Desenvolvimento de Soluções de Monitorização e Automação na CapTemp

Relatório de estágio curricular – CapTemp

Mestrado em Engenharia Informática – Computação Móvel

Juan Carlos Calvopiña Lalangui

Leiria, abril de 2026

Desenvolvimento de Soluções de Monitorização e Automação na CapTemp

Relatório de estágio curricular – CapTemp

Mestrado em Engenharia Informática – Computação Móvel

Juan Carlos Calvopiña Lalangui

Estágio realizado sob a orientação da Professora Doutora Dulce Cristina dos Santos Iria Gonçalves e sob supervisão do Engenheiro João Paulo López Agostinho (supervisor na entidade de acolhimento de estágio).

Leiria, abril de 2026

Originalidade e Direitos de Autor

O presente relatório de estágio é original, elaborado unicamente para este fim, tendo sido devidamente citados todos os autores cujos estudos e publicações contribuíram para o elaborar.

Reproduções parciais deste documento serão autorizadas na condição de que seja mencionado o Autor e feita referência ao ciclo de estudos no âmbito do qual o mesmo foi realizado, a saber, Mestrado em Engenharia Informática – Computação Móvel no ano letivo 2024/2025 da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, Portugal, e, bem assim, à data das provas públicas que visaram a avaliação destes trabalhos.

Dedicatória

Dedico este trabalho à minha família, berço de tudo quanto sou e inspiração do que desejo construir. Aos meus pais, pela coragem quotidiana de acreditar, pelos conselhos que chegavam no tempo certo e pela ternura firme que me ensinou a transformar esforço em caminho. Obrigado pelos sacrifícios discretos, por cada manhã em que me lembraram que estudar também é honra e responsabilidade, e por cada noite em que uma chamada breve bastou para me devolver a confiança. À família alargada, próxima e distante, que me acolheu em cada regresso e me enviou força em cada partida, agradeço o cuidado que não se mede em quilómetros.

Às minhas duas irmãs, pela presença serena, pelo humor que desarma as tensões e por celebrarem comigo as pequenas vitórias que, somadas, fizeram possível esta etapa.

E, de modo muito especial, ao meu irmão, colega do mestrado: obrigado por teres partilhado cadernos e silêncio, dúvidas e descobertas, cansaços e convicções. Pelas sessões de estudo que começaram tarde e acabaram ainda mais tarde; pelos resumos enviados à última hora; pelas correções pacientes de capítulos e pelos *debugs* que só avançaram porque insististe em mais uma hipótese. Obrigado por estares presente nos dias de maior pressão, por lembrares o essencial quando os prazos se impunham, e por me mostrares, com o teu exemplo, que a disciplina pode ser também cuidado. Se este percurso teve solidez, foi porque caminhámos lado a lado — nos corredores, nas bibliotecas e nas madrugadas de código e escrita. Esta conquista tem a tua mão e a tua voz.

Dedico igualmente a Deus, fonte de vida e de sentido, pela saúde, pela perseverança e pela serenidade nos dias mais longos. Se o estudo é um ato individual, o seu percurso é profundamente coletivo: nasce do apoio, cresce no diálogo e cumpre-se no serviço. Este relatório é, por isso, um gesto de gratidão transformado em páginas. À minha família, com amor e respeito: esta conquista é nossa e seguirá a ser compromisso — continuar a aprender, a cuidar e a construir, juntos.

.

Agradecimentos

Agradeço à CapTemp, Lda**.** a oportunidade de realizar este estágio e a confiança depositada na minha capacidade de aprender e contribuir. Trabalhar com sistemas reais, prazos exigentes e equipas multidisciplinares permitiu-me amadurecer técnica e humanamente.

Ao Eng. João Paulo Agostinho, meu supervisor na entidade de acolhimento, expresso reconhecimento pelo acompanhamento próximo, pela clareza nas decisões e pelo rigor prático. As revisões de código, as conversas sobre arquitetura e as perguntas difíceis foram, ao mesmo tempo, desafio e escola.

À Professora Doutora Dulce Cristina dos Santos Iria Gonçalves, minha orientadora académica, agradeço o olhar científico, o sentido de exigência e a generosidade do *feedback*, que me ajudaram a articular prática, escrita e reflexão crítica.

A toda a equipa da CapTemp — desenvolvimento, operações, laboratório e apoio ao cliente — obrigado pela partilha generosa de conhecimento e pelo espírito colaborativo. Aos colegas de escritório, pelos momentos de *pair programming*, e pelo humor que manteve a equipa coesa. Agradeço também às equipas que facilitaram acessos, documentação e *APIs*, e às pessoas que disponibilizaram tempo para demonstrações e validações em contexto real.

Aos colegas do mestrado (edição 2024/2025), obrigado pela entreajuda, pelos debates que abriram novas perspetivas e pela disciplina de quem se responsabiliza pelo trabalho comum. Aos amigos e familiares, pela logística invisível, pelas mensagens nos dias difíceis e pelas celebrações simples que devolveram equilíbrio.

Por fim, deixo um agradecimento expresso ao Instituto Politécnico de Leiria que acompanhou este percurso e a todos os professores que fizeram parte da minha formação académica. A vossa exigência, a generosidade no ensino e o exemplo profissional foram determinantes para consolidar conhecimentos, cultivar sentido crítico e orientar escolhas. Sem o vosso trabalho e dedicação, este relatório e o estágio que lhe deu origem não teriam a mesma solidez. Muito obrigado.

Resumo

Este relatório apresenta o estágio curricular do Mestrado em Engenharia Informática – Computação Móvel, realizado na CapTemp, Lda. (Pombal, Portugal), entre setembro de 2024 e junho de 2025, com duração de nove meses. O trabalho surgiu da necessidade de modernizar o ecossistema *Senslive* e de aumentar a eficiência dos processos técnicos e administrativos, garantindo simultaneamente a rastreabilidade e a qualidade do serviço prestado aos clientes.

Antes da introdução do ecossistema *Senslive*, a solução disponibilizada pela empresa assentava numa infraestrutura local - a cada cliente era fornecido um computador dedicado, responsável por armazenar os dados recolhidos pelos dispositivos de monitorização, bem como por efetuar cópias de segurança para mitigar o risco de perda de informação. Embora esta abordagem permitisse guardar historicamente os dados, implicava uma gestão distribuída, maior complexidade de manutenção e uma escalabilidade limitada.

Com o crescimento das tecnologias de computação em nuvem e a necessidade de centralizar a informação, a CapTemp optou por evoluir para um serviço *cloud*, materializado no ecossistema *Senslive*. Esta plataforma transfere o armazenamento e o processamento de dados para a nuvem, proporcionando uma solução centralizada, acessível a partir de múltiplos dispositivos e mais fácil de manter e escalar. Além de representar uma evolução tecnológica em relação ao sistema anterior, esta migração melhora de forma significativa a experiência dos utilizadores, ao disponibilizar *dashboards* personalizados, consulta e *download* de relatórios e funcionalidades de análise dos dados recolhidos.

Neste contexto, o projeto desenvolveu diversas soluções integradas orientadas à melhoria do ecossistema *Senslive* e ao suporte das operações diárias da empresa. Uma dessas soluções é o projeto Chat/Notificações em tempo real, que disponibiliza um canal conversacional com resposta imediata a pedidos sobre registos e estado de sensores. A solução combina *WebSocket* para comunicação interativa e notificações push (APNs/FCM) para entrega fiável em segundo plano, assegurando baixo tempo de resposta e continuidade do serviço.

A aplicação iLog (Android + *cloud*) permite programar e ler *dataloggers* DS1921G e sincronizar automaticamente as missões e leituras com o *Senslive*, focando-se na operação

em mobilidade e na centralização dos dados para análise, geração de gráficos e emissão de documentação. A aplicação iLog PRO (Android + exportação local) fornece um fluxo *offline-first* para exportação em formato *.xlsx* e elaboração de relatórios com gráfico de linhas e realce visual de excedências, sendo especialmente útil em contextos em que o envio imediato para a cloud não é adequado.

Por fim, a plataforma interna CTFlash/CTFirmware automatiza a programação de *firmware* em massa, disponibiliza um módulo de leitura para triagem de sondas, integra a criação de pedidos de laboratório e suporta a impressão de etiquetas através de API, reduzindo tarefas manuais e reforçando a rastreabilidade dos equipamentos e dos processos.

Palavras-chave: Ecossistema *Senslive*; plataforma *cloud*; *WebSockets* e notificações *push* (APNs/FCM); Android/Kotlin; *dataloggers* DS1921G; automatização de *firmware*; exportação para Excel, API para impressão de etiquetas.

Abstract

This report presents the curricular internship of the Master's in Computer Engineering – Mobile Computing, carried out at CapTemp, Lda. (Pombal, Portugal) between September 2024 and June 2025, with a duration of nine months. The work arose from the need to modernise the Senslive ecosystem and to increase the efficiency of technical and administrative processes, while ensuring traceability and the quality of the service provided to clients.

Before the introduction of the Senslive ecosystem, the solution offered by the company relied on a local infrastructure: each client was provided with a dedicated computer responsible for storing the data collected by monitoring devices and for performing backups to mitigate the risk of data loss. Although this approach enabled historical data storage, it entailed distributed management, higher maintenance complexity and limited scalability.

With the growth of cloud computing technologies and the need to centralise information, CapTemp decided to evolve towards a cloud service, materialised in the Senslive ecosystem. This platform moves data storage and processing to the cloud, providing a centralised solution, accessible from multiple devices and easier to maintain and scale. In addition to representing a technological evolution compared to the previous system, this migration significantly improves the user experience by offering personalised dashboards, access to and download of reports, and data analysis features.

In this context, the project developed several integrated solutions aimed at enhancing the Senslive ecosystem and supporting the company's daily operations. One of these solutions is the Real-time Chat/Notifications project, which provides a conversational channel with immediate responses to queries about records and sensor status. The solution combines WebSocket for interactive communication with push notifications (APNs/FCM) for reliable background delivery, ensuring low response times and service continuity.

The iLog application (Android + cloud) enables the configuration and reading of DS1921G dataloggers and the automatic synchronisation of missions and readings with Senslive, focusing on mobile operation and the centralisation of data for analysis, chart generation and documentation output. The iLog PRO application (Android + local export) provides an offline-first workflow for exporting data in .xlsx format and generating reports

with line charts and visual highlighting of threshold exceedances, which is particularly useful in scenarios where immediate upload to the cloud is not appropriate.

Finally, the internal CTFlash/CTFirmware platform automates mass firmware programming, provides a scanning module for probe screening, integrates the creation of laboratory requests and supports label printing via API, thereby reducing manual tasks and reinforcing the traceability of equipment and processes.

Keywords: Senslive ecosystem; cloud platform; WebSockets and push notifications (APNs/FCM); Android/Kotlin; DS1921G dataloggers; firmware automation; Excel export and reporting; label printing API.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Siglas e Acrónimos

ADB	Android Debug Bridge
ADR	Architecture Decision Record
API	Application Programming Interface
APNs	Apple Push Notifications Service
BD	Base de Dados
BLE	Bluetooth Low Energy
CLI	Command Line Interface
CMD	Command Prompt or cmd.exe
CRC16	16-bit Cyclic Redundancy Check
CRUD	Create, Read, Update and Delete
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheets
ESTG	Escola Superior de Tecnologia e Gestão
FCM	Firebase Cloud Messaging
GMT	Greenwich Mean Time
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IHC	Human-Computer Interaction
IoT	Internet of Things
JSON	JavaScript Object Notation
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
MB	Megabyte
MKT	Mean Kinetic Temperature
NB-IoT	Narrowband Internet of Things
OS	Operating System
OTAA	Over-the-Air Activation
PCB	Printed Circuit Board
PHP	Hypertext Preprocessor
PID	Product ID
QA	Quality Assurance

RBAC	Role-Based Access Control
ROM ID	Read-Only Memory Identifier
SDK	Software Development Kit
SLIs	Service Level Indicators
SLOs	Service Level Objectives
SPA	Single-Page Application
SSL	Secure Sockets Layer
TLS	Transport Layer Security
UI-UX	User Interface/User Experience
URL	Uniform Resource Locator
USB	Universal Serial Bus
VID	Vendor ID
VM	Virtual Machine
WSS	WebSocket Secure
XML	Extensible Markup Language

1. Introdução

O presente relatório enquadra o estágio curricular realizado no âmbito do Mestrado em Engenharia Informática – Computação Móvel, na CapTemp, Lda., empresa especializada no desenvolvimento de soluções de monitorização e controlo remoto baseadas em sensores com e sem fios, bem como numa plataforma *cloud* — o *Senslive* — orientada para a visualização, alarmística e geração automática de relatórios [1].

O ecossistema *Senslive* constitui a componente central da solução tecnológica da empresa, permitindo a gestão integrada de múltiplos dispositivos, a centralização da informação recolhida e o acesso distribuído a funcionalidades de monitorização, consulta e análise de dados [2]. A sua utilização em setores regulados, como a saúde, a logística, o transporte e as infraestruturas técnicas, exige uma conformidade estrita com normas como a EN 12830:2018 [3] e boas práticas de auditoria digital, em que a rastreabilidade, a fiabilidade e a capacidade de resposta assumem um papel crítico.

No âmbito da Computação Móvel, a solução beneficia da recolha de dados em campo, da resiliência offline, da comunicação segura e da integração com serviços *cloud*. Estas áreas são particularmente relevantes em sistemas distribuídos e aplicações móveis modernas, em que a fiabilidade, a segurança e a sincronização de dados assumem um papel central e representam desafios técnicos essenciais em ambientes reais de utilização.

1.1. Ano letivo, duração e período do estágio

O estágio curricular insere-se no ano letivo de 2024/2025 e teve a duração de nove meses, tendo decorrido entre 16 de setembro de 2024 e 5 de junho de 2025, nas instalações da CapTemp, em Pombal, Portugal, em regime presencial. O horário de trabalho observou o regime de oito horas diárias.

1.2.Objetivos do estágio e relação com o programa

No âmbito do estágio, foram definidos os seguintes objetivos, alinhados com o plano curricular do ciclo de estudos:

1. Complementar a formação académica através do contacto com a realidade profissional, em contexto empresarial, na área da Computação Móvel.
2. Aplicar, em contexto real, os conhecimentos e competências teórico-práticas adquiridos ao longo do curso, nomeadamente no desenvolvimento *Android/Kotlin*, na comunicação em tempo real, na integração com serviços *cloud*, em bases de dados móveis e na geração de relatórios.
3. Desenvolver práticas de trabalho adequadas ao contexto empresarial e favorecer a integração profissional e social, em particular em equipas multidisciplinares e em ambientes sujeitos a constrangimentos de produção.
4. Consolidar hábitos de trabalho, sentido de responsabilidade e capacidade de comunicação técnica, com enfoque na qualidade, na rastreabilidade e na segurança.

No plano operacional, o programa de estágio estruturou-se em torno de quatro projetos principais:

- Projeto 1 — Chat/Notificações (tempo real): desenvolvimento de um canal conversacional com *WebSocket* e notificações *push* (*APNs*/*FCM*) para consultas e alertas por local.
- Projeto 2 — iLog (Android + *cloud*): programação e leitura de dispositivos DS1921G, com sincronização automática de dados com o *Senslive*.
- Projeto 3 — iLog PRO (Android + exportação local): implementação de um fluxo *offline-first* com exportação em formato *.xlsx* e relatórios com gráfico de linhas e realce de excedências.
- Projeto 4 — CTFlash/CTFirmware (automação interna): automação de tarefas internas, incluindo programação de *firmware* em massa via *Command Line Interface* (CLI), scanner de sondas, pedidos de laboratório e impressão de etiquetas por API.

A concretização destes objetivos verificou-se de forma transversal ao conjunto dos projetos desenvolvidos, com particular incidência nos Projetos 2 e 3, pela articulação entre

mobilidade, integração com dispositivos e serviços *cloud*, e no Projeto 1, pela disponibilização de um serviço de comunicação em tempo real em ambiente produtivo. Paralelamente, o trabalho em equipa, a planificação em sprints e as entregas incrementais foram transversais aos quatro projetos. Simultaneamente, a documentação técnica, a auditoria de alterações e a produção de relatórios contribuíram para consolidar os resultados alcançados e para reforçar a sua rastreabilidade.

1.3. Estrutura do relatório

O relatório encontra-se organizado em cinco capítulos. O Capítulo 1 apresenta o enquadramento do estágio, os seus objetivos e a estrutura do documento. O Capítulo 2 caracteriza a entidade de acolhimento, descrevendo a CapTemp, o ecossistema *Senslive* e os principais serviços e produtos associados. O Capítulo 3 expõe o programa de estágio, detalhando os quatro projetos desenvolvidos, bem como as respetivas opções técnicas e etapas de implementação. O Capítulo 4 apresenta uma análise crítica do trabalho realizado, identificando limitações, propostas de melhoria e perspetivas de evolução futura. Por fim, o Capítulo 5 reúne as conclusões do relatório, destacando o contributo do estágio para a entidade de acolhimento e para o desenvolvimento de competências académicas e profissionais.

2. Caracterização da Entidade de Acolhimento

Este capítulo apresenta a caracterização da CapTemp, Lda., entidade onde decorreu o estágio curricular. São descritos a sua atividade, evolução histórica, estrutura organizacional, recursos humanos, principais desafios e um exemplo de solução desenvolvida, de forma a contextualizar o ambiente técnico e operacional em que o trabalho foi realizado.

2.1. Ficha técnica CapTemp, Lda

A CapTemp, Lda. é uma empresa portuguesa, especializada no desenvolvimento de soluções *IoT* para monitorização ambiental crítica, nomeadamente controlo de temperatura e humidade em cadeias de frio farmacêuticas, laboratórios e indústrias agroalimentares. A empresa opera desde a Zona Industrial da Formiga, em Pombal, e tem como plataforma central o *Senslive*, um portal *cloud* que agrega dados de registadores remotos, fornece alarmística em tempo real e gera relatórios automáticos de conformidade.

A Tabela 1 - Ficha técnica CapTemp, Lda., resume as principais características da organização:

Tabela 1 - Ficha técnica CapTemp, Lda.

Campo	Informação
Designação social	CapTemp, Lda.
Setor de atividade	Sistemas de monitorização e controlo remoto; software *cloud* de monitorização; soluções IoT para temperatura/humidade e outros parâmetros ambientais.
Localização	Rua Dr. José António Varela Pinto, Armazém 1ª, Zona Industrial da Formiga, 3100-513 Pombal, Portugal.
Forma jurídica	Sociedade por quotas (Lda.).
Portal/Plataforma	*Senslive* — portal *cloud* para visualização, alarmística e relatórios automáticos.

2.2. Evolução histórica

Constituída em 25 de julho de 2013, a CapTemp consolidou-se como referência nacional no desenvolvimento de soluções integradas *end-to-end* (dispositivo–*cloud*) para monitorização contínua. Em paralelo com a evolução do *Senslive* [2], o portal cloud da empresa para gestão centralizada de dados, alarmística e consulta operacional, e das linhas de sensores e conectividade, a empresa investiu em conformidade metrológica, nomeadamente com a norma EN 12830:2018 e o enquadramento WELMEC 7.2, tornando o ecossistema apto para setores regulados, como a saúde, a cadeia do frio e a indústria. A arquitetura atual da solução encontra-se representada na (Figura 1).

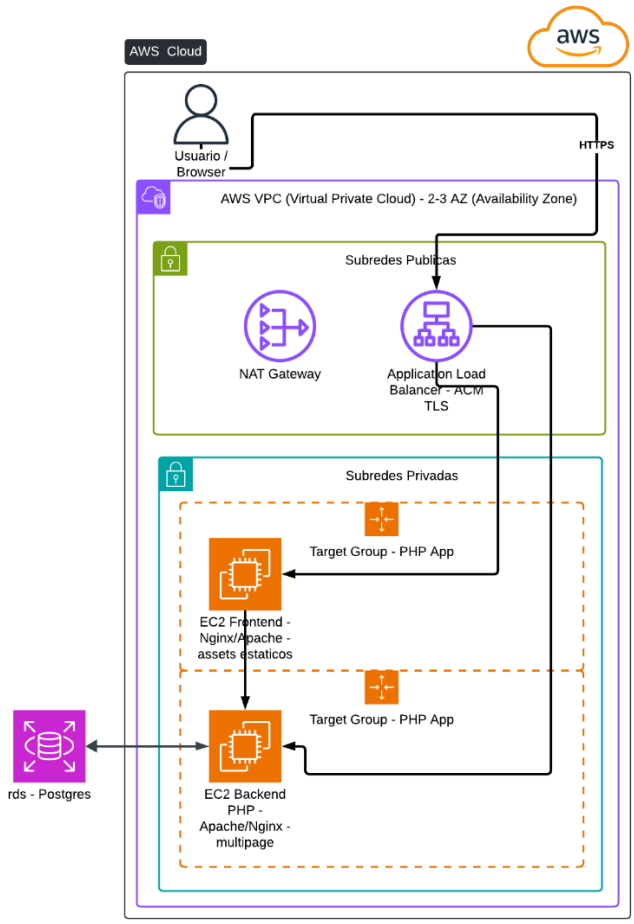


Figura 1 - Arquitetura Senslive da solução atual.

No portefólio de dispositivos, destacam-se registadores de dados e sensores multiparâmetro com opções de medição de temperatura, humidade, CO_2, pressão diferencial e detetor de fugas de água, bem como entradas analógicas (4–20 mA e 0–10 V). Entre os produtos de referência encontra-se a linha AIROLOG, com LCD integrado e conectividade

LoRa (rede privada) ou LoRaWAN (OTAA), concebida para integração nativa com o *Senslive* e alinhada com requisitos de EN 12830:2018, EU Annex 11, FDA 21 CFR Part 11 e GAMP 5, conforme ilustrado na Figura 2.

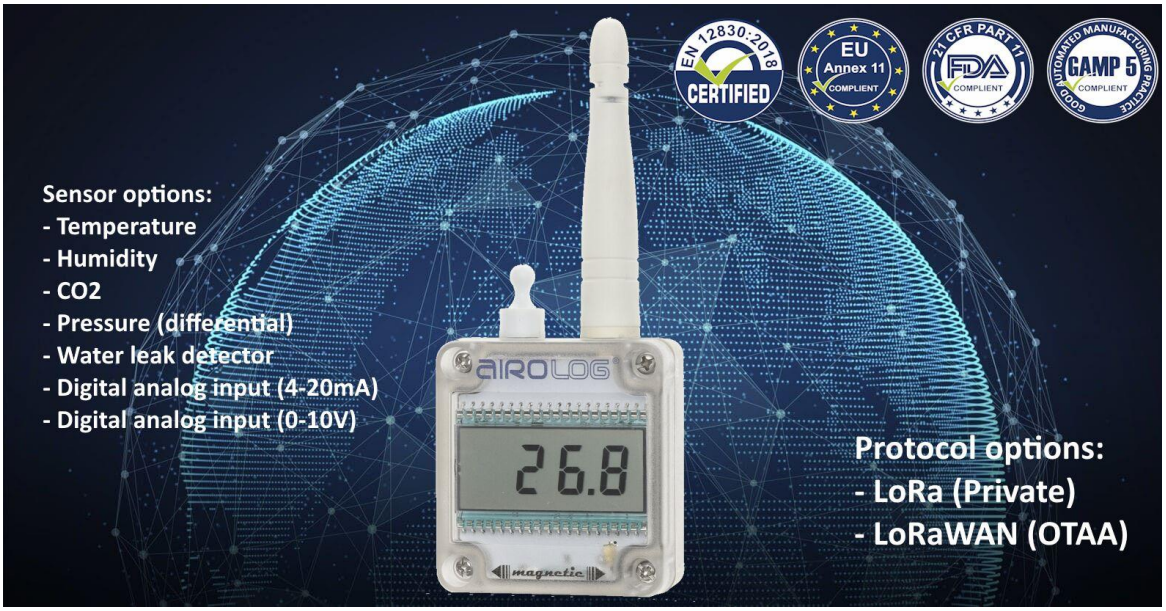


Figura 2 - exemplo de dispositivo AIROLOG [1][4]

2.3. Visão, missão e valores

A atuação da CapTemp é suportada por uma visão de longo prazo, uma missão claramente definida e um conjunto de valores que orientam a tomada de decisão e o relacionamento com clientes e parceiros. Neste contexto, a empresa desenvolve e instala sistemas de monitorização e controlo destinados a ambientes com elevados requisitos de rastreabilidade, qualidade e conformidade, o que torna relevante explicitar os princípios estratégicos que orientam a sua atividade.

Visão. Tornar os dados ambientais operacionais — fiáveis, auditáveis e acionáveis — em qualquer lugar, promovendo qualidade e conformidade.

Missão. Conceber e manter um ecossistema integrado de sensores + redes + *cloud* que permita monitorizar, alertar e certificar com segurança e rastreabilidade.

[1] Figura 2. Captura del post de Captemp en LinkedIn (2025).
Fuente: https://www.linkedin.com/posts/captemp_one-controller-endless-possibilities-activity-7368249788938579968-SxhZ

Valores. Qualidade e conformidade (metrologia legal), inovação prática, segurança e privacidade dos dados, proximidade ao cliente e responsabilidade operacional.

2.4. Estrutura organizacional

A estrutura funcional da CapTemp organiza-se em quatro áreas principais, que asseguram o desenvolvimento, a operação e o suporte das suas soluções. A área de I&D/Engenharia concentra competências em eletrónica, *firmware*, *software* de *backend*, web e mobile, bem como em QA e automação de testes. A área de Operações & Suporte é responsável pelas instalações, assistências técnicas presenciais e remotas, cabendo aos respetivos técnicos a abertura e o acompanhamento de pedidos de verificação em laboratórios acreditados. A área Comercial & Marketing assegura a pré-venda técnica, a gestão de parcerias e a rede de distribuidores, enquanto a área Administrativo-Financeiro apoia as atividades de compras, logística e conformidade

Esta descrição apresenta de forma sintética a estrutura funcional da empresa. A verificação e a calibração são efetuadas por laboratórios externos acreditados, sendo os pedidos e o respetivo acompanhamento operacional conduzidos pelos técnicos da área de Operações & Suporte.

2.5. Recursos Humanos (foco na área do curso)

A CapTemp desenvolve internamente tanto o *hardware* como o *software* dos seus dispositivos de monitorização, realizam instalações dos equipamentos nos clientes e presta assistência técnica em campo. Esta capacidade integrada, que abrange desde a programação de microcontroladores até à implementação final, exige uma equipa multidisciplinar cujas competências técnicas foram fundamentais para o sucesso do estágio em Computação Móvel.

Esta equipa integra perfis de engenharia de software (Kotlin/Android, Laravel/PHP, integrações *cloud*), engenharia eletrónica/telecomunicações (LoRa/ISM/NB-IoT, programação de microcontroladores), QA/metrologia e técnicos de operações e suporte (instalações físicas, assistência técnica e *field service*). Na área de Computação Móvel, salientam-se competências em desenvolvimento Android, comunicação em tempo real (WebSockets/push), armazenamento local e sincronização resiliente com serviços *cloud* (*offline-first*), diretamente relacionadas com os projetos desenvolvidos no âmbito do estágio.

