UAB
Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

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

Vergés Costa, Martí. *Automatització i monitorització de la sala de fatiga d'ESPA*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Grau en Enginyeria de Dades)

This version is available at https://ddd.uab.cat/record/337351

under the terms of the  license.

Automatització i monitorització de la sala de fatiga d'ESPA

Martí Vergés

Resum– ESPA disposa d'una sala de fatiga on s'assagen bombes centrífugues durant llargs períodes per validar-ne la durabilitat. Aquest treball desenvolupa el programari del PC LAB, el node central de supervisió del laboratori. A partir del treball conjunt amb ESPA s'han consolidat els requisits que governen el disseny: lectura periòdica del variador SPEEDRIVE V2 per Modbus RTU (RS-485) i del PLC Siemens S7-1200 per Modbus TCP, unificació en un model de dades comú, validació de qualitat, històric persistent local, visualització, alarmes, exportació i consulta remota, sempre sense escriure mai sobre els equips de camp. El sistema construït s'estructura en una arquitectura de doble NIC amb bus de camp aïllat i una persistència dual SQLite→Oracle resistent a talls de xarxa, amb una consola Qt local i un dashboard web autenticat. Cada decisió no trivial s'ha resolt amb un procediment de quatre passos (requisit, opcions, criteri, decisió) i s'ha validat amb tests automàtics, simuladors intercanviables i un conjunt de dades reals de bombes centrífugues. El resultat és un sistema modular, robust i traçable de cap a peus.

Paraules clau– sala de fatiga, bombes centrífugues, requisits, metodologia d'enginyeria, Modbus, adquisició de dades, SCADA, PLC, variador de freqüència, Oracle, SQLite.

Abstract– ESPA operates a fatigue room where centrifugal pumps are tested over long periods to validate their durability. This work develops the software of the LAB PC, the laboratory's central supervision node. From ESPA's specification, the requirements driving the design were consolidated: periodic reading of the SPEEDRIVE V2 drive over Modbus RTU (RS-485) and the Siemens S7-1200 PLC over Modbus TCP, unification into a common data model, quality validation, persistent local history, visualisation, alarms, export and remote consultation, always without ever writing to field devices. The built system is structured around a dual-NIC architecture with an isolated field bus and a dual SQLite→Oracle persistence resilient to network outages, with a local Qt console and an authenticated web dashboard. Every non-trivial decision was resolved with a four-step procedure (requirement, options, criterion, decision) and validated with automated tests, interchangeable simulators and a real centrifugal-pump dataset. The result is a modular, resilient and fully traceable system.

Keywords– fatigue room, centrifugal pumps, requirements, engineering methodology, Modbus, data acquisition, SCADA, PLC, variable-frequency drive, Oracle, SQLite.

1 Introducció i context

La sala de fatiga d'ESPA sotmet bombes centrífugues a assajos de llarga durada per comprovar-ne el comportament i detectar degradacions abans de portar-les a producció. Fins ara, el seguiment d'aquests assajos depenia d'observacions manuals i no en quedava un registre estructurat i consultable de l'evolució de cada prova. L'objectiu del treball és dotar la sala d'un sistema d'adquisició i monitorització centralitzat al voltant del PC LAB, l'ordinador del laboratori.

La instal·lació inicial consta de tres grups de pressió, cadascun pilotat per un variador SPEEDRIVE V2 [3], amb bombes de 2 a 7,5 HP. El PC LAB ha de llegir de forma periòdica i fiable les variables del variador i del PLC, marcar-les temporalment, validar-les, desar-ne l'històric i posar-les a disposició tant de l'operador de camp com dels consultors remots. Un requisit transversal és la seguretat operativa: el sistema només llegeix, mai escriu sobre els equips (l'escrip-

• E-mail de contacte: marti.verges@espa.com
• Treball realitzat a ESPA
• Curs 2025/26

tura per Modbus del variador està anul·lada pel fabricant).

Vist com un problema d'enginyeria de dades, monitorar una màquina genera una sèrie temporal —una seqüència de mesures (valor, instant, identificador de variable)— que cal adquirir, validar i desar de forma fiable, amb tres propietats desitjables: ritmes d'adquisició, refresc i persistència desacoblats; una etiqueta de qualitat per cada mostra en lloc de descartar-la silenciosament; i robustesa davant talls de xarxa mitjançant un buffer local. Aquestes idees, transversals a qualsevol sistema SCADA o *data historian*, guien tot el disseny. La resta de l'informe demostra el sistema complet: requisits, metodologia de decisions, arquitectura física i de xarxa, protocols de camp, model i flux de dades, interfícies, desplegament real, pla de treball i resultats. El detall extens es desplaça als annexos.

2 REQUERIMENTS

Els requisits s'han consolidat a partir de reunions amb ESPA amb els responsables del laboratori i la sala de fatiga. Es distingeixen els funcionals (què ha de fer el sistema), els no funcionals (amb quina qualitat) i un conjunt de restriccions implícites del context corporatiu que han condicionat el disseny tant com els requisits explícits.

2.1 Requisits funcionals

- **RF-01.** Lectura periòdica (1 s nominal) de totes les variables analògiques i digitals dels variadors i del PLC.
- **RF-02.** Camps mínims per lectura: marca temporal, equip, origen, variable, valor, unitat, qualitat o estat de comunicació i codi d'alarma.
- **RF-03.** Registre històric persistent local, accessible encara que la xarxa corporativa caigui.
- **RF-04.** Visualització de valors actuals, gràfiques en viu, alarmes actives i històric en una aplicació local.
- **RF-05.** Generació d'esdeveniments per canvis d'estat, traspàs de llindars i alarmes del variador.
- **RF-06.** Exportació puntual d'un rang temporal a fitxer CSV.
- **RF-07.** Publicació d'històrics per a consulta remota des de la xarxa corporativa, sense cap accés directe als dispositius de camp.

2.2 Requisits no funcionals

Els requisits no funcionals són els que més han marcat l'arquitectura. Els més determinants:

- **Robustesa (RNF-01).** L'adquisició i el registre han de continuar tot i pèrdues de xarxa amb el PLC o amb Oracle; el servei d'adquisició no s'ha de bloquejar mai per una font caiguda.
- **Traçabilitat (RNF-02).** Cada dada porta marca temporal, origen identificat (variador, PLC o sistema) i una qualitat explícita; quan una lectura es rebaixa, s'explica per què.
- **Configurabilitat (RNF-03).** Ports, equips, llindars, retencions i credencials editables sense recompilar, en un fitxer YAML validat en arrencar.
- **Seguretat (RNF-04).** La consulta remota viu darrere d'un *reverse proxy* corporatiu que termina HTTPS, i té autenticació pròpia; cap dispositiu de camp queda exposat.
- **Cadència i resolució (RNF-07, RNF-08).** Cicle nominal d'1 s amb latència de persistència per sota de 250 ms; un cicle d'1 s aporta més de 3 600 mostres/hora per variable, dos o tres ordres de magnitud per sobre del que exigeix la dinàmica tèrmica i vibratòria d'un assaig de fatiga.
- **Capacitat (RNF-09).** El buffer local ha d'absorbir setmanes d'històric encara que Oracle estigui inaccessible tot aquest temps, sense degradar l'adquisició.

2.3 Restriccions implícites

Altres restriccions no escrites pesen tant com els requisits. Així, el PC LAB usa el seu rellotge local (sense NTP corporatiu), de manera que les mesures d'un mateix assaig es correlacionen dins el mateix host. El sistema ha de poder rearrencar sense pèrdua, fet que imposa que tota inserció sigui transaccional i el sync idempotent.

3 METODOLOGIA

El valor del projecte no rau només en el codi que finalment s'executa, sinó en per què s'ha triat cada peça i no una altra. Gairebé qualsevol problema d'enginyeria admet diverses solucions tècnicament vàlides; el que distingeix una bona decisió no és que «funcioni», sinó que sigui la que millor satisfà els requisits i les restriccions del context.

