

INFORME FINAL DEL PROYECTO “Detección de Metales Pesados en Aguas Residuales”

Versión 8

Fecha: 2025-06-17

Autor: Camilo

1. OBJETIVO GENERAL

Desarrollar un sistema de monitoreo electroquímico capaz de procesar archivos .psession (o datos IoT de Mini PC), extraer curvas, realizar PCA, estimar PPM de metales pesados, almacenarlos en PostgreSQL y mostrarlos en una interfaz gráfica (GUI).

2. ESTRUCTURA GENERAL DEL PROYECTO

Carpeta principal: COINVESTIGACION1

- .venv/
- .vscode/settings.json
- data/
 - archivos .psession de prueba
 - matrices y muestras (.xlsx, .csv)
 - limits_ppm.json (umbrales ppm)
- sdk/PSPythonSDK/pspython (SDK PalmSens)
- src/
 - db_connection.py ← conexión y migración de BD
 - insert_data.py ← inserción de sesiones y mediciones
 - interfaz_grafica.py ← GUI con filtros, curvas, PCA, ppm
 - main.py ← orquestador (si aplica)
 - pstrace_connection.py
 - pstrace_session.py ← lectura y extracción de datos
- schema.sql ← script de creación y ALTER TABLE
- limits_ppm.json ← archivo de umbrales ppm
- requirements.txt
- debug.log, pstrace_debug.log

3. SCRIPT DE BASE DE DATOS (schema.sql)

-- 1) Tabla sessions:

```
CREATE TABLE IF NOT EXISTS sessions (  
  id SERIAL PRIMARY KEY,  
  filename TEXT NOT NULL,  
  loaded_at TIMESTAMP NOT NULL,  
  scan_rate DOUBLE PRECISION,  
  start_potential DOUBLE PRECISION,  
  end_potential DOUBLE PRECISION,  
  software_version TEXT  
);
```

-- 2) Tabla measurements:

```
CREATE TABLE IF NOT EXISTS measurements (
```

```

id SERIAL PRIMARY KEY,
session_id INTEGER NOT NULL REFERENCES sessions(id) ON DELETE
CASCADE,
title TEXT NOT NULL,
timestamp TIMESTAMP NOT NULL,
device_serial TEXT NOT NULL,
curve_count INTEGER NOT NULL,
pca_data DOUBLE PRECISION[] NOT NULL,
ppm_estimations DOUBLE PRECISION[] NOT NULL
);
-- 3) Tabla curves:
CREATE TABLE IF NOT EXISTS curves (
id SERIAL PRIMARY KEY,
measurement_id INTEGER NOT NULL REFERENCES measurements(id) ON
DELETE CASCADE,
curve_index INTEGER NOT NULL,
num_points INTEGER NOT NULL
);
-- 4) Tabla points:
CREATE TABLE IF NOT EXISTS points (
id SERIAL PRIMARY KEY,
curve_id INTEGER NOT NULL REFERENCES curves(id) ON DELETE CASCADE,
potential DOUBLE PRECISION NOT NULL,
current DOUBLE PRECISION NOT NULL
);
-- Índices:
CREATE INDEX IF NOT EXISTS idx_measurements_session ON
measurements(session_id);
CREATE INDEX IF NOT EXISTS idx_curves_measurement ON
curves(measurement_id);
CREATE INDEX IF NOT EXISTS idx_points_curve ON points(curve_id);

```

4. MÓDULOS Y SUS BLOQUES PRINCIPALES

4.1 pstrace_session.py

- cargar_limites(): lee limits_ppm.json
- configurar_sdk(): inicializa Python.NET y SDK PalmSens
- procesar_ciclos(curves): elimina primer ciclo, promedia ciclos 2-5
- extract_session_dict(path): carga .pssession, extrae session_info y lista measurements, genera matriz_pca.csv

4.2 insert_data.py

Bloque 3.2 guardar_session(conn, filename, info): inserta en sessions con metadatos

Bloque 3.3 guardar_mediciones(conn, session_id, measurements): inserta measurements, curves y points

Bloque 3.4 script principal: llama extract_session_dict, luego guarda todo en BD

4.3 interfaz_grafica.py

Bloque 2.2.2 _create_filters_panel: crea filtros ID, fechas y combobox Dispositivo, llama load_devices()

Bloque 2.2.3 query_sessions(): consulta SQL con LATERAL UNNEST de ppm_estimations. Pendiente agregar filtro device_serial

Bloque 2.2.4 _create_meta_panel: muestra cuatro metadatos reales

Bloque 2.4 on_session_select(): lee de sessions scan_rate, start_potential, end_potential, software_version

Bloque load_file(): reemplazado subprocess por llamada in-process a extract_session_dict, inserta en BD y actualiza UI

5. ERRORES CRÍTICOS CORREGIDOS

- Crash 3221225477: eliminar subprocess, usar extract_session_dict internamente
- Columnas faltantes en sessions: añadidas scan_rate, start_potential, end_potential, software_version
- Combobox Dispositivo vacío: load_devices() invocado tras panel de filtros
- Referencias a file_size/processing_time: eliminadas

6. PRUEBAS REALIZADAS

- Inicio de GUI sin errores
- Carga manual de .pssession, extracción de datos y guardado en BD
- Visualización de curvas, PCA y tabla de ppm
- Combobox dispositivo poblado
- query_sessions() muestra sesiones filtradas por fecha e ID

7. PENDIENTES

7.1 Bloque 2.3 (inmediato): implementar filtro por dispositivo en query_sessions()

7.2 Revisar y ajustar limits_ppm.json según estándares oficiales:
Cadmio (Cd): 0,10 mg/L

Cinc (Zn): 3,00 mg/L

Cobre (Cu): 1,00 mg/L

Cromo (Cr): 0,50 mg/L

Níquel (Ni): 0,50 mg/L

7.3 Bloque 4: tooltips explicativos en la GUI

7.4 Bloque 5: exportación automática a PDF

7.5 Bloque 6: conexión IoT en Mini PC (MQTT/WebSocket, buffering, reconexión)

8. RECOMENDACIÓN INMEDIATA

- Actualizar limits_ppm.json con los valores oficiales
- Corregir query_sessions() para filtrar por dispositivo
- Verificar de nuevo carga de .psession y UI