2.6.Análise SWOT

A análise SWOT (*Strengths*/ Forças, *Weaknesses*/ Fraquezas, *Opportunities*/ Oportunidades, *Threats* / Ameaças) constitui uma ferramenta estratégica que permite avaliar os fatores internos e externos que influenciam a posição competitiva da empresa no seu setor de atividade. No caso da CapTemp, esta análise é particularmente relevante porque a empresa atua num mercado técnico e regulado, em que a capacidade de integrar *hardware*, *software* e serviços *cloud*, bem como de responder a exigências de rastreabilidade e conformidade, condiciona a evolução do negócio. Assim, a Tabela 2 - Análise SWOT permite identificar os principais elementos diferenciadores do ecossistema *Senslive*, bem como os desafios associados à expansão e à manutenção da qualidade do serviço.

Tabela 2 - Análise SWOT

Forças	**Fraquezas**
Especialização em monitorização	Dependência de múltiplos protocolos
Plataforma *cloud Senslive* madura	Coexistência com sistemas legados
Enfoque em conformidade metrológica	
Oportunidades	**Ameaças**
Procura por monitorização certificada	Requisitos regulatórios em evolução
Automação de processos internos	Riscos de cibersegurança
Expansão para novos mercados	Conectividade intermitente

2.7. Clientes, fornecedores e evolução recente das atividades

A CapTemp atua num ecossistema de monitorização ambiental certificada que envolve clientes com necessidades críticas de rastreabilidade, uma cadeia de fornecimento

especializada e uma evolução estratégica centrada na *cloud* e na mobilidade. Esta caracterização permite compreender o contexto operacional em que se inseriu o estágio curricular.

Clientes - Organizações dos setores de saúde (hospitais, farmácias, cadeia do frio), logística/transporte (armazenagem refrigerada, veículos), retalho alimentar (supermercados, distribuição) e salas de IT/telecomunicações (*data centers*), todas com necessidade de monitorização 24/7, alarmística instantânea e relatórios automatizados para auditorias regulatórias (EN 12830, GxP).

Fornecedores - Fabricantes e distribuidores de sensores (temperatura, humidade, CO_2), *gateways* e componentes eletrónicos, além de parceiros de conectividade (LoRa/ISM/NB-IoT). A cadeia de fornecimento é rigorosamente selecionada por critérios de conformidade metrológica, qualidade certificada e suporte técnico contínuo, minimizando riscos na produção de dispositivos *end-to-end*.

Evolução recente das atividades - Nos últimos anos, a empresa acelerou a transição para serviços *cloud* (*Senslive*), reforçando a segurança com encriptação *end-to-end* e controlos de acesso avançados, além de investir em mobilidade (aplicações moveis) e interoperabilidade (integrações API). O *portefólio* evoluiu para soluções verticalizadas — transportes/logística (monitorização GPS + temperatura), *healthcare* (conformidade FDA 21 CFR Part 11) e salas de TI(IT-*rooms*) (UPS + deteção de inundações), incorporando relatórios automáticos e alarmística multicanal, tendências diretamente refletidas nos projetos do estágio.

2.8.Principais desafios

O crescimento sustentável da CapTemp enfrenta desafios que exigem uma adaptação contínua da sua estrutura operacional e técnica. Entre os principais, destacam-se:

- Capacidade Operacional e *Onboarding* Técnico - O número de clientes tem aumentado significativamente, impulsionado pela procura por software à medida e por novas instalações. Contudo, a equipa técnica responsável pela instalação e manutenção do ecossistema é reduzida e lida com um produto proprietário de elevada complexidade. A ausência de um plano estruturado de *onboarding* técnico — centrado na arquitetura *Senslive* e na montagem de soluções —, de

documentação operacional — como *runbooks* e *playbooks* — e de certificação interna por níveis acarreta o risco de aumento dos tempos de resposta e de impacto direto na qualidade do serviço prestado.

- Conformidade Normativa e Cibersegurança - Garantir a conformidade contínua com normas de metrologia legal e assegurar a privacidade e a segurança dos dados em arquiteturas *IoT* distribuídas constitui um desafio permanente, especialmente num contexto de requisitos regulatórios em evolução e de ameaças crescentes de cibersegurança.
- Gestão de Conectividade e Operação Offline - A heterogeneidade das tecnologias de rede utilizadas, designadamente LoRa, NB-IoT e interfaces seriais, bem como a necessidade de assegurar resiliência em cenários de conectividade intermitente, exigem mecanismos robustos de sincronização, tolerância a falhas e continuidade de operação. Escalabilidade e Observabilidade - A manutenção da escalabilidade da plataforma *cloud* e a garantia de observabilidade (*observability*) em tempo real dos sistemas implantados são essenciais para suportar o crescimento da base instalada e assegurar níveis de serviço consistentes.

2.9. Exemplo de solução desenvolvida — Abacus: Smart Vision System (contagem de ovos)

Abacus é um sistema de visão inteligente dedicado à contagem automática de ovos, concebido e implementado pela CapTemp. Baseia-se em Inteligência Artificial (aprendizagem profunda) para deteção, *tracking* e controlo de qualidade, integrando-se com a plataforma *cloud Senslive* para supervisão, relatórios e alarmística. [5] ver (Figura 3).

Principais características

- Preparado para Indústria 4.0 (integração *web*/API e telemetria em tempo real).
- Plataforma web responsiva com *preview* em tempo real.
- Algoritmos de IA rápidos e precisos para deteção e seguimento de ovos em tapetes transportadores de qualquer tipo e dimensão.
- Robustez a variação de cor, tamanho e condição (p.ex., sujidade, fissuras).
- Exatidão reportada de 99,8% em cenários operacionais.

- Compatibilidade total com o portal *Senslive* (envio de contagens, eventos e relatórios).

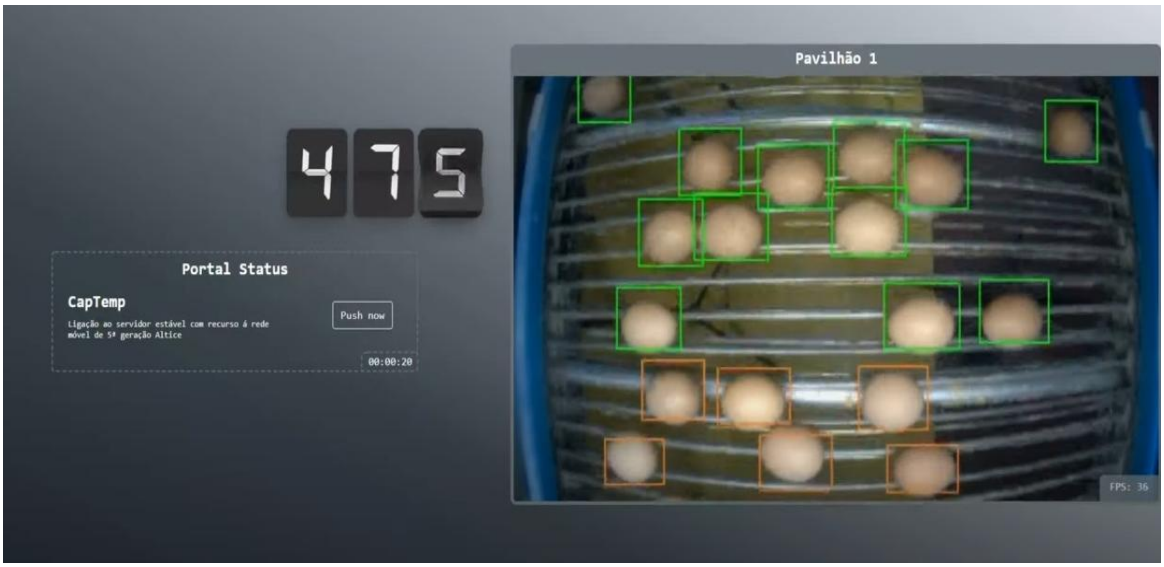


Figura 3 - Abacus: amostra de deteção e seguimento com preview em tempo real. [2][6]

[2] Figura 3. Captura del video "One Controller, Endless Possibilities" (2024).
Fuente: https://youtu.be/REbkOo9d7fI

3. Programa de Estágio

O programa de estágio estruturou-se em torno de quatro projetos principais - ***Chat/*Notificações em tempo real**, **iLog** (Android + *cloud*), **iLog PRO** (Android + exportação local) e **CTFlash**/**CTFirmware** (automação interna) — que cobrem o ciclo completo dispositivo ⇄ *cloud* ⇄ operação. Em todos os projetos foram aplicados conhecimentos do curso de Computação Móvel e áreas afins (Sistemas Distribuídos, Bases de Dados, Redes/Segurança, Engenharia de *Software* e HCI/UI-UX), validando-os em contexto real com requisitos funcionais, constrangimentos técnicos e prazos definidos.

Neste enquadramento, o estágio teve como principais objetivos aproximar o suporte ao cliente dos dados operacionais, através de um canal de *chat* em tempo real; capacitar as operações de campo com aplicações Android para configuração e leitura de dispositivos DS1921G; e automatizar tarefas internas morosas, como a programação de *firmware*, a criação de pedidos de laboratório e a impressão de etiquetas, de forma a agilizar o trabalho dos técnicos e reforçar a rastreabilidade. Em cada projeto foram aplicadas práticas de prototipagem, implementação incremental, testes funcionais e de integração e documentação técnica, garantindo a rastreabilidade e a qualidade do desenvolvimento.

Foi adotada uma abordagem ágil inspirada em *Scrum*, suportada por *Gitea* como ferramenta de gestão de projeto e controlo de versões. Realizaram-se planeamentos semanais com o supervisor da CapTemp (Eng.º João Paulo Agostinho), nos quais se definia e priorizava o *backlog*, se clarificavam critérios de aceitação e se alinhavam dependências. No final de cada semana eram efetuadas *reviews*/demonstrações do incremento entregue e curtas retrospetivas para ajustar o plano seguinte.

Os requisitos emergentes e as provas de conceito/testes (funcionais e de integração) eram conduzidos conjuntamente com o supervisor, que detém um conhecimento aprofundado da infraestrutura de bases de dados e do *Senslive* (sendo atualmente o responsável pelo portal). Esta proximidade acelerou a validação técnica e a resolução de impedimentos. A cada iteração, registei decisões de arquitetura, contratos de interface e evidências de teste (casos e resultados), assegurando a rastreabilidade e preparação dos projetos.

Para o controlo de versões e o planeamento foi utilizado o Gitea corporativo, como repositório central e plataforma de gestão visual. Em cada projeto foi mantido um quadro *Kanban* com as colunas *To Do* → *In Progress* → *Done* (equivalentes a Por Fazer → Em Progresso → Concluído), no qual as *issues* eram criadas e priorizadas por funcionalidade ou tarefa, com *labels* de área e de urgência, e agrupadas em *milestones* semanais (sprints). Cada *issue* dava origem a uma *branch* dedicada (padrões *feature/...*, *fix/...*); o desenvolvimento decorria nessa *branch* com *commits* descritivos, seguindo convenções de mensagem (modo imperativo, referência à *issue*). As alterações eram submetidas por *Merge Request* (*MR*), preservando o histórico de discussão, os critérios de aceitação e as evidências (capturas de ecrã, *logs*, artefactos), o que assegurou rastreabilidade ponta a ponta entre planeamento → execução → validação → integração e facilitou auditorias ou reversões quando necessário.

3.1. Descrição geral das tarefas e articulação teoria–prática

O estágio desenvolveu **quatro projetos interdependentes**, cada um articulando disciplinas específicas da formação académica com necessidades operacionais críticas da CapTemp:

Projeto 1 — Chat e Notificações

Foram aplicados fundamentos de Sistemas Distribuídos e Redes no desenvolvimento de um canal *WebSocket* (WSS) de baixa latência, articulado com *push notifications* (APNs/FCM) para aplicação em *background*. O protocolo implementa mensagens tipificadas, *keep-alive* e reconexão automática com garantia *at-least-once*, enquanto a persistência em PostgreSQL incorpora índices otimizados e paginação eficiente. Na área de Segurança Informática, foram aplicados TLS 1.3 e gestão segura de *tokens* JWT.

Projeto 2 — iLog (Android)

A solução articulou Computação Móvel com Arquiteturas de Dispositivos Embebidos, implementando comunicação USB *Host* com adaptador 1-Wire DS9490R e *driver* de alto nível para DS1921G. O ciclo de programação de missão seguiu o padrão *Write/Read/Copy Scratchpad*, com verificação de integridade através de CRC-16 conforme protocolo MODBUS. O armazenamento *offline-first* adotou *ObjectBox*, selecionado após estudo comparativo com Realm e SQLite — onde se destacou pelo melhor compromisso entre

latência e *footprint* de memória. Os protótipos Adobe XD foram convertidos em *flows* de UX/HCI com validações semânticas da missão.

Projeto 3 — iLog PRO

Consolidou-se um pipeline sem rede, com leitura local, exportação *.xlsx* e relatórios incluindo gráfico temporal e marcação de excedências térmicas. A arquitetura implementa padrões de sincronização assíncrona (desacoplamento por filas ObjectBox, *retries* com *backoff* exponencial e reconciliação posterior), otimizada para cenários de conectividade intermitente em contexto de campo.

Projeto 4 — CTFlash/CTFirmware

Teve o principal foco em Engenharia de Software e *DevOps*, desenvolvendo *Command Line Interface* (CLI) para programação em massa de *firmware* (mapa versão→HEX), *logs* de auditoria, *scanner* de sondas TH3, geração de pedidos de laboratório (PT/ES) e impressão de etiquetas via API. A solução cruza princípios de automação, qualidade e rastreabilidade, sustentando a escalabilidade operacional da CapTemp.

3.2.Síntese técnica dos quatro projetos

3.2.1. Projeto 1 — Chat/Notificações (tempo real)

- Arquitetura com serviço de *WebSocket* próprio (controlo de sessões e *tokens*), complementado por APNs (iOS) e FCM (Android) para *push* em 2.º plano.
- Autenticação: QR Code/PIN → troca por *token* de sessão; escopo por local e cliente.
- Mensagens: *schema* compacto (pedido de estado, resposta com *payload* atual, pedido de documentos/*logs*); serialização JSON; *heartbeats* e *reconnect backoff*.
- Resiliência: verificação de faltas com ***notification_last_fired*** e reenvio; consulta de pendentes em ***alerts_notifications***.
- Persistência local (Android): avaliação Realm/ObjectBox/SQLite.

3.2.2. Projeto 2 — iLog (Android + cloud)

- Programação/leitura de DS1921G; parâmetros de missão: *sample rate*, data/hora de início, limite mínimo e máximo, *rollover*, *offset* de calibração, PIN.

- USB-OTG e deteção de dispositivo; *pipeline* de sincronização com o *Senslive* (envio automático/manual).
- Armazenamento avaliação Realm/ObjectBox/SQLite.
- Persistência local (Android): avaliação Realm/ObjectBox/SQLite.

3.2.3. Projeto 3 — iLog PRO (Android + exportação local)

- Mesma base de missão/leitura que o iLog; exportação para Excel (*.xlsx*) com parâmetros, tabela de leituras e gráfico de linhas.
- Realce visual de excedências (abaixo/entre/acima dos limites) e geração de relatório para partilha rápida.
- Operação *offline-first* (sem envio para *cloud*), histórico local e filtros por missão/intervalo temporal.
- Persistência local (Android): avaliação Realm/ObjectBox/SQLite.

3.2.4. Projeto 4 — CTFlash/CTFirmware (automação interna)

- Programação de *firmware* em massa via CLI; mapeamento versão → ficheiro HEX; registo de auditoria (quem/quando/dispositivo).
- *Scanner* de sondas e apoio a calibração e monitorização dos sensores.
- Criação de pedidos de laboratório com metadados extraídos do registrador.
- Impressão de etiquetas por API (etiquetas dinâmicas por sensor ou registrador).
- Gestão de ativos/MACs (LoRa/ISM; verificados ou não verificados).

3.3. Tecnologias e ferramentas utilizadas

Para garantir coerência entre os quatro projetos — Chat/Notificações, iLog, iLog PRO e CTFlash/CTFirmware — foi organizado um ecossistema de ferramentas que cobre edição de código, teste de APIs e *WebSockets*, desenvolvimento *mobile*, gestão de bases de dados, prototipagem, controlo de versões, instrumentação série, virtualização e observabilidade. Descrevem-se de seguida, em linguagem objetiva, as principais tecnologias e ferramentas utilizadas, bem como a respetiva adequação ao contexto técnico do estágio.

No planeamento e no controlo de versões, toda a gestão do trabalho foi centralizada em Gitea [7], através de *issues* por funcionalidade, *milestones* semanais, quadro Kanban e fluxos

de integração baseados em *branches* dedicadas e *merge requests*, assegurando rastreabilidade entre requisitos, desenvolvimento e integração. Esta abordagem permitiu manter disciplina no ciclo de desenvolvimento e preservar o histórico técnico das alterações efetuadas.

Nos projetos iLog e iLog PRO, o desenvolvimento foi suportado em Android Studio, em articulação com a documentação oficial do Android *Developers* [8], o que permitiu gerir *builds*, variantes de execução, análise de desempenho, *debugging* e validação de compatibilidade entre versões do sistema operativo. Esta ferramenta revelou-se particularmente importante na implementação de funcionalidades relacionadas com USB Host, *foreground services* e interação com hardware externo.

Na validação da camada de comunicação em tempo real do projeto Chat, foram utilizados clientes *WebSocket* [9] em navegador, permitindo testar ligações WSS, envio de *frames*, *handshakes*, *pings/pongs* e inspeção de mensagens em tempo real antes da validação *end-to-end* com a aplicação móvel. Esta etapa foi útil para isolar problemas de transporte e acelerar a depuração inicial do protocolo.

No apoio às integrações com bases de dados *PostgreSQL* [10], foi utilizada a ferramenta *Adminer*, adequada para operações de consulta, manipulação de tabelas, validação de índices e execução de *queries* em ambientes de teste. A sua leveza e simplicidade facilitaram o acesso rápido a esquemas auxiliares e a inspeção de dados sem dependência de ferramentas mais pesadas.

Como apoio à análise comparativa com soluções anteriores da empresa, foi também consultada a documentação pública associada à CapTemp e a aplicações legadas utilizadas historicamente no processamento local de dados (*MonTemp*). Esta referência permitiu comparar fluxos, parâmetros de missão e saídas documentais com as novas soluções *mobile* e *cloud* desenvolvidas no contexto do estágio.

Ao nível da prototipagem de interfaces, os fluxos críticos foram desenhados em Adobe XD [11], nomeadamente para programação de missão, leitura, sincronização, exportação e alertas, permitindo acelerar o processo de *wireframing*, prototipagem e alinhamento visual antes da implementação. O uso desta ferramenta contribuiu para manter consistência de interface e validar interações com antecedência.

Na instrumentação de comunicações série e de entrada/saída, foi utilizada a ferramenta *Hercules SETUP Utility* [12], particularmente útil no Projeto 4 para depuração de portas série, testes de *baud rate*, validação de *frames* e análise de *timeouts* em dispositivos físicos. Esta ferramenta foi importante na fase de teste do scanner de sondas e noutras rotinas ligadas à comunicação com hardware.

Para garantir isolamento, repetibilidade e facilidade de recuperação de ambientes, parte da infraestrutura associada ao CTFlash/CTFirmware foi alojada em máquina virtual com VMware Workstation [13]. A utilização de virtualização permitiu criar ambientes reprodutíveis, preservar *snapshots* de configuração e reduzir o risco operacional durante ensaios e validações.

Na camada de observabilidade e monitorização de desempenho da aplicação CTFlash, baseada em Laravel [14], foi integrada a plataforma *Inspector.dev*, permitindo acompanhar latências, erros, execução de *jobs* e comportamento geral da aplicação através de *dashboards* e alertas. Esta integração facilitou a deteção precoce de problemas de desempenho e contribuiu para uma abordagem mais preventiva na análise operacional.

Em conjunto, estas ferramentas compuseram uma *toolchain* pragmática e ajustada aos objetivos do estágio, articulando produtividade de desenvolvimento, qualidade técnica, rastreabilidade e capacidade de diagnóstico. O seu uso contribuiu para encurtar ciclos de *feedback*, suportar decisões técnicas informadas e melhorar a robustez das soluções desenvolvidas.

3.4. Projeto 1 — Chat em tempo real para o Senslive

O principal objetivo consistiu em disponibilizar um canal conversacional, seguro e de baixa latência, que permita a clientes com subscrição ativa consultar o estado de registradores/sensores, receber alertas em tempo real e solicitar documentos (*logs*, relatórios) por local, integrando-se com o ecossistema *Senslive*.

O sistema é composto por um servidor *WebSocket* (WSS) que comunica com PostgreSQL e por aplicações móveis (Android/iOS) que combinam WebSocket em *foreground* com notificações *push* em *background* (FCM para Android, APNs para iOS). Quando a app está ativa, o diálogo decorre via WSS, caso contrário, as notificações são entregues por *push* (ver Figura 4).

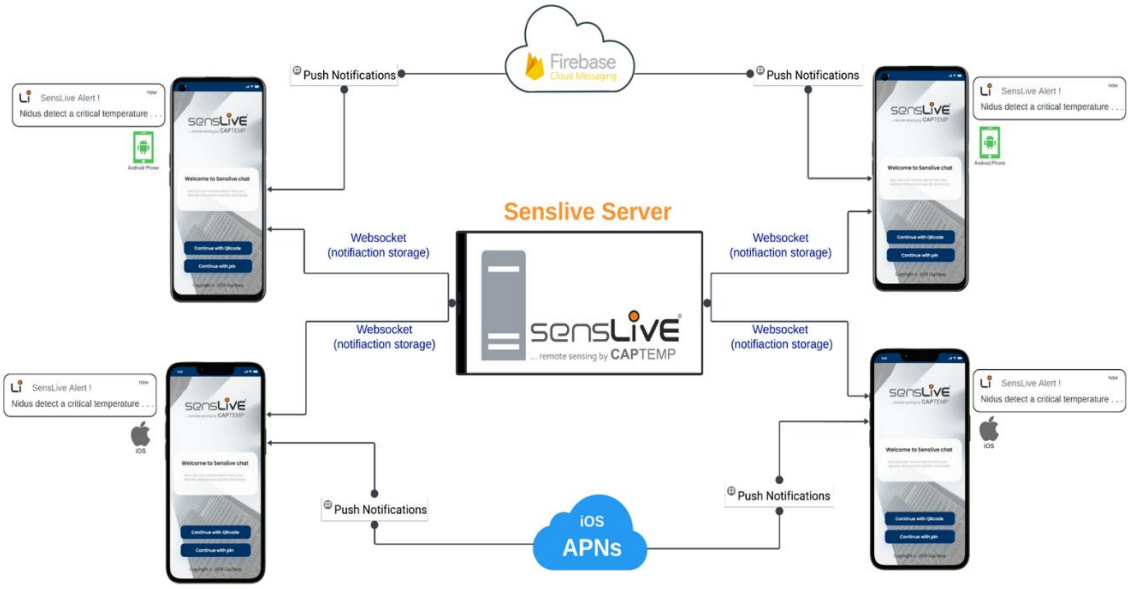


Figura 4 - Arquitetura geral e do servidor Projeto 1 - Chat/Notificações.

3.4.1. Servidor WSS (detalhes técnicos)

1. Configuração e acesso a dados - O servidor inicializa carregando parâmetros operacionais (porta de escuta, limites máximos de clientes, intervalos de *keepalive*) a partir de um ficheiro *config.json*. A ligação à base de dados **PostgreSQL** é estabelecida através do *reverse proxy* interno (porta de serviço 8080), utilizando *connection pooling* para garantir eficiência, isolamento e reutilização segura de ligações.

2. Autenticação - O acesso ao canal conversacional é realizado por *token* ou por QR code/PIN. No caso do *token*, a aplicação móvel gera um envelope cifrado contendo

identificadores temporários, que é desencriptado no servidor através da biblioteca *crypto*. As credenciais correspondentes são validadas na base de dados, garantindo autenticação segura sem exposição de palavras-passe em claro.

3. Gestão de ligações - As sessões ativas são mantidas num registo volátil em memória, estruturado como um mapa de clientes autenticados, sujeito a um limite máximo configurável. O servidor emite *heartbeats* periódicos e monitoriza inatividade; clientes que não enviem *keepalive* dentro da janela definida são terminados automaticamente. O número de clientes ativos é refletido na tabela *notifications_servers*, permitindo *health checks* e monitorização operacional.

4. Notificações e consistência - Para cada cliente autenticado, o servidor consulta notificações pendentes na tabela *alerts_notifications* e envia-as via WSS. Para garantir consistência e evitar perdas, compara-se o campo *notification_last_fired* com o último *acknowledgement* recebido do cliente. Caso exista défice, procede-se ao reenvio das notificações em falta, assegurando reposição após períodos de desconexão. O mecanismo é tolerante a duplicados, sendo as operações idempotentes no lado do cliente.

5. Protocolo de mensagens - O canal utiliza um *payload* compacto em JSON, estruturado segundo um *schema* com tipos de mensagem (*type 0, type 1*, …), cada um associado a rotinas específicas: consulta de estado, pedidos de *logs* ou documentos, *acknowledgements* de receção e mensagens de *keepalive*. O formato JSON foi adotado pela sua simplicidade, interoperabilidade e facilidade de depuração.

6. Segurança - A comunicação entre cliente e servidor é realizada exclusivamente via **WSS**, garantindo cifragem dos dados em trânsito através de certificados **SSL/TLS**. Em ambiente de testes utilizou-se certificado autoassinado; em produção prevê-se gestão segura de chaves, armazenamento protegido e *hardening* do *host*, alinhado com as boas práticas de segurança aplicáveis ao protocolo *WebSocket*.

7. Boas práticas e melhoria contínua - O sistema implementa registo estruturado de erros e eventos, complementando a saída de consola com armazenamento persistente para auditoria e diagnóstico. As consultas à base de dados foram otimizadas através de índices e paginação baseada em cursores, assegurando desempenho consistente mesmo com volumes elevados de notificações.

3.4.2. Aplicações móveis e opções tecnológicas

Seleção arquitetural e desenvolvimento técnico - No âmbito da avaliação de soluções *cross-platform*, procedeu-se à análise comparativa de **Kotlin Multiplatform** e **Flutter**, tendo-se verificado que os requisitos funcionais críticos — designadamente, a receção de *push notifications* com a aplicação totalmente encerrada, o arranque automático aquando do *boot* do dispositivo e a integração profunda com os serviços nativos do sistema operativo — inviabilizavam uma abordagem verdadeiramente *cross-platform*. Assim, optou-se pelo desenvolvimento de duas aplicações nativas distintas: **Android** (em **Kotlin**) e **iOS** (em **Swift**), garantindo o cumprimento integral dos requisitos técnicos essenciais.

Integração de serviços de notificação - Com o objetivo de assegurar a entrega fiável de notificações em todos os estados da aplicação (*foreground*, *background* e *terminated*), foi adotada a stack oficial de cada plataforma: **Firebase Cloud Messaging (FCM)** para Android e **Apple Push Notification Service (APNs)** para iOS. Esta escolha assegura conformidade com as melhores práticas de cada ecossistema e compatibilidade com as políticas de distribuição da *app stores*.

Gestão de comunicação em tempo real - Durante a execução em *foreground*, estabelece-se uma ligação persistente ao ***WebSocket*** **Secure (WSS)**, permitindo comunicação bidirecional com latência mínima. Para garantir robustez e continuidade do serviço, implementou-se um mecanismo sofisticado de reconexão exponencial com *backoff*, complementado por envio periódico de *keepalive*, assegurando a deteção atempada de falhas de conectividade e recuperação automática.

Desenvolvimento de interface e prototipagem - Foram concebidos e implementados protótipos de alta-fidelidade nos modos *light* e *dark*, apresentados na Figura 5 e Figura 6. A base visual dos fluxos críticos — autenticação, listagem por localização e histórico — foi totalmente desenvolvida para validação funcional, mantendo-se alguns ecrãs provisoriamente incompletos em virtude da sua dependência da nova versão do *Senslive*, presentemente em fase de migração tecnológica.

