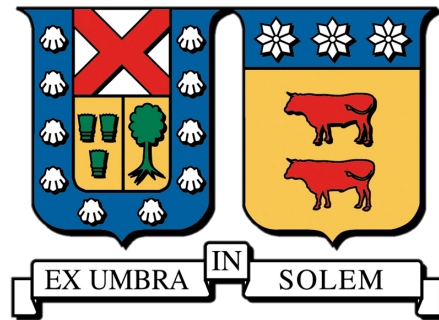


**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA**  
**DEPARTAMENTO DE ELECTRÓNICA**  
**SANTIAGO - CHILE**



**“PLATAFORMA IOT PARA EL MONITOREO DEL  
CONSUMO DE AGUA EN HOGARES”**

**ÁLVARO ALEJANDRO POZO FUENTES**

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL  
TELEMÁTICO**

**PROFESORA GUIA:**

**DRA. PATRICIA FRANCO**

**PROFESORA CORREFERENTE:**

**DRA. PATRICIA MORALES**

**ENERO - 2026**



## CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

### 1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción):  Memoria o trabajo de título  Tesis de Postgrado

Título del trabajo: Plataforma IoT para el monitoreo del consumo de agua en hogares.

Nombre del candidato(a): Álvaro Alejandro Pozo Fuentes.

Carrera / Grado: Grado de Licenciado en Ciencias de la Ingeniería, título profesional de Ingeniero Civil Telemático.

Campus: San Joaquín Departamento: Electrónica

### 2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Patricia Franco Troya, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

### 3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

6 meses  12 meses  2 años  3 años  5 años  10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

---

---

---

### 4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 23/02/2026

Firma: PATRICIA FRANCO TROYA  
Firmado digitalmente por PATRICIA FRANCO TROYA  
Fecha: 2026.02.23 10:14:12 -03'00'

Estudiante o Candidato(a):

Fecha: 23/01/2026

Firma: 

*Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.*

---

# Agradecimientos

En primera instancia, deseo expresar mi profundo agradecimiento a todas las personas que han formado parte de este proceso. Ha sido un camino largo, exigente y, en muchos momentos, complejo. Por lo mismo, llegar a esta instancia representa un logro muy significativo para mí.

Agradezco a mis padres, quienes siempre han estado presentes, brindándome apoyo, compañía y soporte en cada etapa de esta carrera.

Quiero agradecer también a mis amigos, quienes han sido un apoyo y una fuente de motivación durante estos años. De manera especial, menciono a Leonardo, Tabata, y Eduardo, por su compañía, su apoyo y todas las veces que han estado ahí cuando los necesité. A lo largo de este camino, compartimos momentos duros y también momentos llenos de alegría; y, aun así, dentro de todo, siempre logramos salir adelante.

A mi pareja, Kony Carrasco, quiero darle un agradecimiento muy especial, ya que ella ha sido un pilar fundamental en todo este proceso, acompañándome, motivándome y apoyándome incondicionalmente. Y, en realidad, más que solo compañía, ha sido un refugio, una motivación constante y una de las razones por las que este trabajo pudo llegar a término. Su presencia ha significado mucho más de lo que podría expresar en estas líneas.

Finalmente, quiero extender mi agradecimiento a todos los profesores que me han formado a lo largo de la carrera y que, con sus enseñanzas, contribuyeron de manera decisiva a mi crecimiento académico y profesional. De manera particular, agradezco a mi profesora guía, Patricia Franco, por su dedicación, orientación y apoyo constante durante el desarrollo de este trabajo. Su acompañamiento fue clave para finalizar esta memoria.

---

# Resumen

Este proyecto presenta el diseño e implementación de una plataforma IoT orientada al monitoreo inteligente del consumo de agua en hogares, con el objetivo de mejorar la eficiencia hídrica y facilitar la detección temprana de anomalías. El sistema integra un módulo de sensado encargado de obtener lecturas del medidor de agua mediante una arquitectura distribuida basada en dispositivos de borde y un servicio centralizado en la nube. A partir de estas lecturas se construye un conjunto de eventos de consumo que permite la aplicación de técnicas de análisis avanzadas, tales como métodos de desagregación de datos para estimar el consumo individual asociado a distintos sectores del hogar, y técnicas de análisis de patrones orientadas a la identificación de comportamientos anómalos en el uso del recurso hídrico.

El trabajo se centra en el desarrollo de dos componentes principales: un módulo de desagregación de consumos basado en modelos de Machine Learning, y un módulo de detección de anomalías orientado a identificar patrones inusuales asociados a fugas o comportamientos atípicos. Para la desagregación se evaluaron diversos modelos predictivos, comparando su desempeño mediante métricas como precisión, recall y F1-score, seleccionándose el algoritmo con mejor rendimiento para su integración en el pipeline operativo. Adicionalmente, se implementó un esquema híbrido de detección de anomalías que combina Isolation Forest con un sistema de lógica difusa para determinar los niveles de severidad.

El sistema completo fue desplegado en AWS, permitiendo la ejecución continua del procesamiento, la clasificación de consumos y la generación de alertas. Los resultados demuestran la viabilidad de la plataforma para apoyar la gestión del recurso hídrico en hogares, habilitando análisis detallados y detección temprana de eventos anómalos.

**Palabras Clave:** Consumo de Agua; Desagregación de Consumos; Detección de Anomalías; Machine Learning; Monitoreo Inteligente.

---

# Abstract

This project presents the design and implementation of an IoT platform for intelligent monitoring of household water consumption, with the aim of improving water efficiency and facilitating the early detection of anomalies. The system integrates a sensing module responsible for reading water meters using a distributed architecture based on edge devices and a centralized cloud service. From these readings, a set of consumption events is constructed, enabling the application of advanced analytical techniques, such as data disaggregation methods to estimate individual consumption associated with different areas of the home, and pattern analysis techniques to identify anomalous water usage patterns.

