

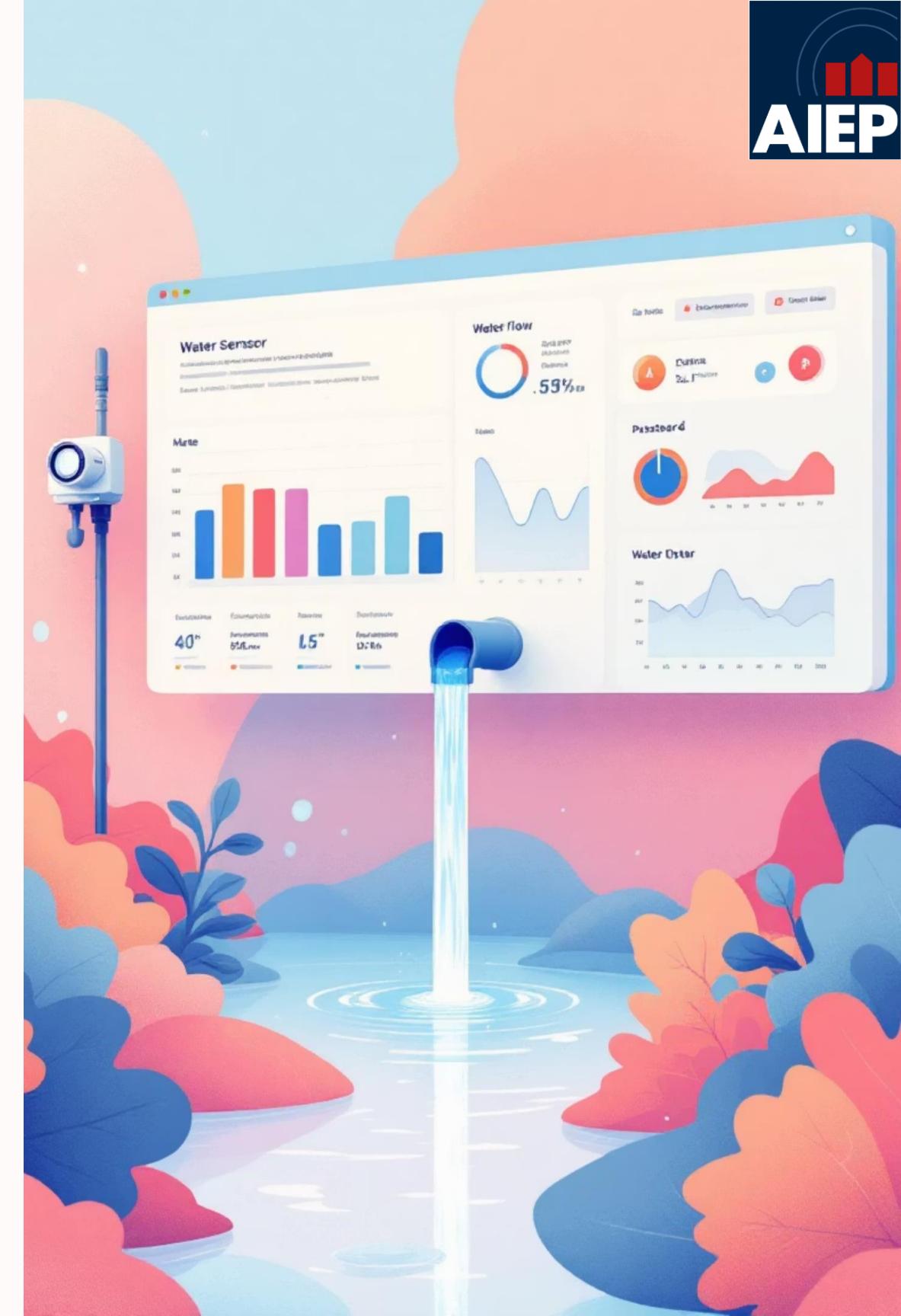


# Zaino Web: Plataforma de Monitoreo Hídrico Inteligente

Proyecto IoT + Web para la Medición de Flujo y Datos Meteorológicos

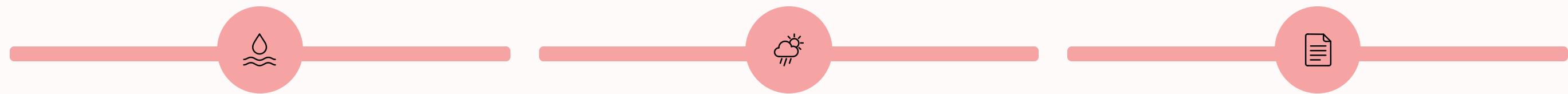
Equipo: Urra Technologies

Fecha: 03/11/2025



# El Desafío de la Monitorización Hídrica No Integrada

En muchas comunidades, la gestión y el uso del agua se ven obstaculizados por la falta de datos continuos y transparentes. Abordamos tres problemas clave que dificultan la toma de decisiones eficiente.



## Monitorización Inconsistente

La medición del caudal de agua (flujo instantáneo y acumulado) es a menudo manual o intermitente, lo que impide un seguimiento preciso y en tiempo real.

## Datos Desconectados

Existe una clara falta de integración entre los datos de flujo hídrico y las variables meteorológicas relevantes, como temperatura y precipitación.

## Reportes Manuales

La generación de informes estructurados y accesibles es laboriosa, sin una herramienta que permita visualizar, exportar y alertar automáticamente.

# Objetivo Central: Integración y Visualización en Tiempo Real

Zaino Web busca revolucionar la monitorización hídrica mediante una solución completa de IoT y desarrollo web.

