ISSN 1688-2806



Universidad de la República
Facultad de Ingeniería



Desarrollo de un sistema prototipo con capacidad de adquisición, procesamiento y visualización de señales eléctricas en vegetales con capacidad para ser utilizado en cultivos

Tesis presentada a la Facultad de Ingeniería de la Universidad de la República por

Rodrigo Fernández, Fabrizio Rafaelli, Gastón Rodríguez

en cumplimiento parcial de los requerimientos para la finalización de la carrera de Ingeniería Eléctrica.

Dirección de Tesis

Leonardo Barboni ..... Universidad de la República

Tribunal

Andrés Seré ..... Universidad de la República
Juan Pablo Oliver ..... Universidad de la República
Mariana Del Castillo ..... Universidad de la República

Montevideo
viernes 29 mayo, 2026

Desarrollo de un sistema prototipo con capacidad de adquisición, procesamiento y visualización de señales eléctricas en vegetales con capacidad para ser utilizado en cultivos, Rodrigo Fernández, Fabrizio Rafaelli, Gastón Rodríguez.

ISSN 1688-2806

Esta tesis fue preparada en LaTeX usando la clase iietesis (v1.2).
Contiene un total de 148 páginas.
Compilada el viernes 29 mayo, 2026.
http://iie.fing.edu.uy/

Do. Or do not. There is no try.

Yoda

Esta página ha sido intencionalmente dejada en blanco.

Agradecimientos

En primer lugar, se agradece al tutor Leonardo Barboni por el acompañamiento, la guía técnica y las sugerencias aportadas a lo largo de todo el proyecto, así como por su disponibilidad para orientar decisiones de diseño y validar criterios de implementación.

Asimismo, se agradece a la Facultad de Agronomía por el apoyo brindado mediante el préstamo del material vegetal utilizado en los ensayos, en particular plantas de vid (incluyendo las variedades Tanat y otras), fundamentales para la etapa experimental y de validación del sistema.

Finalmente, se agradece a la Facultad de Ingeniería por las instalaciones y recursos puestos a disposición, que permitieron realizar tareas de soldadura, pruebas, estudio y desarrollo, indispensables para llevar adelante el prototipo y completar las distintas etapas del trabajo.

Esta página ha sido intencionalmente dejada en blanco.

Resumen

El trabajo consiste en el desarrollo de un dispositivo portátil para la adquisición de señales eléctricas en plantas, pensado para su uso en condiciones de campo. El proyecto surge a partir de una línea de investigación de la Facultad de Agronomía de la Universidad de la República, donde se estudia la electrofisiología de especies vegetales, en particular viñas y otras plantas de tronco leñoso. Dentro de esa línea existe interés en analizar si la actividad eléctrica de las plantas puede estar vinculada con variables fisiológicas relevantes, como su estado hídrico.

Dado que la electrofisiología vegetal sigue siendo un área poco desarrollada y con un fuerte componente experimental, una de las principales dificultades es contar con herramientas adecuadas para registrar señales en condiciones reales de uso, fuera del laboratorio. En ese contexto, la tesis no busca resolver la interpretación biológica de las mediciones, sino aportar una herramienta tecnológica que permita a los equipos de agronomía realizar ensayos de forma más práctica, autónoma y sistemática.

Como resultado, el proyecto propone y construye un prototipo funcional capaz de adquirir, almacenar y poner a disposición estas señales eléctricas para su posterior análisis. El objetivo final es entregar una plataforma de medición que facilite el trabajo experimental en campo y sirva como base para futuras investigaciones en electrofisiología vegetal.

Esta página ha sido intencionalmente dejada en blanco.

Tabla de contenidos

Agradecimientos III

Resumen V

1. Introducción 1
- 1.1. Contexto . . . 1
- 1.2. Motivación . . . 3
- 1.3. Objetivos . . . 4
  - 1.3.1. Objetivo general . . . 4
  - 1.3.2. Objetivos específicos . . . 4
  - 1.3.3. Alcance . . . 5
- 1.4. Contenido . . . 6

2. Descripción del sistema 7

3. Hardware 13
- 3.1. Introducción . . . 13
- 3.2. Diseño de la Electrónica . . . 13
  - 3.2.1. Selección del microcontrolador . . . 14
  - 3.2.2. Selección del amplificador de instrumentación . . . 15
  - 3.2.3. Selección del amplificador operacional auxiliar . . . 16
  - 3.2.4. Selección de conversor USB a TTL . . . 17
  - 3.2.5. Prototipo 1 . . . 17
  - 3.2.6. Prototipo 2 . . . 19
  - 3.2.7. Prototipo 3 (Final) . . . 22
- 3.3. Selección de batería . . . 27
- 3.4. Selección de accesorios IP . . . 27
- 3.5. Electrodos . . . 30

4. Firmware y Software 33
- 4.1. Firmware . . . 33
  - 4.1.1. Introducción . . . 33
  - 4.1.2. Arquitectura general . . . 33
  - 4.1.3. Módulos . . . 34
  - 4.1.4. Bucle Principal . . . 45
- 4.2. Software . . . 53

4.2.1. Introducción 53
4.2.2. Arquitectura general 53
4.2.3. Interfaz y comunicación con el dispositivo 55
4.2.4. Visualización, exportación y análisis de datos 58
4.2.5. Bibliotecas utilizadas 61

5. Pruebas y mediciones **63**
5.1. Medidas de la electrónica 63
5.2. Consumo 66
5.3. Estimación del tamaño de los archivos generados 69
5.4. Pruebas funcionales 71
5.5. Pruebas preliminares en Mimosa Púdica 73
5.5.1. Ensayo prolongado sobre *Mimosa pudica* 74
5.5.2. Respuesta ante estímulo mecánico 77
5.6. Comparación entre electrodos de acero y plata 77
5.7. Análisis temporal por ciclos de 24 horas 79
5.8. Electrodo de hoja 82

6. Conclusiones **83**
6.1. Evaluación de resultados 83
6.2. Reflexiones 85
6.3. Trabajo a futuro 86

A. Manual de Uso **89**

B. Generación del ejecutable de la interfaz en Windows **113**

C. Graficas **117**

D. Ajuste de la medida de V_{REF} **125**

Referencias **127**

Índice de tablas **130**

Índice de figuras **132**

Capítulo 1

Introducción

1.1. Contexto

La agronomía actual enfrenta el desafío de comprender con mayor profundidad el comportamiento fisiológico de los cultivos en condiciones reales de operación, donde las variables ambientales cambian de forma continua y no controlada. Factores como la radiación solar, la temperatura, la humedad, el viento y la disponibilidad de agua afectan de manera directa el estado de la planta y condicionan tanto su desarrollo como su productividad. En este escenario, disponer de herramientas que permitan observar respuestas fisiológicas de forma continua y no destructiva resulta de gran interés para el estudio agronómico y para el eventual desarrollo de nuevas estrategias de monitoreo.

Dentro de ese marco se ubica la electrofisiología vegetal, un área de investigación que estudia la actividad eléctrica presente en las plantas y su posible relación con distintos procesos fisiológicos. Si bien se trata de una disciplina con antecedentes en la literatura [27], su desarrollo sigue siendo limitado en comparación con otras áreas de medición biológica, y aún persisten múltiples interrogantes tanto sobre el origen preciso de las señales registradas como sobre su interpretación y utilidad práctica. En particular, en especies leñosas como la vid, existe interés en analizar si la actividad eléctrica puede aportar información vinculada al estado hídrico de la planta, o al menos acompañar de forma indirecta ciertos cambios fisiológicos de interés agronómico.

Dentro de este campo, la *Mimosa pudica* constituye una especie de interés experimental, debido a que presenta respuestas visibles ante estímulos externos, como contacto mecánico o cambios en la iluminación. Esta característica permite utilizarla como planta de referencia para evaluar sistemas de adquisición de señales bioeléctricas, ya que facilita la asociación entre un estímulo aplicado y una variación registrada. En trabajos previos se han desarrollado sistemas electrónicos para registrar la respuesta eléctrica de *Mimosa pudica* ante estímulos luminosos, lo que refuerza su utilidad en este tipo de ensayos [12].

En la Facultad de Agronomía de la Universidad de la República existe una línea de trabajo orientada al estudio de la electrofisiología de plantas leñosas, entre ellas

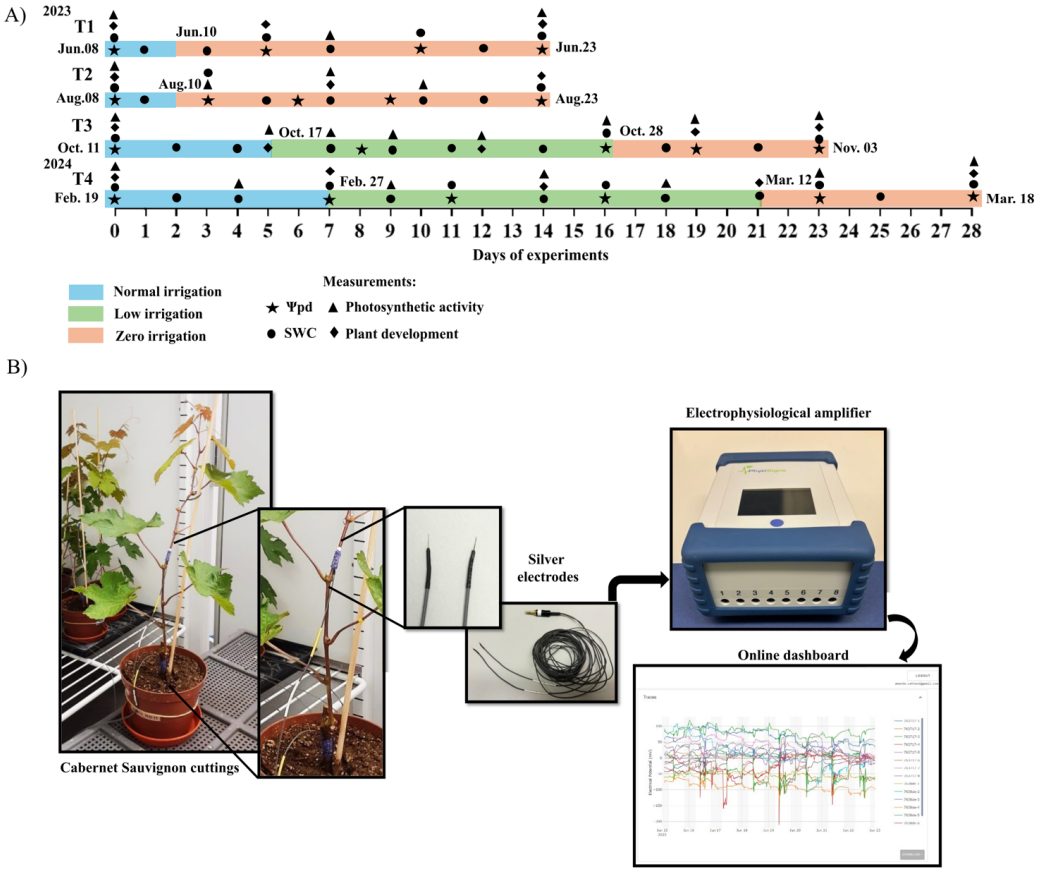


Figura 1.1: Linea de tiempo y diseño experimental de las pruebas de deshidratación en una vid. [10]

viñas, con el objetivo de explorar experimentalmente estas relaciones. Sin embargo, una de las principales limitaciones para avanzar en este tipo de estudios no radica únicamente en la formulación de hipótesis biológicas, sino en la disponibilidad de instrumentación adecuada para realizar mediciones fuera del laboratorio. La adquisición de señales eléctricas vegetales presenta dificultades particulares: se trata de señales muy sensibles al ruido, al movimiento mecánico y a las condiciones del entorno, cuya medición exige cuidado tanto en el contacto con la planta como en el diseño de la electrónica y del sistema de registro.

En un ambiente de laboratorio es posible reducir parte de estas dificultades mediante instrumentación de banco, alimentación controlada y condiciones de ensayo más estables. Un ejemplo de esto es un reciente articulo [10] donde se logró obtener señales representativas en un ambiente controlado, se presenta el montaje en la Figura 1.1.

No obstante, en aplicaciones agronómicas reales resulta necesario medir directamente en campo, donde aparecen nuevas restricciones: alimentación por batería, exposición a condiciones ambientales variables, necesidad de portabilidad, robustez frente a vibraciones e interferencias, facilidad de instalación y operación por

usuarios que no necesariamente son especialistas en electrónica. Estas condiciones convierten al problema de medición en un problema de ingeniería en sí mismo.

Por este motivo, en el contexto de esta tesis el foco principal no se sitúa en la interpretación fisiológica definitiva de las señales adquiridas, sino en el desarrollo de una plataforma instrumental que permita registrarlas de forma confiable en condiciones reales. En otras palabras, el problema central abordado no es demostrar una relación biológica cerrada entre la señal eléctrica y una variable fisiológica determinada, sino diseñar y construir un dispositivo que haga posible la realización de ensayos experimentales de manera práctica, reproducible y autónoma. El proyecto se posiciona, por lo tanto, como un aporte instrumental dentro de una línea de investigación más amplia, en la que la calidad y disponibilidad de los datos constituyen un paso previo indispensable para cualquier análisis agronómico posterior.

1.2. Motivación

La motivación principal de este trabajo surge de la necesidad concreta de contar con un equipo que permita a los grupos de agronomía realizar mediciones de electrofisiología vegetal en condiciones de campo. Que como ya se mencionó anteriormente no siempre se dispone de soluciones que combinen, en un mismo sistema, portabilidad, autonomía, capacidad de almacenamiento, robustez mecánica, facilidad de uso y adecuación a señales.

Esta situación genera una brecha entre el interés científico por estudiar estas señales y la posibilidad efectiva de adquirirlas en escenarios agronómicos reales. Aun cuando exista una hipótesis interesante sobre la relación entre la actividad eléctrica de la planta y su estado fisiológico, esa hipótesis difícilmente pueda explorarse de manera sistemática si no se cuenta con una herramienta que permita medir de forma confiable, repetible y relativamente sencilla. Desde esta perspectiva, el desarrollo de un dispositivo específico deja de ser un complemento accesorio y pasa a ser una condición habilitante para la propia investigación.

Además del interés técnico, el proyecto presenta una motivación académica y formativa importante. El desarrollo de una solución de este tipo exige integrar conocimientos de distintas áreas de la ingeniería eléctrica y de sistemas embebidos. Por un lado, requiere diseñar una cadena analógica apta para adquirir señales, con criterios adecuados de amplificación, filtrado, referencia y adaptación al conversor A/D. Por otro lado, implica implementar firmware de control, adquisición y almacenamiento, contemplando restricciones de consumo, tiempos de muestreo, operación autónoma y manejo de archivos. A esto se suma el desarrollo de software de visualización y descarga, la integración física del equipo, el diseño mecánico de la envolvente y la resolución de detalles prácticos de uso en campo, como conectores, sellado y manipulación.

La motivación del trabajo, por tanto, no fue únicamente construir un circuito o programar una aplicación aislada, sino desarrollar una solución integral. El proyecto implicó diseñar un sistema completo, desde el contacto físico con la planta hasta la disponibilidad final de los datos para el usuario. Esto incluyó tareas de

diseño electrónico, selección de componentes, confección y montaje de la PCB, desarrollo del firmware, programación de la interfaz de usuario, diseño y fabricación de electrodos, adaptación mecánica de la caja y realización de pruebas experimentales. El carácter integral del trabajo fue uno de sus principales atractivos, ya que permitió abordar un problema real con múltiples niveles de complejidad y con una aplicación concreta para un usuario final claramente identificado.

Por otro lado, también existió una motivación asociada al hecho de trabajar sobre un área aún poco explorada a nivel local. La electrofisiología vegetal no constituye hoy una línea de instrumentación ampliamente desarrollada ni estandarizada, por lo que cada avance en este tipo de proyectos aporta no sólo un resultado puntual, sino también experiencia práctica sobre criterios de diseño, dificultades instrumentales y procedimientos de uso. En ese sentido, el proyecto no sólo busca entregar un equipo funcional, sino también dejar documentado un proceso de desarrollo que pueda servir de base para futuras mejoras, adaptaciones o nuevas investigaciones.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar un prototipo de equipo electrónico portátil para electrofisiología vegetal, capaz de adquirir señales bioeléctricas de plantas en condiciones de campo, minimizando el daño en la zona de contacto con los electrodos para no alterar ni perder la señal. El sistema deberá contar con autonomía energética y operar en dos modos: (i) *Live*, con transmisión y visualización de datos en una computadora vía USB, y (ii) *Logger*, con adquisición y almacenamiento interno durante períodos prolongados sin conexión permanente a una computadora. Asimismo, deberá incluir un software de visualización intuitivo, orientado al uso por agrónomos, que facilite el análisis de las señales adquiridas.

1.3.2. Objetivos específicos

- Relevar y analizar bibliografía sobre señales eléctricas en vegetales, identificando rangos típicos de amplitud, contenido espectral, fuentes de ruido y requerimientos de adquisición.
- Definir requerimientos de diseño a partir de la literatura y de las necesidades del usuario final (modalidades de muestreo, ventanas temporales, frecuencia de muestreo y parámetros configurables).
- Seleccionar y definir los tipos de electrodos y su esquema de conexión para diferentes escenarios de uso (tierra, troncos leñosos, tallos herbáceos y hojas), priorizando mínima invasividad y estabilidad de contacto.
- Diseñar las etapas de acondicionamiento analógico previas al muestreo (amplificación, referencia/offset, protección de entrada), adecuando el rango

dinámico a la etapa de conversión A/D.

- Diseñar e implementar filtros para el acondicionamiento de señal (p. ej., pasabajos y supresión de interferencia de red), acordes a la banda de interés definida.
- Seleccionar componentes críticos del sistema (amplificador de instrumentación/operacionales, conversor A/D, microcontrolador, batería y circuito de carga), justificando su elección por prestaciones, consumo y disponibilidad.
- Diseñar, fabricar y ensamblar la PCB del prototipo, incluyendo criterios de ruteo para señales de bajo nivel, desacople, separación analógico/digital y robustez mecánica.
- Programar el microcontrolador para controlar la adquisición y soportar ambos modos de operación:
  - Modo *Live*: transmisión de muestras a la PC vía USB para visualización en vivo.
  - Modo *Logger*: almacenamiento interno continuo durante días, con manejo de archivos y protección ante cortes/condiciones de memoria llena.
- Desarrollar una aplicación en Python para visualización y descarga de datos, con interfaz simple e intuitiva, orientada a usuarios no especialistas en electrónica.
- Definir e implementar el mecanismo de extracción de datos en modo logger (descarga por USB, lectura de memoria/SD, formato de archivo y procedimiento operativo).
- Integrar el sistema en una envolvente con grado de protección IP6X para uso en campo, contemplando conectores, sellado y facilidad de operación.
- Realizar caracterización y pruebas del prototipo en plantas definidas, evaluando calidad de señal, estabilidad, autonomía, robustez y repetibilidad.
- Elaborar la documentación del sistema (manual de usuario, manual técnico y procedimientos de operación/descarga), asegurando reproducibilidad y trazabilidad del diseño.

1.3.3. Alcance

El proyecto abarca el diseño e implementación de un prototipo funcional (hardware y software) orientado a la adquisición de señales bioeléctricas en plantas, con operación portátil y robusta para campo. Incluye: (i) el diseño del analógico, filtros y conversión A/D; (ii) la selección e integración de microcontrolador, memoria/almacenamiento y sistema de alimentación con batería recargable y soporte de alimentación por USB; (iii) el diseño y fabricación de la PCB y el montaje

del prototipo en una caja con protección IP6X; (iv) el desarrollo del firmware para los modos *Live* y *Logger*; (v) el desarrollo de una aplicación en Python para visualización y descarga; y (vi) pruebas de validación y documentación.

Quedan fuera del alcance: la investigación biológica/agrónoma profunda sobre la interpretación fisiológica de las señales (más allá de lo necesario para definir requerimientos), el desarrollo de electrodos comerciales estandarizados para producción en serie, y la certificación formal del dispositivo (normativas EMC/seguridad) propia de un producto final comercial. El prototipo se orienta a validar el concepto, la arquitectura y el desempeño funcional bajo condiciones realistas de uso en campo.

1.4. Contenido

El documento se organiza del siguiente modo: el Capítulo 2 describe el sistema completo y su operación, baja los conceptos a tierra antes de ser presentados en las siguientes secciones en profundidad; el Capítulo 3 describe el diseño de hardware, cuando decimos hardware nos referimos a electrónica, batería, accesorios IP (conectores, usb y botones) y electrodos; el Capítulo 4 detalla el desarrollo de firmware y software; el Capítulo 5 expone las pruebas realizadas y mediciones relevantes; finalmente, los capítulos posteriores resumen resultados, observaciones y conclusiones, junto con material complementario en anexos.

Capítulo 2

Descripción del sistema

En este capítulo se describe la arquitectura del sistema desarrollado para la adquisición de señales eléctricas en plantas. Se presentan los bloques principales, las conexiones físicas entre sus componentes y el flujo de datos según el modo de operación.

El sistema adquiere señales bioeléctricas mediante electrodos colocados sobre la planta. La señal obtenida es acondicionada en una etapa analógica y luego digitalizada por una placa Arduino MKR Zero. A partir de esta plataforma, el sistema puede operar en dos modos: *Live*, en el cual los datos se envían a una computadora, y *Logger*, en el cual los datos se almacenan localmente en una tarjeta SD.

La Figura 2.1 muestra la arquitectura general del sistema. En ella se identifican la computadora, el usuario, el conector de acceso al exterior, el conversor USB–TTL FT232RL, la placa Arduino MKR Zero, la tarjeta SD, la batería Li-Po y el pulsador de selección de modo. El sistema dispone además de un segundo pulsador para el encendido y apagado, que no se representa en la figura.

La conexión con el exterior se realiza mediante una interfaz USB. Por esta vía se proporciona alimentación y comunicación de datos. La alimentación proveniente del USB energiza la placa Arduino y el módulo FT232RL. La comunicación de datos con la computadora se realiza a través del FT232RL, que adapta la interfaz USB a UART. La placa principal gestiona además el almacenamiento local de los datos en la tarjeta SD.

La Figura 2.2 presenta la interconexión física de los principales elementos del sistema. Los electrodos se colocan sobre la planta y se conectan al equipo mediante el conector externo asociado a la etapa de adquisición. El sistema se encuentra integrado en un gabinete que contiene la electrónica principal, los conectores de acceso y los elementos de interacción con el usuario. En la Figura 2.3 se muestra una fotografía del conjunto, donde se observan el gabinete, los electrodos y la planta.

La interacción del usuario con el sistema se realiza mediante dos mecanismos. El primero corresponde a los pulsadores del equipo, utilizados para el encendido, apagado y selección del modo de operación. El segundo corresponde a la computadora, desde la cual se pueden enviar comandos, visualizar datos en vivo y acceder

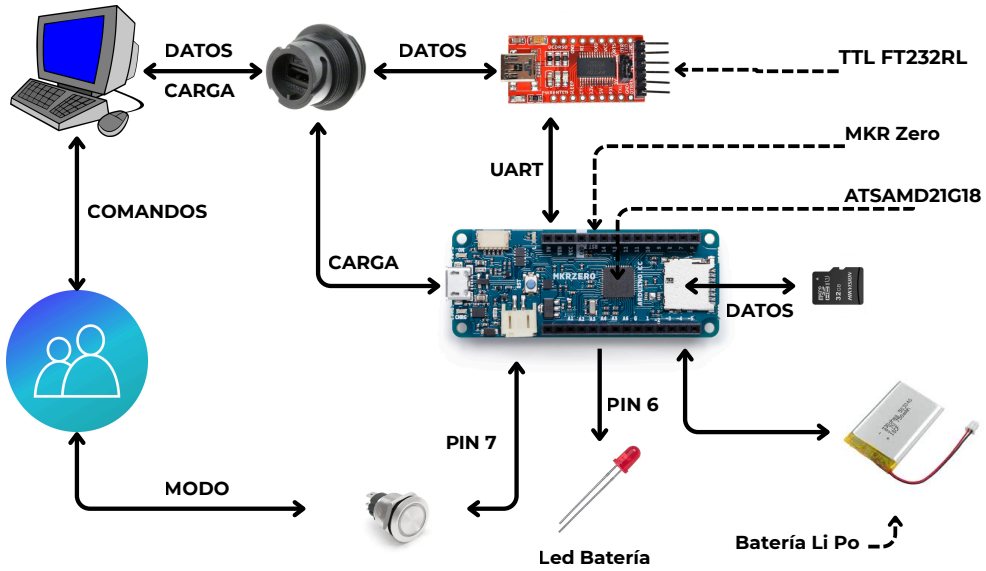


Figura 2.1: Arquitectura general del sistema y conexiones principales en torno a la placa Arduino MKR Zero.

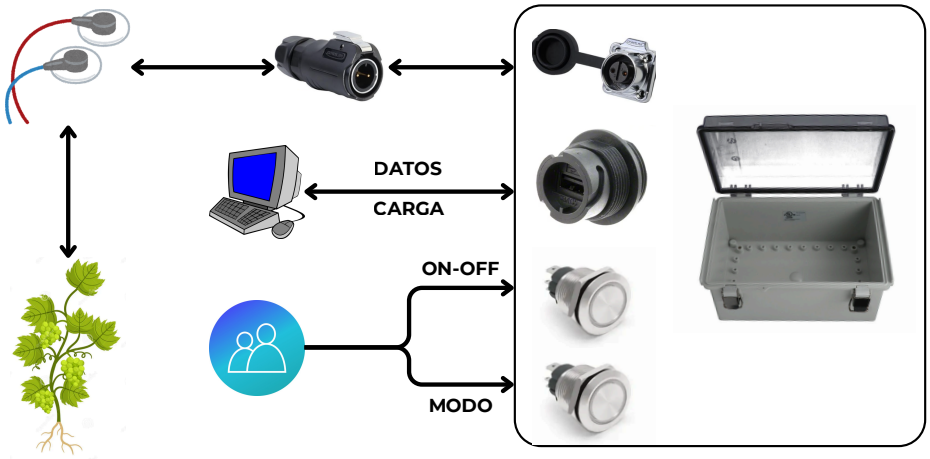


Figura 2.2: Interconexión física de los principales elementos del sistema.



Figura 2.3: Foto del sistema.

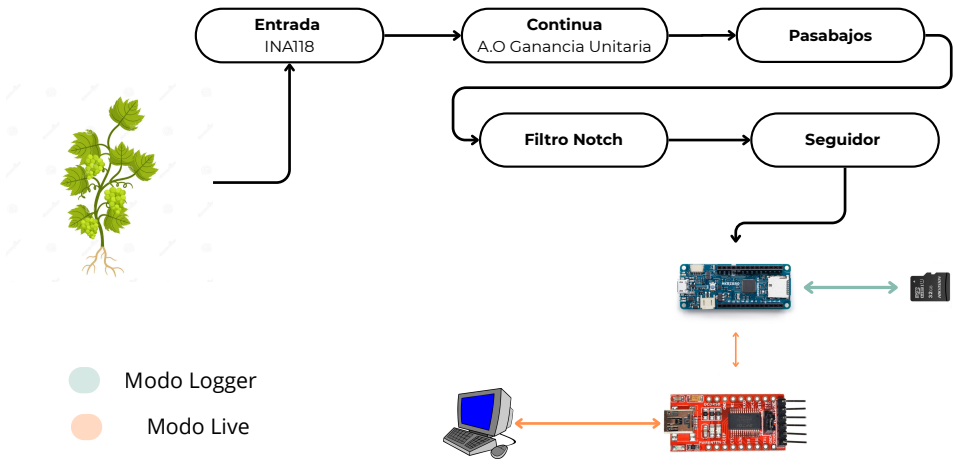


Figura 2.4: Flujo de datos según el modo de operación.

a los registros almacenados. Cuando el sistema se utiliza conectado, la interfaz USB cumple simultáneamente la función de alimentación, carga de batería y comunicación.

Desde el punto de vista funcional, la señal proveniente de la planta atraviesa una cadena de acondicionamiento analógico antes de ser digitalizada. Esta cadena está formada por una entrada basada en un amplificador de instrumentación INA118, una etapa de ajuste de continua con un amplificador operacional en configuración de ganancia unitaria, filtros pasabajos, un filtro notch y una etapa seguidora. La función de esta cadena es amplificar la señal de interés, fijar su nivel de referencia, limitar el ancho de banda y atenuar interferencias antes de su ingreso al conversor analógico-digital del microcontrolador.

La Figura 2.4 resume el flujo de la señal dentro del sistema. En ambos modos de operación, la señal proveniente de la planta atraviesa primero la cadena analógica de acondicionamiento y luego es adquirida por la placa principal. A partir de ese punto, el destino de los datos depende del modo seleccionado.

En modo *Logger*, las muestras se almacenan en la tarjeta SD para su análisis posterior. La recuperación de estos datos puede realizarse mediante la interfaz de comunicación o mediante la extracción de la microSD para su lectura en una computadora.

En modo *Live*, los datos se transmiten en vivo hacia una computadora a través de la interfaz serie. De esta forma, el sistema admite tanto ensayos conectados a una PC como campañas de adquisición autónoma alimentadas mediante batería.

Para ambos modos de operación, una señal adquirida puede visualizarse en la interfaz como se muestra en la Figura 2.5.

Los aspectos relacionados con la operación del sistema se detallan en el manual de usuario incluido en el Anexo A. Allí se especifica la ubicación de los botones

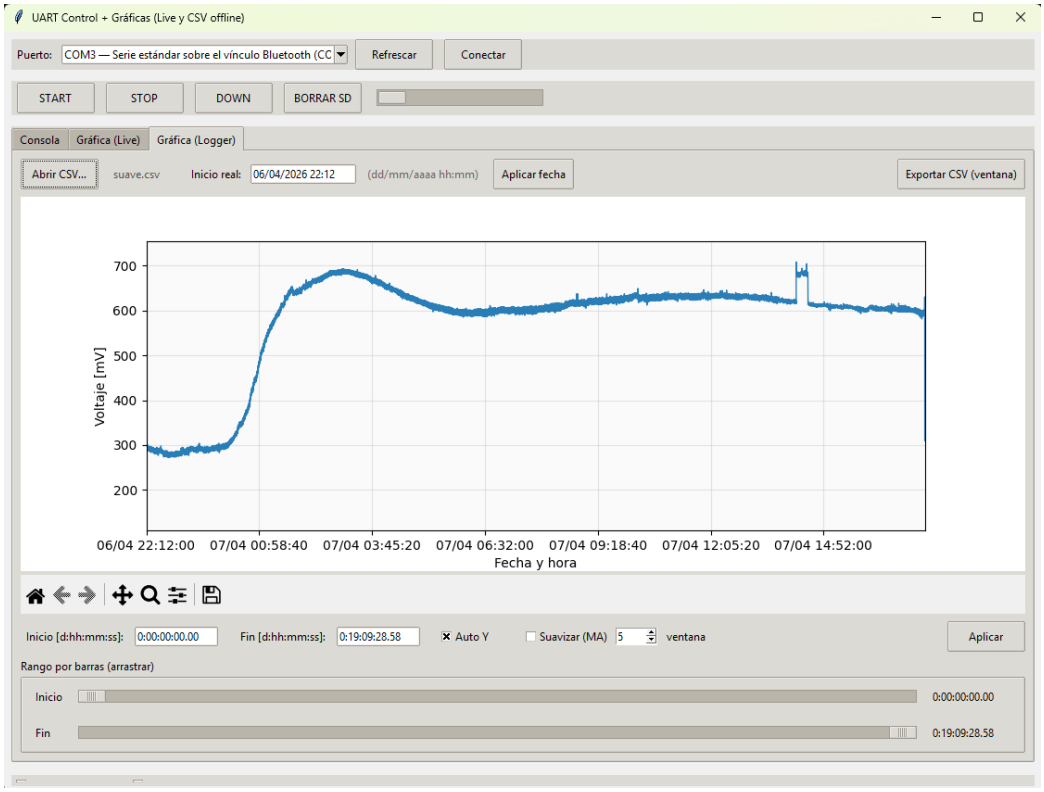


Figura 2.5: Señal capturada en la interfaz.

y conectores, la función de cada uno, los procedimientos de uso y las tareas de mantenimiento.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 3

Hardware

3.1. Introducción

En este capítulo se presenta el diseño de hardware del sistema de adquisición de señales electrofisiológicas vegetales. Se describen los criterios empleados para seleccionar los principales componentes del prototipo, incluyendo la etapa analógica de acondicionamiento, el microcontrolador, la alimentación portátil y los accesorios con grado de certificación de resistencia para uso en campo.

Además, se expone la evolución del diseño a través de tres prototipos sucesivos, mostrando las decisiones adoptadas, los problemas detectados en cada etapa y las modificaciones realizadas hasta alcanzar la arquitectura final. De esta manera, el capítulo no solo documenta la solución implementada, sino también el proceso de desarrollo que permitió converger hacia un sistema más robusto, portable y adecuado para la adquisición de señales en condiciones reales de operación.

3.2. Diseño de la Electrónica

Los principales componentes de la electrónica del sistema son el microcontrolador Arduino MKR Zero, el conversor USB–TTL, los pulsadores, los indicadores LED, la memoria SD y el circuito de acondicionamiento de la señal. En la Figura 3.1 estos elementos se muestran integrados junto con otros componentes del hardware, como los electrodos y la batería.

La figura permite apreciar que el microcontrolador constituye el nodo central del sistema, ya que coordina la adquisición de datos, la gestión de los periféricos y el intercambio de información con los distintos subsistemas.

En modo *Live*, tal como se describió previamente, la PC debe conectarse al puerto USB del gabinete. A través de dicha conexión se alimenta el sistema y, simultáneamente, se posibilita la carga de la batería. Asimismo, la comunicación de datos se realiza mediante el conversor USB–TTL, a través del cual se envían y reciben las tramas correspondientes.

En modo *Logger*, en cambio, no se requiere conexión con la PC ni utilización del conversor USB–TTL. En esta condición, la alimentación proviene de la

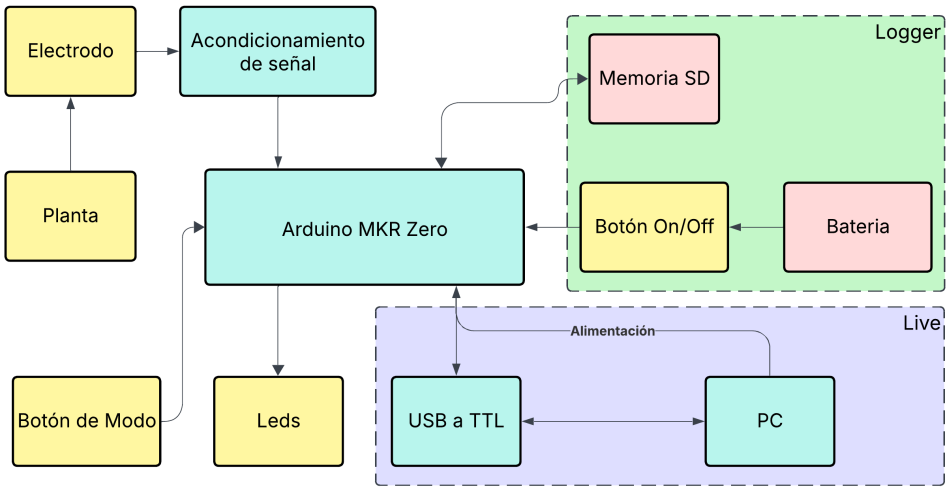


Figura 3.1: Diagrama de la Electrónica.

batería interna, para lo cual el sistema debe permanecer encendido mediante el accionamiento del interruptor correspondiente. Durante este modo de operación, la memoria SD se mantiene activa para el almacenamiento local de los datos adquiridos.

3.2.1. Selección del microcontrolador

La selección del microcontrolador se realizó en función de los requerimientos del sistema: adquisición continua, registro prolongado, portabilidad, bajo consumo y facilidad de desarrollo. Se consideraron tres alternativas: Arduino Nano 33 BLE, Arduino MKR Zero y MSP430G2553.

Arduino Nano 33 BLE

El Arduino Nano 33 BLE incorpora un microcontrolador ARM Cortex-M4 y conectividad Bluetooth. Sin embargo, no dispone de almacenamiento masivo integrado. Para implementar el modo *Logger* habría sido necesario agregar una tarjeta SD externa, junto con su integración eléctrica, mecánica y de firmware. Esto aumentaba la complejidad del sistema sin aportar una ventaja relevante para los objetivos del prototipo.

MSP430G2553

También se consideró el MSP430G2553, utilizado previamente en la asignatura *Sistemas Embebidos para tiempo real*. Este microcontrolador presenta bajo consumo y modos de operación adecuados para sistemas autónomos. No obstante, sus recursos resultan limitados para esta aplicación: memoria Flash y RAM reducidas, ADC de 10 bits y ausencia de almacenamiento masivo integrado. Su uso

habría requerido agregar memoria externa o una tarjeta SD, junto con su gestión, incrementando el tiempo de desarrollo y la complejidad del sistema.

Arduino MKR Zero

La plataforma seleccionada fue Arduino MKR Zero. Esta placa integra lector de tarjeta microSD y circuito de carga para baterías Li-Po, lo que simplifica la implementación del sistema y evita el uso de módulos externos para almacenamiento y alimentación.

El microcontrolador SAMD21 dispone de recursos suficientes para la adquisición, el registro de datos, la comunicación por USB y la implementación de los modos de operación definidos. Además, el entorno de desarrollo utilizado permitió reducir el tiempo de implementación del prototipo.

Por estos motivos, se seleccionó el Arduino MKR Zero para la implementación del sistema.

3.2.2. Selección del amplificador de instrumentación

El amplificador de instrumentación INA118U fue seleccionado como etapa principal de amplificación diferencial del sistema. Si bien este dispositivo es ampliamente utilizado en aplicaciones de adquisición de señales biomédicas, como electrocardiografía (ECG), su elección en este trabajo se fundamenta principalmente en la adecuación de sus prestaciones a los requerimientos de la señal electrofisiológica vegetal.

La adquisición de este tipo de señales impone exigencias particulares sobre la etapa de entrada, ya que se trata de señales diferenciales que pueden tener amplitud cambiante, susceptibles al ruido externo y obtenidas a partir de electrodos con impedancia relativamente elevada. En este contexto, la elección del INA118U se justificó por las siguientes características técnicas:

- **Alto rechazo de modo común (CMRR)**, lo que permite atenuar interferencias externas y perturbaciones presentes simultáneamente en ambas entradas del amplificador.
- **Bajo nivel de ruido**, condición necesaria para no degradar señales diferenciales de pequeña amplitud.
- **Ganancia configurable mediante una única resistencia externa**, lo que simplifica el diseño del circuito y permite ajustar la amplificación de manera sencilla según las necesidades de la aplicación.
- **Alta impedancia de entrada**, característica adecuada para la adquisición de señales provenientes de electrodos, minimizando el efecto de carga sobre la fuente medida.

Adicionalmente, el encapsulado SOIC-8 del dispositivo facilitó su implementación sobre placa de circuito impreso y su montaje durante la etapa de prototipado.

En conjunto, estas características hacen que el INA118U constituya una solución adecuada para la amplificación de señales diferenciales en la aplicación desarrollada.

3.2.3. Selección del amplificador operacional auxiliar

Para las etapas auxiliares de acondicionamiento de señal, tales como buffer, adaptación del nivel de continua y filtro notch, se evaluaron distintas alternativas de amplificadores operacionales con alimentación simple y entrada/salida *rail-to-rail*. Entre las opciones consideradas se analizaron los modelos MCP6002, MCP6022, OPA2333 y ADA4505.

La selección final recayó sobre el ADA4505. Esta decisión no respondió a un único parámetro, sino al compromiso global entre precisión, estabilidad, compatibilidad con alimentación simple, facilidad de integración y costo razonable para la aplicación desarrollada.

Desde el punto de vista técnico, el ADA4505 presenta características especialmente adecuadas para el tratamiento de señales de baja amplitud y baja frecuencia. Según la hoja de datos del fabricante, el dispositivo opera con fuente simple entre 1,8 V y 5 V, posee entrada y salida *rail-to-rail*, corriente de polarización de entrada típica del orden de 0,5 pA, corriente de alimentación menor a 10 µA por amplificador y una tensión de offset máxima del orden de 3 mV. Estas prestaciones resultan particularmente convenientes en un sistema donde el error de continua y la carga sobre etapas de alta impedancia deben mantenerse reducidos. [2]

En comparación, el MCP6002 también admite alimentación simple entre 1,8 V y 5,5 V y dispone de entrada/salida *rail-to-rail*. Sin embargo, presenta una tensión de offset típicamente mayor, del orden de milivoltios, por lo que, si bien constituye una alternativa económica y simple de implementar, resulta menos conveniente cuando se busca minimizar errores de continua en señales pequeñas. [17]

Por su parte, el MCP6022 ofrece mejores prestaciones dinámicas que el ADA4505. Sin embargo, su tensión de offset sigue siendo superior a la del ADA4505 y su corriente de polarización de entrada es mayor, por lo que resulta menos conveniente para esta aplicación específica. [16]

También se consideró el OPA2333, un amplificador operacional de muy alta precisión y extremadamente bajo offset. Sin embargo, esta alternativa fue descartada debido a su mayor costo relativo y a que, durante la etapa inicial del proyecto, no resultó la opción más conveniente desde el punto de vista práctico para simulación, disponibilidad e integración al prototipo. En consecuencia, si bien el OPA2333 representaba una opción técnicamente competitiva, no resultó la más adecuada para la solución final adoptada.

En síntesis, el ADA4505 fue seleccionado no porque superara a todas las alternativas en cada parámetro individual, sino porque ofrecía el mejor equilibrio entre precisión, muy baja corriente de polarización, compatibilidad con alimentación simple y viabilidad práctica de implementación para el sistema desarrollado.