The work focuses on the development of two main components: a consumption disaggregation module based on Machine Learning models, and an anomaly-detection module designed to identify unusual patterns associated with leaks or atypical usage behaviors. Multiple predictive models were evaluated for the disaggregation task, and their performance was compared using metrics such as precision, recall, and F1-score, selecting the model with the best overall performance for operational deployment. Additionally, a hybrid anomaly-detection approach was implemented, combining Isolation Forest with a fuzzy-logic system to determine severity levels.

The complete system was deployed on AWS, enabling continuous processing, consumption classification, and real-time alert generation. The results demonstrate the feasibility of the platform as a tool to support household water-resource management, providing detailed consumption analysis and early identification of abnormal events.

**Keywords:** Anomaly Detection; Consumption Disaggregation; Machine Learning; Smart Monitoring; Water Consumption.

---

# Índice general

<b>1..Introducción</b>	1
1.1. Contextualización del Problema	1
1.1.1. Consumo de agua en hogares chilenos y su relevancia	1
1.1.2. Limitaciones de los medidores tradicionales	2
1.1.3. Problemas de accesibilidad y costo en soluciones comerciales	2
1.1.4. Dificultad para desagregar consumos y detectar anomalías	3
1.1.5. Falta de sistemas IoT económicos para usuarios finales	4
1.1.6. Necesidad de herramientas visuales predictivas para hogares	4
1.2. Propósito	5
1.3. Motivación	5
1.4. Objetivos	5
1.4.1. Objetivo General	5
1.4.2. Objetivos Específicos	6
1.4.3. Contribuciones	6
1.5. Estructura del documento	7
<b>2..Estado del Arte</b>	9
2.1. Soluciones de monitoreo de agua en entornos residenciales	9
2.1.1. Arquitectura IoT y capas del sistema	10
2.1.2. Métodos obstructivos basados en sensores de flujo	13
2.1.3. Métodos no obstructivos e IoT	14
2.2. Técnicas de desagregación del consumo	17
2.3. Datasets para desagregación	21
2.3.1. Resumen comparativo de datasets	22
2.4. Métodos de detección de anomalías	23
2.4.1. Métodos estadísticos basados en percentiles	23
2.4.2. Métodos basados en aprendizaje automático	24
2.4.3. Métodos basados en predicción futura	27

---

2.4.4.	Lógica difusa para evaluación de severidad . . . . .	28
2.5.	Síntesis del estado del arte . . . . .	29
<b>3.</b>	<b>Desagregación del Consumo de Agua . . . . .</b>	<b>32</b>
3.1.	Diseño de los módulos principales . . . . .	32
3.1.1.	Alcance de la implementación realizada . . . . .	33
3.2.	Diseño del módulo de desagregación . . . . .	33
3.3.	Preparación y construcción del dataset . . . . .	34
3.3.1.	Dataset 1: WEUSEDTO . . . . .	35
3.3.2.	Dataset 2: STREaM . . . . .	40
3.4.	Implementación del módulo de desagregación de consumos . . . . .	45
3.4.1.	Modelos de clasificación evaluados . . . . .	46
3.4.2.	Proceso de entrenamiento . . . . .	47
3.4.3.	Métricas de evaluación y comparación de modelos . . . . .	51
3.5.	Experimentos y análisis . . . . .	51
3.5.1.	Diseño experimental . . . . .	52
3.5.2.	Comparación entre datasets reales y sintéticos . . . . .	57
3.5.3.	Impacto de variaciones de características . . . . .	65
3.5.4.	Evaluación cruzada entre datasets . . . . .	68
3.5.5.	Selección final del modelo . . . . .	74
3.5.6.	Discusión de limitaciones y posibles mejoras . . . . .	75
3.6.	Integración y validación del módulo de desagregación en escenario real . . . . .	77
3.6.1.	Despliegue del módulo de desagregación en AWS . . . . .	78
3.6.2.	Flujo operativo del pipeline de desagregación . . . . .	78
3.6.3.	Validación del módulo en entorno productivo . . . . .	78
<b>4.</b>	<b>Detección de Anomalías en el Consumo . . . . .</b>	<b>82</b>
4.1.	Diseño del módulo de detección de anomalías . . . . .	82
4.2.	Implementación del módulo de detección de anomalías . . . . .	82
4.2.1.	Métodos iniciales sin Machine Learning y enfoques explorados . . . . .	83
4.2.2.	Implementación con Isolation Forest . . . . .	86

---

4.2.3.	Complemento con lógica difusa para severidad . . . . .	89
4.2.4.	Complemento con lógica difusa para evaluación de severidad . . . . .	89
4.3.	Integración y despliegue del módulo de anomalías en AWS . . . . .	94
4.3.1.	Validación del sistema en escenario real . . . . .	97
4.4.	Discusión de ventajas, limitaciones y posibles mejoras . . . . .	99
<b>5.</b>	<b>Conclusiones y Trabajos Futuros . . . . .</b>	<b>102</b>
5.1.	Conclusiones . . . . .	102
5.2.	Trabajos Futuros . . . . .	104

---

## Índice de figuras

2.1. Arquitectura IoT por capas, que muestra la separación entre dispositivos, conectividad, procesamiento y aplicaciones. . . . .	11
2.2. Ejemplo de sensor de flujo utilizado para medir el caudal de agua en soluciones de monitoreo obstructivas. . . . .	14
2.3. Principio de medición de caudal mediante sensores ultrasónicos por tiempo de tránsito, donde la diferencia entre las señales emitidas aguas arriba y aguas abajo permite estimar la velocidad del flujo. . . . .	16
2.4. Representación conceptual de una distribución de probabilidad del consumo y la ubicación de distintos percentiles (P90, P95 y P99). Los percentiles elevados permiten definir distintos niveles de umbral para la detección de consumos atípicos o extremos. . . . .	24
2.5. Esquema conceptual del funcionamiento de Isolation Forest, donde las observaciones anómalas son aisladas con un menor número de particiones en comparación con los datos normales. . . . .	25
2.6. Representación conceptual de un modelo one-class SVM, donde una frontera delimita el comportamiento normal y los puntos fuera de esta son considerados anomalías. . . . .	26
3.1. Módulos por capas . . . . .	33
3.2. Módulo de desagregación . . . . .	34
3.3. Visión general del módulo de desagregación de consumos. A partir de los eventos de consumo detectados, se emplean distintos modelos de clasificación entrenados con datasets reales y semi-sintéticos, considerando escenarios single-label y multi-label, para identificar el dispositivo responsable del consumo. . . . .	46
3.4. Arquitectura del modelo CNN utilizado para la desagregación del consumo. .	49
3.5. Arquitectura del modelo LSTM utilizado para la desagregación del consumo.	50
3.6. Ejemplo de pulso cuadrado típico en el dataset WEUSEDTO. . . . .	53