## Objetivo General

Diseñar una plataforma web que visualice datos de flujo hídrico en tiempo real, integre variables meteorológicas cruciales y genere informes mensuales automatizados para la transparencia y gestión.

## Integración Meteorológica

Integrar datos de temperatura, presión y sensación térmica vía API o mediante ingeniería inversa de servicios existentes (Weathercloud).

## Medición de Flujo Hídrico

Medir el flujo instantáneo y acumulado con precisión mediante un sensor conectado a la nube de IoT (Arduino IoT Cloud).

## Informes y Exportación

Desarrollar un frontend responsive que permita la visualización de gráficos y estadísticas, y la generación de informes en formatos JSON y PDF directamente desde el cliente.

# Stack Tecnológico: Soluciones Ligeras y Robustas

Hemos seleccionado herramientas modernas y eficientes para garantizar un rendimiento óptimo y una arquitectura escalable.



## Backend y Lógica de Negocio

**Python 3.x + Flask:** Se utiliza como framework minimalista para el manejo de rutas, lógica de negocio y consumo de APIs externas. Imagen Docker base: `python:3.11-slim`.

## Frontend y Visualización

**HTML5 + CSS3 + JS (ES6):** Se complementa con Bootstrap 5 para el diseño responsive y jQuery para manipulación eficiente del DOM. Se usa un esquema de color azul-agua.