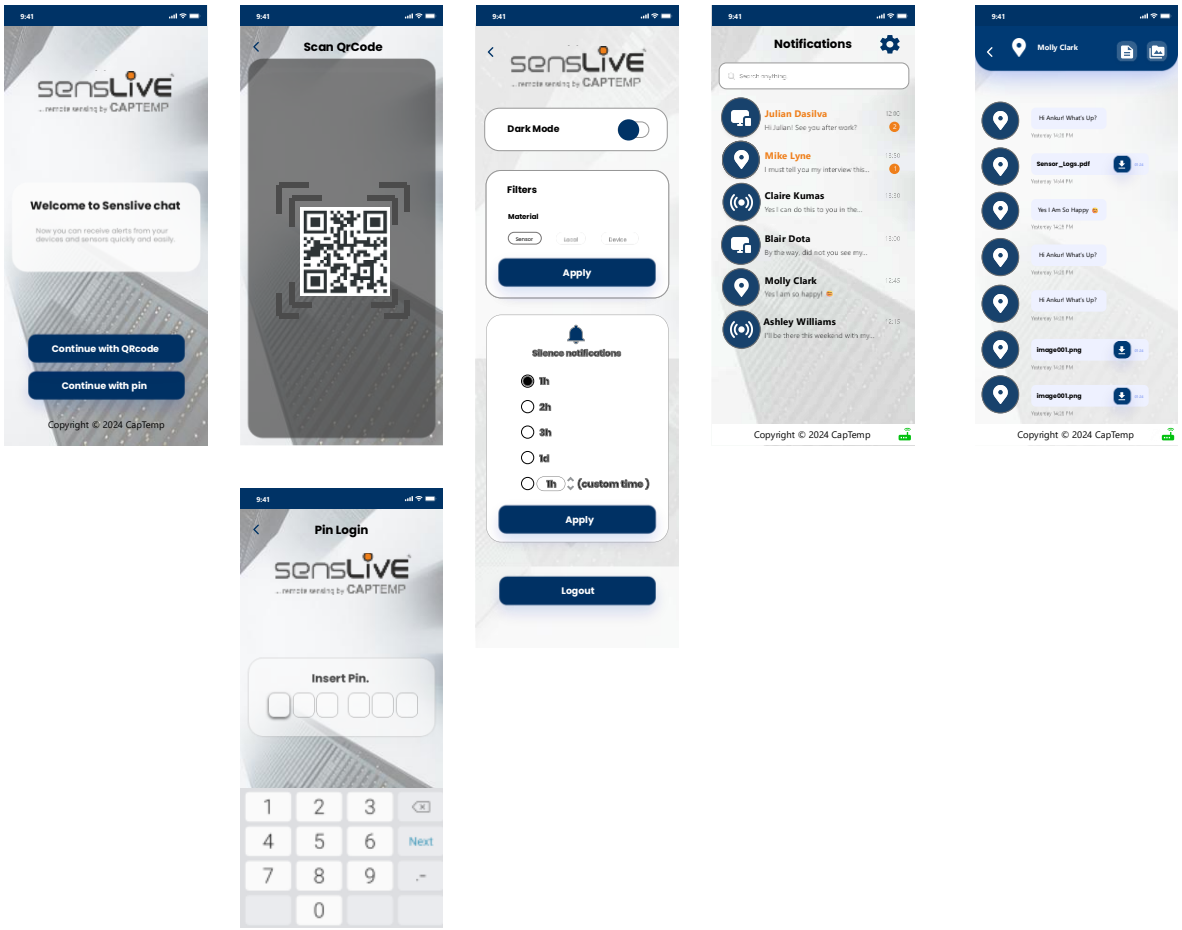


Figura 5 - Protótipos de alta-fidelidade *Light Mode*.

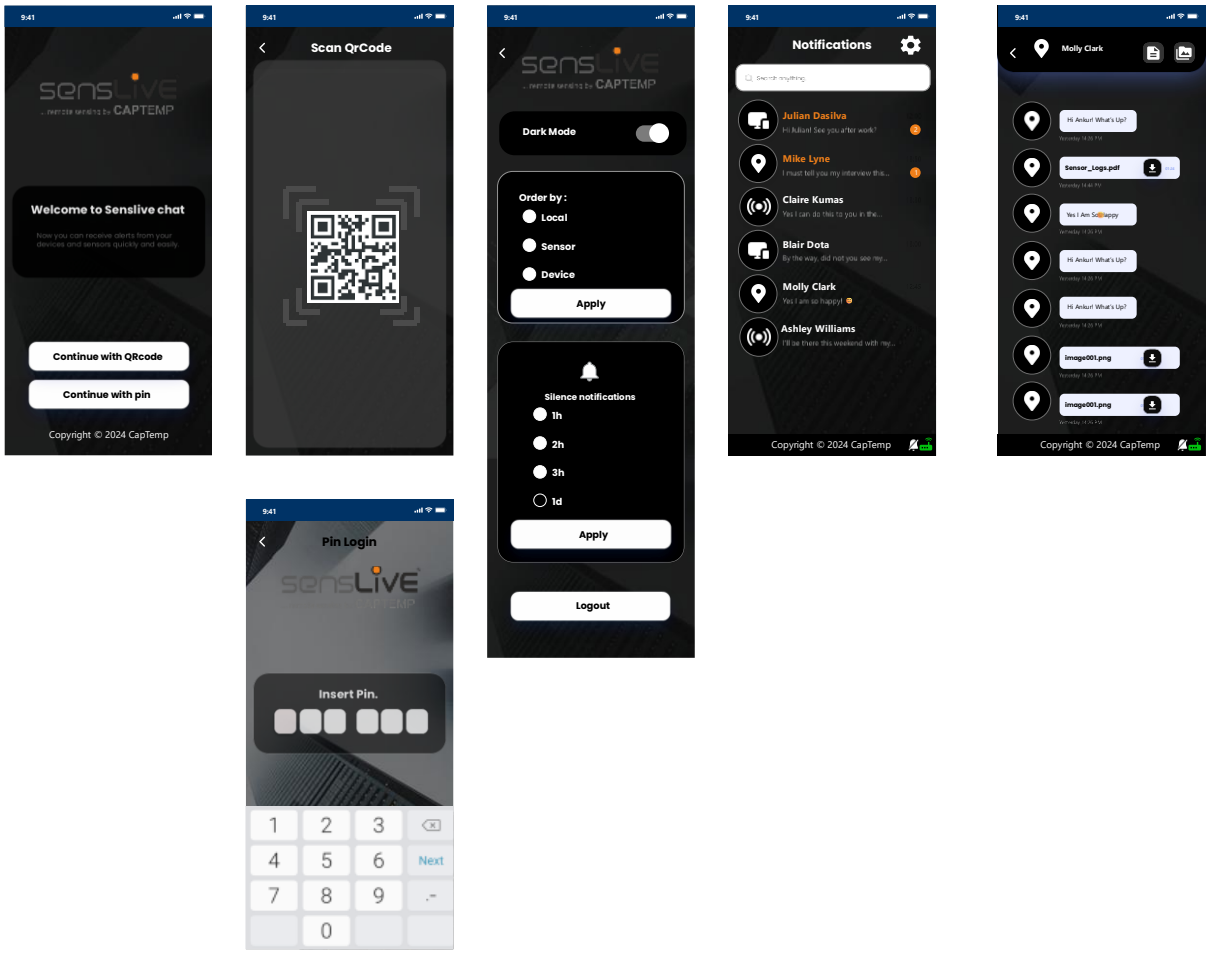


Figura 6 - Protótipos de alta-fidelidade *Dark Mode*.

Persistência local - No contexto do armazenamento local de notificações e do estado no cliente, procedeu-se à avaliação comparativa de três soluções amplamente utilizadas em aplicações móveis e/ou offline: Realm, SQLite e ObjectBox. A análise foi conduzida com base em critérios de desempenho, latência de carregamento e *footprint* global da solução, entendido como a soma do espaço ocupado pela aplicação e pelos respetivos dados persistidos.

Os ensaios foram realizados localmente realizados localmente num dispositivo móvel (Xiaomi Mi 9), com sistema operativo Android, recorrendo a conjuntos de registos de grandes dimensões e relativamente pesados, com aproximadamente 1 MB por registo, de forma a reproduzir condições próximas de utilização real e a testar a robustez das soluções sob carga. Foram comparados os tempos de carregamento e o tamanho total ocupado por cada abordagem, permitindo identificar diferenças relevantes no comportamento de cada

tecnologia em cenário de persistência intensiva. Os resultados encontram-se sistematizados na Tabela 3, onde se apresentam as medições obtidas para 100k e 5000k amostras.

A partir da análise dos dados recolhidos, concluiu-se que **ObjectBox** oferece o compromisso mais favorável entre rapidez de acesso e redução de *footprint*, destacando-se como a solução mais adequada para o caso de uso em questão. Esta decisão revela-se particularmente pertinente em aplicações que exigem arranque rápido, boa responsividade e capacidade de funcionamento *offline*, uma vez que a eficiência no armazenamento local tem impacto direto na experiência do utilizador e no comportamento global do sistema.

Tabela 3 - Resultados obtidos no dispositivo móvel Xiaomi MI 9, com 100k e 5000k amostras.

Base de dados	Tempo de carga 100k (ms)	Tamanho total 100k (MB)	Tempo de carga 500k (ms)	Tamanho total 500k (MB)
Realm	3395–3526	50,7–54,4	22 096–22 396	182–185
ObjectBox	265–267	30,4–33,0	1 582–1 587	111–113
SQLite	1 596–1 599	~27,5–27,7	15 316–15 332	~114

Nota metodológica - Cumpre referir que os resultados apresentados foram obtidos com base em testes realizados num único dispositivo (Xiaomi Mi 9**)**, pelo que poderão verificar-se variações adicionais em função das características específicas do hardware e da versão do sistema operativo utilizada. Neste sentido, os valores obtidos devem ser interpretados como indicativos do comportamento relativo das soluções avaliadas no contexto experimental definido, e não como resultados universalmente generalizáveis. Os gráficos de suporte, relativos ao tempo de carregamento e ao tamanho total por caso de teste, encontram-se anexados e são apresentados na Figura 7 e Figura 8.

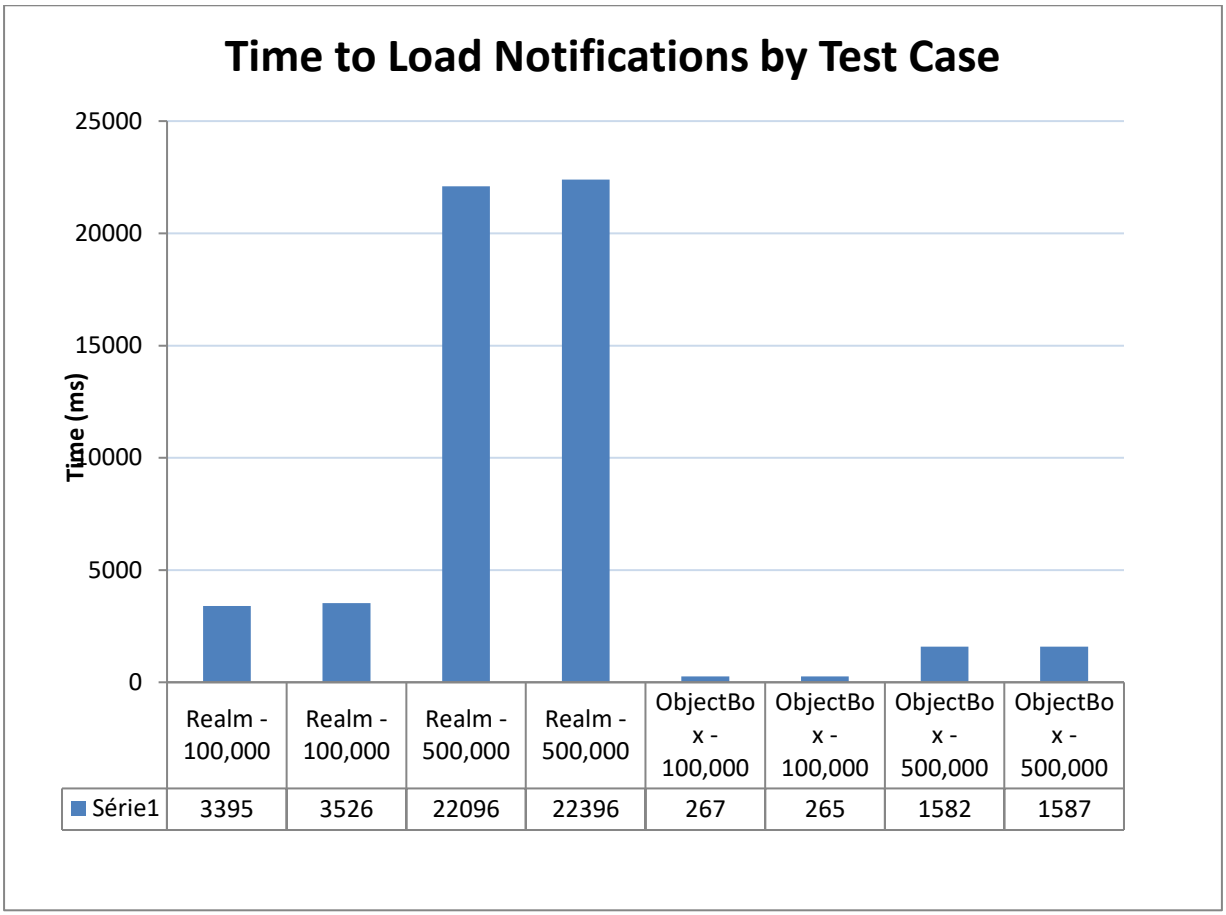
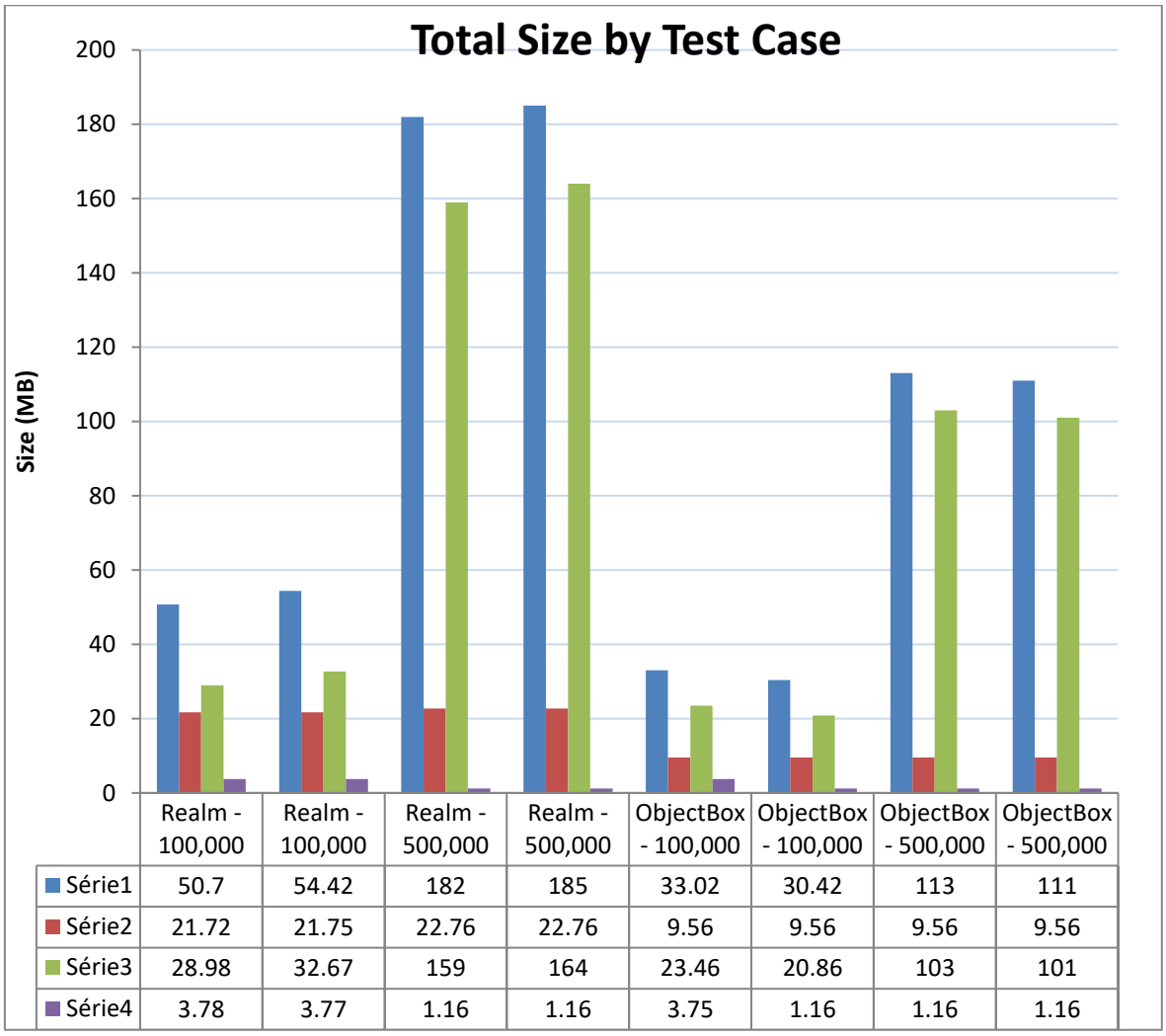


Figura 7 - Gráfico de barras tempo de carga para 100k e 500k amostras.



	Realm - 100,000	Realm - 100,000	Realm - 500,000	Realm - 500,000	ObjectBox - 100,000	ObjectBox - 100,000	ObjectBox - 500,000	ObjectBox - 500,000
Série1	50.7	54.42	182	185	33.02	30.42	113	111
Série2	21.72	21.75	22.76	22.76	9.56	9.56	9.56	9.56
Série3	28.98	32.67	159	164	23.46	20.86	103	101
Série4	3.78	3.77	1.16	1.16	3.75	1.16	1.16	1.16

Figura 8 - Gráfico de Barras tamanho total ocupado para 100k e 500k amostras.

3.4.3. Fluxos principais

1. **Autenticação, autorização e subscrição de eventos -** Fluxo de entrada através de QR *code/PIN* ou *token*, culminando na obtenção do acesso por localização e permissões específicas do cliente. Segue-se o estabelecimento da ligação WSS, cuja sequência de autenticação é apresentada na Figura 9, para subscrição de eventos em tempo real e receção de *push notifications* quando a aplicação se encontra inativa (*background* ou *terminated*).

Figura 9 - Fluxo de Autenticação por PIN.

2. **Consulta de estado e geração documental** - Interrogação do estado dos sensores (ex.: "sensor 3 do registrador 1: 30°C") e solicitação de documentos associados (*logs* e relatórios), conforme ilustrado na (Figura 10).

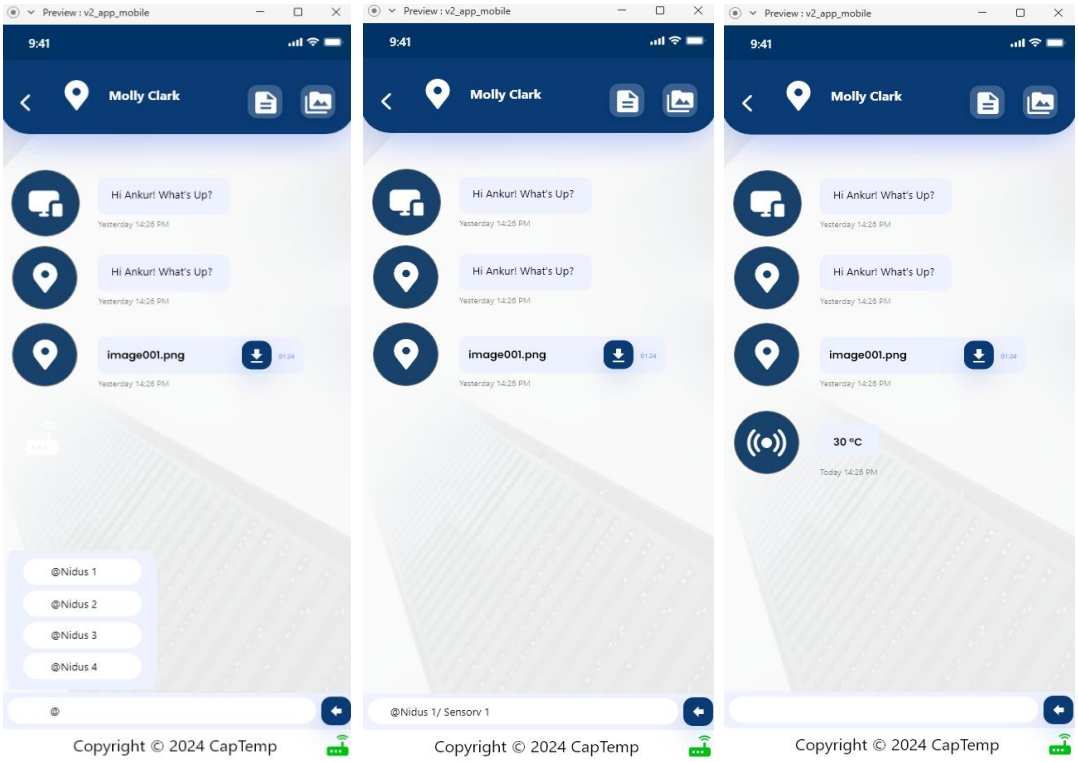


Figura 10 - Consulta de estado de um sensor.

3. **Confirmação de receção** - Registo da receção/visualização através da abertura do *chat*, atualizando o campo *notification_last_fired* para gestão inteligente das notificações subsequentes.

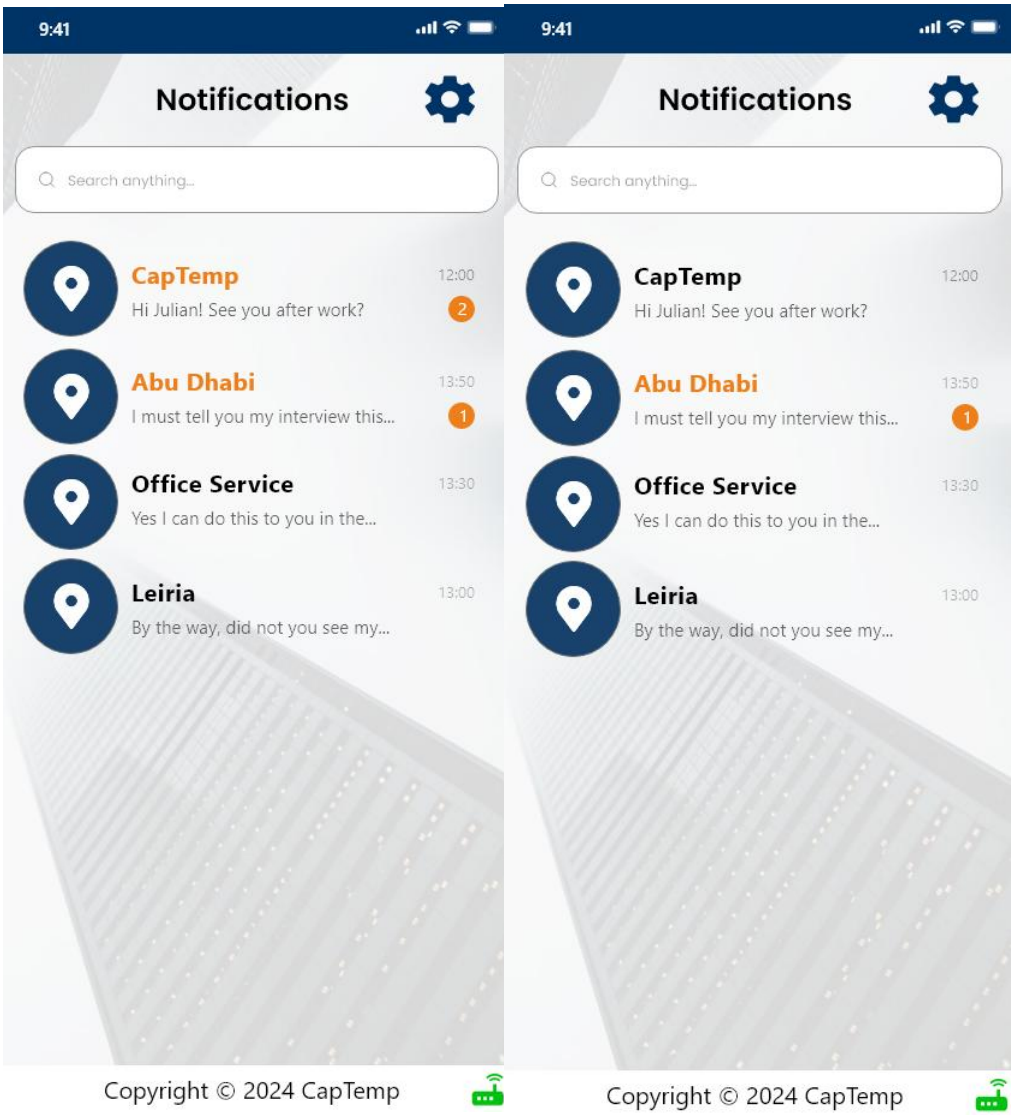


Figura 11 - Receção e visualização de conteúdos.

3.4.4. Estado do projeto

O servidor WSS encontra-se totalmente implementado e operacional no ambiente de testes, utilizando um certificado autoassinado para efeitos de validação preliminar. As aplicações nativas (Android em Kotlin e iOS em Swift) dispõem já dos fluxos essenciais de comunicação bidirecional e integração com serviços de *push notifications* plenamente funcionais.

A interface de utilizador permanece, contudo, em fase de integração com a nova versão do *Senslive*, cuja estabilização se encontra pendente. Esta dependência externa motivou a priorização dos Projetos 3 e 4, desenvolvidos em paralelo, até à disponibilização de uma *release* estável da solução *cloud*, assegurando assim a continuidade do progresso do sistema.

3.4.5. Riscos identificados e respetivas estratégias de mitigação

Riscos operacionais - A possibilidade de perda de mensagens em situações-limite (*edge cases*) foi mitigada através da implementação de mecanismos de *acknowledgement* obrigatórios ao nível da aplicação, garantindo confirmação explícita de receção por parte do destinatário. Complementarmente, foi estabelecida uma *dead-letter queue* individualizada por cliente, destinada ao armazenamento de eventos não processados, assegurando rastreabilidade completa e recuperação automática perante falhas transitórias ou condições anómalas de processamento.

Riscos de segurança - As chaves privadas utilizadas pelo servidor encontram-se devidamente protegidas em armazenamento seguro, enquanto todas as comunicações cliente-servidor são cifradas através do protocolo WSS com certificados SSL/TLS válidos para ambiente de produção. Esta implementação assegura a proteção dos dados em trânsito contra ataques de interceção (*man-in-the-middle*) e cumpre as melhores práticas de segurança preconizadas pelos standards aplicáveis ao protocolo *WebSocket*.