---

3.7. Consumo mensual por dispositivo en WEUSEDTO. . . . .	54
3.8. Distribución del consumo según la hora del día en WEUSEDTO. . . . .	54
3.9. Ejemplo de eventos individuales por dispositivo en el dataset STREaM. Todos los subgráficos comparten un eje temporal común (eje X), mientras que el eje Y representa el caudal instantáneo. Se observan diferencias claras en duración, intensidad del caudal y forma del pulso para lavadora, grifo, inodoro, ducha y lavavajillas. . . . .	55
3.10. Rendimiento de modelos en STREaM bajo configuración single-label con tres características. . . . .	58
3.11. Rendimiento de modelos en WEUSEDTO bajo configuración single-label con tres características. . . . .	58
3.12. Comparativa de modelos multilabel en STREaM utilizando tres características. 60	
3.13. Comparativa de modelos multilabel en WEUSEDTO utilizando tres características. . . . .	61
3.14. Desempeño de modelos multilabel con características extendidas en STREaM. 62	
3.15. Desempeño de modelos multilabel con características extendidas en WEUSEDTO. . . . .	63
3.16. Resultados en WEUSEDTO multilabel con características extendidas y balanceo de clases. . . . .	63
3.17. Resultados en STREaM multilabel con características extendidas y balanceo de clases. . . . .	64
3.18. Matriz de confusión para Random Forest entrenado en WEUSEDTO y evaluado en STREaM. . . . .	70
3.19. Métricas por clase para Random Forest (WEUSEDTO $\rightarrow$ STREaM). . . . .	70
3.20. Matriz de confusión para XGBoost entrenado en WEUSEDTO y evaluado en STREaM. . . . .	71
3.21. Métricas por clase para XGBoost (WEUSEDTO $\rightarrow$ STREaM). . . . .	71
3.22. Matriz de confusión para Random Forest entrenado en STREaM y evaluado en WEUSEDTO. . . . .	72
3.23. Métricas por clase para Random Forest (STREaM $\rightarrow$ WEUSEDTO). . . . .	72

---

3.24. Matriz de confusión para XGBoost entrenado en STREaM y evaluado en WEUSEDTO. . . . .	73
3.25. Métricas por clase para XGBoost (STREaM → WEUSEDTO). . . . .	73
3.26. Arquitectura de integración del módulo de desagregación en AWS y flujo de datos hacia el backend y el dashboard. . . . .	77
3.27. Lecturas reales de caudal recibidas y almacenadas en la instancia EC2. . . . .	79
3.28. Eventos de consumo generados y predicciones del módulo de desagregación. . . . .	79
3.29. Registro del envío de eventos desagregados al backend de la plataforma IoT. . . . .	80
3.30. Visualización del consumo total y desagregado por dispositivo en el dashboard de la plataforma, a partir de datos reales procesados por el módulo de desagregación. . . . .	80
4.1. Módulo de detección de anomalías . . . . .	82
4.2. Distribución empírica del volumen total consumido por dispositivo en el dataset unificado STREaM. . . . .	84
4.3. Esquema general del proceso de detección de anomalías mediante Isolation Forest, donde los eventos desagregados son evaluados en función de su duración y volumen total para ser clasificados como normales o anómalos antes de su envío al módulo de severidad. . . . .	87
4.4. Detección de anomalías en el espacio duración–litros utilizando Isolation Forest (dispositivo: grifo). . . . .	89
4.5. Esquema del módulo de lógica difusa para la evaluación de severidad de eventos anómalos. . . . .	91
4.6. Funciones de membresía triangulares para la entrada <code>score_anomalia</code> . . . . .	92
4.7. Funciones de membresía triangulares para la salida <code>severidad_out</code> . . . . .	93
4.8. Consumo instantáneo en el tiempo. Cada rectángulo representa un evento de consumo; los eventos en rojo son clasificados como anómalos y los amarillos como normales. . . . .	94
4.9. Comparación entre consumo real y consumo esperado sin anomalías. . . . .	94

---

4.10. Esquema de la arquitectura y flujo de datos del módulo de detección de anomalías desplegado en AWS. . . . .	95
4.11. Salida generada por el pipeline de procesamiento en la instancia EC2, mostrando eventos desagregados y resultados del módulo de detección de anomalías durante la operación real del sistema. . . . .	98
4.12. Visualización del consumo instantáneo y eventos procesados durante un período de operación real del sistema, donde no se detectan anomalías. . . . .	98
5.1. Propuesta de solución . . . . .	105
5.2. Modelo de capas . . . . .	106
5.3. Diagrama de contexto . . . . .	109
5.4. Módulo de sensado . . . . .	111
5.5. Módulo de aplicación web . . . . .	112
5.6. Flujo de los módulos . . . . .	113
5.7. Modelo de navegación . . . . .	114
5.8. Página principal . . . . .	115
5.9. Listado de notificaciones . . . . .	116
5.10. Visualización de consumo de agua . . . . .	116

---

## Índice de tablas

2.1. Resumen comparativo de tecnologías de comunicación empleadas en arquitecturas IoT para monitoreo residencial, diferenciando niveles LAN y WAN. . . . .	12
2.2. Resumen comparativo de datasets para desagregación del consumo de agua . . . . .	22
2.3. Comparación de características entre trabajos de desagregación, detección de anomalías y la presente memoria. . . . .	31
3.1. Configuración de parámetros utilizados en los modelos basados en árboles . . . . .	48
3.2. Métricas de desempeño para el escenario single-label con tres características utilizando el dataset STREaM. . . . .	59
3.3. Métricas de desempeño para el escenario single-label con tres características utilizando el dataset WEUSEDTO. . . . .	59
3.4. Métricas de desempeño para el escenario multi-label con tres características utilizando el dataset STREaM. . . . .	61
3.5. Métricas de desempeño para el escenario multi-label con tres características utilizando el dataset WEUSEDTO. . . . .	62
4.1. Cantidad de eventos por dispositivo en el dataset unificado STREaM. . . . .	85
4.2. Estadísticas descriptivas por dispositivo. . . . .	86

