
This is the **published version** of the bachelor thesis:

Cubells Pintado, Àlex; Crespo Yepes, Albert, dir. Sistema per al control d'aforament amb arquitectura modular distribuïda i desplegament en infraestructura al núvol. 2025. 11 pag. (Grau en Gestió de Ciutats Intel·ligents i Sostenibles)

This version is available at <https://ddd.uab.cat/record/315773>

under the terms of the  license

Sistema per al control d'aforament amb arquitectura modular distribuïda i desplegament en infraestructura al núvol

Àlex Cubells Pintado

Universitat Autònoma de Barcelona — Tutor: Albert Crespo Yepes

Grau en Ciutats Intel·ligents i Sostenibles

Facultat d'Enginyeria

20 de Juny de 2025

Resum

Aquest projecte presenta el disseny i desenvolupament d'un sistema propi per al control i la visualització de l'aforament de persones en recintes delimitats com centres comercials o botigues, mitjançant una arquitectura modular i escalable. La solució recull dades en temps real des d'un dispositiu de comptatge de persones, anomenat TAC-B 3D-W, les transmet via MQTT a un núvol d'AWS, on es processen i s'emmagatzemen en una base de dades. Les dades recollides s'integren en panells interactius mitjançant Grafana, facilitant una anàlisi exhaustiva de l'ocupació, els fluxos de visitants i els patrons d'ús de l'espai. En acabat, per evidenciar la viabilitat d'aplicar aquest sistema en contextos reals, es va presentar al congrés MEMEnginy 2025, sent tot un èxit.

Paraules clau: Comptadors de persones, sensors mmWave, IoT, MQTT, AWS, PostgreSQL, Grafana.

Resumen

Este proyecto presenta el diseño y desarrollo de un sistema propio para el control y visualización del aforo de personas en recintos delimitados como centros comerciales o tiendas, mediante una arquitectura modular y escalable. La solución recoge datos en tiempo real desde un dispositivo de conteo de personas, llamado TAC-B 3D-W, los transmite vía MQTT a una nube de AWS, donde se procesan y almacenan en una base de datos. Los datos recogidos se integran en paneles interactivos mediante Grafana, facilitando un análisis exhaustivo del empleo, los flujos de visitantes y los patrones de uso del espacio. Luego, para evidenciar la viabilidad de aplicar este sistema en contextos reales, se presentó en el congreso MEMEnginy 2025, siendo todo un éxito.

Palabras clave: Contadores de personas, sensores mmWave, IoT, MQTT, AWS, PostgreSQL, Grafana.

Abstract

This project presents the design and development of a proprietary system for the control and visualization of the capacity of people in delimited areas such as shopping centers or stores, using a modular and scalable architecture. The solution collects data in real time from a people counting device, called TAC-B 3D-W, transmits it via MQTT to an AWS cloud, where it is processed and stored in a database. The data collected is integrated into interactive panels using Grafana, facilitating an exhaustive analysis of occupancy, visitor flows and space usage patterns. Finally, to demonstrate the feasibility of applying this system in real contexts, it was presented at the MEMEnginy 2025 congress, which was a great success.

Keywords: People counters, mmWave sensors, IoT, MQTT, AWS, PostgreSQL, Grafana.



I. INTRODUCCIÓ

La gestió eficient dels espais públics i privats requereix, cada vegada més, eines intel·ligents capaces de proporcionar informació en temps real sobre el nombre de visitants i l'ocupació d'aquests entorns. En aquest context, el recompte de persones s'ha convertit en un instrument clau per a la presa de decisions en àmbits com l'administració d'esdeveniments, edificis corporatius o equipaments municipals.

Aquesta tesi presenta una solució tecnològica basada en un programari destinat a gestionar eficientment dades de recompte de persones recollides en múltiples espais i períodes temporals diferenciats. Es tracta d'un sistema dissenyat per cobrir totes les etapes del flux d'informació: des de la captura de les dades, la seva transmissió, el seu processament, l'emmagatzematge i finalment la seva visualització en temps real a través d'una interfície gràfica interactiva.

La finalitat d'aquest projecte es centra en l'optimització del procés de recopilació, processament i visualització de les dades obtingudes per mitjà d'un sistema de detecció d'esdeveniments d'entrada i de sortida de persones, anomenat **TAC-B 3D-W**.

II. CONTEXT

Aquest projecte neix a partir de les pràctiques que vaig fer a l'empresa **Quadrex SL**, que fa més de 35 anys que treballa en tecnologia per a la mobilitat, control de trànsit i estudis de trànsit en zones urbanes i interurbanes. L'empresa es dedica a integrar i mantenir sistemes per comptar i classificar vehicles i persones, així com plataformes d'analítica de vídeo per detectar i gestionar incidents en túnels i altres infraestructures.

Durant les pràctiques, es va identificar una necessitat específica no resolta: la manca d'un programari propi i flexible per a la gestió de dades d'aforament de persones. Fins al moment, Quadrex feia ús d'una eina proporcionada pel fabricant del dispositiu TAC-B 3D-W, la qual presentava diverses limitacions, com ara una capacitat analítica molt reduïda, visualitzacions gràfiques senzilles i poc explotables, i un cost derivat del model de llicència.

Amb l'objectiu d'oferir una alternativa més personalitzada, escalable i alineada amb l'arquitectura pròpia de l'empresa, es va plantejar el desenvolupament d'un *software*. Aquesta nova solució integraria l'ús dels serveis al núvol de Quadrex i eliminaria la dependència d'un programari extern tot garantint un control total sobre el flux de dades, des de la captura fins a la seva visualització final.

A més, aquest projecte ha estat una oportunitat per posar en pràctica coneixements assolits de manera complementària, com amb la certificació oficial **AWS Certified Cloud Practitioner**¹, que m'ha permès aprofundir en el desplegament de serveis en el núvol amb garanties de seguretat i eficiència.

¹Certificació bàsica d'AWS que valida coneixements generals sobre el núvol, serveis AWS i bones pràctiques de seguretat i facturació.

III. OBJECTIUS

El propòsit d'aquest projecte és desenvolupar un sistema propi i funcional que permeti gestionar de manera eficient les dades en temps real.

Per assolir aquest objectiu general, s'han plantejat els següents objectius específics:

- 1) Adaptar la comunicació del dispositiu **TAC-B 3D-W** per tal que sigui compatible amb el protocol MQTT, cosa que dona lloc a una connexió fluida amb el broker MQTT de *Mosquitto*.
- 2) Configurar el servei **AWS IoT Core** per a la recepció segura i l'encaminament de les dades cap als serveis de processament.
- 3) Desenvolupar un script que s'executi a una instància EC2 i que processi automàticament les dades rebudes.
- 4) Implementar una base de dades relacional amb **AWS RDS**, que emmagatzemi la informació de manera estructurada.
- 5) Connectar la base de dades amb **Grafana** per generar un *dashboard* interactiu amb visualitzacions gràfiques en temps real.
- 6) Dur a terme proves pràctiques i validacions del sistema per garantir el seu correcte funcionament i optimització en un entorn real.