3.1 Procés de decisió en quatre passos

Cada decisió no trivial s'ha resolt amb el mateix procediment: (1) *requisit o problema* que obliga a decidir —una opció no lligada a cap requisit és una decisió prematura—; (2) *opcions candidates*, no només la triada, perquè una decisió sense alternatives documentades és indistingible d'una suposició; (3) *criteri discriminant*, sovint un de secundari (cost de llicència, dependències, operativa); i (4) *decisió i conseqüència*, l'opció triada amb el que s'hi guanya i —important— el que s'hi perd o es posposa. Aquest raonament s'ha registrat en fitxes de decisió (Requisit / Opcions / Criteri / Decisió) al llarg de la documentació; l'Annex A en recull les principals.

3.2 Jerarquia de criteris

No tots els criteris pesen igual. L'ordre aplicat per desempatar és: (1) *no perdre dades* —una hora perduda invalida setmanes d'assaig—; (2) *robustesa desatesa* —es prefereix degradar amb gràcia abans que fallar en sec—; (3) *encaix amb les restriccions corporatives* —sense privilegis d'administrador, sense serveis nadius, sense dependre de l'AD per als logins—; (4) *cost zero de llicència i de maquinari addicional*; (5) *simplicitat i mantenibilitat*; i (6) *continuïtat amb el que ja existeix*. Un criteri superior només cedeix davant d'un d'inferior si queda igualat entre les opcions. Aquesta jerarquia explica decisions que, aïllades, podrien semblar conservadores: triar Python amb les biblioteques ja usades al laboratori, no migrar a la darrera versió d'una

dependència quan l'actual cobreix el requisit, o no afegir maquinari que el plec no exigeix.

3.3 Validació de les decisions

Una decisió no es dóna per bona fins que es pot comprovar. S'han fet servir tres mecanismes, per ordre de cost creixent: (1) *tests automàtics* —cada decisió que afecta el comportament queda fixada per un test que falla si algú la desfà; uns simuladors intercanviables del variador i del PLC permeten validar tot el recorregut de la dada sense maquinari real i la suite s'executa en pocs segons—; (2) *proves de camp incrementals* —les decisions que depenen de maquinari es validen per etapes (programa sense sensors, canals físics, xarxa i Modbus, integració amb el PC LAB) sense passar a l'etapa següent fins que l'anterior està confirmada, de manera que un error queda localitzat—; i (3) *reversibilitat* —quan una decisió no es pot validar del tot per avançat, es tria l'opció més fàcil de desfer.

4 Arquitectura física i de xarxa

El sistema es vertebra en quatre dominis (camp, PC LAB, xarxa corporativa i usuaris), que la Fig. 1 representa de dalt a baix.

4.1 Capa física

Cada bomba està comandada pel seu variador SPEEDRIVE V2, que exposa variables internes (intensitat, pressió, freqüència, estats) per Modbus RTU sobre RS-485. Els sensors externs de la bomba —temperatures de rodaments DE i NDE, del motor i del fluid, vibració i rpm— no es connecten al PC: s'agrupen físicament al PLC, que els condiciona, escala i exposa com a *input registers* per Modbus TCP. El PC LAB és un Windows situat al laboratori; hi arriba l'RS-485 a través d'un adaptador USB↔RS-485 (apareix com un `COMx`) i el cable Ethernet del PLC. La regla d'or de disseny és que el PC LAB no parla mai directament amb un sensor: tota la lògica de sensors (filtratge, escalat, FFT de vibració) viu al PLC, de manera que canviar un sensor només requereix tocar el programa del PLC, no el del PC LAB.



Fig. 2: El grup de quatre bombes centrífugues del banc de fatiga, cadascuna comandada pel seu variador SPEEDRIVE V2; a la dreta, el quadre elèctric d'alimentació del banc.

4.2 Topologia de xarxa: doble NIC i bus de camp aïllat

El PC LAB ha de ser membre de dues xarxes alhora —la corporativa (Oracle, DNS, *reverse proxy*) i la de camp (PLC)— i una sola NIC no pot estar en dos segments físicament separats. Es resol amb dues targetes:

- NIC corporativa (la integrada): IP fixa `10.4.11.202` cap a la xarxa d'ESPA. El cable va directe del PC a la roseta.
- NIC de camp (adaptador USB–Ethernet): connecta directament amb el PLC màster amb un sol cable Ethernet, sense cap *switch*. És l'única connexió de camp: tots els sensors de la sala pengen del PLC com a mòduls d'ampliació Siemens (SM), no com a equips Ethernet separats, de manera que una sola connexió i una sola IP són suficients.

La regla és que el cable corporatiu no es barreja mai amb la xarxa de camp: els dos segments queden físicament separats i el Modbus TCP del PLC —que no porta autenticació— resta inaccessible des de la xarxa corporativa. La xarxa de camp és privada (RFC 1918), estàtica i sense *gateway* ni DNS: el PLC no necessita sortir-ne i, en no tenir *gateway*, no interfereix amb les rutes de la corporativa (tot el trànsit per defecte continua sortint per la NIC corporativa). El PLC màster ocupa la IP de camp real `10.4.13.58` i la NIC de camp del PC es configura dins la mateixa subxarxa, sense *gateway*. Com que els sensors s'afegeixen com a mòduls SM al mateix bastidor del PLC, no cal cap *switch*: la sala escala sense multiplicar nodes de xarxa, amb una única connexió directa. Aquesta solució no requereix cap acció d'IT (ni IP per al PLC, ni VLAN, ni canvis de *firewall*); l'únic visible des de la corporativa és que el PC té un segon adaptador actiu. El detall (pla d'adreces, material i validació) és a l'Annex C.

5 Protocols de camp: Modbus RTU i TCP

Els dos busos parlen Modbus —el mateix joc de funcions i registres de 16 bits big-endian— però per transports diferents. Tots dos són estrictament de només lectura.

5.1 SPEEDRIVE V2 (Modbus RTU sobre RS-485)

El bus físic és RS-485 a `9600 N 8 1`, mode RTU, amb esclau ID=1 fixat pel fabricant. La trama és `[ID][Funció][Dades][CRC-16]`, amb un silenci de 3,5 caràcters entre trames. El client de l'aplicació (`minimalmodbus+pyserial`) rebutja explícitament les funcions d'escriptura, perquè l'escriptura per Modbus està anul·lada de fàbrica. Els registres es numeren amb la notació `@0XXXX` (coils), `@1XXXX` (entrades digitals) i `@3XXXX` (*input registers*); el codi els converteix a l'adreça 0-based de trama i valida la funció abans de tocar el bus. Es llegeixen, per cada equip `@x` del grup (x=0 . . . 7), intensitat, pressió i freqüència del motor, més els bits d'alarma, automàtic/manual i estat del RS-485 intern; i, globals del màster, les tensions de control i de bus i el registre d'alarma `30049`, un codi enter amb 25 significats catalogats (treball en sec, sobreintensitat, sonda de pressió, etc.). El mapa complet de registres i codis d'alarma és a l'Annex B.