# CAPÍTULO 1: INTRODUCCIÓN

---

## 1.1 Contextualización del Problema

La gestión del agua se ha convertido en un desafío relevante a nivel mundial, particularmente en el ámbito residencial, donde el aumento del consumo y la escasez del recurso han intensificado la necesidad de una gestión más eficiente. Estudios recientes destacan que una parte significativa del consumo doméstico corresponde a usos ineficientes o no intencionados, los cuales se ven agravados por la ausencia de sistemas de monitoreo continuo y de retroalimentación hacia los usuarios. En la mayoría de los hogares, el consumo de agua es medido de forma agregada y con baja frecuencia, lo que dificulta identificar comportamientos específicos de uso y retrasa la detección de eventos anómalos, como fugas o consumos prolongados, generando impactos económicos y ambientales significativos [1, 2].

Este proyecto busca enfrentar dicha problemática, desarrollando una plataforma IoT capaz de monitorear el consumo de agua en tiempo real. La propuesta consiste en instalar un sensor no obstructivo basado en visión por computadora que capture la lectura del medidor analógico tradicional. Con esto, es posible proporcionar al usuario información precisa y continua, desagregar el consumo por tipo de uso y generar alertas automáticas cuando se detecten patrones anormales. El objetivo es acercar tecnologías de eficiencia hídrica a cualquier hogar sin necesidad de reemplazar la infraestructura existente.

A continuación se detallan los aspectos más relevantes de este problema para el contexto Chileno y de las técnicas a utilizar.

### 1.1.1 Consumo de agua en hogares chilenos y su relevancia

Chile ha enfrentado una crisis hídrica severa durante más de una década, conocida como la "megasequía", que afectó gran parte del país principalmente entre los años 2010 y 2020. Este periodo marcó una disminución sostenida en las precipitaciones y en la disponibilidad de agua superficial, situando a varias regiones en niveles críticos. Según la Superintendencia de Servicios Sanitarios (SISS), el consumo promedio en zonas urbanas varía entre 130 y 170 litros diarios por persona, dependiendo de la región y del nivel socioeconómico [1].

Comprender y monitorear este consumo es clave, ya que existe una brecha entre la dispo-

nibilidad del recurso y la demanda residencial. Durante el verano, por ejemplo, el consumo puede aumentar en torno a un 30 % por riego y actividades recreativas, lo que pone aún más presión sobre la red de distribución. A esto se suma que la mayoría de las personas solo conocen su nivel de consumo cuando reciben la boleta mensual. Diversos estudios muestran que, al entregar retroalimentación inmediata, el consumo puede disminuir entorno a un 5 % simplemente porque el usuario toma conciencia [4]. En un contexto como el chileno, donde el agua es cada vez más costosa y escasa, no contar con herramientas de control es un riesgo tanto económico como ambiental.

### 1.1.2 Limitaciones de los medidores tradicionales

La mayoría de los hogares en Chile utiliza medidores mecánicos. Estos funcionan bien para la facturación, pero presentan varias limitaciones para una gestión eficiente:

1. **Baja frecuencia de medición:** Normalmente, se leen una vez al mes, lo que impide analizar consumos diarios u horarios [1].
2. **Proceso manual:** La lectura depende de una visita, por lo que es susceptible a errores humanos, estimaciones o retrasos [1].
3. **No detectan eventos:** Si ocurre una fuga o si un artefacto queda funcionando, el medidor simplemente registra el paso del agua, pero no informa al usuario. Muchas veces el problema se descubre solo al recibir una boleta con sobreconsumo [1].

### 1.1.3 Problemas de accesibilidad y costo en soluciones comerciales

Aunque existen tecnologías modernas como los medidores inteligentes (AMI), su adopción masiva sigue siendo limitada y poco accesible para los usuarios residenciales. La mayoría de estas soluciones está orientada a empresas sanitarias y no a hogares particulares, lo cual dificulta el uso doméstico.<sup>1</sup>

El reemplazo de los medidores mecánicos tradicionales por dispositivos ultrasónicos o electromagnéticos implica una inversión alta, que generalmente termina traspasándose al

---

<sup>1</sup> Véase <https://www.ibm.com/mx-es/think/topics/advanced-metering-infrastructure>.

usuario final. Además, varios sistemas comerciales utilizan arquitecturas cerradas que restringen el acceso directo a los datos en tiempo real. Tal como señalan en la investigación [5], la implementación de medidores inteligentes enfrenta barreras relevantes asociadas al costo de adquisición, el tiempo de instalación y problemas de compatibilidad tecnológica, factores que complican su despliegue a gran escala. Por otra parte, muchas alternativas disponibles para instalación en hogares requieren intervenir la red de agua (cortar o modificar tuberías), lo cual aumenta los costos y genera riesgos [3].

### 1.1.4 Dificultad para desagregar consumos y detectar anomalías

Uno de los desafíos más complejos es la desagregación del consumo, conocida como NILM (Non-Intrusive Load Monitoring) aplicada al agua. A diferencia de la electricidad, donde cada equipo presenta una "firma" particular, los patrones de flujo de agua son mucho más parecidos entre sí [24].

- **El problema del consumo agregado:** El medidor principal solo registra el volumen total que ingresa al hogar. Sin algoritmos avanzados, es imposible distinguir, por ejemplo, si 50 litros consumidos corresponden a una ducha, al llenado de una lavadora o al uso del lavamanos [23].
- **Detección de anomalías:** Las fugas pequeñas pueden pasar desapercibidas e incluso no ser detectadas por medidores antiguos que no registran flujos pequeños [1]. Además, para identificar comportamientos inusuales, es importante definir criterios claros. En este proyecto, una anomalía será considerada principalmente cuando un consumo se prolongue en el tiempo o presente un volumen de litros inusualmente alto [28].