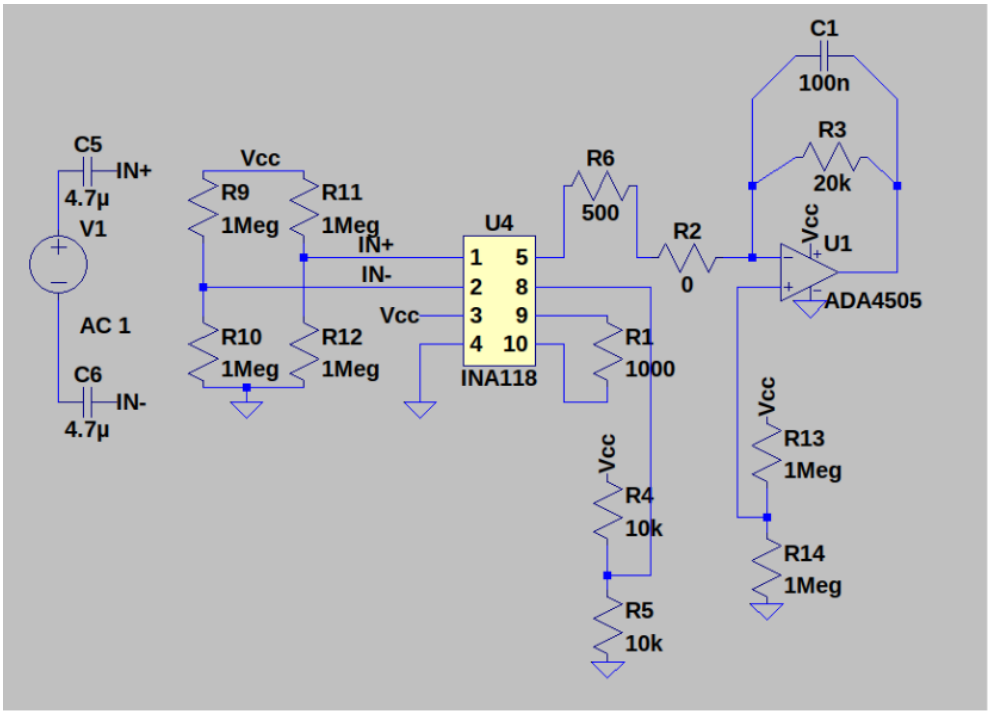


Figura 3.2: Esquemático eléctrico del Prototipo 1. Se observa la etapa diferencial implementada con INA118 (siendo iN- y iN+ sus entradas) y la etapa de filtrado activo realizada con ADA4505.

3.2.4. Selección de conversor USB a TTL

Para comunicar el microcontrolador con una PC, se hizo uso de un modulo conversor USB-TTL encontrado en plaza con el chip FT232RL [11]. El chip soporta hasta USB 2.0 full speed.

Este modulo en particular tiene la posibilidad de operar en $5\,V$ y $3,3\,V$. Dado que nuestro microcontrolador opera $3{,}3\,V$ se eligió ese modo de funcionamiento.

3.2.5. Prototipo 1

El primer prototipo se desarrolló a partir de una topología clásica de adquisición de biopotenciales, basada en una etapa de amplificación diferencial seguida por una etapa adicional de filtrado y ganancia.

A partir de esta base conceptual se implementó el **Prototipo 1** presentado en la Figura 3.2. En esta versión se utilizó un amplificador de instrumentación **INA118** como etapa de entrada y un amplificador operacional **ADA4505** para la etapa de filtrado activo. El INA118 se configuró con una ganancia $G_1 = 20$, mientras que el ADA4505 se empleó para implementar un filtro pasabajos activo con una ganancia adicional $G_2 = 40$.

En esta primera aproximación se decidió polarizar ambas entradas a una tensión continua de $V_{CC}/2$, es decir, a la referencia de mitad de alimentación, con el

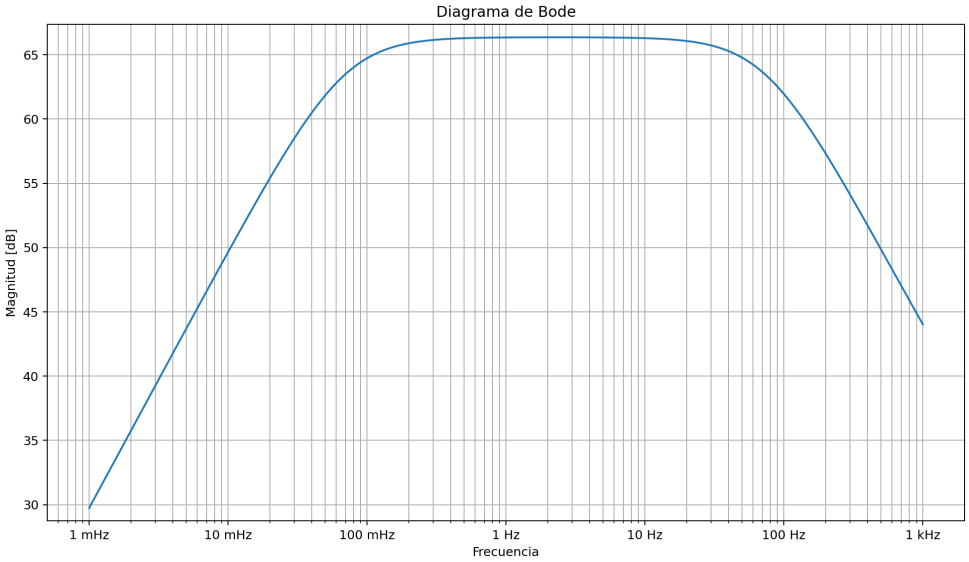


Figura 3.3: Respuesta en frecuencia del filtro pasabajos del Prototipo 1.

objetivo de centrar la señal dentro del rango dinámico disponible para el sistema.

Además, se incorporaron capacitores de desacople en serie con las entradas para desacoplar la componente continua asociada a la planta.

En síntesis, el Prototipo 1 quedó conformado por:

- Un amplificador de instrumentación INA118 con ganancia $G_1 = 20$.
- Un filtro pasabajos activo implementado con ADA4505, con ganancia $G_2 = 40$.

La etapa de filtrado fue diseñada con puentes configurables que permitían modificar la frecuencia de corte, habilitando valores aproximados de 100 Hz, 40 Hz, 20 Hz y 2 Hz.

El diagrama de bode se presenta en la Figura 3.3. Los resultados experimentales mostraron las siguientes prestaciones:

- Polo de alta frecuencia regulable entre 13 y 310 Hz.
- Polo de baja frecuencia de 68 mHz.
- Ganancia regulable entre 40 y 67 dB.

Este prototipo constituyó el primer paso en el desarrollo del sistema de adquisición, ya que permitió validar la estrategia general de amplificación, polarización y filtrado.

Sin embargo, una vez ensamblado y ensayado en condiciones reales, se observó que la combinación de los capacitores de desacople con el divisor resistivo utilizado

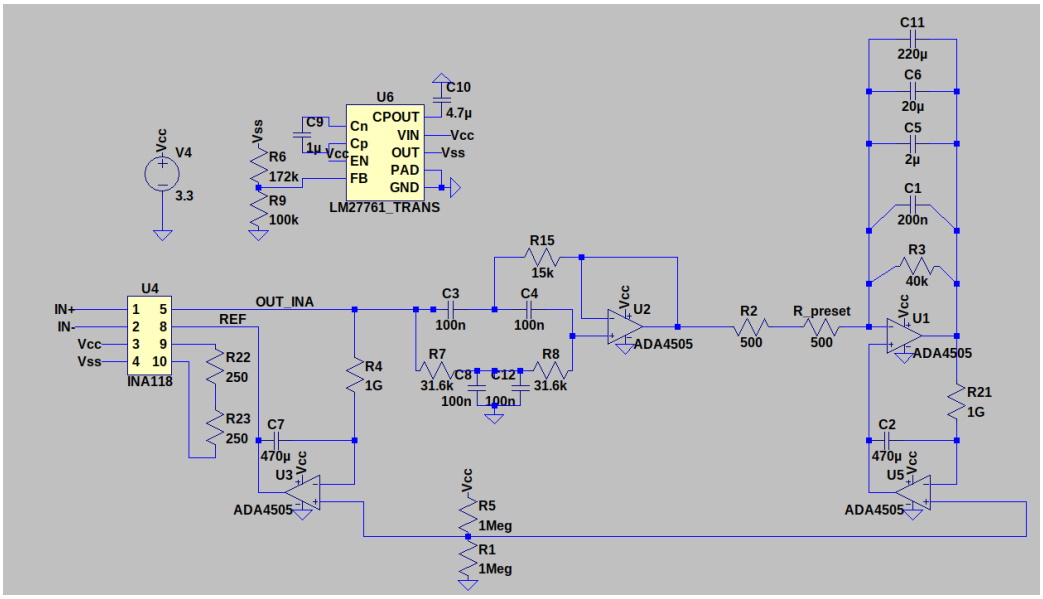


Figura 3.4: Esquemático correspondiente al Prototipo 2.

para fijar la componente continua introducía un efecto de filtrado no deseado en bajas frecuencias.

En particular, el acoplamiento en alterna, junto con las resistencias de polarización de entrada, forma un filtro pasaaltos cuya frecuencia de corte depende del producto RC. Como consecuencia, parte del contenido espectral de muy baja frecuencia, relevante en señales electrofisiológicas vegetales, podía resultar atenuado o incluso eliminado.

A esta limitación se sumó la presencia de interferencia de red a 50 Hz, la cual deterioró significativamente la calidad de las mediciones realizadas sobre la planta e imposibilitó obtener registros confiables.

En consecuencia, si bien el Prototipo 1 permitió validar la arquitectura general de adquisición, también puso en evidencia que la estrategia de acondicionamiento de entrada no era adecuada para preservar de forma robusta la información de baja frecuencia de interés. Por esta razón, en la siguiente etapa de diseño se replanteó tanto el tratamiento de la componente continua como la inmunidad frente al ruido de red.

3.2.6. Prototipo 2

Dado que en el **Prototipo 1** no fue posible obtener mediciones relevantes sobre la planta, en el **Prototipo 2** se optó por desarrollar una nueva placa que permitiera desacoplar etapas y ensayar distintas estrategias de acondicionamiento de señal.

Como se muestra en la Figura 3.4, en esta versión se eliminaron los capacitores de desacople y el divisor resistivo de polarización en la entrada, de modo que la señal proveniente de la planta se aplicara directamente a las entradas diferenciales

del INA118. Con esta modificación se buscó evitar el efecto de filtrado pasaaltos introducido en el prototipo anterior y preservar el contenido de muy baja frecuencia presente en la señal vegetal.

Asimismo, se mantuvo la posibilidad de ajustar la etapa de filtrado pasabajos mediante puentes y se conservó un potenciómetro de ajuste entre etapas.

La principal modificación conceptual de este prototipo consistió en trasladar el control de la componente continua desde la entrada hacia la referencia interna del amplificador de instrumentación. Para ello, se tomó como base la recomendación de la hoja de datos del **INA118**, donde se propone utilizar el pin *REF* para fijar el nivel de referencia de la salida. Mediante esta estrategia, la salida del amplificador podía desplazarse al valor deseado, en este caso $V_{CC}/2$, sin necesidad de introducir acoplamiento capacitivo en la entrada.

De esta manera, el sistema buscaba:

- Preservar el contenido de muy baja frecuencia de la señal vegetal.
- Evitar la atenuación de componentes lentas producida por el acoplamiento capacitivo.
- Fijar el nivel de continua de la salida a través del pin *REF* del INA118.

Por otro lado, el análisis de las condiciones de alimentación del INA118 mostró que este dispositivo no podía operar adecuadamente con una fuente simple de $3{,}3\,V$. Por esta razón, se incorporó el inversor LM27761 [26], con el fin de generar una alimentación diferencial de $\pm 3{,}3\,V$.

Adicionalmente, se incorporó un filtro notch centrado en 50 Hz, correspondiente a la frecuencia de la red eléctrica nacional, con el fin de atenuar la interferencia proveniente de la red.

Finalmente, también se añadió una etapa de regulación de continua a la salida del sistema, para asegurar que la señal quedara efectivamente centrada en $V_{CC}/2$.

En conjunto, esta arquitectura representó una mejora conceptual respecto al Prototipo 1, ya que trasladaba el control del nivel de continua hacia la referencia interna del amplificador y evitaba alterar el contenido de baja frecuencia en la entrada. Sin embargo, como se detalla a continuación, varias de las dificultades que se pretendían resolver persistieron debido a errores conceptuales en el diseño.

Análisis crítico del Prototipo 2

Uno de los primeros inconvenientes detectados estuvo asociado a la generación de la tensión negativa. El circuito integrado seleccionado para este propósito, el LM27761, presentaba un encapsulado poco conveniente para el proceso de prototipado manual adoptado en el proyecto. En la práctica, su montaje resultó dificultoso y no se logró obtener a partir de él una alimentación diferencial funcional. Debido a esta limitación, durante los ensayos fue necesario recurrir a una fuente externa para generar la tensión negativa requerida.

El principal problema del prototipo estuvo vinculado a los dos lazos de realimentación destinados a fijar la componente continua de la señal. Durante el diseño

no se consideró adecuadamente el tiempo de establecimiento asociado a los filtros RC empleados en dichos lazos.

La dinámica de esta realimentación estaba gobernada por la constante de tiempo

$$\tau = R \cdot C$$

donde $R \approx 1\,\mathrm{G\Omega}$ y $C = 470\,\mu\mathrm{F}$, por lo que se obtenía

$$\tau = 4{,}7 \times 10^5\,\mathrm{s} \approx 130\,\mathrm{horas}$$

Este valor resulta completamente incompatible con un sistema experimental que requiere alcanzar condiciones estables en escalas de tiempo del orden de minutos. En consecuencia, cualquier variación en la señal de entrada persistía durante períodos excesivamente prolongados, impidiendo una corrección efectiva de la componente continua. En la práctica, esto provocó que el lazo de realimentación no lograra estabilizar el nivel de referencia de manera robusta ni reproducible.

Durante la etapa de diseño también se intentó analizar el comportamiento del circuito mediante simulaciones en *LTspice*. Sin embargo, no fue posible obtener resultados confiables debido a problemas de convergencia numérica. En particular, el modelo del inversor encargado de generar la alimentación negativa no presentaba un comportamiento satisfactorio en simulación, y los lazos de realimentación de muy baja frecuencia introducían constantes de tiempo extremadamente grandes, dificultando aún más la convergencia. Como consecuencia, no fue posible validar el comportamiento dinámico del circuito mediante simulación antes de la construcción del prototipo.

Cabe señalar que el diseño permitía desacoplar estos lazos de realimentación, lo que hizo posible continuar con algunas pruebas parciales a pesar de estas limitaciones.

Por otra parte, el potenciómetro ubicado entre la salida del INA118 y la entrada del ADA4505 fue concebido originalmente como un elemento de ajuste de nivel. Sin embargo, al encontrarse en un nodo que transportaba simultáneamente la señal amplificada y la componente de continua, cualquier modificación de su posición afectaba de manera simultánea la ganancia efectiva del sistema y el punto de operación de la etapa siguiente.

En un sistema alimentado con fuente simple y referenciado a $1{,}65\,V$, el punto de operación resulta crítico, ya que un desplazamiento del nivel de continua puede acercar la señal a los rieles de alimentación, reduciendo el margen dinámico disponible y provocando saturación en la etapa operacional.

Durante los ensayos se observó, además, que el punto de operación adecuado solo se obtenía para una posición muy específica del potenciómetro. Pequeñas variaciones en su ajuste provocaban desplazamientos significativos del nivel de continua, lo que hacía que el sistema fuera difícil de calibrar y poco estable en el tiempo.

En otras palabras, el prototipo perdió independencia entre dos grados de libertad fundamentales: el control de ganancia y la polarización del circuito. Desde

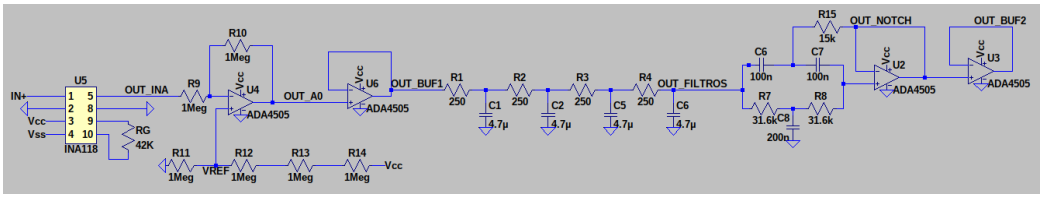


Figura 3.5: Esquemático del Prototipo 3 a nivel de simulación en LTSpice.

el punto de vista del diseño analógico, ambos parámetros deberían resolverse mediante mecanismos independientes. En este caso, al quedar acoplados en un mismo nodo eléctrico, se obtuvo una arquitectura particularmente sensible a ajustes manuales y, en consecuencia, poco robusta para un sistema experimental.

3.2.7. Prototipo 3 (Final)

El **Prototipo 3** corresponde a la versión final del sistema, en la cual se resolvieron los principales inconvenientes identificados en los prototipos anteriores. Su esquemático y su implementación se presentan en las Figuras 3.5 y 3.6, respectivamente. Asimismo, en la Tabla 3.1 se detallan los componentes utilizados en esta versión del circuito, los nombres corresponden de la Figura 3.6, que es el circuito diseñado en Fusion, siendo este un software de diseño asistido por computadora (CAD) que integra herramientas para el diseño de placas de circuito impreso (PCB), permitiendo la definición del esquemático y el enrutado del circuito [9].

En primer lugar, se reemplazó el inversor de tensión utilizado en la versión anterior por el integrado **LTC1044** [3]. Este cambio representó una mejora tanto en simulación como en implementación práctica. Por un lado, el modelo empleado en LTSpice presentó un comportamiento numérico a nivel de simulación más estable, los anteriores tomaban mucho tiempo para converger. Por otro, su encapsulado de mayor tamaño resultó más conveniente para el montaje manual sobre PCB.

De acuerdo con las recomendaciones del fabricante, se dispusieron capacitores entre los terminales $CAP+$ y $CAP-$, así como en la salida V_{out}. Para esta implementación se seleccionaron C_{10} y C_{11} con igual valor, de $4{,}7\,\mu F$, manteniendo simetría entre ambos componentes y empleando valores del mismo orden de magnitud que los sugeridos en la hoja de datos. Por otra parte, en el pin OSC se dejó prevista la posibilidad de incorporar el capacitor C_5 en caso de requerirse un ajuste futuro de la frecuencia de oscilación interna del inversor, aunque durante el desarrollo no fue necesario utilizarlo.

Por otro lado, se eliminó completamente la realimentación de continua aplicada sobre el INA118 en versiones anteriores. En su lugar, el ajuste de offset se realiza en una etapa posterior, mediante un amplificador de ganancia unitaria que desplaza la señal al nivel $V_{CC}/2$. De esta forma, se desacopló el control de continua de la etapa principal de amplificación diferencial.

Este amplificador se configura como una etapa inversora de ganancia unitaria para la señal proveniente de la planta. Sin embargo, respecto de la entrada no inversora, la misma configuración presenta una ganancia igual a 2. Por este motivo,

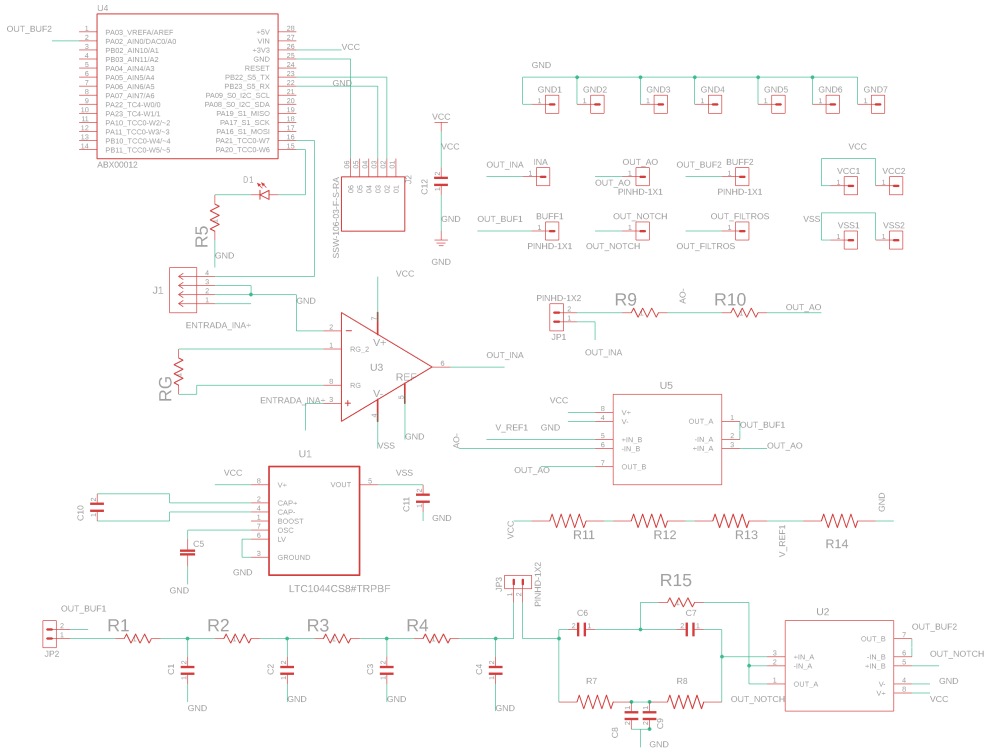


Figura 3.6: Implementación del Prototipo 3 (final).

se aplica una tensión de referencia de $V_{CC}/4$ en la entrada no inversora, obtenida mediante una cadena resistiva. De esta forma, la contribución de dicha referencia a la salida es:

$$V_{\text{out,ref}} = 2 \cdot \frac{V_{CC}}{4} = \frac{V_{CC}}{2}$$

Adicionalmente, se incorporaron seguidores de tensión entre las distintas etapas del circuito, específicamente a la salida de la etapa de ajuste de offset y a la salida del filtro notch, que es antes del ADC. Esta decisión permite aislar eléctricamente las etapas entre sí, minimizando interacciones indeseadas asociadas a las impedancias de entrada y salida.

En cuanto a la configuración general, se optó por fijar tanto la ganancia como la frecuencia de corte del sistema. Esta decisión se fundamenta en la revisión bibliográfica sobre electrofisiología vegetal [10], donde se reporta que las señales de interés presentan dinámicas de muy baja frecuencia, del orden de horas. En consecuencia, no resultaba necesario mantener una arquitectura con múltiples configuraciones de filtrado o grandes ganancias ajustables, como en etapas anteriores del desarrollo.

En particular, la bibliografía consultada utiliza un amplificador de ganancia unitaria. En este trabajo, tomando el experimento de la bibiografía, se adoptó una ganancia total cercana a la unitaria. Suficiente para acondicionar la señal sin sobreamplificar componentes no deseadas. La ganancia total del sistema quedó

Ref.	Valor
R_1	$250\,\Omega$
R_2	$250\,\Omega$
R_3	$250\,\Omega$
R_4	$250\,\Omega$
R_5	$330\,\Omega$
R_G	$42\,k\Omega$
R_7	$31{,}6\,k\Omega$
R_8	$31{,}6\,k\Omega$
R_9	$1\,M\Omega$
R_{10}	$1\,M\Omega$
R_{11}	$1\,M\Omega$
R_{12}	$1\,M\Omega$
R_{13}	$1\,M\Omega$
R_{14}	$1\,M\Omega$
R_{15}	$15\,k\Omega$

(a) Resistencias

Ref.	Valor
C_1	$4{,}7\,\mu F$
C_2	$4{,}7\,\mu F$
C_3	$4{,}7\,\mu F$
C_4	$4{,}7\,\mu F$
C_5	-
C_6	$100\,nF$
C_7	$100\,nF$
C_8	$100\,nF$
C_9	$100\,nF$
C_{10}	$4{,}7\,\mu F$
C_{11}	$4{,}7\,\mu F$
C_{12}	$4{,}7\,\mu F$

(b) Capacitores

Ref.	Valor
U_1	LTC1044
U_2	ADA4505
U_3	INA118
U_4	ARDUINO MKR ZERO
U_5	ADA4505

(c) Circuitos integrados

Tabla 3.1: Listado de componentes del Prototipo 3.

determinada por el INA118 a través de la resistencia R_G.

$$G = 1 + \frac{50\,k\Omega}{R_G} \tag{3.1}$$

La respuesta en frecuencia simulada del sistema se presenta en la Figura 3.7. En ella se observa que el comportamiento en altas frecuencias está determinado por una cascada de cuatro filtros RC pasabajos, los cuales introducen una pendiente total aproximada de $-80\,\mathrm{dB/dec}$.

La incorporación de estas cuatro etapas se justificó a partir del análisis de la simulación. Si bien el filtro notch atenúa la interferencia de red en 50 Hz, se verificó que, para frecuencias superiores, la ganancia del sistema volvía a incrementarse. Este comportamiento resultaba indeseable, ya que generaba un nuevo máximo en torno a 100 Hz. En consecuencia, se añadieron varias etapas RC en cascada con el objetivo de suprimir ese realce y reforzar la atenuación fuera de la banda de interés.

Simulaciones

Tal como se aprecia en la Figura 3.7, cada una de las etapas de filtrado aporta atenuación adicional en esa región de frecuencias. En conjunto, las cuatro etapas

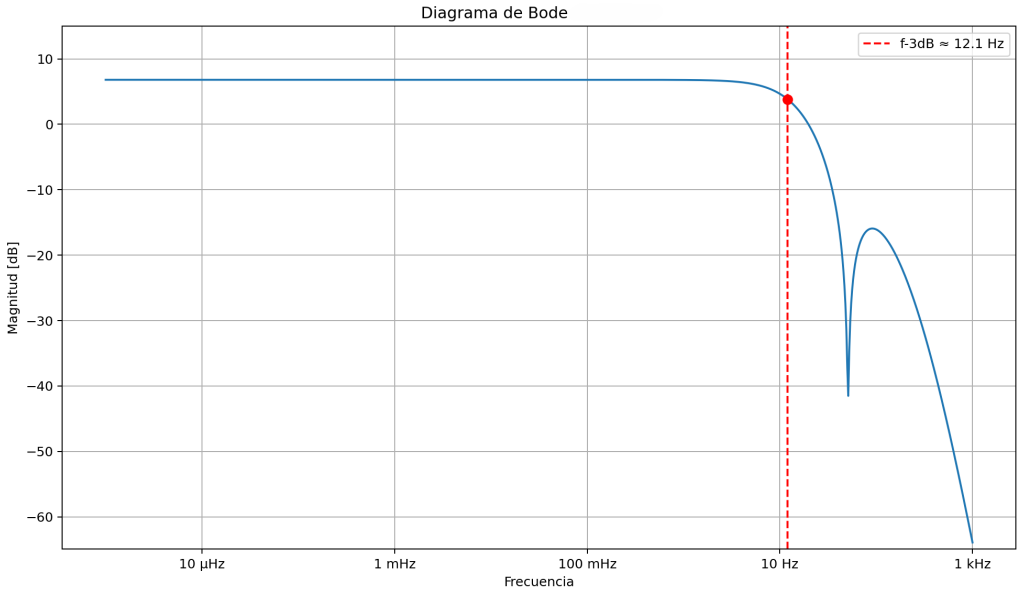


Figura 3.7: Diagrama de Bode del Prototipo 3 simulado en LTSpice.

Ganancia total	$2{,}19\,V/V$
Frecuencia de corte	$12{,}1\,\mathrm{Hz}$
Rechazo a $50\,\mathrm{Hz}$	$-41\,\mathrm{dB}$
Tensión de salida (offset)	$1{,}65\,V$

Tabla 3.2: Especificaciones teóricas del Prototipo 3.

permiten reducir aproximadamente en $-15\,\mathrm{dB}$ la ganancia alrededor de $100\,\mathrm{Hz}$, correspondiente al máximo posterior a la acción del notch.

Asimismo se presenta la Tabla 3.2, donde se listan algunas de las especificaciones de interés para el circuito.

La Figura 3.8 muestra la respuesta temporal del sistema ante una señal de entrada senoidal de $10\,\mathrm{mV}$ a $1\,\mathrm{Hz}$. Se aprecia que la señal esta centrada en $V_{cc}/2$ y la amplificación coincide con la reportada previamente.

PCB

En la Figura 3.9 se presenta la PCB diseñada para implementar el prototipo descripto anteriormente, corresponde a la Figura 3.6 . En ella además de la electrónica se dejó lugar para conectar el conversor USB-TTL, microcontrolador y el conector JST. También se agregaron bastantes pines para facilitar el debug, por este mismo motivo también se dejaron puentes para poder desacoplar etapas.

Por lo general a la hora de diseñar una PCB se trata de reducir el area para que sea menos costoso. En nuestro caso ya que la soldadura sería manual se decidió dejar bastante espacio entre componentes para facilitar la soldadura.

Las medidas experimentales sobre la PCB se detallan en la Sección 5.1.

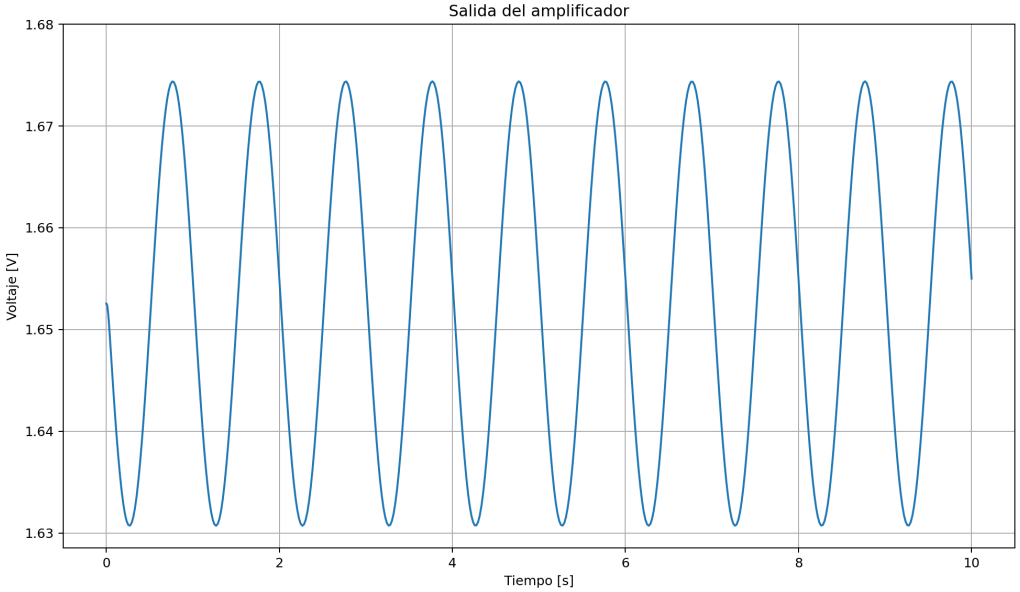


Figura 3.8: Salida del amplificador para una entrada de $10\,\text{mV}$ a $1\,\text{Hz}$.

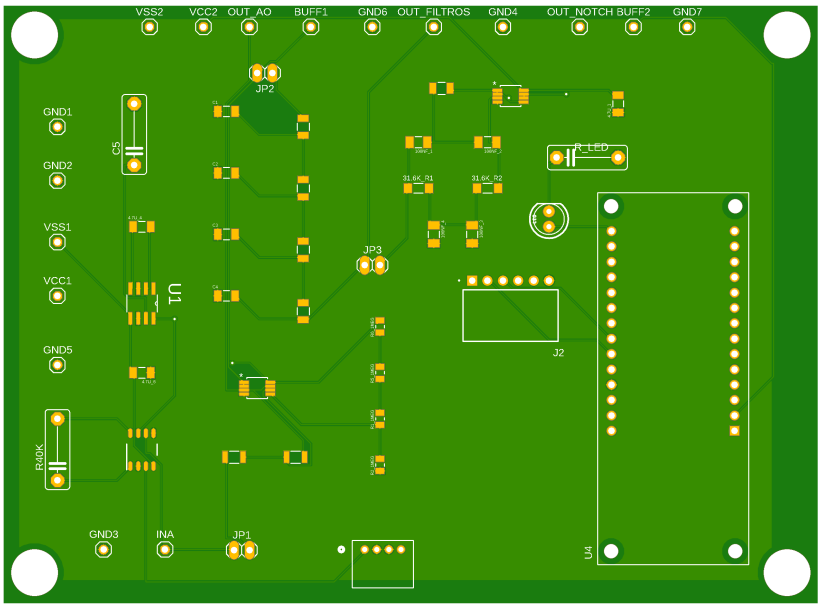


Figura 3.9: PCB diseñada para el Prototipo 3.

3.3. Selección de batería

La necesidad de incorporar una batería surge de dotar al sistema de portabilidad y permitir jornadas de adquisición en condiciones de campo. Para ello, se aprovechó que el microcontrolador elegido incorpora un circuito integrado especializado para la carga de baterías Li-Po de 3,7 V, con una corriente de carga predefinida de 350 mA [1]. Esto simplifica el diseño del prototipo, ya que evita el desarrollo de un módulo externo de carga y protección, y reduce los riesgos asociados a la operación en campo, donde se requiere una recarga práctica mediante conexión USB a una notebook o a un cargador estándar.

Asimismo, dicho circuito está configurado con un tiempo máximo de carga de 4 horas, tras el cual pasa automáticamente a modo de reposo. En consecuencia, la carga transferida en una ronda completa es, como máximo, del orden de 1400 mAh. Esto implica que una batería de capacidad superior a ese valor no necesariamente se cargará por completo en una única ronda. En este trabajo se seleccionó una batería Li-Po de 3,7 V y 1800 mAh, en lugar de una de 1400 mAh, principalmente por disponibilidad en el mercado.

En el Capítulo 5 se presentan la estimación y la medición del consumo del sistema en sus distintos modos de operación, a partir de las cuales se evalúa la autonomía efectiva obtenida con la batería seleccionada.

3.4. Selección de accesorios IP

Uno de los requisitos de diseño del prototipo fue asegurar un grado mínimo de protección IP65 para los elementos externos del sistema. De acuerdo con la norma IEC 60529 [13], este grado de protección implica estanqueidad total frente a polvo (primer dígito 6) y protección frente a chorros de agua proyectados desde cualquier dirección (segundo dígito 5). Este requisito resulta especialmente relevante debido a que el dispositivo fue concebido para operar en condiciones de campo, donde puede estar expuesto a humedad, lluvia, polvo o salpicaduras.

Para cumplir con este requerimiento, se seleccionaron componentes industriales con grado de protección igual o superior al especificado.

La electrónica del sistema se aloja en una caja estanca de policarbonato con grado de protección IP67, equipada con junta perimetral de sellado como se muestra en la Figura 3.10. Este tipo de gabinete se utiliza comúnmente en instalaciones eléctricas exteriores y proporciona protección completa frente al ingreso de polvo y contra inmersión temporal en agua. Donde para fijar la PCB y la batería dentro de la caja, se diseñó en OnShape [19] un soporte interno atornillable. El modelo se muestra en la Figura 3.11 y fue fabricado posteriormente mediante impresión 3D con filamento PETG.

La conexión de los electrodos al sistema se realiza mediante un conector circular industrial tipo aviación, modelo EW-LP20, Figura 3.12, con dos contactos eléctricos. Este conector presenta un grado de protección IP68 en condición acoplada, garantizando estanqueidad frente al ingreso de agua y polvo. El conector utiliza un sistema de acoplamiento con inserción automática y un mecanismo de autoblo-

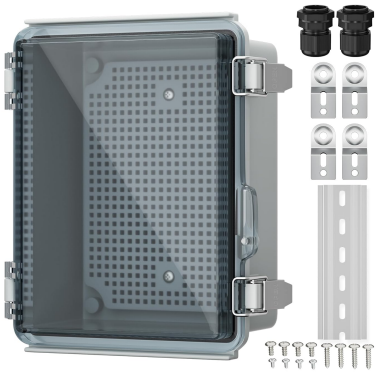


Figura 3.10: Caja IP67 seleccionada.

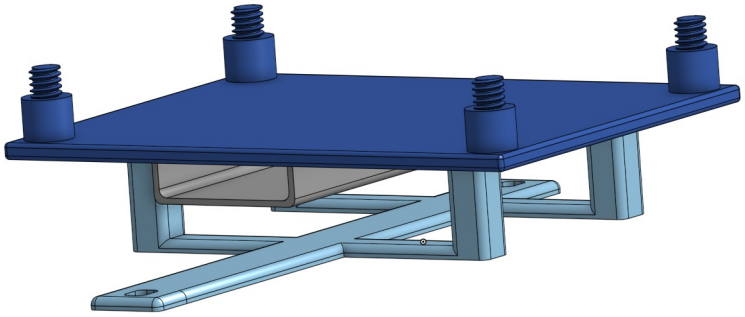


Figura 3.11: Modelo 3D del soporte interno en herramienta OnShape.

queo, con sellado mediante un anillo de silicona. Sus contactos están fabricados en aleación de cobre con recubrimiento de oro para reducir la resistencia de contacto (menor a 5 mΩ) y mejorar la estabilidad eléctrica en aplicaciones de señal. El conector admite corrientes nominales de hasta 5 A y tensiones de operación de 250 V_{RMS}, con una resistencia de aislamiento superior a 1000 $M\Omega$. Además, presenta una durabilidad mecánica superior a 1900 ciclos de acoplamiento y desacoplamiento, lo que lo hace adecuado para aplicaciones experimentales donde se requieren múltiples conexiones y desconexiones.

Para la interfaz de comunicación con la computadora se emplea un conector USB tipo A montado en panel con grado de protección IP67, Figura 3.13. Este conector cumple con la especificación USB 2.0 y posee resistencia de contacto máxima de 30 $m\Omega$, resistencia de aislamiento mínima de 1000 $M\Omega$ y corriente nominal de 1,5 A. El sellado se logra mediante una junta de silicona entre el conector y el panel de la caja, permitiendo mantener la estanqueidad del sistema incluso en entornos exteriores.

Finalmente, el sistema incorpora pulsadores metálicos antivandalismos presentados en la Figura 3.14 con grado de protección IP65, diseñados para operar en ambientes exteriores. Estos pulsadores presentan un sellado interno que evita la



Figura 3.12: Conector para electrodos seleccionado.



Figura 3.13: Conector hembra USB tipo A seleccionado.

penetración de polvo y agua, asegurando la integridad del sistema incluso bajo condiciones ambientales adversas.

En conjunto, la selección de estos componentes permite que el sistema completo mantenga un grado de protección ambiental igual o superior a IP65, cumpliendo así con los requisitos establecidos para su operación en campañas de medición en



Figura 3.14: Pulsador metálicos seleccionado.

campo.

3.5. Electrodos

El sistema de medición desarrollado se basa en la adquisición de señales bioeléctricas mediante electrodos en contacto directo con el tejido vegetal. En este sentido, el electrodo constituye un elemento crítico del sistema, ya que define el acoplamiento entre la planta y la electrónica de adquisición, influyendo tanto en la calidad de la señal como en la presencia de artefactos.

Desde el punto de vista funcional, el electrodo puede dividirse en dos componentes principales: (i) el elemento conductor que se inserta en el tejido vegetal o permanece en contacto con este y permite la captación de la señal eléctrica, y (ii) el sistema de soporte mecánico, encargado de mantener una posición estable durante la medición y de reducir perturbaciones externas.

En este trabajo se desarrollaron dos alternativas de electrodo: un electrodo tipo aguja para mediciones sobre tallo y un electrodo de contacto para mediciones sobre hoja. El electrodo tipo aguja fue diseñado para su inserción en el tallo de la planta. Se emplearon electrodos metálicos de pequeño diámetro, fabricados en acero inoxidable y plata, materiales comúnmente utilizados en mediciones bioeléctricas debido a su buena conductividad y resistencia a la corrosión. En el caso de la aguja de plata, se recurrió a un joyero para fabricar una varilla fina de dicho material. Posteriormente, se realizó un posprocesado manual, que consistió en cortar la varilla a la longitud deseada y limar uno de sus extremos para obtener una punta tipo aguja. Finalmente, tanto la aguja de acero inoxidable como la de plata se unieron a los cables de los electrodos mediante soldadura con estaño. La comparación entre ambos materiales se analiza en el Capítulo 5.

Por otra parte, se diseño un electrodo específico para hoja, con el objetivo de explorar una alternativa de contacto menos invasiva que la inserción mediante aguja. Este diseño buscó mantener el elemento conductor en contacto con la superficie de la hoja mediante un soporte mecánico liviano, reduciendo el daño sobre el tejido y permitiendo evaluar la posibilidad de adquirir señales desde hojas. Los resultados obtenidos con esta variante se presentan en el Capítulo 5.

Los soportes mecánicos de los electrodos fueron diseñados específicamente para aplicaciones en campo, donde las condiciones ambientales pueden introducir perturbaciones significativas en la medición. Dicho soporte fue modelado mediante diseño asistido por computadora utilizando la herramienta OnShape [19] como se puede observar en la figura 3.15 y 3.16.

Posteriormente, para la fabricación de las piezas se utilizó el programa *Creality Print* como herramienta de laminado (*slicer*), configurando una altura de capa de 0,2 mm. La impresión se realizó mediante una impresora 3D Creality K1C [25], empleando filamento PETG, seleccionado por su mejor resistencia térmica y su mayor aptitud para uso en exteriores en comparación con otros materiales de impresión comunes.

El diseño de los soporte busca cumplir con los siguientes objetivos: (i) asegurar una fijación firme del electrodo al tejido vegetal, (ii) minimizar el movimiento

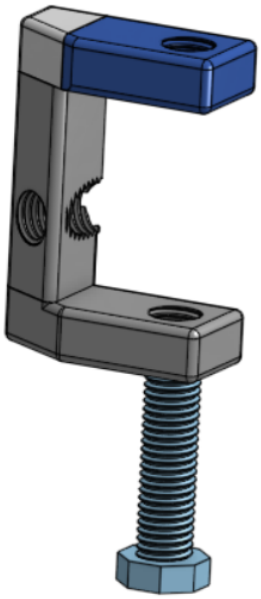


Figura 3.15: Modelo 3D del electrodo para el tallo en herramiento OnShape.