IV. FUNDAMENTS TECNOLÒGICS

Abans d'entrar a detall és important aclarir alguns conceptes que formen part del desenvolupament d'aquest projecte:

- El *Cloud Computing* de l'anglès computació al núvol és un model de prestació de serveis informàtics sota demanda que permet l'accés remot a recursos com ara emmagatzematge, processament i programari a través d'Internet. Aquest fet elimina la necessitat de què les persones i les empreses gestionin els seus propis recursos físics i paguin només pel que facin servir. [1]
- *Amazon Web Services (AWS)* és el núvol més complet i àmpliament adoptat del món, i inclou ofertes d'infraestructura com a servei (*IaaS*²) i plataforma com a servei (*PaaS*³). [4] Els serveis d'AWS ofereixen solucions escalables per a la computació, emmagatzematge, bases de dades, anàlisi i molt més.
- El terme *Internet of Things (IoT)*, o Internet de les coses, fa referència a la xarxa col·lectiva de dispositius connectats i a la tecnologia que facilita la comunicació entre els dispositius i el núvol, així com entre els mateixos dispositius. [5] I això permet que objectes quotidians recopilin, comparteixin i actuïn sobre dades sense intervenció humana directa.

²*Infrastructure as a Service (IaaS)* és un model de computació al núvol que proporciona recursos d'infraestructura virtualitzats, com servidors, emmagatzematge i xarxes, a través d'Internet [2].

³*Platform as a Service (PaaS)* és un model de serveis al núvol que ofereix entorns de desenvolupament i desplegament complets per crear aplicacions sense haver de gestionar la infraestructura subjacent [3].

- **MQTT (Message Queuing Telemetry Transport)** és un protocol de missatgeria basat en estàndards, o un conjunt de regles, que s'utilitza per comunicar un equip a un altre. Els sensors intel·ligents, els dispositius portàtils i altres dispositius IoT generalment han de transmetre i rebre dades a través d'una xarxa amb recursos restringits i una amplada de banda limitada. Aquests dispositius IoT utilitzen MQTT per a la transmissió de dades, ja que és fàcil d'implementar i pot comunicar dades IoT de manera eficient. MQTT admet la missatgeria entre dispositius al núvol i el núvol al dispositiu. [6]

V. ARQUITECTURA DEL SISTEMA

La Figura 1, descriu breument l'arquitectura general del sistema:

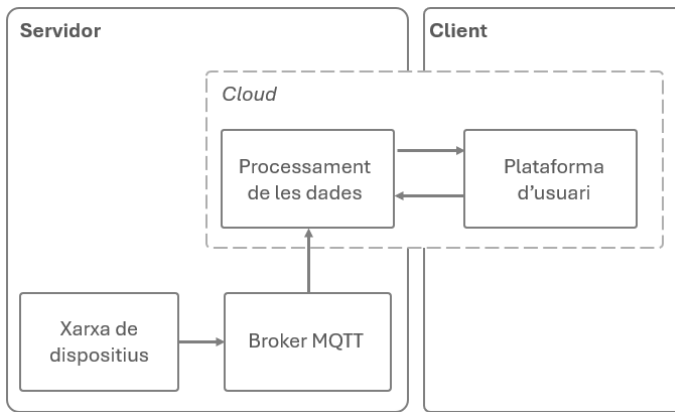


Figure 1: Esquema general del sistema.

L'arquitectura del sistema integra una xarxa de dispositius que capturen dades de cada esdeveniment d'entrada o sortida d'una persona i les transmeten, mitjançant el protocol MQTT, a un broker MQTT públic i de codi obert, que actua com a intermediari. Tot seguit, aquestes dades són reenviades, a un mòdul de processament al núvol en el nostre cas, al núvol d'Amazon Web Services, a partir d'aquest punt es verifiquen, es filtren i es guarden utilitzant serveis escalables (com AWS EC2 o AWS RDS). Finalment, la informació processada es presenta als usuaris a través d'una interfície intuïtiva, accessible des d'una plataforma web, que permetent als clients monitoritzar i observar les dades en temps real.

VI. DESENVOLUPAMENT DEL SISTEMA

Aquesta secció detalla els components tecnològics implicats i el procediment metodològic seguit durant el desenvolupament del projecte. Amb l'objectiu de proporcionar una visió global i estructurada de la solució implementada, s'ha dissenyat un diagrama d'arquitectura del sistema, on es representen els principals mòduls, la seva interconnexió i el flux de dades entre els diferents elements. Aquesta representació es pot consultar a la Figura 2.

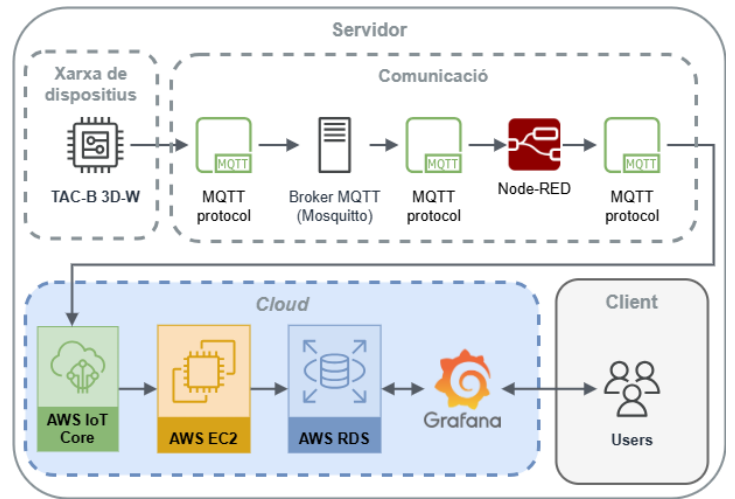


Figure 2: Arquitectura del sistema.

A. Dispositiu TAC-B 3D-W

El dispositiu **TAC-B 3D-W**, tal com es mostra a la Figura 3, ha estat desenvolupat i subministrat per **SensMax Ltd.**, una empresa europea especialitzada en solucions de recompte de persones, sensors IoT i anàlisi d'espais. Amb seu a Letònia, SensMax disposa d'una àmplia experiència en el desenvolupament de dispositius de monitoratge en temps real per al sector del comerç minorista, l'administració pública i espais oberts.

Una de les tecnologies clau emprades per aquest dispositiu és el sensor **mmWave de 60 GHz**. Aquest dispositiu utilitza tecnologia *millimeter-wave*, basada en l'emissió d'ones electromagnètiques d'alta freqüència —concretament a 60 GHz— per detectar amb gran precisió la presència, la trajectòria i la velocitat d'objectes dins del seu camp de visió [7].

És important remarcar que aquest sensor no utilitza ni ultrasons ni cap sistema òptic: no capta imatges ni vídeo, i per tant és completament anònim i totalment compatible amb la normativa *GDPR*⁴. El seu funcionament es basa en l'emissió contínua d'impulsos electromagnètics que reboten contra cossos en moviment, i en l'anàlisi dels senyals reflectits per calcular distància, direcció i velocitat. Aquest principi físic és similar al del radar, però amb una resolució més alta i capacitat per operar en condicions adverses com foscor, fum, pluja o escenes parcialment ocultes per materials semitransparents [7].

El TAC-B 3D-W presenta un camp de visió de **120° horitzontal i 60° vertical**, amb un abast màxim de **10 metres** i una resolució de detecció de fins a 5 cm per a escenes de curta distància. Pot identificar objectes que es desplacen a una velocitat de fins a 35 km/h, cosa que el fa apte per a entorns dinàmics tant interiors com exteriors [7].