## Hardware y Conectividad

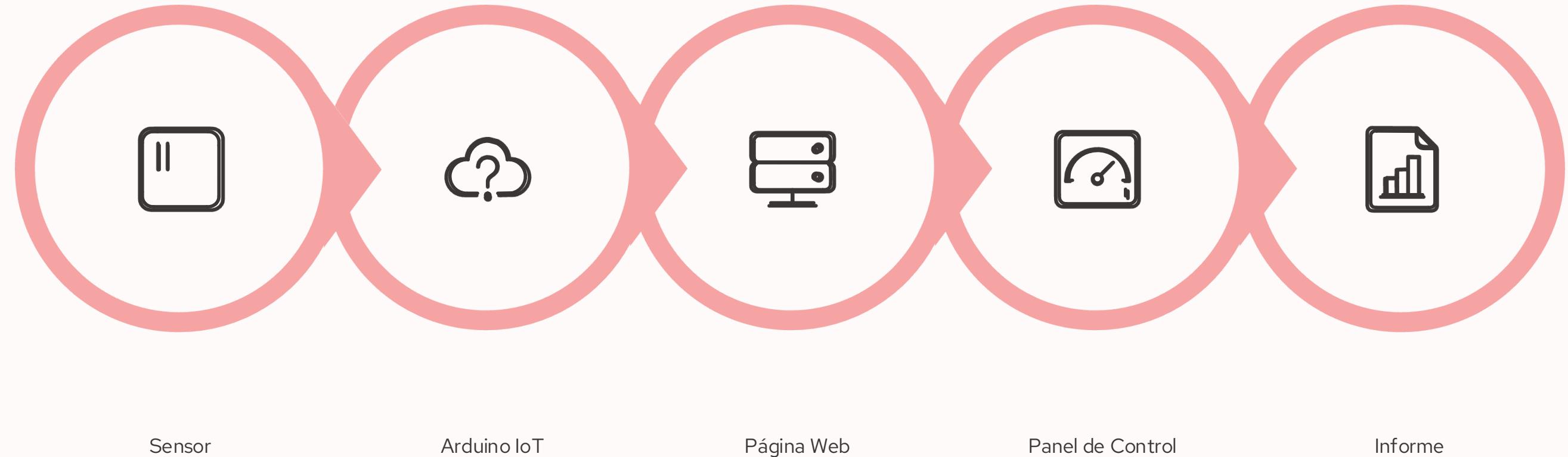
**Sensor de Flujo + Arduino IoT Cloud:** La infraestructura IoT gestiona la conexión del sensor, permitiendo que los datos sean consumidos por el backend.

## Exportación y Rendimiento

**html2canvas + jsPDF:** Utilizados en el cliente para la generación de PDFs. Se implementó un Cache TTL de 8 segundos en el endpoint del flujómetro para optimizar la latencia.

# Arquitectura del Sistema: Un Flujo de Datos Limpio

El sistema se ha diseñado con una arquitectura modular, desde el sensor hasta el informe final, priorizando la eficiencia del flujo de datos y la baja latencia.



- El Frontend actualiza la visualización (gauge, gráfico histórico y estadísticas) cada ~10 segundos.
- El **cacheo TTL** (Time-To-Live) minimiza la carga en el backend al reducir las consultas repetitivas al servicio de IoT.
- El despliegue en producción se realiza mediante **Docker**, asegurando un entorno consistente y portable.

# Endpoints Críticos y Abstracción de APIs

La API RESTful centraliza el acceso a los datos de flujo y meteorológicos, manteniendo una clara separación de responsabilidades.

## Estructura de la API

GET /api/arduino/flowmeter	Flujo actual y acumulado (TTL 8s).
GET /api/weather	Datos meteorológicos (temperatura, presión, sensación).

### Ingeniería Inversa

El archivo `weathercloud_py.py` implementa la lógica de ingeniería inversa necesaria para integrar datos de Weathercloud, ya que no existe una API pública directa.

## Organización de Código

### app/controllers

Manejo de rutas y lógica principal.

### app/static

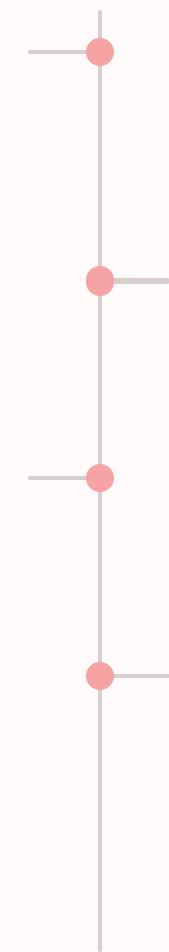
Archivos estáticos: JS, CSS, imágenes.