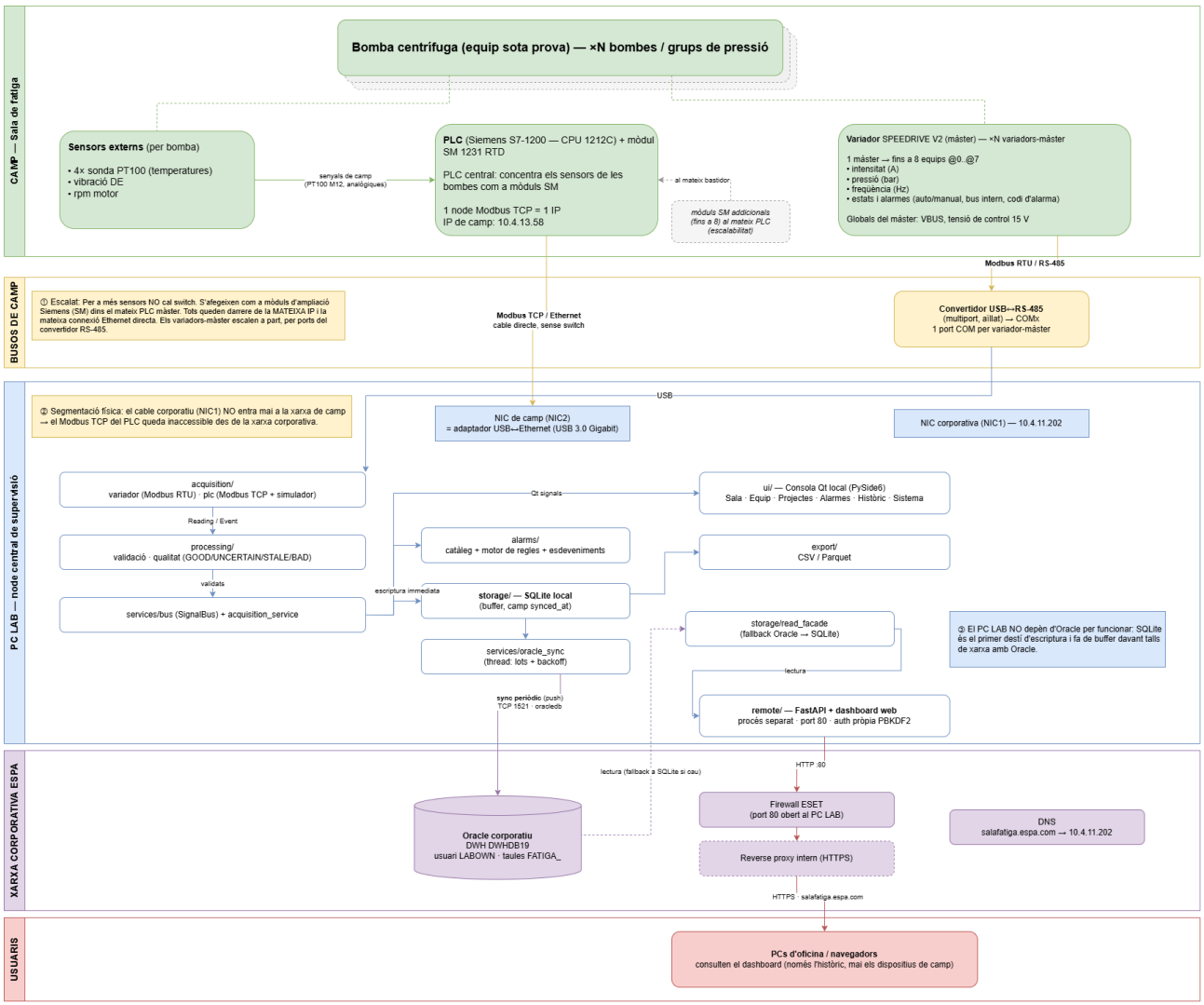

Fig. 1: Arquitectura completa del sistema: domini de camp (variador per RS-485 i PLC per Ethernet), PC LAB com a node central de supervisió (adquisició, validació, bus de senyals, persistència dual i interfícies), xarxa corporativa (Oracle, *reverse proxy*) i usuaris remots.

5.2 Diversos variadors-màster

El protocol fixa l'esclau extern a ID=1, de manera que no es poden encadenar diversos variadors-màster al mateix bus RS-485 (tots respondrien com a ID=1 i col·lidirien). El que sí encadena el SPEEDRIVE és el seu RS-485 *intern* de grup: un sol màster governa fins a 8 equips, que es llegeixen d'aquest únic màster. Per tant, el comptador de ports sèrie és el nombre de variadors-màster, no el de bombes: una bomba sola i un grup de pressió sencer ocupen, cadascun, un sol port `COMx`. L'escalat físic es fa amb un convertidor USB↔RS-485 multiport i, en programari, amb una llista `variadors:` a la configuració: a l'arrencada es construeix una font d'adquisició per element, cadascuna amb el seu `source_id`, de manera que un error de comunicació s'atribueix al bus concret. Com que l'`equip_id` és globalment únic, la resta del sistema és transparent a la topologia.

5.3 PLC Siemens S7-1200 (Modbus TCP)

El PLC és una CPU 1212C [5] amb un mòdul `SM 1231 RTD` de 4 canals [6] i 4 sondes PT100, que concentra sis senyals externs i els publica per Modbus TCP (port 502) mitjançant la instrucció `MB_SERVER` [4]. El PC LAB és el client i el PLC el servidor. Modbus TCP substitueix l'adreçat i el CRC del bus sèrie per la capçalera MBAP, perquè la pila TCP/IP ja garanteix l'entrega i la integritat; per això el codi no necessita cap temporització de bus ni cap CRC. El PLC exposa els sis valors ja processats com a *input registers* consecutius des de `30001`, cadascun amb un factor d'escala fix per no enviar decimals: quatre temperatures als canals RTD i dues entrades 0–10 V (vibració DE i rpm). A diferència del variador, aquest mapa no el dicta cap fabricant: és l'especificació que cal programar al PLC a TIA Portal. El procediment complet de preparació elèctrica, configuració i la prova que decideix entre servir per FC04/`30001` o migrar a `MB_HOLD_REG`/FC03/`40001` és a l'Annex B.

5.4 Atribució de variables a equips

El variador identifica cada bomba pel seu índex `@x`, però el PLC central reporta sis senyals «despullats» d'equip. Per repartir-los entre les bombes en assaig s'usa un mapa `plc.equip_map` (`variable_id`→equip) amb *fallback* a l'equip per defecte. Es va triar indexar per variable, i no per canal RTD físic, perquè així el mateix mecanisme co-

breix qualsevol senyal que el PLC reporti (temperatures, vibració, rpm) i no només les quatre entrades RTD.

6 MODEL I FLUX DE DADES

Un cop la dada surt dels busos de camp, deixa de ser específica de cap protocol i passa a recórrer el nucli del PC LAB sempre pel mateix camí. Aquesta secció descriu aquest recorregut de cap a peus: el model comú al qual es normalitza tota mesura, la qualitat amb què s'etiqueta, les tres freqüències desacoblades a les quals treballa el sistema i la persistència dual que el fa resistent a talls de xarxa.

6.1 Model de dades comú

Tota dada que entra al PC LAB es normalitza immediatament a un de dos tipus. Una *lectura* (Reading) és la mesura puntual d'una variable d'un equip, amb marca temporal UNIX en UTC, equip, origen, variable del catàleg, valor en unitats d'enginyeria, unitat, qualitat, valor cru opcional, codi d'estat i nota. Un *esdeveniment* (Event) representa una alarma, un canvi d'estat o una incidència de comunicació, amb tipus, severitat i un codi de convenció estable (`VFD_<n>` per al variador, `ALARM_<var>` per a llindars, `COMM_PLC...`). Fixar aquest model abans de qualsevol processament és arquitectònic: a partir del moment en què una lectura deixa la capa d'adquisició, cap consumidor ha de saber si ve d'un variador, d'un PLC o d'un simulador. Aquesta uniformitat permet substituir fonts reals per simulades als tests i afegir una nova font implementant només un contracte comú (DataSource). Els camps coincideixen amb els mínims del plec (RF-02).