⁴El Reglament General de Protecció de Dades (GDPR, per les sigles en anglès) és la normativa europea que garanteix el dret a la privacitat de les persones pel que fa al tractament de dades personals. El sensor no capta imatges ni enregistraments que puguin identificar individus, fet que assegura el seu compliment.

La configuració del dispositiu es realitza de manera senzilla a través d'una interfície web integrada, accessible via WiFi en mode *Access Point (AP)*⁵. Aquesta interfície permet configurar fins a 5 línies de comptatge i 5 zones de monitoratge, així com ajustar paràmetres avançats com la sensibilitat, el tipus d'objecte a detectar (persona, bicicleta o vehicle), la predicció de trajectòria i la discriminació d'objectes en entorns densos, evitant dobles comptatges i falsos positius [7].

A nivell d'integració, el TAC-B 3D-W ofereix connectivitat completa per a arquitectures IoT modernes. Suporta nativament el protocol **MQTT**, permet l'enviament de notifikacions mitjançant l'*API de Telegram*, i inclou funcionalitats per a l'ús de *Webhooks*⁶, fet que facilita la seva incorporació a sistemes de control i monitoratge existents [7].

Pel que fa a la instal·lació, el dispositiu pot muntar-se tant en paret com en trípode, amb una alçada recomanada entre 2,5 i 3,5 metres. S'aconsella una inclinació lleugera cap avall (entre 5 i 15 graus) per optimitzar la detecció de trajectòries llargues sense perdre precisió en objectes propers [7].

Per a una descripció més detallada de les característiques tècniques del dispositiu, es pot consultar la **fitxa tècnica** inclosa a l'**Apèndix A. Fitxa tècnica del sensor**.

En acabat, cal destacar que el desenvolupament i proves d'aquest projecte, s'ha utilitzat un dispositiu TAC-B 3D-W amb la següent identificació tècnica:

- `collector_serial: 051001471`
- `sensor_serial: 151001471Z1`



Figure 3: Dispositiu TAC-B 3D-W.

B. Format i tipus de dades MQTT del sensor

Per aquest projecte el **TAC-B 3D-W** utilitzarà el protocol **MQTT** per transmetre dades d'aforament i informació d'estat. Aquesta transmissió es realitza mitjançant una estructura de *tòpics* jeràrquics que permeten diferenciar els tipus d'informació enviada. Segons la documentació oficial del protocol de SensMax [8], que és l'empresa proveïdora de l'aparell, el dispositiu pot enviar diferents paquets amb informació diversa, entre els quals destaquen:

⁵El mode *Access Point (AP)* permet que el dispositiu actuï com a punt d'accés, creant una xarxa WiFi pròpia a la qual altres dispositius es poden connectar directament i d'aquest mode poder configurar el dispositiu.

⁶Els *webhooks* són mecanismes que permeten que una aplicació envii dades automàticament a una altra aplicació quan es produeix un esdeveniment determinat, mitjançant una petició HTTP. Són útils per integrar serveis de manera eficient i en temps real.

- **single_ts**: mostra dades generades en temps real cada vegada que es detecta un esdeveniment (una entrada o sortida).
- **5_min_data**: dades agrupades automàticament cada 5 minuts, amb informació resumida de visites i ocupació per períodes de 5 minuts.
- **inside_avg**, **inside_max**, **occupancy_begin**, **occupancy_duration**: informació estadística detallada sobre l'ocupació de zones.
- **ping** i **info**: paquets periòdics per monitorar l'estat del dispositiu.
- **notifications**: notifikacions definides per l'usuari o el sistema (per exemple, activacions de triggers o reinicis).

Per a aquest projecte s'ha optat per utilitzar el **tòpic single_ts**, ja que permet obtenir dades immediates cada cop que el dispositiu detecta una acció d'entrada o sortida. Això facilita el processament d'esdeveniments individuals en temps real i simplifica l'estructura del flux de dades, també redueix la complexitat respecte a altres formats agregats, com `5_min_data`, que requereixen tractament per lots [8].

Aquesta decisió es reflecteix a la següent secció en la configuració del node `mqtt in` de Node-RED, el qual se subscriu exactament al *tòpic* corresponent per a rebre aquest tipus de missatge. Això permet capturar únicament els missatges d'interès, reduint la càrrega de processament i garantint una resposta ràpida del sistema.

C. Broker de MQTT: Mosquitto i integració amb Node-RED

Per a la transmissió de dades entre l'aparell i la infraestructura al núvol, s'ha utilitzat el protocol **MQTT**. Aquest protocol es basa en un model de comunicació *publish/subscribe*, on els dispositius poden publicar missatges a un *tòpic* determinat o bé subscriure's per rebre'n.

En aquest projecte, el **TAC-B 3D-W** s'ha configurat per publicar els missatges en format JSON al **broker públic de Mosquitto** (test.mosquitto.org), mitjançant el port estàndard 1883 sense autenticació ni xifrat. Aquest broker actua com a intermediari lleuger i de codi obert, molt adequat per a entorns de prova i desenvolupament [9].

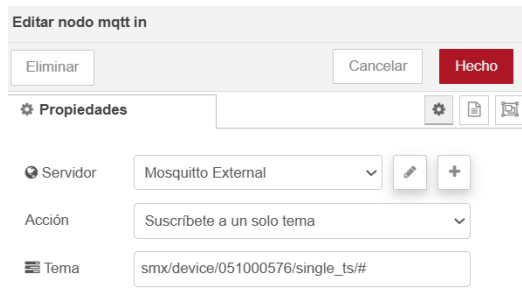
Per tal de gestionar i redirigir les dades rebudes, s'ha utilitzat **Node-RED**, una eina de programació visual basada en fluxos, que permet construir diagrames de tractament de dades mitjançant nodes o blocs configurables [10].

El flux de dades dins Node-RED es basa en dos components fonamentals:

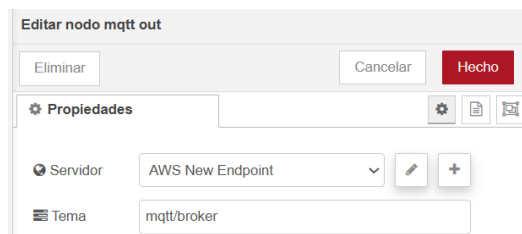
- El node `mqtt in`, tal com es mostra a la Figura 4, s'ha configurat per a **subscriure's** a un *tòpic* concret del broker MQTT Mosquitto. Per tant, aquest node rebirà tots els missatges provinents del *tòpic* especificat, que és:

Tòpic 1: Tòpic MQTT configurat per a la comunicació entre el dispositiu i Node-RED.

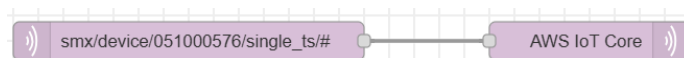
`smx/device/051000576/single_ts/#`

Figure 4: Configuració del node `mqtt in` a Node-RED.

- El node `mqtt out`, tal com es mostra a la Figura 5, publica les dades al tòpic `mqtt/broker` utilitzant el servidor definit com a `AWS New Endpoint`. Cal destacar que `AWS IoT Core` no proporciona un broker MQTT genèric, sinó que disposa d'un *client de prova MQTT* i una interfície MQTT gestionada. Aquesta interfície permet als dispositius autenticar-se, publicar i subscriure's a missatges mitjançant temes específics, amb l'objectiu de comunicar l'estat del dispositiu i rebre missatges. Per establir la connexió, és necessari carregar, al Node-RED, els **certificats de seguretat proporcionats per AWS IoT Core (certificat de client, clau privada i arxiu CA Root)**, així com l'**endpoint personalitzat associat al compte i regió d'AWS**, i utilitzar el port 8883, que permet connexions segures mitjançant TLS/SSL.