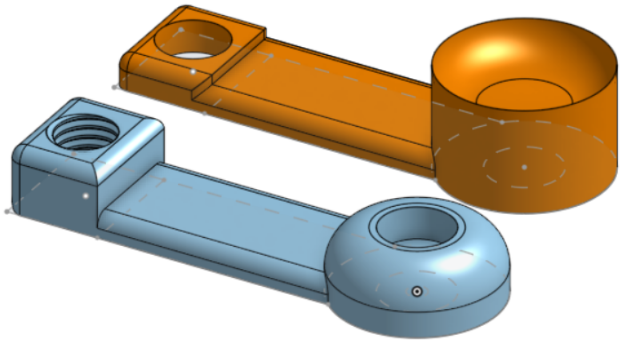


Figura 3.16: Modelo 3D del electrodo para la hoja en herramiento OnShape.

relativo entre el electrodo y la planta, y (iii) reducir la transmisión de vibraciones externas, tales como las generadas por viento o manipulación del cableado. Estos aspectos son fundamentales para disminuir la aparición de artefactos mecánicos en la señal adquirida.

En conjunto, la combinación de electrodos adecuados y soportes mecánicos específicos permite mejorar la estabilidad de la medición en condiciones de campo, contribuyendo a la obtención de registros más confiables.

En el "Manual de uso", en el anexo A, se encuentra la guía de uso y colocación de estos electrodos de forma correcta.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 4

Firmware y Software

4.1. Firmware

4.1.1. Introducción

Esta sección describe la arquitectura y el funcionamiento del firmware implementado en el sistema. Se presenta la organización general del software, los módulos que lo componen, la configuración de los periféricos utilizados y la lógica del bucle principal.

El firmware fue desarrollado sobre la plataforma Arduino para el microcontrolador ATSAMD21G18A (MKR ZERO) y se estructuró de forma modular. Su función es coordinar la adquisición periódica de la señal analógica, el almacenamiento de datos en tarjeta SD, la transmisión serie en vivo, la supervisión del estado de la batería y la gestión de modos de bajo consumo.

La descripción se organiza en tres partes. Primero se presenta la arquitectura general del firmware. Luego se describen los módulos principales del sistema. Finalmente, se detalla el funcionamiento del bucle principal y la interacción entre adquisición, almacenamiento, comunicación y bajo consumo.

4.1.2. Arquitectura general

La Figura 4.1 muestra la arquitectura general del firmware. El sistema se organiza alrededor de un bucle principal de tipo *round-robin* con interrupciones, que coordina la ejecución de módulos de software encargados de la adquisición, el almacenamiento, la comunicación y la supervisión del sistema.

La implementación se estructuró en capas, separando la lógica de aplicación, los módulos de software, los periféricos internos del microcontrolador y las interfaces externas del sistema. Esta organización permite desacoplar la lógica del programa principal de los detalles de implementación de cada módulo.

En este esquema, los módulos exponen funciones específicas y no dependen del modo de operación del sistema. La coordinación entre ellos se realiza desde `main.ino`, que actúa como punto de integración y define la secuencia de ejecución según el estado del sistema.

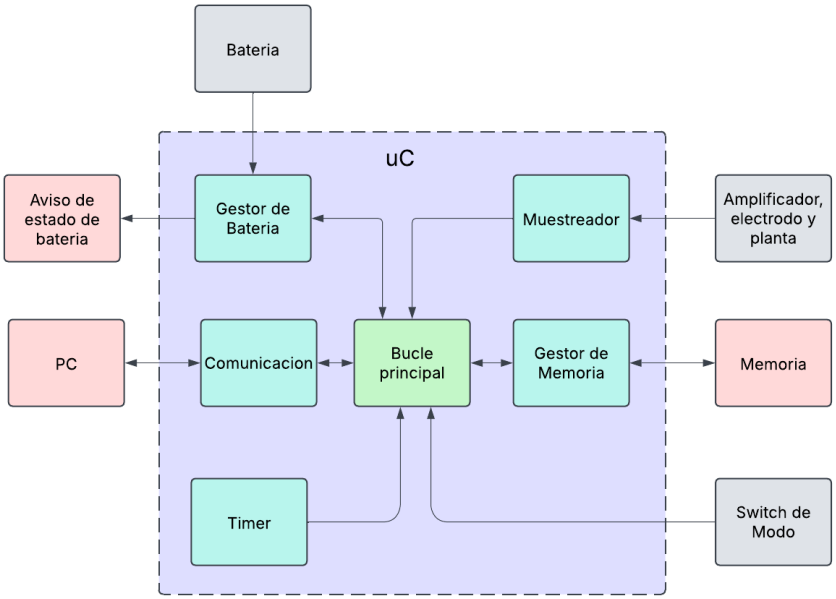


Figura 4.1: Diagrama alto nivel de la Arquitectura de Firmware

La Figura 4.2 presenta esta organización jerárquica. En el nivel superior se ubica `main.ino`, donde se implementan el bucle principal, la selección de modo, el control de estados y el flujo general de adquisición. En el nivel intermedio se encuentran los módulos de software descritos en la Sección 4.1.3, mientras que en los niveles inferiores se ubican los periféricos del microcontrolador y las interfaces externas utilizadas por el sistema.

4.1.3. Módulos

Timer

Para implementar la base de tiempo del sistema se utilizó el temporizador TC3 del microcontrolador SAMD21, alimentado por el oscilador interno de 32.768 Hz.

El módulo desarrollado contempla la configuración de los periféricos involucrados en la generación de la base de tiempo: `SYSCTRL`, encargado de habilitar y estabilizar la fuente de reloj; `GCLK`, utilizado para distribuir dicha señal hacia el temporizador; `TC3`, empleado como contador principal del sistema; y `NVIC`, responsable de habilitar la interrupción asociada al temporizador.

Además, se implementan funciones auxiliares para iniciar y detener el conteo, así como para obtener el tiempo total transcurrido desde la puesta en marcha del temporizador. A continuación, se describe la configuración realizada en cada uno de estos bloques más relevantes.

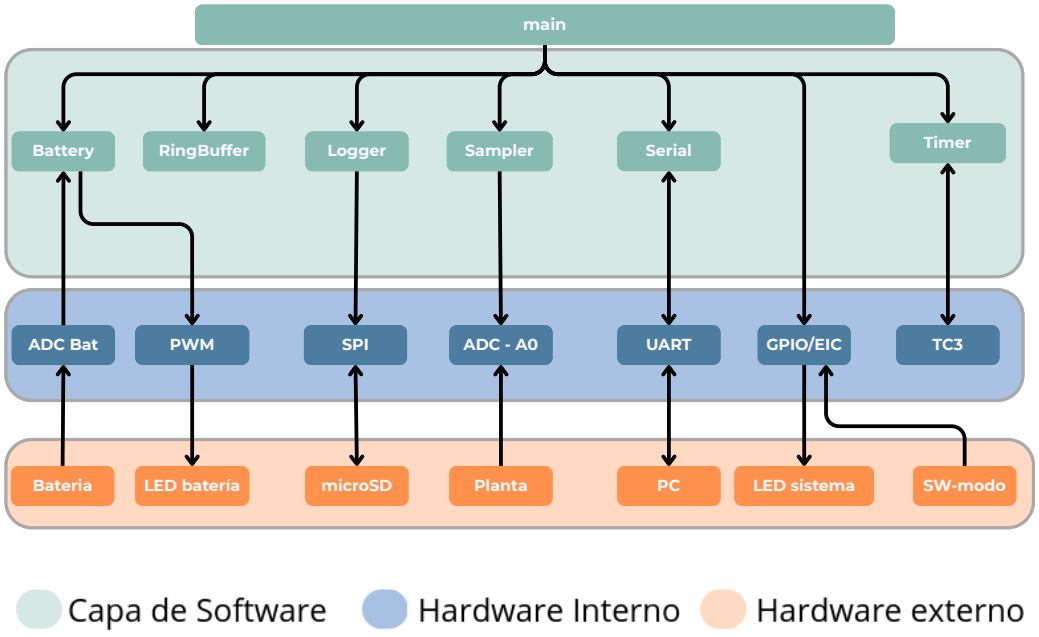


Figura 4.2: Diagrama jerarquico del Firmware

Registro	Campo	Valor	Efecto
OSC32K	STARTUP	0x6	Tiempo de arranque del oscilador interno 32 kHz
OSC32K	EN32K	1	Habilita salida de 32.768 Hz
OSC32K	ENABLE	1	Enciende el oscilador OSC32K
OSC32K	ONDEMAND	0	Oscilador siempre activo
PCLKSR	OSC32KRDY	Lectura	Espera a que el oscilador esté estable

Tabla 4.1: Configuración del periférico SYSCTRL

Configuración del oscilador La función `configure_osc32k()` configura el oscilador interno OSC32K del microcontrolador, el cual constituye la fuente de reloj de 32.768 Hz utilizada por el sistema. Configura el tiempo de arranque del oscilador, habilita su salida hacia el sistema de distribución de relojes y desactiva el modo on-demand para garantizar su funcionamiento continuo. Además se realiza la verificación del bit `OSC32KRDY`, esta evita que los periféricos sean alimentados con una señal de reloj aún no válida.

En la Tabla 4.1 se presentan los registros configurados, junto con los valores aplicados y el efecto esperado en cada caso.

La selección de `STARTUP = 0x6` define un tiempo de arranque mayor al mínimo. Este margen permite que la fuente de reloj alcance un régimen estable antes de ser empleada por los módulos periféricos que dependen de ella.

Registro	Campo	Valor	Efecto
GENDIV	ID	2	Selecciona generador GCLK2
GENDIV	DIV	0	Sin división
GENCTRL	ID	2	Configura generador GCLK2
GENCTRL	SRC	OSC32K	Fuente del generador: reloj interno 32 kHz
GENCTRL	GENEN	1	Habilita el generador
GENCTRL	RUNSTDBY	1	Generador activo en modo standby
CLKCTRL	ID	TCC2_TC3	Asigna clock al periférico TC3
CLKCTRL	GEN	GCLK2	Fuente del TC3 = GCLK2
CLKCTRL	CLKEN	1	Habilita clock hacia TC3

Tabla 4.2: Configuración del periférico GCLK

El bit `ENABLE` activa físicamente el bloque OSC32K, mientras que `EN32K` habilita la propagación de la señal de 32.768 Hz hacia el controlador de relojes genérico (GCLK).

La desactivación del modo `ONDEMAND` garantiza que el oscilador permanezca activo de forma continua, evitando apagados automáticos que podrían introducir discontinuidades temporales en la base de tiempo.

Enrutamiento del reloj La función `setupGCLK2()` configura el controlador (`GCLK`) para alimentar al temporizador `TC3` a partir del oscilador interno `OSC32K`. Para ello, se emplea el generador `GCLK2`, configurado sin división adicional. En esta configuración se activa el bit `RUNSTDBY` en el generador, ya que permite que la señal de reloj permanezca disponible aun cuando el microcontrolador ingresa en estados de bajo consumo. Finalmente, mediante el registro `CLKCTRL`, el generador `GCLK2` se asocia al canal de reloj correspondiente al bloque `TCC2_TC3`. Quedando así asignado el reloj al periférico TC3.

En la Tabla 4.2 se detallan los registros configurados para el enrutamiento del reloj hacia el temporizador.

Configuración del temporizador La función `timer_ini()` es la que configura todo el temporizador (la que debe usar el bucle principal), configura el oscilador y el enrutamiento del reloj, llamando a las funciones comentadas anteriormente.

A continuación, configura el TC3 en modo contador de 16 bits (`COUNT16`), con prescaler igual a 1, de modo que el contador incremente directamente a 32.768 Hz.

En la Tabla 4.3 se detallan los registros configurados para el temporizador.

Se seleccionó el modo Match Frequency (`MFRQ`), donde el período de interrupción queda determinado por el valor del registro `CC[0]`.

El período de muestreo del sistema está dado por:

$$T = \frac{\texttt{SAMP_TICKS} - 1}{32768}$$

Registro	Campo	Valor	Efecto
CTRLA	MODE	COUNT16	Modo contador de 16 bits
CTRLA	PRESCALER	DIV1	Sin prescaler (clock directo 32.768 Hz)
CTRLA	WAVEGEN	MFRQ	Modo Match Frequency
CTRLA	RUNSTDBY	1	Timer activo en standby
CC[0]	CC0	SAMP_TICKS-1	Define período de interrupción
COUNT	COUNT	0	Reinicia contador antes de iniciar
INTENSET	MC0	1	Habilita interrupción por compare match
INTFLAG	MC0	Escritura 1	Limpia bandera de interrupción

Tabla 4.3: Configuración del periférico TC3

El registro `COUNT` se deja en cero para no comenzar la cuenta del timer con un valor indefinido.

El registro `INTENSET.MC0` habilita la interrupción por coincidencia en el canal 0, mientras que la escritura de un 1 en `INTFLAG.MC0` permite limpiar explícitamente la bandera de interrupción dentro de la rutina de servicio.

Gestión de interrupciones Las funciones `timer_start()` y `timer_stop()` controlan la operación del temporizador `TC3` desde el bucle principal. La primera habilita la interrupción asociada al periférico en el `NVIC` y activa el contador, iniciando así la generación periódica de eventos de muestreo. La segunda detiene el contador y deshabilita la interrupción a nivel del `NVIC`, evitando que se ejecuten nuevas rutinas de atención.

La rutina `TC3_Handler()` corresponde a la función de servicio de interrupción del temporizador. Esta se ejecuta automáticamente cuando ocurre una coincidencia entre el contador interno y el valor configurado en `CC[0]`. Dentro de la rutina se verifica la bandera `MC0`, se limpia el evento de interrupción y se actualiza la base temporal interna del sistema. Además, se incrementa un contador de muestras pendientes, dejando al bucle principal la responsabilidad de realizar efectivamente la adquisición.

En la Tabla 4.4 se resumen las acciones realizadas sobre el Nested Vectored Interrupt Controller (NVIC).

Sampler

El módulo de muestreo es el encargado de adquirir la señal analógica proveniente de la etapa de acondicionamiento y convertirla en una magnitud digital

Elemento	Acción	Efecto
TC3_IRQn	Enable	Habilita la interrupción del TC3 en el NVIC
TC3_IRQn	Disable	Deshabilita la interrupción del TC3 en el NVIC

Tabla 4.4: Configuración del NVIC para TC3

expresada en milivoltios y centrada en cero.

Durante la inicialización del módulo se configura una resolución de 12 bits mediante la función `analogReadResolution(12)` de Arduino, de forma que cada conversión entregue un valor digital comprendido entre 0 y 4095.

Durante cada muestreo, la función `analogRead(A0)` de Arduino realiza la conversión de la señal presente en el canal analógico correspondiente. El valor crudo obtenido se convierte posteriormente a milivoltios utilizando como referencia una tensión de 3,3 V, según:

$$V_{ADC} = \frac{raw \cdot 3300}{4095}$$

donde raw es el valor digital entregado por el ADC.

Debido a que la etapa analógica de acondicionamiento polariza la señal alrededor de $V_{REF}/2$, como se explica en la Sección 3.2.7, el valor convertido se centra restando 1650 mV. De esta forma, la salida del módulo representa directamente la desviación de la señal respecto al punto medio:

$$V_{muestra} = V_{ADC} - \frac{V_{REF}}{2}$$

Esto permite obtener una señal bipolar en milivoltios, con rango aproximado entre $-1650\,mV$ y $+1650\,mV$, compatible con el formato utilizado por los módulos de almacenamiento y transmisión.

Con el objetivo de reducir el consumo energético del sistema, una vez finalizada cada conversión el ADC se deshabilita explícitamente mediante acceso directo al registro `CTRLA.ENABLE`.

Luego se espera a que finalice la sincronización interna del periférico verificando el bit `SYNCBUSY` del registro `STATUS`. Este mismo procedimiento se aplica también durante la inicialización, dejando el ADC apagado cuando no se encuentra realizando conversiones.

En la Tabla 4.5 se resumen los elementos de configuración utilizados por el módulo.

Ring Buffer

El firmware utiliza una estructura de datos tipo *ring buffer* (buffer circular) para almacenar temporalmente datos en memoria RAM de forma eficiente. En este caso, dicha estructura fue implementada dentro del entorno Arduino como una clase en C++.

Elemento	Tipo	Valor	Efecto
`analogReadResolution()`	Función Arduino	12	Configura resolución del ADC en 12 bits
`analogRead(A0)`	Función Arduino	A0	Ejecuta la conversión sobre el canal analógico de entrada
CTRLA	ENABLE	0	Deshabilita el periférico ADC al finalizar la conversión
STATUS	SYNCBUSY	Lectura	Espera a que la deshabilitación se sincronice

Tabla 4.5: Configuración del ADC

Un *ring buffer* es un arreglo de tamaño fijo que se reutiliza de forma circular, evitando la necesidad de desplazar datos en memoria cuando se agregan o eliminan elementos.

En esta implementación, el buffer almacena datos a nivel de byte (`uint8_t`) y mantiene tres variables principales para controlar su estado:

- **head**: índice donde se escribirá el próximo dato.
- **tail**: índice desde donde se leerá el próximo dato.
- **count**: cantidad de elementos actualmente almacenados en el buffer.

El almacenamiento real de los datos se realiza en un arreglo dinámico de bytes (`_buf`), cuyo tamaño es definido durante la construcción del objeto. La memoria se reserva dinámicamente utilizando `malloc()` y se libera en el destructor mediante `free()`.

Funcionamiento del buffer circular Cuando se inserta un nuevo dato en el buffer mediante la operación `push()`, el byte se escribe en la posición indicada por el índice `head`.

Luego, el índice se incrementa y se aplica una operación módulo con la capacidad del buffer para mantenerlo dentro del rango válido de posiciones. De esta manera, cuando el índice alcanza el final del arreglo vuelve automáticamente al inicio, reutilizando las posiciones previamente liberadas.

De forma similar, cuando se extrae un dato mediante la operación `pop()`, el valor se obtiene desde la posición indicada por el índice `tail`. Posteriormente, el índice `tail` se incrementa también de manera circular.

Este comportamiento se ilustra conceptualmente como un arreglo que se conecta consigo mismo en sus extremos, formando una estructura circular donde los índices de lectura y escritura avanzan independientemente.

El uso de un contador interno (`count`) permite distinguir entre las condiciones de buffer vacío y buffer lleno:

- El buffer está **vacío** cuando `count = 0`.
- El buffer está **lleno** cuando `count = capacity`.

Operaciones principales

La clase implementa las siguientes operaciones fundamentales:

- **push**: inserta un byte en el buffer si existe espacio disponible. Si el buffer está lleno, la operación falla.
- **pop**: extrae el byte más antiguo almacenado en el buffer.
- **peek**: permite observar el próximo byte a leer sin eliminarlo del buffer.
- **writeBlock**: inserta múltiples bytes consecutivos hasta que el buffer se llena o se escriben todos los datos solicitados.
- **readBlock**: extrae múltiples bytes consecutivos hasta que el buffer se vacía o se alcanza la cantidad solicitada.

Adicionalmente, la clase incluye funciones auxiliares para consultar el estado del buffer, tales como `isFull()`, `isEmpty()`, `size()`, `capacity()` y `freeSpace()`, así como la función `clear()` que reinicia completamente el buffer.

Logger

El módulo *Logger* es el encargado de gestionar el almacenamiento persistente de las muestras adquiridas por el sistema en una tarjeta SD. Su función principal es registrar las mediciones en archivos organizados de forma secuencial, permitiendo posteriormente su descarga completa hacia un dispositivo externo o la eliminación total del contenido de la tarjeta cuando sea necesario. Para comunicarse con la SD se hizo uso de la librería de Arduino "SD.h" [7].

Organización de archivos

Los datos se almacenan en archivos con formato CSV ubicados en la raíz de la tarjeta SD. Cada nuevo ciclo de registro genera automáticamente un archivo con un nombre secuencial siguiendo el esquema:

```
data1.csv, data2.csv, data3.csv, ...
```

De esta manera se evita sobrescribir información previamente registrada y se mantiene un orden creciente según el índice de archivo asignado durante la creación.

Para determinar el nombre del nuevo archivo, el sistema recorre el contenido de la tarjeta SD durante la inicialización y detecta el mayor índice existente, generando el siguiente número disponible.

Estructura del archivo de datos Cada archivo generado incluye inicialmente una línea de encabezado que define las columnas del archivo CSV:

```
tiempo,voltaje_mV
```

Posteriormente, cada línea del archivo corresponde a una muestra adquirida por el sistema, donde se almacena el instante de tiempo de la medición y el valor de voltaje medido en milivoltios.

Inicialización del sistema de registro Durante la inicialización del módulo se realizan varias etapas:

1. Inicialización del sistema de archivos de la tarjeta SD.
2. Exploración de los archivos existentes en la raíz de la tarjeta.
3. Determinación del índice más alto de archivo existente.
4. Generación automática del nombre del nuevo archivo de registro.
5. Creación y apertura del archivo correspondiente.
6. Escritura del encabezado CSV si el archivo es nuevo.

Si alguna de estas etapas falla, el sistema deshabilita el registro de datos para evitar errores de escritura o corrupción del sistema de archivos.

Registro de datos El módulo permite registrar datos de dos formas diferentes. Por un lado, es posible almacenar directamente líneas de texto ya formateadas en formato CSV, lo cual resulta útil cuando las muestras han sido previamente procesadas o formateadas. Por otro lado, el sistema permite escribir bloques de datos binarios directamente en el archivo, lo que puede resultar más eficiente en términos de velocidad o uso de memoria cuando se manejan estructuras de datos predefinidas.

En ambos casos, el objetivo es permitir una interfaz flexible entre el módulo de adquisición de datos y el sistema de almacenamiento.

Sincronización del almacenamiento El modulo permite forzar la sincronización del archivo con la tarjeta SD. Esta operación asegura que los datos que se encuentran temporalmente en memoria intermedia se escriban físicamente en la tarjeta, reduciendo el riesgo de pérdida de información en caso de reinicio inesperado del sistema o interrupción de energía.

Cierre del archivo Cuando el registro de datos finaliza o el sistema cambia de modo de operación, el archivo abierto se cierra correctamente. Este procedimiento asegura que el sistema de archivos permanezca consistente y que los datos almacenados puedan ser accedidos posteriormente sin riesgo de corrupción.

Descarga completa de datos

El modulo permite transmitir todos los archivos de datos almacenados en la tarjeta SD hacia un dispositivo externo mediante un flujo de comunicación (*stream*), típicamente a través de un puerto serie. En este esquema, el microcontrolador actúa como emisor de los datos, mientras que la computadora o dispositivo conectado al *stream* actúa como receptor.

Esta funcionalidad se implementa mediante la función `logger_dump_all`, la cual recibe como argumento un *stream* de salida y envía a través de él la totalidad de los archivos de datos almacenados en la tarjeta SD, siguiendo un protocolo de transmisión definido.

Para ello, el sistema identifica los archivos de datos presentes en la tarjeta y los transmite en orden secuencial. Cada archivo se envía completo de forma que el receptor pueda reconstruirlo correctamente.

Antes de transmitir el contenido de cada archivo, se envía una línea de encabezado que indica el nombre del archivo y su tamaño en bytes:

```
BEGIN_FILE:nombre_del_archivo,tamaño_en_bytes
```

A continuación, se transmite el contenido completo del archivo y, al finalizar, se envía un marcador de cierre:

```
END_FILE
```

De esta manera, el receptor puede identificar sin ambigüedad el inicio y el final de cada archivo, conocer su tamaño exacto y reconstruir los archivos originales a partir del flujo recibido.

Eliminación completa de datos

El módulo incluye también una funcionalidad que permite eliminar completamente el contenido de la tarjeta SD. El procedimiento consiste en recorrer todos los archivos y directorios presentes en la raíz de la tarjeta y eliminarlos de forma recursiva.

Esta operación permite reiniciar completamente el almacenamiento del sistema y preparar la tarjeta para una nueva campaña de adquisición de datos.

Serial

Para la implementación del Serial se utilizó la libreria de Arduino *Serial.h* [4]. En particular se hizo uso del objeto Serial1 en el bucle principal a posterior.

Battery

El sistema incorpora un módulo encargado de supervisar periódicamente el nivel de tensión de la batería del dispositivo. Este módulo permite detectar condiciones de batería baja y señalizar dicho estado mediante un indicador visual basado en un LED.

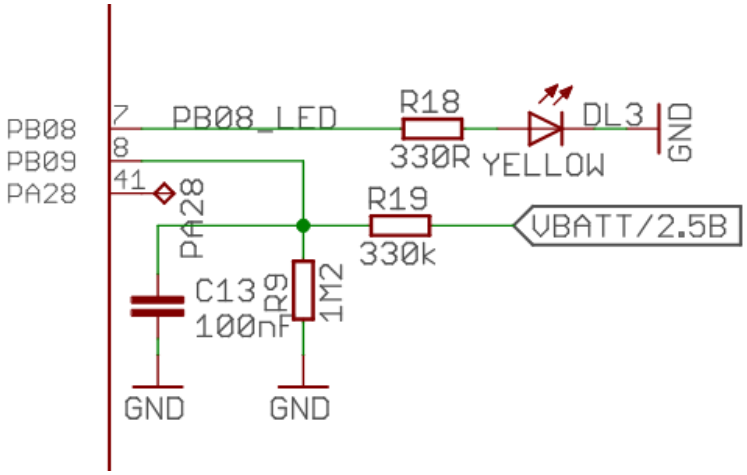


Figura 4.3: Divisor resistivo utilizado para la medición de la batería [6].

Medición de la tensión de batería La medición de la tensión se realiza utilizando el convertidor analógico–digital (ADC) del microcontrolador. Debido a que la tensión de la batería puede superar el rango de entrada permitido por el ADC, se utiliza previamente un divisor resistivo que viene incluido en el MKRZERO que reduce la tensión antes de ser medida.

El divisor está compuesto por resistencias de 1,2 MΩ y 330 kΩ como se muestra en la Figura 4.3, que está conectado pin *PB09*, la etiqueta *VBATT/2.5B* es el voltaje directo de la batería.

Esto genera un factor de división aproximado de:

$$k = \frac{V_{ADC}}{V_{BAT}} \approx 0{,}784$$

Por lo tanto, la tensión real de la batería puede estimarse a partir del valor digital obtenido por el ADC mediante:

$$V_{ADC} = \frac{raw \cdot V_{ref}}{4095}$$

$$V_{BAT} = \frac{V_{ADC}}{k}$$

donde:

- raw es el valor digital entregado por el ADC (0–4095).
- V_{ref} es la tensión de referencia del convertidor ADC ($3{,}3\,V$).
- k es la relación del divisor resistivo.

En este sistema se utiliza una resolución de 12 bits, característica del microcontrolador SAMD21.

Verificación de la batería El modulo no realiza la medición de la batería en cada iteración del programa principal. En su lugar, se implementa un mecanismo de verificación periódica basado en tiempo.

El módulo compara el tiempo actual del sistema con el instante de la última verificación realizada. Si el intervalo transcurrido es menor que un tiempo mínimo predefinido, la función finaliza sin realizar una nueva medición.

Este mecanismo reduce el consumo de energía y evita realizar conversiones ADC innecesarias.

Inicialización del sistema de monitoreo Durante la inicialización del módulo se configuran los parámetros necesarios para la medición y señalización del estado de la batería. En particular se realiza:

- Configuración de la resolución del ADC en 12 bits.
- Inicialización de la variable de referencia temporal utilizada para el control de los intervalos de verificación.
- Configuración del pin de salida que controla el LED indicador de batería baja.

La lectura de la tensión de batería se realiza sobre un canal interno del ADC, por lo que no es necesario configurar explícitamente el modo del pin correspondiente.

Detección de batería baja Una vez realizada la conversión ADC, la tensión estimada de la batería se compara con un umbral predefinido. Este umbral representa la tensión mínima aceptable de operación del sistema.

Si la tensión estimada cae por debajo de este valor, el sistema considera que la batería se encuentra en estado de baja carga.

Señalización mediante LED El estado de la batería se comunica al usuario mediante un LED indicador. El comportamiento del LED depende del nivel de carga detectado.

- **Batería en estado normal:** el LED realiza un breve encendido periódico cada varios segundos. Este patrón funciona como señal de funcionamiento normal del sistema.
- **Batería baja:** el LED realiza una secuencia de parpadeos más frecuente durante el ciclo de señalización, lo que permite diferenciar visualmente el estado de batería baja.

El control del LED se realiza mediante modulación de ancho de pulso (PWM), permitiendo ajustar la intensidad luminosa del indicador.

Ciclo temporal de señalización El comportamiento del LED se organiza en ciclos temporales. Dentro de cada ciclo se definen diferentes intervalos de tiempo en los que el LED se enciende o apaga según el estado de la batería.

Al finalizar cada ciclo completo de señalización, el temporizador interno se reinicia y comienza nuevamente el patrón correspondiente.

Este enfoque permite implementar patrones visuales de señalización sin necesidad de utilizar temporizadores adicionales o interrupciones dedicadas.

4.1.4. Bucle Principal

El firmware del sistema se organiza mediante un bucle principal (*main loop*) implementado bajo una arquitectura *Round Robin* asistida por interrupciones. El diagrama general se muestra en la Figura 4.4.

En este esquema las tareas críticas en tiempo real son realizadas mediante interrupciones, mientras que el procesamiento principal se realiza en el contexto normal del programa dentro del bucle principal. Esto permite mantener un comportamiento determinista sin necesidad de un sistema operativo en tiempo real.

El funcionamiento general del bucle principal puede resumirse en las siguientes etapas:

1. Lectura del switch de modo y determinación del modo de operación.
2. Detección de cambios de modo y reconfiguración del sistema.
3. Ejecución de los muestreos pendientes indicados por el temporizador.
4. Procesamiento del flujo de datos según el modo de operación (*Logger* o *Live*).
5. Supervisión del estado de la batería.
6. Entrada en modo de bajo consumo hasta la siguiente interrupción.

A continuación se describe cada una de estas etapas.

Inicialización del Sistema Durante la ejecución de la función `setup()` de Arduino se realiza la inicialización de todos los módulos necesarios para el funcionamiento del sistema. En esta etapa se configuran los periféricos del microcontrolador, se inicializan los módulos de adquisición y almacenamiento, y se establecen los estados iniciales de las variables de control.

El proceso de inicialización se compone de las siguientes etapas:

1. **Configuración del pin de selección de modo**

   El pin `MODE_PIN` se configura como entrada con resistencia *pull-up* interna mediante la función `pinMode(MODE_PIN, INPUT_PULLUP)`. Este pin determina el modo de operación del sistema:

   - **Modo Live**: transmisión de datos en vivo a través de la interfaz UART.

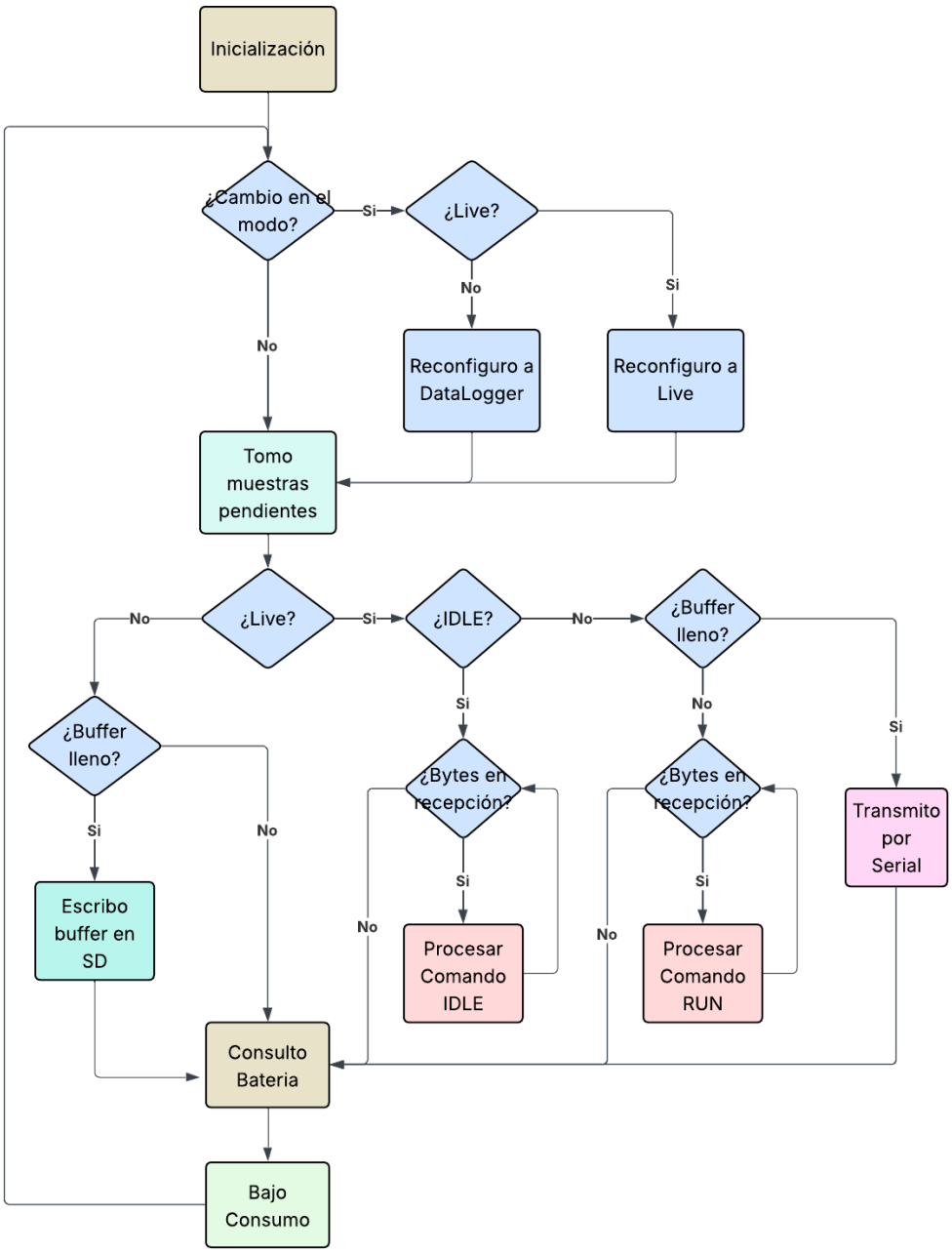


Figura 4.4: Diagrama de flujo del bucle principal

- **Modo Logger**: registro de datos en la tarjeta SD.

Además, se configura una interrupción asociada a este pin mediante `attachInterrupt`. Esta interrupción no ejecuta ninguna lógica específica, sino que se utiliza para despertar al microcontrolador desde los modos de bajo consumo cuando se produce un cambio en el estado del pin.

2. **Inicialización del sistema de temporización**

   Se inicializa el módulo `timer` mediante la función `timer_ini()`. Este módulo es responsable de generar la base temporal del sistema y definir el período de muestreo. El temporizador permite además determinar cuántas muestras deben tomarse cuando el sistema sale de un estado de bajo consumo.

3. **Inicialización del módulo de adquisición**

   El módulo de adquisición se inicializa mediante la función `sampler_ini()`. Este módulo se encarga de realizar las conversiones analógicas utilizando el ADC del microcontrolador y devolver el valor medido de la señal en milivoltios.

4. **Configuración del indicador visual**

   El LED integrado del microcontrolador (`LED_BUILTIN`) se configura como salida digital. Este LED se utiliza como indicador visual del modo de operación activo del sistema.

5. **Inicialización del sistema de monitoreo de batería**

   El módulo de monitoreo de batería se inicializa mediante la función `battery_init()`. Este módulo permite supervisar el estado de la batería durante la operación del sistema y detectar condiciones de bajo voltaje.

6. **Inicialización del estado del sistema**

   Finalmente se inicializan las variables de estado globales utilizadas por el sistema. En particular, el sistema comienza en estado *idle*, lo que significa que la adquisición se encuentra detenida hasta que se reciba un comando de inicio en modo *Live* o se active el modo *Logger*.

Selección de modo

El modo de operación del sistema se determina mediante un switch físico conectado al pin `MODE_PIN`. Este pin se configura con resistencia interna y se lee en cada iteración del bucle principal.

El firmware implementa dos modos de operación:

- **MODE_LOGGER**: el sistema funciona como registrador autónomo de datos, almacenando las muestras adquiridas en la tarjeta microSD.
- **MODE_LIVE**: el sistema transmite las muestras en vivo hacia una computadora mediante comunicación serie.

Además, el pin del switch posee una interrupción asociada configurada en modo `CHANGE`. Esta interrupción no ejecuta lógica de aplicación, sino que se utiliza únicamente como mecanismo de *wake-up* para despertar al microcontrolador cuando el usuario cambia el modo mientras el sistema se encuentra en bajo consumo.

Detección de cambio de modo

El firmware mantiene un registro del modo anterior de operación. Si se detecta que el modo actual es diferente al modo anterior, se realiza una reconfiguración del sistema.

Durante esta transición se realizan las siguientes acciones:

- Se limpia el buffer de adquisición para evitar mezclar datos entre modos.
- Se detiene el temporizador de adquisición.
- Se inicializan o deshabilitan los periféricos necesarios según el modo.

En particular:

- Al entrar en modo *Live* se inicializa la comunicación serie y se deshabilita el sistema de registro en SD.
- Al entrar en modo *Logger* se inicializa el módulo de almacenamiento en la tarjeta SD y se habilita el temporizador de adquisición.

Este mecanismo evita conservar datos pendientes, configuraciones activas o periféricos habilitados del modo anterior.

Gestión del muestreo

La adquisición de datos es controlada por un temporizador periódico que genera interrupciones cada $20\,ms$. Sin embargo, la interrupción no ejecuta directamente la adquisición de datos, sino que únicamente incrementa un contador de muestreos pendientes.

Durante cada iteración del bucle principal, el sistema consulta cuántas muestras pendientes existen. Por cada muestra pendiente se ejecuta la adquisición correspondiente.

Cada muestra adquirida contiene:

- El tiempo de adquisición en milisegundos.
- El valor de voltaje medido en milivoltios.

Estas muestras se almacenan temporalmente en un buffer circular de memoria. Este buffer desacopla el proceso de adquisición del proceso de transmisión o almacenamiento, evitando la pérdida de datos cuando alguna de estas operaciones requiere más tiempo.

Cuando el buffer se llena, se activa una bandera que indica al bucle principal que debe vaciar el contenido del buffer. Sin embargo, debido a la demora que puede ocurrir al escribir en la SD, puede ocurrir que se acumulen varias interrupciones del temporizador antes de que el sistema procese las muestras pendientes. En estas situaciones, si se acumulan interrupciones antes de que el bucle principal procese la adquisición, el sistema ejecuta varias conversiones de forma consecutiva y reconstruye sus marcas temporales a partir del período configurado.

Para evitar que todas estas muestras queden registradas con la misma marca de tiempo, el sistema reconstruye marcas de tiempo equiespaciadas utilizando el período del temporizador. De esta forma, si existen n muestras pendientes, las primeras muestras del lote reciben marcas de tiempo anteriores separadas por el período del temporizador, mientras que la última muestra del lote conserva la marca de tiempo actual entregada por el temporizador. Este procedimiento permite mantener una base temporal consistente aun cuando varias muestras se procesen de manera consecutiva.

Cada muestra adquirida se almacena en un registro binario compacto de 6 bytes con el siguiente formato:

- 4 bytes: tiempo de adquisición en milisegundos (`uint32_t`)
- 2 bytes: voltaje medido en milivoltios (`int16_t`)

Estos registros se almacenan temporalmente en un *buffer circular* implementado en memoria RAM.

El buffer circular tiene una capacidad de 8190 bytes, lo que corresponde a 1365 registros completos de 6 bytes cada uno. Este tamaño fue seleccionado para permitir almacenar aproximadamente 27 segundos de adquisición continua a una frecuencia de muestreo de $50\,Hz$ antes de requerir el vaciado del buffer.

Cuando el espacio libre del buffer es insuficiente para almacenar un nuevo registro completo, el sistema activa una bandera de drenado. Esta bandera indica al bucle principal que debe vaciar el contenido del buffer, ya sea escribiendo los datos en la tarjeta SD (modo *Logger*) o transmitiéndolos por UART (modo *Live*). Una vez que el buffer es completamente procesado, la bandera se limpia y el sistema continúa con la adquisición normal de datos.

Modo Logger

En modo *Logger* el sistema funciona como registrador autónomo de datos. En este modo el bucle principal verifica periódicamente si el buffer de adquisición ha alcanzado su capacidad máxima.

Cuando el espacio libre ya no permite almacenar un registro completo de 6 bytes, se activa una bandera de drenado, el contenido se procesa en bloques y se convierte a formato CSV. Cada muestra se transforma en una línea con el formato:

```
tiempo_ms,voltaje_mV
```

Las líneas generadas se almacenan en archivos dentro de la tarjeta microSD utilizando el módulo de registro descrito en la Sección 4.1.3.

Este mecanismo permite que el proceso de adquisición continúe mientras se escriben los datos en memoria externa, evitando pérdidas de muestras.

Modo Live

En modo *Live* el sistema se comunica con una computadora mediante el puerto serie `Serial1` a una velocidad de 921600 bps.

Este modo posee dos subestados de operación:

- **Idle**: el sistema espera comandos del usuario sin realizar transmisión de datos.
- **Run**: el sistema adquiere datos continuamente y transmite las muestras en vivo.

Los comandos se reciben como mensajes ASCII terminados en `\n`. Entre los comandos implementados se encuentran:

- `START`: inicia la adquisición y transmisión continua.
- `STOP`: detiene la adquisición.
- `STATUS`: informa el estado actual del sistema.
- `DOWN`: descarga el contenido de la tarjeta SD.
- `DELETE`: elimina todos los archivos de la tarjeta SD.

Durante el estado *Run*, cuando el buffer de adquisición se llena, las muestras almacenadas se transmiten por el puerto serie utilizando el mismo formato CSV utilizado en el modo *Logger*.

En la Figura 4.5 se visualiza como se dan los cambios de estado.

Supervisión de batería

En cada iteración del bucle principal el sistema consulta el estado de la batería mediante el módulo de monitoreo descrito en la Sección 4.1.3. Este módulo se encarga de medir la tensión de la batería y controlar el LED indicador de estado.

Bajo consumo

Finalmente, el sistema entra en un modo de bajo consumo hasta que ocurre una nueva interrupción. Según el modo en el que nos encontremos se utilizan distintas tecnicas de bajo consumo. Esto se debe a que para enviar por serial necesitamos tener abierto el puerto durante toda la sesión de envio para que no se cierre el puerto del lado de la PC en la interfaz.

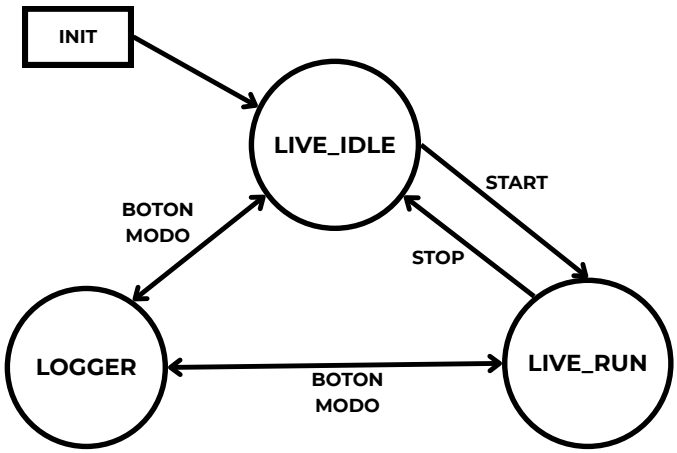


Figura 4.5: Estados del sistema

El microcontrolador utilizado en el sistema es el *ATSAMD21G18A*, basado en un núcleo *ARM Cortex-M0+* [8]. Este núcleo permite reducir el consumo energético mediante instrucciones de suspensión del procesador, principalmente la instrucción `WFI` (*Wait For Interrupt*). Cuando esta instrucción es ejecutada, el procesador detiene la ejecución del código hasta que ocurre una interrupción habilitada.

El sistema implementa dos modos de bajo consumo diferenciados, utilizados según el modo de operación del dispositivo:

- **Modo Live**: utiliza el modo *IDLE*, que detiene únicamente el núcleo del procesador pero mantiene activos la mayoría de los periféricos.
- **Modo Logger**: utiliza el modo *STANDBY*, que corresponde a un modo de suspensión profunda donde se apagan más dominios de reloj y se reduce significativamente el consumo energético.

El modo IDLE se utiliza durante la operación *Live* del sistema, donde se requiere mantener activos los periféricos del microcontrolador (como interfaces de comunicación), pero se busca reducir el consumo cuando el procesador no está ejecutando tareas.

La función esta presente en la librería *LowPower.h* de Arduino [5]:

```
void ArduinoLowPowerClass::idle() {
 SCB->SCR &= ~SCB_SCR_SLEEPDEEP_Msk;
 PM->SLEEP.reg = 2;
 __DSB();
 __WFI();
}
```

El funcionamiento de esta función es el siguiente:

- Se limpia el bit `SLEEPDEEP` del registro `SCR` (*System Control Register*) del núcleo ARM. Esto indica que el procesador debe entrar en un modo de suspensión normal y no en un modo profundo.
- Se configura el registro `PM->SLEEP` del *Power Manager* del SAMD21 para seleccionar el modo *IDLE2*. Este modo detiene el núcleo del CPU pero mantiene operativos varios periféricos del sistema.
- La instrucción `__DSB()` (*Data Synchronization Barrier*) asegura que todas las operaciones de escritura a registros se completen antes de que el procesador entre en suspensión.
- Finalmente, la instrucción `__WFI()` detiene la ejecución del procesador hasta que ocurre una interrupción.

Este modo es adecuado para el funcionamiento en vivo, ya que permite que los periféricos continúen operando mientras el núcleo permanece inactivo cuando no hay tareas pendientes.

El modo STANDBY se utiliza durante el funcionamiento autónomo del sistema en modo *Logger*. En este caso se prioriza minimizar el consumo energético entre mediciones.

La función implementada es la siguiente:

```
void sleep_standby(void) {
    SysTick->CTRL &= ~(SysTick_CTRL_ENABLE_Msk |
                       SysTick_CTRL_TICKINT_Msk);

    SCB->SCR |= SCB_SCR_SLEEPDEEP_Msk;
    __DSB();
    __WFI();
}
```

- Se deshabilita el temporizador `SysTick`, el cual es un temporizador interno del núcleo ARM utilizado frecuentemente como base de tiempo del sistema. Esto evita que interrupciones periódicas despierten innecesariamente al microcontrolador.
- Se establece el bit `SLEEPDEEP` en el registro `SCR` del núcleo ARM. Esto indica que al ejecutar la instrucción de suspensión el procesador debe entrar en un modo de suspensión profunda.
- La instrucción `__DSB()` asegura la correcta sincronización de los registros antes de entrar en suspensión.
- Finalmente, la instrucción `__WFI()` detiene el procesador y lo coloca en modo *STANDBY*. En este estado se detienen múltiples dominios de reloj y el consumo energético se reduce significativamente.

El microcontrolador permanecerá en este estado hasta que ocurra una interrupción válida proveniente de un periférico configurado como fuente de *wake-up*, en nuestro caso será el Timer o el Botón de modo.

Este modo permite extender considerablemente la autonomía del sistema durante períodos prolongados de adquisición de datos en campo.

4.2. Software

4.2.1. Introducción

Para la operación del sistema se desarrolló una aplicación de escritorio en *Python*. Esta herramienta permite establecer la comunicación con el dispositivo, supervisar la adquisición en vivo, descargar archivos generados en modo *Logger* y analizar registros almacenados en formato CSV. El objetivo fue concentrar en una única aplicación las tareas de control, visualización y procesamiento inicial de los datos, de forma que el uso del sistema no dependa de herramientas externas adicionales.

La implementación se realizó con una estructura modular, separando las funciones de comunicación, procesamiento, visualización y almacenamiento local. Esta organización facilita el mantenimiento del código y, además, permite extender la interfaz sin modificar la lógica básica de adquisición. En particular, se buscó evitar que la interfaz gráfica quedara bloqueada durante tareas prolongadas, como la lectura continua del puerto serie o la transferencia de archivos desde el dispositivo. Para ello se recurrió a hilos de trabajo y a colas de mensajes, de modo que las operaciones de entrada y salida se ejecuten de forma desacoplada respecto a la actualización de la ventana principal.

Desde el punto de vista del usuario, la aplicación reúne en una misma interfaz la conexión al equipo, el envío de comandos, la inspección de la señal, el registro local de muestras, la descarga de archivos y el análisis posterior de registros ya existentes. Esto permitió utilizar la misma herramienta tanto durante el desarrollo del prototipo como en las pruebas funcionales y en el procesamiento posterior de las capturas obtenidas.

4.2.2. Arquitectura general

La arquitectura del programa se organiza en tres bloques principales: rutinas auxiliares, hilos de trabajo e interfaz principal. Las rutinas auxiliares incluyen funciones de detección de puertos serie, conversión de tiempo, formateo de ejes y procesamiento de líneas recibidas. Los hilos de trabajo se emplean en las tareas que no deben ejecutarse sobre el hilo principal de la interfaz, como la lectura continua del puerto serie y la descarga de archivos. Finalmente, la ventana principal concentra la construcción de la GUI, el estado general de la aplicación, los buffers de datos y la actualización de gráficos y controles.

La lectura en vivo se implementa mediante un hilo específico, denominado `SerialWorker`. Este hilo abre el puerto serie seleccionado, recibe datos de forma

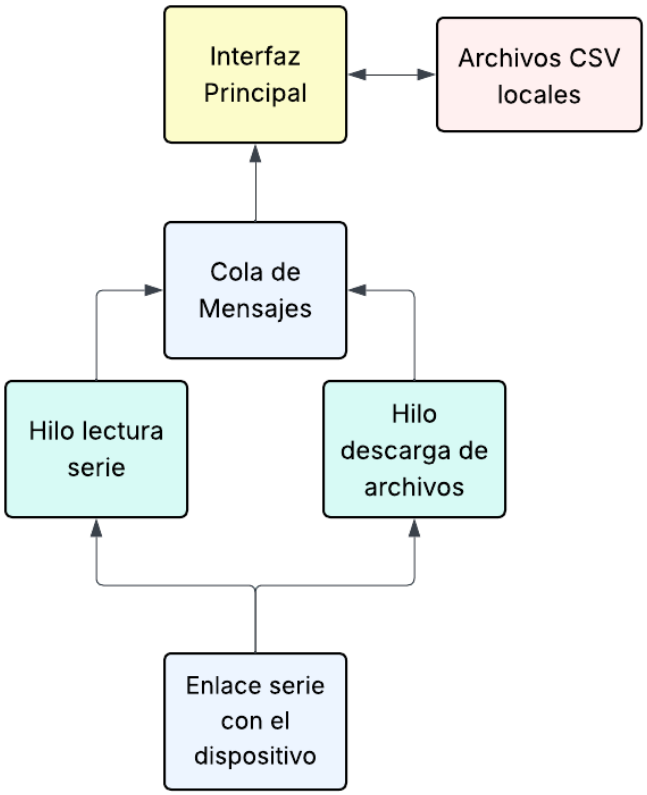


Figura 4.6: Arquitectura general de la aplicación desarrollada.

continua y reconstruye líneas completas a partir de un delimitador de fin de línea. En la implementación actual se adopta una transmisión con `EOL_TX = "\r\n"` y recepción con `EOL_RX = "\n"`, lo que permite mantener compatibilidad con distintas variantes del firmware. La velocidad de operación definida en la aplicación es de 921600 bps, valor elegido para acompañar el caudal de muestras del modo *Live* y reducir el riesgo de acumulación de datos en la interfaz.

La descarga de archivos se resuelve mediante un segundo hilo, `DownloadWorker`. Su función es encargarse del protocolo de transferencia de archivos utilizado por el firmware, para descargar los archivos generados en el modo *Logger*, evitando que el resto de la aplicación quede detenido durante esa operación. Ambos hilos intercambian información con la ventana principal a través de colas de mensajes. Este esquema evita el acceso directo de los hilos secundarios a los widgets de la interfaz, reduciendo problemas de sincronización y manteniendo una actualización ordenada del estado interno del programa.

En la ventana principal, los datos del modo *Live* se almacenan temporalmente en buffers circulares basados en `deque` (double-ended queue). En la implementación actual, estos buffers poseen un tamaño máximo de 10000 puntos, lo que permite limitar el consumo de memoria durante adquisiciones prolongadas. A partir de

estos buffers se generan las vistas, las exportaciones parciales y, si corresponde, el almacenamiento local en CSV.

Desde el punto de vista funcional, la arquitectura puede resumirse en los siguientes elementos, también se presenta en forma de diagrama en la Figura 4.6:

- detección y selección del puerto serie;
- lectura continua de datos y envío de comandos;
- descarga de archivos del modo *Logger*;
- cola de mensajes para comunicación interna;
- buffers de adquisición en memoria;
- herramientas de visualización, exportación y análisis.

Esta separación resulta útil no solo para el funcionamiento actual del sistema, sino también para su eventual réplica o ampliación. Por ejemplo, la lógica de descarga puede reutilizarse sin cambios si se mantiene el mismo protocolo de transferencia, mientras que las herramientas de visualización pueden adaptarse a nuevos formatos de señal sin alterar el esquema general de comunicación.

4.2.3. Interfaz y comunicación con el dispositivo

La interfaz principal se organiza en una barra superior de conexión, una barra de comandos, una consola de eventos y dos vistas principales de trabajo: adquisición en vivo y análisis fuera de línea. Esta organización permite acceder de forma directa a las acciones habituales durante el uso del sistema, evitando menús o ventanas auxiliares innecesarias.

La barra de conexión permite seleccionar el puerto serie, refrescar la lista de puertos detectados y establecer o cerrar la conexión con el dispositivo. La detección de puertos se realiza de forma distinta según el sistema operativo. En Linux se priorizan dispositivos del tipo `/dev/ttyACM*` y `/dev/ttyUSB*`; en macOS se filtran `/dev/tty.usb*` y `/dev/cu.usb*`; en Windows se listan directamente los puertos `COM`. Además, la aplicación construye una etiqueta descriptiva para cada puerto, agregando la descripción del dispositivo y, si está disponible, el identificador `VID:PID`. Posteriormente, las entradas se ordenan dando prioridad a dispositivos cuya descripción contenga cadenas como `arduino`, `mkr`, `samd`, `cdc` o `zero`. Este criterio reduce la probabilidad de seleccionar un puerto incorrecto cuando existen varios dispositivos conectados.

Una vez establecida la conexión, la aplicación puede enviar comandos de control al firmware. En la interfaz actual se incluyen, entre otros, los comandos `START`, `STOP`, `DOWN` y `DELETE`. Los dos primeros se utilizan para iniciar y detener la adquisición en vivo. El comando `DOWN` se emplea para solicitar la transferencia de archivos almacenados en la tarjeta SD, mientras que `DELETE` se reserva para el borrado de esos archivos, previa confirmación del usuario.

La lectura del puerto serie se realiza mediante el hilo `SerialWorker`. Este hilo abre el puerto con un tiempo de espera corto y lee los datos en bloques, acumulándolos en un buffer hasta detectar el delimitador de fin de línea. Una vez reconstruida una línea completa, la misma se decodifica y se reenvía al hilo principal. Este procedimiento evita depender de una lectura carácter por carácter y mejora la robustez frente a fragmentación de mensajes.

Las líneas recibidas pueden corresponder a datos válidos o a mensajes de estado. Para el procesamiento de muestras, la aplicación admite dos formatos principales. El primero corresponde a la forma

```
12345,2010
```

donde el primer campo representa el tiempo en milisegundos y el segundo el voltaje en milivoltios. El segundo formato corresponde a

```
0:00:12.345,2010
```

donde el tiempo se expresa como horas, minutos y segundos con resolución subsegundo. Si la línea coincide con alguno de estos formatos, se convierte internamente a una tupla de tiempo en segundos y voltaje en milivoltios. Si no coincide, la línea se conserva únicamente en la consola de eventos.

La consola constituye una parte importante de la interfaz, se presenta en la Figura 4.7. Cada mensaje se muestra junto con una marca temporal generada por la propia aplicación, lo que permite seguir en forma cronológica la secuencia de conexión, envío de comandos, respuestas del firmware, advertencias y errores. Esto simplifica el uso del sistema durante la operación normal y, además, aporta una herramienta de apoyo durante la depuración.

En el caso de la descarga de archivos mediante el comando *DOWN*, la transferencia de archivos se apoya en un protocolo basado en marcas de inicio y fin. El encabezado de un archivo adopta la forma:

```
BEGIN_FILE:nombre,size
```

seguido por el contenido del archivo y una marca final `END_FILE`. Si el tamaño del archivo se recibe correctamente, la descarga se realiza por conteo de bytes. Si no se dispone de un tamaño válido, la aplicación pasa a un modo alternativo en el que consume líneas hasta detectar la marca de finalización. Para hacer esta operación más robusta, el hilo de descarga incorpora distintos tiempos máximos de espera: uno para la llegada del primer archivo, otro para la inactividad entre bloques y otro para la espera del marcador `END_FILE`. De esta forma se evita que la interfaz quede indefinidamente esperando una transferencia incompleta.

Durante la descarga, la ventana principal muestra el nombre del archivo en curso, la cantidad de bytes transferidos y el progreso de la operación, como se presenta en la Figura 4.8. Antes de iniciar este proceso, la aplicación desconecta temporalmente la lectura en vivo para liberar el puerto serie. Al finalizar, el puerto puede volver a utilizarse para tareas de supervisión o adquisición. Esta decisión evita conflictos entre dos operaciones concurrentes sobre el mismo enlace físico.

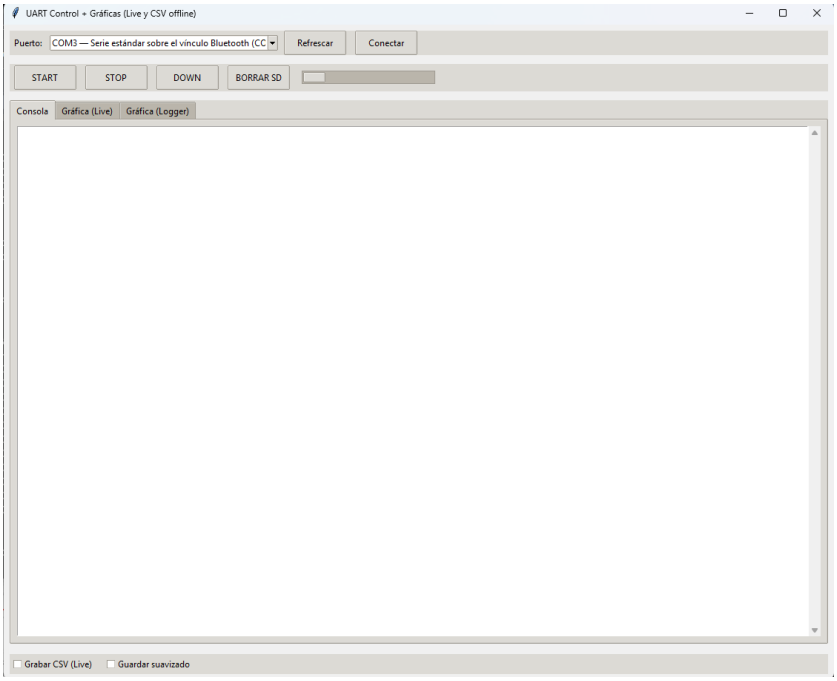


Figura 4.7: Ventana principal de la interfaz gráfica.

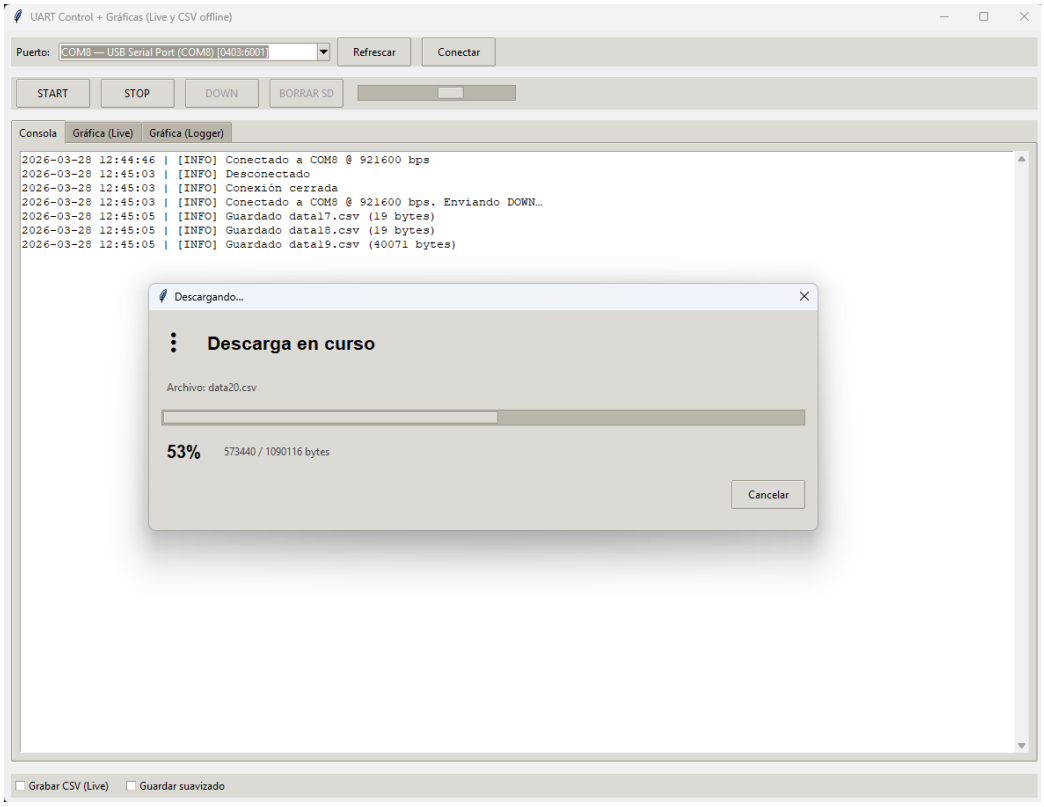


Figura 4.8: Interfaz utilizada durante la descarga de archivos del modo *Logger*.

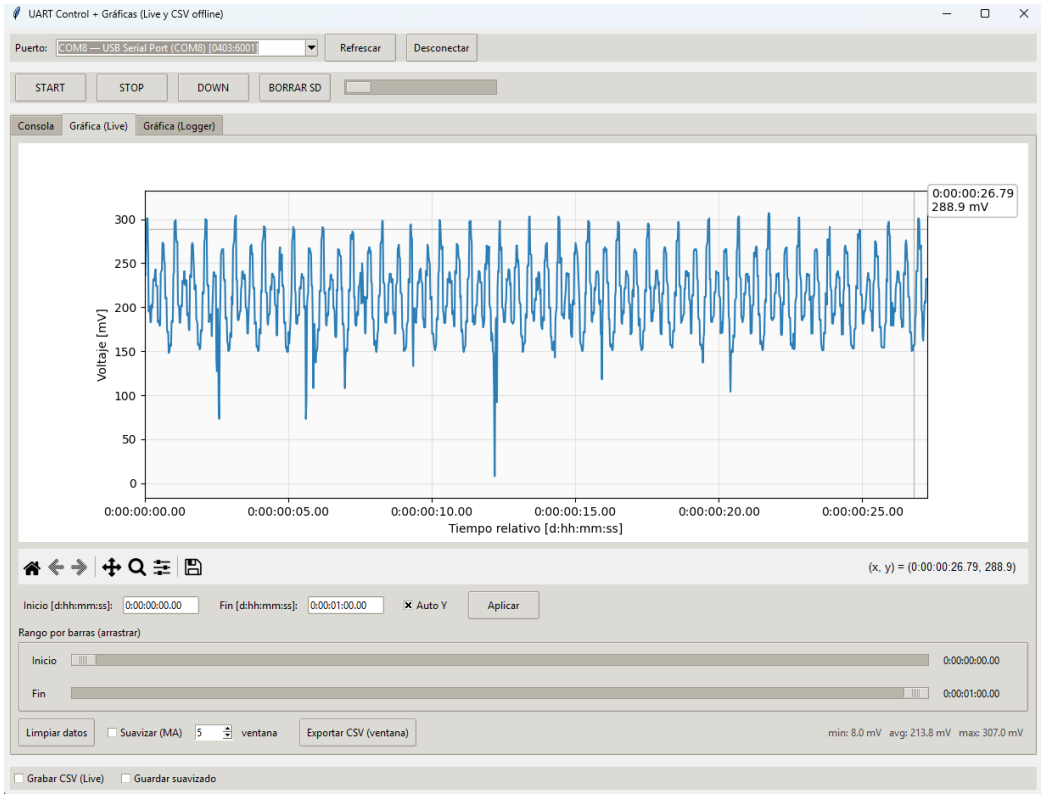


Figura 4.9: Vista de adquisición en *Live*.

4.2.4. Visualización, exportación y análisis de datos

La vista de adquisición en vivo representa la señal como voltaje en función del tiempo relativo transcurrido desde la primera muestra válida, la vista se muestra en la Figura 4.9. Para esto se utiliza `Matplotlib` embebido en la interfaz `Tkinter`. El eje horizontal se expresa mediante una representación temporal del tipo `d:hh:mm:ss`, de manera que el usuario pueda trabajar con adquisiciones largas sin necesidad de convertir manualmente los tiempos.

La aplicación incluye dos mecanismos complementarios para seleccionar la ventana temporal visible. El primero consiste en campos de entrada donde el usuario puede escribir el tiempo inicial y final en formato `d:hh:mm:ss`. El segundo consiste en barras deslizantes que modifican esos mismos límites de manera interactiva. Ambas herramientas se mantienen sincronizadas: si el usuario escribe un valor, los deslizadores se actualizan; si mueve un deslizador, también se actualizan los campos de texto. Esto mejora la navegación sobre registros largos y permite un uso más preciso cuando se desea inspeccionar un intervalo corto.

Además de la selección de ventana, la vista *Live* incorpora autoajuste del eje vertical, suavizado opcional por media móvil, cálculo de estadísticas básicas y herramientas de interacción directa sobre la figura. Entre estas últimas se encuentran un cursor cruzado, una anotación flotante con tiempo y voltaje y un menú contextual para exportar imágenes o activar y desactivar la grilla. La presencia de

estas funciones no altera la lógica principal del programa, pero mejora la utilidad práctica de la interfaz durante la inspección de la señal.

Dado que el número de muestras puede crecer rápidamente, la aplicación implementa un procedimiento de reducción de datos para la visualización mediante la función `downsample_minmax()`. Esta técnica agrupa las muestras en ventanas y conserva mínimos y máximos locales, reduciendo la carga gráfica sin perder la forma general de la señal ni sus variaciones extremas. Este criterio resulta más adecuado que un simple submuestreo uniforme, ya que evita descartar picos o valles aislados que podrían ser relevantes desde el punto de vista experimental.

Durante la adquisición en vivo, la aplicación también puede registrar las muestras en un archivo CSV local. Esta función se activa desde la interfaz mediante una casilla de selección. Cuando el registro se habilita, se solicita al usuario la ruta del archivo y se crea un CSV con cabecera normalizada:

```
h,m,s,ms,mV
```

Cada nueva muestra válida se escribe línea a línea en ese formato, descomponiendo el tiempo absoluto en horas, minutos, segundos y milisegundos. Por ejemplo, una fila válida puede ser

```
0,12,34,250,2010
```

donde el último campo representa el voltaje en milivoltios. Después de cada escritura se fuerza un *flush* del archivo, minimizando la pérdida de datos ante un cierre inesperado. De forma opcional, el usuario puede almacenar una versión suavizada de la señal en lugar del valor instantáneo, utilizando la misma media móvil configurada para la visualización.

La aplicación también permite exportar la ventana visible de la gráfica en vivo a un nuevo archivo CSV. En este caso, el tiempo se reescribe relativo al inicio de la ventana seleccionada, manteniendo el mismo formato de cabecera. Esta opción resulta útil cuando se desea conservar un tramo particular del registro sin exportar toda la adquisición.

La vista de análisis fuera de línea reutiliza gran parte de estas herramientas y se muestra en la Figura 4.10, pero aplicada a archivos ya existentes. La aplicación admite tres variantes principales de entrada: archivos CSV normalizados con cabecera `h,m,s,ms,mV` y archivos con columnas del tipo `tiempo,voltaje_mV`. Esto vuelve a la herramienta más flexible frente a distintas fuentes de datos y simplifica el reprocesamiento de registros generados en etapas anteriores del desarrollo.

Una vez cargado un archivo, los tiempos se normalizan respecto a la primera muestra y se almacenan en arreglos `numpy`. A partir de ahí, la señal se representa en una gráfica independiente con las mismas opciones de selección de intervalo, autoescala y suavizado. También en este caso la ventana visible puede exportarse a un CSV. Si el usuario definió una fecha base válida, la exportación se realiza con cabecera `datetime,mV`; de lo contrario, se mantiene el formato temporal relativo `h,m,s,ms,mV`.

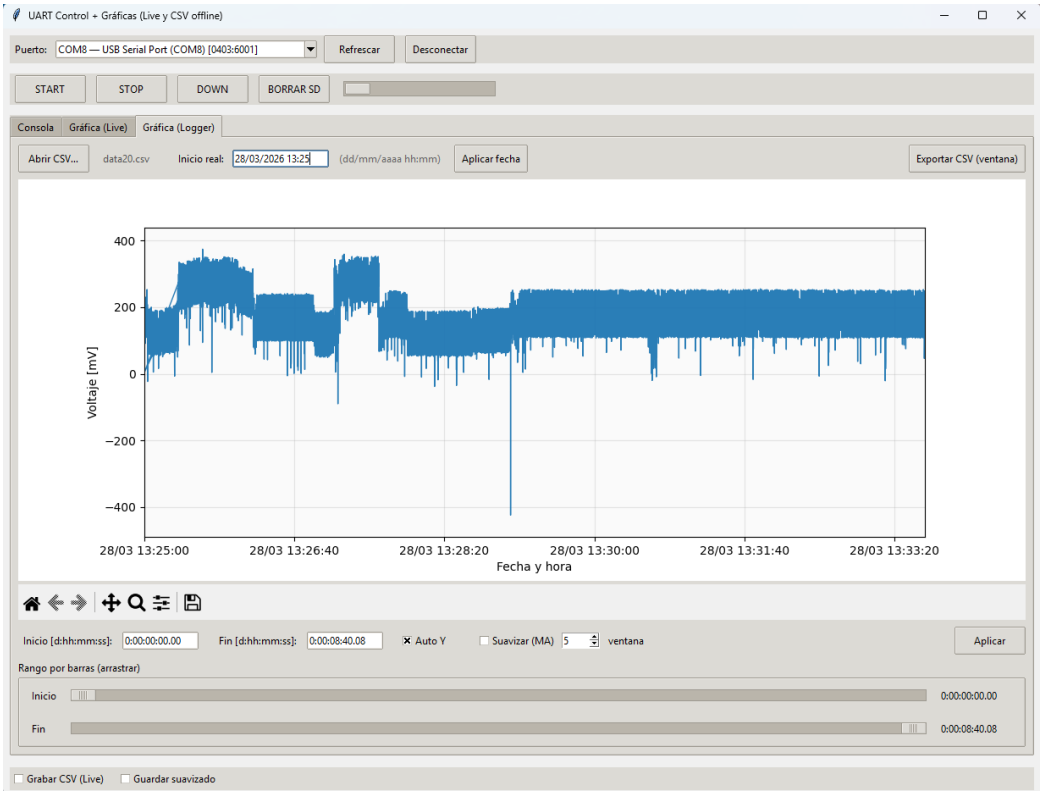


Figura 4.10: Vista de análisis fuera de línea.

La aplicación incorpora además una función para asociar al registro una fecha y hora de inicio reales. Esta fecha puede ingresarse en distintos formatos, por ejemplo `dd/mm/aaaa hh:mm`, `dd/mm/aa hh:mm`, `dd/mm hh:mm` o `aaaa-mm-dd hh:mm`. Si la fecha es válida, el eje horizontal de la gráfica offline deja de expresarse como tiempo relativo y pasa a mostrarse como fecha y hora absolutas. Por ejemplo, si se fija como inicio real:

```
06/02/2026 14:30
```

una muestra ubicada a 3600 segundos del comienzo pasa a corresponder a las 15:30 del mismo día. Esta capacidad resulta especialmente útil al trabajar con archivos descargados desde el modo *Logger*, donde interesa relacionar el registro con eventos externos ocurridos durante el ensayo.

Por último, el cierre de la aplicación se realiza de forma controlada. Antes de destruir la ventana principal, se solicita la detención de los hilos activos, se cierra la conexión serie, se cierran los archivos CSV que puedan estar abiertos y se liberan los recursos asociados a la interfaz. Esto reduce la probabilidad de corrupción de archivos y evita dejar el puerto serie en un estado inconsistente.

4.2.5. Bibliotecas utilizadas

La aplicación fue implementada en *Python* utilizando bibliotecas estándar y de terceros. La interfaz gráfica se desarrolló con `tkinter` [24], que proporciona los widgets principales de la ventana, los cuadros de diálogo y los controles de interacción. La concurrencia se resolvió mediante `threading` [23] y `queue` [22], utilizadas para ejecutar la lectura serie y la descarga en hilos separados y transferir la información al hilo principal de forma segura.

La comunicación serie con el dispositivo se implementó con *PySerial* [14]. Esta biblioteca se emplea tanto para la apertura del puerto y el envío de comandos como para la detección de puertos disponibles en el sistema. El procesamiento numérico de las señales se realizó con `numpy` [18], utilizado en las operaciones de suavizado, cálculo estadístico, conversión a arreglos y manejo de máscaras temporales. La visualización gráfica se implementó con `matplotlib` [15], embebiendo las figuras directamente dentro de la interfaz.

Para la lectura y escritura de archivos se utilizaron las bibliotecas `csv` [20] y `datetime` [21]. La primera se emplea en el registro local de adquisiciones, en la apertura de archivos externos y en la exportación de ventanas visibles. La segunda se utiliza en la interpretación de fechas base y en la reconstrucción de ejes temporales absolutos para los registros offline.

En conjunto, estas bibliotecas permitieron implementar una herramienta única para conexión, adquisición, descarga y análisis, manteniendo una estructura relativamente simple y fácilmente replicable. Desde el punto de vista de un desarrollo futuro, la aplicación puede reutilizarse como base para otros sistemas de adquisición que mantengan el mismo esquema de comunicación y formatos de datos.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 5

Pruebas y mediciones

5.1. Medidas de la electrónica

Los nombres de los puntos de operación en continua presentados en la Tabla 5.1, coinciden con los de la Figura 3.6. Se observa que las alimentaciones del sistema resultan prácticamente simétricas, con valores de $V_{CC} = 3{,}29$ V y $V_{SS} = -3{,}24$ V, lo que indica un funcionamiento adecuado de la etapa de generación de tensiones. Asimismo, bajo condición de entrada nula, las distintas etapas de la cadena analógica presentan prácticamente el mismo nivel DC: $V_{\text{out}}^{\text{Buff1}}$, $V_{\text{out}}^{\text{Notch}}$, $V_{\text{out}}^{\text{Filtros}}$ y $V_{\text{out}}^{\text{A0}}$ toman un valor de 1,646 V, mientras que la salida final resulta 1,645 V.

Estos resultados indican que no se producen desplazamientos apreciables del punto de operación entre etapas, ni acumulación significativa de offset en continua.

Además, el valor observado en la salida coincide prácticamente con la mitad de la alimentación positiva, es decir, $V_{CC}/2$, lo cual confirma que la cadena principal de señal se encuentra correctamente centrada en ese nivel y, por tanto, se aprovecha al máximo el rango de $0\,V$ a $3,3\,V$ para la entrada del conversor analógico digital.

El único valor de tensión que, en primera instancia, no resulta coherente es V_{REF1}. Este debería coincidir aproximadamente con $V_{A0-} \approx V_{CC}/4$, dado que ambos puntos corresponden a las entradas del amplificador operacional y, bajo la hipótesis de cortocircuito virtual, sus tensiones deberían ser prácticamente iguales.

La diferencia observada se atribuye a la resistencia de entrada del voltímetro utilizado durante la medición. Dicho instrumento presenta una resistencia de entrada de $10\,M\Omega$, valor que no puede despreciarse frente a las resistencias empleadas para generar V_{REF1}, de $3\,M\Omega$ y $1\,M\Omega$. Por lo tanto, al conectar el voltímetro, se modifica el divisor resistivo original y se obtiene una tensión medida distinta de la esperada.

En el Anexo D se desarrolla esta justificación de forma analítica. Allí se demuestra que, al considerar el efecto de carga introducido por el voltímetro, el valor corregido de la tensión resulta $V_{REF1}^{AJUSTADO} \approx 0{,}82\,V$, lo cual es consistente con el resto de las tensiones medidas en el circuito.

Las tres mediciones de respuesta en frecuencia, presentadas en las Figuras 5.1, 5.2 y 5.3, exhiben un comportamiento global muy similar. En todos los casos se

Tabla 5.1: Tensiones medidas en distintos nodos del circuito.

Nodo	**Valor**
V_{CC}	3,29 V
V_{SS}	−3,24 V
$V_{\mathrm{A0-}}$	0,82 V
V_{REF1}	0,77 V
$V_{\mathrm{out}}^{\mathrm{BUF1}}$	1,646 V
$V_{\mathrm{out}}^{\mathrm{Notch}}$	1,646 V
$V_{\mathrm{out}}^{\mathrm{Filtros}}$	1,646 V
$V_{\mathrm{out}}^{\mathrm{A0}}$	1,646 V
$V_{\mathrm{out}}^{\mathrm{BUF2}}$	1,645 V

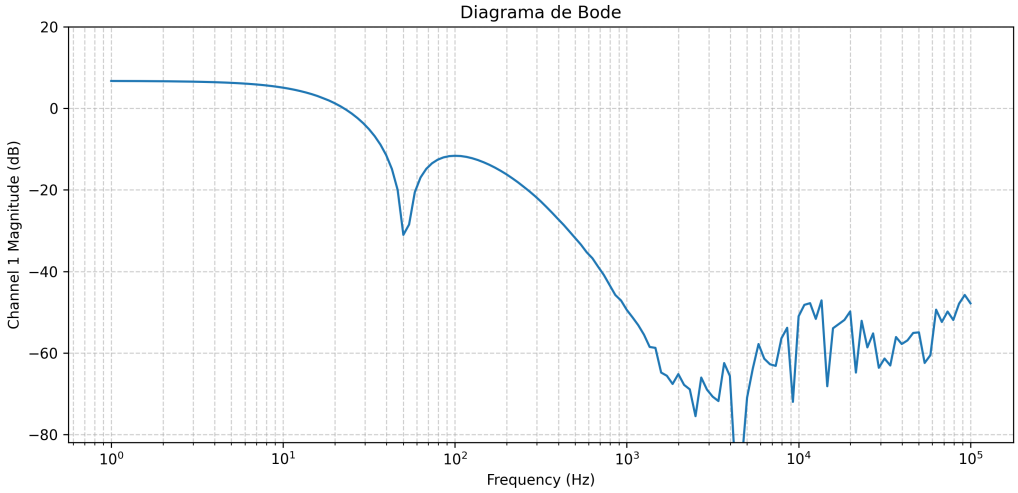


Figura 5.1: Diagrama de Bode medido para la configuración con batería y electrodos de aguja de plata, procesado con python

observa una ganancia aproximadamente constante $G = 7\,dB$ a bajas frecuencias, seguida de un rechazo pronunciado en torno a la frecuencia de red y una posterior atenuación creciente hacia frecuencias más elevadas. La coincidencia en la posición del mínimo asociado al filtro *notch*, visible en las tres configuraciones ensayadas, así como la similitud de la pendiente de atenuación posterior, indican que la dinámica principal del sistema se encuentra dominada por la cadena de filtrado diseñada, sin variaciones sustanciales atribuibles ni al tipo de electrodo ni al modo de alimentación.

Al comparar en particular las Figuras 5.1 y 5.2, correspondientes a mediciones realizadas con batería y con electrodos de distinto material, no se aprecian diferencias relevantes dentro del rango de frecuencias de mayor interés para el filtrado, que son hasta $1\,kHz$ aproximadamente.

De forma análoga, la comparación entre las Figuras 5.2 y 5.3 sugiere que la

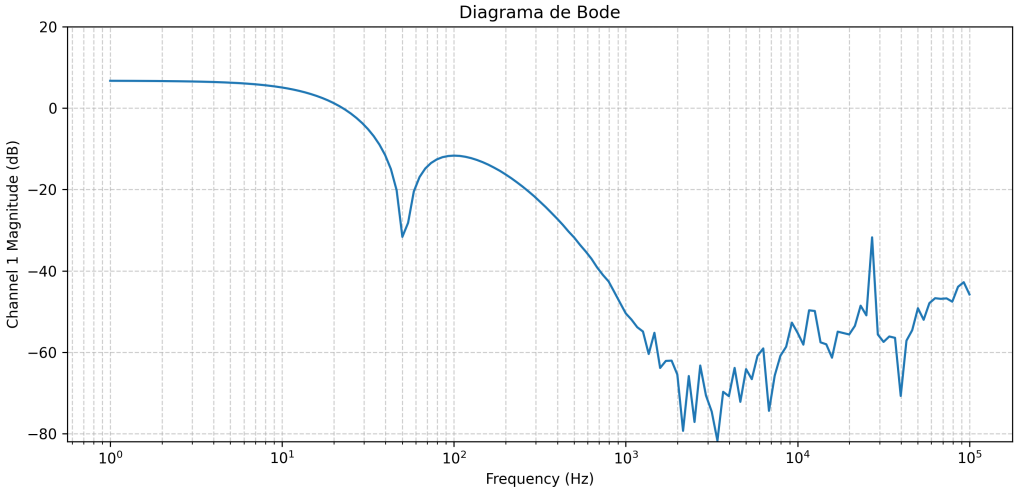


Figura 5.2: Diagrama de Bode medido para la configuración con batería y electrodos de aguja de acero, procesado con python

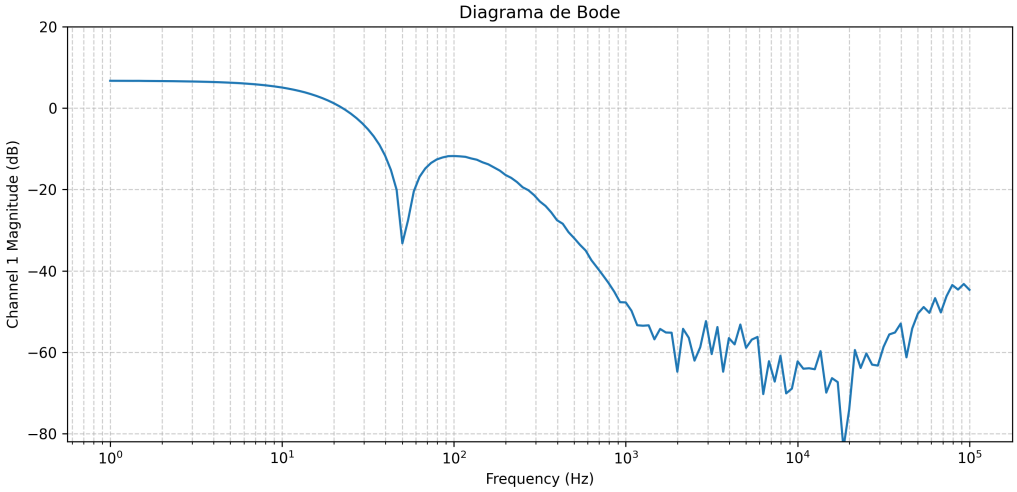


Figura 5.3: Diagrama de Bode medido para la configuración sin batería y electrodos de aguja de acero, procesado con python

presencia o ausencia de batería tampoco introduce modificaciones apreciables en la respuesta global del sistema dentro de la banda analizada.

Las diferencias entre curvas se vuelven más notorias únicamente a frecuencias elevadas, donde aparecen picos y valles de distinta magnitud según la configuración, como puede observarse en las Figuras 5.1, 5.2 y 5.3. Dado que estas discrepancias se presentan fuera de la banda de mayor interés para la aplicación, resulta razonable asociarlas principalmente a efectos parásitos del montaje, del cableado o del propio procedimiento de medición, más que a una modificación relevante del comportamiento funcional del sistema.

En consecuencia, a partir de las mediciones mostradas en las Figuras 5.1, 5.2 y 5.3, puede concluirse que tanto el uso de batería como el material del electrodo no introducen cambios significativos en la respuesta en frecuencia dentro del rango útil evaluado. No obstante, debe señalarse que las mediciones realizadas comienzan en 1 Hz, por lo que estos resultados no permiten extraer conclusiones directas sobre el comportamiento del sistema a frecuencias inferiores a dicho valor.

Por otra parte, la respuesta temporal del sistema ante una excitación senoidal de 2 Hz de frecuencia y 750 mV de amplitud, mostrada en la Figura 5.4, evidencia que la salida conserva el carácter periódico de la señal de entrada y estable, sin recortes notorios ni deformaciones abruptas en el dominio temporal. Por su parte, el análisis espectral de dicha respuesta, presentado en la Figura 5.5, permite identificar una componente fundamental dominante en 2 Hz, correspondiente a la frecuencia de excitación aplicada. Asimismo, se observan armónicos de orden superior y otras componentes de menor amplitud, lo que indica que la señal de salida no es perfectamente senoidal. Sin embargo, la diferencia de magnitud entre la componente fundamental y estas componentes secundarias muestra que la mayor parte de la energía de la señal se concentra en la frecuencia de excitación, mientras que las restantes contribuciones permanecen acotadas.

A efectos de este trabajo, se adopta como criterio práctico para definir una excursión de salida aceptable que la diferencia entre la componente fundamental y el armónico de mayor amplitud sea de al menos 40 dB.

En este sentido, la señal ya presentada en la Figura 5.4 la cual su FFT se encuentra en la Figura 5.5. Es la maxima excursión de salida siendo esta [0,04, 3,25].

5.2. Consumo

Para medir el consumo electrico del sistema operando en modo *Logger* se colocó una resistencia de $1,5\,\Omega$ en serie con la alimentación de la batería.

Con un osciloscopio digital se registó la caida de voltaje en la resistencia. En la Figura 5.6 se registra la caida en forma de corriente por la resistencia. Se pueden ver picos cada aproximadamente $20\,ms$ de $14\,mV/1,5\,\Omega = 9,3\,mA$. Estos corresponden al periodo de las interrupciones generadas por el timer, que despiertan al microcontrolador. El resto del tiempo, que el microcontrolador se encuentra en bajo consumo se tiene un consumo de aproximadamente $4\,mV/1,5\,\Omega = 2,67\,mA$. Cabe aclarar que aca no solo se registra lo que consume el microcontrolador si no que es la suma con toda la electrónica.

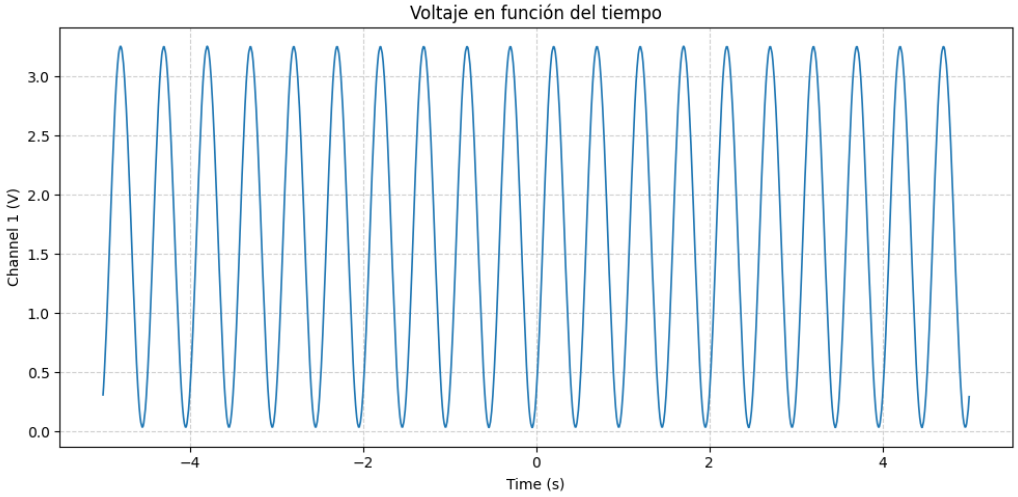


Figura 5.4: Excursión temporal de la salida obtenida a partir del procesamiento en Python.

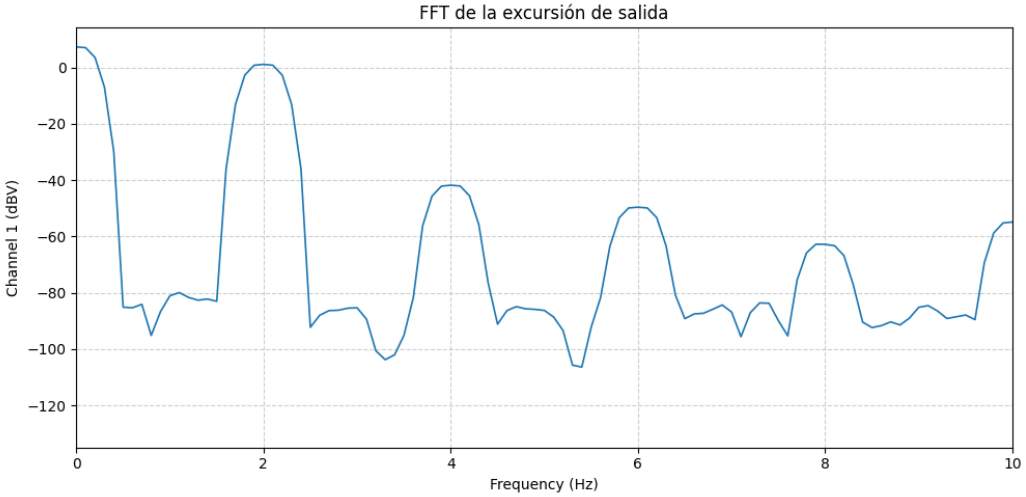


Figura 5.5: FFT de la excursión de salida obtenida a partir del procesamiento en Python.

Ampliando el rango de tiempo de captura se observa lo que se muestra en la Figura 5.7. Aqui se observa que cada $5\,s$ se tienen pequeños picos de consumo que corresponden al momento en donde se prende el led de encendido. Además se tienen dos grandes picos de aproximadamente $25\,mA$, en la Figura 5.8 se muestra ampliado en este pico, este es el mayor consumo del sistema y corresponde a el momento donde se guardan las muestras en la SD.

Este estudio del consumo es coherente con el comportamiento esperado del sistema. En particular permite comprobar que el microcontrolador esta efectivamente entrando en bajo consumo, lo cual ha quedado evidenciado ser fundamental para prolongar la autonomía del sistema.

Adicionalmente, con el objetivo de estimar la autonomía del sistema operando

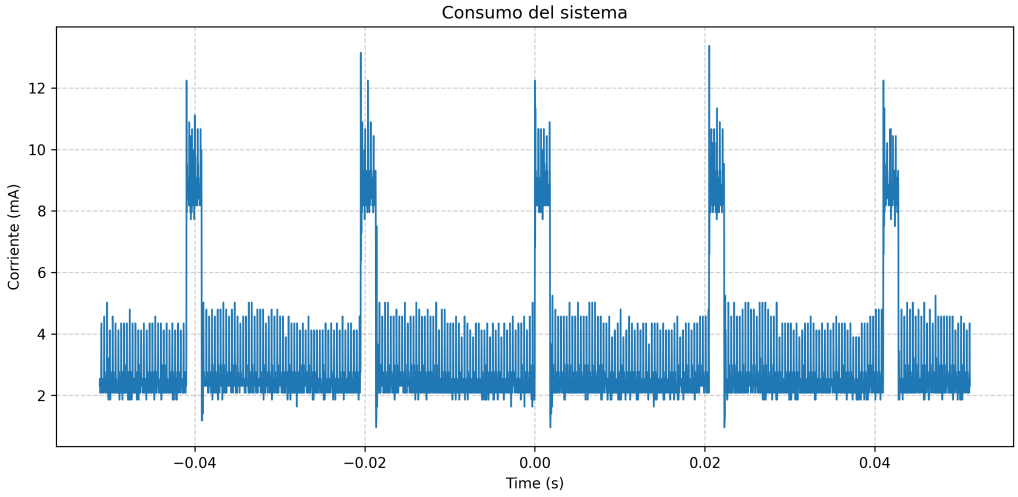


Figura 5.6: Consumo del temporizador, procesada en Python.

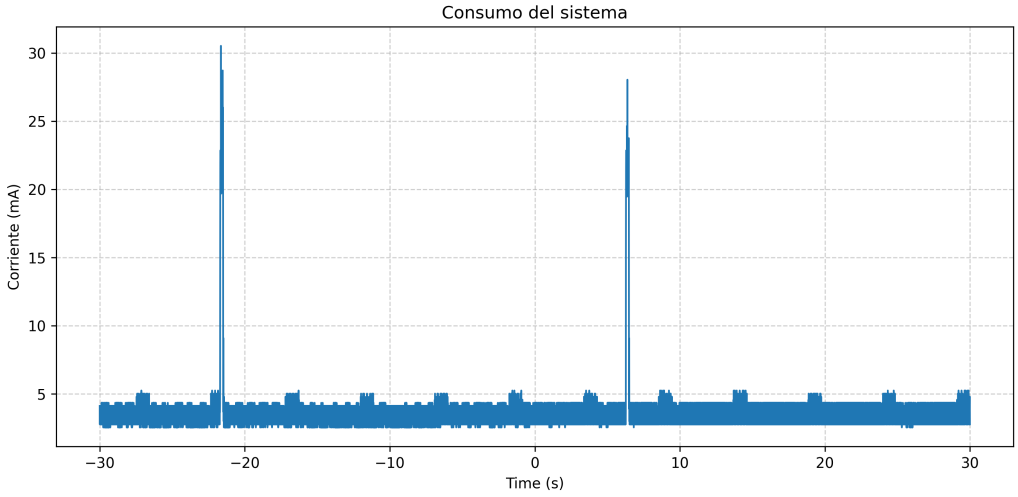


Figura 5.7: Consumo total del sistema en función del tiempo, procesado en Python.

de forma continua en modo *Logger*, se realizó un ensayo específico de descarga de batería. Para ello, se modificó el firmware del dispositivo, sustituyendo temporalmente la señal adquirida por la medición del voltaje de la batería. Esta modificación no altera de forma apreciable el consumo del sistema, ya que únicamente se cambió la entrada analógica leída por el microcontrolador, manteniéndose igual el resto de la lógica de adquisición, almacenamiento y funcionamiento general del equipo. De esta forma, el sistema registró la evolución de la tensión de alimentación durante su operación normal, hasta que la batería se descargó y el dispositivo se apagó por completo.

El resultado del ensayo se presenta en la Figura 5.9. A partir de la medición realizada, se obtuvo que el sistema demoró 20 días, 8 horas y 53 minutos en apa-

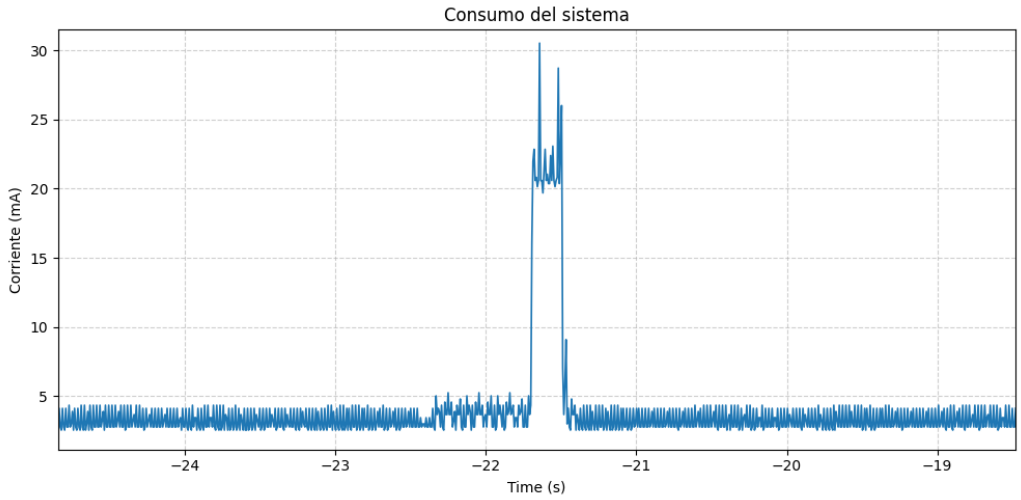


Figura 5.8: Detalle del pico de consumo cercano a 30 mA obtenido a partir del procesamiento en Python.

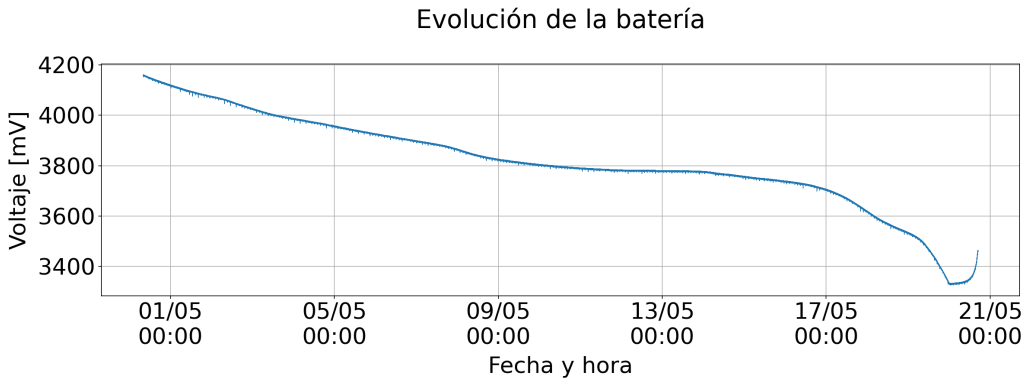


Figura 5.9: Voltaje de la batería en función del tiempo

garse por completo. Cabe aclarar que, para realizar este experimento, se partió de la batería cargada mediante dos ciclos de carga, como se mencionó en la Sección 4.1.3. Esta condición inicial se considera equivalente a una carga aproximada de $1400\,mAh$.

5.3. Estimación del tamaño de los archivos generados

Cada muestra adquirida por el sistema se almacena en la tarjeta SD en formato CSV, utilizando dos campos por línea: el tiempo transcurrido en milisegundos y el valor de la señal en milivoltios. En consecuencia, cada registro se escribe con la estructura:

```
tiempo_ms,voltaje_mV
```

seguida por un salto de línea.

A diferencia de un almacenamiento binario, en el cual cada variable ocupa una cantidad fija de bytes, en un archivo CSV el tamaño de cada muestra depende de la cantidad de caracteres necesarios para representar numéricamente ambos valores. Por tanto, el tamaño de cada línea puede estimarse como:

$$B_{\text{muestra}} = d(ms) + d(mV) + 2$$

donde $d(ms)$ es la cantidad de dígitos del tiempo en milisegundos, $d(mV)$ es la cantidad de caracteres del valor en milivoltios, y el término adicional 2 corresponde a la coma separadora y al salto de línea.

Para un registro continuo de 24 horas, el tiempo máximo almacenado es:

$$24 \cdot 60 \cdot 60 \cdot 1000 = 86\,400\,000 \text{ ms}$$

por lo que el campo temporal requiere hasta 8 caracteres. Considerando además un valor de tensión del tipo `-1650`, el campo de amplitud puede requerir hasta 5 caracteres. En ese caso, una muestra en el peor caso ocupa aproximadamente:

$$8 + 5 + 2 = 15 \text{ bytes}$$

Si el sistema opera a una frecuencia de muestreo de 50 Hz, la cantidad total de muestras generadas en 24 horas es:

$$N = 24 \cdot 60 \cdot 60 \cdot 50 = 4\,320\,000$$

Por lo tanto, el tamaño aproximado máximo del archivo diario resulta:

$$B_{\text{día}} = 4\,320\,000 \cdot 15 = 64\,800\,000 \text{ bytes}$$

es decir, aproximadamente 64,8 MB.

Esta estimación constituye una cota superior razonable para un archivo de 24 horas de duración. Si el registro se prolongara por más tiempo dentro de un mismo archivo, el tamaño de cada muestra aumentaría, ya que el valor temporal demandaría una mayor cantidad de dígitos. En consecuencia, para un archivo diario, resulta razonable considerar un tamaño aproximado de 65 MB.

La decisión de utilizar archivos CSV se basó en la simplicidad del esquema de almacenamiento adoptado. Cada muestra queda registrada mediante dos campos, correspondientes al tiempo y al voltaje medido, sin requerir estructuras adicionales ni etapas de interpretación complejas. Esto permite acceder al contenido de los archivos de forma directa y simplifica su procesamiento posterior.

Si bien el almacenamiento en formato binario habría permitido reducir el tamaño total de los archivos, dicha ventaja no constituía una restricción crítica para el sistema desarrollado. En contrapartida, esta alternativa habría requerido incorporar una etapa adicional de decodificación en la interfaz. Por este motivo, se optó por priorizar un formato simple, legible y de uso directo, aun a costa de un mayor tamaño de almacenamiento.

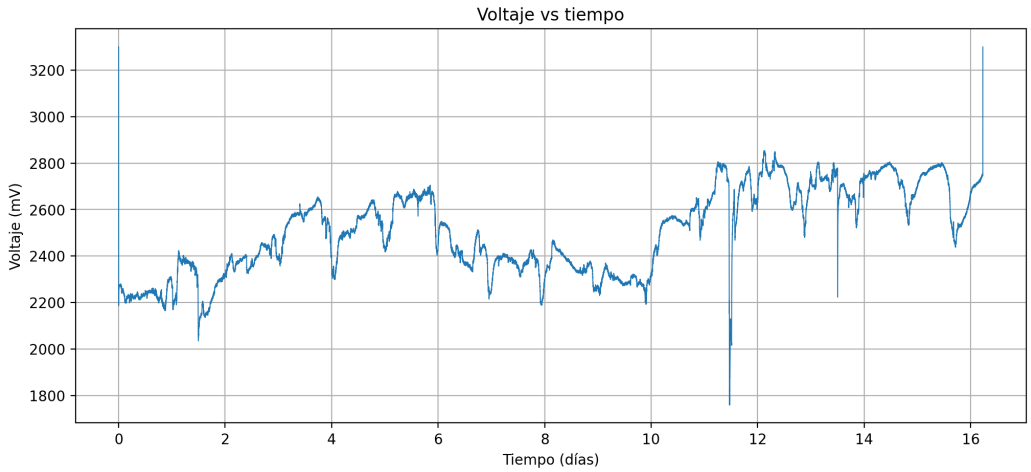


Figura 5.10: Primera prueba realizada con electrodos tipo aguja de acero inoxidable.

5.4. Pruebas funcionales

A continuación se presenta la primera prueba realizada con el sistema completo, mostrada en la Figura 5.10.

Para esta medición se utilizaron electrodos tipo aguja de acero inoxidable, insertados directamente en la planta. Uno de ellos se ubicó en la base del tronco y el otro en una rama alejada de ese punto, con el objetivo de medir la diferencia de potencial entre ambas posiciones.

En la señal temporal obtenida no se identificó un patrón claro. Por este motivo, se realizó una Transformada Rápida de Fourier (FFT) con el fin de analizar su contenido en frecuencia.

Antes de calcular la FFT se eliminó el valor medio de la señal y se aplicó una ventana de Hann. La eliminación del valor medio permite suprimir la componente continua dominante, mientras que la ventana reduce los efectos de discontinuidad asociados al análisis de un registro temporal finito.

La magnitud de la FFT se representó en decibeles, tomando como referencia el valor máximo del espectro mostrado. De esta forma, el pico dominante queda ubicado en 0 dB y el resto de las componentes se expresa de forma relativa respecto a dicho máximo. Esta representación facilita la comparación entre componentes de distinta magnitud, ya que permite visualizar con mayor claridad señales de menor peso espectral. En este contexto, una componente ubicada a -20 dB presenta una magnitud diez veces menor que la componente de referencia, mientras que una componente a -40 dB presenta una magnitud cien veces menor.

El resultado se muestra en la Figura 5.11. Dado que el contenido de interés se concentra en frecuencias muy bajas, el eje horizontal se expresó en ciclos por día en lugar de Hertz.

En la FFT se observa un pico principal próximo a cero ciclos por día. También aparecen componentes de menor magnitud en torno a 1 y 2 ciclos por día. La componente próxima a 1 ciclo por día corresponde a una variación con período

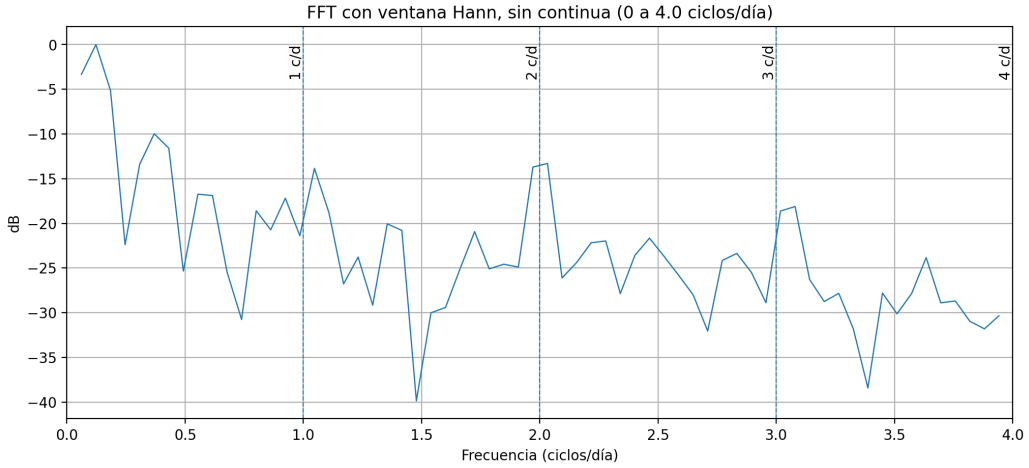


Figura 5.11: FFT de la primera captura, expresada en ciclos por día, sin componente continua y con ventana de Hann.

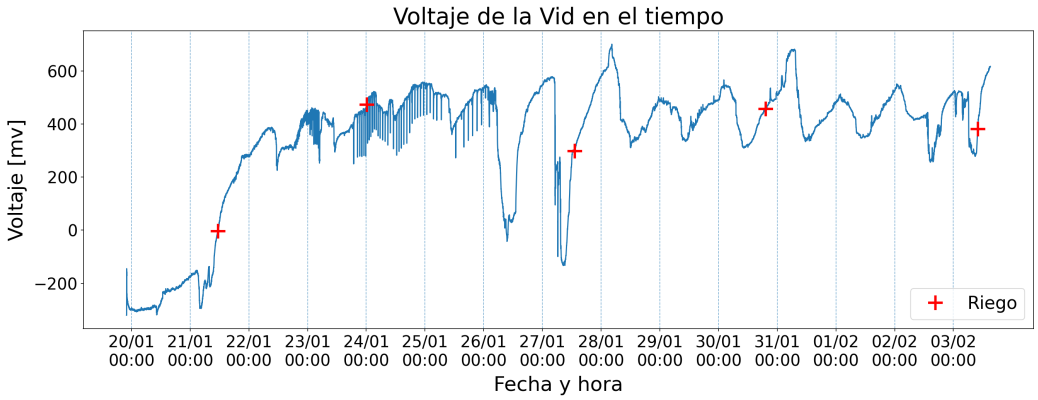


Figura 5.12: Segunda captura realizada con el dispositivo. Indicando con cruces rojas los momentos de riego.

aproximado de 24 horas, mientras que la componente próxima a 2 ciclos por día corresponde a una variación con período aproximado de 12 horas.

La figura se presenta ampliada en la zona de bajas frecuencias, ya que no se observaron componentes relevantes a frecuencias mayores. Debe señalarse, además, que en este rango la FFT contiene pocos puntos, debido a que la separación entre componentes frecuenciales depende de la duración total del registro. En consecuencia, una captura de mayor duración permitiría mejorar la resolución espectral y distinguir con mayor precisión las componentes de muy baja frecuencia.

En la segunda captura se utilizó el mismo tipo de electrodo, pero se mejoró la forma de colocación. En este caso, los electrodos se introdujeron de forma casi tangencial a las ramas y al tronco, y además se fijaron con cinta para reducir movimientos durante la medición. El resultado se muestra en la Figura 5.12.

Un aspecto a señalar es la componente continua de la señal. Para esta instancia, en el firmware se decidió restar la continua impuesta por la electrónica, asociada

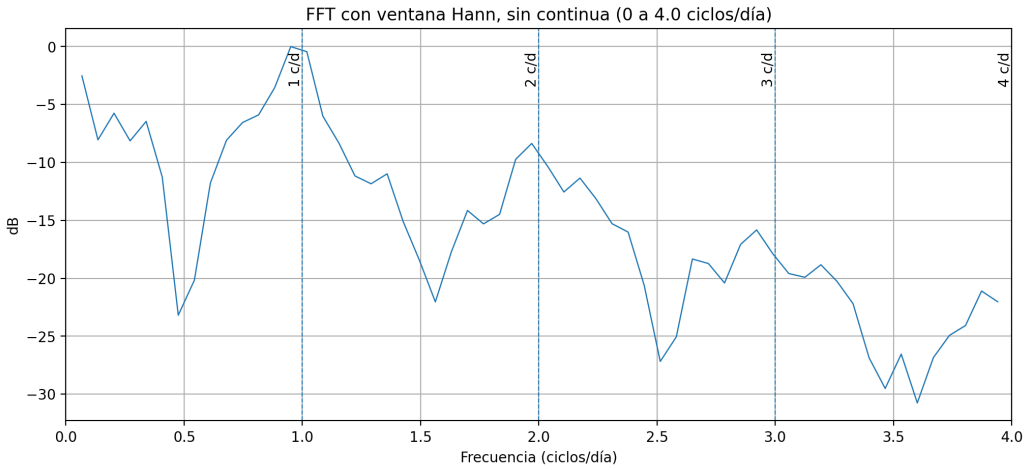


Figura 5.13: FFT de la segunda captura, expresada en ciclos por día, sin componente continua y con ventana de Hann.

al sesgo de salida en $V_{cc}/2$. Por este motivo, la señal quedó centrada en cero.

Como se observa en la figura, esta segunda captura presenta un resultado más claro que el anterior. Se distinguen ciclos de aproximadamente 24 horas, así como una diferencia entre los períodos diurno y nocturno. En particular, la señal presenta un máximo alrededor de la medianoche y un mínimo en torno al mediodía.

En la misma figura se indican además los eventos de riego mediante cruces rojas, con el objetivo de evaluar si estos producen variaciones en la señal. A partir de esta observación, se aprecia que durante los períodos de mayor deshidratación la amplitud de la señal tiende a aumentar.

Una evidencia adicional de este comportamiento se presenta en la Figura 5.13, donde se muestra la FFT correspondiente a esta segunda captura. En este caso, se observa una componente dominante próxima a 1 ciclo por día, consistente con la periodicidad aproximada de 24 horas observada en la señal temporal. También se identifican componentes de menor magnitud en frecuencias superiores, que pueden asociarse a armónicos de la variación diaria o a otros efectos de menor escala temporal.

5.5. Pruebas preliminares en Mimosa Púdica

Antes de disponer de ejemplares de vid para los ensayos principales, se realizaron pruebas preliminares con una *Mimosa pudica*, con el objetivo de evaluar el comportamiento general de los primeros prototipos del sistema de adquisición. La elección de esta planta estuvo motivada por su conocida respuesta visible frente a estímulos mecánicos, lo que permitía contar con una referencia cualitativa inmediata durante las primeras verificaciones experimentales.

En este contexto, la Figura 5.14 presenta un registro de aproximadamente 30 minutos obtenido durante una serie de estímulos mecánicos aplicados con un objeto de madera sobre distintas zonas de la planta. En cada caso se observó visualmente

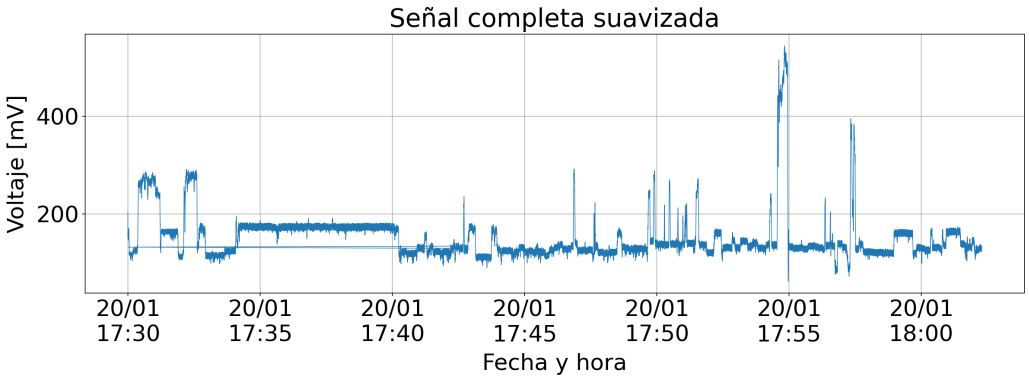


Figura 5.14: Registro de 30 minutos obtenido sobre una *Mimosa pudica* mediante electrodos de acero. La señal fue suavizada con una ventana de 50 muestras.

el cierre de las hojas, y entre estímulos se dejó a la planta en reposo durante algunos minutos.

La señal registrada presenta variaciones transitorias de amplitud y algunos picos de mayor magnitud. Si bien en esta experiencia no se registraron de forma precisa los instantes de aplicación de cada estímulo, el ensayo resultó útil como validación preliminar del funcionamiento del prototipo y como antecedente de las pruebas posteriores realizadas sobre vid. La señal mostrada fue suavizada con una ventana de 50 muestras para facilitar su visualización.

5.5.1. Ensayo prolongado sobre *Mimosa pudica*

Una vez integrado el prototipo final, se realizaron nuevos ensayos sobre una planta de *Mimosa pudica*. El registro se llevó a cabo desde el 30 de abril de 2026 a las 8:00 h hasta el 5 de mayo de 2026 aproximadamente a las 16:00 h. El objetivo fue evaluar el comportamiento del sistema en una adquisición prolongada, utilizando ambos prototipos disponibles y comparando el uso de electrodos de acero y electrodos de plata.

El armado experimental se muestra en la Figura 5.15. Ambos prototipos fueron conectados simultáneamente sobre la misma planta, ubicando los electrodos en posiciones próximas. De esta forma, las señales fueron adquiridas bajo condiciones ambientales y fisiológicas equivalentes, diferenciándose principalmente por el material del electrodo utilizado.

Cabe señalar que el ensayo fue realizado durante el otoño, período en el cual la actividad de la *Mimosa pudica* es reducida. Esta especie presenta una respuesta más marcada ante la presencia de radiación solar directa, por lo que era esperable obtener señales de baja amplitud y variaciones lentas durante el registro.

Las señales adquiridas fueron procesadas mediante una media móvil de 50 muestras, con el fin de reducir ruido y facilitar la observación de las tendencias principales. En las Figuras 5.16 y 5.17 se muestran los registros completos obtenidos con electrodos de acero y plata, respectivamente. En ambos casos se observa

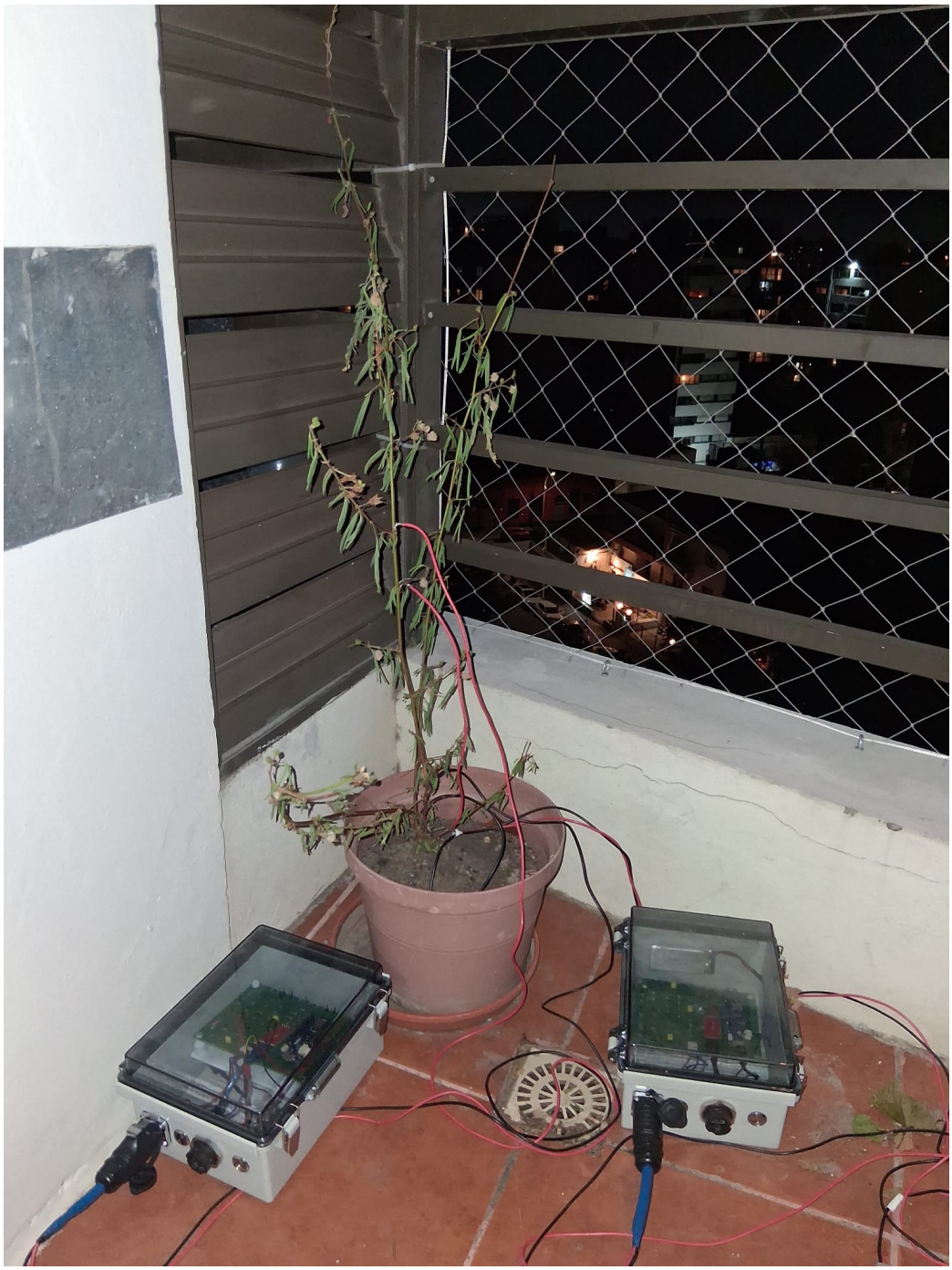


Figura 5.15: Armado experimental de la prueba simultánea sobre *Mimosa pudica*.



Figura 5.16: Registro completo obtenido sobre *Mimosa pudica* mediante electrodos de acero. La señal fue suavizada mediante una media móvil de 50 muestras.



Figura 5.17: Registro completo obtenido sobre *Mimosa pudica* mediante electrodos de plata. La señal fue suavizada mediante una media móvil de 50 muestras.

que la señal presenta variaciones lentas a lo largo de varios días, junto con transitorios de mayor amplitud en instantes puntuales.

La comparación entre ambos registros se presenta en la Figura 5.18. Se observa una evolución temporal muy similar en las dos mediciones, lo que respalda que la señal registrada está asociada al potencial eléctrico de la planta y no únicamente a ruido propio del sistema de adquisición o a un artefacto aislado de un electrodo. Esta semejanza resulta especialmente relevante dado que las mediciones fueron realizadas con dos prototipos independientes.

Si bien ambas señales presentan una tendencia general semejante, se aprecian algunas diferencias. En particular, el registro obtenido con electrodos de plata presenta inicialmente un desplazamiento de continua mayor, que tiende a reducirse con el transcurso del tiempo. Esto puede estar asociado a la estabilización de la interfaz electrodo-planta durante los primeros instantes de contacto. Además, la señal correspondiente al electrodo de plata presenta una evolución ligeramente más suave, con menor presencia de variaciones abruptas.

En términos generales, ambos electrodos permitieron registrar un comporta-

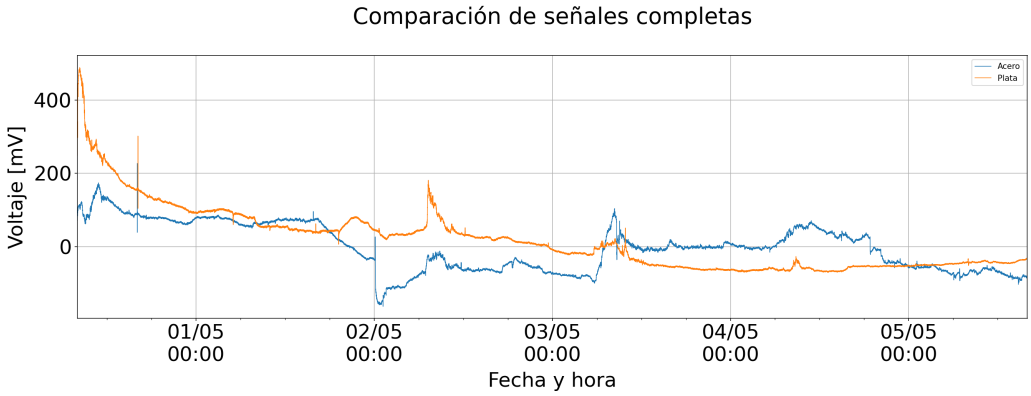


Figura 5.18: Comparación entre los registros obtenidos con electrodos de acero y plata sobre *Mimosa pudica*.

miento compatible con el esperado para esta condición experimental. Sin embargo, las diferencias observadas evidencian la importancia de la conexión electrodo y planta en este tipo de mediciones. Por este motivo, para trabajos futuros resulta necesario profundizar en la repetibilidad de los ensayos y controlar mejor las condiciones de contacto.

5.5.2. Respuesta ante estímulo mecánico

Además del registro prolongado, se analizaron segmentos temporales asociados a estímulos mecánicos aplicados sobre la planta. En particular, se consideraron instantes en los que se produjo contacto o perturbación sobre la *Mimosa pudica*, con el fin de observar si el sistema era capaz de registrar variaciones transitorias de la señal.

En las Figuras 5.19 y 5.20 se muestran ejemplos de estos eventos para electrodos de acero y plata. En ambos casos se observa una variación abrupta de la tensión medida, seguida de una recuperación parcial hacia un nuevo nivel de equilibrio. Este comportamiento es consistente con la presencia de un evento transitorio registrado por el sistema.

5.6. Comparación entre electrodos de acero y plata

Se realizó una comparación entre la señal obtenida en la vid utilizando dos electrodos de tipo aguja, uno de acero y otro de plata. Las mediciones no se efectuaron de forma simultánea, primero se realizó una captura con uno de los electrodos y luego con el otro. Debe señalarse que estas mediciones se llevaron a cabo a fines de Marzo, es decir, en otoño. Dado que la vid reduce marcadamente su actividad fisiológica en esta época del año, no se esperaba obtener una señal representativa del comportamiento diario de la planta. En este contexto, el objetivo de la prueba no fue identificar ciclos fisiológicos definidos, sino verificar la presencia de una

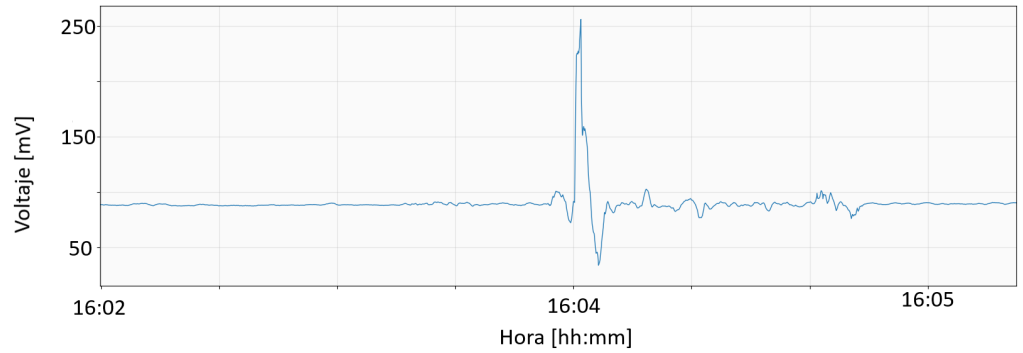


Figura 5.19: Segmento temporal asociado a un estímulo mecánico sobre *Mimosa pudica*, registrado mediante electrodos de acero. La señal fue suavizada mediante una media móvil de 50 muestras.

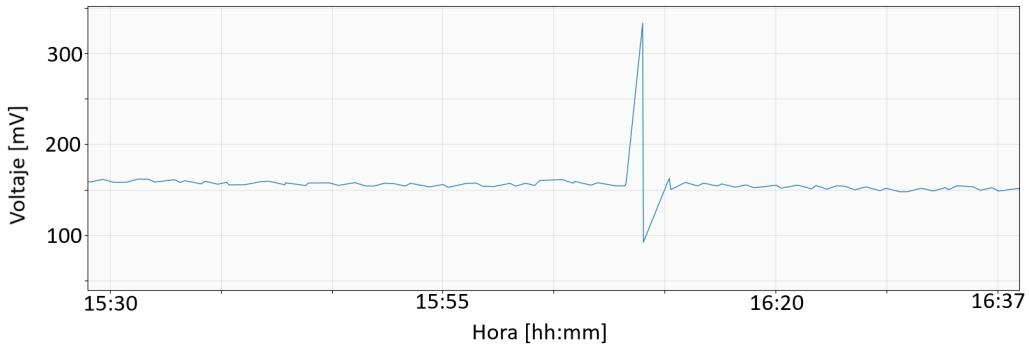


Figura 5.20: Segmento temporal asociado a un estímulo mecánico sobre *Mimosa pudica*, registrado mediante electrodos de plata. La señal fue suavizada mediante una media móvil de 50 muestras.

señal continua, sin fijación en un valor constante y con una cantidad acotada de discontinuidades.

En la Figura 5.21 se presenta la señal registrada durante una semana utilizando el electrodo de acero. Se trata de una captura de corta duración, realizada con el propósito de evaluar de forma rápida si el sistema era capaz de registrar una señal continua y sin comportamientos anómalos evidentes. En la figura no se observa un ciclo claro, lo cual resulta consistente con la época del año en que se efectuó la medición. No obstante, la señal presenta continuidad temporal y una cantidad reducida de discontinuidades.

En la Figura 5.22 se presenta la señal registrada durante una semana utilizando el electrodo de plata. En este caso se observa una mayor cantidad de discontinuidades respecto de la medición realizada con el electrodo de acero. Este resultado no coincide con lo reportado en la bibliografía, donde la plata suele considerarse un material adecuado para este tipo de aplicaciones. Sin embargo, en la implementación realizada se identificó una limitación mecánica asociada a este material.

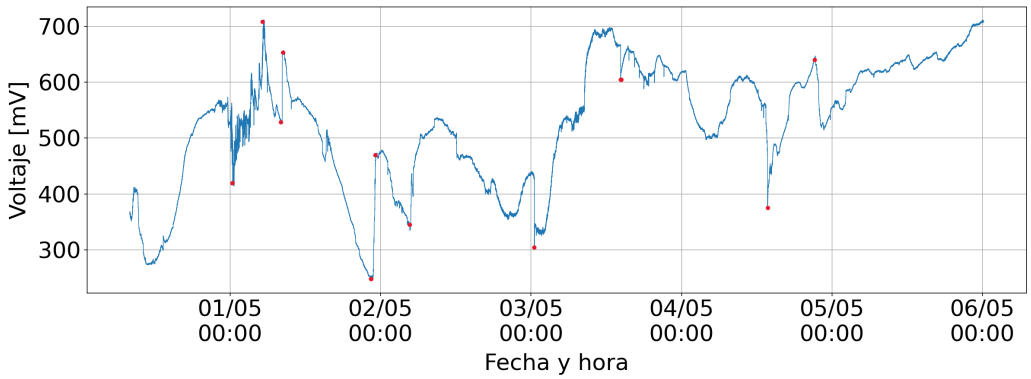


Figura 5.21: Señal registrada con electrodo de acero durante una semana. En rojo se indican algunos saltos observados en la señal.

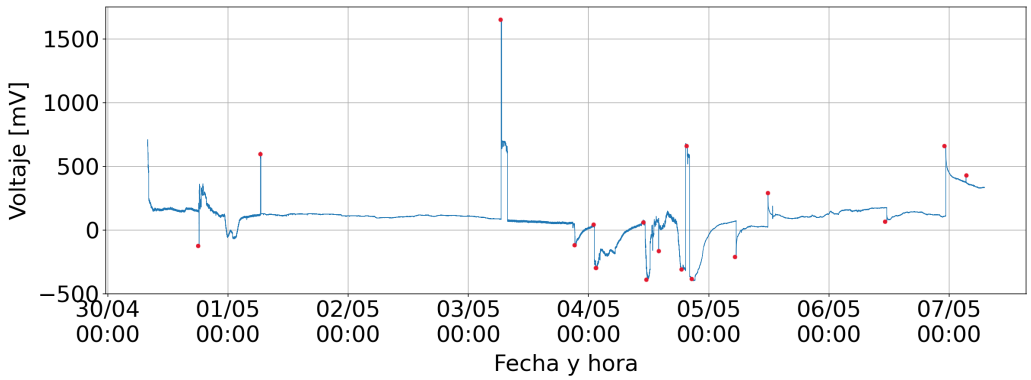


Figura 5.22: Señal registrada con electrodo de plata durante una semana. En rojo se indican algunos saltos observados en la señal.

La plata presenta menor rigidez que el acero, lo que dificulta su inserción en el tronco y reduce su estabilidad una vez colocada. En consecuencia, perturbaciones externas como el viento o la lluvia pueden producir desplazamientos del electrodo y afectar la continuidad del registro.

Aunque en ninguna de las dos mediciones se identificó un comportamiento claro de 24 horas, los resultados obtenidos permiten comparar el desempeño práctico de ambos electrodos en las condiciones de ensayo. A partir de esta prueba se resolvió que utilizar el electrodo de acero es la mejor opción, ya que proporcionó un registro con menor cantidad de discontinuidades y mayor estabilidad mecánica durante la medición. En estas condiciones, la estabilidad del electrodo resultó un criterio más relevante que la ventaja teórica asociada al material del conductor.

5.7. Análisis temporal por ciclos de 24 horas

En esta sección se analiza un registro de aproximadamente 14 días correspondiente a una planta de vid, adquirido mediante electrodos de aguja de acero. Con

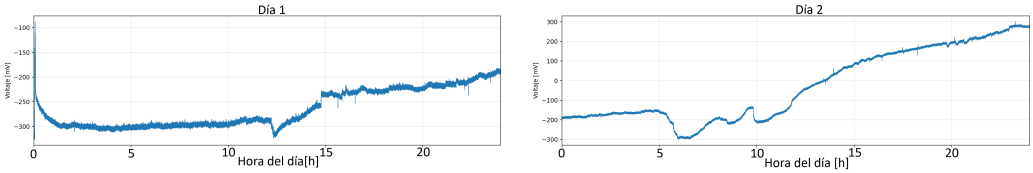


Figura 5.23: Ciclos diarios correspondientes a los días 1 y 2 del registro.

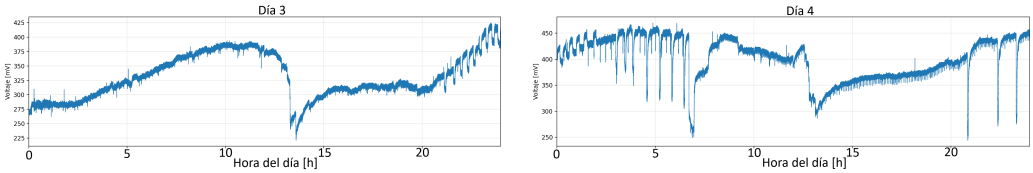


Figura 5.24: Ciclos diarios correspondientes a los días 3 y 4 del registro.

el objetivo de facilitar la observación de patrones temporales de escala diaria, la señal completa se dividió en segmentos de 24 horas, representados en las Figuras 5.23 a 5.29.

A partir de los segmentos diarios se identifica una componente de baja frecuencia con estructura aproximadamente circadiana. En varios ciclos, la señal presenta niveles superiores durante la noche y las primeras horas del día, especialmente entre 0 h y 5 h, así como hacia el final del período diario, entre 20 h y 24 h. En cambio, durante las horas centrales, aproximadamente entre 10 h y 15 h, se observa una reducción del nivel de señal.

La comparación entre días muestra que este patrón no es estrictamente periódico, pero mantiene una estructura temporal semejante entre ciclos. Sobre esta componente lenta aparecen perturbaciones locales, visibles como transitorios, picos o caídas abruptas. Por lo tanto, el registro no debe interpretarse como una oscilación senoidal ni como una señal periódica ideal, sino como una variación diaria de baja frecuencia con eventos superpuestos.

La superposición de los ciclos diarios permite visualizar esta tendencia de forma más clara, al evidenciar la recurrencia parcial del aumento nocturno y la disminución durante las horas centrales. La dispersión entre ciclos indica que la señal también está afectada por condiciones no estacionarias, posiblemente asociadas a variaciones ambientales, cambios en la interfaz electrodo-planta, etc.

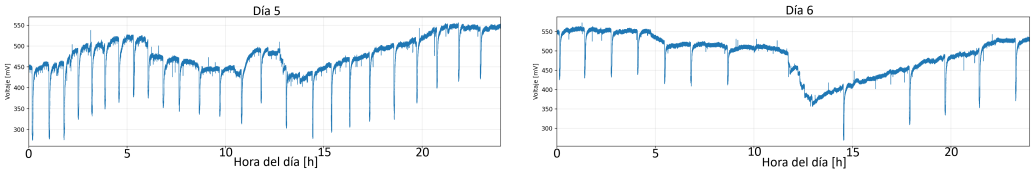


Figura 5.25: Ciclos diarios correspondientes a los días 5 y 6 del registro.

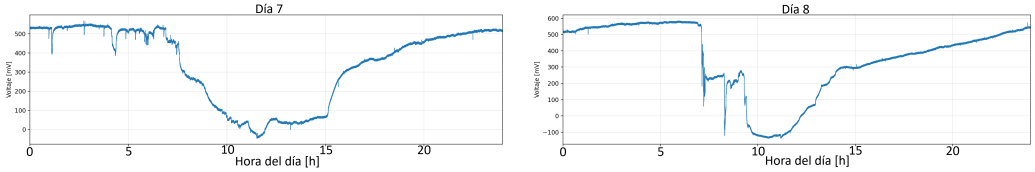


Figura 5.26: Ciclos diarios correspondientes a los días 7 y 8 del registro.

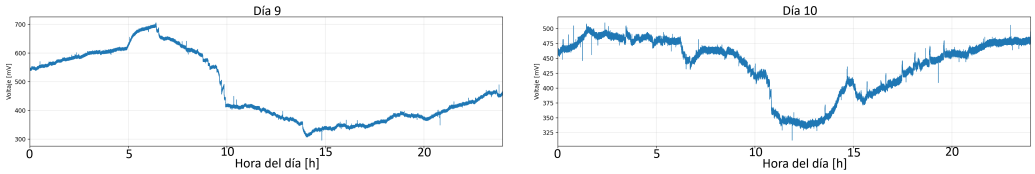


Figura 5.27: Ciclos diarios correspondientes a los días 9 y 10 del registro.

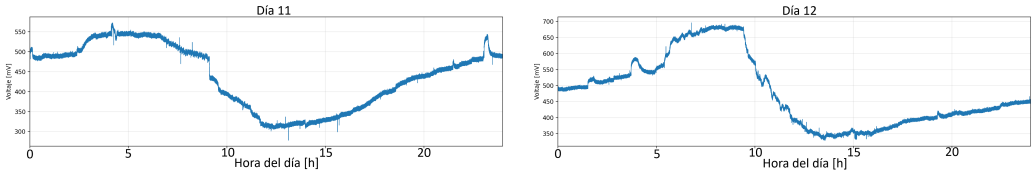


Figura 5.28: Ciclos diarios correspondientes a los días 11 y 12 del registro.

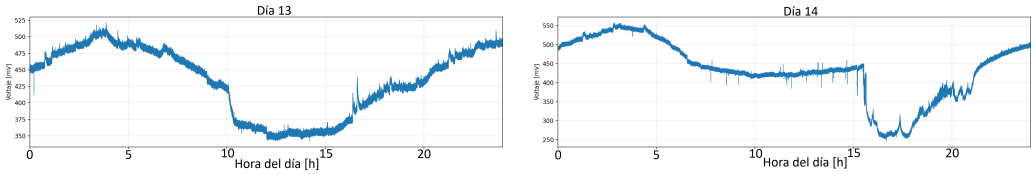


Figura 5.29: Ciclos diarios correspondientes a los días 13 y 14 del registro.

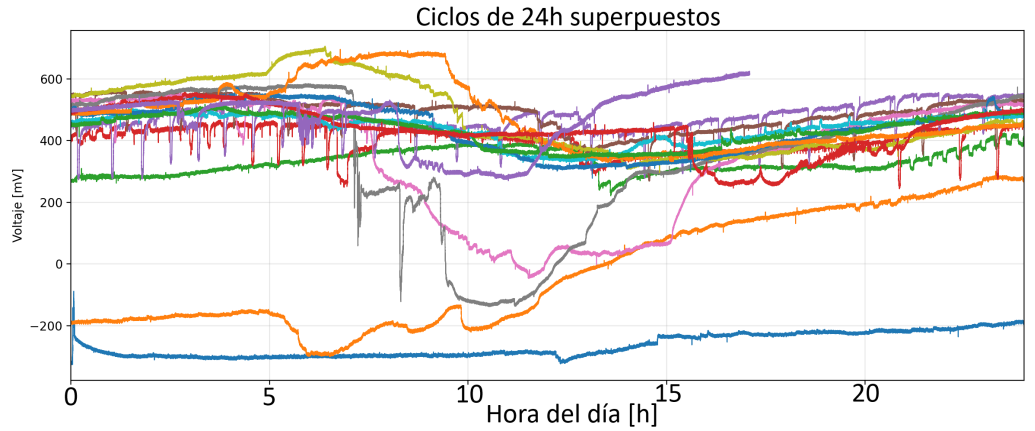


Figura 5.30: Superposición de los ciclos diarios de 24 horas correspondientes a los 14 días analizados.



Figura 5.31: Registro obtenido utilizando un electrodo comercial adaptado sobre la hoja y un electrodo de plata ubicado sobre el tronco. La señal fue suavizada mediante una media móvil de 500 muestras.

5.8. Electrodo de hoja

Inicialmente se había previsto el desarrollo de un electrodo específico para medición sobre hoja. Sin embargo, debido a restricciones de tiempo, no se llegó a fabricar una versión propia de este electrodo. En su lugar, se optó por adaptar un electrodo comercial, con el objetivo de realizar una prueba preliminar de adquisición en hoja.

Para este ensayo, uno de los electrodos se colocó directamente sobre la hoja de la planta, mientras que el segundo electrodo, de plata, se ubicó sobre el tronco. La señal registrada se presenta en la Figura 5.31. Para facilitar la visualización de la evolución temporal, la señal fue suavizada mediante una media móvil de 500 muestras.

El registro tuvo una duración aproximada de dos días y medio. A diferencia de los ensayos realizados con electrodos insertados en el tallo o tronco, en este caso la señal obtenida presentó variaciones poco apreciables y no permitió identificar con claridad eventos o patrones temporales relevantes.

Este resultado puede estar asociado a diversos factores, entre ellos la menor estabilidad mecánica del contacto sobre la hoja, la geometría del electrodo utilizado, la presión de contacto, la impedancia de la interfaz electrodo-planta. Además, al tratarse de una adaptación de un electrodo comercial y no de un diseño optimizado para este tipo de medición, las condiciones de contacto no fueron las más adecuadas para una evaluación concluyente.

Por este motivo, luego de discutir los resultados con el tutor, se decidió no profundizar en esta línea dentro del alcance del presente trabajo. El desarrollo de un electrodo específico para hoja, junto con un protocolo de fijación mecánica adecuado y ensayos comparativos sistemáticos, queda planteado como una línea de trabajo futuro.

Capítulo 6

Conclusiones

En este trabajo se desarrolló un prototipo portátil para la adquisición, almacenamiento, visualización y descarga de señales eléctricas en vegetales, orientado a su utilización en condiciones de campo. El sistema implementado integra una cadena circuital de acondicionamiento, una plataforma de adquisición basada en Arduino MKR Zero, un modo de operación *Live* para la visualización de la señal en vivo, un modo *Logger* para el registro autónomo sobre tarjeta SD, y una interfaz de usuario en Python para la interacción con el dispositivo y el análisis inicial de los datos. En este sentido, se considera que el objetivo general del proyecto fue alcanzado, ya que se obtuvo una herramienta funcional capaz de registrar señales vegetales fuera del laboratorio y ponerlas a disposición para su análisis posterior.

Uno de los principales aportes del trabajo fue la resolución integral del problema de instrumentación. El proyecto no se limitó al diseño de un circuito o de una interfaz aislada, sino que abarcó desde la selección del microcontrolador, la electrónica analógica y la alimentación portátil, hasta el diseño de electrodos, la implementación del firmware, la fabricación de la PCB, la integración mecánica del gabinete y el desarrollo de la interfaz de usuario. Esta integración permitió construir una plataforma coherente con las necesidades del contexto de uso, donde la portabilidad, la autonomía, la robustez y la facilidad de operación resultan tan importantes como la propia adquisición de la señal.

6.1. Evaluación de resultados

La evaluación del prototipo fue positiva, tanto desde el punto de vista electrónico como funcional. Las mediciones realizadas sobre la cadena analógica mostraron un comportamiento coherente con el diseño previsto. Las mediciones de respuesta en frecuencia también resultaron consistentes con la función esperada del sistema. En las distintas configuraciones ensayadas se observó una ganancia aproximadamente constante a bajas frecuencias, un rechazo marcado en torno a la frecuencia de red y una atenuación progresiva hacia frecuencias más elevadas. Además, no se identificaron diferencias relevantes dentro de la banda de interés al comparar el uso de batería, la ausencia de batería o el cambio entre electrodos de acero y

plata. Esto indica que la dinámica principal del sistema queda determinada por la cadena de filtrado diseñada y no por variaciones externas al circuito.La respuesta temporal ante una excitación conocida permitió verificar que el sistema conserva el carácter periódico de la señal de entrada sin recortes notorios ni deformaciones abruptas.

En cuanto al consumo, las pruebas realizadas en modo *Logger* permitieron comprobar que el microcontrolador entra efectivamente en bajo consumo entre interrupciones. Se observaron picos asociados al temporizador, al encendido periódico del LED y a las escrituras en la tarjeta SD, siendo estas últimas los eventos de mayor demanda de corriente. Aun así, el comportamiento medido fue compatible con una operación autónoma prolongada. La prueba de descarga completa de batería permitió estimar una autonomía de aproximadamente 20 días, 8 horas y 53 minutos, lo que valida el uso del equipo para campañas de adquisición de varios días. Sumado a que el esquema de almacenamiento en archivos CSV también resultó adecuado para el objetivo del prototipo. Si bien este formato ocupa más espacio que una codificación binaria, permite que los datos sean legibles, simples de procesar y directamente utilizables desde herramientas externas.

Las pruebas funcionales sobre plantas permitieron validar el comportamiento general del sistema en condiciones reales. Los primeros ensayos en *Mimosa pudica* fueron útiles como verificación preliminar del prototipo y permitieron observar variaciones transitorias durante estímulos mecánicos. Posteriormente, las mediciones sobre vid permitieron obtener registros continuos durante varios días, confirmando la capacidad del sistema para adquirir, almacenar y recuperar señales vegetales en campo.

La comparación entre electrodos de acero y plata permitió extraer una conclusión práctica relevante. Si bien la plata aparece en la bibliografía como un material adecuado para este tipo de mediciones, en las condiciones de uso del prototipo presentó mayores dificultades mecánicas. Su menor rigidez favorece deformaciones y movimientos durante la colocación, lo que puede aumentar la cantidad de discontinuidades en el registro. En cambio, los electrodos de acero mostraron mayor estabilidad mecánica, menor costo y menor complejidad de preparación, por lo que resultaron más adecuados para esta versión del sistema.

Finalmente, el análisis de un registro de aproximadamente 14 días en vid permitió observar una tendencia de baja frecuencia compatible con una modulación diaria. Al dividir la señal en ciclos de 24 horas y superponerlos, se identificaron similitudes entre varios días, con niveles relativamente más altos durante la noche y primeras horas del día, y disminuciones hacia las horas centrales. Si bien estos resultados no permiten establecer una interpretación fisiológica definitiva, sí muestran que el sistema es capaz de registrar variaciones lentas y continuas durante períodos prolongados.

En conjunto, las pruebas realizadas permiten concluir que el prototipo cumple con su función instrumental principal: adquirir señales eléctricas vegetales, almacenarlas de forma autónoma, transmitirlas o descargarlas mediante USB y facilitar su visualización posterior. El sistema no constituye todavía una herramienta de diagnóstico fisiológico, pero sí una plataforma funcional y reproducible para con-

tinuar realizando campañas experimentales en electrofisiología vegetal.

6.2. Reflexiones

Uno de los principales aprendizajes del trabajo fue comprender que la electrofisiología vegetal no depende únicamente de amplificar señales lentas. La dificultad principal está en construir un sistema completo que mantenga condiciones de medición suficientemente estables en un entorno real. En este sentido, decisiones aparentemente secundarias, como la forma de fijación del electrodo, el material utilizado, el sellado de la envolvente, el manejo del almacenamiento, la referencia de continua, la autonomía energética o la interfaz de usuario, tienen un impacto directo sobre la calidad del resultado final.

El proyecto permitió recorrer un proceso completo de ingeniería, desde la formulación inicial del problema hasta la obtención de un prototipo funcional. Esto implicó relevar bibliografía, definir requerimientos, tomar decisiones de diseño, construir versiones sucesivas del circuito, detectar errores, corregir limitaciones, fabricar varias PCBs, programar el firmware, desarrollar una interfaz de usuario, integrar el sistema en una envolvente y realizar pruebas experimentales. El resultado final no fue únicamente un circuito electrónico, sino un equipo utilizable por una persona del área agronómica que no necesariamente posee conocimientos de electrónica.

Este aspecto resulta especialmente importante, ya que uno de los propósitos del proyecto era acercar la adquisición de señales eléctricas vegetales a usuarios vinculados al estudio de plantas, no solo a usuarios técnicos. La interfaz desarrollada, los modos de operación y el procedimiento de descarga fueron pensados para reducir la complejidad operativa del sistema. De esta forma, el prototipo permite que el usuario se concentre en la planta, en el ensayo y en la interpretación de los datos, sin necesidad de intervenir sobre detalles internos de adquisición, comunicación o almacenamiento.

También se destaca el aprendizaje asociado a la gestión de un proyecto de desarrollo tecnológico. A lo largo del trabajo fue necesario ajustar decisiones, rediseñar partes del sistema, abandonar soluciones que no resultaron adecuadas y priorizar criterios prácticos frente a alternativas teóricamente atractivas. La comparación entre electrodos, las iteraciones del circuito, la elección del formato de almacenamiento y la integración mecánica del prototipo son ejemplos de decisiones que se fueron consolidando a partir de la experiencia experimental y no únicamente desde el diseño inicial.

Otro aspecto relevante fue la familiarización con un área poco conocida al comienzo del proyecto. La electrofisiología vegetal combina elementos de electrónica analógica, adquisición de datos, fisiología vegetal, procesamiento de señales y experimentación en campo. Esta combinación obligó a trabajar con incertidumbre, ya que no siempre existe una separación clara entre efectos propios de la planta, ruido eléctrico, condiciones ambientales y artefactos mecánicos. En consecuencia, el trabajo permitió adquirir una visión más amplia del problema y reconocer la

importancia de diseñar instrumentos confiables antes de extraer conclusiones fisiológicas fuertes.

En términos generales, la tesis permitió transformar una necesidad experimental en una plataforma concreta de medición. El sistema desarrollado no resuelve por completo el problema de la interpretación de las señales eléctricas vegetales, pero sí proporciona una herramienta funcional para obtener registros, analizarlos y diseñar nuevas campañas de experimentación. Ese aporte instrumental constituye la base necesaria para trabajos posteriores de mayor profundidad agronómica y fisiológica.

6.3. Trabajo a futuro

A partir de los resultados obtenidos, una primera línea de trabajo futuro consiste en ampliar la experimentación con el sistema. Sería conveniente realizar campañas de medición más extensas, con mayor cantidad de plantas, repeticiones controladas y registro simultáneo de variables ambientales como temperatura, humedad, radiación, riego, viento y estado hídrico. Esto permitiría contextualizar mejor las señales adquiridas y avanzar hacia una interpretación más sólida de los patrones observados.

También sería relevante realizar ensayos comparativos más sistemáticos entre tipos de electrodos, posiciones de colocación y especies vegetales. Si bien en este trabajo se observó que el electrodo de aguja de acero resultó más conveniente que el de plata para el prototipo desarrollado, todavía queda margen para estudiar otras geometrías, materiales y mecanismos de fijación. En particular, sería deseable reducir aún más los artefactos asociados al movimiento, mejorar la repetibilidad entre ensayos y minimizar el daño mecánico sobre la zona de contacto.

Asimismo, sería conveniente profundizar en el estudio de electrodos aplicados directamente sobre la hoja. Si bien se realizaron algunas pruebas preliminares con este tipo de contacto, no fue posible obtener señales relevantes o suficientemente estables durante los ensayos realizados. Por lo tanto, esta alternativa queda abierta para trabajos futuros, especialmente en lo referido al diseño mecánico del electrodo y la presión de contacto.

Desde el punto de vista electrónico, una mejora importante sería rediseñar la entrada del sistema para aprovechar completamente la capacidad diferencial del amplificador de instrumentación. En el prototipo actual, la arquitectura de conexión no explota todo el potencial de la medición diferencial. Una posible mejora sería implementar un esquema de tres electrodos, donde uno funcione como referencia común entre la planta y el sistema de adquisición, mientras que los otros dos se conecten a las entradas diferenciales del amplificador. Esta modificación permitiría medir diferencias de potencial entre dos puntos de la planta con mayor rechazo de ruido común, siempre que se acompañe de un diseño adecuado de protección, referencia y estabilidad de contacto.

Otra línea de mejora corresponde a la arquitectura general del sistema. El prototipo desarrollado está pensado como un equipo individual de adquisición, pero en aplicaciones agronómicas puede resultar útil medir varias plantas en simultáneo y

en distintos puntos del cultivo. En ese contexto, podría evaluarse una arquitectura distribuida basada en nodos sensores de bajo costo comunicados mediante LoRa hacia un concentrador central. Este enfoque permitiría aumentar la cantidad de plantas monitoreadas y reducir el cableado necesario en campo. Sin embargo, también requeriría un desarrollo adicional en comunicación inalámbrica, sincronización temporal, gestión energética, almacenamiento distribuido y robustez frente a pérdidas de enlace.

La interfaz de usuario también puede ampliarse en futuras versiones. Entre las mejoras posibles se encuentran la incorporación de herramientas automáticas de análisis temporal y espectral, la generación de reportes, la anotación de eventos experimentales, la comparación entre registros y la asociación directa entre señales vegetales y variables ambientales. Estas funcionalidades permitirían que el sistema no solo sirva para adquirir datos, sino también para facilitar etapas posteriores de interpretación.

Finalmente, sería deseable avanzar hacia una versión más compacta, robusta y optimizada del prototipo. Esto podría incluir una reducción del tamaño del PCB, mejoras en el consumo energético, optimización del sistema de carga, incorporación de un panel solar para extender la autonomía en campo, mayor protección frente a condiciones ambientales, conectores más adecuados para uso prolongado y procedimientos de calibración más definidos. Estas mejoras permitirían acercar el desarrollo a una herramienta más madura para campañas experimentales de larga duración.

Esta página ha sido intencionalmente dejada en blanco.

Apéndice A

Manual de Uso

ElektroGrün - Manual de uso

v. 2026-4

Rodrigo Fernandez, Fabrizio Raffaelli,
Gastón Rodriguez

18 de mayo de 2026



Resumen

ElektroGrün es un dispositivo pensado para medir y registrar cómo cambia el voltaje de una planta a lo largo del tiempo, de forma automática y en condiciones reales (como en una maceta o en el campo). Se coloca en la planta mediante electrodos y guarda las mediciones para que después puedas ver gráficas y analizar si los cambios se relacionan con cosas como luz, temperatura, humedad o riego. Es una herramienta para observar el comportamiento eléctrico de la planta sin necesidad de equipos de laboratorio.

Índice

1. Descripción general **3**

2. Colocación de electrodos **5**
2.1. Separación entre electrodos 5
2.2. Presión y sujeción 5
2.3. Evitar artefactos por movimiento 5
2.4. Errores comunes 5
2.5. Colocación de electrodo tipo “aguja-aguja” 6

3. Preparación de planta-electrodo-dispositivo **7**

4. Uso general de la interfaz **7**
4.1. Conectarse al dispositivo mediante PC 7
4.2. Interacciones online 9
4.2.1. Descargar datos guardados 9
4.2.2. Gráfica en Vivo 12
4.2.3. Borrar memoria 15
4.3. Interacciones offline 17
4.3.1. Gráfica *Logger* 17

5. Batería **20**

6. Tarjeta microSD **21**

1. Descripción general

La solución ElektroGrün dispone de dos modos de funcionamiento para la adquisición de señales bioeléctricas: *Live* y *Logger*. En el modo *Live*, el dispositivo transmite las mediciones a una PC para su visualización en vivo. En el modo *Logger*, registra las señales adquiridas en memoria interna para su análisis posterior.

El sistema permite adquirir y almacenar el **voltaje eléctrico** medido en un vegetal a lo largo del tiempo, con el objetivo de proporcionar datos útiles para análisis agronómicos en condiciones de campo.

Qué entrega: series temporales en formato **csv**, con tiempo (ms) y voltaje (mV).

Qué no entrega: el equipo no realiza diagnósticos automáticos sobre el estado de la planta; los datos requieren un análisis posterior.

Cuenta con 4 conexiones al mundo exterior como muestra la Figura 1 y Cuadro 1:

Número	Concepto	Descripción
1	Electrodos	Conexión de los electrodos que van en la planta.
2	Botón de modo	Botón para cambiar de modo.
3	USB	Carga de bateria y comunicación con PC.
4	Botón ON / OFF	Botón para prender y apagar el dispositivo

Cuadro 1: Conexiónes del dispositivo

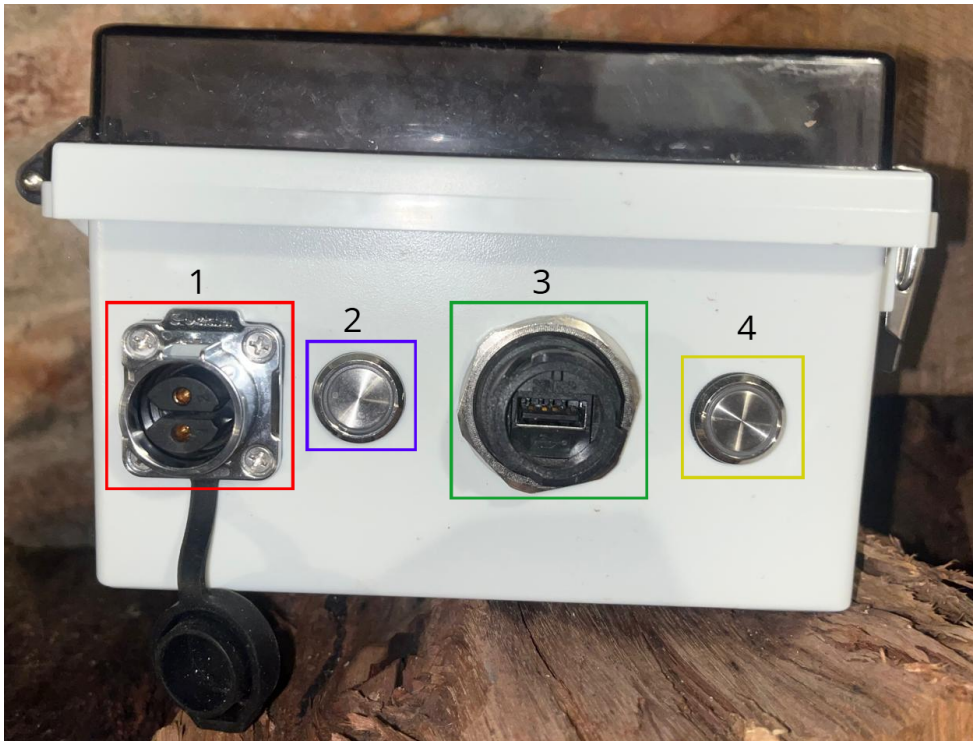


Figura 1: Conexiones del dispositivo.

A su vez, desde la tapa transparente de la caja se pueden ver los componentes. Entre ellos

algunos leds que sirven de indicadores, estos se describen en el Cuadro 2 y se visualizan en la Figura 2.

Número	Concepto	Descripción
1	Encendido/Bateria	Led rojo: ▪ Si la batería está en buen estado, el LED parpadea durante 1 s y luego permanece apagado 4 s. ▪ Si la batería requiere carga, el LED realiza la secuencia: 0,5 s parpadea, 0,5 s apagado, 0,5 s parpadea y luego permanece apagado 3,5 s.
2	Modo	Led naranja que enciende cuando el dispositivo se encuentra en modo *Live*.
3	Encendido	Led verde que enciende siempre que este encendido, cuando se alimente desde la bateria será muy tenue. Al alimentar desde la PC será más brillante
4	Carga	Led rojo, que se enciende mientras se esta cargando la bateria.

Cuadro 2: Leds indicadores

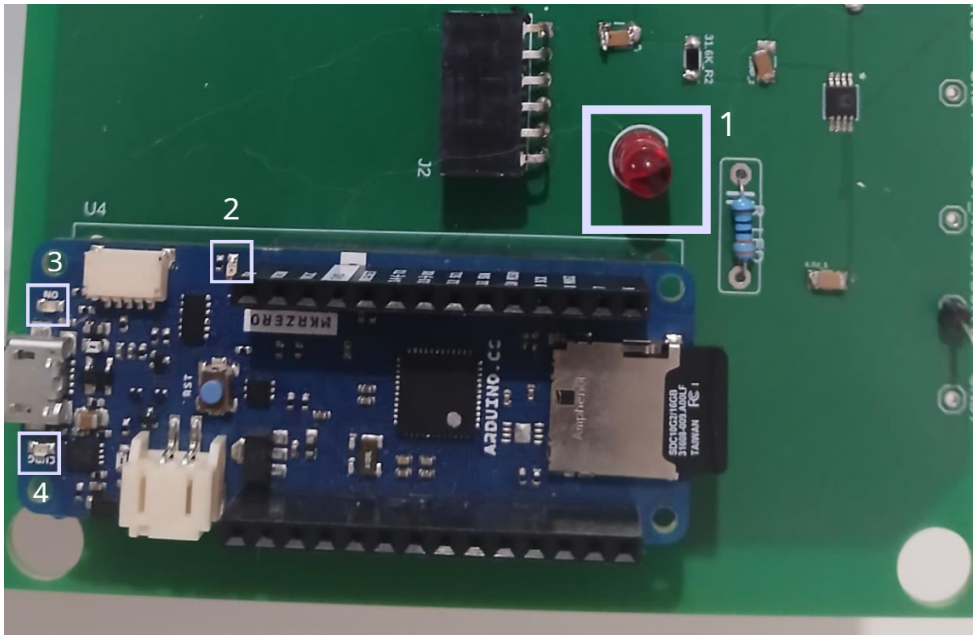


Figura 2: Leds indicadores.

2. Colocación de electrodos

Una correcta colocación de los electrodos es clave para obtener una medición estable y con bajo nivel de ruido. Las siguientes recomendaciones buscan mejorar el contacto eléctrico y reducir artefactos por movimiento.

- **Cable negro:** electrodo de **referencia**.
- **Cable rojo:** electrodo de **medición**.

2.1. Separación entre electrodos

- Evite colocarlos **demasiado cerca** (menos de 2 cm), ya que disminuye la diferencia de potencial medida y la señal puede volverse muy pequeña.
- Evite colocarlos **excesivamente lejos** (más de 2 metros), eso obliga a tender cables con tensión y/o aumentar movimientos mecánicos.

2.2. Presión y sujeción

- Antes de colocar, asegure que los electrodos estén **limpios y secos**.
- El contacto debe ser **firme** pero sin dañar el tejido: no estrangular, no perforar en exceso ni aplastar la hoja.
- Una sujeción excesiva puede generar estrés mecánico o alterar la fisiología local, afectando la medición.
- Asegure que el punto de contacto no se desplace con el viento. Si es necesario, fije suavemente el cable para que **no traccione** del electrodo.

2.3. Evitar artefactos por movimiento

- Evite que los cables queden colgando o tirantes: sujételos de forma que el peso del cable no actúe sobre los electrodos.
- Durante la adquisición, evite tocar la planta o los cables.

2.4. Errores comunes

- No coloque ambos electrodos en el **mismo punto** o excesivamente cerca: se reduce la señal diferencial.
- No deje el cable con tensión: si el cable tira del electrodo, es probable que se despegue.

2.5. Colocación de electrodo tipo “aguja-aguja”



Figura 3: Ejemplo de colocación de electrodos.

En el ejemplo que se muestra en la Figura 3, se ejemplifica la colocación de dos electrodos tipo “aguja”. Estos van levemente clavados en el tronco de forma perpendicular al mismo.

A continuación se describe el procedimiento para clavarlos en la planta:

1. Cada electrodo cuenta con dos tornillos, uno con punta y otro sin punta. Ajustar ambos tornillos hasta que el espacio entre ellos quede un poco más grande que el tronco.

2. Apoyar suavemente, sin girar el tornillo, la punta de plata en el tronco o rama de la planta.

3. Girar el tornillo sin punta hasta que el electrodo quede firmemente agarrado a la planta y la punta insertada total o parcialmente en el tronco.

3. Preparación de planta-electrodo-dispositivo

- Con el equipo apagado, conecte los electrodos a la planta, siguiendo las recomendaciones de colocación indicadas anteriormente.
- El electrodo asociado al cable negro se utiliza como **referencia**, mientras que el electrodo asociado al cable rojo corresponde a la **señal a medir**.
- Verifique que los electrodos queden firmemente sujetos en la planta y que los conductores estén correctamente ajustados en la bornera.
- Si va a registrar en modo *Logger*, verifique que se encuentra con una tarjeta SD y en caso contrario, debe colocar una de hasta 32 GB en formateada en FAT32.
- Encienda el equipo mediante el botón de encendido.
- Verifique que el Led 1 definido en el Cuadro 2 funcione de manera adecuada.
- Presione el botón de modo para seleccionar: **Logger** o **Live**.
- Para detener la medición, se recomienda apagar primero el equipo y luego desconectar los electrodos.

4. Uso general de la interfaz

4.1. Conectarse al dispositivo mediante PC

Para conectarse al dispositivo, se debe tener conectado el mismo a la PC mediante un cable USB.

Se debe seleccionar el puerto que corresponde al dispositivo, como se muestra en la Figura 4 y presionar el botón de ”Conectar“.

El nombre del puerto a seleccionar depende enteramente de la PC, por lo que el usuario deberá identificarlo. En este caso se llama ”/dev/ttyUSB0 – FT232R USB UART“ ya que se está usando una distribución Linux.

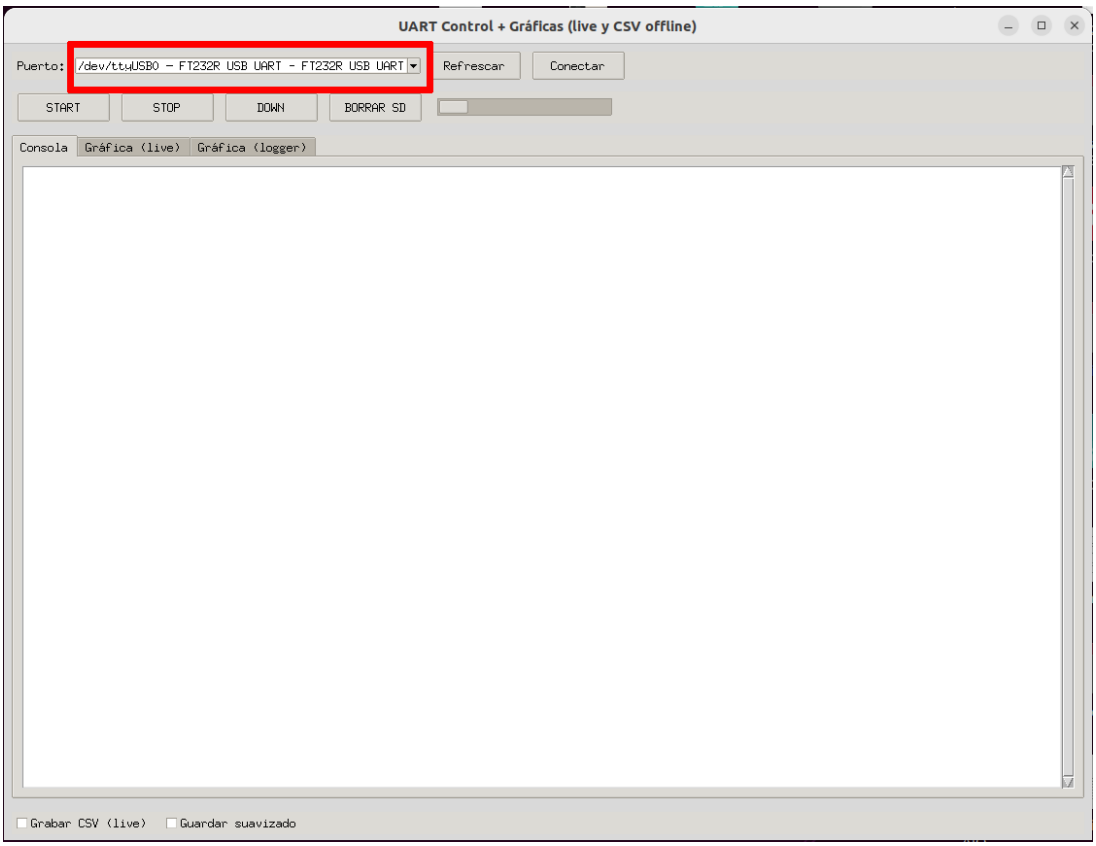


Figura 4: Selección de puertos.

Una vez conectado el dispositivo manda el mensaje de exito que se muestra en la Figura 5.

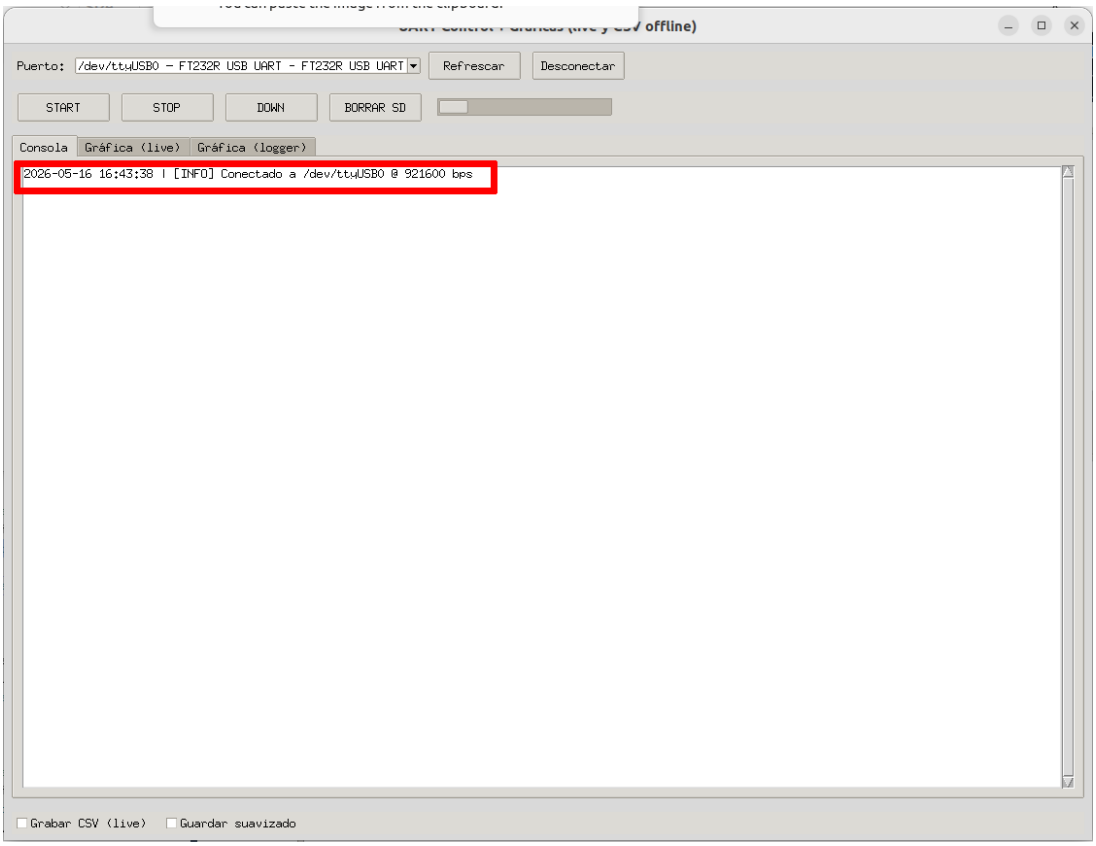


Figura 5: Dispositivo Conectado.

4.2. Interacciones online

Para las siguientes subsecciones se debe tener en cuenta que el dispositivo debe estar correctamente conectado a la PC como se indica en la sección 4.1. Y en modo *Live*, es decir con el botón de modo presionado.

4.2.1. Descargar datos guardados

Para descargar los datos guardados en la memoria del dispositivo se debe presionar el botón de ”DOWN“ de la interfaz y este desplegará un menú para elegir la carpeta en el filesystem (Figura 6) donde se desee guardar los datos obtenidos en formato csv.

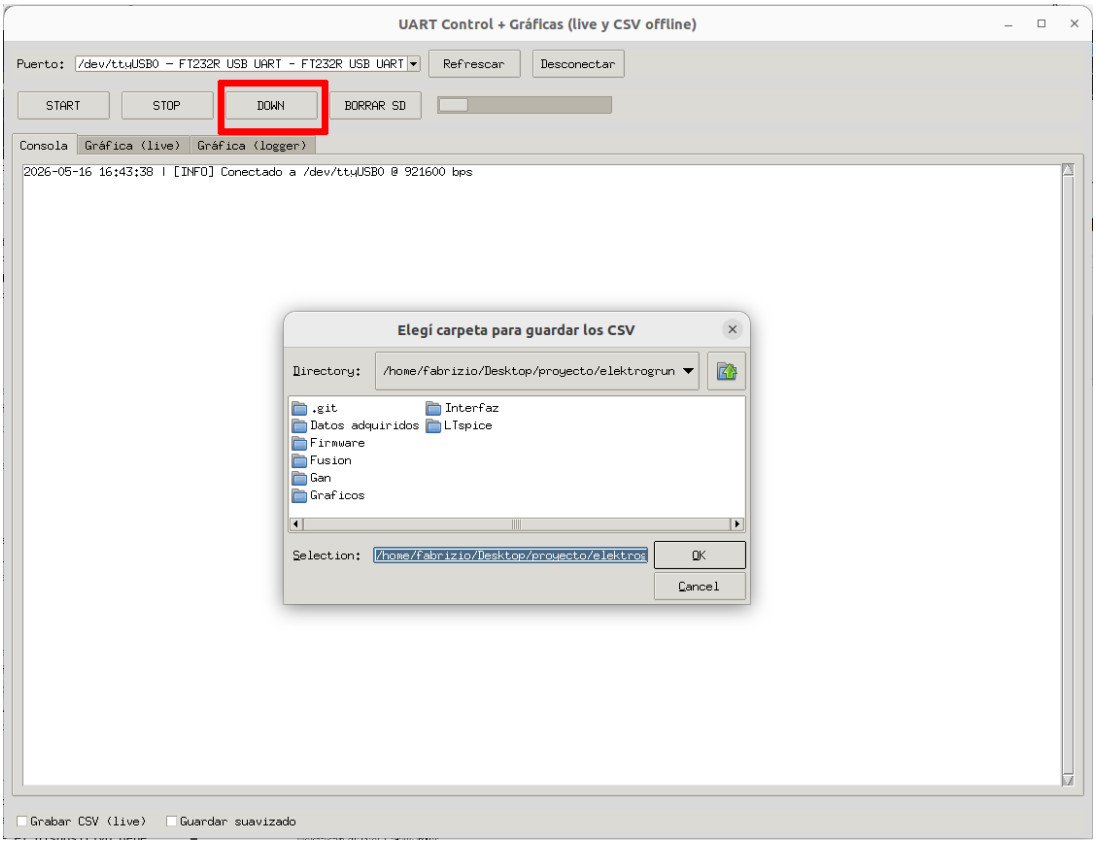


Figura 6: Ubicación de los archivos

Durante la descarga se desplegará el cuadro de descarga que se muestra en la Figura 7.

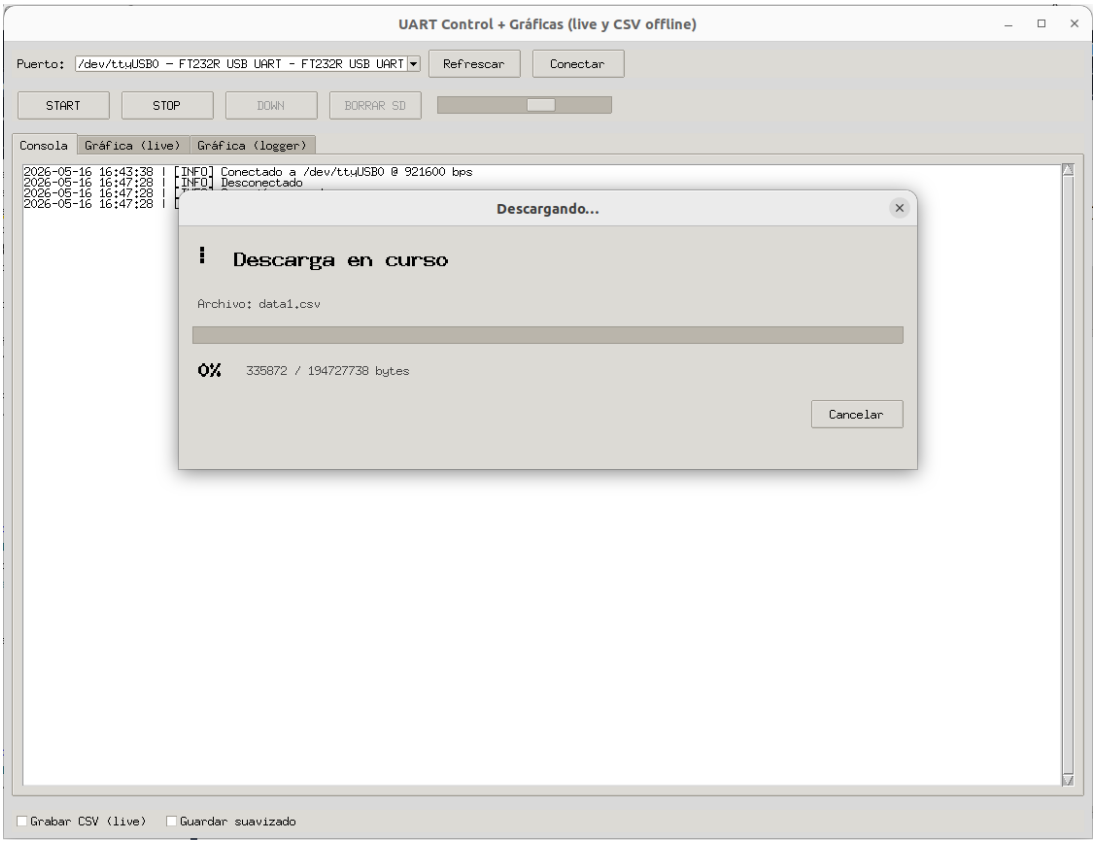


Figura 7: Descarga en Proceso

Cuando se terminan de descargar los archivos se manda el mensaje de exito que se muestra en la figura 8.

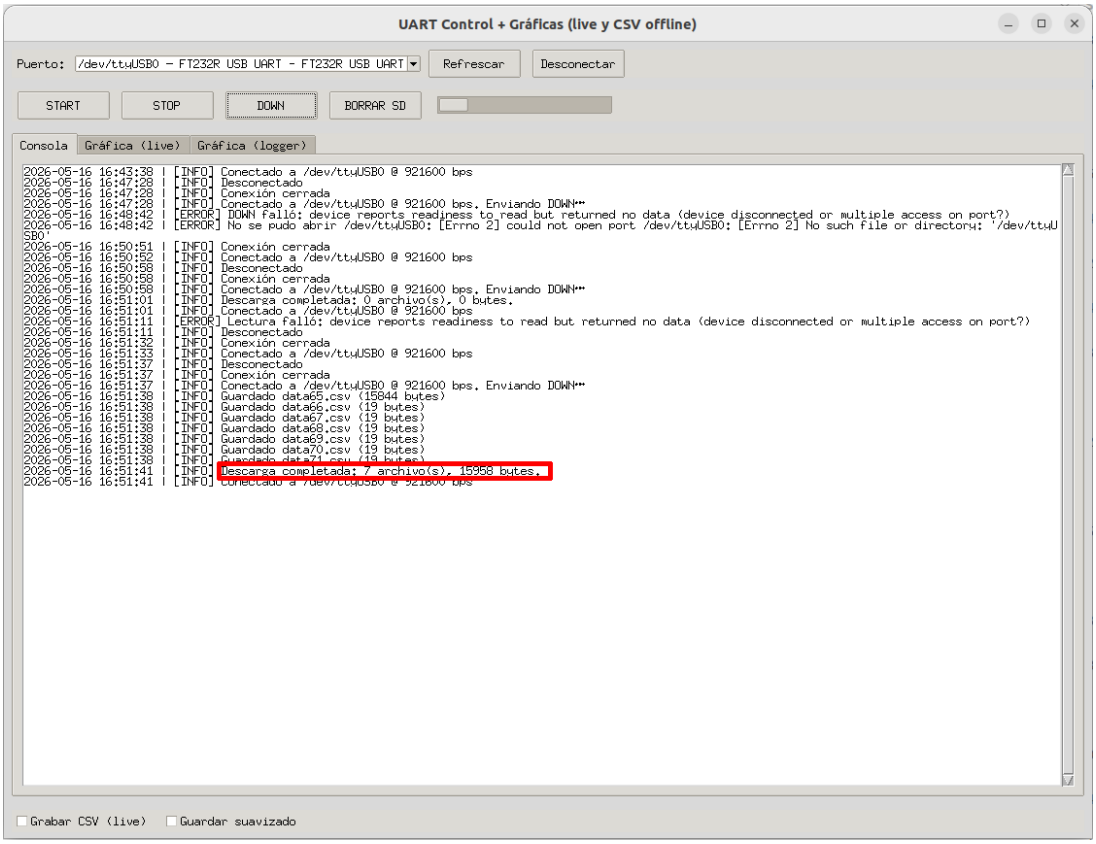


Figura 8: Mensaje de éxito al descargar

4.2.2. Gráfica en Vivo

Para graficar en vivo primero se debe asegurar que los electrodos estén correctamente colocados y el USB conectado.

Para iniciar la gráfica se debe presionar el botón de ”START“ como muestra la Figura 9.

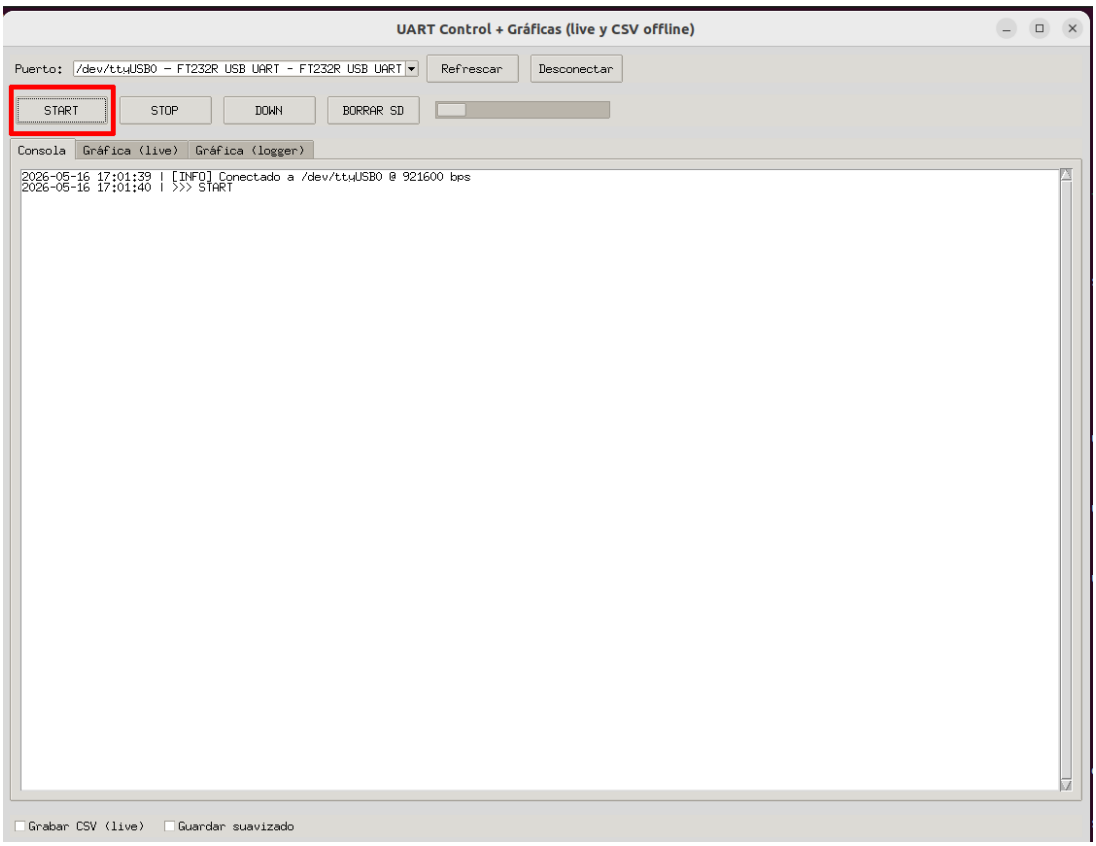


Figura 9: Inicio de transmision de datos Live.

Para visualizar las mediciones en vivo se debe cambiar a la pestaña ”Gráfica (tiempo real)“, esta gráfica se actualizará periodicamente (cada 30s aproximadamente) con las mediciones más recientes (Figura 10).

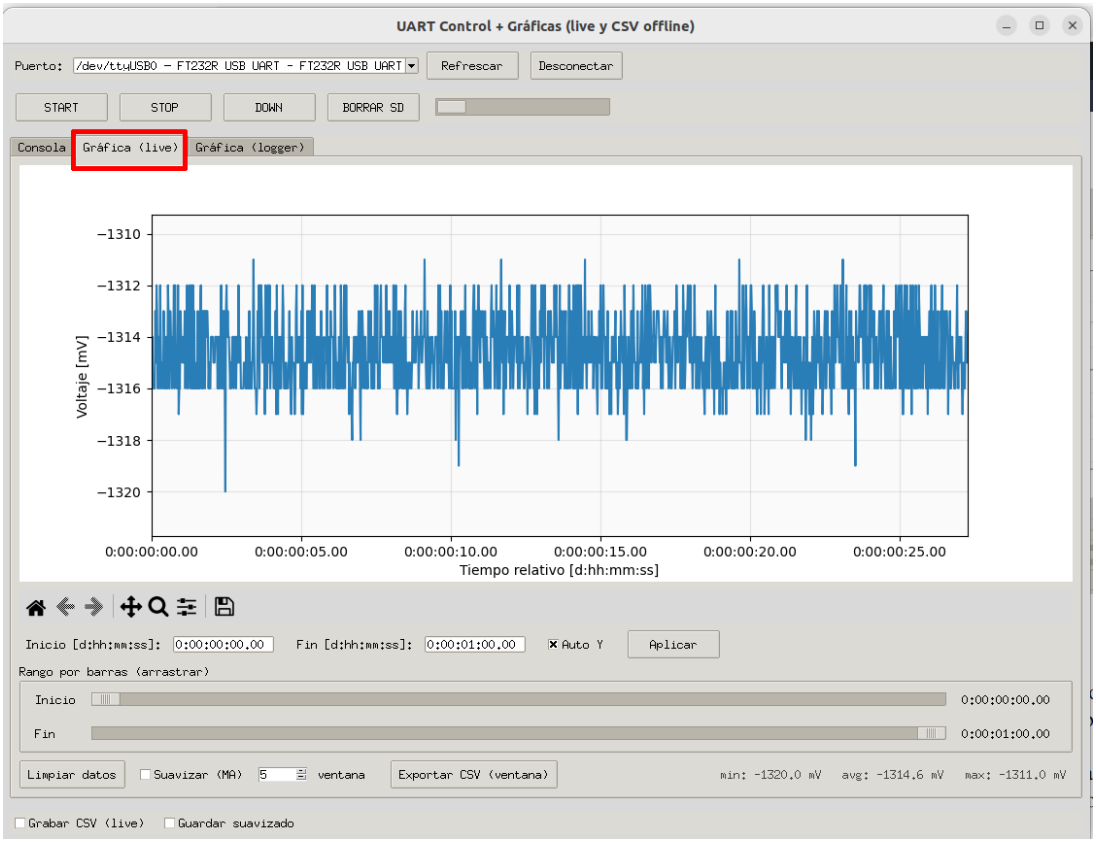


Figura 10: Inicio de transmisión de datos Live.

Las mayor parte de las funciones de esta ventana están descriptas en la sección 4.3.1. La única diferencia reside en el botón ”Limpiar datos“ el cual reinicia la captura. Borra las mediciones que se hicieron hasta el momento.

Una vez que se termina con la captura se debe detener la misma, para ello se debe dirigir a la pestaña de ”Consola“ y presionar el botón de ”STOP“ (Figura 11).

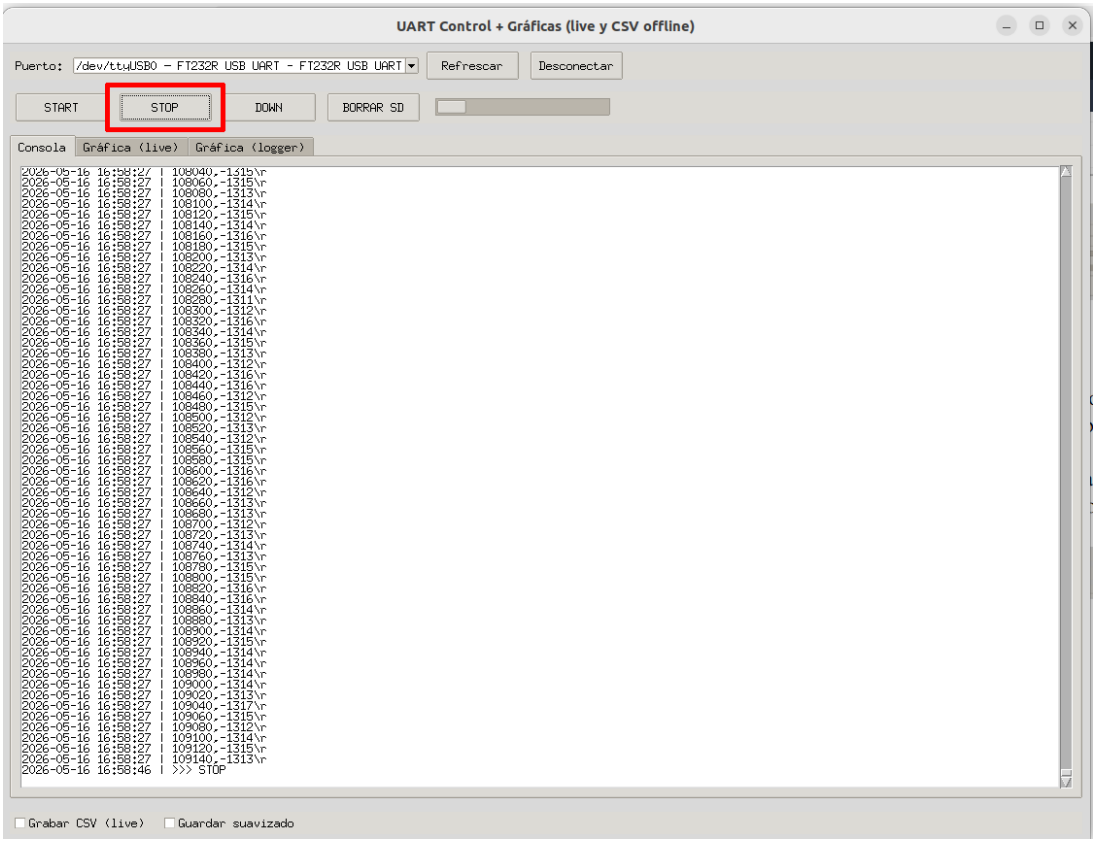


Figura 11: Detener Captura.

4.2.3. Borrar memoria

Para borrar la memoria SD del dispositivo, osea todos los datos capturados en modo *Logger*, se debe presionar el botón de *Borrar SD* como aparece en la Figura 12.

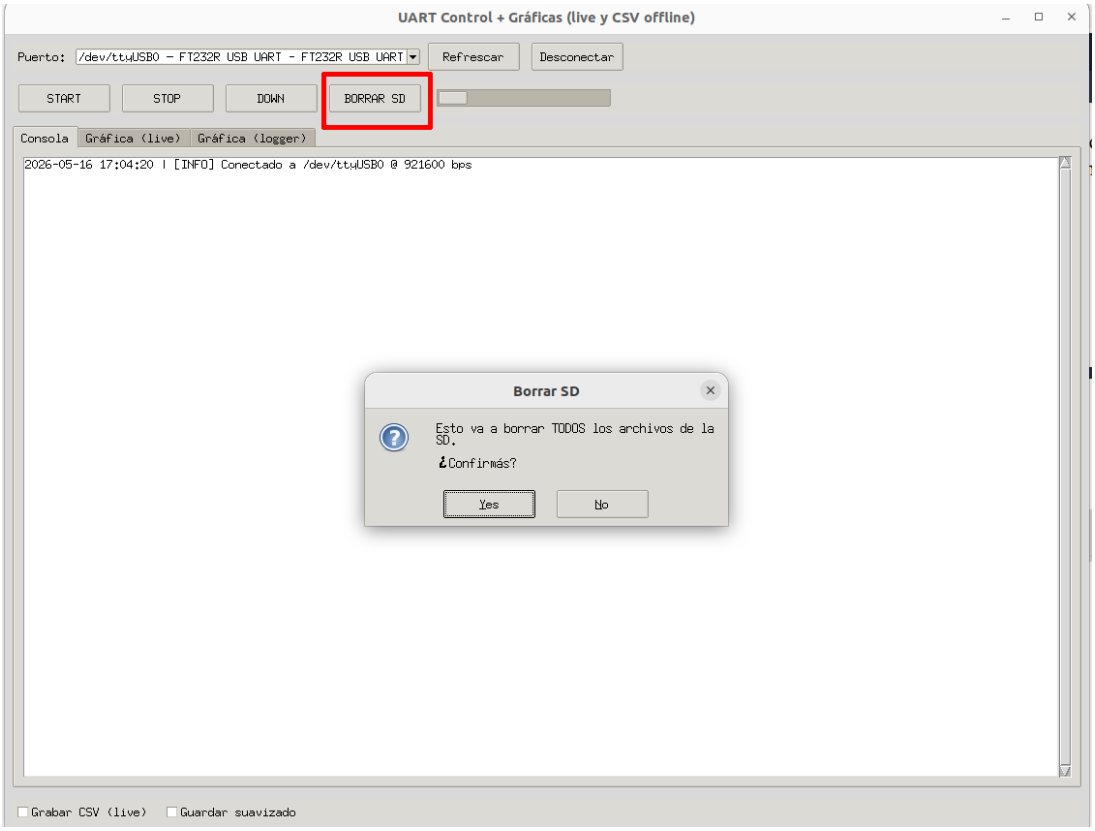


Figura 12: Borrar SD.

Esta accion procedera a ir borrando todos los archivos de la SD. Una vez que termine se entregará el mensaje de éxito mostrado en la Figura 13.

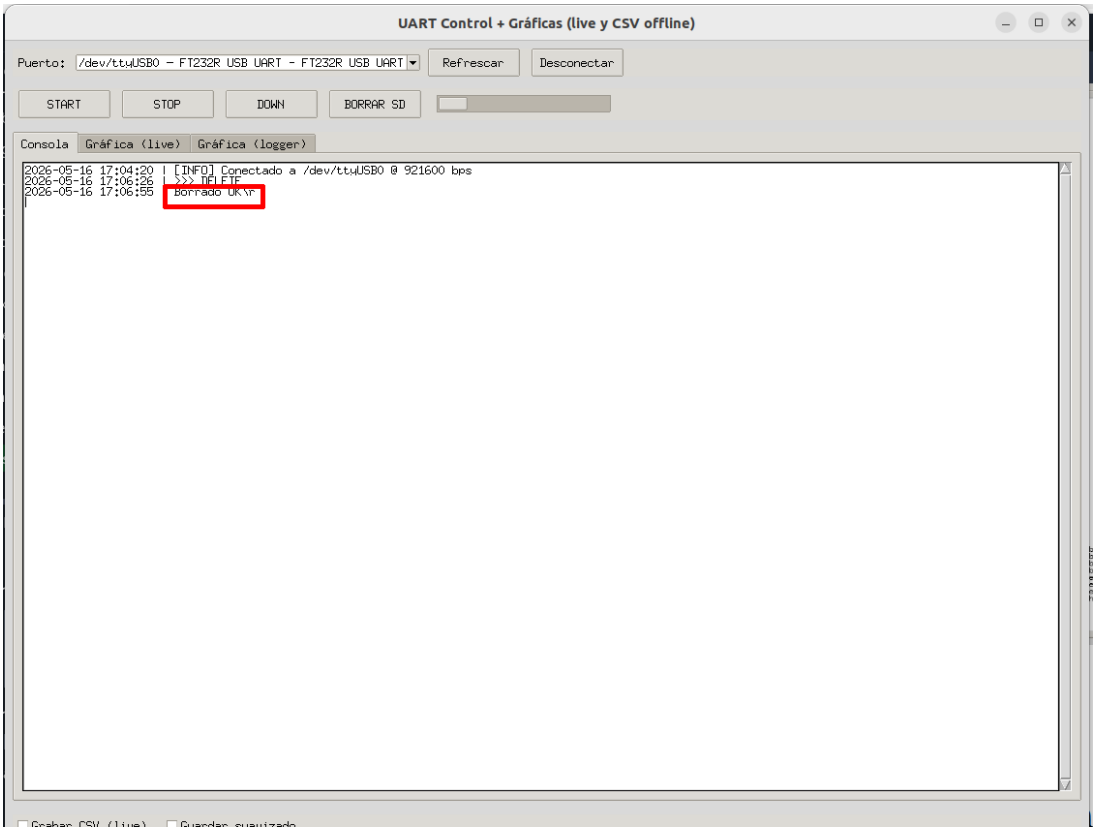


Figura 13: Borrado OK.

4.3. Interacciones offline

Para las siguientes subsecciones no es necesario tener conectado el dispositivo.

4.3.1. Gráfica *Logger*

Abrir la pestaña bajo el nombre “Gráfica (logger)” y presionar el botón que dice “Abrir CSV” como se muestra en la figura 14.

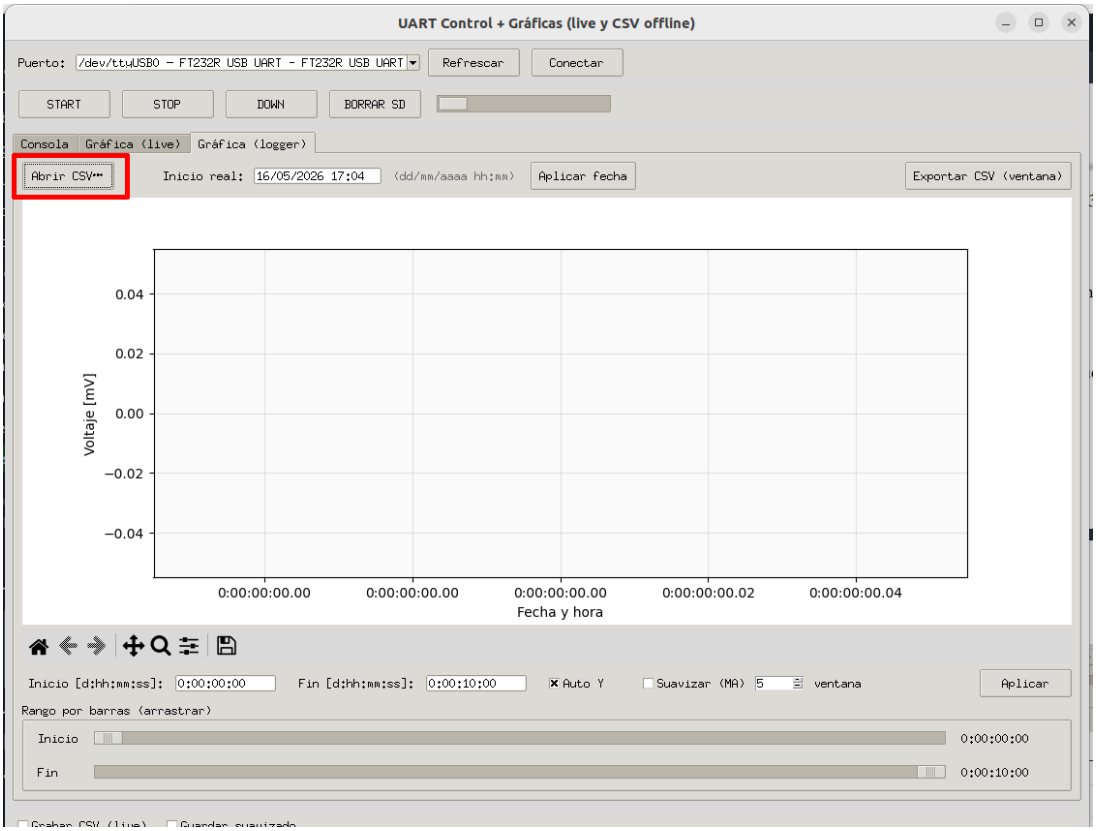


Figura 14: Abrir `.csv` desde la interfaz

Seleccionar un archivo fuente en formato `.csv` para desplegar en la interfaz. El archivo `.csv` debe tener el siguiente formato, donde el tiempo está en milisegundos y el voltaje en milivolts:

```
tiempo,voltaje_mV
19,-549
39,-551
59,-551
79,-549
```

El dispositivo solo guarda archivos `.csv` con este formato, por lo que cualquier archivo descargado desde el dispositivo se puede abrir desde la interfaz.

Una vez abierto el archivo, se mostrará como indica la Figura 15.

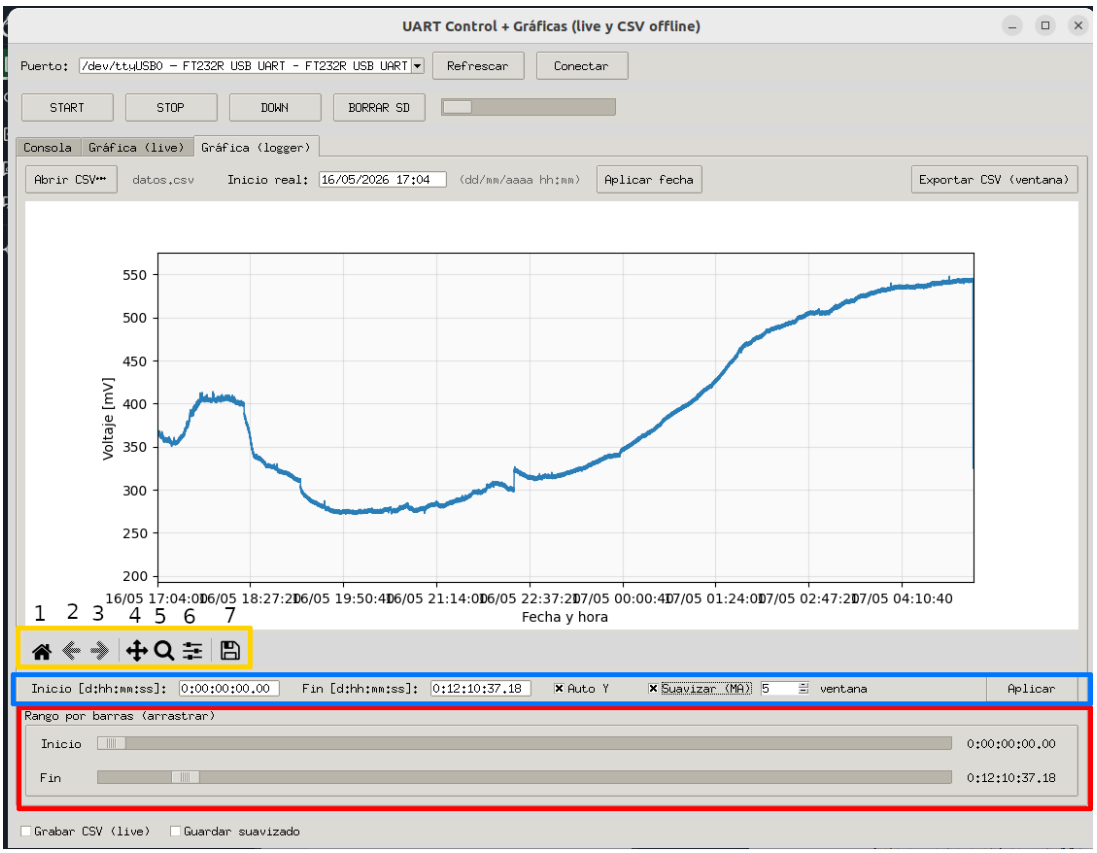


Figura 15: Editar gráfica logger

El recuadro de color **rojo** engloba dos barras, una de inicio y otra de fin. Estas permiten moverse temporalmente en la gráfica permitiendo visiones mas amplias de toda la señal o más precisas.

En el recuadro de color **azul** también se tiene un campo de inicio y otro de fin, se agregan para poner directamente la hora de inicio y fin de ventana que se quiera analizar, sin tener que utilizar las barras. Adicionalmente, se tienen dos campos más:

- Checkbox “Auto Y”: Para que ajuste automaticamente el voltaje a medida que se ajusta el tiempo
- Suavizar (MA): Es un checkbox en conjunto con un número que determina la ventana. Este sirve para aplicar un Moving Average (MA) de la ventana seleccionada a la señal.

En el recuadro de color **amarillo** se brindan herramientas para mejorar la imagen que se quiere capturar de la señal:

1. Retorna a la imagen inicial.
2. Vuelve una vista para atrás.
3. Va una vista para adelante.
4. Permite mover la vista en todas direcciones.
5. Permite hacer zoom a un rectángulo elegido con el ratón.

6. Abre la configuración del plot.

7. Guarda la imágen visualizada en formato .png.

También se puede exportar la imagen visualizada pero en formato .csv, presionando en el botón “Exportar CSV (Ventana)”. En caso de querer guardarlo con suavizado se debe marcar el checkbox “Guardar Suavizado”. Esto se muestra en la Figura 16.

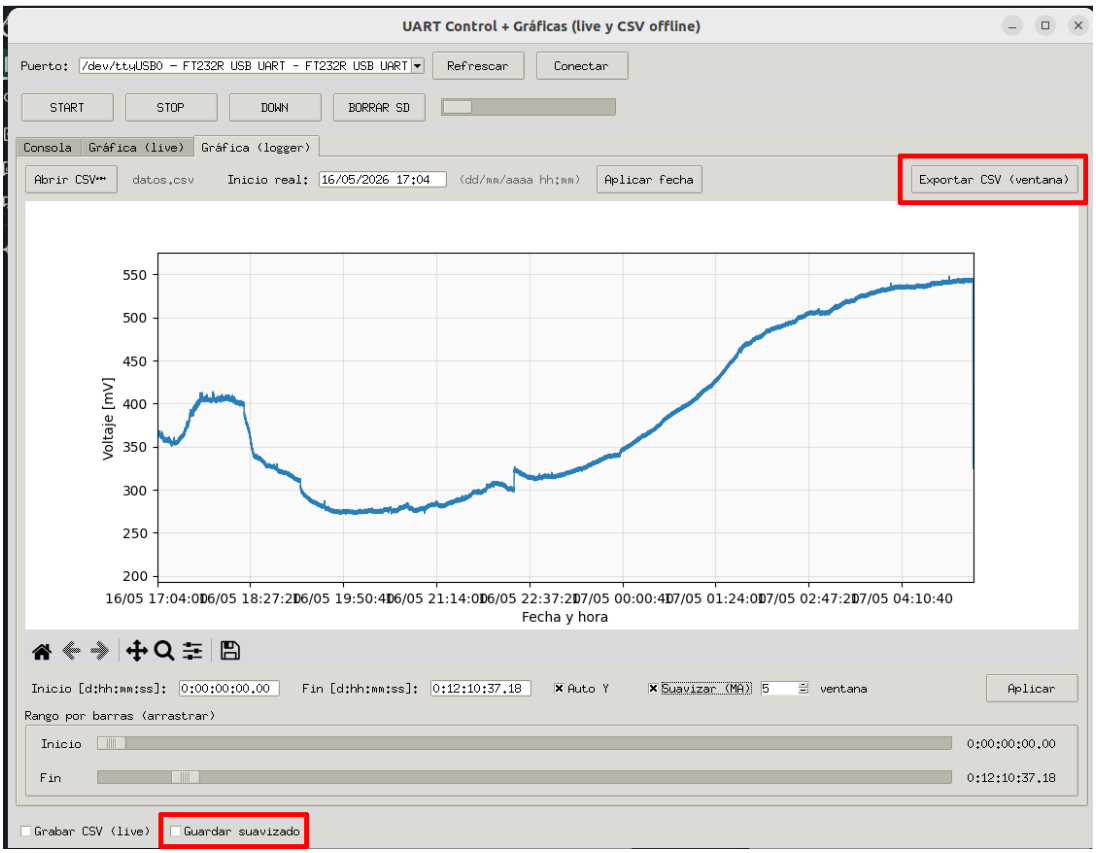


Figura 16: Editar gráfica logger

5. Batería

El dispositivo incorpora una batería recargable de polímero de litio (Li-Po) de $3{,}7\,V$ y $1800\,mAh$, la cual alimenta al sistema en modo autónomo.

La recarga se realiza mediante el conector USB del equipo. Para cargarla, el usuario únicamente debe conectar el dispositivo a una computadora. Si la batería no se encuentra al 100 % de su capacidad, el sistema iniciará automáticamente el proceso de carga.

El tiempo estimado para una carga completa es de aproximadamente 4 horas.