6.2 Qualitat i validació

Cada mostra porta una etiqueta de qualitat —bona, incerta, antiga o dolenta— assignada per un validador amb estat que recorda l'última mostra bona de cada parella equip-variable. Les comprovacions s'apliquen en ordre: existència al catàleg, valor present i finit, rang físic, coherència de digitals, salt sobtat i antiguitat. La regla de disseny és rebaixar i no descartar: les lectures antigues o dolentes es desen igualment a l'històric perquè el rastre de l'assaig quedi complet, però no contribueixen a tendències ni a alarmes. La qualitat final és sempre la pitjor aplicada, i seqüències de lectures dolentes consecutives es poden promocionar a una incidència de comunicació.

6.3 Les tres freqüències

Un punt sovint mal entès és a quin ritme treballa el sistema; en realitat hi ha tres freqüències desacoblades (Taula 1). El ritme base d'adquisició és d'1 s: un únic temporitzador llegeix totes les bombes del grup pel mateix bucle Modbus i escriu el resultat immediatament a SQLite, dins una transacció per lot. L'empenta del buffer cap a Oracle és cada 30 s, en un fil en segon pla. I la resolució de consulta per projecte és variable i *no* és cap ritme d'adquisició: només defineix la resolució amb què es re-mostregen en memòria les dades consultades.

TAULA 1: LES TRES FREQÜÈNCIES I EL SEU PAPER.

Freqüència	Valor	Què controla
Adquisició base	1 s	Lectura Modbus de totes les bombes + escriptura a SQLite. Compartida.
Push a Oracle	30 s	Bolcat asíncron del buffer cap a Oracle, en lots, amb *backoff*.
Resolució de projecte	variable	Només presentació: re-mostreig en memòria de la consulta.

6.4 Persistència dual i consulta

La persistència aplica el criteri d'integritat. Les dades s'escriuen sempre primer al buffer SQLite local, on cada fila duu una marca de sincronització nul·la fins que s'empeny a Oracle; un servei en segon pla la bolca per lots a l'històric Oracle corporatiu —la font de veritat— amb *backoff* exponencial si Oracle cau, i purga del buffer les files ja sincronitzades més velles que la retenció. Així l'adquisició no es bloqueja mai per Oracle: escriu local i ràpid (típicament < 5 ms per lot) i empeny de forma asíncrona. El volum previst, d'unes 15 variables per bomba i fins a 8 bombes, ronda les 120 mesures/s; per això sèries i esdeveniments viuen en taules separades amb índexs per temps i per equip-variable-temps. L'esquema Oracle replica el del SQLite i es crea de forma idempotent, igual que el sync, de manera que un reinici després d'un tall no duplica ni perd cap fila. Per a la lectura, una façana prova Oracle primer i, si falla, cau automàticament al SQLite local marcant un estat degradat durant un *cooldown*, de manera que cap interfície no queda mai sense dades. En conjunt, el recorregut de cada mesura és: adquisició Modbus a 1 s $\rightarrow$ normalització $\rightarrow$ validació amb qualitat $\rightarrow$ escriptura immediata a SQLite $\rightarrow$ sincronització asíncrona a Oracle $\rightarrow$ consulta amb re-mostreig per projecte.

7 INTERFÍCIES: CONSOLA LOCAL I DASHBOARD

El sistema té dues interfícies amb rols diferents, organitzades totes dues al voltant del concepte de «projecte», que delimita una finestra temporal i un conjunt de bombes sense duplicar cap mesura.

7.1 Consola Qt del PC LAB

És la consola d'operació local (PySide6, gràfiques amb QPainter sense dependències externes). Té una capçalera operativa persistent amb el *badge* d'adquisició i sis pestanyes: *Sala* (resum de tota la sala amb estat per bomba i comptadors d'alarmes), *Equip* (detall i tendència en viu), *Projectes*, *Alarmes i events*, *Històric i exportació* (amb exportació CSV) i *Sistema* (diagnòstic per bus RS-485, PLC, Oracle i configuració efectiva). Deixa molt clar que aturar l'adquisició és diferent d'aturar una prova, i no escriu mai al camp: no modifica el mapa Modbus, no edita el YAML en calent ni administra contrasenyes.

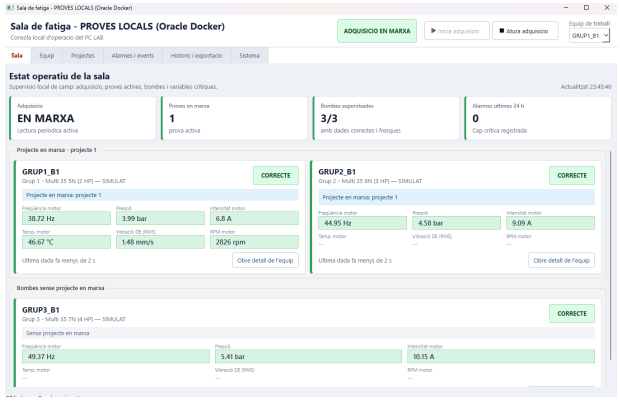

Fig. 3: Consola Qt local del PC LAB (pestanya *Sala*): capçalera operativa amb el *badge* d'adquisició i el control de proves, i estat per bomba amb valors en viu agrupats per projecte en marxa.

7.2 Dashboard web autenticat

El dashboard (FastAPI [9] + HTML/JS sense framework, en un procés separat) és per a la consulta remota des d'oficina (Fig. 4). Mostra l'estat de la sala amb les bombes agrupades per projecte, indicadors de variador/PLC/Oracle i valors en viu. L'autenticació és pròpia (PBKDF2-HMAC-SHA256 amb 600.000 iteracions segons OWASP [8], i sessió per cookie signada): és una decisió d'IT per a ús intern sense SSO O365, que permet logins compartits d'operari sense alta a l'AD. Sobre les dades d'un projecte, el botó *Genera estudi* obre una pàgina imprimible amb resum estadístic estès (mitjana, desviació, percentils, pendent i R^2 de la tendència), correlacions de Pearson entre variables i conclusions derivades per regles deterministes i auditables.

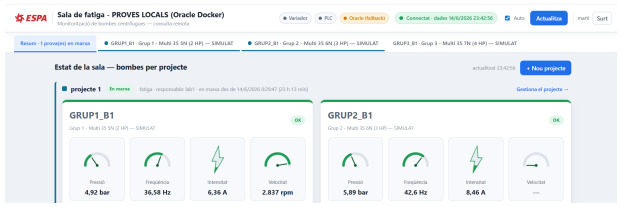

Fig. 4: Dashboard web de consulta remota: estat de la sala amb les bombes agrupades per projecte, indicadors de variador/PLC/Oracle i valors en viu de cada equip.

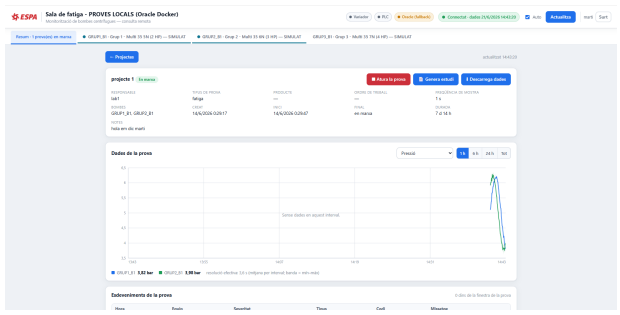

Fig. 5: Dashboard web, vista de gestió d'un projecte: fitxa de la prova en marxa (responsable, bombes, finestra temporal), gràfica de la variable seleccionada amb re-mostreig per projecte i accessos a aturar la prova, generar estudi i descarregar dades.

8 Desplegament real al PC LAB

El desplegament és una feina a dues mans amb el departament d'IT d'ESPA: IT prepara la infraestructura corporativa (base de dades, xarxa, publicació) i el projecte en gestiona el contingut i la posada en marxa al PC LAB.