Figure 5: Configuració del node `mqtt out` per publicar a `AWS IoT Core`.

Aquesta estructura permet actuar com a pont entre el dispositiu i `AWS`, mantenint separades les responsabilitats: el dispositiu només ha de conèixer el broker Mosquitto, mentre que la connexió segura amb `AWS` es gestiona completament des de Node-RED. A continuació s'observa la Figura 6, en la que podem observar el flux de dades, mitjançant els dos nodes comentats:

Figure 6: Flux de dades entre Mosquitto, Node-RED i `AWS`.

Aquest esquema aporta flexibilitat i modularitat al sistema, facilitant tant les proves locals com la migració a entorns de producció. En conjunt, la integració entre **Mosquitto**, **Node-RED** i **AWS IoT Core** proporciona una infraestructura de comunicació lleugera, escalable i adequada per a sistemes d'aforament en temps real.

D. Amazon Web Services (AWS)

La Figura 7 mostra els serveis que formaran l'arquitectura d'AWS i les connexions entre elles.

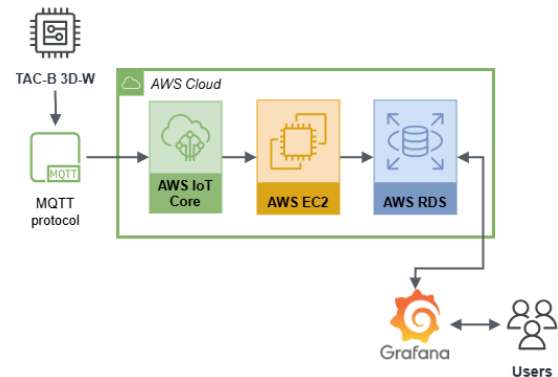


Figure 7: Arquitectura AWS.

Les dades són rebudes pel servei **AWS IoT Core**, que actua com a punt d'entrada al núvol i permet la gestió, filtratge i encaminament dels missatges. A partir d'aquest punt, es poden distingir tres fases clares dins del sistema:

- Processament:** Les dades capturades per `IoT Core` són redirigides a una instància **Amazon EC2**, on s'executa un script Python encarregat de parsejar i validar el contingut dels missatges.
- Emmagatzematge:** Un cop processats, els valors rellevants s'insereixen en una base de dades relacional allotjada a **Amazon RDS** (PostgreSQL), la qual serveix com a repositori central d'històric de dades.
- Visualització:** Finalment, la base de dades es connecta amb la plataforma de **Grafana**, que permet visualitzar les dades en temps real a través de panells interactius.

1) AWS IAM (Identity and Access Management): Per a garantir un control segur de l'accés als recursos utilitzats (`EC2`, `RDS`, `IoT Core`...), s'ha fet servir `AWS Identity and Access Management (IAM)`, el servei d'Amazon responsable de la gestió d'identitats, rols i polítiques de seguretat. `IAM` permet definir qui (identitat) pot fer què (accions) sobre quins recursos, seguint el principi del menor privilegi [11].

En aquest projecte, `IAM` s'ha utilitzat per a:

- Crear un *rol IAM* associat a la instància `EC2`, amb permisos específics per accedir a `AWS IoT Core` i a la base de dades a `AWS RDS` [12].
- Establir polítiques gestionades (*managed policies*), com `AmazonRDSFullAccess` i `AWSIoTDataAccess`, que atorguen permisos per a les funcions definides.
- Millorar la seguretat mitjançant la definició de grups i permisos amb àmbit limitat, evitant l'ús de l'usuari *root* i reduint l'exposició a riscos innecessaris.

Gràcies a IAM, s'ha assegurat que només les entitats autenticades i autoritzades puguin interactuar amb els serveis crítics (EC2, RDS, IoT Core), i amb els mínims permisos exigits [11]. Aquesta configuració contribueix també a una gestió més sòlida, i això permet aplicar canvis i revocar permisos amb facilitat sense afectar la infraestructura operativa.

2) **AWS IoT Core**: És un servei que facilita la connexió, comunicació, gestió i escalat de dispositius IoT de manera segura, senzilla i altament fiable [5]. En el marc d'aquest projecte, AWS IoT Core s'ha utilitzat com a element central per a la recepció i distribució de dades provinents del TAC-B 3D-W.

El primer pas en la implementació és la **configuració d'un dispositiu o Thing** dins d'AWS IoT Core. Aquest *Thing* és una representació virtual del dispositiu físic dins de l'entorn AWS. Això permet que el núvol reconegui el dispositiu i gestioni les seves comunicacions d'una manera segura i controlada.

Durant el procés de creació del *Thing*, es duen a terme dues accions fonamentals:

- **Generació de certificats de seguretat**: imprescindibles per garantir la comunicació segura entre el sensor i AWS. AWS permet autogenerar certificats, incloent-hi la clau pública, la clau privada i l'arxiu de certificat propi [5].
- **Creació i assignació de polítiques d'accés**: per defecte, els certificats tenen totes les operacions denegades. És necessari crear una política (*Policy*) que defineixi què pot fer el dispositiu, indicant camps com *Effect* (Allow o Deny), *Action* (p.e. *iot:Publish*) i *Resource*.

Un cop creats i descarregats els certificats i claus, aquests es carreguen al client intermediari (en aquest cas, al node `mqtt out` integrat amb Node-RED). A partir d'aquest moment, el Node-RED està capacitat per connectar-se de manera segura a AWS IoT Core i transmetre dades en temps real.

Per verificar la connexió amb el servei, **AWS IoT Core** posa a disposició un client MQTT de proves que permet subscriure's a un *topic* específic, visualitzar en temps real els missatges en format JSON rebuts pel broker i realitzar proves de publicació i subscripció.

Per tal de monitorar la informació enviada pel dispositiu, s'ha realitzat una subscripció al *tòpic* configurat en el node `mqtt out` de Node-RED, utilitzant el client MQTT de proves que proporciona AWS IoT Core. El *tòpic* utilitzat en aquest cas ha estat:

`mqtt/broker`

Tot seguit, es mostra un exemple del missatge JSON rebut a través d'aquest *tòpic*:

Missatge 3: Exemple de JSON rebut per AWS IoT Core

```
{
  "messageType": "single_ts",
  "id": "CBE13A33",
  "collector_id": "TAC-B",
  "collector_serial": "051001471",
  "collector_frm": "3.2",
  "sensor_serial": "151001471Z1",
  "sensor_name": "ACCESO 1",
  "direction": "IN",
  "date": "2025-05-06",
  "time": "12:31:05.295",
  "in_today": 23,
  "out_today": 18,
  "object_id": 12,
  "object_speed": "1.8",
  "last_5min_sum": 2,
  "first_entry_time": "11:44:55"
}
```

3) **AWS EC2 (Elastic Compute Cloud)**: Per al tractament i processament de les dades en temps real, s'ha utilitzat una instància d'**AWS EC2**, un servei que proporciona capacitat de computació escalable sota demanda [13]. En aquest projecte, la instància EC2 ha estat configurada amb un rol IAM específic, adreça IP pública i connectivitat directa amb els serveis AWS utilitzats, com IoT Core i RDS. El tipus d'instància seleccionat ha estat `t2.micro`, que és suficient per executar scripts lleugers en entorns de desenvolupament i proves [13].