Una alternativa para abordar este desafío es el uso de técnicas de análisis de datos y algoritmos de aprendizaje automático (Machine Learning), lo que permite inferir el tipo de dispositivo asociado al consumo basándose en la velocidad del flujo, la duración de cada evento y otros atributos derivados. Este nivel de detalle y capacidad de interpretación no existe en los sistemas de medición tradicionales.

### 1.1.5 Falta de sistemas IoT económicos para usuarios finales

Hoy existe un vacío en el mercado de soluciones del Internet de las Cosas (IoT). En el contexto residencial, la mayoría de estas soluciones requiere hardware costoso o redes especializadas, lo que limita su adopción [13].

La propuesta de utilizar una cámara para leer el medidor existente elimina la necesidad de intervenir las tuberías y reduce significativamente los costos [16]. Sin embargo, este tipo de dispositivos aún no es común en el mercado, por lo que muchos hogares siguen dependiendo de lecturas manuales o estimaciones [1].

### 1.1.6 Necesidad de herramientas visuales predictivas para hogares

Finalmente, solo entregar datos en bruto no es suficiente para generar cambios en el comportamiento del usuario. Para que la información sea útil, es fundamental transformarla en visualizaciones que sean claras y que permitan comprender rápidamente el consumo diario y detectar situaciones anómalas [4].

En un contexto residencial, las personas necesitan plataformas que presenten la información de forma simple y directa, permitiendo interpretar el uso del agua sin conocimientos técnicos. En este sentido, una plataforma de monitoreo orientada a hogares debería cumplir con tres características principales:

1. **Intuitiva:** Mostrar el consumo histórico y en tiempo real mediante gráficos simples y fáciles de entender.
2. **Informativa:** Destacar patrones de uso relevantes o cambios bruscos en el comportamiento del consumo.
3. **Proactiva:** Notificar automáticamente cuando se detecten consumos prolongados o volúmenes inusuales que puedan indicar fugas.

A partir del análisis de las soluciones existentes y del alcance de la revisión desarrollada en este trabajo, no se observa la presencia de una solución accesible que combine un sensor

económico y no obstructivo. Con un procesamiento de datos en tiempo real y una visualización clara dirigida al usuario final. Todo esto es precisamente lo que este proyecto busca cubrir, integrando hardware, análisis y visualización en una sola plataforma.

### **1.2 Propósito**

El propósito de este proyecto es desarrollar una solución tecnológica que permita monitorear de manera inteligente y no obstructiva el consumo de agua en hogares, con el fin de mejorar la eficiencia hídrica y apoyar una gestión más informada del recurso. Para ello, se plantea la construcción de un sistema capaz de obtener lecturas continuas del medidor de agua, procesarlas y transformarlas en información útil para el usuario final.

### **1.3 Motivación**

La motivación de esta iniciativa surge ante la creciente necesidad de contar con herramientas que entreguen visibilidad sobre los patrones de consumo en los hogares, y que permitan identificar comportamientos anómalos que puedan generar pérdidas económicas o desperdicio del recurso. A partir de los datos sensados, el sistema busca realizar análisis avanzados mediante técnicas de desagregación y modelos de detección de anomalías, entregando así una plataforma que contribuya tanto a la sostenibilidad ambiental como a la toma de decisiones informadas dentro del hogar.

### **1.4 Objetivos**

#### **1.4.1 Objetivo General**

Diseño e implementación de una plataforma IoT que permita monitorear en tiempo real el consumo de agua en un hogar mediante técnicas que no interrumpan el comportamiento habitual de los usuarios.

### 1.4.2 Objetivos Específicos

1. **Diseñar la arquitectura del sistema** permitiendo el monitoreo en tiempo real del consumo de agua en un hogar, considerando los distintos módulos necesarios para la adquisición, procesamiento y visualización de la información.
2. **Obtener el consumo de agua por sector en el hogar mediante la desagregación del consumo total** aplicando técnicas de desagregación sobre los datos recolectados por el sensor basado en visión por computadora, con el objetivo de estimar el consumo de agua asociado a cada sector del hogar y permitir al usuario conocer el uso del recurso en dichos puntos.
3. **Detectar patrones anómalos en el consumo de agua para la generación automática de alertas** mediante el análisis estadístico de patrones históricos de consumo, con el fin de informar al usuario ante la presencia de anomalías como fugas o incrementos inusuales en el uso del recurso hídrico.
4. **Desarrollar una interfaz web amigable orientada al usuario** mostrando información de consumo en tiempo real e histórico, con el propósito de entregar al usuario una herramienta visual que facilite la comprensión de sus patrones de utilización de agua.

### 1.4.3 Contribuciones

Las principales contribuciones de este trabajo se pueden resumir en los siguientes puntos:

1. **Elaboración y procesamiento de un dataset estructurado a partir de eventos de consumo reales y simulados.** Se desarrolló un pipeline de preprocesamiento que transforma las lecturas crudas del medidor en eventos de consumo, incorporando características relevantes como volumen, duración, caudal y patrones temporales. Además, se integraron datos sintéticos provenientes de modelos estocásticos para enriquecer el proceso de entrenamiento y evaluación.
2. **Implementación y comparación de modelos de desagregación de consumo basados en Machine Learning.** Se evaluaron distintos algoritmos supervisados, comparando su

rendimiento mediante métricas como precisión, recall y F1-score. Este análisis sistemático permitió seleccionar el modelo con el mejor desempeño para su uso en un entorno operativo.

3. **Desarrollo de un módulo de detección de anomalías.** Se implementó un esquema combinado que integra Isolation Forest con un sistema de lógica difusa para estimar niveles de severidad. Esta contribución permite la identificación temprana de patrones de consumo atípicos con mayor robustez.