Riscos de desempenho. Para mitigar latências elevadas e reduzir carga sobre a base de dados, as consultas à tabela *alerts_notifications* foram otimizadas através da criação de índices específicos nas colunas mais utilizadas em filtros e ordenações, bem como pela adoção de paginação baseada em cursores, evitando custos de *OFFSET*. Antecipando cenários de escalabilidade horizontal, foi definido que o estado efémero da aplicação — como sessões ativas, sinais de *keepalive* e filas de eventos transitórios — será externalizado para um sistema de *cache* distribuído (ex.: Redis) e para um *message broker* assíncrono (ex.: RabbitMQ), garantindo consistência, baixa latência e desacoplamento entre serviços.

3.4.6. Contributo esperado

O projeto estabelece a infraestrutura de tempo real alinhada com a migração tecnológica do *Senslive*: reduz carga de suporte (autoatendimento), melhora a experiência do cliente e cria base para novos serviços (p. ex., relatórios *on-demand* e automação de tarefas por chat).

3.5. Projeto 2 — *iLog* (Android, integração *cloud*)

A motivação principal para o desenvolvimento do iLog resulta da operação dos *dataloggers* DS1921G [15] dos clientes em modo *self-hosted*, dependentes de instalação e processamento local através da aplicação MONTEMP (PC) para descarregar e visualizar dados, funcionando completamente fora do ecossistema *cloud* Senslive. Este fluxo fragmentado — baseado em software local, não integrado e sujeito a intervenção manual — gerava atrasos operacionais, maior risco de erro humano, dificuldades na auditoria de medições e fraca integração com os processos de alarmística e relatórios do Senslive.

Em setores regulados (cadeia do frio, saúde, indústria), onde a rastreabilidade metrológica e a visibilidade em tempo real são críticas, a inexistência de um *pipeline* unificado comprometia a conformidade normativa e limitava a capacidade de monitorização contínua.

Para ultrapassar estas limitações, foi concebida uma solução móvel *cloud-based*, totalmente integrada no ecossistema *Senslive*. Esta abordagem substitui o modelo *self-hosted* tradicional por um *pipeline* centralizado, escalável e gerido na *cloud*, colocando o *smartphone* Android no centro das operações de campo. Através de USB-C/OTG, o dispositivo liga-se diretamente ao adaptador 1-Wire DS9490R e, deste, às sondas DS1402D-DR8+ e aos DS1921G (Figura 12), garantindo sincronização imediata, redução de latências e integração nativa com a plataforma *Senslive*.

Figura 12 - *Hardware* necessário para usar o projeto iLog

3.5.1. Objetivos do projeto iLog (Projeto 2)

Os objetivos gerais do projeto foram:

- Centralizar missões e leituras no *Senslive*, eliminando o ciclo PC-dependente e garantindo integração nativa com o *Portal Push*.
- Reduzir o *lead time* entre recolha e disponibilização de dados para gráficos, alertas e relatórios.
- Aumentar a qualidade, integridade e auditabilidade das medições, assegurando autenticidade do emissor e rastreabilidade metrológica.
- Melhorar a experiência do técnico em campo, disponibilizando uma aplicação móvel robusta, simples e *plug-and-play*.

Para concretizar estes objetivos gerais, foram definidos os seguintes requisitos:

- Programação completa de missões diretamente no *logger*, incluindo início (até 40 dias), intervalo (1–255 min), limites de temperatura (−40,0…85,0 °C em passos de 0,5 °C), *rollover*, *offset* de calibração e PIN, com validações e pré-visualização.

- Leitura de registos e metadados, seguida de sincronização automática com o Senslive via HTTPS, com *acknowledgement*, fila de reenvio *offline* e seleção de *endpoints* (produção/ensaio).
- Deteção automática do adaptador DS9490R por VID/PID e lançamento imediato da aplicação ao *plug-in*, assegurando uma experiência verdadeiramente *plug-and-play* (Figura 13).
- Suporte para múltiplos *loggers* em paralelo, através de *splitter* RJ11(Figura 15).
- Implementação de mecanismos de integridade e rastreabilidade, incluindo CRC-16/MODBUS e *bitmap* de memória em *little-endian* para metadados de missão.
- Produção de protótipos de alta-fidelidade, manual do utilizador e guia técnico, garantindo manutenibilidade e transferência de conhecimento.

Foram ainda definidas metas para os requisitos não funcionais tal como apresentado na Tabela 4

Tabela 4 - Metas não funcionais

Qualidade	**Meta definida**
Fiabilidade	0 perdas de dados com mecanismo de reenvio
Desempenho	Latência perceptível baixa em operações críticas
Segurança	Controlo de permissões USB e comunicação segura
Usabilidade	Fluxos curtos, *feedback* claro e operação intuitiva

Estes objetivos alinham a solução com os princípios da Computação Móvel e com a estratégia da CapTemp de migrar para a *cloud* o legado, suportando escala, conformidade e valor imediato para o cliente.

Figura 13 - Lançamento automático da aplicação ao detectar o adaptador (filtro intent).

3.5.2. Arquitetura (software/sistema)

A aplicação Android foi concebida para detetar a ligação do dispositivo DS9490R e iniciar automaticamente o fluxo de comunicação com o adaptador 1-Wire. Para isso, a app recorre à *Android USB Host API* [16], com *UsbManager* e um *BroadcastReceiver* para os eventos *DEVICE_ATTACHED* e *DEVICE_DETACHED*, permitindo reagir à ligação e remoção do dispositivo [17]. As operações críticas decorrem num *foreground service*, de forma a evitar a suspensão pelo sistema operativo e a garantir a continuidade da leitura.

A aplicação Android foi desenvolvida para detetar automaticamente a ligação do adaptador DS9490R através da Android USB Host API [16]. Este processo inicia-se com um *UsbManager* configurado com *BroadcastReceiver* para os eventos DEVICE_ATTACHED e DEVICE_DETACHED, permitindo resposta imediata à inserção e remoção do dispositivo [17]. Todas as operações críticas são executadas num *Foreground Service*, garantindo imunidade à suspensão pelo sistema operativo Android e assegurando a continuidade da comunicação.

O fluxo de comunicação USB segue sequência rigorosa:

- Identificação do adaptador pelo par VID/PID específico do DS9490R;
- Solicitação de permissões USB ao utilizador;
- Reclamação da interface USB com abertura dos *endpoints* BULK IN e BULK OUT;
- Manutenção da sessão através de *pings* 1-Wire a cada 30 segundos (*keep-alive*);
- Recuperação automática após *hot-plug*, com revalidação da interface.

Esta abordagem garante sincronização contínua entre aplicação e adaptador, mesmo em cenários de conectividade instável, conforme ilustrado na (Figura 14).

A camada de protocolo 1-Wire encontra-se modularizada em funções especializadas descoberta de dispositivos, leitura/escrita por blocos e transferências com validação CRC16.

A execução é orquestrada por Kotlin *Coroutines* e *Flows*, com a implementação de:

- *Timeouts* de 5 segundos por operação;
- 3 tentativas de *retry* com *backoff* exponencial (100 ms, 250 ms, 500 ms);
- Processamento assíncrono sem bloqueio da interface utilizador.

Cada operação retorna um par envio/resposta, isolando falhas e prevenindo estados inconsistentes.

Para concorrência segura, a leitura organiza-se por porta RJ11 através de fila dedicada por porta com isolamento total de erros. Uma falha na sonda DS1402D-DR8+ numa porta específica não compromete as restantes sondas do sistema.

Finalmente, os dados recolhidos seguem para sincronização *cloud* através do Portal Push sobre HTTPS com confirmação de receção (ACK). Em caso de falha de rede, os registos persistem numa fila local ObjectBox e são reenviados automaticamente com a mesma estratégia de *backoff* exponencial. O *endpoint* de destino (produção ou *staging*) é configurável via *BuildConfig* no momento da compilação.

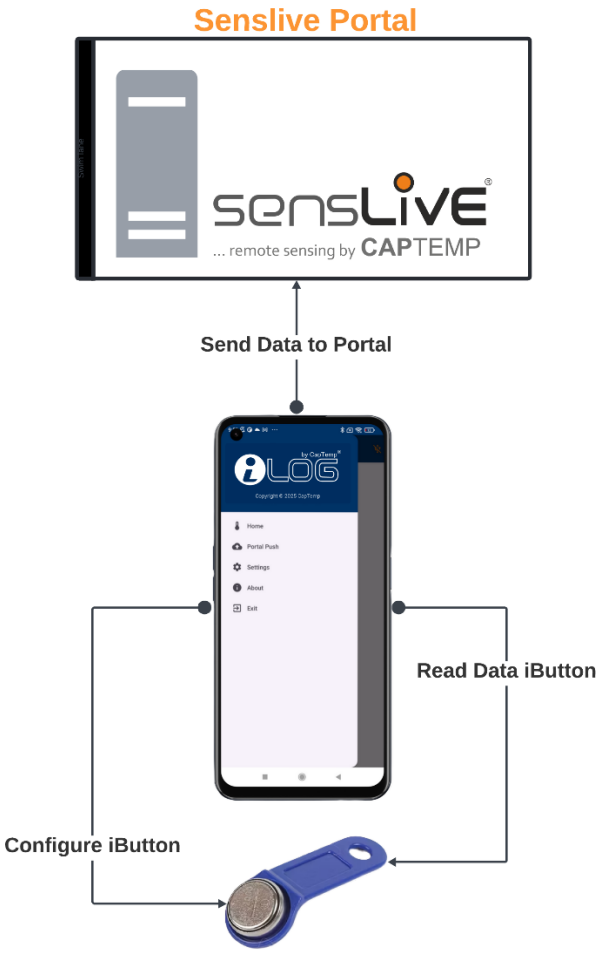


Figura 14 - Arquitetura geral Projeto 2 - iLog (Android, integração cloud)

3.5.3. Integrações de hardware (cadeia física e metadados no dispositivo)

A cadeia física do sistema é composta por quatro elementos principais: o dispositivo Android com suporte *USB-OTG*, o adaptador DS9490R, a sonda DS1402D-DR8+ e o registador DS1921G (*iButton*) [18]. O Android atua como anfitrião (*USB host*) e comunica com o DS9490R através da ligação USB-C/OTG, enquanto o DS9490R faz a ponte entre o universo USB e o barramento 1-Wire. A sonda DS1402D-DR8+ funciona como interface RJ11 para a ligação física dos registadores DS1921G. Em testes realizados, foi validado o uso de *splitter* RJ11 com até quatro registadores em paralelo, devidamente listados na aplicação (ver Figura 18).

O fluxo de utilização foi desenhado para ser simples e reprodutível. Primeiro, o utilizador liga o DS9490R ao dispositivo Android; em seguida, a aplicação é aberta automaticamente, sendo depois possível escolher entre programar ou ler os registadores. Durante a execução, o progresso é apresentado numa barra de estado, suportada por um

serviço em primeiro plano [19], e o resultado é enviado para o *Senslive*. Esta sequência reduz erros operacionais e torna o processo mais previsível em contexto de campo.

Relativamente aos metadados guardados no dispositivo, foi definida uma zona de memória personalizada na *general-purpose* [20] *SRAM* do DS1921G para armazenar informação de controlo e rastreabilidade. Essa estrutura foi organizada em *little-endian* para evitar ambiguidades na interpretação de campos *multi-byte* entre módulos e versões futuras. O bloco inclui, entre outros elementos, a versão do formato, o *epoch* de início da missão, *flags* de controlo como *PE* (*PIN Enable*) e *CE* (*Calibration Enable*), o valor de calibração com sinal e, quando configurado, o PIN de quatro dígitos. O PIN é validado pela aplicação antes de operações sensíveis, como a leitura ou exportação de registos, funcionando como uma barreira operacional adicional contra acessos indevidos.

Nota importante - Os dados armazenados na *general-purpose SRAM* são utilizados exclusivamente pela aplicação desenvolvida. Em particular, o PIN configurado é validado apenas por esse software, funcionando como uma camada de proteção operacional específica da aplicação criada. Caso seja utilizada outra ferramenta para leitura da camada 1-Wire, esse PIN não tem qualquer efeito, uma vez que não corresponde a um mecanismo de proteção nativo do dispositivo DS1921G nem a uma configuração geral do sistema [20].

Esta abordagem assegura compatibilidade futura, verificabilidade dos dados e maior robustez operacional, enquanto preserva a rastreabilidade dos registos associados a cada dispositivo.

Figura 15 - RJ11 splitter com quatro data loggers DS1921G.

3.5.4. Base técnica e migração de código

Como ponto de partida, foi utilizada a aplicação de referência do fabricante, intitulada *"Interface 1-Wire USB Adapter with Android"* [21], originalmente escrita em Java. Esta aplicação foi migrada para Kotlin, removendo APIs obsoletas e reorganizando o código em módulos bem definidos para facilitar manutenção e testes unitários:

- *Device discovery* (detecção automática do DS9490R);
- *Permission flow* (pedido e gestão de permissões USB);
- *Claim/release* da interface do dispositivo;
- *BULK IN/OUT* (transferências emparelhadas);
- Gestão de *endpoints*, *timeouts* e *retries* com estratégia de *backoff*.

Antes de consolidar a arquitetura final, a viabilidade funcional foi validada através de dois projetos de referência. Primeiramente, testou-se uma aplicação desenvolvida por um estudante africano que já implementava operações de leitura e escrita no registador DS1921G através do adaptador DS9490R, utilizando *Android USB Host*. Esta validação confirmou a exequibilidade do percurso tecnológico no ecossistema alvo da CapTemp. Posteriormente, a ferramenta interna da empresa, designada por MONTEMP (também referida como MONTEMP), foi utilizada para cotejar temperaturas instantâneas, registos e parâmetros de missão. As leituras obtidas pela aplicação *iLog* foram comparadas cruzadamente com as do MONTEMP, garantindo consistência de valores e paridade de comportamento nos casos de teste.

Esta dupla validação — técnica (projeto estudante) e operacional (ferramenta interna CapTemp) — assegurou que a solução desenvolvida era tanto viável como compatível com os processos e padrões da empresa, reduzindo riscos de implementação em produção.

3.5.5. Comunicação 1-Wire/USB-C

Nos ensaios iniciais com Android USB *Host* (telefone → DS9490R → DS1921G) foi identificado um comportamento que podia bloquear a pilha USB quando a comunicação não era encerrada corretamente [22]. Em determinados fluxos, era enviado apenas o pedido (*BULK OUT*) sem a leitura correspondente da resposta (*BULK IN*). A repetição desta sequência — por exemplo, ao enumerar dispositivos e a disparar comandos sucessivos sem consumir a resposta do dispositivo — levava o *host* Android a um estado em que as

transferências ficavam em *timeout* e a porta USB-C deixava de aceitar tráfego de dados para esse dispositivo [22].

Para confirmar a origem do problema, foi ligado um osciloscópio ao barramento 1-Wire. Durante o bloqueio, não foram observados pulsos no barramento (ver Figura 16), o que indica que a falha ocorria no lado USB/*host*, sem que a comunicação chegasse ao adaptador. Em paralelo, observou-se que certas políticas de segurança do Android [22] podem suspender temporariamente a comunicação quando são detetados padrões considerados anómalos ou quando existem pedidos repetidos de acesso sem interação do utilizador [23].

Assim, o problema foi interpretado como uma falha de sincronização na camada USB/*host*, causada pela ausência de consumo da resposta após cada comando, e não como uma avaria do adaptador ou do dispositivo 1-Wire. Esta análise permitiu ajustar o fluxo de comunicação para garantir o emparelhamento correto entre pedidos e respostas, evitando novos bloqueios da interface.

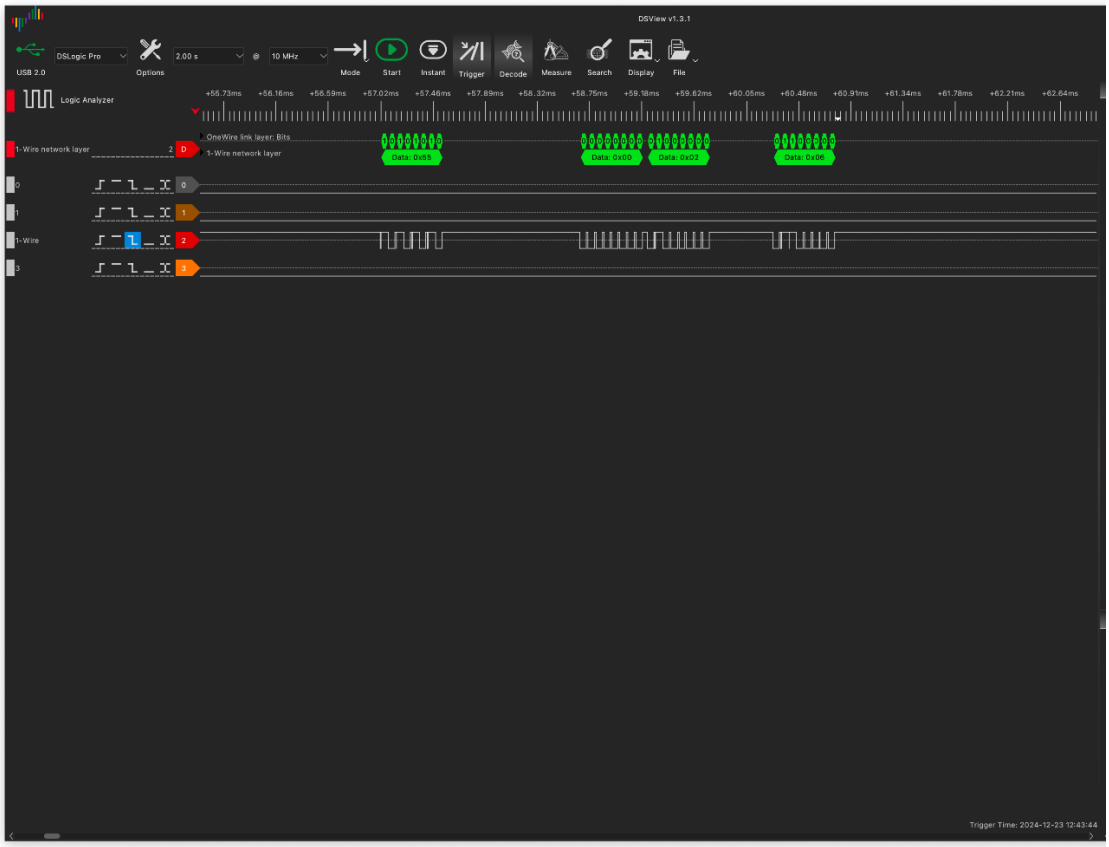


Figura 16 - Osciloscópio a ler o barramento 1-Wire.

As medições com osciloscópio foram realizadas sob a supervisão de um técnico sénior da CapTemp, com vasta experiência em programação de microcontroladores e desenvolvimento de placas eletrónicas (*PCBs*). Esta supervisão garantiu a correção do *setup* experimental e a fiabilidade na leitura dos sinais 1-Wire. Os resultados confirmaram que, durante os bloqueios da porta USB, não havia atividade no barramento 1-Wire, indicando que o problema se localizava na camada USB/*host* e não no adaptador ou no dispositivo.

Medidas de mitigação implementadas:

Para resolver o problema de bloqueio da comunicação, foram adotadas as seguintes medidas técnicas e operacionais:

1. **Pareamento obrigatório OUT/IN** - Toda operação passou a ser obrigatoriamente bidirecional — cada pedido (*BULK OUT*) é sempre seguido da leitura da resposta correspondente (*BULK IN*), evitando *deadlocks* na pilha USB.

2. **Gestão de *timeouts* e *retries*** - Implementados *timeouts* explícitos com estratégia de *backoff* exponencial para todas as transferências.

3. **Mensagens de erro claras** - Em caso de falha, a aplicação apresenta mensagens orientadas ao utilizador, como “Ligação USB bloqueada/perdida” ou “Reconectar o dispositivo” tal como ilustrado na (Figura 17).

4. **Procedimento de recuperação** - Quando o sistema operativo bloqueia a porta USB — cenário reproduzível em testes —, a recuperação efetiva consiste em remover e voltar a ligar o adaptador DS9490R (ciclo de energia), após o qual a comunicação retoma normalmente.

Estas medidas — pareamento obrigatório *OUT/IN*, gestão de *timeouts*, reconexão manual e mensagens orientadas ao utilizador — garantiram a robustez da comunicação em cenários reais de utilização em campo, mesmo com conectividade intermitente ou políticas de segurança ativas do sistema operativo [22].

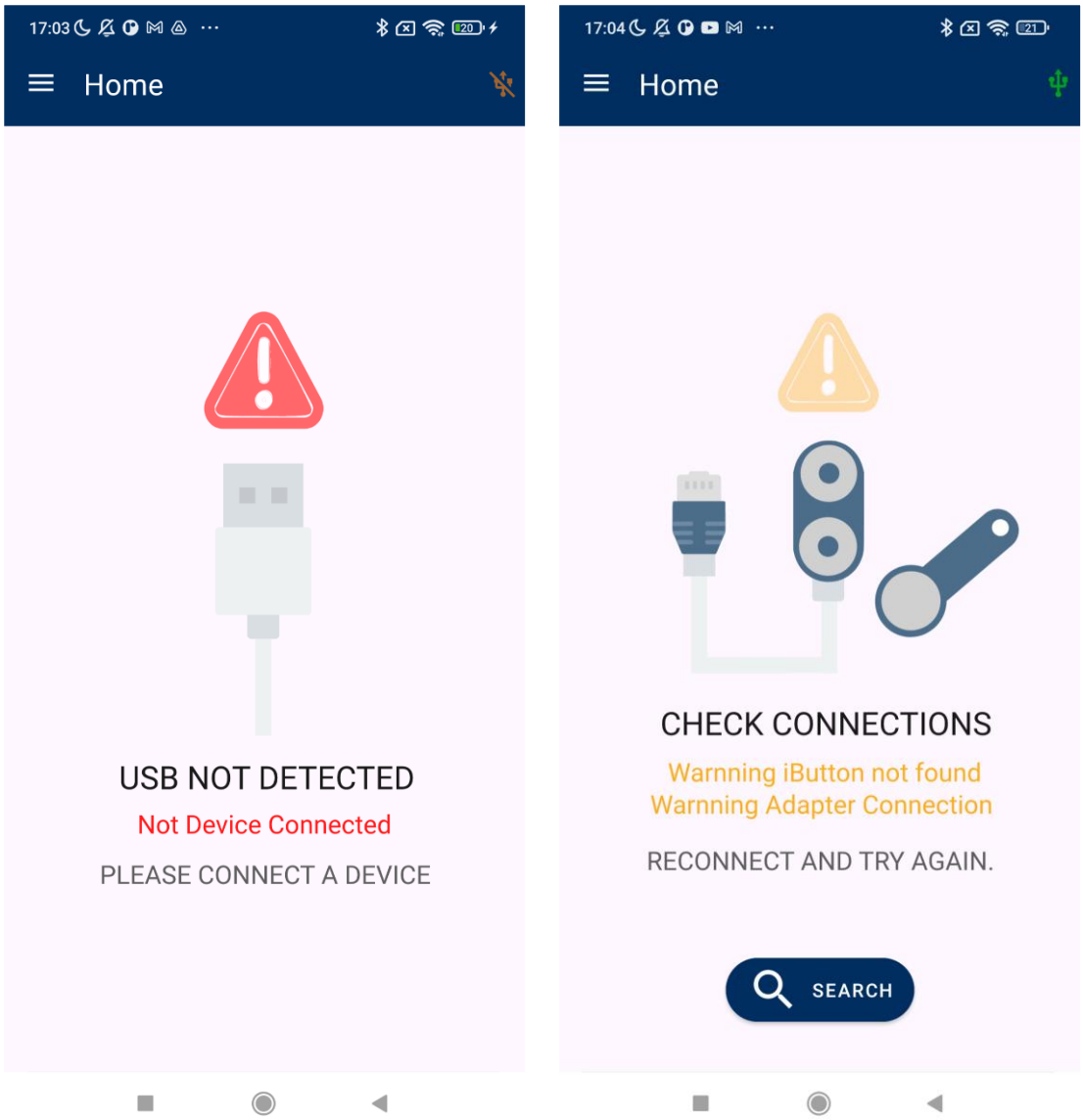


Figura 17 - Tratamento explícito de erros com mensagens claras.

3.5.6. Fluxos principais

Para facilitar a identificação imediata do estado operacional dos registadores DS1921G em contexto de campo e minimizar erros de configuração, foi implementado um sistema de três estados visuais distintos, codificados por cores específicas:

- **Verde** - registador em missão ativa, com todos os parâmetros dentro dos limites definidos;
- **Vermelho** - registador disponível para nova missão (sem missão ativa, memória disponível, parâmetros por definir);

- **Laranja** - missão programada em espera, com contagem decrescente até ao arranque automático.

Os estados são exibidos na interface principal da aplicação quando acionado o botão de pesquisa de dispositivos, conforme ilustrado na (Figura 18). Esta codificação visual, associada à informação temporal, permite ao técnico identificar instantaneamente o estado operacional de cada registador, otimizando as operações de campo e reduzindo significativamente o risco de configuração inadequada.

Figura 18 - Estado operacionais dos dataloggers DS1921G.

Ao aceder a um registador DS1921G através do menu de leitura, a aplicação implementa um mecanismo de autenticação por PIN de 4 dígitos sempre que o dispositivo tenha sido previamente configurado com proteção ativa. Sem a introdução do PIN correto, o utilizador não pode visualizar informação de missão nem proceder à descarga de registos, garantindo a confidencialidade dos dados.

Uma vez autenticado, o ecrã de *Detalhes da Missão* (ver Figura 19) apresenta a seguinte informação para registadores em missão ativa (estado verde):

- Temperatura atual (°C);
- Identificador único do registador (12 dígitos);
- Hora do dispositivo no momento do arranque da missão;
- Data prevista de término (calculada pela capacidade de memória disponível).

Adicionalmente são exibidos os parâmetros configurados da missão: intervalo de amostragem, número de leituras já recolhidas, total de registos em memória (incluindo missões anteriores), estado do *rollover* (se ativo, as leituras mais antigas são sobrescritas ao encher a memória; caso contrário, deixam de ser armazenadas), estado atual da missão (ativa/agendada), limites de temperatura (mínimo/máximo) e *offset* de calibração.

Sempre que este ecrã é aberto, as novas leituras são guardadas automaticamente na memória interna da aplicação, assegurando a consulta posterior mesmo em condições de conectividade intermitente. Esta funcionalidade otimiza a experiência do técnico em campo, combinando segurança, informação completa e resiliência operacional.