8.1 Repartiment de feines amb IT

IT es va encarregar de lliurar tot el que el PC LAB necessita per arrencar contra la infraestructura real: posada a punt de l'equip amb una política d'ESET que obre el port 80 entrant (el port pel qual es serveix el dashboard); IP fixa `10.4.11.202`; accés a Oracle amb l'usuari `LABOWN` sobre la base de dades de producció `DWHDB19` (host SCAN, port 1521), amb la contrasenya en una variable d'entorn i mai al fitxer de configuració; i el subdomini `salafatiga.espa.com` resolent per DNS cap al PC. Com que `LABOWN` viu dins el *data warehouse* compartit, totes les taules del projecte es prefixen `FATIGA_` per no xocar amb el magatzem. Finalment, IT es va encarregar de muntar el *reverse proxy* intern amb HTTPS (un cop el dashboard va respondre pel port 80 directe).

8.2 Posada en marxa

La instal·lació al PC LAB és Python estàndard sobre un entorn virtual, amb la configuració en un sol YAML. Activar Oracle i el sync és posar dos indicadors a `true` i definir la contrasenya com a variable d'entorn permanent; l'autenticació es configura generant la clau de signatura i el hash de cada operari amb un *script* dedicat. L'auto-arrencada s'aconsegueix amb el *Windows Task Scheduler* natiu (sense NSSM ni serveis instal·lats): el dashboard arrenca al *boot* com a `SYSTEM` i la consola Qt al *login*. Un requisit de pes és la recuperació desatesa després d'un tall de llum: quan torna el corrent, el PC LAB ha de tornar a recollir dades sol; les tasques programades són condició necessària però no suficient, i cal completar-les amb l'arrencada automàtica de la BIOS i l'*autologon*. El procediment de desplegament complet és a l'Annex C.

9 Pla de treball

El projecte s'ha executat per etapes incrementals i validades una a una. Aquest apartat en recull el cronograma i la metodologia de treball seguida.

9.1 Cronograma per etapes

La implementació s'ha ordenat de manera que cada capa es pugui validar abans d'apilar la següent, sense dependències circulars: cap pas comença abans que l'anterior tingui els tests verds (Taula 2). Les nou etapes base van de l'esquelet (configuració, model de dades, catàleg) fins a l'històric Oracle amb sync, passant per l'emmagatzematge SQLite, l'adquisició, els simuladors, el processament i les alarmes, la interfície Qt, l'exportació i el dashboard web. Després s'hi han afegit, de forma oportunista i sense alterar l'estructura, millores post-pas-9: autenticació pròpia, auto-arrencada, indicador d'estat d'Oracle, redisseny de la consola local,

gestió de projectes de prova (pas 11), diversos variadors-màster (pas 12), atribució de variables del PLC a equips (pas 13) i l'estudi imprimible del projecte (pas 14).

TAULA 2: ETAPES DEL PLA DE TREBALL.

Pas	Contingut
1	Esquelet: config YAML, model de dades, catàleg, logs
2	Emmagatzematge SQLite (esquema + repositori)
3	Adquisició: fonts variador i PLC, bus, servei
4	Simulador del PLC autònom (Modbus TCP)
5	Processament i alarmes (validació + motor de regles)
6	Interfície Qt (sala, equip, gràfiques en viu…)
7	Exportació CSV
8	Dashboard web (FastAPI + pàgina estàtica)
9	Oracle corporatiu i sync (thread, *backoff*, façana)
11–14	Projectes, multivariador, mapa PLC–equip, estudi

9.2 Metodologia i traçabilitat

Tota la feina s'ha dut amb control de versions (Git, repositori privat amb el tutor) i una suite de tests automàtics que actua com a primer mecanisme de regressió i que ha crescut amb el sistema (135 tests al tancament del dossier). Els simuladors intercanviables del variador i del PLC han permès desenvolupar i validar tot el recorregut de la dada sense maquinari de camp disponible, i les decisions s'han documentat amb fitxes Requisit/Opcions/Criteri/Decisió que mantenen la traçabilitat entre cada requisit i la solució adoptada.

10 RESULTATS I CONCLUSIONS

S'ha implementat un sistema complet i funcional que cobreix tot el recorregut de la dada i que satisfà els requisits del plec: des de l'adquisició dels dos busos de camp fins a la consulta local i remota, passant per la validació, l'històric dual i la gestió de proves. Més enllà dels simuladors, el sistema s'ha validat amb un conjunt de dades reals de bombes centrífugues (CIRA [13]), carregat com a projecte de prova: les Fig. 6 i 7 en mostren l'evolució tèrmica i de pressió al llarg d'un assaig de gairebé deu hores, processada pel mateix pipeline de re-mostreig i estadística que faria servir un assaig real. La gràfica de temperatura del motor mostra clarament l'escalfament progressiu fins a un règim estacionari i el refredament posterior —exactament el tipus de deriva que un assaig de fatiga ha de capturar—, i l'estudi automàtic en deriva conclusions traçables (per exemple, increments de temperatura del motor del 34–47 % i correlacions fortes pressió↔vibració amb $r \geq 0{,}9$).

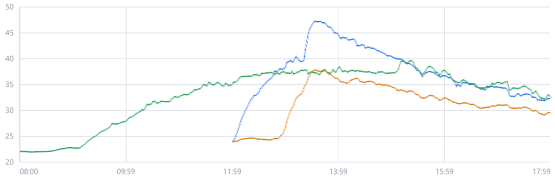

Fig. 6: Temperatura del motor de tres bombes al llarg d'un assaig real (dataset CIRA carregat com a projecte): escalfament progressiu, règim estacionari i refredament, re-mostrejat a la resolució del projecte.

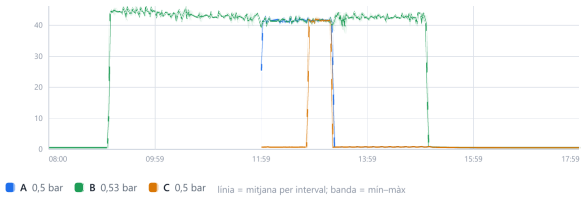

Fig. 7: Pressió de les tres bombes durant el mateix assaig: les arrencades i aturades de cada bomba queden ben delimitades pel registre continu a 1 s.

El resultat és una eina que substitueix el seguiment manual per un registre estructurat, traçable i consultable de cada assaig, respectant la premissa de no actuar mai sobre els equips i robust davant talls de xarxa gràcies al buffer SQLite local i a l'històric Oracle amb caiguda automàtica. La metodologia de decisió i validació ha permès justificar cada peça.

REFERÈNCIES

[1] Modbus Organization. *MODBUS Application Protocol Specification V1.1b3*. Modbus Organization, Inc., 2012. https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.

[2] Modbus Organization. *MODBUS over Serial Line Specification and Implementation Guide V1.02*. Modbus Organization, Inc., 2006. https://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf.

[3] ESPA. *Protocol Modbus SPEEDRIVE V2*. Document oficial del fabricant del variador (mapa de registres i codis d'alarma), s. d.

[4] Siemens AG. *Instrucció MB_SERVER per a la comunicació Modbus TCP amb SIMATIC S7-1200/S7-1500*. SIOS — Siemens Industry Online Support, ID 109759862. https://support.industry.siemens.com/cs/mdm/109759862?c=90609542795&lc=en-ES.

[5] Siemens AG. *SIMATIC S7-1200 CPU 1212C DC/DC/DC — fitxa de producte (6ES7212-1AE40-0XB0)*. SiePortal, Siemens AG. https://sieportal.siemens.com/en-ww/products-services/detail/6ES7212-1AE40-0XB0?tree=CatalogTree#overview.