### 1.5 Estructura del documento

A continuación se presenta la organización del presente informe, en donde se indica de manera breve el contenido de cada capítulo:

- **Capítulo 1 Introducción:** En este capítulo se presenta el contexto general del problema abordado, destacando la relevancia del monitoreo del consumo de agua en el contexto chileno. Se describen las principales limitaciones de los sistemas tradicionales, la motivación del proyecto, su propósito, los objetivos planteados y las contribuciones esperadas, estableciendo así el marco conceptual que sustenta el desarrollo de la investigación.
- **Capítulo 2 Estado del arte:** En este capítulo se revisan los principales trabajos relacionados con el monitoreo inteligente del consumo hídrico, la desagregación de uso de agua y los métodos más utilizados para la detección de anomalías. Aquí se describen las principales técnicas para cada una de las tecnologías utilizadas.
- **Capítulo 3 Desagregación del consumo de agua:** Aquí se describe el módulo encargado de indicar el tipo de dispositivo hídrico utilizado a partir de los datos obtenidos desde el medidor. Se describe la preparación de los datasets, el preprocesamiento aplicado, los modelos evaluados, el diseño experimental y los resultados obtenidos. Este capítulo corresponde a uno de los módulos centrales del trabajo.
- **Capítulo 4 Detección de anomalías en el consumo:** En este capítulo se muestra el módulo que está orientado a identificar los patrones anómalos del consumo de agua,

ya sean consumos excesivos o fugas. Se muestran los modelos implementados, los experimentos realizados y los resultados obtenidos. Este apartado constituye el segundo módulo principal de la memoria.

- **Capítulo 5 Conclusiones y trabajos futuros:** Aquí se concluye acerca de los principales resultados obtenidos a lo largo de la investigación, destacando cuales fueron los aportes correspondientes de los módulos desarrollados. Además, trabajos futuros orientados a mejorar el desempeño y robustez del sistema.

# CAPÍTULO 2: ESTADO DEL ARTE

---

El monitoreo del consumo de agua de forma inteligente en entornos tanto residenciales como urbanos ha crecido de forma importante en los últimos años gracias a la integración de tecnologías IoT, sensores de bajo costo y plataformas en la nube. Estos sistemas permiten convertir lecturas puntuales y manuales del medidor en un seguimiento continuo del comportamiento de consumo, lo que abre la puerta a estrategias de eficiencia hídrica mediante una mejor gestión del recurso, la detección de patrones anómalos e incluso la detección temprana de fugas [33].

Los organismos internacionales han advertido desde hace ya décadas la necesidad de contar con información más precisa sobre el uso del agua y la relación entre consumo y sostenibilidad, destacando la importancia de herramientas que permitan monitorear, analizar y gestionar este recurso de forma más eficiente [6]. Sin embargo, en la actualidad sigue predominando el uso de medidores analógicos que solo entregan una lectura acumulada y que, en general, se leen de forma mensual. Esto limita la visibilidad que tiene el usuario sobre su propio consumo y dificulta la detección de anomalías o usos excesivos [1].

Entonces es en este escenario, en que los sistemas de monitoreo inteligente basados en IoT cobran relevancia, llenando este vacío mediante la captura de datos con mayor frecuencia, su envío mediante redes de comunicación inalámbrica y su procesamiento en plataformas que permiten generar indicadores, visualizaciones y alertas [13]. Este proyecto se sitúa precisamente en esta línea de trabajo, con el foco puesto en soluciones no obstructivas instaladas en el medidor general del hogar.

## 2.1 Soluciones de monitoreo de agua en entornos residenciales

Las soluciones de monitoreo de consumo de agua se pueden agrupar en dos grandes áreas:

- **Métodos obstructivos:** Los cuales requieren intervenir las tuberías e instalar sensores de flujo.[22]
- **Métodos no obstructivos:** Los cuales buscan sensar los datos sin modificar la infraestructura hidráulica existente ni intervenir con el comportamiento habitual de los usuarios.[23]

Independientemente del tipo de sensado empleado, las soluciones modernas de monitoreo de consumo de agua se implementan típicamente como sistemas IoT, en los cuales la captura de datos, su transmisión y su procesamiento forman parte de un flujo integrado. En este contexto, el tipo de sensado utilizado (obstructivo o no obstructivo) condiciona directamente el diseño de la arquitectura del sistema, los requerimientos de comunicación y las capacidades de análisis de datos que pueden implementarse posteriormente.

Por esta razón, es necesario describir primero el marco general de una arquitectura IoT, identificando sus principales capas y funciones, para luego analizar cómo distintos métodos de sensado y tecnologías de comunicación se integran dentro de dicha arquitectura en aplicaciones residenciales de monitoreo de agua.

### 2.1.1 Arquitectura IoT y capas del sistema

Para diferenciar sensores, tecnologías de comunicación y plataformas, es útil describir una solución IoT mediante una arquitectura por capas, tal como se muestra en la Figura 2.1. En general, una arquitectura IoT considera: una capa de **dispositivos** encargada de la captura de datos; una capa de **conectividad** responsable del transporte de la información, y una **plataforma IoT** donde los datos son almacenados, procesados y presentados mediante indicadores, visualizaciones y alertas [34].

En este esquema, los **medidores inteligentes** operan principalmente en la capa de **dispositivos**, ya que integran el proceso de medición y, en muchos casos, un módulo de comunicaciones para transmitir lecturas hacia una plataforma centralizada [11, 33].