### app/utils

Funciones de ayuda (e.g., lógica de Weathercloud).

### app/templates

Vistas HTML del dashboard.



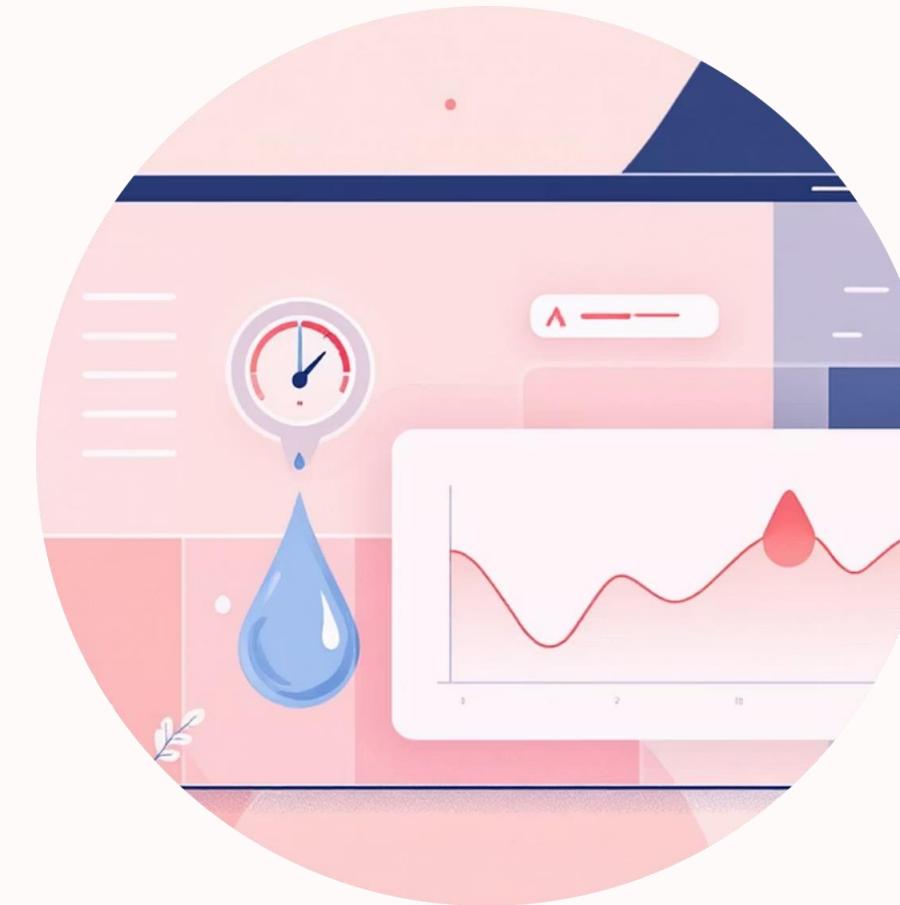
# UI/UX: Transparencia y Adaptabilidad Visual

El diseño prioriza la claridad de los datos críticos y la accesibilidad en cualquier dispositivo, utilizando un esquema de color azul-agua para reforzar la temática.



## Diseño Responsive

Adaptabilidad total para dispositivos móviles y de escritorio, crucial para la consulta en campo.



## Visualización de Datos

Panel principal con **gauge** (flujo instantáneo), gráfico histórico y estadísticas en vivo para una comprensión inmediata.



## Generación de Informes

Botón destacado 'Generar Informe' con feedback visual (spinner) durante el proceso de exportación cliente (html2canvas + jsPDF).

# Navegando los Desafíos Técnicos Clave

El desarrollo de Zaino Web requirió soluciones ingeniosas para mantener el rendimiento y la interoperabilidad con servicios externos.

## Mantenimiento de la API Meteorológica

La dependencia de la ingeniería inversa para la API de Weathercloud implica un riesgo de **mantenimiento continuo** ante posibles cambios en la estructura del servicio externo.