Dins la instància EC2 s'executa un script desenvolupat en Python que s'encarrega de gestionar tot el flux de dades rebut des del dispositiu. Per a una millor comprensió del funcionament intern del programa executat a la instància EC2, es pot consultar el diagrama de flux inclòs a l'**Apèndix B. Diagrama de flux del codi executat a la instància EC2**, on es mostra de manera clara i esquemàtica el procés complet de connexió, recepció de missatges i inserció de dades a la base de dades.

Tot seguit, es descriuen de manera resumida les tasques principals que executa aquest programa.

Instal·lació de dependències

Prèviament a l'execució del script, ha estat necessari instal·lar les següents biblioteques:

- **AWSIoTPythonSDK**, per establir la connexió MQTT amb AWS IoT Core [14].
- **psycopg2-binary**, utilitzada per establir la connexió amb la base de dades PostgreSQL allotjada a AWS RDS i per realitzar insercions de dades. **Psycopg2** és l'adaptador més popular per a PostgreSQL en entorns Python, destacant per la seva eficiència, estabilitat i compatibilitat amb consultes SQL complexes [15].
- **boto3**, per gestionar automàticament les credencials IAM i accedir a serveis AWS.

D'aquestes biblioteques, cal destacar especialment l'ús de **boto3**, que és el *SDK (Software Development Kit)* oficial d'Amazon per al llenguatge Python. Un SDK és un conjunt d'eines que permet als desenvolupadors crear aplicacions per a un sistema o plataforma específics [16]. En aquest cas, boto3 ofereix totes les funcionalitats necessàries per interactuar de manera eficient i programàtica amb múltiples serveis d'AWS des de Python, com ara la gestió de credencials IAM, la connexió amb serveis com RDS, S3, EC2, entre d'altres [17].

Connexió segura a AWS IoT Core

La comunicació amb AWS IoT Core es duu a terme mitjançant el protocol MQTT sobre WebSocket, emprant el sistema d'autenticació **SigV4** amb credencials temporals obtingudes automàticament pel rol IAM associat a la instància EC2. Aquesta combinació permet establir connexions segures sense necessitat d'un certificat de dispositiu propi.

Tot i això, és imprescindible incloure el **certificat de l'autoritat arrel (Root CA)** proporcionat per AWS IoT Core, ja que és necessari per validar la identitat del servidor durant l'establiment de la connexió TLS. Sense aquest certificat, la connexió no seria considerada segura ni es podria establir.

Per establir correctament la connexió, cal:

- Definir l'endpoint del broker d'AWS IoT Core.
- Carregar el certificat d'autoritat arrel (*CA Root*) d'AWS.
- Obtenir les credencials temporals mitjançant la biblioteca boto3.
- El *tòpic* MQTT al qual es vol subscriure: `mqtt/broker`.

Processament i inserció de dades.

Quan arriba un missatge, s'activa una funció *callback* que analitza el contingut JSON rebut i verifica que es tracti d'un missatge de tipus `single_ts`. En aquest cas, el script extreu els valors rellevants com el número de sèrie del sensor, la direcció detectada, els comptadors acumulats, la data i hora del registre, la velocitat de l'objecte i el nom de l'accés. Tota aquesta informació s'insereix a la taula `tac_b` de la base de dades `db_sensmax`, mitjançant una consulta SQL parametritzada.

Execució contínua.

El script funciona en un bucle permanent, mantenint la connexió amb AWS IoT Core activa i escoltant nous missatges de forma indefinida, ja que roman en execució com a procés de fons dins la instància EC2.

Gràcies a aquesta automatització, s'aconsegueix una gestió eficient i robusta del flux de dades des del sensor fins a la base de dades, assegurant la persistència i la disponibilitat de la informació per a la seva posterior explotació analítica.

4) RDS (Relational Database Service): Per a l'emmagatzematge de les dades d'aforament recollides pel dispositiu s'ha utilitzat Amazon RDS (Relational Database Service), un servei gestionat que permet desplegar i operar bases de dades relacionals al núvol amb alta disponibilitat, seguretat i escalabilitat [18].

En aquest projecte s'ha optat pel motor de base de dades **PostgreSQL**, per la seva robustesa, compatibilitat amb consultes SQL estàndard i el seu bon rendiment en entorns amb volum mitjà-alt de dades. La base de dades implementada allotja una taula principal anomenada `tac_b`, que conté els registres generats per cada missatge rebut del dispositiu.

Els camps emmagatzemats a la taula inclouen informació com:

- El tipus de missatge.
- El número de sèrie del sensor.
- La direcció detectada (entrada o sortida).
- Els comptadors acumulats d'entrades i sortides.
- La data i l'hora del registre.
- La velocitat de l'objecte detectat, entre d'altres.

Un dels principals avantatges d'utilitzar **Amazon RDS** és que es tracta d'un servei de base de dades completament gestionat, cosa que permet delegar tasques administratives crítiques com ara la realització de còpies de seguretat periòdiques, l'aplicació automàtica de parches de seguretat i l'escalabilitat vertical dels recursos com la CPU, la memòria o l'emmagatzematge. A més, si es desitja, el sistema pot desplegar-se en múltiples zones de disponibilitat (Multi-AZ) per garantir una alta disponibilitat i tolerància a fallades, fet que augmenta considerablement la fiabilitat del sistema en producció [18].

A més a més, AWS RDS permet definir grups de seguretat (firewall virtual) que restringeixen l'accés a la base de dades, assegurant que només la instància EC2 autoritzada pugui connectar-s'hi.

E. Visualització de dades amb Grafana

Per a la visualització i anàlisi de les dades d'aforament en temps real, s'ha utilitzat **Grafana**, una eina de codi obert àmpliament adoptada en l'àmbit del monitoratge i la visualització de dades. Grafana permet crear *dashboards* interactius i personalitzables, mitjançant gràfics, panells i filtres, connectant-se a diferents fonts de dades com bases de dades SQL, sistemes de sèries temporals o serveis al núvol [19].

Entre les visualitzacions destacades es troba el **nivell d'ocupació actual** (Figura 8), representat amb un indicador de tipus gauge. Aquest valor es calcula com la diferència acumulada entre les entrades i sortides registrades pel sistema, de manera que:

$$\text{Ocupació actual} = \sum \text{IN} - \sum \text{OUT} \quad (1)$$

Aquest valor és actualitzat constantment i permet detectar situacions d'alta densitat o superar límits establerts. A més, es mostra l'evolució d'aquest valor al llarg del dia (Figura 8), amb un gràfic de barres que permet visualitzar els períodes de màxima afluència.

També s'ha incorporat una mètrica de **visitants totals**, calculada com la mitjana entre entrades i sortides:

$$\text{Visitants} = \frac{\text{IN} + \text{OUT}}{2} \quad (2)$$



Figure 8: Imatge en què s'observa dos gràfics, un situat a l'esquerra que representa l'ocupació actual en temps real, mentre que a la dreta es visualitza la seva evolució durant la jornada.