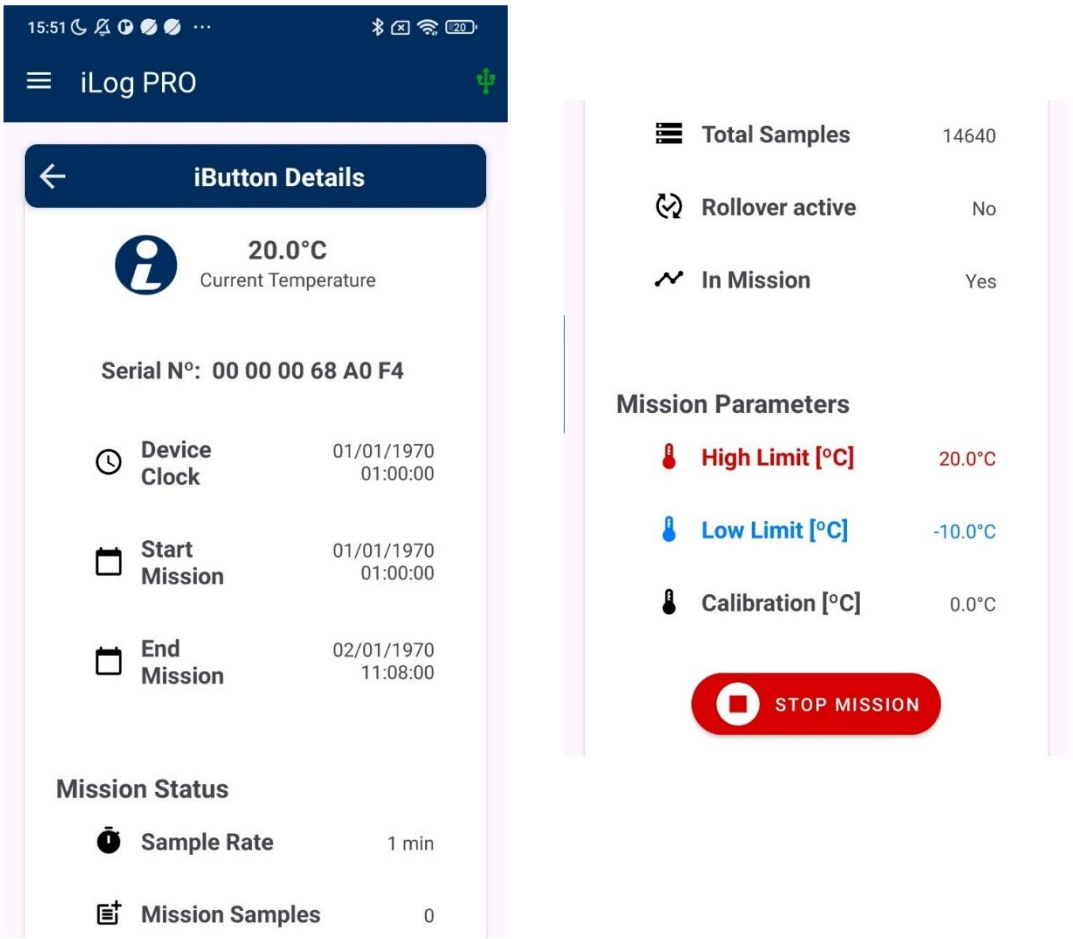


Figura 19 - Detalhes da missão.

A paragem de uma missão ativa requer confirmação explícita do utilizador, implementada através de uma contagem decrescente de 3 segundos tal como ilustrado na (Figura 20). Após a confirmação e término da operação, o estado do registador DS1921G deixa de ser verde (*missão ativa*), passando a vermelho (*disponível para nova missão*).

Para missões agendadas (estado laranja), o utilizador pode rever a configuração antes do início automático. Até à primeira leitura registada, os campos de data exibem valores *placeholder* (época de 1970, Unix *epoch* 0). O utilizador tem a possibilidade de cancelar a missão agendada, revertendo imediatamente o estado para "Sem missão" e permitindo a reprogramação do registador.

Estas funcionalidades garantem controlo total sobre o ciclo de vida da missão, prevenindo paragens acidentais e oferecendo flexibilidade operacional ao técnico em campo.

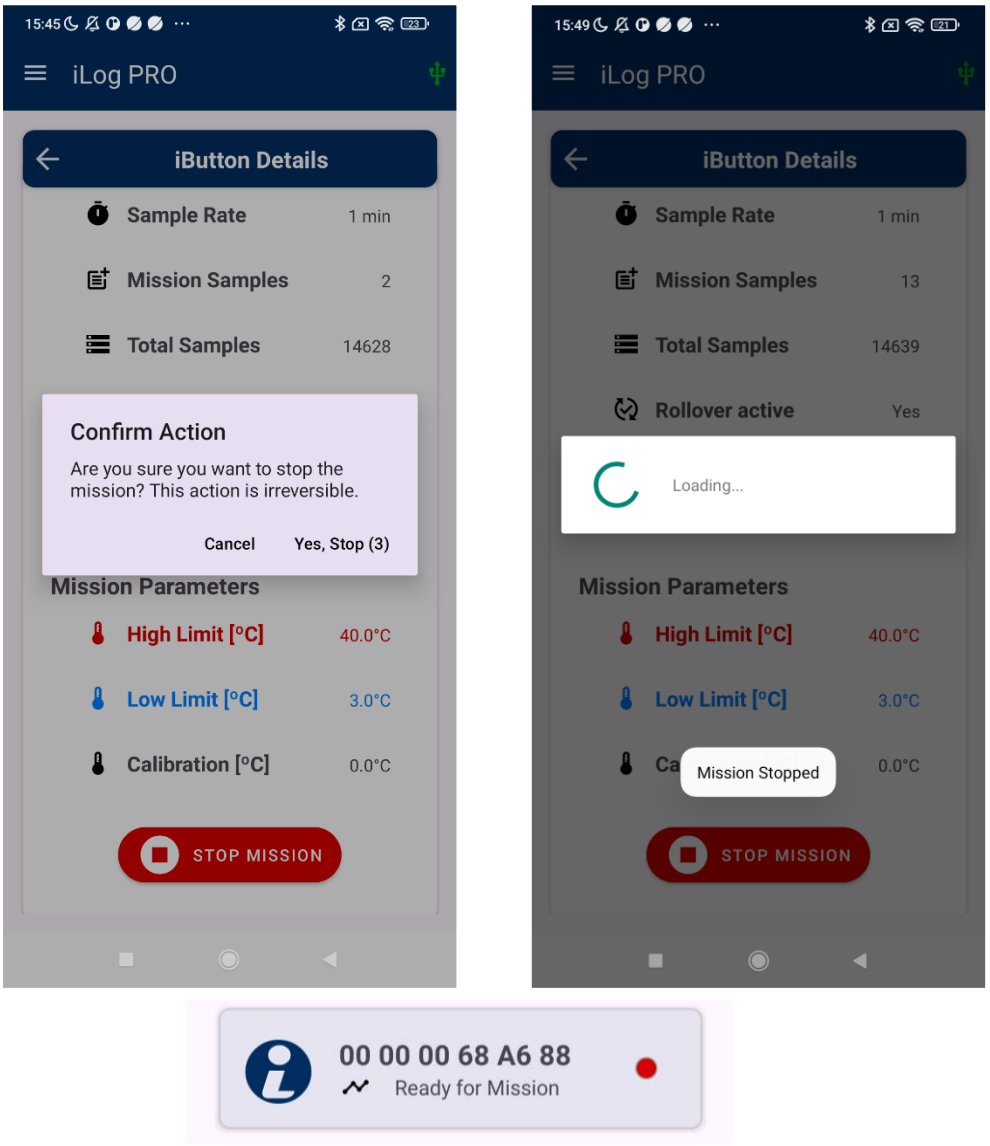


Figura 20 - Paragem de uma missão em curso.

O formulário de programação reúne, de forma guiada e intuitiva, todos os parâmetros necessários para configurar uma missão no registador DS1921G:

- **Nome/descrição** - dados de controlo essenciais para o portal *Senslive*;
- **Data/hora de início** - programável até 40 dias no futuro, com validação de fuso horário GMT0;
- **Intervalo de amostragem** - configurável entre 1 e 255 minutos;
- **Limites de temperatura** - intervalo de −40,0 a 85,0 °C, com passo de 0,5 °C;
- **Offset de calibração** - correção da leitura do sensor;
- **Opção *rollover*** - sobrescrita circular da memória quando cheia;

- **PIN de proteção** - 4 dígitos (0000–9999) para operações sensíveis.

O formulário implementa validações cruzadas automáticas, incluindo: arredondamento dos limites de temperatura para múltiplos de 0,5 °C; verificação que limite mínimo é inferior ao máximo; coerência entre data de início, intervalo de amostragem e capacidade disponível de memória do dispositivo. Estas validações previnem configurações inválidas e garantem a exequibilidade da missão programada (ver Figura 21).

A interface guiada reduz significativamente o risco de erro humano e assegura que todos os registadores sejam configurados de forma consistente e compatível com os requisitos operacionais da CapTemp.

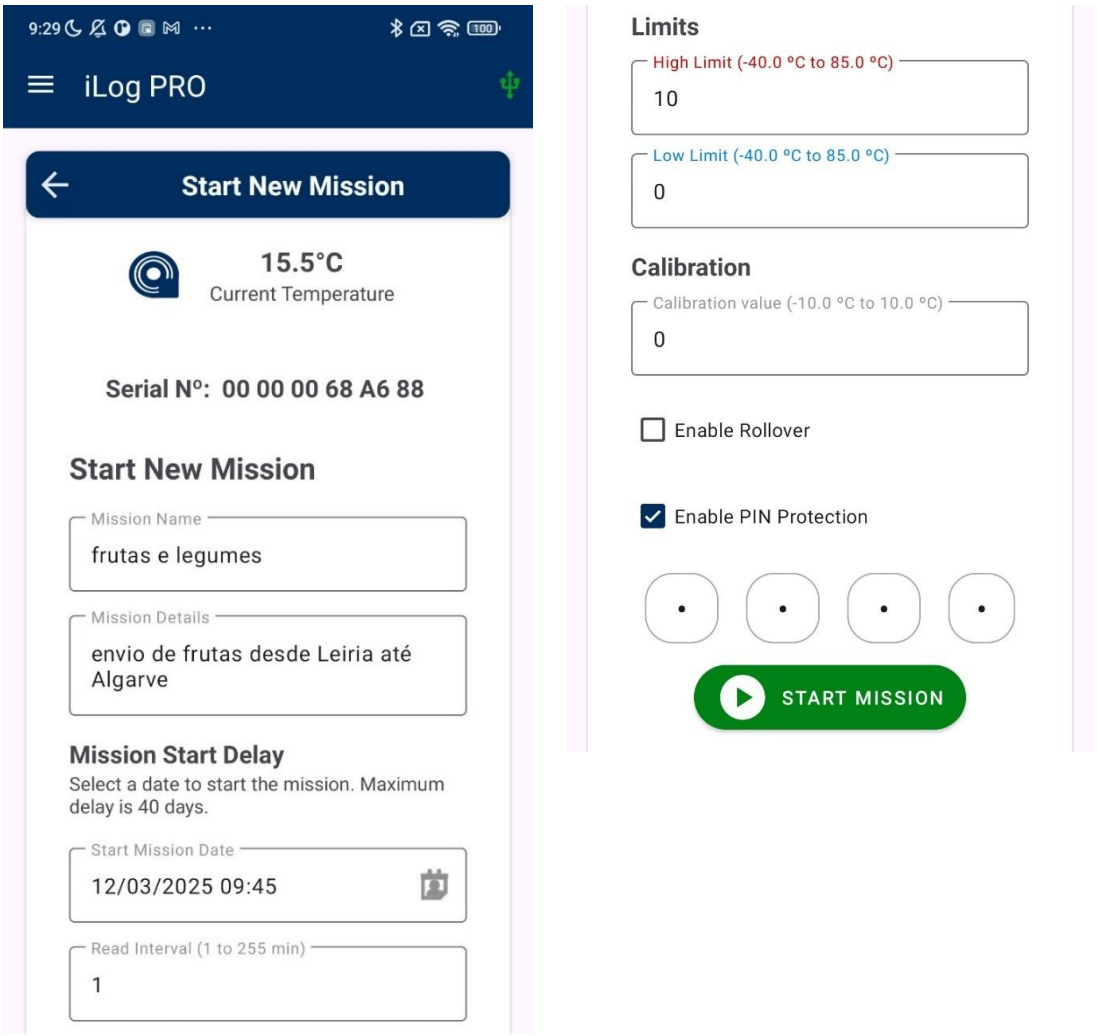


Figura 21 - Formulário de programação de missão.

Antes de proceder à gravação da missão no registador DS1921G, a aplicação apresenta uma pré-visualização do plano de missão que resume todos os parâmetros configurados, a

duração expectável da missão e a hora prevista de término. Esta funcionalidade permite ao utilizador confirmar a configuração de forma informada antes da execução definitiva.

Em caso de erro de validação, são apresentadas mensagens descritivas junto ao campo afetado, acompanhadas de *tooltips* com ajuda contextual. Os dados já introduzidos são preservados, permitindo correção rápida sem perda de informação (ver Figura 22). O processo de gravação só é autorizado na ausência de inconsistências; após sucesso, o utilizador recebe confirmação explícita e o estado do registador é atualizado automaticamente: laranja (missão agendada) se o início for futuro, ou verde (missão ativa) se o arranque for imediato.

Estas salvaguardas — pré-visualização, validação contextual e feedback claro — minimizam erros operacionais e asseguram configurações consistentes e viáveis em contexto de campo.

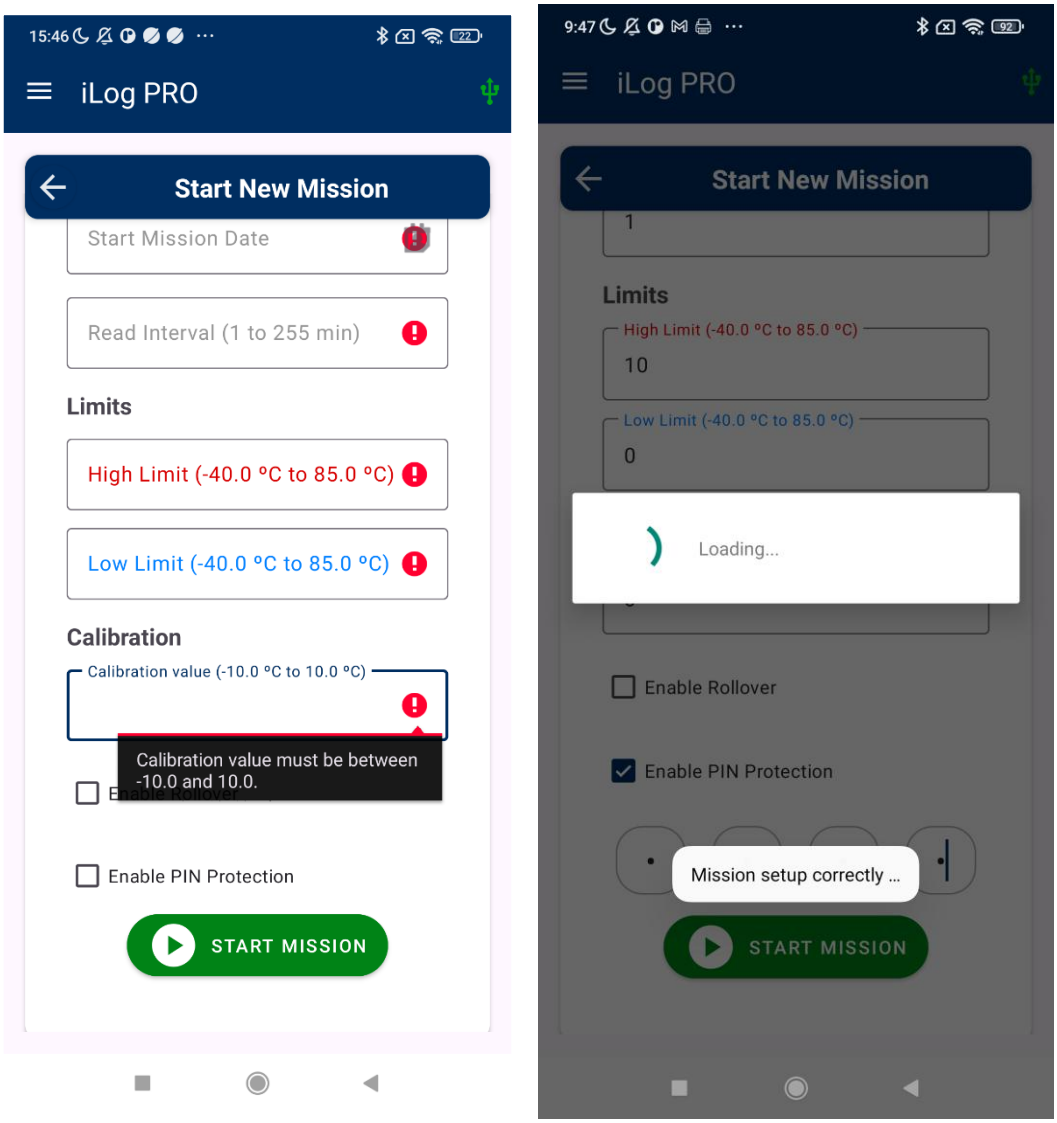


Figura 22 - Mensagens de erro no formulário de programar uma nova missão.

Para validar a integridade das leituras da estrutura implementada na zona de escrita livre *GENERAL-PURPOSE SRAM* (16 *pages*) do DS1921G, foi implementado o algoritmo CRC-16/MODBUS, com valor inicial 0xFFFF, operação *XOR* por byte e processamento *bitwise* com o polinómio 0xA001, aplicando máscara de 16 bits no resultado (ver Figura 23). Este mecanismo permite detetar alterações ou erros de leitura nos dados armazenados, reforçando a fiabilidade da informação processada pela aplicação.

Com base nessa estrutura, foi definido um mapa de memória em *little-endian* para armazenamento de metadados e informação de controlo, assegurando rastreabilidade, verificabilidade e compatibilidade futura entre versões da aplicação. O bloco inclui, entre outros elementos, a versão do formato, o *epoch* de início da missão, *flags* de controlo (*PE* –

PIN Enable; *CE – Calibration Enable*), o valor de calibração com sinal e o PIN de 4 dígitos, quando configurado.

O PIN é armazenado na mesma zona *GENERAL-PURPOSE SRAM* e é validado exclusivamente pela aplicação desenvolvida, antes da execução de operações sensíveis, como a leitura ou a exportação de registos. Importa esclarecer que este PIN não corresponde a um mecanismo de proteção nativo do DS1921G nem coloca o dispositivo, por si só, num modo protegido ao nível da camada 1-Wire. Assim, caso seja utilizada outra ferramenta para aceder diretamente ao dispositivo, o PIN configurado não produz qualquer efeito fora do contexto da aplicação criada. Deste modo, esta funcionalidade deve ser entendida como uma camada adicional de proteção operacional, específica do software desenvolvido, e não como um mecanismo de segurança intrínseco ao hardware.

A imagem apresentada (Figura 23) corresponde a um diagrama de fluxo que descreve o funcionamento do mecanismo de validação por CRC [24] aplicado aos dados armazenados na zona de escrita livre (*GENERAL-PURPOSE SRAM*) do DS1921G. Este mecanismo é utilizado exclusivamente para os dados personalizados da aplicação desenvolvida, com o objetivo de garantir a integridade da informação armazenada e detetar eventuais corrupções durante os processos de escrita e leitura.

No processo de escrita, a aplicação começa por construir um *array* de bytes com os novos dados que o utilizador pretende guardar nessa área de memória. Em seguida, é calculado o valor CRC para esse conjunto de dados, com base nos parâmetros previamente definidos. Após esse cálculo, os dados e o respetivo CRC são gravados em conjunto na memória do dispositivo.

No processo de leitura, a aplicação recupera da memória os dados armazenados e o valor CRC associado. Posteriormente, separa o *array* de bytes do CRC guardado e calcula novamente o CRC a partir dos dados lidos. Por fim, compara o valor CRC recém-calculado com o valor CRC armazenado na memória. Se ambos coincidirem, os dados são considerados válidos; caso contrário, conclui-se que os dados se encontram corrompidos.

Importa salientar que este procedimento é aplicado apenas à área de memória personalizada utilizada pela aplicação, não interferindo com as restantes estruturas internas do DS1921G. Desta forma, assegura-se que os metadados e parâmetros adicionais

introduzidos pela solução desenvolvida sejam verificados de forma consistente antes de serem utilizados pela aplicação.

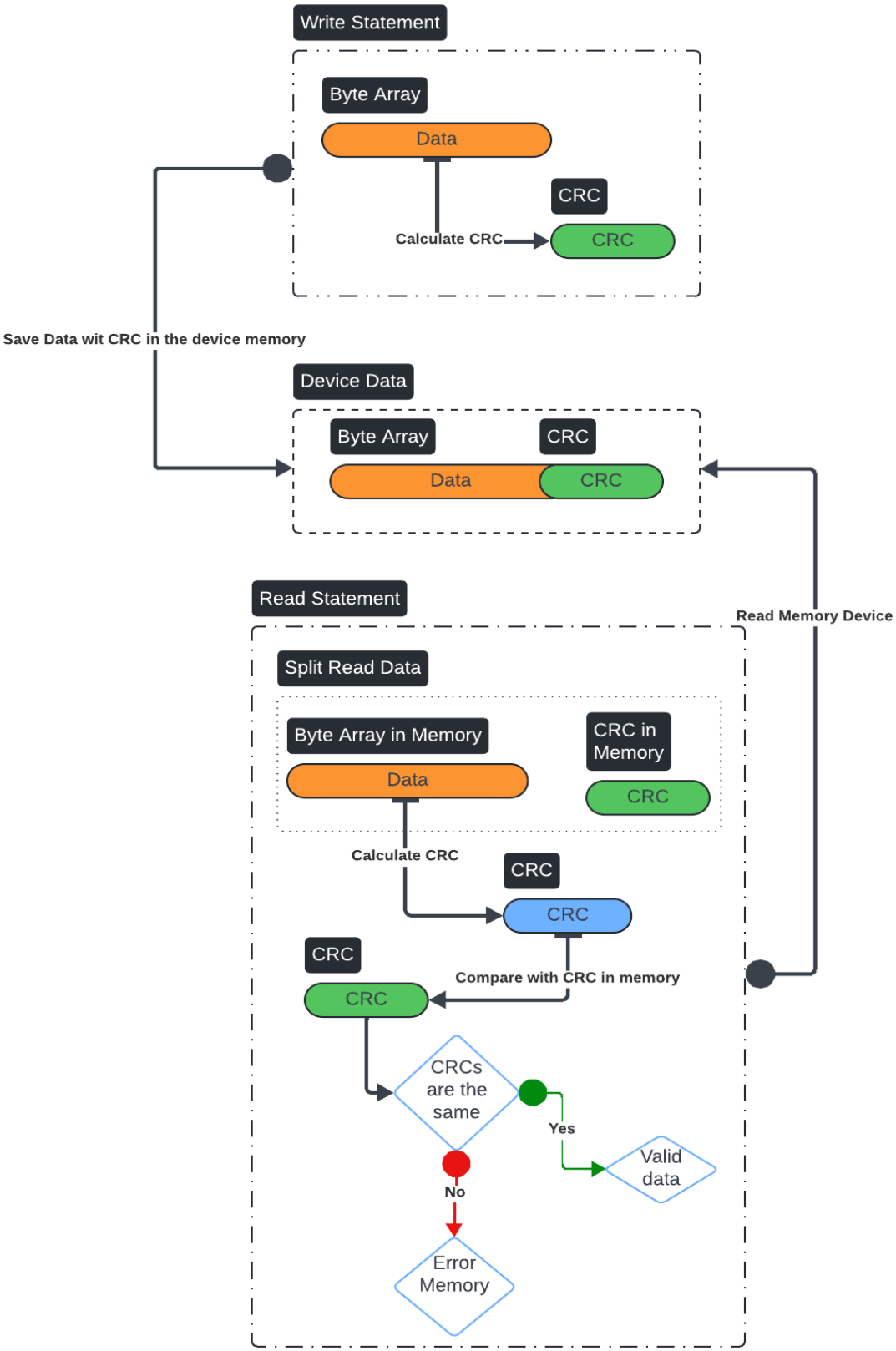


Figura 23 - Validação de integridade de dados em leituras da estrutura implementada.

Cada envio para o portal *Senslive* inclui o identificador único ROM (12 dígitos), *timestamp*, parâmetros da missão configurada e o bloco de amostras recolhidas. O *endpoint* do *Senslive* devolve um ACK explícito para confirmar a receção; caso o ACK não seja recebido ou contenha erro, as leituras permanecem numa fila local persistente, sendo reenviadas automaticamente na próxima ligação à Internet ou manualmente pelo utilizador quando solicitado.

Para configurar esta funcionalidade, o utilizador deve definir previamente, no menu de Definições, o URL do *endpoint* (ambiente de produção ou ensaio) e a frequência de envio automático (a partir de 1 minuto), conforme ilustrado na (Figura 25). Quando pretende forçar o envio manual, o ecrã *Portal Push* (ver Figura 24) apresenta informação operacional relevante:

- Hora do último envio;
- Número de registos guardados localmente;
- Número de missões iniciadas e terminadas no dispositivo e disponibiliza ações como "*Send Data*", além de uma área dedicada à visualização da resposta do servidor.

Esta estratégia *offline-first* garante que os dados recolhidos sejam sempre preservados e enviados, mesmo em condições de conectividade intermitente, assegurando a continuidade operacional em contexto de campo.

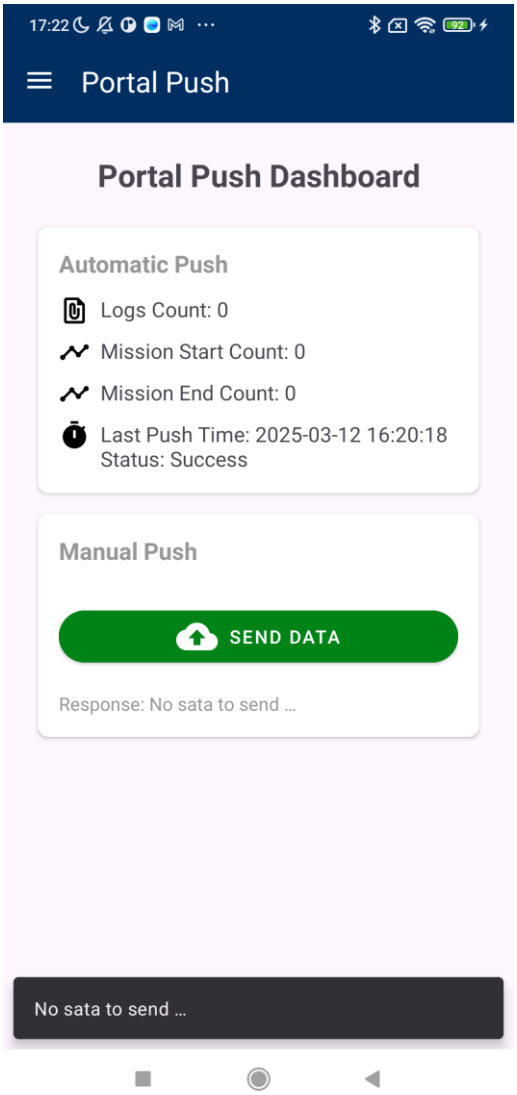


Figura 24 - Menu para envios manuais ao portal *Senslive*.

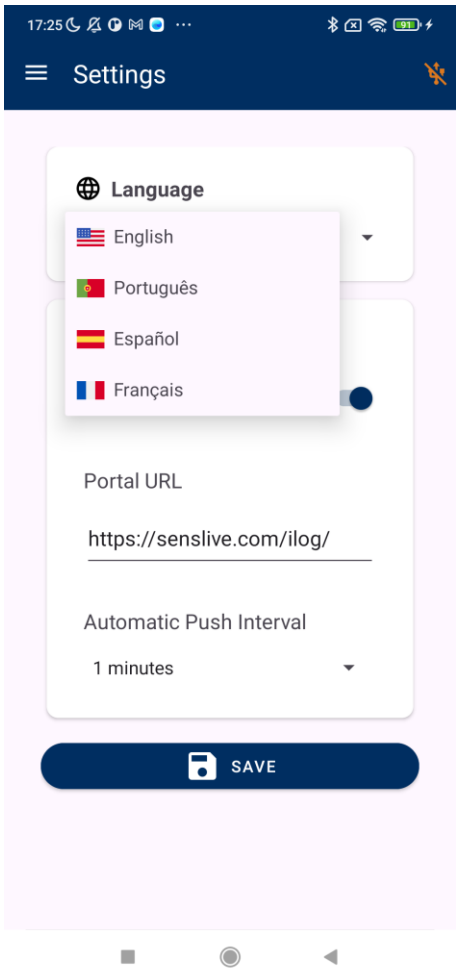


Figura 25 - Menu Configurações para o envio automático.