[6] Siemens AG. *SIMATIC S7-1200 mòdul d'entrades analògiques SM 1231 RTD 4×AI — fitxa de producte (6ES7231-5PD32-0XB0)*. SiePortal, Siemens AG. https://sieportal.siemens.com/en-ww/products-services/detail/6ES7231-5PD32-0XB0?tree=CatalogTree#overview.

[7] International Organization for Standardization. *ISO 10816 — Mechanical vibration: evaluation of machine vibration by measurements on non-rotating parts*. ISO, Ginebra.

[8] OWASP Foundation. *OWASP Cheat Sheet Series: Password Storage Cheat Sheet*. The OWASP Foundation, 2023. https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html.

[9] Ramírez, Sebastián. *FastAPI (documentació oficial)*. https://fastapi.tiangolo.com/.

[10] The Qt Company. *Qt for Python (PySide6) — documentació*. The Qt Company. https://doc.qt.io/qtforpython-6/.

[11] Hipp, D. Richard, i el SQLite Consortium. *SQLite — documentació oficial*. https://www.sqlite.org/docs.html.

[12] Oracle Corporation. *python-oracledb — documentació oficial*. https://python-oracledb.readthedocs.io/.

[13] Martone, Angelo; Zazzaro, Gaetano. *Centrifugal Pump Dataset* (conjunt de dades), versió 1.0. Italian Aerospace Research Centre (CIRA). Zenodo, 2025. DOI 10.5281/zenodo.15301820.

[14] Verges, Martí. *ESPA — Automatització de la sala de fatiga* (codi font del projecte). Repositori Git. https://github.com/MartiVC/ESPA_automatitzacio_sala_fatiga.

ANNEXOS

Els annexos resumeixen la matèria que el cos de l'informe només enuncia. El detall exhaustiu —codi font, esquemes SQL complets i taules de registres senceres— viu al dossier tècnic del projecte, al qual es remet en cada cas.

A DECISIONS D'ENGINYERIA (FITXES REQUISIT/OPCIONS/CRITERI/DECISIÓ)

Cada fila resumeix una decisió no trivial pel procediment de quatre passos; les alternatives descartades es conserven al dossier perquè no es reobrin sense aportar res nou.

Decisió	Criteri	Resultat
SQLite buffer, Oracle veritat	No perdre dades	S'escriu sempre a SQLite i un fil empeny a Oracle amb *backoff*.
Sistema *read-only*	Robustesa + restricció del fabricant	Es prohibeix tota escriptura cap a camp; el control queda fora d'abast.
Un port sèrie per variador-màster	Col·lisió d'ID al bus (ID=1 fix)	Un bus i un `COMx` per màster, amb convertidor multiport.
Dues NIC, connexió directa al PLC	Aïllar el bus de camp (RNF-04)	Segments separats físicament; cable directe al PLC sense *switch*; xarxa de camp privada sense *gateway*.
Auth pròpia (no SSO O365)	Logins d'operari sense AD	PBKDF2 + cookie signada amb *stdlib*.
Task Scheduler natiu	Polítiques corporatives	Tasques al *boot*/*login*; cal BIOS + *autologon* per a la recuperació total.

B MAPES MODBUS DE CAMP

El variador SPEEDRIVE V2 (RTU) exposa, per equip `@x`, intensitat ($30013+2x$), pressió ($30014+2x$) i freqüència ($30029+x$) com a *input registers*, i els bits d'alarma, auto/manual i estat del RS-485 intern com a entrades digitals ($10021+3x\ldots$). El registre global `30049` és un codi enter d'alarma (no un camp de bits): 0 vol dir absència d'alarma i hi ha 25 codis catalogats, com treball en sec (33), sobreintensitat (21) o canonada rebentada (41). Els espais d'adreçament són `@0XXXX` (coils, FC01), `@1XXXX` (digitals, FC02) i `@3XXXX` (*input registers*, FC04).

El PLC Siemens S7-1200 (TCP, port 502) publica sis senyals ja processats com a *input registers* consecutius des de `30001`: quatre temperatures (rodaments DE/NDE, motor, fluid) als canals RTD PT100 amb escala 0,1, més vibració DE (escala 0,01) i rpm (escala 1,0) a dues entrades 0–10 V. Aquest mapa no el dicta cap fabricant: és l'especificació que cal programar a TIA Portal, on resta per confirmar si els valors es poden servir amb FC04/`30001` o cal migrar a FC03/`40001`. El mapa complet de registres i els 25 codis d'alarma són al dossier.

C DESPLEGAMENT: XARXA DE CAMP I POSADA EN MARXA

La xarxa de camp és privada i estàtica (sense *gateway* ni DNS): el PLC màster ocupa la IP de camp real `10.4.13.58` i la NIC de camp del PC es configura dins la mateixa subxarxa. El material és un adaptador USB–Ethernet i cable Cat 5e/6, connectat directament de l'adaptador al PLC (les NIC modernes fan auto-MDI/MDI-X) sense cap *switch*: la sala escala afegint sensors com a mòduls d'ampliació Siemens (SM) al mateix bastidor (el S7-1200 n'admet fins a vuit en sèrie), que queden darrere de la mateixa IP, de manera que mai cal multiplicar nodes de xarxa. La validació és `New-NetIPAddress` sense *gateway*, `ping` i `Test-NetConnection` al port 502; al programari només canvien tres línies de configuració (`mode/host/port`).

La posada en marxa al PC LAB és Python estàndard sobre entorn virtual: clonar el repositori, activar Oracle i el sync amb dos indicadors i la contrasenya en variable d'entorn, configurar l'auth amb `hash_password.py`, servir el dashboard pel port 80 darrere del *reverse proxy* i registrar les tasques del Task Scheduler. La recuperació desatesa després d'un tall de llum es completa amb l'arrencada automàtica per BIOS i l'*autologon*. El procediment pas a pas és al dossier.

D VALIDACIÓ AMB DADES REALS: ESTUDI DEL DATASET CIRA

Per validar tot el recorregut de la dada sense esperar el maquinari de camp, s'ha carregat com a projecte de prova un conjunt de dades reals de bombes centrífugues (CIRA [13]): tres bombes durant un assaig de gairebé deu hores. L'estudi automàtic en treu un resum estadístic estès, la matriu de correlacions de Pearson i conclusions etiquetades, amb resultats físicament coherents (la temperatura del motor amb tendència sostinguda, correlacions pressió↔vibració amb $r \geq 0{,}9$). L'exercici confirma que el pipeline és reproduïble i les conclusions traçables fins a la dada que les dispara.

E MODEL DE DADES COMÚ

Tota dada que entra al PC LAB es normalitza immediatament a un de dos tipus (`salafatiga/core/datamodel.py`): una Reading (mesura puntual d'una variable) i un Event (alarma,

canvi d'estat o incidència de comunicació). Fixar aquest contracte abans de qualsevol processament és el que permet que cap consumidor posterior hagi de saber d'on ve la dada i que afegir una font nova sigui només implementar el contracte DataSource.

TAULA 3: CAMPS DE LES ESTRUCTURES READING I EVENT.

Estructura	Camp	Contingut
Reading	`ts`	Marca temporal UNIX (s, UTC).
	`equip_id`	Equip que ha generat la mesura.
	`origin`	Origen: variador, PLC o sistema.
	`variable_id`	Variable del catàleg.
	`value`, `unit`	Valor en unitats d'enginyeria i unitat (`None` si falla).
	`quality`	Qualitat assignada pel validador.
	`raw`, `status_code`	Valor cru del registre i codi d'estat.
	`note`	Motiu de qualsevol rebaixa de qualitat.
Event	`ts`, `equip_id`, `origin`	Com a Reading (o sistema si és global).
	`type`	Tipus d'esdeveniment (EventType).
	`severity`	Severitat (`INFO`…`CRITICAL`).
	`code`	Codi estable: `VFD_<n>`, `ALARM_<var>`, `COMM_PLC`…
	`message`	Text llegible per a l'operari.
	`variable_id`, `value`	Variable i valor associats (opcionals).