Aquesta fórmula s'utilitza perquè un mateix visitant pot generar una entrada i una sortida, i així s'evita el doble comptatge. Aquesta dada es representa tant de forma agregada per dia com per accés individual (Figura 9).



Figure 9: Recompte acumulat de visitants desglossat per accés i per jornada.

A nivell d'històric, es mostren gràfics que representen tant l'**ocupació mitjana diària** (Figura 10) com els **visitants totals per jornada**, durant els últims 7 dies (Figura 11). Aquestes dades s'utilitzen per fer seguiment de tendències, detectar canvis d'hàbit i optimitzar la gestió d'espais segons l'afluència real.

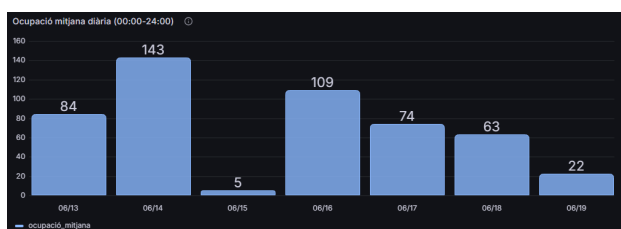


Figure 10: Ocupació mitjana diària, durant els últims 7 dies.

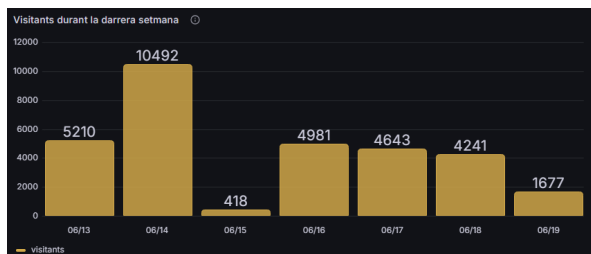


Figure 11: Visitants totals per jornada, durant els últims 7 dies.

Finalment, per monitorar en temps real l'afluència de persones a cada accés, s'ha desenvolupat un panell de visualització amb Grafana que integra múltiples gràfiques a partir de les dades emmagatzemades. A la Figura 12 es mostra un exemple corresponent a l'Accés 1. El gràfic de l'esquerra representa l'evolució del flux net, les entrades, les sortides i el nombre de visitants al llarg de la jornada. El gràfic circular de la dreta resumeix el total acumulat d'entrades i sortides.

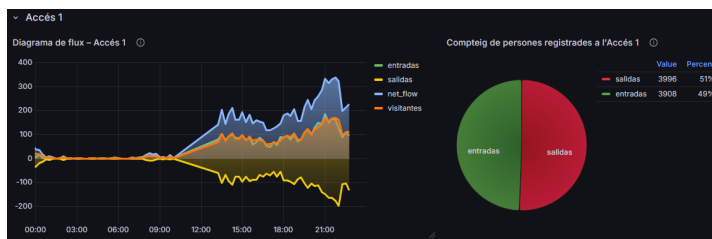


Figure 12: Visualització de flux i repartiment d'entrades i sortides a l'Accés 1. A l'esquerra, es mostra l'evolució temporal de les dades; a la dreta, la distribució percentual entre entrades i sortides acumulades.

Aquest conjunt de visualitzacions ofereix una visió completa i intuïtiva del comportament del recinte, afavorint la presa de decisions informades i la gestió eficaç de l'espai. La combinació de gràfics en temps real i mètriques històriques permet detectar patrons de comportament, comparar l'ús dels diferents accessos i anticipar situacions de saturació. Això facilita, per exemple, la redistribució del personal, la programació d'horaris de neteja o l'ajust dinàmic de l'aforament màxim segons franges horàries i dies de la setmana.

VII. PRESENTACIÓ PÚBLICA DEL PROGRAMARI A LA MEMENGINY 2025

Amb l'objectiu de donar visibilitat al projecte i apropar-lo tant a l'entorn acadèmic com professional, el programari d'aforament va ser presentat públicament durant l'edició de 2025 del congrés MEMEnginy, celebrat a la Universitat Autònoma de Barcelona.

Aquesta fira tecnològica d'àmbit universitari, va reunir més de 1.500 estudiants i 80 empreses dels sectors TIC i enginyeria [20]. L'estand de Quadrex SL va permetre mostrar el funcionament real del sistema desenvolupat, així com explicar la seva arquitectura tècnica, els avantatges operatius i les aplicacions potencials.

Al llarg de la jornada es va mostrar un *dashboard* en temps real mitjançant Grafana, el qual visualitzava el flux de persones que circulaven davant l'estand. El sistema va aconseguir registrar més de **4.000 esdeveniments** durant el dia, evidenciant així la capacitat operativa i la fiabilitat del sistema en un entorn d'alta concurrència.

Aquesta participació va representar una oportunitat clau per validar i testear el programari en condicions reals i exigents, reforçant la seva viabilitat tècnica i demostrant el seu potencial.

VIII. LIMITACIONS DEL PROGRAMARI INICIAL

Tot i que la versió del sistema d'aforament desenvolupada en aquest treball complia amb els requisits funcionals inicials, l'experiència pràctica i la seva aplicació en entorns reals van posar en evidència una sèrie de limitacions i punts crítics que han motivat la necessitat de redissenyar el programari. A continuació, es detallen les principals debilitats identificades:

- **Dependència de Node-RED:** tot i la seva facilitat, la interfície visual de Node-RED limita la flexibilitat i escalabilitat del sistema. L'estructura dels fluxos podia resultar poc eficient en escenaris amb múltiples dispositius o fluxos complexos.
- **Escassa robustesa davant errors:** la gestió d'errors en temps real era mínima. Davant de missatges mal formats o interrupcions de connexió, el sistema no disposava de mecanismes de recuperació automàtica ni de registre detallat dels problemes.
- **Dificultats en la gestió i reutilització de configuracions:** cada connexió i subscripció MQTT es feia manualment. Això volia dir que cada vegada que volies fer un desplegament, s'ha de configurar cada element del sistema dispositiu a dispositiu.
- **Complexitat d'instal·lació i desplegament:** la configuració de Node-RED, el broker Mosquitto, els certificats, les regles d'IoT Core i els scripts Python a EC2 requerien coneixements tècnics elevats, dificultant un desplegament ràpid.

Aquest conjunt de limitacions ha estat el punt de partida per a la creació d'una nova versió del programari, més centrada en la robustesa, la usabilitat i l'escalabilitat.

IX. DESENVOLUPAMENT D'UNA NOVA VERSIÓ MILLORADA DEL PROGRAMARI

Amb l'objectiu de superar les limitacions detectades a la primera versió del sistema, s'ha desenvolupat una nova versió del programari molt més avançada i escalable. Aquesta versió ha estat dissenyada amb tecnologies web modernes, utilitzant el framework **React**⁷ combinat amb **Vite**⁸, per construir un *frontend*⁹ interactiu. El *backend*¹⁰, desenvolupat en **JavaScript**¹¹, segueix una arquitectura client-servidor que garanteix una gestió eficient del flux de dades, des de la recepció en temps real fins a la seva permanència en una base de dades.

⁷React és una biblioteca de JavaScript per construir interfícies d'usuari, desenvolupada inicialment per Facebook. S'utilitza àmpliament en el desenvolupament de frontends moderns.

⁸Vite és una eina de construcció (build tool) i servidor de desenvolupament ultraràpid per a projectes web, especialment pensada per treballar amb frameworks com React, Vue o Svelte.