Nos protótipos iniciais, foi utilizado o Adobe XD para criar protótipos de alta-fidelidade que representavam a navegação geral da aplicação, os ecrãs principais e os diferentes estados operacionais. Estas maquetes funcionaram como ponto de partida para discutir requisitos funcionais e interações com o supervisor, permitindo alinhar expectativas e identificar potenciais melhorias desde uma fase inicial do desenvolvimento (ver Figura 26).

Ao longo do desenvolvimento, os *mockups* foram ajustados de forma iterativa, incorporando o *feedback* do supervisor e as restrições técnicas identificadas, nomeadamente a comunicação *USB Host*, a sincronização *offline-first* e a gestão de estados do registador. Este processo culminou na estabilização dos protótipos finais, que refletem tanto o aspeto visual como o comportamento funcional previsto para a versão de produção.

Na implementação final, foram privilegiados fluxos curtos e intuitivos, complementados por mensagens claras de erro e sucesso. A descoberta de dispositivos é representada

visualmente através de indicadores de ligação e deteção, enquanto os formulários incorporam validações que impedem configurações inválidas, como limites de temperatura incoerentes, intervalos de amostragem inadequados ou inconsistências temporais. Os estados operacionais do registador são codificados por cores para identificação imediata: verde (*em missão*), vermelho (*pronto para programar*) e laranja (*missão agendada com contagem decrescente*).

Esta abordagem centrada no utilizador assegura que a aplicação seja intuitiva para técnicos em contexto de campo, mesmo sem formação específica, minimizando erros operacionais e otimizando a eficiência do processo.

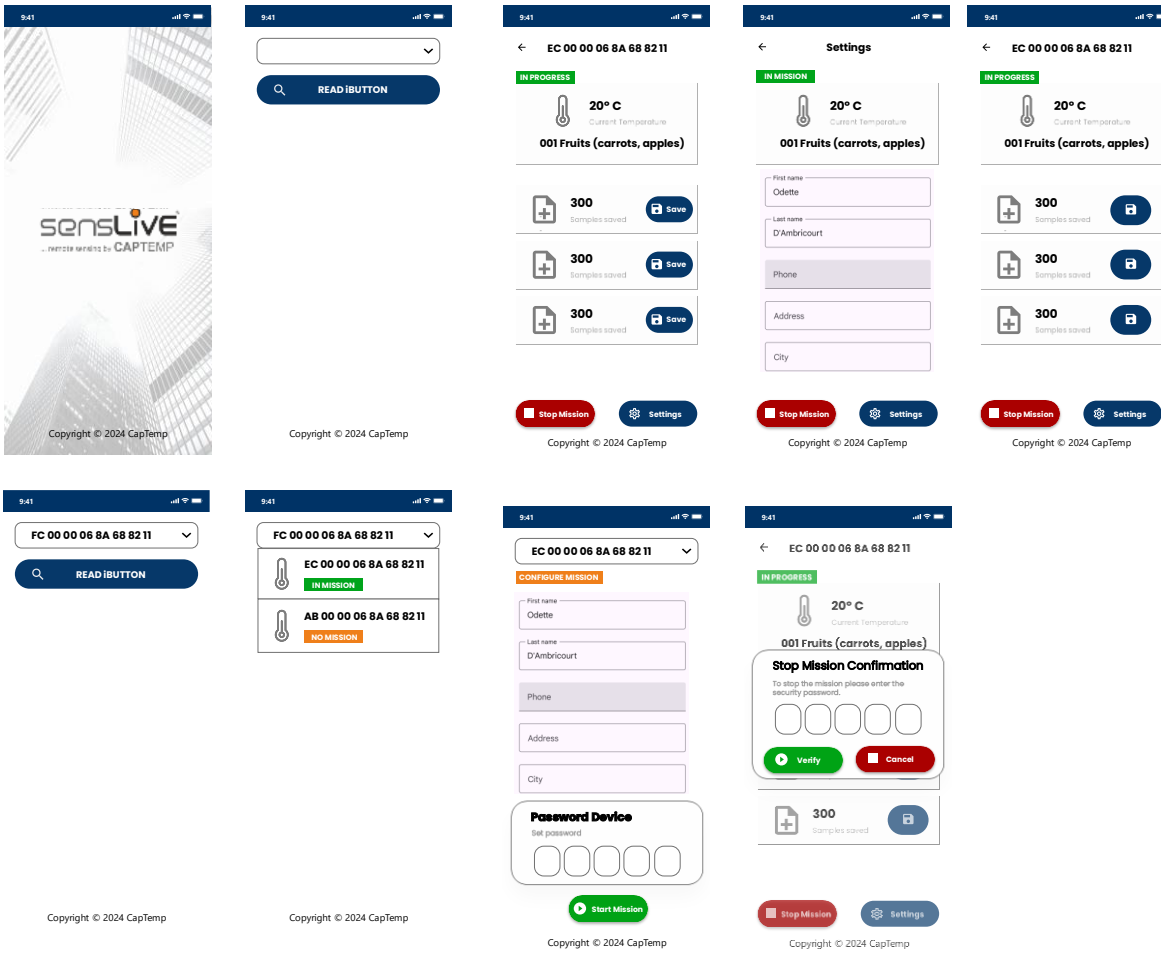


Figura 26 - Protótipos iniciais para aplicação iLog.

Ensaios funcionais e avaliação de desempenho

Em dispositivos Android 10 e superiores, foram realizados ensaios que demonstraram a fiabilidade do auto-arranque da aplicação ao ligar o adaptador DS9490R (através de filtro *VID/PID*), a consistência da programação e leitura do registador DS1921G e a estabilidade da sincronização com o portal *Senslive*, mesmo em condições de conectividade intermitente. Com o *splitter* RJ11, foram obtidos ganhos significativos de *throughput* em rotinas de bancada, permitindo processar vários registadores em paralelo com isolamento completo de erros por porta.

Foram documentados casos de teste abrangentes que validaram os seguintes cenários:

- Missão válida (programação, execução e leitura completas);
- Missão fora dos limites de temperatura configurados;
- PIN inválido (bloqueio de acesso às operações sensíveis);
- *Rollover* ligado e desligado;
- Perda de conectividade durante envio, com reconexão automática;
- *Timeouts* USB em diferentes condições operacionais.

Limitações identificadas e matriz de compatibilidade

Em alguns modelos de dispositivos Android, políticas de otimização de energia mais agressivas podem interferir em operações prolongadas, exigindo a configuração de exceções à otimização de bateria para assegurar a continuidade da comunicação e da leitura. Durante os ensaios, foram identificadas diferenças de comportamento entre dispositivos testados, o que levou ao ajuste de tempos de *timeout* e à definição de procedimentos de recuperação adequados a determinados *chipsets* e versões do sistema operativo. A implementação para iOS permaneceu em fase de estudo, devido às limitações atualmente impostas pelo *stack* USB-C da plataforma. Esta avaliação permitiu identificar e mitigar os principais constrangimentos operacionais, reforçando a robustez da solução em contexto de produção.

3.5.7. Documentação e manutenção

Foram elaborados os seguintes artefactos documentais para suportar a implementação, formação de equipas e manutenção da solução:

- *Manual do utilizador* e *guia técnico* (instalação, configuração de permissões, ligações físicas, resolução de problemas, com capturas de ecrã ilustrativas);
- Diagramas de arquitetura da solução;
- Registo de decisões arquitetónicas (*Architecture Decision Records* – ADR), que documentam as principais opções consideradas, as decisões adotadas e as respetivas implicações técnicas.

Estes materiais facilitam a formação de novos elementos da equipa, reduzem a carga de suporte técnico e asseguram a rastreabilidade das decisões arquitetónicas ao longo do ciclo de vida da aplicação (ver Anexo A).

3.5.8. Impacto e valor para a CapTemp

A solução iLog uniformiza o *pipeline* de recolha de dados dos registadores DS1921G, substituindo gradualmente a ferramenta interna MONTEMP e reduzindo significativamente o *lead time* de processamento. Ao eliminar passos manuais e centralizar a informação no portal *Senslive*, cria uma fonte única de verdade que suporta a geração automática de gráficos, alertas, relatórios e documentação para auditoria.

Em articulação com a componente iLog *PRO* projeto 3 (destinada a cenários *offline*), a solução permite escalabilidade operacional e controlo de qualidade aprimorado, abrindo caminho a contratos com exigências elevadas de conformidade metrológica e facilitando a oferta de serviços geridos baseados em dados centralizados e verificados.

3.6. Projeto 3 — *iLog PRO* (Android, exportação local e relatório)

O iLog *PRO* constitui uma variante especializada do projeto 2 iLog, desenhada para cenários onde o envio imediato de dados para a *cloud Senslive* não é viável ou desejável. Mantendo o núcleo funcional de programação e leitura dos registadores DS1921G, esta versão introduz capacidades de exportação local para Excel (*.xlsx*) e geração autónoma de relatórios gráficos e parametrizados, suportando operações em contextos de conectividade intermitente, políticas restritivas de segurança ou necessidade de partilha rápida no terreno. A (Figura 27) ilustra a arquitetura diferenciada desta solução, destacando o processamento local em substituição do *Portal Push*.

3.6.1. Motivação e objetivos

O desenvolvimento do iLog PRO responde a três constrangimentos operacionais críticos identificados na CapTemp:

- Conectividade intermitente em locais remotos sem acesso estável à *cloud*;
- Independência da plataforma *cloud Senslive* em ocasiões específicas onde a dependência da infraestrutura externa não é desejável;
- Restrições de segurança/privacidade que impedem transmissão de dados sensíveis;
- Exigência de partilha imediata de relatórios com equipas de campo ou clientes on-site.

O objetivo foi preservar integralmente as funcionalidades centrais do iLog (detecção USB, programação/leitura de missões, validações de integridade), redirecionando o fluxo de dados: em vez de *push* para o portal *Senslive*, os dados são processados localmente, filtrados pelo utilizador e exportados em formato Excel, pronto para análise ou partilha direta.

3.6.2. Arquitetura e fluxos (diferenças face ao iLog)

Reutilizei a base técnica do iLog (Android USB Host, detecção do *DS9490R* e operação com *DS1921G*),

Ambas as aplicações reutilizam a mesma *stack* técnica:

- **Persistência**: *ObjectBox* para armazenamento local transacional dos registos DS1921G;

- **Comunicação**: *Android USB Host* com filtro *VID/PID* para detecção automática do DS9490R;
- **Validações**: verificação de integridade (*CRC*, *bitmap* de metadados) e coerência temporal.

As principais diferenças implementadas entre os projetos encontram-se resumidas na Tabela 5 - Diferenças entre as implementações dos projetos iLog e iLog PRO

Tabela 5 - Diferenças entre as implementações dos projetos iLog e iLog PRO

Componente	iLog	iLog PRO
Saída de dados	Sincronização automática Senslive	Exportação Excel + gráficos locais
Dependência de rede	Obrigatória para envio	Nenhuma (100% offline)
Fluxo pós-leitura	*Push* para portal	Geração de relatório parametrizável

O controlo por PIN mantém-se idêntico: se o registador DS1921G estiver protegido, a exportação de detalhes exige autenticação, garantindo simetria de segurança e experiência de utilizador entre ambas as variantes. Esta arquitetura dual assegura flexibilidade operacional total para a CapTemp em todos os contextos.

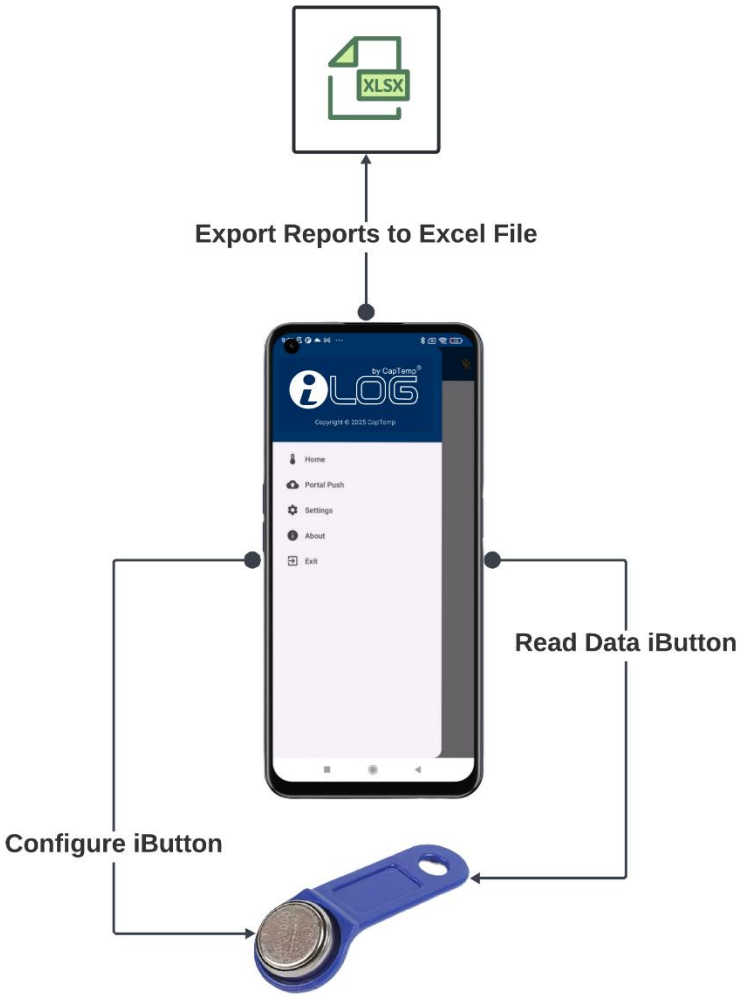


Figura 27 - Arquitetura iLog PRO.

3.6.3. Menu Reports — visão geral e agrupamento

O menu *Reports* apresenta uma vista abrangente de todas as missões guardadas no dispositivo, agrupadas por *logger* (identificador ROM ID de 12 dígitos). Esta organização facilita a navegação por histórico, reduz duplicação e oferece contexto temporal por dispositivo (várias missões, estados distintos). A listagem é paginada e responde a filtros, permitindo encontrar rapidamente a missão de interesse sem percorrer a totalidade do histórico (ver Figura 28).

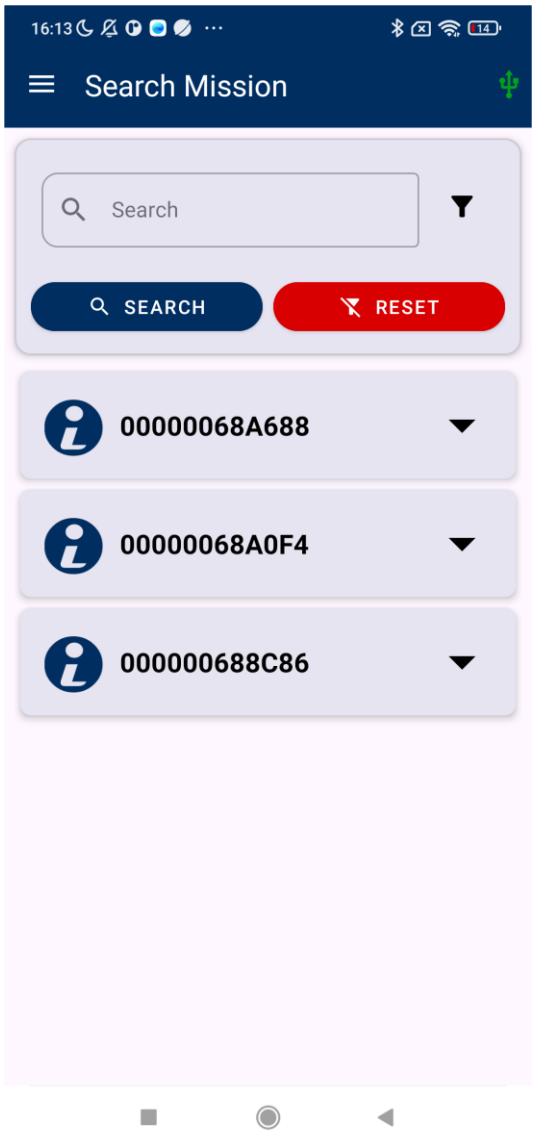


Figura 28- Missões guardadas no dispositivo, agrupadas por datalogger.

3.6.3. Pesquisa e filtros (ID/nome/descrição e intervalo temporal)

A recuperação de informação foi otimizada através de pesquisa textual realizada nos seguintes campos principais: Identificador (12 dígitos hexadecimais do registador DS1921G), Nome da missão (texto atribuído pelo técnico) e Descrição da missão (notas operacionais). Esta funcionalidade opera em tempo real durante a digitação, proporcionando resultados imediatos e reduzindo significativamente o tempo de triagem.

Complementarmente, foi implementado um filtro temporal por intervalo de datas (início/fim) que restringe a listagem às missões do período relevante. A interação é

simplificada através de duplo toque na barra de filtros, que abre o seletor de datas (*date picker*), facilitando a operação em contexto de campo.

O fluxo é completado por dois botões de ação na barra superior: ***SEARCH*** (aplica todos os filtros ativos e atualiza a listagem) e ***RESET*** (restaura a listagem completa sem filtros). Estes controlos, ilustrados na (Figura 29), previnem erros de seleção e asseguram eficiência operacional na gestão de missões.

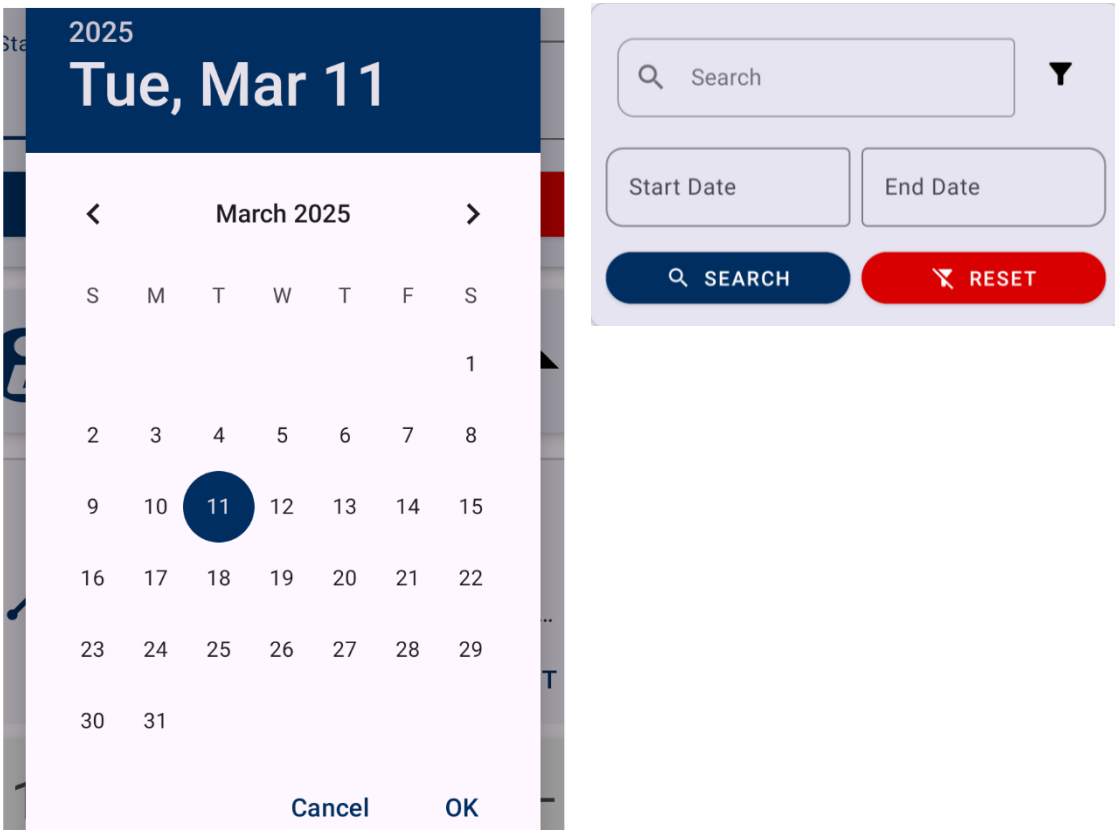


Figura 29 - Pesquisa e filtros (ID/nome/descrição e data)

3.6.4. Exportação de missões para .xlsx

A exportação foi estruturada segundo um fluxo guiado e seguro, de modo a garantir clareza operacional e a reduzir a probabilidade de erro durante a interação. O processo inicia-se com a localização da missão pretendida através dos mecanismos de pesquisa e filtragem previamente descritos. De seguida, o utilizador seleciona a opção *Export*, sendo então apresentado um diálogo de confirmação antes de prosseguir com a operação.

Após a confirmação, a aplicação abre o gestor de ficheiros do dispositivo, permitindo ao utilizador escolher explicitamente a pasta de destino e o nome do ficheiro a gravar.

Concluído o processo de gravação, a aplicação disponibiliza, de forma opcional, mecanismos de partilha direta do ficheiro exportado, nomeadamente por correio eletrónico, serviços de *cloud storage* ou aplicações de mensagens.

Quando a missão já se encontra terminada, é apresentado um pedido adicional de confirmação para remoção dos dados locais associados, com o objetivo de libertar espaço de armazenamento no dispositivo. Pelo contrário, quando a missão ainda está em execução, essa eliminação não é permitida, assegurando a continuidade da recolha de dados e evitando a perda prematura de informação. A (Figura 30) ilustra este fluxo de exportação e as respetivas salvaguardas operacionais.

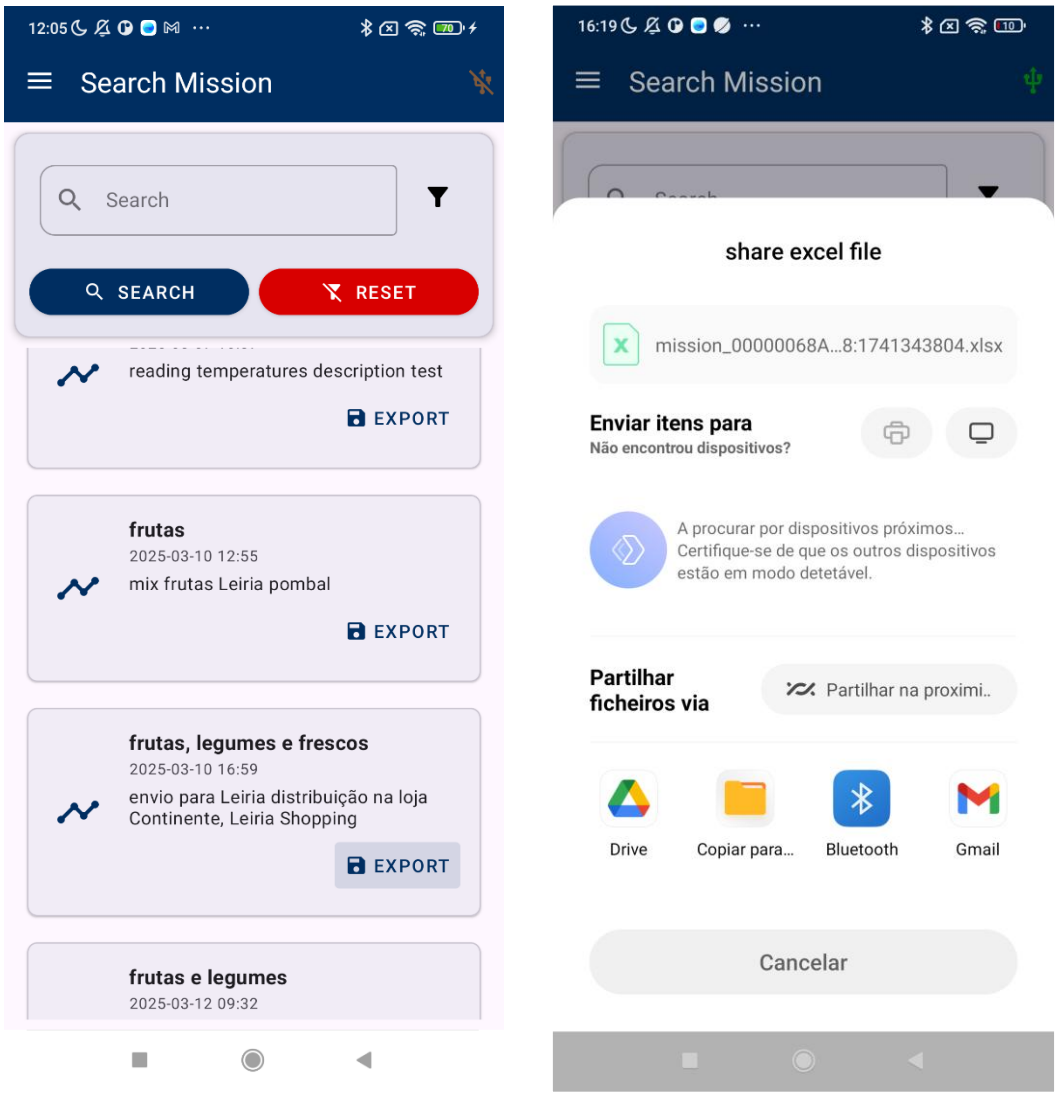


Figura 30 - Exportação de missões para .xlsx.

3.6.5. Estrutura do relatório Excel

O ficheiro Excel gerado apresenta duas secções funcionais distintas: (i) parâmetros de missão e (ii) dados e análise visual. Esta organização assegura rastreabilidade técnica e conformidade metrológica.

Secção 1 — Parâmetros de missão

Contém as configurações técnicas da missão, incluindo:

- **Intervalo de amostragem** (segundos);
- **Data/hora de início e fim** (formato DD/MM/YYYY HH:MM, N/A para missões ativas);
- **Limites operativo** (temperatura mínima/máxima, °C);
- **Offset de calibração** aplicado (±0.1°C);
- **Estado do *rollover*** (Ativo/Inativo).

Esta informação fornece **contexto técnico-jurídico** essencial para auditoria e validação dos dados.

Secção 2 — Dados e análise visual

Tabela de leituras: *Timestamp* | Temperatura (°C) com codificação por cor condicional: estas codificações encontram-se resumidas na Tabela 6 - codificação por cor do ficheiro Excel.