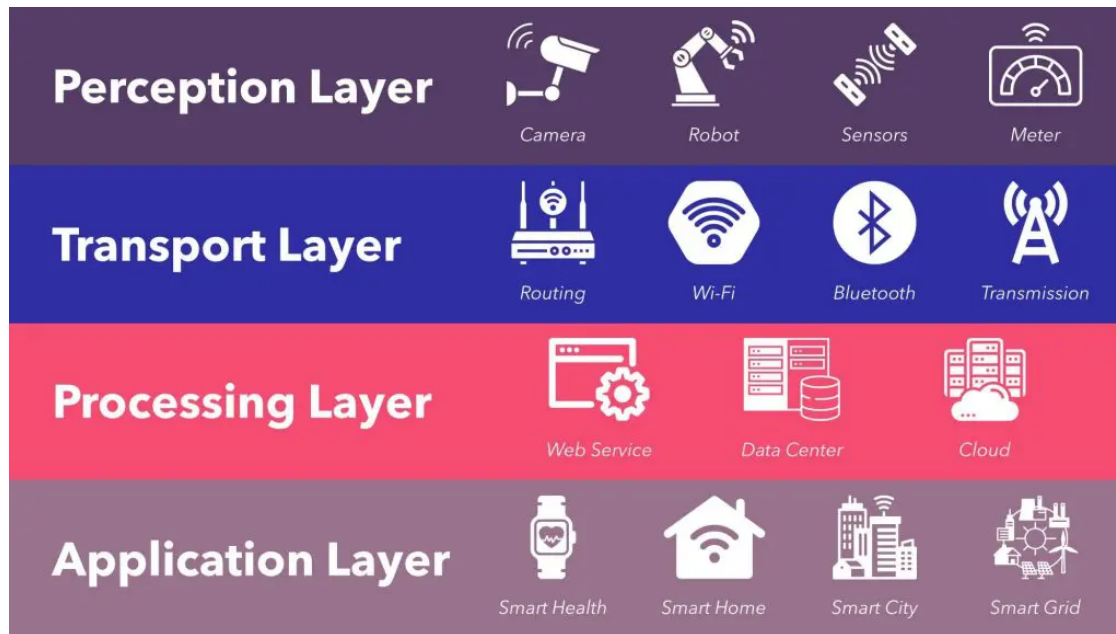


Fig. 2.1: Arquitectura IoT por capas, que muestra la separación entre dispositivos, conectividad, procesamiento y aplicaciones.

Fuente: <https://dgtlinfra.com/internet-of-things-iot-architecture/>

### Comunicación de datos y plataformas IoT

Las soluciones no obstructivas, suelen tener una arquitectura IoT que combina un nodo de sensado local, un medio de comunicación inalámbrico y una plataforma de procesamiento en la nube. En este contexto, se han utilizado distintos protocolos y tecnologías de comunicación según el alcance, el consumo energético y la disponibilidad de infraestructura [34, 21].

En la investigación [13] se destaca el uso de protocolos como Zigbee y LoRaWAN para entornos de monitoreo distribuido, mientras que en [14] y [15] profundizan en aplicaciones específicas de LoRaWAN, Sigfox y otras tecnologías LPWAN en la gestión de agua. En el caso de soluciones que son residenciales, también se considera el uso de WiFi combinado con protocolos de mensajería como MQTT, aprovechando la infraestructura doméstica existente y facilitando la integración con servicios en la nube [19, 20, 21].

Desde el punto de vista arquitectónico, la comunicación en sistemas de monitoreo residencial puede dividirse en dos niveles los cuales son una red local (LAN), correspondiente al entorno del hogar, y una red de área amplia (WAN), encargada de transportar la información hacia servicios externos en la nube. En la capa LAN, tecnologías como WiFi, Zigbee

o Bluetooth Low Energy permiten la comunicación entre el nodo de sensado y un gateway local, aprovechando la infraestructura doméstica existente. En la capa WAN, tecnologías como LoRaWAN, Sigfox o redes celulares facilitan el envío de datos a plataformas remotas, priorizando bajo consumo energético y amplio alcance [13, 14, 15].

La elección de la tecnología de comunicación depende del alcance, consumo energético, latencia y disponibilidad de infraestructura. Mientras que soluciones basadas en WiFi y MQTT ofrecen baja latencia y facilidad de integración en entornos residenciales, tecnologías LPWAN como LoRaWAN o Sigfox son más adecuadas en escenarios donde se requiere bajo consumo energético y transmisión a larga distancia [14, 15, 21]. En todos los casos, el objetivo común es recopilar datos del medidor y transmitirlos de forma confiable para habilitar etapas posteriores de visualización, desagregación del consumo y detección de anomalías. Este proyecto se centra específicamente en el análisis de los datos ya digitalizados, independientemente de la tecnología de comunicación empleada.

A continuación se tiene una tabla resumen con cada una de las tecnologías y sus características:

Tecnología	Nivel	Alcance típico	Consumo	Ventajas	Limitaciones / uso típico
WiFi (802.11)	LAN	10–30 m	Medio/alto	Alta tasa de datos; infraestructura doméstica existente; integración simple con MQTT/HTTP	No ideal para baterías; depende del router del hogar. Adecuado cuando hay energía disponible y se requiere baja latencia.
Zigbee (802.15.4)	LAN	10–100 m (malla)	Bajo	Soporte mesh; bajo consumo; robusto para redes de sensores	Requiere gateway; menor tasa que WiFi. Adecuado para múltiples sensores distribuidos.
Bluetooth Low Energy (BLE)	LAN	5–30 m	Muy bajo	Muy bajo consumo	Alcance limitado; requiere gateway o teléfono para salida a Internet. Útil para configuración local.
LoRaWAN (LPWAN)	WAN	2–15 km	Muy bajo	Gran alcance; excelente autonomía; ideal para telemetría periódica	Baja tasa y mayor latencia; depende de cobertura LoRaWAN. Útil sin WiFi disponible.
Sigfox (LPWAN)	WAN	3–10 km	Muy bajo	Operación simple; consumo mínimo	Dependiente del operador; límite de mensajes diarios. Adecuado para envíos muy esporádicos.
Celular (NB-IoT / LTE-M / 2G–4G)	WAN	Cobertura operador	Bajo–medio	Conectividad directa a nube; amplia cobertura	Costos de SIM y plan; mayor consumo que LPWAN. Usado en medidores inteligentes comerciales.
Ethernet (cable)	LAN	Hogar/edificio	Bajo	Alta estabilidad; baja latencia; alta confiabilidad	Requiere cableado; poco práctico en medidores residenciales. Útil en gateways fijos.

Tab. 2.1: Resumen comparativo de tecnologías de comunicación empleadas en arquitecturas IoT para monitoreo residencial, diferenciando niveles LAN y WAN.