## Optimización de Rendimiento con Caché

Implementación de **Cache TTL** (8s) en el endpoint del flujómetro para prevenir la sobrecarga del backend por parte de múltiples usuarios en el frontend.

## Limitaciones de Exportación PDF Cliente

Aunque la exportación PDF basada en el cliente (**jsPDF**) reduce la carga del servidor, presenta límites en el tamaño y la complejidad de los datos exportables.

## Sincronización Hardware-Software

Asegurar una **baja latencia** en la integración del sensor con Arduino IoT Cloud y el backend Flask, manteniendo la integridad del flujo en tiempo real.

# Resultados y Beneficios Clave del Proyecto

Zaino Web ofrece valor inmediato a los gestores de proyectos y equipos técnicos al transformar datos brutos en inteligencia accionable.

10s

## Actualización de Datos

Supervisión en tiempo real del flujo de agua con una tasa de refresco rápida y optimizada.

1

## Arquitectura Única

Diseño ligero y moderno que integra IoT, Backend Flask y Frontend en un solo sistema escalable.

2

## Formatos de Exportación

Transparencia total y accesibilidad mediante la generación de informes en PDF y JSON.

- ❑ Zaino Web es un **caso educativo de éxito** que demuestra cómo una pila tecnológica conocida puede implementarse para resolver problemas ambientales y sociales concretos, siendo **escalable** a más sensores o variables.

ZAINO



Inicio



Informes

## Flujometro



FLUJO ACUMULADO  
**834460.94 L**

## Tiempo Actual



- °C

## Visitas Totales

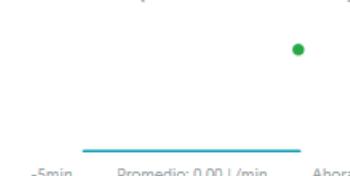
**20**

## Flujo en Tiempo Real

## CAUDAL ACTUAL

**80.26**  
L/min

## HISTORIAL (ÚLTIMOS 5 MINUTOS)



## MÁXIMO

**100.00**

## MÍNIMO

**0.00**

 Inicio Informes

## Lista de Informes

Nombre del Informe

Generar Nuevo Informe Mensual 

ⓘ Se generará un informe del último mes. Solo se permite un informe por mes.

El informe incluirá datos de caudal, estadísticas y análisis del período mensual.

Cancelar

 Generar Informe Generar Informe

Opciones

 Inicio Informes

## Lista de Informes

1 informe(s)



Informe de Caudal - Último Mes  
03/11/2025  
Último Mes

## Informe de Caudal - Último Mes

X

## Fecha de Generación

03/11/2025 02:27:53

## Período

Último Mes

## Fecha Inicio

01/10/2025

## Fecha Fin

03/11/2025

## Datos del Flujómetro



## Flujo Instantáneo

80.26

L/min



## Flujo Acumulado

834460.94

Litros



## Promedio Diario

25286.70

Litros/día

Cerrar

Descargar JSON

Generar Nuevo Informe

Generar

Ver

# Demo de la página web



<https://aguaconecta.urrahost.app>



# Conclusión: Un Futuro de Monitoreo Hídrico Conectado



## Integración Total

Zaino Web demuestra una integración fluida de IoT, backend y frontend para la monitorización hídrica esencial.

## Eficiencia y Limpieza

La arquitectura es limpia y eficiente, enfocada en el rendimiento y la exportación de informes desde el cliente.

## Escalabilidad y Impacto

El proyecto es escalable a nuevas variables y tiene un alto impacto técnico y potencial social.

## Preguntas y Contacto

Agradecemos su atención. ¿Tiene alguna pregunta sobre la implementación técnica o el potencial de expansión de Zaino Web?

Código Fuente (GitHub):

<https://github.com/IdkBemja/zaino-web>

Contacto: [contact@urrahost.app](mailto:contact@urrahost.app)