⁹El frontend fa referència a la part visible i interactiva d'una aplicació web, és a dir, tot allò amb què l'usuari pot interactuar directament.

¹⁰El backend és la part del sistema que opera en segon pla. Gestiona la lògica de l'aplicació, el processament de dades, les connexions a bases de dades i la comunicació amb serveis externs.

¹¹JavaScript és un llenguatge de programació interpretat, àmpliament utilitzat en el desenvolupament web tant en el frontend com en el backend (especialment amb Node.js).

La nova arquitectura del sistema es basa en una instància EC2 d'AWS que actua com a servidor principal i allotja un **broker MQTT** propi, capaç de rebre dades directament dels dispositius a través d'una **IP pública**. Aquest servidor centralitza la comunicació entre els dispositius físics i l'aplicació web, mantenint una connexió constant al broker MQTT, per a la recepció de missatges en temps real. El dispositiu TAC-B 3D-W s'hi connecta per publicar dades en *tòpics* preconfigurats, els quals són escoltats pel *backend* per al seu processament i emmagatzematge.

Configuració de brokers, bases de dades i tòpics

Una de les principals millores d'aquesta nova versió és la capacitat de configurar i gestionar múltiples brokers i bases de dades des de la mateixa interfície web. Per registrar un **broker MQTT**, s'ha d'introduir un nom, l'adreça IP o URL i el port de connexió. Aquestes configuracions poden quedar guardades de forma permanent, de manera que no cal tornar-les a introduir en iniciar l'aplicació, facilitant una connexió automàtica.

Pel que fa a la connexió a una **base de dades PostgreSQL**, s'ha de proporcionar les credencials necessàries, com la *host*, port, usuari, contrasenya i nom de la base de dades. Aquestes connexions també poden ser desades per reutilitzar-les en futures sessions sense necessitat de reconfiguració manual.

El sistema permet, a més, el registre i la gestió de **tòpics MQTT**, els quals s'associen a un broker prèviament registrat. Per afegir un nou tòpic, cal seleccionar el broker corresponent, introduir el nom del tòpic.

Assignació de destinacions i configuració genèrica

Una altra funcionalitat clau és la **configuració genèrica**, que permet definir com s'han de tractar els missatges rebuts per cada tòpic. En aquesta configuració s'ha de seleccionar el broker, el tòpic i la taula de destinació dins d'una base de dades registrada anteriorment, on s'emmagatzemaran les dades d'aforament. Aquesta lògica, totalment configurable per l'usuari, s'executa automàticament des del *backend* i assegura una correcta distribució de la informació.

Funcionalitats addicionals: alertes i extensibilitat

A banda de la configuració del sistema base, s'han implementat noves funcionalitats orientades a millorar el control i el monitoratge dels espais. Una d'aquestes és el sistema d'**alertes personalitzades**, que permet registrar una alerta vinculada a un broker i un tòpic específic. Es pot triar entre dos modes d'alerta (per nombre d'entrades o per llinar d'ocupació), definir una marca concreta i associar-hi un número de telèfon per rebre notificacions quan se superi aquest valor. Finalment, també s'ha previst la incorporació d'una funcionalitat d'**estimació de dades**, actualment en desenvolupament, que té com a objectiu inferir valors aproximats en cas de pèrdua de dades o funcionament irregular dels dispositius.

Conclusió

La nova versió del programari suposa una millora significativa respecte al sistema original. S'ha eliminat la dependència de Node-RED, substituint-la per un *backend* propi més flexible i escalable. També s'han implementat mecanismes de gestió d'errors i reconexió automàtica, augmentant la robustesa en temps real. La configuració dels brokers i tòpics ara es desa de forma centralitzada, facilitant la reutilització i el desplegament en diversos entorns. A més, el procés d'instal·lació ha estat simplificat per reduir les barreres tècniques. En conjunt, la nova arquitectura ha cobert totes les limitacions detectades anteriorment.

X. RESULTATS

Els resultats obtinguts al llarg del desenvolupament del projecte mostren que s'han assolit amb èxit tots els objectius proposats. El sistema dissenyat permet capturar esdeveniments d'entrada i sortida de persones mitjançant el TAC-B 3D-W, transmetre aquestes dades en temps real mitjançant el protocol MQTT i processar-les de manera automatitzada a través d'una instància EC2 a AWS. Aquest procés finalitza amb l'emmagatzematge estructurat de la informació en una base de dades PostgreSQL gestionada mitjançant Amazon RDS. Paral·lelament, s'ha aconseguit connectar aquesta base de dades amb Grafana per a la visualització de les dades en temps real, creant un quadre de control intuïtiu i dinàmic que permet entendre el comportament dels espais monitoritzats.

Un dels resultats més rellevants del projecte ha estat la seva validació en un entorn real durant la fira tecnològica MEMEnginy 2025. En aquesta jornada, el sistema va registrar més de 4.000 esdeveniments d'aforament sense incidències, fet que demostra la seva robustesa i fiabilitat.

XI. CONCLUSIONS

En acabat, l'execució d'aquest projecte ha donat lloc a una solució tecnològica completa i eficient per al control d'aforament en entorns reals. A partir d'una necessitat concreta detectada a l'empresa Quadrex SL, s'ha creat un sistema propi, escalable i adaptat a les seves necessitats, que elimina la dependència de programaris externs i ofereix una gestió total del flux de dades. El sistema desenvolupat cobreix tot el cicle de la informació, des de la captació fins a la seva visualització, i ha estat dissenyat tenint en compte criteris de seguretat, confiança i escalabilitat.

A més, el projecte ha evolucionat més enllà dels objectius inicials amb la creació d'una nova versió millorada del programari. Aquesta versió supera les limitacions detectades en la primera implementació, com ara la dependència de Node-RED, la manca de persistència de configuracions o la baixa tolerància a errors. La nova arquitectura incorpora un *backend* propi, desenvolupat amb tecnologies web modernes, que centralitza la gestió dels brokers, tòpics i bases de dades, i que permet configurar alertes, optimitzar el desplegament i afegir funcionalitats de manera modular. Aquesta evolució consolida el projecte com una peça potent i preparada per a ser utilitzada en entorns productius, tot aportant valor real a l'àmbit de la gestió d'espais i la monitorització d'aflluència.

Per concloure, aquest treball no sols ha arribat als objectius acadèmics que m'havia fixat, sinó que ha creat una resposta tecnològica que funciona bé i que les empreses poden utilitzar directament. A més, ha estat una experiència d'aprenentatge molt enriquidora i ha aportat alguna cosa valuosa al camp de l'Internet de les Coses.

AGRAÏMENTS

Vull començar aquests agraïments expressant el meu més profund reconeixement a l'Albert Crespo. Gràcies per la teva paciència, per confiar en mi des del primer moment i per acompanyar-me amb criteri i empatia durant tot aquest camí.

A la meua mare, no hi ha prou paraules per agrair-te tot el que has fet per mi. Gràcies per ser el meu refugi, el meu motor i el meu exemple. El teu amor incondicional, el teu suport silencios, però constant i la teua força m'han ajudat a seguir endavant en els moments més difícils. Aquest treball també és teu.

I als meus amics, els de sempre i els que han arribat pel camí, gràcies per escoltar-me quan ho necessitava i per creure en mi fins i tot quan jo dubtava. Sense vosaltres, aquest viatge hauria estat molt més solitari. Aquest projecte és el resultat de moltes hores, però també de moltes converses, abraçades i complicitats. Gràcies de tot cor.

REFERENCES

- [1] Google Cloud, "¿qué es la computación en la nube?," <https://cloud.google.com/learn/what-is-cloud-computing?hl=es>, 2025. Accés el 9 de juny de 2025.
- [2] A. W. Services, "¿qué es iaaS? - infrastructure as a service," 2025. Accés el 9 de juny de 2025.
- [3] A. W. Services, "¿qué es paas? - platform as a service," 2025. Accés el 9 de juny de 2025.
- [4] A. W. Services, "Introducción a aws," <https://aws.amazon.com/es/getting-started/>, 2025. Accedido: 8-mar-2025.
- [5] A. W. Services, "¿qué es iot?," <https://aws.amazon.com/es/what-is/iot/>, 2025. Accedido: 8-mar-2025.
- [6] A. W. Services, "¿qué es mqtt?," <https://aws.amazon.com/es/what-is/mqtt/>, 2025. Accés el 8 de març de 2025.
- [7] S. Ltd., "Tac-b 3d-w sensor datasheet," https://soft.sensmax.eu/downloads/tac-b_datasheet.pdf, 2025. Accés el 13 de juny de 2025.
- [8] S. Ltd., "Sensmax mqtt protocol (for tac-b sensor) - document version 3.7," https://soft.sensmax.eu/downloads/tac-b_datasheet.pdf, 2025. Accés el 13 de juny de 2025.
- [9] Eclipse Foundation, "Eclipse mosquitto - an open source mqtt broker," 2025. Accés el 9 de juny de 2025.
- [10] OpenJS Foundation, "Node-red: Low-code programming for event-driven applications," 2025. Accés el 4 de juny de 2025.
- [11] Amazon Web Services, "¿qué es aws identity and access management (iam)?," 2024. Accés el 9 de juny de 2025.
- [12] Amazon Web Services, "Uso de roles de iam para amazon ec2," 2024. Accés el 9 de juny de 2025.
- [13] Amazon Web Services, "¿qué es amazon ec2?," 2024. Accés el 9 de juny de 2025.
- [14] Amazon Web Services, "AWS iot device sdk for python v2 documentation," 2025. Accés el 9 de juny de 2025.
- [15] T. P. Team, "Psychopg 2 documentation," 2025. Accés el 2 de juny de 2025.
- [16] Amazon Web Services, "¿qué es un sdk?," 2023. Accés el 5 de juny de 2025.
- [17] A. Geller, "boto3: How to use any aws service with python," juliol 2021. Accés el 6 de juny de 2025.
- [18] Amazon Web Services, "Amazon RDS para PostgreSQL," <https://aws.amazon.com/es/rds/postgresql/>, 2025. Accés el 14 de maig de 2025.
- [19] G. Labs, "Grafana: The open observability platform," 2025. Accés el 7 de juny de 2025.
- [20] M. UAB, "MemEnginy — la fira tecnològica de la uab," 2025. Accés el 9 de juny de 2025.

APÈNDIX A. FITXA TÈCNICA DEL SENSOR

A continuació, es mostren les especificacions tècniques del dispositiu *TAC-B 3D-W*, extretes de la seva fitxa tècnica [7], utilitzat en el present treball.

Característica	Especificació tècnica
Carcassa	Plàstic ABS, IP54
Precisió del recompte	Per a un abast de 10 m: >99% (amb densitat d'1 persona per m ²)
Camp de visió	120° horitzontal, 60° vertical
Abast màxim	10 metres (per detecció de vianants)
Resolució de distància	5 cm per a escenes de 5 m / 10 cm per a escenes de 10 m
Velocitat màxima de l'objecte	35 km/h
Compliment amb la Llei de Privacitat (GDPR)	100% conforme (proporciona anonimat complet)
Notificacions i informes	Notificacions per Telegram, protocol MQTT (correus i informes si connectat a my.sensmax.eu)
Connexió a Internet	WiFi 2.4 GHz / WPA2-PSK / WPA2-ENTERPRISE (PEAP-MSCHAPv2)
Emmagatzematge de dades	Targeta SD de 8GB (suficient per a >100 anys de dades amb pas de 5 minuts)
Alimentació	12VDC 0,5A (consum mitjà: 4WH)
Temperatura de funcionament	-25 °C a +65 °C
Dimensions	80 x 80 x 35 mm

Figure 13: Taula resum de les especificacions tècniques del sensor.

APÈNDIX B. DIAGRAMA DE FLUX DEL CODI EXECUTAT A LA INSTÀNCIA EC2

El següent diagrama de flux representa el funcionament del script executat a la instància EC2 d'AWS. Aquest script permet la subscripció a un tòpic MQTT mitjançant AWS IoT Core i la inserció automàtica de les dades rebudes en una base de dades PostgreSQL allotjada a Amazon RDS.

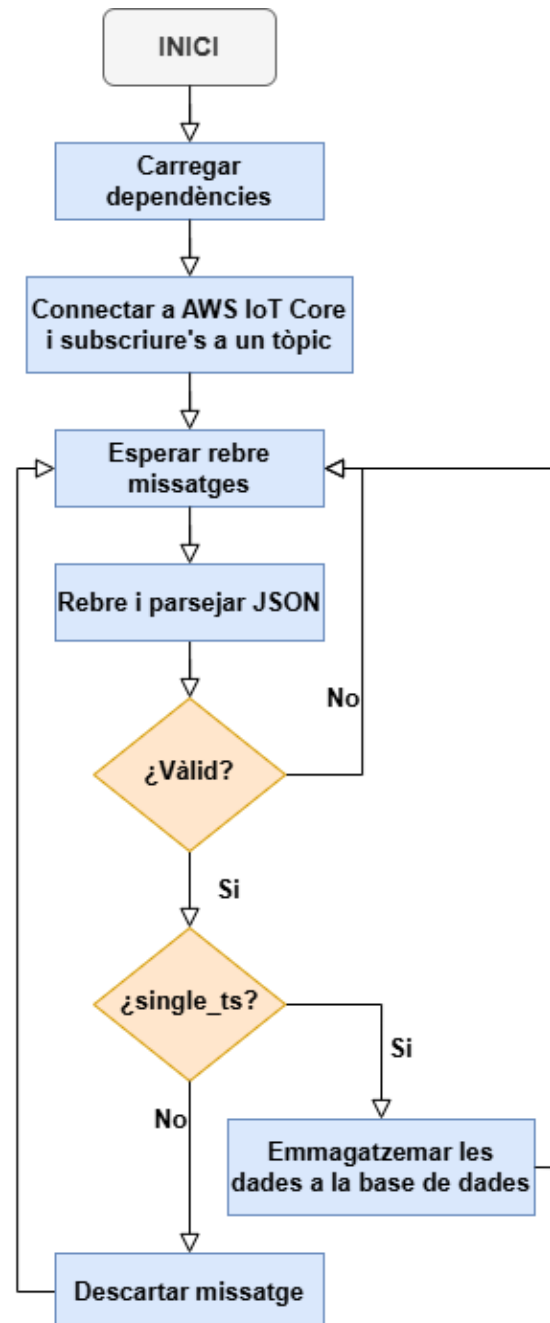


Figure 14: Diagrama de flux del script de Python que s'executa a l'EC2