Reemplazo de la batería

En caso de ser necesario sustituir la batería, se recomienda seguir cuidadosamente los siguientes pasos:

1. Verificar que el equipo se encuentre apagado. Para ello, el botón 4 debe estar sin presionar.

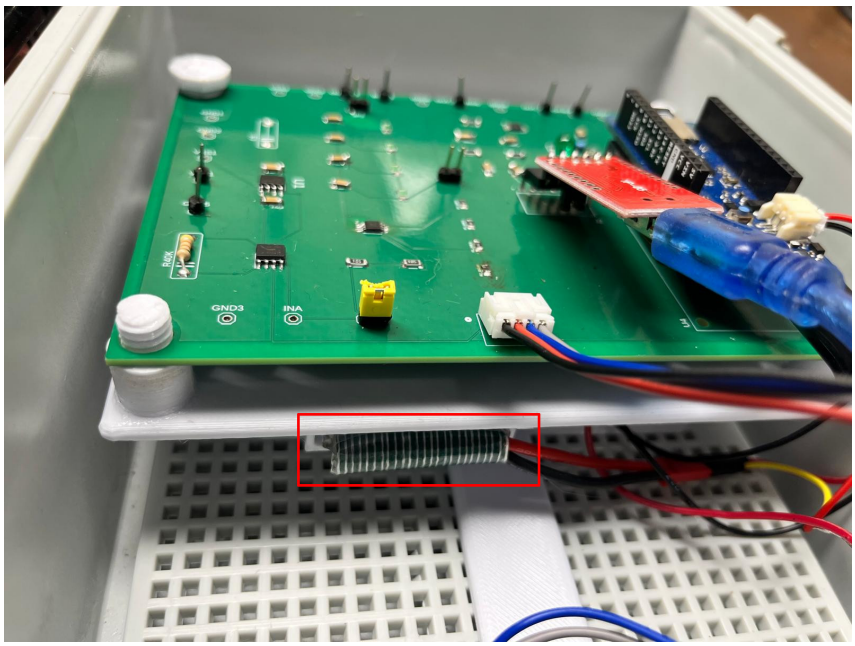


Figura 17: Ubicación de la batería

2. Abrir la caja protectora y retirar cuidadosamente la placa PCB para trabajar con mayor comodidad.
3. Desoldar el polo negativo y el polo positivo de la batería existente.
4. Colocar la nueva batería, asegurándose de respetar la polaridad correcta, y soldar nuevamente ambos polos.
5. Verificar la continuidad eléctrica de las soldaduras mediante un multímetro.
6. Encender el equipo y comprobar el correcto funcionamiento del sistema de alimentación.

Ubicación dentro de la caja

La batería debe colocarse firmemente dentro de la caja , en una posición que evite movimientos durante el transporte y el uso en campo. Asegurando que no interfiera con la placa PCB ni con el cableado interno.

Debajo del apoyo de la electronica la bateria tiene un compartimento donde entra justa y queda fija, como se muestra en la Figura 17.

6. Tarjeta microSD

El sistema utiliza una tarjeta microSD para el almacenamiento de datos. La tarjeta se encuentra alojada en el lector integrado del módulo principal, accesible desde la placa ver Figura 18

Extracción de la tarjeta microSD

1. Apagar el equipo (botón 4 sin presionar).

2. Abrir la caja protectora.
3. Acceder al módulo donde se encuentra la microSD.
4. Presionar suavemente la tarjeta hacia adentro hasta sentir el mecanismo de liberación.
5. Retirar la tarjeta cuidadosamente.

Importante: Nunca retirar la tarjeta con el equipo encendido o durante escritura de datos, ya que puede producirse corrupción del sistema de archivos.

Colocación de la tarjeta

1. Verificar que el equipo esté apagado.
2. Insertar la microSD en el lector hasta que quede firmemente trabada.
3. Encender el equipo.

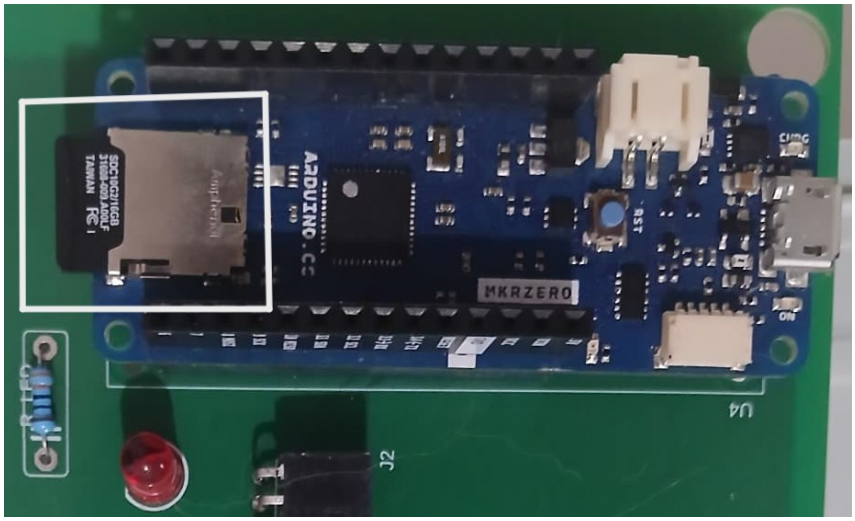


Figura 18: Ubicación de MicroSD

Esta página ha sido intencionalmente dejada en blanco.

Apéndice B

Generación del ejecutable de la interfaz en Windows

Para facilitar el uso de la aplicación, se generó un archivo ejecutable a partir del script principal `interfaz.py`. De esta forma, la interfaz puede ejecutarse directamente en Windows mediante doble clic, sin que el usuario final tenga que instalar Python ni configurar el entorno de desarrollo.

La herramienta utilizada para este proceso fue `PyInstaller`, que permite empaquetar el script de Python junto con las librerías necesarias en un único archivo `.exe`.

El procedimiento se realizó en una computadora con Windows 10/11, conexión a internet y Python instalado. Durante la instalación de Python es importante habilitar la opción *Add Python to PATH*, ya que esto permite ejecutar Python desde la consola de Windows.

Primero se verificó que Python estuviera correctamente instalado. Para ello, se abrió una consola de comandos en Windows y se ejecutó:

```
py --version
```

Luego se colocó el archivo `interfaz.py` en una carpeta local del proyecto. Desde el Explorador de Windows se abrió dicha carpeta, se escribió `cmd` en la barra de dirección y se presionó Enter. Esto permite abrir una consola directamente en la ubicación del archivo. Para verificar que el script estuviera presente, se ejecutó:

```
dir
```

Como buena práctica, se creó un entorno virtual dentro de la carpeta del proyecto. Esto permite aislar las dependencias utilizadas para generar el ejecutable y evita interferencias con otras instalaciones de Python presentes en el sistema.

```
py -3 -m venv .venv
.\.venv\Scripts\activate
```

Una vez activado el entorno virtual, el inicio de la línea de comandos muestra el prefijo `(.venv)`. Luego se actualizó `pip` mediante:

```
python -m pip install --upgrade pip
```

A continuación se instaló `PyInstaller`. Si la interfaz utiliza únicamente Tkinter y librerías estándar de Python, alcanza con ejecutar:

```
pip install pyinstaller
```

En caso de que el programa utilice librerías externas, como `pyserial`, `numpy` o `matplotlib`, estas también deben instalarse dentro del entorno virtual antes de generar el ejecutable. Por ejemplo:

```
pip install pyinstaller pyserial numpy matplotlib
```

Con las dependencias instaladas, se generó el ejecutable mediante el siguiente comando:

```
pyinstaller --noconfirm --clean --windowed --onefile --name Interfaz interfaz.py
```

La opción `--onefile` indica que se genere un único archivo ejecutable. La opción `--windowed` evita que se abra una consola adicional al ejecutar la interfaz, lo cual resulta adecuado para una aplicación gráfica desarrollada con Tkinter.

Al finalizar el proceso, PyInstaller crea las carpetas `build` y `dist`. El ejecutable final queda ubicado en:

```
dist\Interfaz.exe
```

Para verificar el resultado, se ejecutó el archivo `Interfaz.exe` con doble clic. También es recomendable copiar el ejecutable a otra ubicación, por ejemplo al Escritorio, y probarlo desde allí. Esto permite comprobar que la aplicación no dependa de rutas locales de la carpeta original del proyecto.

Una vez validado el funcionamiento, el archivo que debe entregarse al usuario final es:

```
dist\Interfaz.exe
```

El usuario final puede ejecutar este archivo directamente, sin necesidad de instalar Python ni las librerías utilizadas durante el desarrollo.

En caso de que el comando `pyinstaller` no sea reconocido por la consola, puede ejecutarse PyInstaller como módulo de Python:

```
py -m PyInstaller --noconfirm --clean --windowed --onefile --name Interfaz interfaz.py
```

Si el ejecutable se abre y se cierra inmediatamente, conviene generar una versión de prueba con consola habilitada. Esto permite observar los mensajes de error y corregir posibles problemas de ejecución.

```
pyinstaller --noconfirm --clean --onefile --name Interfaz_dbg interfaz.py
```

Luego se ejecuta:

```
dist\Interfaz_dbg.exe
```

Si durante la ejecución aparecen errores relacionados con módulos faltantes, especialmente en programas que utilizan `numpy` o `matplotlib`, puede ser necesario indicar explícitamente a PyInstaller que incluya dichas librerías:

```
pyinstaller --noconfirm --clean --windowed --onefile --name Interfaz ^
  --collect-all matplotlib ^
  --collect-all numpy ^
  interfaz.py
```

Finalmente, si se desea recompilar desde cero, se pueden eliminar las carpetas y archivos generados previamente:

```
build\
dist\
Interfaz.spec
```

Luego se repite el comando de compilación. Esto permite generar una nueva versión limpia del ejecutable.

Esta página ha sido intencionalmente dejada en blanco.

Apéndice C

Graficas

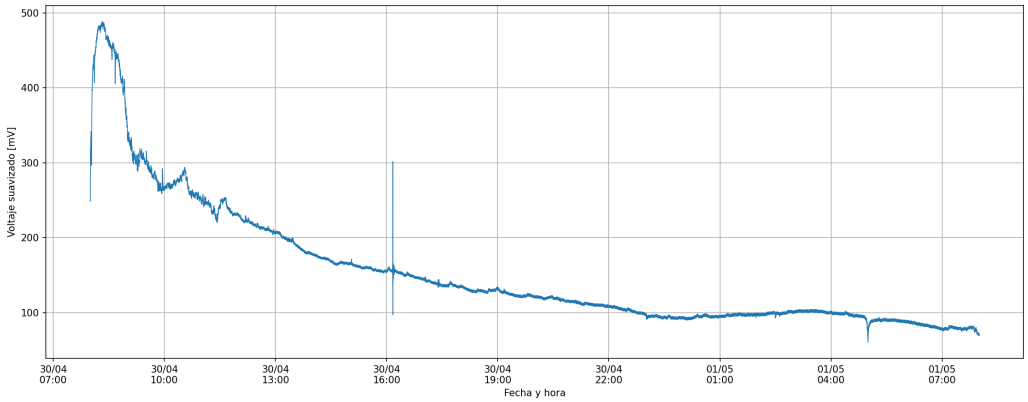


Figura C.1: Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de plata, correspondiente al intervalo entre el 30/04/2026 a las 08:00 h y el 01/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras.

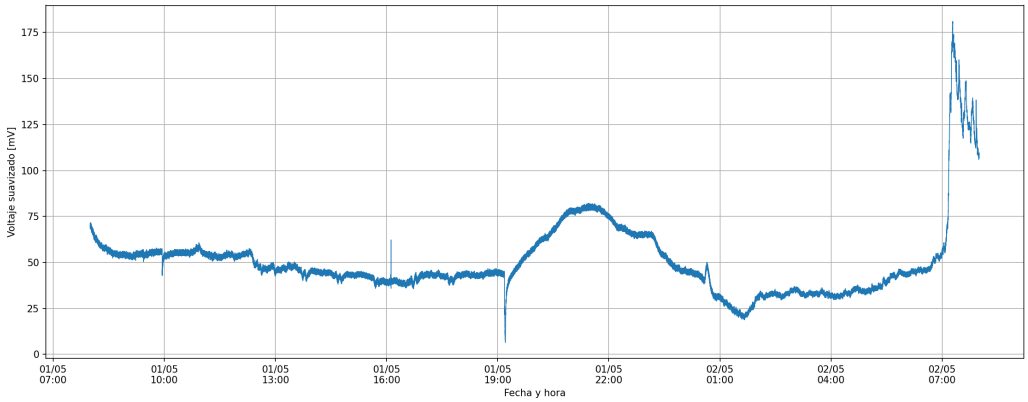


Figura C.2: Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de plata, correspondiente al intervalo entre el 01/05/2026 a las 08:00 h y el 02/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras.

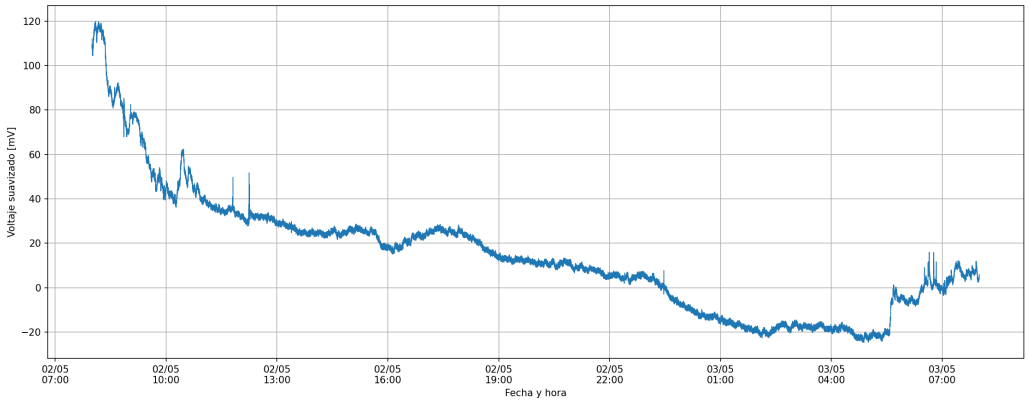


Figura C.3: Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de plata, correspondiente al intervalo entre el 02/05/2026 a las 08:00 h y el 03/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras.

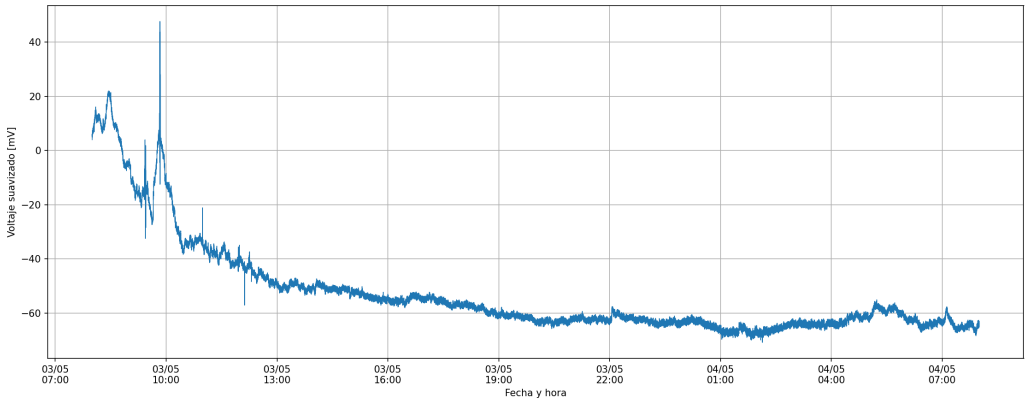


Figura C.4: Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de plata, correspondiente al intervalo entre el 03/05/2026 a las 08:00 h y el 04/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras.

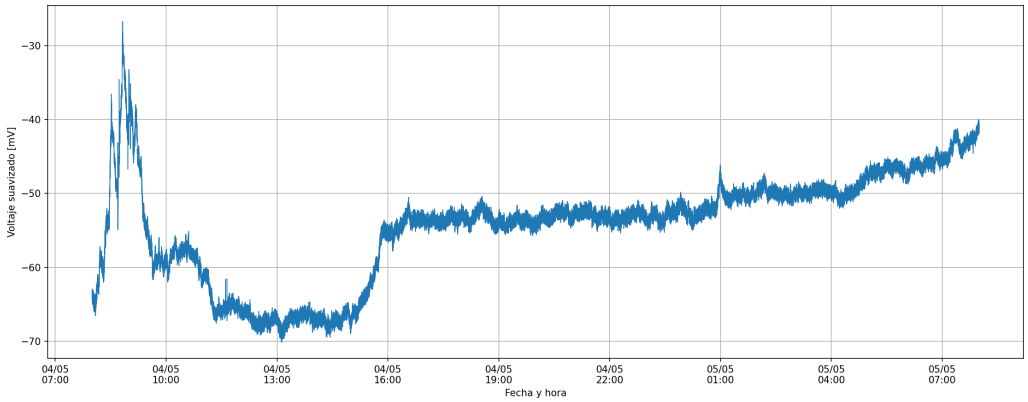


Figura C.5: Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de plata, correspondiente al intervalo entre el 04/05/2026 a las 08:00 h y el 05/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras.

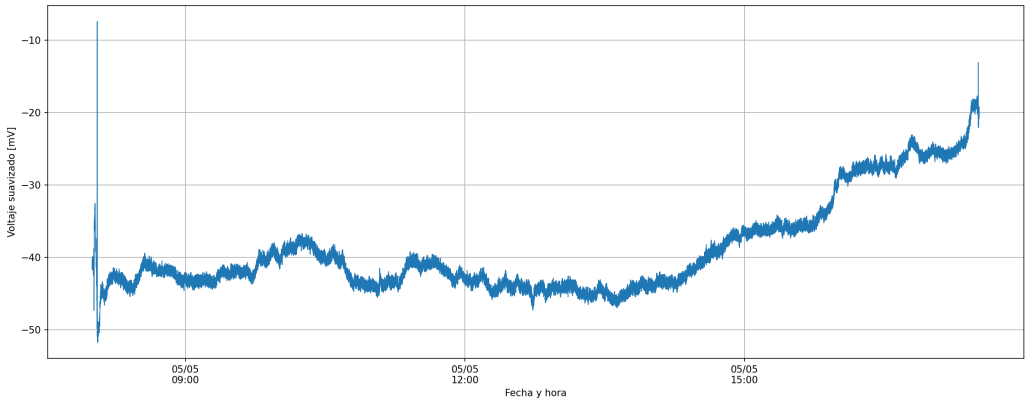


Figura C.6: Segmento final del registro sobre *Mimosa pudica* mediante electrodos de plata, correspondiente al intervalo entre el 05/05/2026 a las 08:00 h y aproximadamente las 16:00 h del mismo día. La señal fue suavizada mediante una media móvil de 50 muestras.

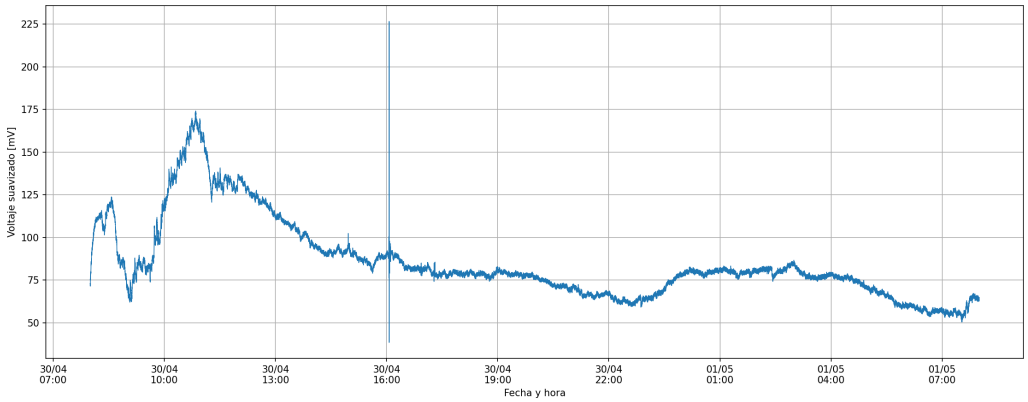


Figura C.7: Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de acero, correspondiente al intervalo entre el 30/04/2026 a las 08:00 h y el 01/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras.

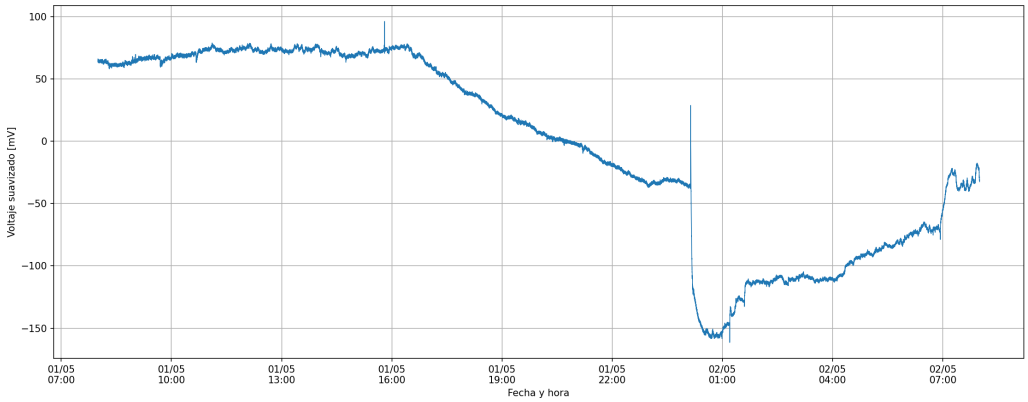


Figura C.8: Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de acero, correspondiente al intervalo entre el 01/05/2026 a las 08:00 h y el 02/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras.

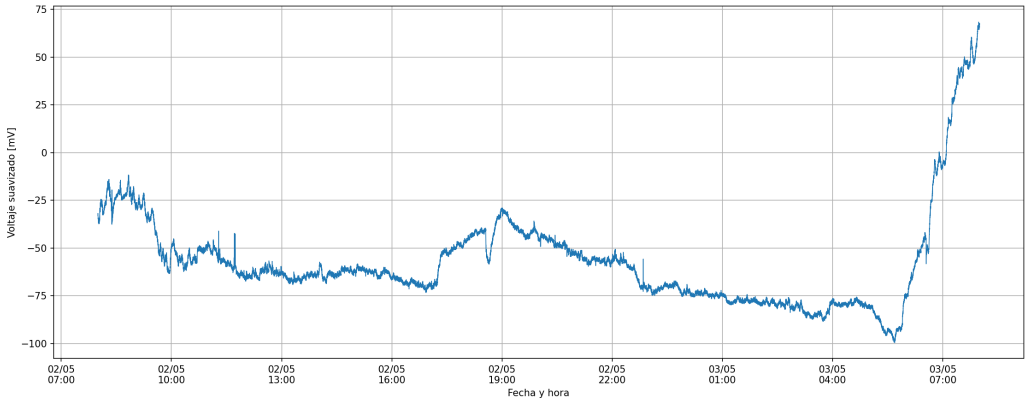


Figura C.9: Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de acero, correspondiente al intervalo entre el 02/05/2026 a las 08:00 h y el 03/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras.

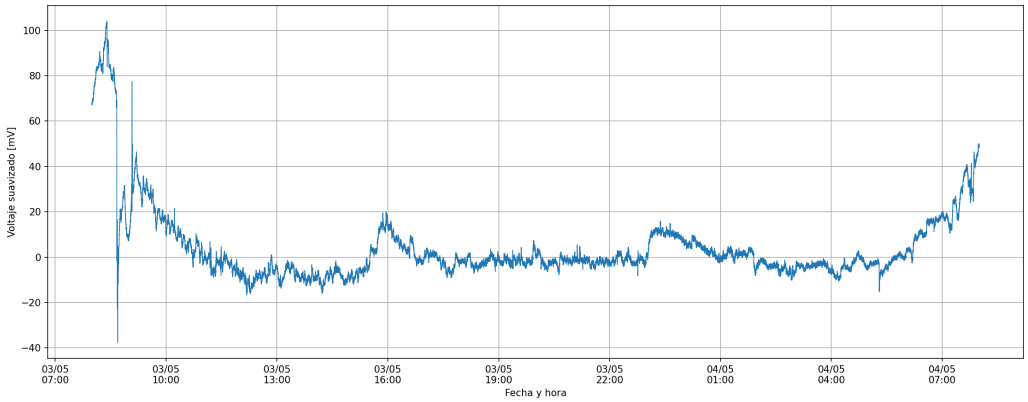


Figura C.10: Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de acero, correspondiente al intervalo entre el 03/05/2026 a las 08:00 h y el 04/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras.

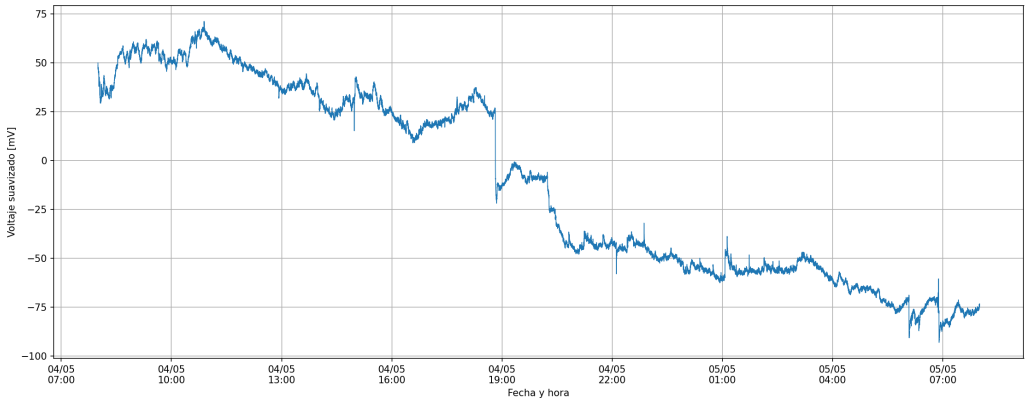


Figura C.11: Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de acero, correspondiente al intervalo entre el 04/05/2026 a las 08:00 h y el 05/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras.

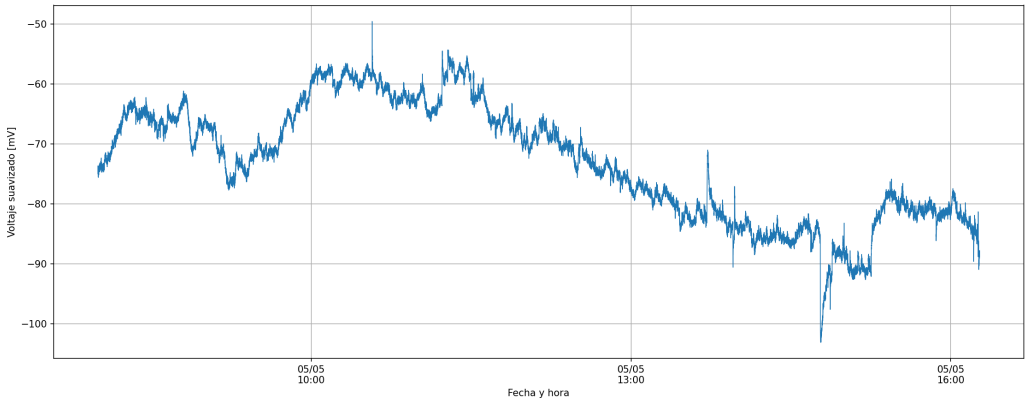


Figura C.12: Segmento final del registro sobre *Mimosa pudica* mediante electrodos de acero, correspondiente al intervalo entre el 05/05/2026 a las 08:00 h y aproximadamente las 16:00 h del mismo día. La señal fue suavizada mediante una media móvil de 50 muestras.

Esta página ha sido intencionalmente dejada en blanco.

Apéndice D

Ajuste de la medida de V_{REF}

Para justificar la diferencia observada en la medición de V_{REF1}, se considera el efecto de carga introducido por el voltímetro. El nodo V_{REF1} se obtiene mediante un divisor resistivo formado por resistencias de $3\,M\Omega$ y $1\,M\Omega$. Sin embargo, al medir dicho nodo, el voltímetro introduce una resistencia de entrada de $10\,M\Omega$, comparable con los valores del divisor.

Aplicando la ley de corrientes de Kirchhoff en el nodo V_{REF1}, se obtiene:

$$\frac{V_{CC} - V_{REF1}}{3\,M\Omega} = \frac{V_{REF1} - V_V}{10\,M\Omega} + \frac{V_{REF1}}{1\,M\Omega} \tag{D.1}$$

donde V_V corresponde al valor indicado por el voltímetro. Como todas las resistencias se encuentran expresadas en $M\Omega$, la ecuación puede escribirse como:

$$\frac{V_{CC} - V_{REF1}}{3} = \frac{V_{REF1} - V_V}{10} + V_{REF1} \tag{D.2}$$

Reordenando los términos:

$$\frac{V_{CC}}{3} + \frac{V_V}{10} = V_{REF1}\left(\frac{1}{3} + \frac{1}{10} + 1\right) \tag{D.3}$$

Por lo tanto, el valor corregido de la tensión en el nodo resulta:

$$V_{REF1}^{AJUSTADO} = \left(\frac{V_{CC}}{3} + \frac{V_V}{10}\right)\left(\frac{1}{\frac{1}{3} + \frac{1}{10} + 1}\right) \tag{D.4}$$

Sustituyendo los valores medidos:

$$V_{REF1}^{AJUSTADO} = \left(\frac{3{,}29}{3} + \frac{0{,}771}{10}\right)\left(\frac{1}{\frac{1}{3} + \frac{1}{10} + 1}\right) \approx 0{,}8189\,V \tag{D.5}$$

De esta forma, al considerar la resistencia de entrada del voltímetro, el valor ajustado de V_{REF1} resulta aproximadamente $0{,}82\,V$, lo cual es coherente con el valor esperado de $V_{CC}/4$.

Esta página ha sido intencionalmente dejada en blanco.

Referencias

[1] AG Electrónica S.A.P.I. de C.V. ABX00012: Arduino MKR Zero. https://agelectronica.com/detalle?busca=ABX00012, 2021. Ficha técnica / documento de producto.

[2] Analog Devices. ADA4505-1/ADA4505-2/ADA4505-4 Data Sheet. https://www.analog.com/media/en/technical-documentation/data-sheets/ada4505-1_4505-2_4505-4.pdf, 2017. Amplificador operacional micropower, rail-to-rail.

[3] Analog Devices. *LTC1044: Switched-Capacitor Voltage Converter*. Analog Devices, 2023. Datasheet.

[4] Arduino. Arduino language reference: Serial. https://www.arduino.cc/reference/en/language/functions/communication/serial/, 2024. Accessed: 2026-03-06.

[5] Arduino. Arduinolowpower.cpp: Low power management for samd boards. https://github.com/arduino-libraries/ArduinoLowPower/blob/master/src/samd/ArduinoLowPower.cpp, 2024. Accessed: 2026-03-04.

[6] Arduino. Mkr zero schematic. https://docs.arduino.cc/resources/schematics/ABX00012-schematics.pdf, 2026. Accessed: 2026-03-21.

[7] Arduino. SD Library for Arduino. https://github.com/arduino-libraries/SD, 2026. Repositorio oficial de la librería SD de Arduino. Accedido: 18 de marzo de 2026.

[8] ARM Ltd. *Cortex-M0+ Devices Generic User Guide*. ARM Limited, 2012. Available online.

[9] Autodesk, Inc. Autodesk Fusion 360, 2026. Software de diseño asistido por computadora (CAD/CAM/CAE) con capacidades de diseño electrónico.

[10] A. Cattani, L. de Riedmatten, J. Roulet, T. Smit-Sadki, E. Alfonso, A. Kurenda, M. Graeff, E. Remolif, and M. Rienth. Water status assessment in grapevines using plant electrophysiology. *OENO One*, 58(4):8209, 2024.

[11] Future Technology Devices International Limited. *FT232R USB UART IC Datasheet*. FTDI, 2020. Applies to the FT232R family, including FT232RL.

[12] Alberto Hernández Pérez and David Viñas Elías. Electronic system design for recording the electrical response of Mimosa pudica due to white light stimulus. *Revista Técnica de la Facultad de Ingeniería Universidad del Zulia*, 38(2):57–63, 2015.

[13] International Electrotechnical Commission. IEC 60529: Degrees of protection provided by enclosures (IP Code), 2013. International Standard.

[14] Chris Liechti. Welcome to pyserial's documentation. `https://pyserial.readthedocs.io/`, 2026. Documentación oficial. Consultado en marzo de 2026.

[15] Matplotlib Development Team. Matplotlib documentation. `https://matplotlib.org/stable/`, 2026. Documentación oficial. Consultado en marzo de 2026.

[16] Microchip Technology. MCP6021/1R/2/3/4 10 MHz Rail-to-Rail Input/Output Op Amps Data Sheet. `https://ww1.microchip.com/downloads/en/devicedoc/20001685e.pdf`, 2016. Amplificador operacional rail-to-rail de 10 MHz.

[17] Microchip Technology. MCP6001/1R/1U/2/4 1 MHz Low-Power Op Amp Data Sheet. `https://ww1.microchip.com/downloads/en/DeviceDoc/MCP6001-1R-1U-2-4-1-MHz-Low-Power-Op-Amp-DS20001733L.pdf`, 2019. Amplificador operacional de bajo consumo.

[18] NumPy Developers. Numpy documentation. `https://numpy.org/doc/stable/`, 2026. Documentación oficial. Consultado en marzo de 2026.

[19] Onshape, Inc. Onshape. `https://www.onshape.com/en/`, 2026. Cloud-native CAD platform.

[20] Python Software Foundation. csv — csv file reading and writing. `https://docs.python.org/3/library/csv.html`, 2026. Documentación oficial. Consultado en marzo de 2026.

[21] Python Software Foundation. datetime — basic date and time types. `https://docs.python.org/3/library/datetime.html`, 2026. Documentación oficial. Consultado en marzo de 2026.

[22] Python Software Foundation. queue — a synchronized queue class. `https://docs.python.org/3/library/queue.html`, 2026. Documentación oficial. Consultado en marzo de 2026.

[23] Python Software Foundation. threading — thread-based parallelism. `https://docs.python.org/3/library/threading.html`, 2026. Documentación oficial. Consultado en marzo de 2026.

[24] Python Software Foundation. tkinter — python interface to tcl/tk. `https://docs.python.org/3/library/tkinter.html`, 2026. Documentación oficial. Consultado en marzo de 2026.

[25] Shenzhen Creality 3D Technology Co., Ltd. Creality Print, 2026. Software de laminado (*slicer*) para impresión 3D.

[26] Texas Instruments. *LM27761 Low-Noise, Regulated, Switched-Capacitor Voltage Inverter Datasheet*. Texas Instruments, 2025. Rev. D.

[27] Alexander G. Volkov, Tolulope Adesina, Vsevolod S. Markin, and Emil Jovanov. Signal transduction in Mimosa pudica: biologically closed electrical circuits. *Plant, Cell & Environment*, 33(5):816–827, 2010.

Esta página ha sido intencionalmente dejada en blanco.

Índice de tablas

3.1. Listado de componentes del Prototipo 3. 24
3.2. Especificaciones teóricas del Prototipo 3. 25

4.1. Configuración del periférico SYSCTRL 35
4.2. Configuración del periférico GCLK 36
4.3. Configuración del periférico TC3 37
4.4. Configuración del NVIC para TC3 38
4.5. Configuración del ADC 39

5.1. Tensiones medidas en distintos nodos del circuito. 64

Esta página ha sido intencionalmente dejada en blanco.

Índice de figuras

1.1. Linea de tiempo y diseño experimental de las pruebas de deshidratación en una vid. [10] 2

2.1. Arquitectura general del sistema y conexiones principales en torno a la placa Arduino MKR Zero. 8
2.2. Interconexión física de los principales elementos del sistema. 8
2.3. Foto del sistema. 9
2.4. Flujo de datos según el modo de operación. 10
2.5. Señal capturada en la interfaz. 11

3.1. Diagrama de la Electrónica. 14
3.2. Esquemático eléctrico del Prototipo 1. Se observa la etapa diferencial implementada con INA118 (siendo iN- y iN+ sus entradas) y la etapa de filtrado activo realizada con ADA4505. 17
3.3. Respuesta en frecuencia del filtro pasabajos del Prototipo 1. 18
3.4. Esquemático correspondiente al Prototipo 2. 19
3.5. Esquemático del Prototipo 3 a nivel de simulación en LTSpice. . . 22
3.6. Implementación del Prototipo 3 (final). 23
3.7. Diagrama de Bode del Prototipo 3 simulado en LTSpice. 25
3.8. Salida del amplificador para una entrada de 10 mV a 1 Hz. 26
3.9. PCB diseñada para el Prototipo 3. 26
3.10. Caja IP67 seleccionada. 28
3.11. Modelo 3D del soporte interno en herramienta OnShape. 28
3.12. Conector para electrodos seleccionado. 29
3.13. Conector hembra USB tipo A seleccionado. 29
3.14. Pulsador metálicos seleccionado. 29
3.15. Modelo 3D del electrodo para el tallo en herramiento OnShape. . . 31
3.16. Modelo 3D del electrodo para la hoja en herramiento OnShape. . . 31

4.1. Diagrama alto nivel de la Arquitectura de Firmware 34
4.2. Diagrama jerarquico del Firmware 35
4.3. Divisor resistivo utilizado para la medición de la batería [6]. 43
4.4. Diagrama de flujo del bucle principal 46
4.5. Estados del sistema 51
4.6. Arquitectura general de la aplicación desarrollada. 54
4.7. Ventana principal de la interfaz gráfica. 57

4.8. Interfaz utilizada durante la descarga de archivos del modo *Logger*. 57
4.9. Vista de adquisición en *Live*. 58
4.10. Vista de análisis fuera de línea. 60

5.1. Diagrama de Bode medido para la configuración con batería y electrodos de aguja de plata, procesado con python 64
5.2. Diagrama de Bode medido para la configuración con batería y electrodos de aguja de acero, procesado con python 65
5.3. Diagrama de Bode medido para la configuración sin batería y electrodos de aguja de acero, procesado con python 65
5.4. Excursión temporal de la salida obtenida a partir del procesamiento en Python. 67
5.5. FFT de la excursión de salida obtenida a partir del procesamiento en Python. 67
5.6. Consumo del temporizador, procesada en Python. 68
5.7. Consumo total del sistema en función del tiempo, procesado en Python. 68
5.8. Detalle del pico de consumo cercano a 30 mA obtenido a partir del procesamiento en Python. 69
5.9. Voltaje de la batería en función del tiempo 69
5.10. Primera prueba realizada con electrodos tipo aguja de acero inoxidable. 71
5.11. FFT de la primera captura, expresada en ciclos por día, sin componente continua y con ventana de Hann. 72
5.12. Segunda captura realizada con el dispositivo. Indicando con cruces rojas los momentos de riego. 72
5.13. FFT de la segunda captura, expresada en ciclos por día, sin componente continua y con ventana de Hann. 73
5.14. Registro de 30 minutos obtenido sobre una *Mimosa pudica* mediante electrodos de acero. La señal fue suavizada con una ventana de 50 muestras. 74
5.15. Armado experimental de la prueba simultánea sobre *Mimosa pudica*. 75
5.16. Registro completo obtenido sobre *Mimosa pudica* mediante electrodos de acero. La señal fue suavizada mediante una media móvil de 50 muestras. 76
5.17. Registro completo obtenido sobre *Mimosa pudica* mediante electrodos de plata. La señal fue suavizada mediante una media móvil de 50 muestras. 76
5.18. Comparación entre los registros obtenidos con electrodos de acero y plata sobre *Mimosa pudica*. 77
5.19. Segmento temporal asociado a un estímulo mecánico sobre *Mimosa pudica*, registrado mediante electrodos de acero. La señal fue suavizada mediante una media móvil de 50 muestras. 78
5.20. Segmento temporal asociado a un estímulo mecánico sobre *Mimosa pudica*, registrado mediante electrodos de plata. La señal fue suavizada mediante una media móvil de 50 muestras. 78

5.21. Señal registrada con electrodo de acero durante una semana. En rojo se indican algunos saltos observados en la señal. 79
5.22. Señal registrada con electrodo de plata durante una semana. En rojo se indican algunos saltos observados en la señal. 79
5.23. Ciclos diarios correspondientes a los días 1 y 2 del registro. 80
5.24. Ciclos diarios correspondientes a los días 3 y 4 del registro. 80
5.25. Ciclos diarios correspondientes a los días 5 y 6 del registro. 80
5.26. Ciclos diarios correspondientes a los días 7 y 8 del registro. 81
5.27. Ciclos diarios correspondientes a los días 9 y 10 del registro. 81
5.28. Ciclos diarios correspondientes a los días 11 y 12 del registro. 81
5.29. Ciclos diarios correspondientes a los días 13 y 14 del registro. 81
5.30. Superposición de los ciclos diarios de 24 horas correspondientes a los 14 días analizados. 81
5.31. Registro obtenido utilizando un electrodo comercial adaptado sobre la hoja y un electrodo de plata ubicado sobre el tronco. La señal fue suavizada mediante una media móvil de 500 muestras. 82

C.1. Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de plata, correspondiente al intervalo entre el 30/04/2026 a las 08:00 h y el 01/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras. 118
C.2. Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de plata, correspondiente al intervalo entre el 01/05/2026 a las 08:00 h y el 02/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras. 118
C.3. Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de plata, correspondiente al intervalo entre el 02/05/2026 a las 08:00 h y el 03/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras. 119
C.4. Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de plata, correspondiente al intervalo entre el 03/05/2026 a las 08:00 h y el 04/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras. 119
C.5. Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de plata, correspondiente al intervalo entre el 04/05/2026 a las 08:00 h y el 05/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras. 120
C.6. Segmento final del registro sobre *Mimosa pudica* mediante electrodos de plata, correspondiente al intervalo entre el 05/05/2026 a las 08:00 h y aproximadamente las 16:00 h del mismo día. La señal fue suavizada mediante una media móvil de 50 muestras. 120
C.7. Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de acero, correspondiente al intervalo entre el 30/04/2026 a las 08:00 h y el 01/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras. 121

C.8. Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de acero, correspondiente al intervalo entre el 01/05/2026 a las 08:00 h y el 02/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras. 121
C.9. Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de acero, correspondiente al intervalo entre el 02/05/2026 a las 08:00 h y el 03/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras. 122
C.10. Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de acero, correspondiente al intervalo entre el 03/05/2026 a las 08:00 h y el 04/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras. 122
C.11. Segmento de 24 horas del registro sobre *Mimosa pudica* mediante electrodos de acero, correspondiente al intervalo entre el 04/05/2026 a las 08:00 h y el 05/05/2026 a las 08:00 h. La señal fue suavizada mediante una media móvil de 50 muestras. 123
C.12. Segmento final del registro sobre *Mimosa pudica* mediante electrodos de acero, correspondiente al intervalo entre el 05/05/2026 a las 08:00 h y aproximadamente las 16:00 h del mismo día. La señal fue suavizada mediante una media móvil de 50 muestras. 123

svg

Esta es la última página.
Compilado el viernes 29 mayo, 2026.
http://iie.fing.edu.uy/