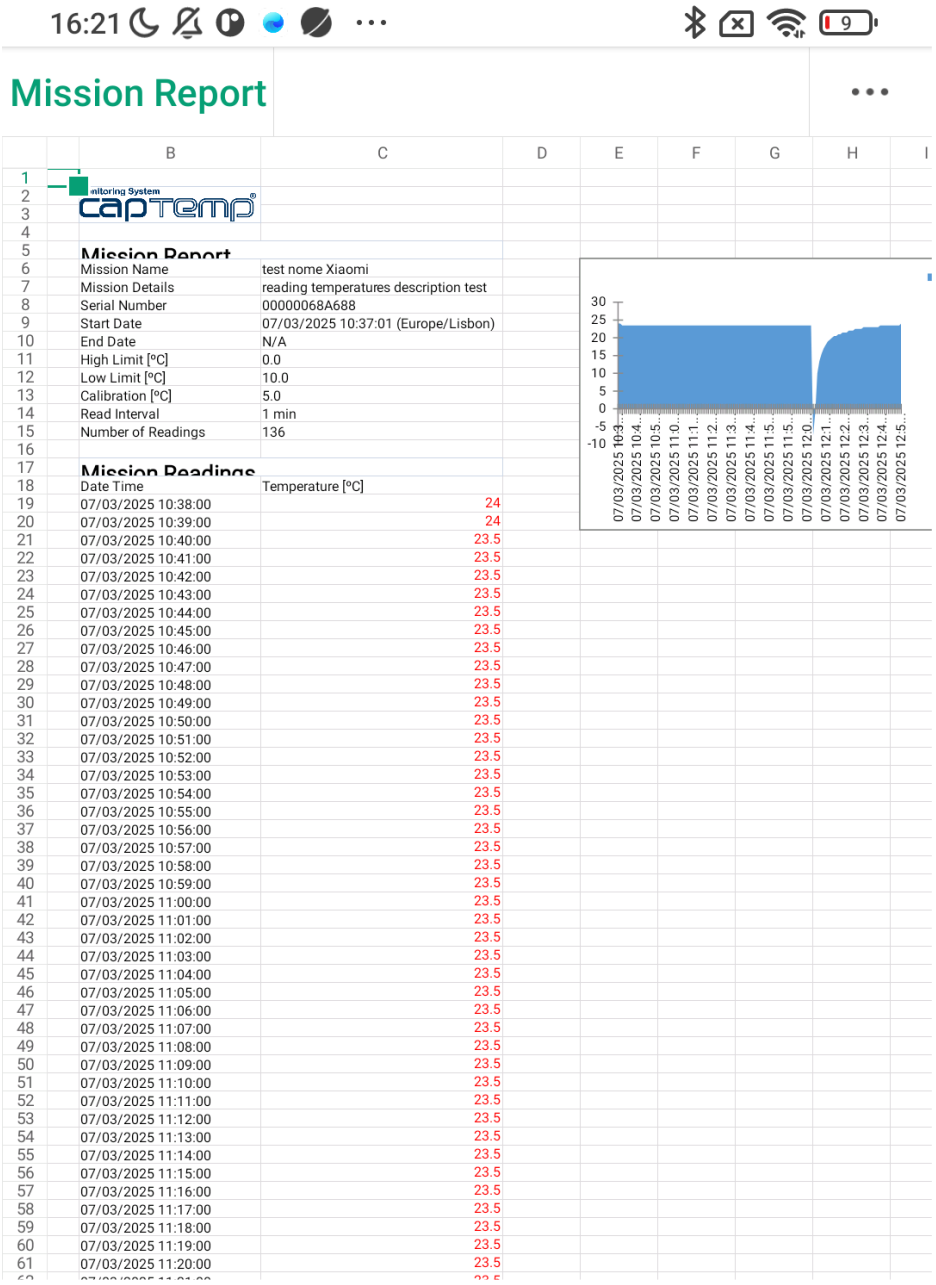
Tabela 6 - codificação por cor do ficheiro Excel.

Estado	Cor	Critério
CONFORME	**Preto**	Dentro dos limites configurados
CRÍTICO	**Vermelho**	Acima do limite máximo
SUBCRÍTICO	**Azul**	Abaixo do limite mínimo

O gráfico temporal de linhas (eixo X: tempo; eixo Y: temperatura °C) apresenta as seguintes funcionalidades:

- Suporta missões ativas (dados parciais);
- Permite zoom interativo para análise detalhada;
- Inclui bandas de limites (mínimo /máximo) para avaliação visual rápida.

A (Figura 31) exemplifica a estrutura completa do ficheiro com os três elementos integrados e a respetiva codificação visual.



Mission Report

	B	C
1		
2		
3		
4		
5	Mission Report	
6	Mission Name	test nome Xiaomi
7	Mission Details	reading temperatures description test
8	Serial Number	00000068A688
9	Start Date	07/03/2025 10:37:01 (Europe/Lisbon)
10	End Date	N/A
11	High Limit [ºC]	0.0
12	Low Limit [ºC]	10.0
13	Calibration [ºC]	5.0
14	Read Interval	1 min
15	Number of Readings	136
16		
17	Mission Readings	
18	Date Time	Temperature [ºC]
19	07/03/2025 10:38:00	24
20	07/03/2025 10:39:00	24
21	07/03/2025 10:40:00	23.5
22	07/03/2025 10:41:00	23.5
23	07/03/2025 10:42:00	23.5
24	07/03/2025 10:43:00	23.5
25	07/03/2025 10:44:00	23.5
26	07/03/2025 10:45:00	23.5
27	07/03/2025 10:46:00	23.5
28	07/03/2025 10:47:00	23.5
29	07/03/2025 10:48:00	23.5
30	07/03/2025 10:49:00	23.5
31	07/03/2025 10:50:00	23.5
32	07/03/2025 10:51:00	23.5
33	07/03/2025 10:52:00	23.5
34	07/03/2025 10:53:00	23.5
35	07/03/2025 10:54:00	23.5
36	07/03/2025 10:55:00	23.5
37	07/03/2025 10:56:00	23.5
38	07/03/2025 10:57:00	23.5
39	07/03/2025 10:58:00	23.5
40	07/03/2025 10:59:00	23.5
41	07/03/2025 11:00:00	23.5
42	07/03/2025 11:01:00	23.5
43	07/03/2025 11:02:00	23.5
44	07/03/2025 11:03:00	23.5
45	07/03/2025 11:04:00	23.5
46	07/03/2025 11:05:00	23.5
47	07/03/2025 11:06:00	23.5
48	07/03/2025 11:07:00	23.5
49	07/03/2025 11:08:00	23.5
50	07/03/2025 11:09:00	23.5
51	07/03/2025 11:10:00	23.5
52	07/03/2025 11:11:00	23.5
53	07/03/2025 11:12:00	23.5
54	07/03/2025 11:13:00	23.5
55	07/03/2025 11:14:00	23.5
56	07/03/2025 11:15:00	23.5
57	07/03/2025 11:16:00	23.5
58	07/03/2025 11:17:00	23.5
59	07/03/2025 11:18:00	23.5
60	07/03/2025 11:19:00	23.5
61	07/03/2025 11:20:00	23.5

110		07/03/2025 12:09:00	23.5
111		07/03/2025 12:10:00	23.5
112		07/03/2025 12:11:00	-7.5
113		07/03/2025 12:12:00	-1
114		07/03/2025 12:13:00	10
115		07/03/2025 12:14:00	13.5

Figura 31 - Estrutura do relatório Excel.

3.6.6. Partilha e controlo do ciclo de vida

Após conclusão da exportação, é apresentado um diálogo de partilha opcional que permite o envio imediato do ficheiro através dos canais disponíveis no dispositivo: correio eletrónico, serviços de *cloud storage* ou aplicações de mensagens. Caso o utilizador opte por não partilhar imediatamente, o ficheiro permanece acessível no local de destino selecionado, permitindo acesso posterior ou reenvio a qualquer momento.

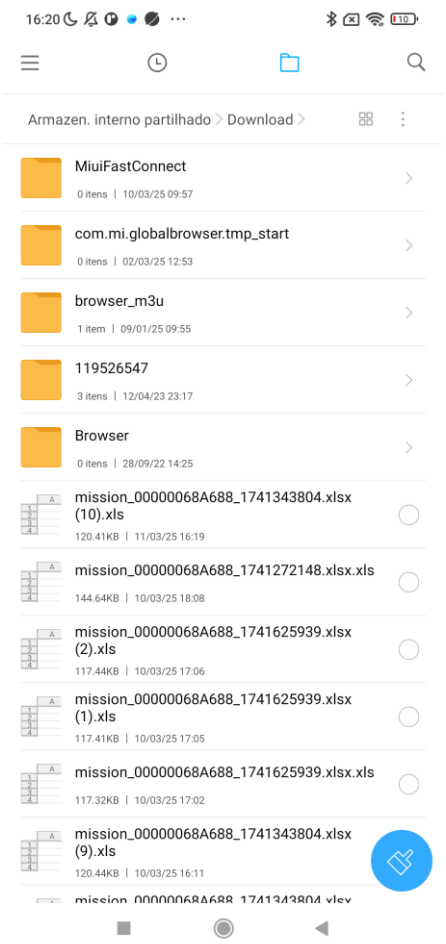


Figura 32 - Ficheiros Excel na memória do dispositivo.

3.6.7. Integridade, segurança e *offline-first*

Tal como projeto 2 iLog, todas as leituras passam por verificação de integridade (CRC/*bitmap*), e a aplicação mantém estado transacional local. Ao abrir os Detalhes da Missão, novas amostras são persistidas automaticamente, garantindo que a exportação reflete o estado mais recente. Se a missão tiver PIN, a leitura/relatório exige autenticação, impedindo visualização/extração por terceiros. O modo *offline-first* assegura que nenhum dado é perdido e que a produção de relatórios não depende de conectividade (ver Figura 23).

3.6.8. Ensaios e resultados

Os testes funcionais abrangem as seguintes funcionalidades principais, todas validadas com sucesso:

- Listagem agrupada por ROM ID dos registadores DS1921G;
- Pesquisa textual por Identificador (12 dígitos), Nome e Descrição da missão;
- Filtro temporal por intervalo de datas, ativado por duplo toque na barra de filtros (*date picker*);
- Exportação .xlsx com geração automática de gráfico temporal e codificação condicional por cor;
- Partilha direta através dos *intents* nativos do Android;
- Eliminação condicionada dos dados locais, permitida apenas para missões concluídas.

Resultados operacionais obtidos:

- Fluxo estável em todos os cenários testados;
- Redução significativa do tempo de análise de missões;
- Melhoria da comunicação entre equipas através de ficheiros portáteis;
- Preservação da operação offline com sincronização diferida quando aplicável.

Esta validação confirma a robustez funcional da solução em contexto de produção.

3.6.9. Documentação e manutenção

Foram produzidos o manual do utilizador e o guia técnico (instalação, permissões, ligações físicas, resolução de problemas, capturas de ecrã), diagramas de arquitetura e registo de decisões ADR (*Architecture Decision Record*). Estes artefactos suportam formação de equipas e reduzem carga de suporte (ver Anexo B).

3.6.10. Limitações e trabalho futuro

Como a exportação é puramente local, a consistência entre múltiplos dispositivos depende estritamente do procedimento operacional adotado: localização de armazenamento, convenção de nomenclatura e responsável pela partilha. Esta limitação operacional foi intencionalmente aceite para priorizar a autonomia *offline*, mas impõe un dispositivo único

por missão: as missões programadas devem ser iniciadas e terminadas no mesmo dispositivo para geração de relatórios completos.

Embora esta restrição possa parecer limitante, revela-se vantajosa em cenários de dispositivo dedicado à leitura dos registadores DS1921G, onde um único equipamento permanece associado aos registadores de um cliente ou localização específica, simplificando a gestão de inventário e reduzindo riscos de perda de dados entre dispositivos.

Perspetivas de evolução incluem: nomenclatura automática padronizada de ficheiros (Cliente_ID_Missao_YYYYMMDD.xlsx), modelos de relatório personalizados por cliente, assinatura digital do ficheiro *.xlsx* (conformidade GDPR) e sincronização diferida com o *Senslive* (*upload* opcional quando rede disponível), preservando a vocação *offline-first* que distingue o iLog PRO.

3.7. Projeto 4 — CTFlash/CTFirmware (automação interna)

O CTFlash/CTFirmware foi concebido para agilizar processos internos críticos da CapTemp—programação de *firmware*, preparação de pedidos de laboratório, scanner de sondas, gestão de MACs e impressão de etiquetas—reduzindo tempos operacionais, erros manuais e dispersão de ferramentas. O objetivo central foi concentrar numa única plataforma fluxos antes heterogéneos (aplicações distintas, criação de etiquetas, edição manual de documentos), assegurando rastreabilidade ponta-a-ponta, governança de acessos e padronização de saídas (documentos por país, etiquetas, registos de auditoria).

3.7.1. Arquitetura de referência e *stack* tecnológico

A solução opera numa VM VMware dedicada com Apache como servidor web e Laravel 12 sobre PHP 8.3 como *framework backend*. Adotou-se renderização por rotas Laravel em vez de SPA completa, combinada com Vue 3 para componentes reativos no *frontend*, Laravel Jetstream com Inertia.js para *scaffolding* de autenticação e navegação fluida entre páginas. A *stack* técnica integra PostgreSQL como base de dados transacional, Tailwind CSS para estilização utilitária e Vite com Laravel Sail (Docker) para *bundling* e ambiente de desenvolvimento *containerizado* (ver Figura 33).

A plataforma implementa autenticação completa, gestão de perfis e permissões RBAC (*Role-Based Access Control*) com auditoria detalhada de todas as operações (quem, quando, o quê). As funcionalidades encontram-se restringidas por papel organizacional: técnico (gestão registadores), laboratório (validações metrológicas), gestão (relatórios administrativos). Esta segmentação assegura conformidade com políticas internas da CapTemp.

O núcleo operacional estabelece comunicação bidirecional *WebSocket* com um *gateway* de campo Node.js multiprotocolo (LoRaWAN, NB-IoT, *Serial*) e delega operações específicas — HTTP, *serial* ou CLI — para equipamentos externos conforme necessário. Esta arquitetura híbrida garante escalabilidade e flexibilidade na integração com infraestrutura *IoT* heterogénea da CapTemp.

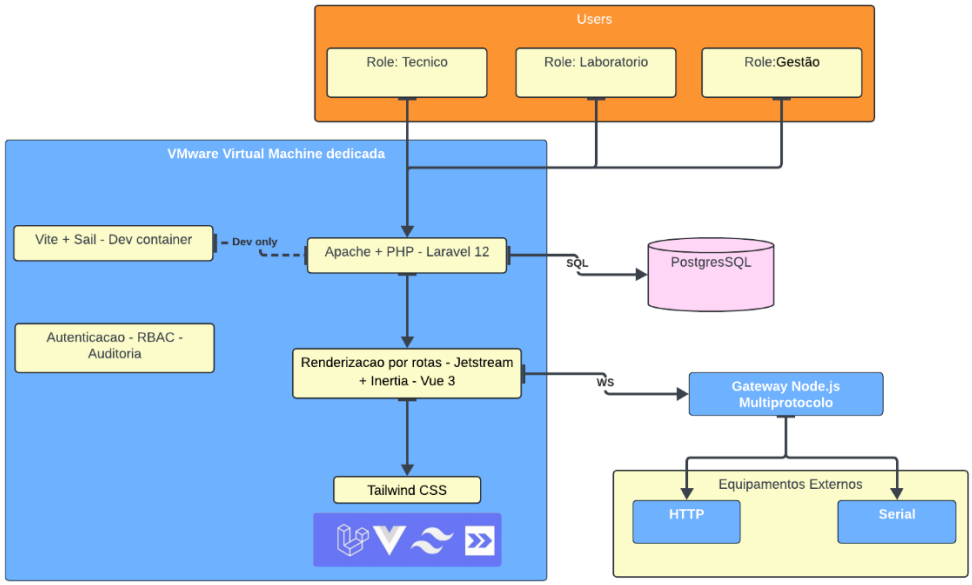


Figura 33 - Arquitetura do projeto CTFlash

3.7.2. Pedidos de laboratório (Portugal/Espanha)

Antes da implementação do CTFlash, os pedidos de verificação e calibração eram preparados manualmente, um a um, implicando duplicação de esforços e elevada propensão a erros humanos. Cada formulário requeria preenchimento repetitivo de dados comuns (cliente, local, técnico responsável), aumentando o *lead time* operacional e comprometendo a consistência documental.

No CTFlash, os formulários oficiais foram replicados digitalmente em *Blade.php* para Portugal (verificação/calibração registadores) e Espanha (sondas *wired*, *wireless*, registadores), com parametrização por país de campos obrigatórios, *logos* institucionais e cláusulas legais específicas ver (Figura 34). O técnico inicia o pedido na aplicação web e adiciona automaticamente as sondas através de ligação WebSocket bidirecional:

- Registo automático a partir de registador conectado: o *gateway* lê as sondas via HTTP/Serial/CLI e preenche automaticamente a *checklist*;
- Leitura por "pistola" de código de barras, adicionando sensores individualmente à lista.

Com a lista de equipamentos completa, o CTFlash gera o documento final (PDF/impresso) em segundos, eliminando completamente a edição manual, garantindo

consistência absoluta entre múltiplos pedidos e assegurando conformidade com modelos oficiais de ambos os países.

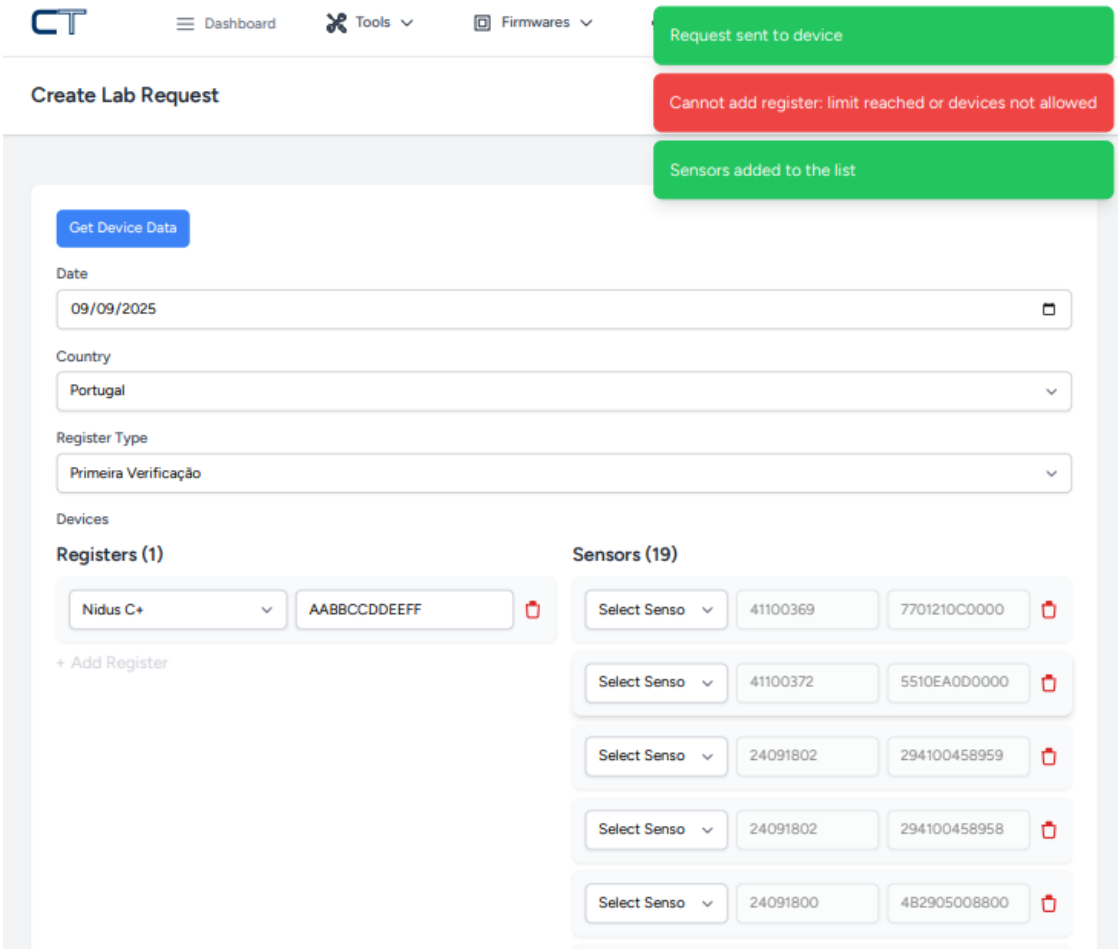


Figura 34 - Exemplo de um pedido de laboratório para Primeira Verificação.

3.7.3. Scanner de sondas (TH3 RS485 + 1-Wire via série)

Com o objetivo de substituir o executável legado e concentrar o diagnóstico numa única plataforma, foi integrado um scanner de sondas na solução ver (Figura 35). A comunicação física é realizada através do TH3 RS485 Converter (ver Anexo D), um controlador modular ligado à porta série por meio de adaptador, garantindo a interface entre o *software* e os dispositivos de campo.

Foram implementados os comandos 1-Wire e as sequências de leitura de acordo com os manuais técnicos da CapTemp, permitindo detetar sondas, ler valores de medição e sinalizar anomalias de forma consistente. A primeira fase do desenvolvimento centrou-se na leitura fiável, validada funcionalmente em contexto operativo, ficando previstas para fases futuras

funcionalidades adicionais, como calibrações, *wizards* de pontos e outras operações do ecossistema.

A comunicação entre o *frontend* e o *gateway* é assegurada por *WebSocket*, que atua como canal bidirecional em tempo real. A aplicação envia ordens de leitura, incluindo o IP e a porta do equipamento configurado, e recebe de imediato os respetivos resultados. Além de consolidar o fluxo de trabalho num único ecrã, a nova solução melhora o layout face à ferramenta anterior em Lazarus, ao apresentar estados claros, *feedback* imediato e histórico de medições para assegurar a rastreabilidade.

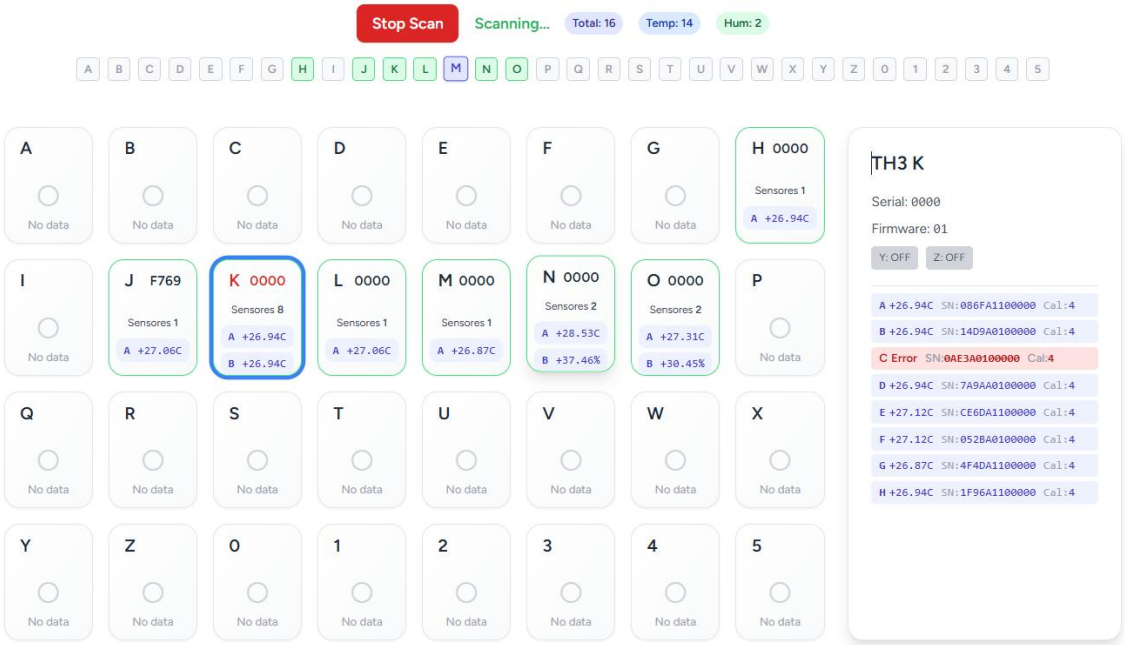


Figura 35 - Scanner de sondas TH3.

3.7.4. Impressão de etiquetas via API (sem edição manual)

Historicamente, a impressão de etiquetas requeria edição manual de imagens/ficheiros, um a um, gerando elevado consumo de tempo e inconsistências documentais entre diferentes técnicos e sessões. No CTFlash, o processo de impressão foi reformulado como declarativo e totalmente automático.

O fluxo implementado segue três etapas sequenciais: (1) o técnico seleciona o tipo de etiqueta (sonda ou registrador); (2) a aplicação obtém automaticamente os campos necessários — via *WebSocket* do registador conectado ou leitura por pistola de

código de barras — e adiciona os dados à fila de impressão; (3) o servidor efetua POST para a API da impressora, utilizando *templates* pré-configurados com nome e *endpoint* específicos.

Para gestão da fila foi desenvolvido um quadro *Kanban* com três estados visuais: Em Progresso, Impresso e Erro, incluindo funcionalidades de repetição automática, *logs* de tentativa e reimpressão rápida (ver Figura 36). Esta solução assegura visibilidade do *throughput*, isolamento de falhas e controlo total do processo. O impacto operacional manifesta-se na redução drástica do tempo de preparação de encomendas e na uniformização total das etiquetas impressas.

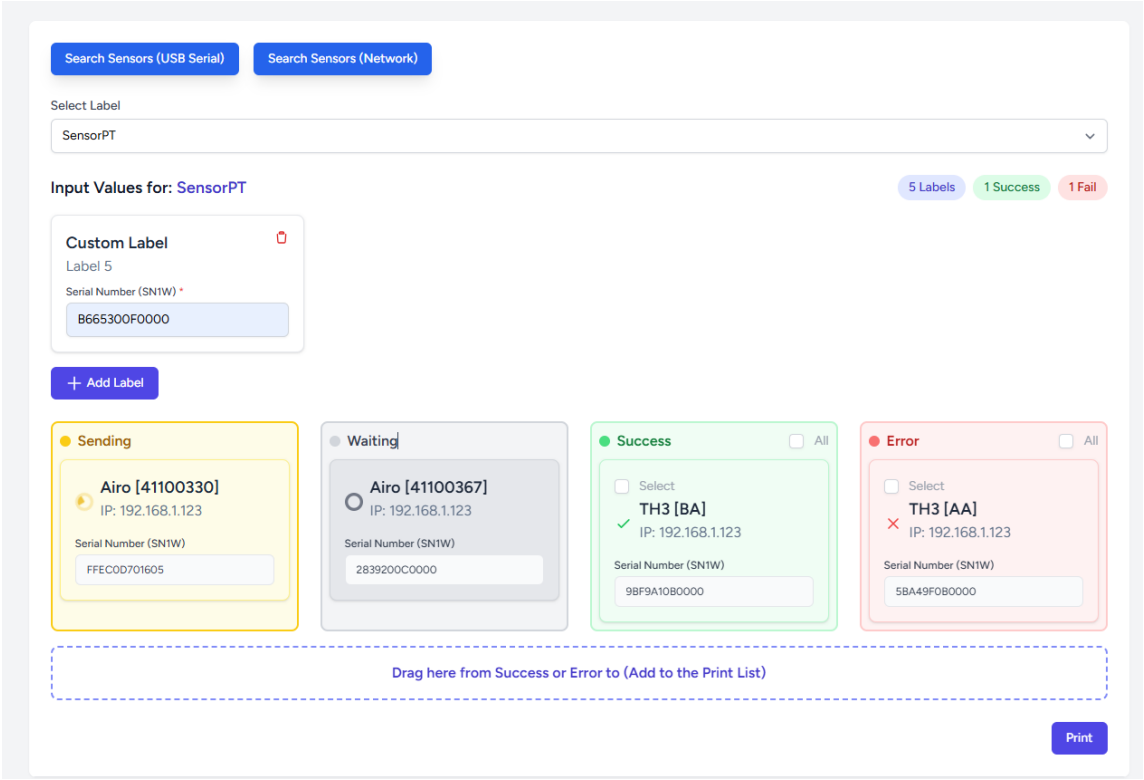


Figura 36 - Atual modelo para impressão de etiquetas.