La propietat `is_usable` de la Reading marca què compta per a tendències i alarmes (valor present i qualitat `GOOD`/`UNCERTAIN`); les lectures `STALE` o `BAD` es desen igualment perquè el rastre quedi complet. El camp `code` de l'esdeveniment segueix una convenció estable perquè l'històric pugui filtrar i agrupar.

Quatre enumeracions donen suport al model: Origin (`VARIADOR`/`PLC`/`SISTEMA`), Quality (`GOOD`<`UNCERTAIN`<`STALE`<`BAD`, on el validador només rebaixa), EventType (`ALARM_SET/CLEAR`, `STATE_CHANGE`, `COMM_LOST/RESTORED`…) i Severity (`INFO` a `CRITICAL`, com a `IntEnum` ordenable).

E.1 Catàleg de variables

Cada variable té una definició immutable (VariableDef) amb identificador estable, origen, naturalesa, unitat, factors d'escala/offset i llindars opcionals; la conversió de cru a enginyeria és lineal ($\text{valor} = \text{raw} \cdot \text{scale} + \text{offset}$). El catàleg és la font de veritat única per a validació, alarmes i etiquetes, i afegir-hi una variable no demana tocar cap altre mòdul. Cobreix tres variables analògiques i quatre digitals del variador i fins a set senyals del PLC; el maquinari adquirit (un `SM 1231 RTD` de 4 canals més 2 entrades de CPU) en llegeix sis. Els llindars són inicials —referenciats a la ISO 10816 [7] per a vibració i a valors conservadors per a temperatures— i s'afinaran amb dades reals. El catàleg complet és al dossier.

Catàleg fix, equips configurables. El catàleg és únic i viu al codi; el que es configura per instal·lació són els equips (bombes), no les variables, perquè el parc és homogeni (tot SPEEDRIVE V2) i totes les bombes comparteixen el mateix mapa. Si el parc deixés de ser homogeni, fer-lo configurable és una extensió ja prevista.

F VALIDACIÓ, QUALITAT I ALARMES

Cada lectura passa pel ReadingValidator abans del servei d'adquisició. És *stateful* (recorda l'última lectura bona de cada parella equip-variable per calcular salts) i aplica sis comprovacions en ordre —existència al catàleg, valor present i finit, rang físic, coherència digital, salt sobtat i antiguitat—, amb qualitat final la pitjor aplicada (`GOOD`<`UNCERTAIN`<`STALE`<`BAD`). La regla de disseny és rebaixar i no descartar: un `BAD` amb nota «Salt sobtat (62 ¿ 50)» és informació útil de diagnòstic, i seqüències de `BAD` consecutives es poden promocionar a una incidència `COMM_LOST`.

El motor de regles (AlarmEngine) processa cada lectura validada i emet esdeveniments quan l'estat d'alarma canvia respecte de l'estable previ. Té dos camins: l'alarma del variador (codi del registre 30049), sense antirebot perquè el variador ja és la font autoritativa; i el llindar per variable analògica (`warn`/`alarm` min/max), amb antirebot configurable (`debounce_count`) perquè una oscil·lació al voltant del llindar no generi una pluja d'esdeveniments. El registre 30049 és un codi enter amb 25 alarmes catalogades —treball en sec (33), sobreintensitat (21), temperatura del motor excessiva (34), canonada rebentada (41)…—, cadascuna publicada com a AlarmDefinition amb identificador `VFD_<n>`. La taula sencera de codis és al dossier. Quan el variador rebutja una sol·licitud retorna el bit `0x80` i un codi d'excepció Modbus, que el client tradueix a un missatge llegible i, si és persistent, a un `COMM_LOST`.

G ESQUEMA D'EMMAGATZEMATGE: SQLITE I ORACLE

L'històric local viu en un SQLite amb dues taules de sèrie (`measurements`, `events`) i dues de catàleg (`devices`, `variables`) que repliquen el catàleg de Python perquè una consulta SQL directa resolgui noms i unitats sense sortir

de la base de dades. La clau del patró buffer+sync és la columna `synced_at`: `NULL` mentre la fila està pendent i amb marca temporal un cop confirmada a Oracle (i, per tant, purgable passada la retenció). Els índexs cobreixen les consultes previstes —per temps, per (equip, variable, temps) i per (variable, temps), més un per `code` als esdeveniments— i no hi ha `FOREIGN KEY` contra les sèries per no penalitzar la inserció a 1 Hz. Tot l'accés passa per un únic StorageRepository que insereix per lots dins una transacció (si una fila falla, es desfà el lot sencer).

L'esquema Oracle replica el del SQLite —mateixos noms de columna i d'índex— amb tipus natius i execució idempotent: cada `CREATE` s'embolica en un bloc PL/SQL que captura l'error `ORA-00955` (≪objecte ja existeix≫) i el silencia, de manera que `auto_create_schema` es pot executar tantes vegades com calgui. Al desplegament real, totes les taules reben el prefix `FATIGA_` perquè l'usuari `LABOWN` viu en un *data warehouse* compartit i no poden col·lidir amb les existents. El repositori Oracle (OracleRepository) ofereix la mateixa API de lectura que el local i escriu per lots amb `executemany`; la connexió usa `python-oracledb` en mode *thin* —sense Oracle Client natiu, alineat amb la restricció de no instal·lar binaris— i aïlla qualsevol error en una excepció pròpia OracleUnavailable. Els esquemes DDL complets (SQLite i Oracle) són al dossier.

H PERSISTÈNCIA DUAL: SINCRONITZACIÓ I FAÇANA DE LECTURA

El OracleSyncService corre en un fil en segon pla i empeny el buffer cap a Oracle sense bloquejar mai l'adquisició. Cada `push_period_s` (30 s) llegeix un lot de mesures pendents (`synced_at IS NULL`), les insereix a Oracle i marca les files locals com a sincronitzades, i fa el mateix amb els esdeveniments; si el lot surt ple continua de seguida amb el següent, per buidar ràpid el buffer després d'una caiguda llarga. Periòdicament purga del SQLite les files ja sincronitzades més antigues que la retenció, sense tocar mai les pendents. Si Oracle no respon (OracleUnavailable), aplica *backoff* exponencial (es duplica el període fins a `backoff_max_s`, 300 s) i el restableix en recuperar-se. El seu estat (`last_push_ok`, `last_error`, comptadors) s'exposa com a atributs públics que la consola Qt llegeix cada 5 s per pintar el *badge* ≪Oracle (històric)≫.

Tota lectura del dashboard passa per la ReadFacade, que prova Oracle primer i, si falla, cau automàticament al SQLite local. Per no pagar la latència del *timeout* en peticions consecutives durant una caiguda, aplica un *cooldown* (30 s): si Oracle ha fallat fa poc, va directament al local. Exposa un `mode` (`remote`/`degraded`/`local`) que es propaga al `/api/health` i al *chip* visual de la interfície. El recorregut complet de cada mesura queda, doncs: adquisició Modbus a 1 s → normalització → validació amb qualitat → escriptura immediata a SQLite ($<$ 5 ms per lot) → sincronització asíncrona a Oracle amb *backoff* → consulta amb re-mostreig per projecte i caiguda automàtica al buffer si Oracle no respon.

I AUTENTICACIÓ PRÒPIA DEL DASHBOARD

El dashboard té autenticació pròpia per usuari i contrasenya (`salafatiga/remote/auth.py`), una decisió d'IT després d'avaluar l'SSO O365: el laboratori vol logins compartits d'operari sense alta a l'Active Directory, i integrar el proxy amb O365 no es justifica per a un grup petit en una eina interna. Només depèn de la *stdlib* de Python (`hashlib`, `hmac`, `secrets`) més `itsdangerous` —*pure-Python* i ja transitiu de FastAPI— per signar la cookie, sense afegir cap binari natiu. La taula d'usuaris és una llista plana al YAML: el sistema té autenticació però no autorització, distingeix qui ets (per a la traçabilitat al camp `owner`), no què pots fer.