La información comparativa se basa en revisiones y documentación técnica sobre arquitecturas IoT, tecno-

logías LPWAN y protocolos de mensajería aplicados al monitoreo de agua. [13, 14, 15, 21, 19, 20]

Ya teniendo en mente como funciona lo anteriormente visto, a continuación se revisan los principales enfoques de sensado utilizados en monitoreo residencial, diferenciando entre soluciones obstructivas y no obstructivas.

### 2.1.2 Métodos obstructivos basados en sensores de flujo

Los métodos obstructivos se basan en utilizar sensores de flujo que son instalados directamente en la tubería. Un enfoque típico consiste en implementar estos sensores por punto de consumo, donde cada artefacto o zona del hogar (ducha, lavamanos, lavadora, cocina, etc.) cuenta con un sensor de flujo dedicado. En [7] se propone, por ejemplo, un prototipo de medición de consumo de agua en entornos residenciales utilizando un sensor de caudal bajo el estándar IEEE 802.15.4, con registro de la información en la nube. Este tipo de solución permite obtener datos de forma detallada por punto, pero a cambio de una modificación importante en la red hidráulica.

Un ejemplo representativo de este tipo de dispositivos corresponde a los sensores de flujo instalados en línea, comúnmente utilizados en prototipos IoT de bajo costo para la medición de caudal, como el ilustrado en la Figura 2.2, cuya imagen se utiliza únicamente con fines ilustrativos<sup>1</sup>.

---

<sup>1</sup> <https://arduino.cl/producto/sensor-de-flujo-de-agua/>



*Fig. 2.2:* Ejemplo de sensor de flujo utilizado para medir el caudal de agua en soluciones de monitoreo obstructivas.

Dentro de esta categoría hay distintos sensores de efecto Hall como el YF-S201, los cuales son ampliamente utilizados por su bajo costo y facilidad de integración con microcontroladores [8]. Estos sensores permiten medir el caudal en litros por minuto a partir de la velocidad de giro de una turbina que tienen de forma interna. Si bien estos dispositivos ofrecen buena precisión, su instalación implica cortar tuberías, adaptar conexiones y, en muchos casos, realizar modificaciones permanentes en la infraestructura.

En aplicaciones de calidad de agua también se han estudiado sensores multiparamétricos de bajo costo para medir variables como pH, turbidez o conductividad, los cuales pueden integrarse a plataformas IoT [10]. No obstante, desde el punto de vista de un usuario residencial, el despliegue masivo de soluciones obstructivas presenta desventajas claras: costo de instalación, necesidad de intervención en la red de agua y mayor complejidad de mantenimiento.

### **2.1.3 Métodos no obstructivos e IoT**

Debido a las grandes limitaciones de los métodos obstructivos, han surgido soluciones que son no obstructivas, las cuales buscan medir el consumo sin intervenir físicamente las

tuberías. Dentro de esta categoría se incluyen dispositivos como los medidores inteligentes, sensores ultrasónicos y soluciones basadas en visión por computador [9, 16, 33].

### **Medidores inteligentes**

En países como Estados Unidos, Reino Unido o Australia ya existen medidores inteligentes comerciales los cuales permiten la medición y transmisión casi en tiempo real del consumo total de agua utilizando redes celulares o LPWAN [11]. Estos dispositivos reemplazan al medidor tradicional por uno digital, los cuales son capaces de registrar el caudal con mayor resolución temporal y enviar los datos a la empresa sanitaria o a una plataforma centralizada.

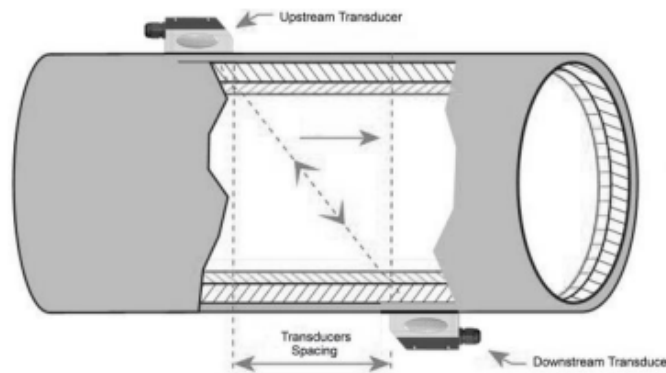
En distintos estudios se han analizado arquitecturas de monitoreo de redes de agua basadas en IoT, donde se integran módulos de comunicación de bajo consumo y algoritmos de análisis para la detección de fugas y la optimización del consumo [12, 13, 14, 15]. En [12], por ejemplo, se propone GraphSmart, el cual es una arquitectura que combina múltiples sensores y técnicas de procesamiento distribuido para lograr monitoreo eficiente y con bajo consumo energético, pensada para despliegues a gran escala. Además en [13] se revisan tecnologías IoT aplicadas a la gestión y control inteligente del agua, en donde hacen énfasis en la importancia de la interoperabilidad y de plataformas modulares y escalables que puedan adaptarse a distintos contextos (residencial, agrícola, industrial).

En las investigaciones también muestran un gran interés creciente por el uso de LoRaWAN para la gestión inteligente del agua, debido a su bajo consumo energético y gran alcance, lo que lo hace adecuado para escenarios donde no se dispone de WiFi o en donde la cobertura es limitada [14]. También se han explorado arquitecturas que integran sensores multiparamétricos con tecnologías como Sigfox, Zigbee y LoRaWAN, enfatizando la eficiencia energética y la necesidad de algoritmos avanzados para el análisis de datos y la detección temprana de anomalías [15].

A pesar de estas ventajas, en Chile el uso de medidores inteligentes aún está limitada por factores regulatorios, de infraestructura y de costo [11]. Esto abre espacio para soluciones alternativas basadas en dispositivos no obstructivos de menor costo, instalados sobre los medidores existentes.

### Sensores ultrasónicos

Los sensores ultrasónicos permiten estimar el caudal y la velocidad del flujo de agua sin necesidad de cortar tuberías, lo que los convierte en una alternativa no obstructiva desde el punto de vista estructural. