalitats necessàries per processar les dades del sensor i executar la classificació del model exportat.

No obstant això, per tal d'adaptar el codi a les necessitats específiques del projecte, es van realitzar algunes modificacions clau. Una de les més importants va ser la introducció d'un llindar d'anomalia per a la detecció de caigudes. Aquest llindar es va establir després de múltiples proves, que van determinar que un valor de 5.0 era adequat per identificar caigudes de manera fiable sense generar falsos positius.

El canvi es va implementar dins de la funció encarregada d'analitzar i mostrar les prediccions del model i es van comentar les parts que no eren necessàries, ja que l'únic realment important en la sortida és si el sistema ha detectat una caiguda. A l'annex es troba el codi corresponent a aquesta funcionalitat ([codi 2](#)).

Aquest codi afegeix una condició per verificar si l'*anomaly score* (puntuació d'anomalia) és superior a 5.0. En cas afirmatiu, el sistema imprimeix el missatge "**caiguda**" al port sèrie, permetent així la detecció en temps real d'una situació potencialment crítica.

### 3.2.3 MQTT i Raspberry Pi

El protocol **MQTT** és l'encarregat de garantir la comunicació eficient entre els diferents components del sistema, connectant els sensors Arduino Nano 33 BLE Sense amb la interfície gràfica d'usuari (GUI) a través d'un broker. Tot i que inicialment es va considerar allotjar el broker en la Raspberry Pi, finalment es va optar per instal·lar-lo en un ordinador amb sistema operatiu Windows, des d'on també s'executa la GUI. Aquesta configuració centralitzada simplifica la gestió del sistema i optimitza el seu funcionament.

El broker seleccionat per al projecte ha estat **Mosquitto**, una solució lleugera i altament fiable àmpliament utilitzada en aplicacions IoT. La instal·lació es va realitzar seguint els passos recomanats pel fabricant, descarregant el programari des de la pàgina oficial de Mosquitto (<https://mosquitto.org>). Després de la instal·lació, es va procedir a configurar el fitxer `mosquitto.conf`, activant l'escolta al port 1883 per a comunicacions TCP sense xifrat, ja que el projecte es troba en una fase de desenvolupament i proves. Aquesta configuració simplificada permet una comunicació fluida dins de la xarxa local. El servei es va iniciar des de la línia de comandes amb l'ordre `mosquitto -v`, que activa el mode de depuració, facilitant la supervisió de les connexions i missatges publicats.

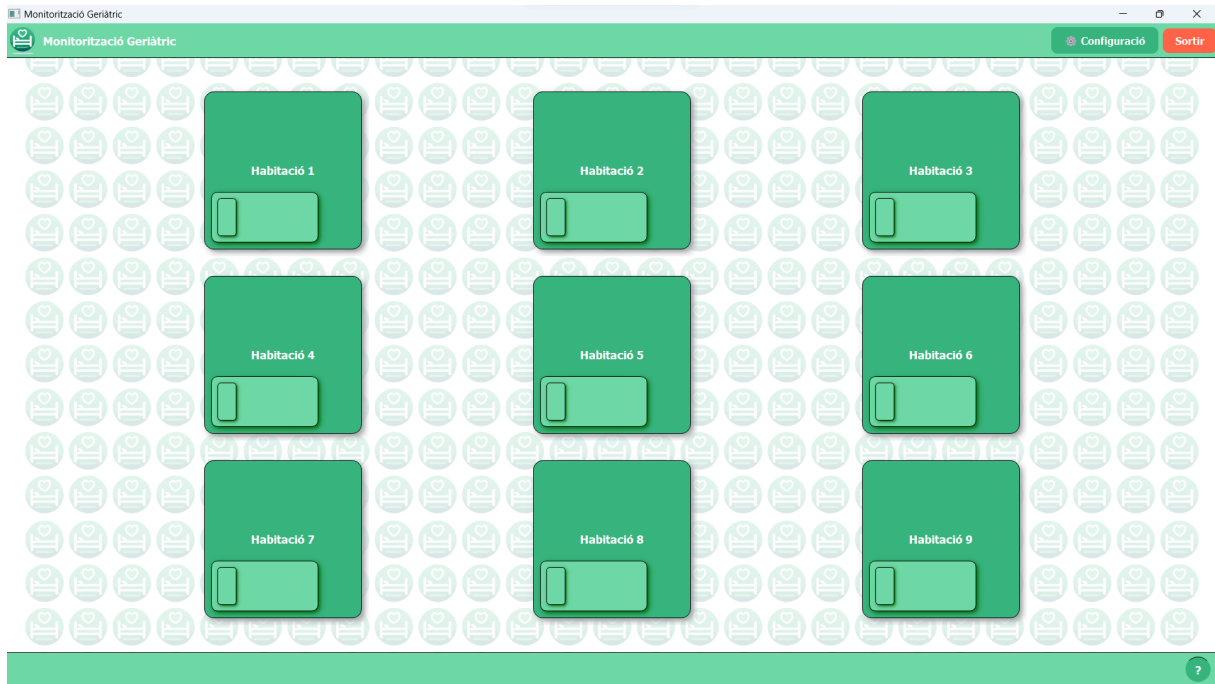
D'altra banda, la **Raspberry Pi 4** té un paper fonamental com a node intermediari en el sistema. Aquest dispositiu està configurat per gestionar la recepció de dades provinents dels sensors Arduino mitjançant connexions serials USB i publicar-les al broker MQTT instal·lat en el Windows. Aquesta arquitectura permet que la Raspberry Pi funcioni com un pont que assegura la integritat i l'accessibilitat de les dades enviades pels sensors.

El programari desenvolupat per a la Raspberry Pi està escrit en Python, utilitzant les biblioteques `paho-mqtt` per gestionar la comunicació MQTT i `pyserial` per interactuar amb els sensors connectats als ports sèrie, tal i com es pot veure al [codi 3](#). Aquest script està configurat per manejar dos canals principals: un per a la lectura de temperatura i humitat i un altre per a la detecció de caigudes. Les dades de temperatura i humitat es publiquen als temes `geriatric/habitacio1/temperatura` i `geriatric/habitacio1/humitat`, respectivament, mentre que, en cas de detectar una caiguda, es publica una alerta al tema `geriatric/habitacio1/alerta`. La utilització del tema `habitacio1` respon al fet que les proves s'estan realitzant en una maqueta a escala que simula l'entorn d'una habitació única d'un centre geriàtric. Per evitar duplicats en les notificacions de caigudes consecutives, el codi inclou un mecanisme basat en intervals de temps mínims entre alertes, establint un lllindar de 10 segons. Aquesta funcionalitat assegura que cada tipus de dades s'envie al broker MQTT de manera adequada, garantint la integritat de la informació en temps real. El fragment del codi que gestiona la lectura de dades i l'enviament de missatges per wi-fi via MQTT es pot consultar també a l'annex ([codi 4](#)). Cal remarcar que s'ha editat el fitxer de configuració de tasques de cron de la Raspberry Pi per a que aquest programa s'execute automàticament quan s'encén la placa.

Gràcies a aquesta configuració, les dades generades pels sensors són accessibles en temps real per la GUI. Aquesta comunicació permet al sistema reaccionar immediatament davant de situacions crítiques com anomalies en la temperatura, la humitat o possibles caigudes. La configuració descrita no només assegura una integració fluida entre els components del sistema, sinó que també facilita una futura escalabilitat, ja que permet afegir nous sensors o funcionalitats simplement publicant dades en nous temes MQTT.

### 3.2.4 Interfície Gràfica (GUI)

La interfície gràfica desenvolupada per a aquest projecte és un element central que integra totes les funcionalitats del sistema, oferint una experiència visual, interactiva i en temps real per al personal sanitari. La GUI, construïda amb la biblioteca `PyQt5`, permet supervisar múltiples habitacions simultàniament i generar alertes visuals i sonores per a situacions crítiques.



*Figura 30: Pantalla principal del programa, on es mostren les habitacions del geriàtric*

Per tal de garantir un desenvolupament organitzat i escalable, la GUI està estructurada en un directori jeràrquic, amb carpetes i arxius que separen clarament les diferents funcionalitats. Aquesta divisió permet mantenir un codi més modular, facilitant futures modificacions o ampliacions del sistema. A continuació, es descriu breument aquesta estructura i la funció principal de cada component:

- **Directori assets/:** Conté els recursos visuals i multimèdia necessaris per a la GUI. Inclou fitxers com:
  - `icono.png`: La icona principal de l'aplicació, mostrada en diverses pantalles.
  - `fondo.png`: Una imatge de fons que aporta coherència visual al disseny.
  - `alarma.wav`: El so d'alerta utilitzat per notificar el personal sanitari de possibles caigudes.
- **Directori configuracio/:** Agrupa arxius relacionats amb la configuració del sistema.
  - `constants.py`: Defineix valors constants utilitzats en tota l'aplicació.
  - `gestio_config.py`: Conté funcions per carregar i guardar configuracions.

- **Directori components/:** Inclou els components principals que defineixen les pantalles i funcionalitats de la GUI.
  - `config.habitacio.py`: Permet personalitzar paràmetres específics per a cada habitació, com el nom o els valors límit de temperatura i humitat.
  - `habitacio.py`: Defineix la classe que representa les habitacions, incloent-hi els seus estats visuals (verd, roig) i la interacció amb les dades rebudes.
  - `pantalla_inici.py`: Implementa la pantalla inicial de benvinguda, on es mostra el logotip del projecte.
  - `configuracio.py`: Gestiona la finestra de configuració global, permetent ajustar valors per a totes les habitacions.
  - `aplicacio_principal.py`: Defineix la interfície principal on es supervisen les habitacions en temps real.
- **Arxius a nivell arrel:**
  - `main.py`: És el punt d'entrada de l'aplicació, encarregat d'inicialitzar tots els components i establir la connexió amb el sistema.
  - `config.json`: Emmagatzema la configuració guardada de manera que es mantinga entre sessions encara que el programa es tanque.
  - `README.md`: Proporciona informació sobre com executar el projecte, útil per a desenvolupadors.
  - `requirements.txt`: Inclou una llista de dependències que s'han d'instal·lar per executar l'aplicació.

Aquesta organització garanteix un desenvolupament eficient i permet mantenir un flux de treball clar. Per exemple, l'ús del fitxer `config.json` assegura que la configuració de la GUI persista de sessió a sessió, oferint una experiència consistent per als usuaris. Aquesta estructura també facilita la separació de responsabilitats entre els diferents components del projecte, millorant la mantenibilitat del codi i assegurant una base sòlida per a futures ampliacions.

Quant a la funcionalitat, un element essencial del sistema és la integració amb el protocol *MQTT* per gestionar les comunicacions entre els sensors i la GUI. El codi inclou un client *MQTT* que s'executa en un fil separat per garantir que la recepció de missatges no interferisca amb el rendiment de la interfície gràfica. La connexió al broker *MQTT*, configurat per funcionar en un ordinador central, permet subscriure's als temes de temperatura, humitat i alertes de caigudes per habitació. Els missatges rebuts es processen de forma eficient i es transmeten a la interfície gràfica mitjançant senyals (*signals*) i ranures (*slots*), assegurant que les actualitzacions es reflectisquen en temps real sense bloquejar el fil principal ([codi 5](#)).

Cada habitació està representada per una targeta visual en una graella central que s'adapta dinàmicament al nombre d'habitacions configurades pel personal. Les targetes mostren el nom de l'habitació i inclouen un llit, el qual canvia de color per indicar l'estat del pacient. Els colors utilitzats són verd per a un estat normal i vermell per a situacions d'alerta, com valors anòmals de temperatura o humitat, tal i com es pot veure a la [figura 31](#). En el cas d'alerta per possible caiguda el terra de l'habitació és el que canvia de color a roig, com es mostra en la [figura 32](#). Aquestes targetes també inclouen efectes visuals, com ombres i estilitzacions aplicades mitjançant *Qt Style Sheets* (QSS), per millorar l'estil del programa ([codi 6](#) i [codi 7](#)).

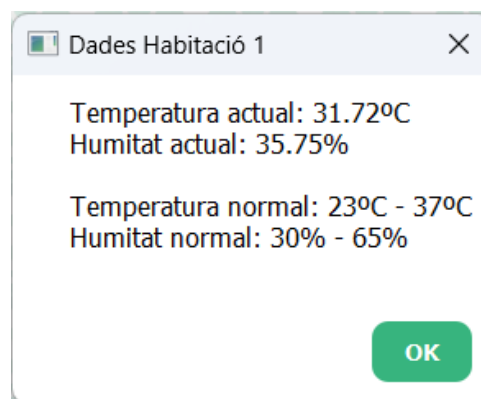


**Figura 31:** Valors anòmals de temperatura o humitat



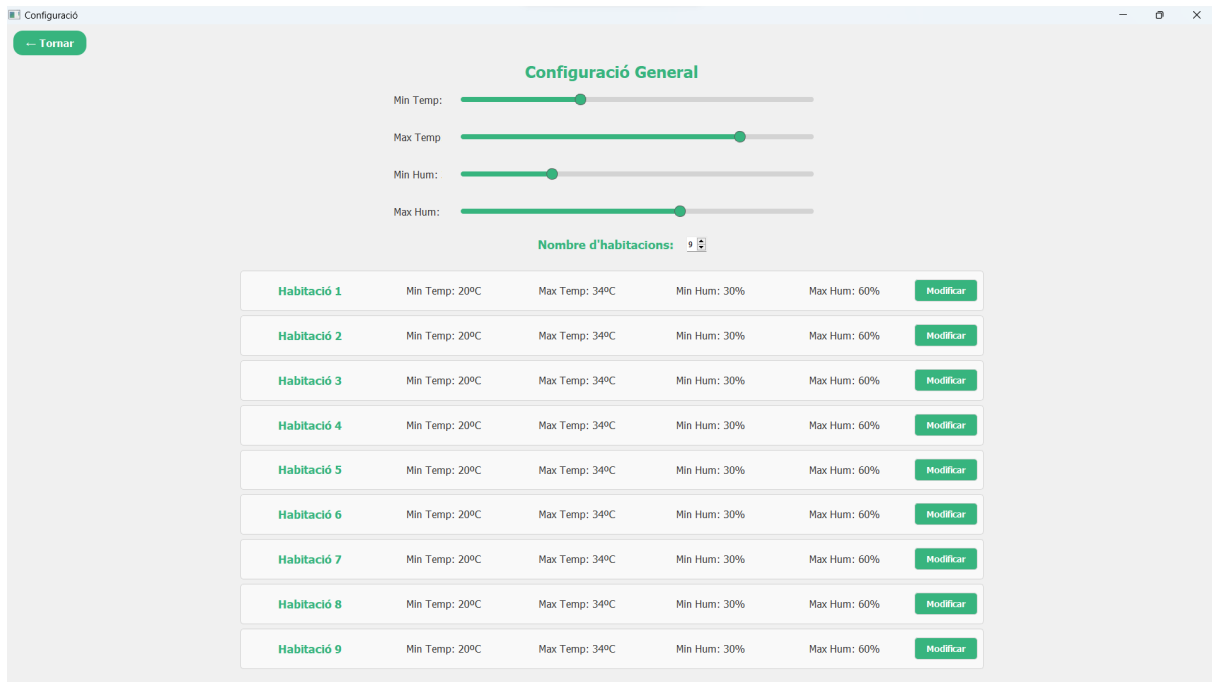
**Figura 32:** Possible caiguda detectada

Quan es fa clic sobre el llit de qualsevol habitació, es desplega una finestra que mostra informació detallada sobre l'habitació seleccionada, com mostra la [figura 33](#). En aquesta finestra, es poden consultar en temps real els valors actuals de temperatura i humitat detectats pels sensors, així com els paràmetres de configuració específics establerts per a aquesta habitació, com els llindars de temperatura i humitat normals. Aquesta funcionalitat permet al personal sanitari accedir fàcilment a dades crítiques i assegurar-se que les condicions de l'habitació són adequades per al pacient.

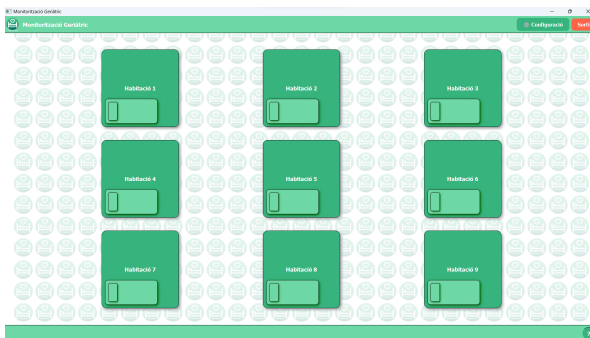


**Figura 33:** Finestra on es mostren les dades de temperatura i humitat en temps real

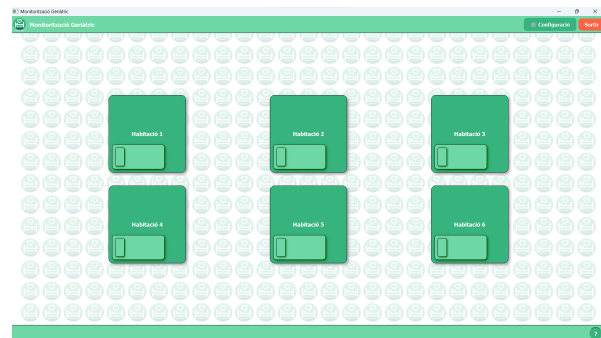
Pel que fa a la configuració, la interfície ofereix una finestra específica per ajustar els valors normals de temperatura i humitat de forma global o individual per habitació tal i com es pot veure a la [figura 34](#). A més, permet definir el nombre d'habitacions visibles, adaptant la disposició d'aquestes. Una funcionalitat útil per adaptar-se a les necessitats dels diferents centres geriàtrics on es puga implantar el projecte ([figura 35](#) i [figura 36](#)). Els ajustos fets a la finestra de configuració es propaguen immediatament al sistema gràcies a la comunicació bidireccional entre els components ([codi 8](#)).



*Figura 34: Pantalla de configuració del programa*



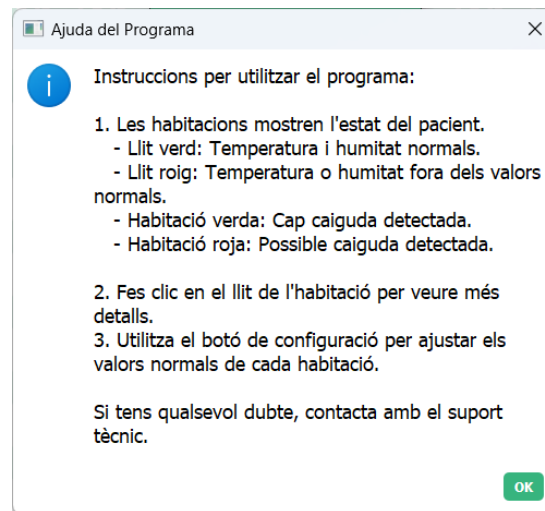
*Figura 35: Distribució amb 9 habitacions*



*Figura 36: Distribució amb 6 habitacions*

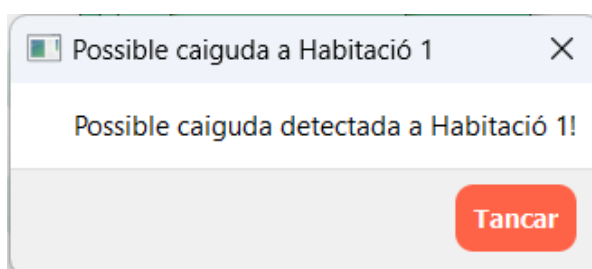


A la pantalla principal, la barra superior proporciona accés directe a funcionalitats importants com ara el menú de configuració i el botó de tancament de l'aplicació. A més, una barra inferior inclou un botó d'ajuda que ofereix instruccions detallades sobre com utilitzar el sistema (figura 37), pensat per facilitar l'adopció per part de personal sanitari amb poca experiència en tecnologia.

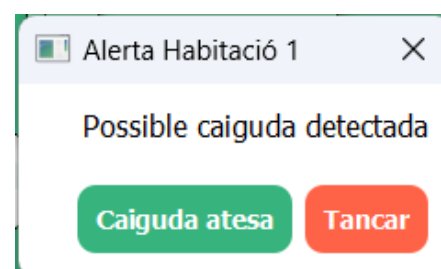


*Figura 37: Pestanya amb les isntruccions del programa*

Un altre aspecte clau és la gestió de les alertes de caigudes. Quan es detecta una caiguda, la GUI emet una notificació emergent i un so d'alarma (figura 38), alhora que actualitza visualment l'estat de l'habitació afectada. Quan es fa clic sobre una habitació on s'ha detectat una possible caiguda es mostren opcions per a marcar-la com a atesa quan el personal sanitari ja haja revisat la situació (figura 39). La funcionalitat d'alertes està implementada mitjançant un mecanisme de fils (QThread), que assegura que les notificacions es gestionen de manera asíncrona sense interrompre el funcionament general de la interfície, tal i com s'ha comentat abans.

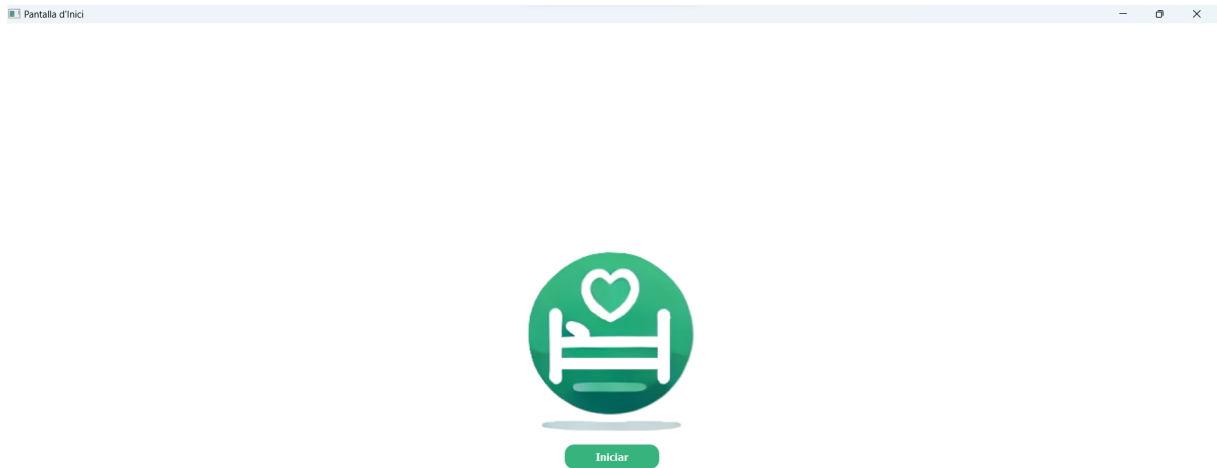


*Figura 38: Notificació de caiguda*



*Figura 39: Pestanya de caiguda atesa*

El disseny de la GUI també inclou una pantalla d'inici que actua com a punt d'entrada a l'aplicació, mostrant el logotip del projecte i un botó per accedir a la finestra principal. Aquesta pantalla d'inici és útil per fer més amigable l'experiència inicial de l'usuari i per garantir que la interfície principal s'inicia de manera ordenada després de carregar tots els components necessaris.



*Figura 40: Pantalla inicial amb el logotip del programa*

En resum, la GUI és un sistema complet que combina funcionalitats avançades amb un disseny intuïtiu, assegurant que el personal sanitari pugui supervisar de manera eficient l'estat dels pacients i respondre ràpidament a situacions crítiques. La integració del protocol *MQTT*, les notificacions visuals i sonores i les opcions de configuració personalitzades fan que aquesta interfície siga no només una eina de monitorització, sinó també una solució escalable i adaptativa per a l'atenció geriàtrica.

## Principis de la Interacció Persona-Computador

Aquesta interfície busca complir una sèrie de principis que assegurin la seva usabilitat i facilitat d'ús. En aquest projecte, s'han seguit les normes bàsiques establertes en el camp de la Interacció Persona-Ordinador (IPC) per garantir que la GUI siga intuïtiva, accessible i eficient.

Un dels principis més importants de la IPC és mantenir la **consistència visual i funcional** en tota l'aplicació. En aquesta GUI, s'ha utilitzat un esquema de colors consistent, on el verd indica un estat normal i el roig una alerta. Aquesta associació s'aplica de manera uniforme a totes les habitacions i components gràfics, assegurant que l'usuari pugui identificar fàcilment l'estat d'una habitació amb només una ullada. A més, les icones, fonts i estils visuals com els botons amb vores arrodonides i efectes d'ombra es mantenen coherents en totes les pantalles, reforçant l'experiència visual cohesionada.

La interfície, a més, està dissenyada per ser fàcil d'utilitzar, amb un enfocament en la **simplicitat**. Els elements essencials són prominents i accessibles, com ara les targetes de les habitacions, que mostren informació clara i directa sobre l'estat del pacient. Per exemple, el llit de cada habitació canvia de color segons els paràmetres monitoritzats, oferint una forma intuïtiva d'interpretar les dades. A més, l'ús de botons amb icones descriptives, com el botó d'ajuda o el botó de configuració, milloren l'experiència de l'usuari.

També es busca proporcionar **retroalimentació immediata** davant de qualsevol acció o canvi en l'estat del sistema. Quan es detecta una caiguda, l'habitació corresponent canvia immediatament de color, i es genera una notificació sonora per alertar el personal sanitari. Aquesta retroalimentació és crucial per assegurar que els usuaris reben informació en temps real, millorant així la rapidesa de resposta davant de situacions crítiques.

El disseny de la GUI s'ha centrat en **evitar la sobrecàrrega cognitiva** de l'usuari. La informació es presenta de manera ordenada i prioritzada, utilitzant una graella per organitzar les habitacions. Les alertes visuals i sonores es limiten a situacions realment crítiques, com la detecció d'una caiguda, per evitar que el personal sanitari siga aclaparat per notificacions innecessàries. A més, els detalls addicionals, com la temperatura i la humitat exactes, només són visibles si l'usuari fa clic en l'habitació corresponent, permetent un accés fàcil a la informació sense saturar la pantalla principal.

Un altre principi clau respectat és proporcionar **control a l'usuari**. L'aplicació inclou una finestra de configuració que permet personalitzar els paràmetres normals de temperatura i humitat per a cada habitació. Aquesta funcionalitat assegura que el sistema es pot ajustar a les necessitats específiques de cada situació. A més, l'usuari pot marcar manualment una alerta com a resolta o ajustar el nombre d'habitacions visibles segons les necessitats operatives.

Finalment, la interfície està dissenyada per ser **fàcilment comprensible** per a usuaris amb diferents nivells d'experiència tècnica. L'ús de colors vius, icones clares i fonts llegibles contribueixen a una experiència visual accessible. A més, el botó d'ajuda ofereix instruccions pas a pas sobre com utilitzar l'aplicació, cosa que facilita l'aprenentatge inicial per part del personal menys familiaritzat amb la tecnologia.

### 3.2.5 Integració de Components

La integració de tots els components del sistema ha sigut un dels punts més crucials en el desenvolupament d'aquest projecte, assegurant que el conjunt de sensors, unitats de processament, mecanismes de comunicació i interfície gràfica treballen de manera coordinada i eficient. Aquesta secció descriu com s'han interconnectat i sincronitzat els elements del sistema per aconseguir una funcionalitat plena, mantenint una arquitectura modular i escalable.

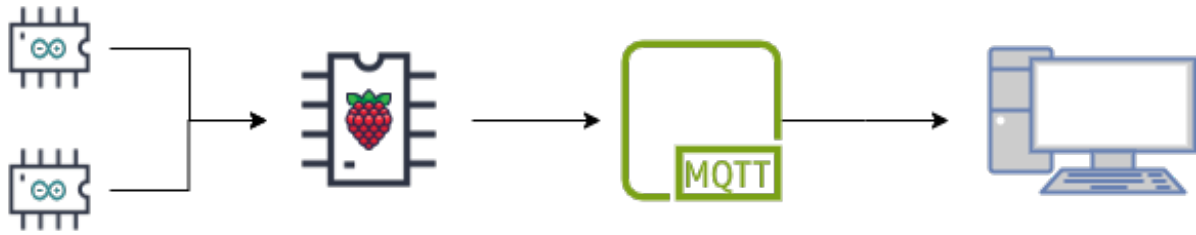
El sistema es construeix sobre una base d'interconnexió que combina components de maquinari i programari. En la part de maquinari, els **Arduino Nano 33 BLE Sense** actuen com els nodes de recollida de dades, capturant informació crítica sobre la temperatura, la humitat i els patrons de moviment. Cada Arduino està equipat amb sensors integrats que detecten variacions en el llit del pacient o en el terra de l'habitació. Aquesta informació es processa localment mitjançant models optimitzats de TinyML abans de ser enviada a la **Raspberry Pi 4**, que funciona com un node intermedi de processament i comunicació.

La Raspberry Pi 4 reuneix les dades enviades pels sensors a través de connexions sèrie i les prepara per ser transmeses al broker **MQTT**. Aquest broker, instal·lat en un ordinador central, actua com a punt de distribució de la informació cap a la interfície gràfica d'usuari (GUI). El protocol MQTT facilita una comunicació eficient, assegurant que les dades es publiquen en temps real i que la GUI reba informació actualitzada sobre l'estat de cada habitació. El flux de dades es realitza utilitzant temes específics per a cada habitació i cada tipus de sensor, assegurant una gestió organitzada i escalable de les dades.

La **GUI**, construïda amb PyQt5, rep els missatges publicats pel broker MQTT i actualitza visualment l'estat de les habitacions. Aquesta interfície no només mostra les condicions normals dels sensors, sinó que també genera alertes visuals i sonores quan es detecten situacions anòmales, com una possible caiguda o una alteració significativa en els paràmetres de temperatura o humitat. Els elements visuals de la GUI es redissenyen en temps real mitjançant mecanismes de senyals i ranures, garantint que les actualitzacions no interferisquen amb el rendiment global del sistema.

La integració de tots aquests elements està dissenyada per garantir una operació contínua i fiable. A més, el sistema és escalable, permetent afegir noves habitacions o sensors simplement configurant nous temes MQTT i ajustant la GUI perquè mostre els nous elements. Aquesta modularitat assegura que el projecte pot ser ampliat fàcilment per a satisfer les necessitats específiques de cada centre geriàtric.

Per a proporcionar una visió general de la connexió entre els components, es presenta el diagrama de la [figura 41](#). Aquest esquema il·lustra les relacions entre els sensors, la Raspberry Pi, el broker MQTT i la GUI, destacant el flux de dades i la sincronització entre els elements del sistema.



*Figura 41: Diagrama de la integració dels components del sistema.*

En aquest diagrama, es pot observar com els sensors Arduino transmeten les dades a la Raspberry Pi, que després publica la informació al broker MQTT. Aquest broker distribueix els missatges cap a la GUI, on es processen per ser representats gràficament. Aquesta arquitectura permet que cada component opere de manera independent, mantenint una coordinació global per oferir una experiència robusta i en temps real al personal sanitari.

En resum, la integració dels components en aquest projecte ha sigut dissenyada amb una atenció especial a la modularitat, la fiabilitat i l'eficiència. El resultat és un sistema que combina hardware i software de manera harmònica.

## 4 Validació del sistema

En aquest capítol es descriu el procés de validació del sistema, que inclou l'avaluació dels sensors, la comunicació entre components i el funcionament complet del sistema. Es presenten les proves realitzades per verificar la precisió dels sensors, la robustesa de la comunicació amb MQTT i la resposta en temps real de la GUI. També es valida la integració dels diferents elements mitjançant escenaris simulats en la maqueta física, assegurant que el sistema compleix els requisits i objectius establerts.

### 4.1 Validació de Sensors

La validació dels sensors és un pas fonamental per garantir que el sistema de monitorització compleix amb els requisits establerts, oferint dades fiables i responent de manera adequada a les diferents situacions que es poden produir en una habitació geriàtrica. Aquesta validació s'ha realitzat utilitzant la maqueta física dissenyada per al projecte, reproduint condicions que simulen escenaris reals en un centre geriàtric.

#### 4.1.1 Validació del sensor de temperatura i humitat

El sensor de temperatura i humitat integrat en l'Arduino Nano 33 BLE Sense ha estat sotmès a diverses proves que busquen assegurar la seua capacitat per detectar canvis en els paràmetres ambientals del llit del pacient. Aquestes proves inclouen l'exposició del sensor a variacions de temperatura i humitat controlades, simulant situacions com un increment sobtat d'humitat al llit causat per una possible incontinència.

Durant les proves, el sensor ha demostrat respondre de manera consistent als canvis introduïts, proporcionant lectures que permeten identificar anomalies en pocs segons. Per exemple, quan es simulava un increment d'humitat, el sistema activava automàticament una alerta en la interfície gràfica, destacant l'habitació afectada per tal de notificar al personal sanitari. Aquesta resposta ràpida confirma que el sensor és adequat per a l'ús previst i que pot contribuir a millorar la detecció d'aquestes situacions en un entorn real.

#### 4.1.2 Validació de l'acceleròmetre per a la detecció de caigudes

El sensor d'acceleròmetre, responsable de detectar moviments al terra de l'habitació, ha sigut provat simulant tant activitats normals com caigudes sobtades. Es van reproduir escenaris habituals, com passos al voltant del llit, així com situacions més crítiques on es representaven moviments que podrien ser considerats anòmals, com caigudes.

El sistema ha demostrat una bona capacitat per diferenciar entre aquests escenaris. En situacions normals, el sensor registrava els moviments sense activar cap alerta, mentre que davant d'una simulació de caiguda, es generava una notificació immediata. Aquest comportament confirma que el model de detecció d'anomalies entrenat amb Edge Impulse és fiable i que el sensor pot proporcionar informació rellevant per garantir la seguretat dels pacients.

### 4.1.3 Robustesa dels Sensors

Per assegurar que els sensors funcionen de manera consistent en diferents condicions, s'han realitzat proves per verificar la seua robustesa davant de possibles interferències ambientals. Per exemple, es van introduir canvis sobtats en la temperatura i petites vibracions en el terra per simular situacions quotidianes que podrien afectar la precisió dels sensors.

Els resultats han mostrat que els sensors són capaços de mantenir lectures estables i fiables. Això és especialment important ja que els sensors poden estar exposats a variacions constants sense que això afecte negativament la seua funció principal.

## 4.2 Validació de la Comunicació

La comunicació entre els diferents components del sistema és un element clau per garantir el correcte funcionament de la monitorització en temps real. Per validar aquesta part del projecte, s'han realitzat proves exhaustives per assegurar que el flux de dades entre els sensors, la Raspberry Pi, el broker MQTT i la interfície gràfica (GUI) funciona de manera fluida i fiable en totes les condicions previstes.

### 4.2.1 Proves de Connexió entre Arduino i Raspberry Pi

En primer lloc, es va verificar la connexió entre els sensors Arduino Nano 33 BLE Sense i la Raspberry Pi. Aquesta comunicació es basa en una interfície sèrie, on les dades capturades pels sensors es transmeten a la Raspberry Pi en temps real. Durant les proves, es van simular escenaris reals amb canvis constants en els paràmetres capturats pels sensors, com variacions de temperatura o simulacions de moviment.

Les proves han demostrat que la transmissió de dades és consistent, amb una latència mínima i sense pèrdua d'informació. Aquesta estabilitat és essencial, ja que garanteix que les dades arriben a la Raspberry Pi en el moment precís, permetent la seua posterior publicació al broker MQTT.

### 4.2.2 Proves de Publicació i Subscripció amb el Broker MQTT

El broker MQTT instal·lat en un ordinador central actua com a nucli de comunicació del sistema, gestionant la publicació i distribució de missatges entre els diferents components. Per validar el seu funcionament, es van crear diferents temes MQTT que corresponen a les dades dels sensors, com `geriatric/habitacio1/temperatura` o `geriatric/habitacio1/alerta`.

Durant la configuració inicial, es va identificar un problema relacionat amb l'accés al port 1883, utilitzat pel broker per gestionar les connexions entrants. El sistema operatiu bloquejava aquest port per defecte, impeding que la Raspberry Pi i la GUI pogueren connectar-se correctament al broker. Per resoldre aquesta situació, es va crear manualment una regla d'entrada en el tallafocs de l'ordinador central, permetent l'accés al port necessari. Aquesta configuració es va realitzar mitjançant les opcions de configuració del tallafocs del sistema, assegurant que el broker MQTT funcionara sense interferències en totes les proves posteriors.

Un cop resolta aquesta incidència, la Raspberry Pi publicava contínuament dades en els temes definits, mentre que la GUI es subscribia per rebre la informació. S'ha comprovat que el broker gestiona correctament tant la publicació com la distribució dels missatges, fins i tot quan es produeixen canvis constants o simultanis en diverses habitacions. Aquest comportament assegura que el sistema és escalable i pot suportar un increment en el nombre de sensors o habitacions monitoritzades sense comprometre el rendiment.

### **4.2.3 Validació de la Sincronització amb la GUI**

La interfície gràfica d'usuari (GUI) és l'element final del flux de comunicació, responsable de processar i mostrar la informació en temps real. Per validar aquesta sincronització, es van realitzar proves on es publicaven dades en els temes MQTT simulant situacions típiques d'un centre geriàtric, com l'augment d'humitat al llit o la detecció d'una possible caiguda.

Els resultats van mostrar que la GUI respon immediatament a les actualitzacions dels sensors, canviant els colors de les habitacions i generant alertes visuals i sonores sense retard perceptible. Aquest rendiment en temps real és crucial per assegurar que el personal sanitari pot actuar ràpidament davant de qualsevol situació crítica.

### **4.2.4 Proves d'Estabilitat i Robustesa de la Comunicació**

Finalment, es van realitzar proves prolongades per garantir l'estabilitat de la comunicació en el sistema. Aquestes proves incloïen la simulació de condicions on la xarxa podria experimentar interrupcions temporals o fluctuacions en l'amplada de banda. Es va observar que, en cas de desconexió temporal entre algun dels components, el sistema és capaç de recuperar la comunicació automàticament sense necessitat d'intervenció manual.

Aquesta capacitat de resiliència confirma que el sistema pot operar de manera fiable en entorns reals, on poden produir-se eventualitats tècniques que afecten la xarxa o els dispositius hardware.



### 4.3 Validació de la Interfície Gràfica d'Usuari (GUI)

La interfície gràfica d'usuari (GUI) constitueix el punt central d'interacció entre el sistema i el personal sanitari, de manera que la seva validació s'ha centrat en garantir que compleix amb els requisits d'usabilitat, fiabilitat i eficiència en temps real. Per avaluar aquestes característiques, es van dur a terme proves centrades en tres àmbits principals: funcionalitat general, gestió d'alertes i resposta a situacions simulades.

La validació funcional de la GUI es va iniciar comprovant el correcte desplegament i visualització de les targetes associades a les habitacions monitoritzades. Es van simular entorns amb diferents nombres d'habitacions, ajustant les configuracions globals per assegurar que la interfície era capaç d'adaptar-se dinàmicament a un increment o disminució del nombre d'habitacions. En tots els casos, la GUI va mostrar les habitacions correctament, amb les seves corresponents icones, colors i efectes visuals segons els paràmetres configurats.

En segon lloc, es va validar la gestió d'alertes simulant situacions anòmales. Per a això, es van introduir manualment dades MQTT que simulaven paràmetres anòmals, com ara temperatures elevades, humitat excessiva o caigudes. La GUI va reaccionar immediatament a cada alerta, modificant els colors de les targetes corresponents i emetent notificacions visuals i sonores per a cridar l'atenció de l'usuari. A més, es va comprovar que el sistema permetia al personal marcar manualment les alertes com a ateses, actualitzant l'estat de la targeta i eliminant la notificació de l'habitació corresponent. Durant les proves també es va analitzar el rendiment de la GUI en condicions d'alta càrrega, simulant l'arribada simultània de missatges MQTT de diverses habitacions. Gràcies a la implementació de fils separats per a la recepció de dades i la representació gràfica, es va observar que la interfície mantenia una fluïdesa òptima sense bloquejos o retards perceptibles, fins i tot en situacions de màxima activitat.

```
C:\Program Files\mosquitto>mosquitto_pub -h localhost -t geriatric/habitacio1/temperatura -m "23.5"  
C:\Program Files\mosquitto>mosquitto_pub -h localhost -t geriatric/habitacio1/humetat -m "50"  
C:\Program Files\mosquitto>mosquitto_pub -h localhost -t geriatric/habitacio1/alerta -m "Caiguda detectada"
```

*Figura 42: Exemple d'una prova de missatges MQTT per terminal*

En conclusió, les proves realitzades confirmen que la GUI no només és funcional i fiable, sinó també accessible i fàcil d'utilitzar, complint amb els objectius establerts per al projecte. Aquesta validació assegura que el sistema és capaç d'oferir una experiència d'usuari òptima, facilitant la supervisió i resposta a situacions de risc en els centres geriàtrics.

#### 4.4 Validació del Sistema Complet

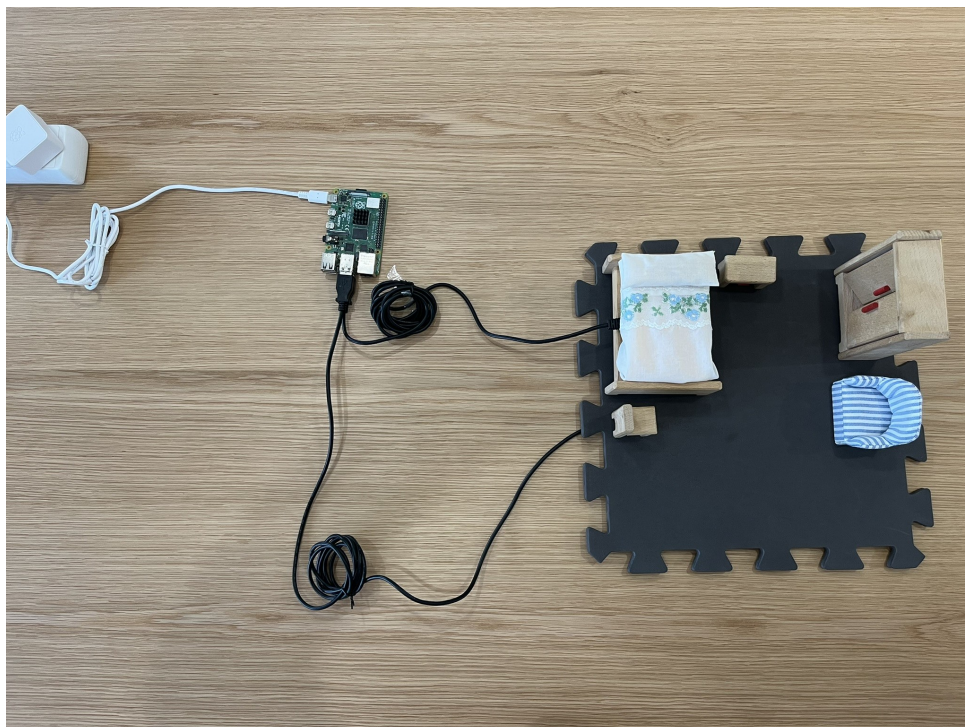
Una vegada completades les proves individuals de cada component, es va procedir a validar el sistema complet utilitzant la maqueta a escala, que simula una habitació d'un centre geriàtric. Aquesta validació es va dur a terme per avaluar el comportament del sistema quan tots els elements treballen de manera integrada, incloent les plaques Arduino, la Raspberry Pi, el broker MQTT i la interfície gràfica d'usuari (GUI).

La maqueta va ser configurada amb els sensors col·locats en les seues posicions finals: el sensor de temperatura i humitat instal·lat al llit i el sensor de moviment col·locat al terra. Tots els components estaven connectats al sistema central mitjançant la Raspberry Pi, que transmetia les dades al broker MQTT, i posteriorment, aquestes es mostraven a la GUI.

Per validar el sistema complet, es van realitzar proves pràctiques simulant escenaris reals dins de la maqueta. Els escenaris incloïen l'alteració controlada de la temperatura i la humitat al llit, així com la simulació de caigudes mitjançant moviments bruscs. En cada cas, es va observar que les dades generades pels sensors es transmetien correctament a través de la Raspberry Pi fins al broker MQTT i, finalment, es reflectien en temps real a la GUI.

Els resultats obtinguts amb la maqueta van ser satisfactoris. Per exemple, en el cas de detectar una caiguda simulada, la GUI canviava immediatament el color de l'habitació corresponent, emetia un so d'alerta i mostrava una notificació visual. Igualment, es va comprovar que els canvis en temperatura i humitat es registraven correctament, i quan es superaven els límits establerts, el sistema generava les alertes adequades.

Finalment, aquesta validació va demostrar que el sistema complet, quan es troba en operació conjuntament amb la maqueta, és capaç de simular correctament els escenaris reals previstos. La coordinació dels components garanteix el seu funcionament, proporcionant una base sòlida per a la seua implementació en un entorn real.



*Figura 43: Imatge de la prova amb el sistema complet amb tots els elements connectats.*

## 5 Conclusions i Treball Futur

### 5.1 Conclusions

El projecte desenvolupat, centrat en el disseny i implementació d'un sistema de sensors intel·ligent per a la supervisió de pacients en habitacions de centres geriàtrics, ha complert amb els objectius generals i específics plantejats inicialment, oferint una solució pràctica, assequible i escalable per a millorar la seguretat i el benestar dels residents en aquest tipus d'instal·lacions.

Pel que fa a l'objectiu general, el sistema ha demostrat la seua capacitat per detectar situacions de risc com caigudes, episodis d'incontinència o anomalies en la temperatura corporal, alertant el personal sanitari de manera ràpida i efectiva. La combinació de sensors basats en Arduino Nano 33 BLE Sense, un node intermedi de processament com la Raspberry Pi i una interfície gràfica amigable ha proporcionat una solució completa que respon a la necessitat de monitorització constant i no intrusiva dels pacients. Dins dels objectius específics, la mesura de paràmetres crítics com la temperatura i la humitat del llit ha estat altament satisfactòria gràcies al sensor HTS221 integrat. La detecció de caigudes, implementada amb un model optimitzat de *TinyML* entrenat mitjançant *Edge Impulse*, ha aconseguit una precisió destacable, oferint resultats fiables en condicions simulades amb una maqueta que reproduïx l'entorn d'una habitació geriàtrica real.

Un altre punt clau és l'ús del protocol *MQTT* com a mecanisme central de comunicació. Aquesta elecció ha permès establir una infraestructura de transmissió de dades en temps real, assegurant que els sensors connectats a la Raspberry Pi envien la informació al broker i, des d'allí, a la GUI. El sistema ha demostrat ser robust i funcional, proporcionant una experiència fluida en totes les fases de monitorització.

La GUI desenvolupada amb PyQt5 ha complert les expectatives pel que fa a usabilitat i disseny intuïtiu. Els elements visuals, com el canvi de colors en funció de l'estat del pacient i les alertes visuals i sonores, han creat una eina efectiva per al personal sanitari. Aquesta interfície no només proporciona informació clara i directa, sinó que també és adaptable a diferents entorns, permetent configurar paràmetres com el nombre d'habitacions o els llindars de temperatura i humitat. Un aspecte fonamental del projecte ha estat la seua alineació amb els principis de no intrusivitat i privacitat. La decisió d'utilitzar sensors col·locats estratègicament al llit i al terra de l'habitació evita sistemes més invasius, garantint que el sistema compleix amb els estàndards ètics i les expectatives dels pacients. A més, la modularitat del sistema permet la seua escalabilitat, possibilitant la seua implementació en centres geriàtrics amb diferents necessitats.

Finalment, l'ús de components accessibles com Arduino i Raspberry Pi assegura que aquesta tecnologia pugui ser implementada en centres geriàtrics amb recursos limitats. Aquest enfocament, juntament amb la simplicitat en la instal·lació i configuració del sistema, fa que la solució siga una opció viable i atractiva per a un sector amb creixent necessitat de solucions tecnològiques eficients.

En resum, aquest treball ha demostrat que la combinació de tecnologies assequibles i estratègies centrades en modularitat, escalabilitat i privacitat poden donar lloc a un sistema robust i eficient. Tot i que les proves s'han limitat a condicions simulades, els resultats proporcionen una base sòlida per considerar l'aplicació d'aquest sistema en entorns reals, amb possibles millores segons les necessitats dels centres geriàtrics.

## 5.2 Treball Futur

Tot i els resultats satisfactoris aconseguits durant el desenvolupament d'aquest projecte, es poden identificar diverses àrees de millora i ampliació que podrien reforçar encara més la funcionalitat, escalabilitat i aplicabilitat del sistema en entorns reals. Aquestes propostes de treball futur tenen com a objectiu abordar limitacions actuals i explorar noves possibilitats per optimitzar el rendiment i ampliar l'abast del sistema.

En primer lloc, una ampliació natural seria la realització de proves exhaustives en entorns reals. Encara que les proves realitzades amb la maqueta han sigut satisfactòries, implementar el sistema en un centre geriàtric real permetria obtenir dades més completes sobre el seu rendiment en situacions del dia a dia. Això inclouria l'avaluació de la robustesa del sistema en condicions ambientals diverses i la interacció amb pacients i personal sanitari. Aquestes proves podrien identificar possibles millores en la interfície gràfica o en la configuració dels sensors per adaptar-se millor a les necessitats reals.

Una altra àrea clau de millora seria permetre la gestió de múltiples plantes dins d'un mateix centre geriàtric. Aquesta funcionalitat seria especialment útil per a centres de grans dimensions, on es podrien definir diferents configuracions per a cada planta. La interfície gràfica podria incloure un selector que permetera alternar entre les plantes, cadascuna amb el seu nombre d'habitacions i configuracions personalitzades. Aquesta ampliació requeriria una adaptació del sistema de temes MQTT per incloure identificadors específics per a cada planta, assegurant una gestió eficient i organitzada de les dades.

En termes d'adaptabilitat, el sistema també podria ser ajustat per al seu ús en cases particulars. Aquesta adaptació implicaria reduir l'escala del projecte per a funcionar en entorns amb un nombre menor d'habitacions, oferint una solució personalitzada per a famílies que necessiten monitoritzar persones majors o amb necessitats especials a casa. A més, es podria dissenyar una versió més senzilla de la interfície gràfica, optimitzada per a dispositius mòbils o tauletes, que permetera una supervisió fàcil i accessible en aquests entorns més reduïts. Fins i tot es podria considerar l'escalabilitat del projecte per a altres aplicacions més enllà de l'àmbit geriàtric. Per exemple, el sistema podria adaptar-se per a la monitorització de pacients en hospitals, persones amb discapacitats a les seues llars o fins i tot per a la supervisió d'indrets com guarderies o residències temporals.

Pel que fa al consum energètic, encara que els components seleccionats són de baix consum, implementar estratègies d'estalvi energètic, com l'ús de modes de repòs en els sensors o la reducció de la freqüència de mostreig quan no es detecten anomalies, podria prolongar la vida útil dels dispositius sense comprometre la seua eficàcia. Aquesta característica seria especialment rellevant per a aplicacions on els dispositius no disposen d'una font d'alimentació contínua.

En resum, les possibilitats per a treballs futurs són àmplies i prometedores. La base desenvolupada en aquest projecte proporciona un punt de partida sòlid per continuar innovant i millorant, amb l'objectiu final de crear un sistema encara més eficaç, adaptable i integrador per a la monitorització de pacients en diferents entorns.

## 6 Bibliografia

Arduino. (2024a). Arduino Integrated Development Environment (IDE). <https://www.arduino.cc/en/software>

Arduino. (2024b). Arduino Nano 33 BLE Sense. <https://docs.arduino.cc/hardware/nano-33-ble-sense/>

Arduino. (2024c). What is Arduino? <https://www.arduino.cc/en/Guide/Introduction>

Arduino Team. (2023). Arduino Nano 33 BLE Sense: Technical Features and Use Cases [Accessed: 2024-11-20]. <https://www.arduino.cc/en/Guide/NANO33BLESense>

Cavagnis, F. (2023). TensorFlow Lite Workflow. <https://www.researchgate.net/publication/376715530/figure/fig1/AS:11431281213855128@1703190167572/Tensorflow-Lite-Workflow-adapted-from-Cavagnis-2023.jpg>

Centre Tecnològic Cartif. (2021). Sistemes basats en la visió per computador per a la monitorització de pacients geriàtrics. *UVA Tech Report*. <https://ucc.uva.es/un-sistema-de-monitorizacio-no-intrusiva-permite-una-mayor-autonomia-de-las-personas-mayores-en-el-hogar/>

Conselleria de Sanitat Universal i Salut Pública, G. V. (2021). Guia de cures per a persones amb incontinència. <https://www.san.gva.es/documents/d/farmacia-i-productes-sanitaris/01032021-guia-cuidados-personas-incontinencia-valenciano>

DataCamp. (2024). What is TinyML? Tiny Machine Learning Overview. <https://www.datacamp.com/blog/what-is-tinyml-tiny-machine-learning>

Eclipse Foundation. (2024). Eclipse Mosquitto: An open source MQTT broker. <https://mosquitto.org/>

Edge Impulse. (2023). *Spectral features*. <https://docs.edgeimpulse.com/docs/edge-impulse-studio/processing-blocks/spectral-features>

Edge Impulse. (2024). Edge Impulse: The Leading Machine Learning Platform for Embedded Devices. <https://www.edgeimpulse.com>

Edge Impulse Forum. (2024). Error: INFO: HW RFFT failed, falling back to SW. <https://forum.edgeimpulse.com/t/error-info-hw-rfft-failed-falling-back-to-sw/12613>

Fundació iSocial. (2022). InvisiCare: Monitorització sense sensors de les llars de les persones grans. <https://isocial.cat/invisicare-monitoritzacio-sense-sensors-de-les-llars-de-les-persones-grans/>

Institut Nacional d'Estadística (INE). (2023). Defuncions per caigudes accidentals 2023. <https://www.regio7.cat/societat/2024/07/25/les-morts-per-caiguda-persones-106053344.html>

Keras. (2024). Keras: The Python Deep Learning API. <https://keras.io/>

López Chaveinte, I. (2015). Sistema de monitorización de pacientes mediante una red de sensores inalámbricos. <https://openaccess.uoc.edu/handle/10609/42770?locale=es>

Medium. (n.d.-a). Image of Tensorflow Workflow Overview. [https://miro.medium.com/v2/resize:fit:828/format:webp/1\\*jMDbfgdwc9VIj5JHwY\\_y9g.jpeg](https://miro.medium.com/v2/resize:fit:828/format:webp/1*jMDbfgdwc9VIj5JHwY_y9g.jpeg)

Medium. (n.d.-b). Tensorflow Lite Overview Image. [https://miro.medium.com/v2/resize:fit:828/format:webp/0\\*\\_f9A4lvebQzp2JTX.png](https://miro.medium.com/v2/resize:fit:828/format:webp/0*_f9A4lvebQzp2JTX.png)

MQTT.org. (2024). MQTT: The Standard for IoT Messaging. <https://mqtt.org/>

Raspberry Pi Foundation. (2024a). Què és Raspberry Pi? <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>

Raspberry Pi Foundation. (2024b). Raspberry Pi 4 Model B Specifications. <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>

TensorFlow. (2024a). TensorFlow Lite for Microcontrollers. <https://www.tensorflow.org/lite/microcontrollers>

TensorFlow. (2024b). TensorFlow: An end-to-end open source platform for machine learning. <https://www.tensorflow.org/>

Universitat de Saragossa. (2019). Sistemes basats en iBeacons per a la monitorització de moviment. <https://zagan.unizar.es/record/85910>

Universitat Politècnica de València. (2023). Sistema híbrid de monitorització de pacients mitjançant sensors i micròfons ambientals. <https://www.upv.es/investigacio/hybrid-systems.html>

Wikipedia. (2023a). Loss functions for classification. [https://en.wikipedia.org/wiki/Loss\\_functions\\_for\\_classification](https://en.wikipedia.org/wiki/Loss_functions_for_classification)

Wikipedia. (2023b). *Rectifier (neural networks)*. [https://en.wikipedia.org/wiki/Rectifier\\_\(neural\\_networks\)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks))

## Annex I - Codi

*Codi 1: Codi per a la lectura de la temperatura i la humitat*

```
#include <Arduino_HTS221.h> // Biblioteca per al sensor HTS221

void setup() {
  // Inicialitzar la comunicació sèrie
  Serial.begin(9600);
  while (!Serial);

  // Inicialitzar el sensor HTS221
  if (!HTS.begin()) {
    Serial.println("Error inicialitzant el sensor HTS221!");
    while (1);
  }

  Serial.println("Sensor HTS221 inicialitzat correctament.");
}

void loop() {
  // Llegir temperatura i humitat del sensor HTS221
  float temperatura = HTS.readTemperature();
  float humitat = HTS.readHumidity();

  // Enviar els valors per sèrie
  Serial.print("Temperatura:");
  Serial.print(temperatura);
  Serial.print(";Humitat:");
  Serial.println(humitat);

  // Esperar 5 segons abans de la següent lectura
  delay(5000);
}
```

*Codi 2: Fragment del codi per a la detecció de caigudes*

```

/**
 * @brief      Run inferencing in the background.
 */
void run_inference_background()
{
    // wait until we have a full buffer
    delay((EI_CLASSIFIER_INTERVAL_MS * EI_CLASSIFIER_RAW_SAMPLE_COUNT)
    + 100);

    // This is a structure that smoothens the output result
    // With the default settings 70% of readings should be the same before
    classifying.
    ei_classifier_smooth_t smooth;
    ei_classifier_smooth_init(&smooth, 10 /* no. of readings */, 7 /* min.
    readings the same */, 0.8 /* min. confidence */, 0.3 /* max anomaly */);

    while (1) {
        // copy the buffer
        memcpy(inference_buffer, buffer, EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE
        * sizeof(float));

        // Turn the raw buffer in a signal which we can the classify
        signal_t signal;
        int err = numpy::signal_from_buffer(inference_buffer,
        EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE, &signal);
        if (err != 0) {
            ei_printf("Failed to create signal from buffer (%d)\n", err);
            return;
        }

        // Run the classifier
        ei_impulse_result_t result = { 0 };

        err = run_classifier(&signal, &result, debug_nn);
        if (err != EI_IMPULSE_OK) {
            ei_printf("ERR: Failed to run classifier (%d)\n", err);
            return;
        }

        // print the predictions
        /*ei_printf("Predictions ");
        ei_printf("(DSP: %d ms., Classification: %d ms., Anomaly: %d ms.,
        anomaly score: %.3f\n)",
            result.timing.dsp, result.timing.classification,
            result.timing.anomaly, result.anomaly);
        ei_printf(": ");
*/
        if (result.anomaly > 5.0) {
            Serial.println("caiguda");
        }

        ...
    }
}

```



**Codi 3:** *Fragment del codi per a la configuració MQTT i dels ports sèrie a la Raspberry Pi*

```
# Configuració MQTT
MQTT_BROKER = "192.168.1.89" # IP del broker MQTT
MQTT_PORT = 1883
TOPIC_TEMPERATURE = "geriatric/habitacio1/temperatura"
TOPIC_HUMIDITY = "geriatric/habitacio1/humitat"
TOPIC_ALERTA_CAIGUDA = "geriatric/habitacio1/alerta"

# Configuració ports sèrie
SERIAL_PORT_TEMPUM = '/dev/ttyACM1' # Port de temperatura i humitat
BAUD_RATE_TEMPUM = 9600

SERIAL_PORT_CAIGUDA = '/dev/ttyACM0' # Port per a la detecció de caigudes
BAUD_RATE_CAIGUDA = 115200

# Inicialitzar els ports sèrie
ser_temphum = serial.Serial(SERIAL_PORT_TEMPUM, BAUD_RATE_TEMPUM,
timeout=1)
ser_caiguda = serial.Serial(SERIAL_PORT_CAIGUDA, BAUD_RATE_CAIGUDA,
timeout=1)

# Funció per enviar dades MQTT
def send_mqtt_data(client, temperature, humidity):
    client.publish(TOPIC_TEMPERATURE, temperature)
    client.publish(TOPIC_HUMIDITY, humidity)

def send_mqtt_alert(client):
    client.publish(TOPIC_ALERTA_CAIGUDA, "Caiguda detectada")

# Inicialitzar el client MQTT
client = mqtt.Client()
client.connect(MQTT_BROKER, MQTT_PORT, 60)
```

**Codi 4:** Fragment del codi per a llegir les dades dels Arduinos i enviar-les via MQTT

```

# Bucle per llegir dades dels ports sèrie i enviar-les per MQTT
try:
    while True:
        # Llegir dades de temperatura i humitat
        if ser_temphum.in_waiting > 0:
            line = ser_temphum.readline().decode('utf-8').strip()
            if line:
                print(f"Dades rebudes: {line}")
                # Dividir les dades rebudes
                parts = line.split(';')
                if len(parts) == 2:
                    temperatura = parts[0].split(':')[1]
                    humitat = parts[1].split(':')[1]
                    print(f"Temperatura: {temperatura} C , Humitat:
                    {humitat} %")
                    # Enviar les dades per MQTT
                    send_mqtt_data(client, temperatura, humitat)

        # Llegir dades de caigudes
        if ser_caiguda.in_waiting > 0:
            line = ser_caiguda.readline().decode('utf-8').strip()
            if line == "caiguda":
                current_time = time.time()
                # Avisar de la caiguda una sola vegada
                if current_time - last_caiguda_time >= CAIGUDA_INTERVAL:
                    print("Caiguda detectada")
                    send_mqtt_alert(client)
                    last_caiguda_time = current_time

            time.sleep(1)

except KeyboardInterrupt:
    print("Programa interromput.")

```

**Codi 5:** Fragment del codi per a subscriure's als temes definits i inicia el bucle del client MQTT en un fil separat

```

def connect_mqtt(self):
    client = mqtt.Client()
    client.on_message = self.on_message
    client.connect(MQTT_BROKER, MQTT_PORT, 60)

    for topics in HABITACIONES.values():
        client.subscribe(topics["temp"])
        client.subscribe(topics["hum"])
        client.subscribe(topics["alerta"])

    client.loop_start() # Executar el client MQTT en un fil separat
    return client

```

**Codi 6:** Fragment del codi per a canviar el color del llit dependent de les dades dels sensors

```

if (self.temp_min_norm <= self.temperatura_actual <= self.temp_max_norm) and \
    (self.hum_min_norm <= self.humitat_actual <= self.hum_max_norm):
    color = "#6DD8A5" # Verd clar per a estat normal
else:
    color = "#FF6347" # Roig per a estat anòmal

self.llit_label.setStyleSheet(f"""
    QPushButton {{
        background-color: {color}; /* Canvia segons l'estat */
        border-radius: 10px;
        border: 1px solid black;
    }}
""")

```

**Codi 7:** Fragment del codi que aplica l'estil desitjat a les habitacions

```

# Quadrat que representa l'habitació
self.habitacio_label = QPushButton(self.nom, self)
self.habitacio_label.setFixedSize(250, 250)
self.habitacio_label.setStyleSheet("""
    QPushButton {
        background-color: #37B47E;
        color: white;
        border-radius: 15px;
        border: 1px solid black;
        font-family: 'Eurostile', sans-serif;
        font-size: 18px;
        font-weight: bold;
    }
""")

# Efecte d'ombra
ombra = QGraphicsDropShadowEffect()
ombra.setBlurRadius(20) # Nivell de difuminat de l'ombra
ombra.setOffset(5, 5) # Offset de l'ombra (desplaament)
ombra.setColor(Qt.gray) # Color de l'ombra

```

**Codi 8:** Fragment del codi que mostra com s'actualitzen immediatament els valors de configuració

```

def crear_slider(self, prefix, min_val, max_val, valor_actual, tipus):
    slider = QSlider(Qt.Horizontal)
    slider.setRange(min_val, max_val)
    slider.setValue(valor_actual)
    label = QLabel(f"{prefix}: {slider.value()}")

    def update_label(value):
        label.setText(f"{prefix}: {value}")
        self.actualitzar_habitacions(tipus, value) # Actualitza
        els valors al sistema

    slider.valueChanged.connect(update_label) # Connecta els canvis del
    slider amb el sistema
    return slider, label

```

## Annex II - Relació del TFG amb els ODS

*Taula 4: Relació del treball amb els ODS*

Objectius de Desenvolupament Sostenible	Alt	Mitjà	Baix	No procedeix
ODS 1. Fi de la pobresa.				X
ODS 2. Fam zero.				X
ODS 3. Salut i benestar.	X			
ODS 4. Educació de qualitat.				X
ODS 5. Igualtat de gènere.				X
ODS 6. Aigua neta i sanejament.				X
ODS 7. Energia assequible i no contaminant.			X	
ODS 8. Treball digne i creixement econòmic.				X
ODS 9. Indústria, innovació i infraestructures.		X		
ODS 10. Reducció de les desigualtats.	X			
ODS 11. Ciutats i comunitats sostenibles.				X
ODS 12. Producció i consum responsables.			X	
ODS 13. Acció pel clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida d'ecosistemes terrestres.				X
ODS 16. Pau, justícia i institucions sòlides.				X
ODS 17. Aliances per aconseguir els objectius.				X

### Grau de relació del treball amb els ODS

**ODS 3. Salut i benestar** El projecte contribueix directament a aquest objectiu, ja que proporciona una solució innovadora per a la supervisió i el benestar dels pacients en centres geriàtrics. Amb el sistema de sensors intel·ligent, es detecten anomalies com caigudes o problemes relacionats amb la salut de forma més eficient, assegurant una resposta més ràpida i adequada del personal sanitari.

**ODS 7. Energia assequible i no contaminant** El projecte es basa en l'ús de tecnologia eficient, com els microcontroladors Arduino i les Raspberry Pi, que tenen un consum energètic reduït. Aquesta optimització ajuda a minimitzar l'impacte ambiental del sistema.

**ODS 9. Indústria, innovació i infraestructures** El projecte aposta per la innovació tecnològica en el sector sanitari, introduint una solució escalable i aplicable a diferents entorns. Aquesta contribució afavoreix el desenvolupament d'infraestructures més modernes i eficients.

**ODS 10. Reducció de les desigualtats** Aquest projecte contribueix a reduir les desigualtats en l'atenció sanitària, ja que proposa una solució tecnològica assequible i adaptable per a centres geriàtrics de diferents nivells de recursos. Això permet millorar l'atenció als pacients en situacions de risc, independentment de les capacitats econòmiques dels centres on s'implementa.

**ODS 12. Producció i consum responsables** L'ús de components amb baix consum energètic i la possibilitat de reutilitzar el maquinari en diverses aplicacions contribueix a una producció més responsable.