Cada contrasenya es desa com a `pbkdf2_sha256$<iter>$<salt>$<hash>`, amb PBKDF2-HMAC-SHA256 a 600.000 iteracions (OWASP [8]), salt aleatori de 16 bytes i comparació amb `hmac.compare_digest`; el nombre d'iteracions queda dins el hash per poder pujar-lo sense invalidar els antics. La verificació no llança mai excepció (un hash mal format equival a contrasenya equivocada) i la contrasenya en clar no es desa mai enlloc.

La sessió es manté amb el SessionMiddleware de Starlette, que signa la cookie amb `itsdangerous`; res es persisteix al servidor. La `secret_key` es llegeix d'una variable d'entorn i el loader en rebutja les de menys de 32 caràcters; la cookie és `same_site=lax` i `Secure` darrere TLS, amb durada de 8 h, i rotar-la invalida totes les sessions de cop. Dues mitigacions complementen el disseny: contra l'*open redirect*, el paràmetre `next` només s'accepta si és una ruta interna; contra la *user enumeration* per timing, davant un usuari inexistent s'executa igualment un PBKDF2 contra un hash *dummy*. El control d'accés es concentra en una sola dependència FastAPI (`get_current_user`); amb l'auth desactivada retorna un usuari `"anonymous"`, i `GET /api/health` no demana mai sessió perquè el proxy i el Task Scheduler l'usen com a *health check*.

No hi ha pantalla d'administració: les operacions es fan editant el `config.yaml` i reiniciant, amb l'script `scripts/hash_password.py` que genera la clau i els hashes.

TAULA 4: OPERACIONS DE GESTIÓ D'USUARIS DEL DASHBOARD.

Operació	Procediment
Alta / canvi de contrasenya	Generar el hash i afegir o substituir l'entrada a `auth.users`; reiniciar.
Baixa d'un usuari	Esborrar l'entrada i reiniciar; `get_current_user` buida la sessió i respon 401 (revocació individual).
Tancar totes les sessions	Rotar la `secret_key`: invalida totes les cookies anteriors.
Caducitat	Automàtica passat `session_max_age_s` (8 h).

Renúncies conscients: sense rols ni permisos, sense recuperació de contrasenya, recàrrega en fred i sense *rate limiting* a l'aplicació (el cost de PBKDF2 ja encareix la força bruta; endurir-ho aniria al reverse proxy). Si l'eina creixés, la migració natural seria una taula d'usuaris a base de dades i, possiblement, SSO O365 al proxy.

J API REST I FORMAT DE DADES

L'API FastAPI (`salafatiga/remote/api.py`) consulta sempre l'històric a través de la ReadFacade; cap endpoint parla amb dispositius de camp. Excepte `/api/health`, `/login` i els estàtics, totes les rutes exigeixen sessió vàlida quan l'auth està activa.

TAULA 5: ENDPOINTS HTTP EXPOSATS PEL DASHBOARD.

Mètode	Ruta	Descripció
GET	`/`	Dashboard HTML; redirigeix a `/login` si auth activa i sense sessió.
GET	`/static/*`	Recursos estàtics (`app.js`, `styles.css`, `login.html`).
GET/POST	`/login`	Pàgina de login i validació de credencials (estableix la cookie signada).
POST	`/logout`	Buida la sessió.
GET	`/api/health`	Estat global: instal·lació, equips, `db_path`, `storage_mode`, `storage_error`, `ts`. Públic.
GET	`/api/me`	Usuari autenticat actual (o ≪anonymous≫).
GET	`/api/variables`	Catàleg complet de variables.
GET	`/api/latest`	Últimes lectures, filtrables per `equip_id` i `variable_id`.
GET	`/api/measurements`	Mesures amb filtres (`ts_from/to`, `equip_id`, `variable_id`, `origin`, `quality`, `limit`, `newest_first`).
GET	`/api/events`	Esdeveniments amb filtres (`ts_from/to`, `equip_id`, `code`, `limit`, `newest_first`).

El mateix objecte de filtre (MeasurementFilter) s'usa a la UI local i a l'API, garantint que la consulta serialitzada per HTTP sigui idèntica a la de la consola. Les lectures i esdeveniments es serialitzen a JSON pla, amb el nom de la variable resolt des del catàleg. Quan `web.behind_proxy` és cert, el servidor arrenca amb `proxy_headers=True` perquè confiï en les capçaleres `X-Forwarded-*` del reverse proxy corporatiu, que termina HTTPS amb certificat intern. L'autenticació, però, no la fa el proxy sinó la pròpia app, de manera que els hashes i la clau de sessió viuen al PC LAB i el proxy només actua com a terminador TLS i validador d'origen.

K ESTUDI IMPRIMIBLE DEL PROJECTE

Un cop una prova té dades, el detall del projecte ofereix el botó *Genera estudi*, que obre una pàgina HTML autònoma per imprimir o desar com a PDF des del navegador (regles `@media print` per a A4). Es va triar HTML imprimible i no un PDF generat al servidor per no afegir cap dependència pesada: l'estudi reaprofita el motor de gràfiques SVG, l'autenticació i la paleta del dashboard. L'endpoint `GET /api/projects/{id}/study` separa el càlcul (determinista) de la interpretació (regles aïllades) i és reproduïble: dos estudis del mateix projecte donen el mateix resultat.

El resum estadístic per sèrie inclou posició i dispersió (`avg`/`min`/`max`, `std`, coeficient de variació i percentils 5/95), variació relativa (`delta_pct`) i tendència (`slope_per_h` i `trend_r2`). La tendència s'obté per mínims quadrats del valor mitjà de cada interval contra el seu instant en hores: el pendent diu quant deriva la variable cada hora i l'R^2 si la deriva és real, de manera que es detecta una deriva lenta que el `delta_pct`, mirant només els extrems, pot amagar. `correlate_variables()` quantifica l'acoblament físic amb el coeficient de Pearson entre cada parell de variables analògiques d'una mateixa bomba, i reporta com a relacions detectades els parells amb $|r| \geq 0{,}70$. Les fórmules completes són al dossier.

`build_conclusions()` és una funció pura: cada frase es dispara per una regla concreta i porta a dins el valor i el llindar que l'han activada, sempre traçable fins a la dada.

TAULA 6: REGLES QUE GENEREN LES CONCLUSIONS DE L'ESTUDI.

Regla	Què la dispara
Durada de la prova	`effective_s` vs. `planned_s`: completa si $\geq 99\,\%$, si no avís de parada anticipada.
Esdeveniments d'alarma	Recompte d'events de severitat `ALARMA` dins la finestra.
Superació de llindar	El pic de la sèrie iguala o supera el `warn_max`/`alarm_max` del catàleg.
Deriva	$\lvert$`delta_pct`$\rvert \geq 10\,\%$ (avís) o $\geq 25\,\%$ (alarma).
Tendència sostinguda	`trend_r2`$\geq 0{,}5$ i variació projectada $\geq 10\,\%$.
Desacoblament físic	Una relació esperada entre dues variables no apareix (o hi apareix amb signe contrari).

La regla de desacoblament és la més diagnòstica: si dues variables que físicament s'han d'acoblar no confirmen la correlació esperada, ho marca com a possible fuita, cavitació o desgast. D'acord amb el tutor, les conclusions surten primer de regles deterministes perquè un informe de laboratori ha de ser reproduïble i auditable; la porta per a un paràgraf interpretatiu redactat per un model de llenguatge ja és oberta i marcada al codi, amb les regles actuals com a *fallback*.