3.7.5. Gestão automática de MACs (LoRa/ISM)

A atribuição de endereços MAC aos registadores constituía, anteriormente, um processo manual, com reduzida capacidade de auditoria e maior probabilidade de erro operacional. No CTFlash, esta operação foi automatizada através de um mecanismo de gestão de *pools* por tecnologia — nomeadamente LoRa e ISM — com distinção entre endereços verificados

e não verificados, prevenindo colisões e assegurando consistência no processo de atribuição (ver Figura 37).

A integração com o *WebSocket* permite obter, em tempo real, o endereço MAC atualmente configurado no dispositivo e, complementarmente, acionar um segundo processo que procura na base de dados um endereço disponível compatível com o tipo de tecnologia e com o estado do equipamento. Desta forma, a plataforma consegue não apenas identificar o estado atual do registador, mas também atribuir automaticamente um novo endereço válido, assegurando coerência entre o dispositivo físico e o inventário lógico mantido em base de dados.

Cada atribuição fica registada com informação de quem executou a operação e quando esta ocorreu, reforçando a rastreabilidade e a capacidade de auditoria. Adicionalmente, um *cron job* executado periodicamente envia uma notificação por correio eletrónico à gestão do departamento com o resumo do estado operacional, incluindo MACs atribuídos, MACs reservados e falhas de atribuição, como por exemplo interrupções no fluxo de processamento.

A base de dados mantém, assim, uma representação atualizada do estado dos endereços disponíveis e utilizados, permitindo monitorização contínua da capacidade e emissão de alertas quando se aproximam limites críticos de *stock*. Esta abordagem melhora o controlo operacional, reduz riscos de duplicação e aumenta a visibilidade do inventário técnico.

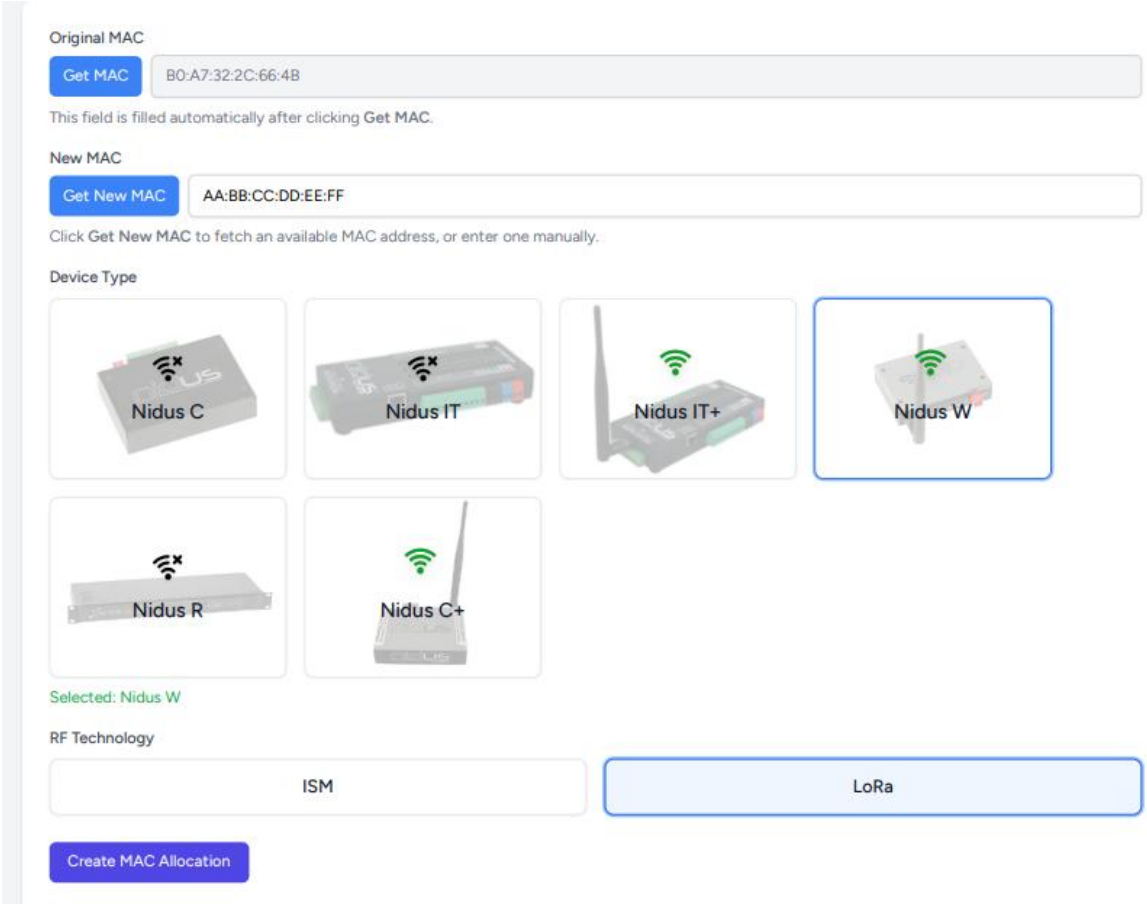


Figura 37 - Atribuição de endereços MAC para os registradores.

3.7.6. Programação de *firmware* (CTFirmware) e integração futura

Na fase final do estágio, foi desenvolvida uma aplicação autónoma destinada à programação rápida de *firmware* (ver Figura 38), com integração futura prevista no CTFlash. O fluxo operacional foi concebido para ser direto e eficiente: o técnico seleciona um ficheiro *.hex* de base, edita os campos necessários correspondentes aos parâmetros do *build*, gera um novo ficheiro *.hex* e procede à programação do dispositivo através do programador adequado, como, por exemplo, o PICkit 3.

Esta solução foi entregue em contexto de necessidade urgente de cliente, associado à existência de um elevado número de dispositivos LoRaWAN com *firmware* desatualizado. A ferramenta permitiu, assim, viabilizar um processo de atualização em lote, reduzindo o tempo de intervenção técnica e tornando o procedimento mais consistente e controlado.

Como evolução futura, prevê-se a integração desta funcionalidade no CTFlash, com o objetivo de unificar o processo de gestão e programação de *firmware* numa única plataforma.

O *roadmap* inclui a criação de um catálogo estruturado versão → ficheiro HEX, mecanismos de auditoria (quem, quando e em que dispositivo foi efetuada a operação), *rollbacks* controlados e validação de *checksums*, reforçando a segurança e a rastreabilidade do processo.

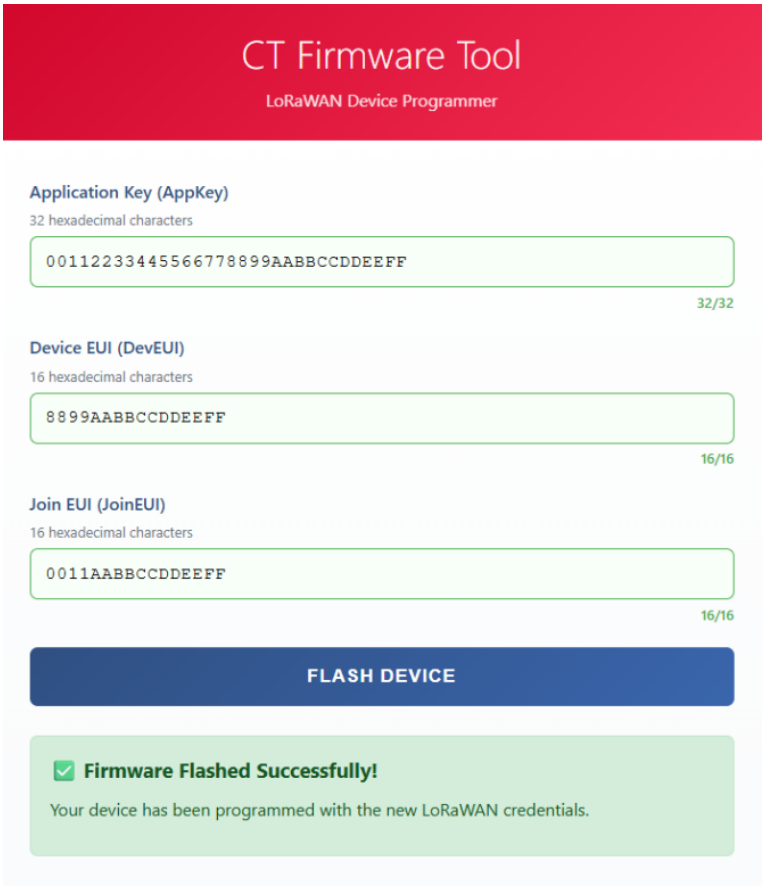


Figura 38 - Ferramenta para programar ficheiros .hex.

3.7.7. Impacto e valor para a CapTemp

A plataforma implementa RBAC (*Role-Based Access Control*) para segmentação granular de permissões por perfil organizacional, com *logs* de auditoria registados para todas as ações críticas. As ligações *WebSocket* e HTTP utilizam TLS 1.3, complementadas por proteções CSRF e rate *limiting* na camada Laravel. O ambiente VMware assegura isolamento entre instâncias e suporta *snapshots* para *rollbacks* rápidos. A monitorização de desempenho inclui rastreio de latências, exceções e consultas N+1, mantendo os *Service Level Objectives* (SLO) internos definidos.

O controlo de versões e planeamento ágil foram geridos através do Gitea, utilizando *issues*, pull *requests* e quadro *Kanban* para acompanhamento de progresso e revisão de código. Durante o estágio, tornaram-se operacionais os módulos críticos: scanner de sondas, pedidos de laboratório (Portugal/Espanha) e impressão de etiquetas.

Em produção interna, estes módulos produziram impactos mensuráveis: redução do tempo de ciclo operacional, diminuição de erros de transcrição humana e aumento da rastreabilidade documental através de documentos padronizados e *logs* auditáveis. O projeto mantém desenvolvimento contínuo, com prioridade para integração do *CTFirmware*, calibração no scanner TH3, leitura de registadores Bluetooth e expansão para fluxos de aprovação automatizados e atualizações de *firmware*.

4. Análise crítica, proposta de melhorias e trabalho futuro

O conjunto de projetos desenvolvidos durante o estágio permitiu intervir em diferentes níveis do ecossistema técnico da CapTemp, desde a comunicação em tempo real e a operação móvel em campo até à automação de tarefas internas e à produção documental. Esta diversidade tornou possível aplicar conhecimentos de Computação Móvel, integração com *hardware*, sincronização *cloud*, persistência local e engenharia de software em contexto real, mas também evidenciou limitações estruturais e oportunidades de melhoria relevantes para a evolução futura das soluções.

4.1. Análise crítica

De forma global, os quatro projetos desenvolvidos revelaram coerência com os objetivos do estágio e responderam a necessidades concretas da entidade de acolhimento, nomeadamente ao nível da mobilidade, rastreabilidade, automação e suporte operacional. A principal mais-valia técnica consistiu na articulação entre dispositivos físicos, aplicações móveis, serviços *cloud* e plataformas internas, permitindo uma visão integrada do ciclo dispositivo ⇄ *mobile* ⇄ *cloud* ⇄ operação.

No Projeto 1, relativo ao Chat/Notificações em tempo real, a solução baseada em *WebSocket* Secure e notificações *push* mostrou-se adequada para suportar comunicação bidirecional de baixa latência e entrega assíncrona de alertas. Ainda assim, a evolução para contexto produtivo alargado exige reforço de observabilidade, definição de métricas operacionais e maior formalização dos mecanismos de monitorização e resposta a incidentes.

No Projeto 2, a aplicação iLog demonstrou a viabilidade de integrar um dispositivo Android com adaptadores USB/1-Wire e registadores DS1921G, assegurando operações de programação, leitura e sincronização com o *Senslive*. A robustez alcançada após a mitigação de bloqueios na comunicação USB e a adoção de uma estratégia *offline-first* com persistência local constituíram resultados relevantes; no entanto, a solução permanece dependente de conhecimento técnico específico e de componentes fortemente acoplados ao hardware atualmente suportado.

No Projeto 3, a aplicação iLog PRO respondeu de forma eficaz a cenários em que a exportação local e a produção imediata de relatórios são mais adequadas do que a sincronização direta com a *cloud*. O suporte a ficheiros *.xlsx* com estrutura organizada, representação gráfica e realce visual de excedências acrescenta valor operacional, embora a solução possa ainda evoluir em aspetos de padronização documental, personalização por cliente e reforço da validade formal dos relatórios gerados.

No Projeto 4, a plataforma CTFlash/CTFirmware permitiu concentrar num único ambiente várias tarefas anteriormente dispersas, incluindo programação de *firmware*, leitura de sondas, criação de pedidos laboratoriais, impressão de etiquetas e gestão de identificadores. O principal contributo deste projeto esteve na redução de tarefas manuais e no aumento da rastreabilidade interna, embora subsistam desafios relacionados com a manutenção evolutiva, a integração de módulos ainda não concluídos e a necessidade de maior uniformização dos processos internos.

Em termos transversais, uma das limitações mais relevantes identificadas ao longo do estágio prende-se com a coexistência entre soluções novas e ferramentas legadas, situação que aumenta a complexidade de manutenção e dificulta a homogeneização de processos. A isso somam-se constrangimentos ligados à capacidade de suporte técnico, à heterogeneidade dos dispositivos utilizados em campo e à necessidade de reforçar documentação, formação e mecanismos de observação contínua das aplicações em ambiente real.

4.2. Propostas de melhoria (curto/médio prazo)

Como primeira linha de evolução, recomenda-se o reforço da observabilidade técnica do ecossistema, através da definição de indicadores operacionais coerentes e de mecanismos centralizados de monitorização para aplicações móveis, serviços *backend*, fluxos de sincronização e componentes de automação interna. Esta melhoria permitiria detetar falhas mais cedo, reduzir o tempo de diagnóstico e apoiar decisões técnicas com base em evidência operacional.

No caso das aplicações iLog e iLog PRO, seria vantajosa a extração de uma camada partilhada de comunicação e acesso a dispositivos 1-Wire, sob a forma de módulo reutilizável ou *SDK* interno, de modo a reduzir duplicação de código e facilitar a extensão futura a novos equipamentos. Esta abordagem contribuiria para maior modularidade, menor esforço de manutenção e melhor controlo da evolução tecnológica.

Para o iLog PRO, recomenda-se ainda o reforço do subsistema de geração documental, nomeadamente através da criação de modelos de relatório ajustáveis por cliente ou contexto de utilização, bem como da introdução de mecanismos de validação formal, como assinatura digital ou controlo de integridade dos ficheiros gerados. Tal melhoria aumentaria a confiança nos relatórios produzidos e aproximaria a solução de cenários com exigências mais fortes de auditoria e conformidade.

No CTFlash/CTFirmware, a prioridade deverá centrar-se na consolidação dos módulos ainda em evolução, sobretudo na integração completa da programação de *firmware*, na formalização do catálogo versão–ficheiro e no reforço dos mecanismos de auditoria e recuperação. Em paralelo, a expansão de fluxos guiados para operações laboratoriais e calibração poderá reduzir dependência de conhecimento tácito e tornar o sistema mais acessível a diferentes perfis de utilizador.

A nível organizacional, recomenda-se também o investimento em documentação operacional, formação interna estruturada e criação de materiais de apoio orientados para instalação, diagnóstico e suporte. Esta medida é particularmente importante num contexto em que o crescimento do número de instalações e clientes aumenta a pressão sobre equipas técnicas reduzidas.

A Tabela 7 - Síntese de desafios e ações propostas, sintetiza a correspondência entre desafios, ações e impacto esperado.

Tabela 7 - Síntese de desafios e ações propostas

Desafio prioritário	**Ação proposta**	**Impacto esperado**
Suporte a crescer com equipa reduzida	Academia, *runbooks*, *triage* 1.º nível	Menos tempo médio de resolução; qualidade estável
Heterogeneidade e legado	SDK 1-Wire + *adapters* de migração	Menos retrabalho; integração mais rápida
Observabilidade limitada	SLIs/SLOs, alertas e *post-mortems*	Prevenção e resposta mais rápidas

Conformidade documental (iLog PRO)	*Templates* e assinatura digital	Auditoria facilitada; confiança do cliente
Automação de firmware/TH3 por concluir	Integrar CTFirmware + *wizards*	Escala e rastreabilidade em produção
Governança e internacionalização	Retenção/lineage + *localization packs*	Menos risco e melhor adaptação a mercados

4.3. Trabalho futuro

O trabalho futuro poderá seguir várias direções complementares, de acordo com os resultados obtidos e com as necessidades identificadas durante o estágio. Uma dessas direções consiste na consolidação de uma arquitetura mais reutilizável para integração com dispositivos físicos, permitindo suportar novos registadores e novos protocolos de comunicação sem necessidade de reimplementação extensiva.

No domínio das aplicações móveis, será relevante continuar a evoluir o suporte a operações *offline-first*, a resiliência a falhas de comunicação e a compatibilidade com diferentes perfis de dispositivos Android. Em paralelo, a exploração de alternativas para integração com outras plataformas móveis, incluindo iOS, poderá ser retomada no futuro, embora atualmente existam limitações técnicas relevantes ao nível do acesso ao *stack* USB-C dessa plataforma.

Para o iLog PRO, o trabalho futuro poderá incluir a expansão dos formatos de saída, a melhoria da personalização dos relatórios e a integração com mecanismos de arquivo, assinatura e rastreabilidade documental. Estas evoluções são particularmente relevantes em contextos regulados, nos quais a qualidade da evidência documental assume importância central.

No caso do CTFlash/CTFirmware, a continuação do desenvolvimento deverá privilegiar a integração plena da programação de *firmware*, o alargamento do suporte a novos dispositivos e a evolução dos módulos de calibração, leitura e expedição. A médio prazo, esta plataforma poderá afirmar-se como núcleo operacional unificado para tarefas técnicas internas, reduzindo fragmentação aplicacional e reforçando a consistência dos processos.

De forma transversal, a evolução futura das soluções beneficiará da combinação entre modularidade técnica, observabilidade, documentação e capacitação interna. Estes vetores serão determinantes para escalar o ecossistema de forma sustentável, mantendo níveis adequados de fiabilidade, auditabilidade e qualidade de serviço.

5. Conclusão

O estágio curricular realizado na CapTemp, Lda., no âmbito do Mestrado em Engenharia Informática – Computação Móvel, permitiu aplicar em contexto real conhecimentos de desenvolvimento móvel, integração com *hardware*, comunicação em tempo real, persistência local e automação de processos internos. Os quatro projetos desenvolvidos — Chat/Notificações, iLog, iLog PRO e CTFlash/CTFirmware — responderam a necessidades concretas do ecossistema *Senslive*, abrangendo desde a recolha e programação de dispositivos em campo até à consolidação documental, operacional e administrativa. Esta diversidade permitiu intervir em diferentes camadas de um sistema real, reforçando a compreensão global do ciclo dispositivo → *mobile* → *cloud* → operação.

Os objetivos definidos no início do estágio foram plenamente atingidos. A experiência prática complementou a formação académica, permitindo transformar competências teórico-práticas em soluções com impacto direto no funcionamento da empresa. Foram consolidados hábitos de trabalho alinhados com ambientes de produção, incluindo planeamento incremental, controlo de versões rigoroso, documentação estruturada e integração contínua com equipas multidisciplinares. A metodologia SCRUM semanal, suportada por Gitea, assegurou ciclos curtos de valor e *feedback* constante, contribuindo para uma prática profissional mais madura e orientada à qualidade.

O contributo técnico para a CapTemp foi particularmente evidente em três eixos. No domínio da mobilidade e integração de campo, os projetos iLog e iLog PRO estabeleceram uma ponte robusta entre registadores DS1921G e o *Senslive*, suportando cenários *online* e *offline*, garantindo integridade dos dados e permitindo a geração autónoma de relatórios estruturados. Ao nível da comunicação operacional, o Chat/Notificações demonstrou a viabilidade de um canal de baixa latência baseado em *WebSocket Secure*, articulado com notificações *push* (APNs/FCM), reduzindo dependência de interação humana em alertas e consultas críticas. Por fim, na automação interna, o CTFlash/CTFirmware consolidou fluxos laboratoriais, etiquetagem e programação de *firmware*, aumentando rastreabilidade, reduzindo tarefas manuais e uniformizando processos antes dispersos.

Do ponto de vista formativo, o estágio aprofundou competências essenciais em Sistemas Distribuídos, incluindo WSS, *backpressure* e mecanismos *at-least-once*; em engenharia de fiabilidade, com CRC-16 e validação de integridade; em persistência móvel, com ObjectBox

e estratégias *offline-first*; em UX operacional, orientada a contextos de campo; e em práticas DevOps aplicadas, como observabilidade, *rollbacks* e RBAC. A exposição a casos de uso regulados reforçou a importância da rastreabilidade, da segurança por desenho e da documentação rigorosa, aspetos centrais em setores como cadeia do frio e metrologia.

Naturalmente, subsistem limitações e oportunidades de evolução. A heterogeneidade de dispositivos Android, a coexistência com ferramentas legadas, as restrições do iOS ao USB-C e a necessidade de reforçar observabilidade (SLIs/SLOs) e assinatura digital em relatórios constituem desafios que não comprometem os resultados alcançados, mas apontam para uma linha de evolução estratégica. A consolidação de módulos ainda em desenvolvimento, nomeadamente no CTFlash/CTFirmware, e o reforço da documentação e formação interna surgem igualmente como passos relevantes para a maturação contínua do ecossistema.

Ao nível pessoal e profissional, o estágio permitiu uma evolução clara em termos de autonomia, comunicação técnica e colaboração com o supervisor e com as diferentes equipas envolvidas. Entre as principais aprendizagens, destacam-se a importância de medir, através de mecanismos de observabilidade; de documentar, para facilitar a disseminação, a manutenção e a escalabilidade; e de prototipar, como suporte à tomada de decisão em fases iniciais de desenvolvimento. Estas competências constituem uma base sólida para uma participação responsável e eficaz em projetos que exigem fiabilidade, segurança e rastreabilidade de ponta a ponta.

Em síntese, o estágio permitiu transformar conhecimento académico em entregas com impacto real, incluindo o desenho de protocolos, a integração de *hardware*, a consolidação de fluxos *cloud* e a automatização de operações. O trabalho desenvolvido reforça a capacidade de atuar em *IoT* aplicada e consolida um perfil profissional preparado para contextos de mobilidade industrial, integração *hardware*/*software* e desafios característicos da Indústria 4.0. Teve também impacto direto na formação, ao aprofundar conhecimentos em *IoT* aplicada, alargar competências tecnológicas e desenvolver uma consciência mais clara do que significa operar sistemas 24/7 em contexto industrial, onde pequenas falhas podem ter consequências operacionais relevantes.

Bibliografia

[1] "CapTemp- Monitoring Systems with Wireless Sensor & sync Technology." Accessed: Sep. 22, 2025. [Online]. Available: https://captemp.com/

[2] "SensLive Portal [Powered by CapTemp]." Accessed: Sep. 22, 2025. [Online]. Available: https://senslive.com/

[3] "EN 12830:2018 - Temperature Recorders for Cold Chain Transport Storage." Accessed: Apr. 27, 2026. [Online]. Available: https://standards.iteh.ai/catalog/standards/cen/b6c1bd4b-fafb-41ee-9f9c-e729e4ca7c97/en-12830-2018?srsltid=AfmBOoqaV8PRph4jnV3m8K4H8cYCnTrWDcnoIYjalO9vU85BQDwOP1cO

[4] "Post | LinkedIn." Accessed: Apr. 27, 2026. [Online]. Available: https://www.linkedin.com/posts/wwwcaptempcom-share-7368249788296781824-HOyk/

[5] "(310) Abacus is a Smart Vision System dedicated for Egg counting [SVS] - Full Video - YouTube." Accessed: Sep. 22, 2025. [Online]. Available: https://www.youtube.com/watch?v=REbkOo9d7fI

[6] "(1239) Abacus is a Smart Vision System dedicated for Egg counting [SVS] - Full Video - YouTube." Accessed: Apr. 27, 2026. [Online]. Available: https://www.youtube.com/watch?v=REbkOo9d7fI

[7] "Gitea Official Website." Accessed: Sep. 24, 2025. [Online]. Available: https://about.gitea.com/

[8] "Android Mobile App Developer Tools – Android Developers." Accessed: Sep. 24, 2025. [Online]. Available: https://developer.android.com/

[9] "PieHost - Scalable Realtime, Cloud Servers, Databases and Hosting." Accessed: Sep. 24, 2025. [Online]. Available: https://piehost.com/

[10] "Adminer - Database management in a single PHP file." Accessed: Sep. 24, 2025. [Online]. Available: https://www.adminer.org/en/

[11] “Adobe XD Learn & Support.” Accessed: Sep. 24, 2025. [Online]. Available: https://helpx.adobe.com/support/xd.html

[12] “Hercules | HW-group.com.” Accessed: Sep. 24, 2025. [Online]. Available: https://www.hw-group.com/product-version/hercules

[13] “Fusion and Workstation | VMware.” Accessed: Sep. 24, 2025. [Online]. Available: https://www.vmware.com/products/desktop-hypervisor/workstation-and-fusion

[14] “Real-Time Web Application Debugging.” Accessed: Sep. 29, 2025. [Online]. Available: https://inspector.dev/

[15] M. Integrated, “Thermochron iButton Device”, Accessed: Sep. 22, 2025. [Online]. Available: www.maximintegrated.com

[16] “USB host overview | Connectivity | Android Developers.” Accessed: Apr. 09, 2026. [Online]. Available: https://developer.android.com/develop/connectivity/usb/host

[17] “java - Android 3.1 USB-Host - BroadcastReceiver does not receive USB_DEVICE_ATTACHED - Stack Overflow.” Accessed: Apr. 11, 2026. [Online]. Available: https://stackoverflow.com/questions/6981736/android-3-1-usb-host-broadcastreceiver-does-not-receive-usb-device-attached

[18] “Interface 1-Wire USB Adapter with Android | Analog Devices.” Accessed: Apr. 11, 2026. [Online]. Available: https://www.analog.com/en/resources/app-notes/interface-1wire-usb-adapter-with-android.html

[19] “Tipos de serviços em primeiro plano | Background work | Android Developers.” Accessed: Apr. 11, 2026. [Online]. Available: https://developer.android.com/develop/background-work/services/fgs/service-types?hl=pt-br

[20] M. Integrated, “Thermochron iButton Device”, Accessed: Apr. 11, 2026. [Online]. Available: www.maximintegrated.com

[21] "Interface 1-Wire USB Adapter with Android | Analog Devices." Accessed: Sep. 22, 2025. [Online]. Available: https://www.analog.com/en/resources/app-notes/interface-1wire-usb-adapter-with-android.html

[22] "Usb host bulk transfer misses packets when called continuosly. [37002652] - Issue Tracker." Accessed: Apr. 12, 2026. [Online]. Available: https://issuetracker.google.com/issues/37002652?pli=1

[23] "Protect your Android device from USB threats with Advanced Protection Mode - Android Help." Accessed: Apr. 12, 2026. [Online]. Available: https://support.google.com/android/answer/16778864?hl=en

[24] "cyclic redundancy check (CRC) - Glossary | CSRC." Accessed: Apr. 12, 2026. [Online]. Available: https://csrc.nist.gov/glossary/term/cyclic_redundancy_check

Anexos

Anexo	Descrição	Ficheiro
Anexo A	Manual iLog	CT_iLog_manual.pfd
Anexo B	Manual iLog PRO	CT_iLog_PRO_manual.pfd
Anexo C	Manual Upload Firmwares	CT_firmware_upload_tool.pdf
Anexo D	TH3 Datasheet	captemp_th3_datasheet.pdf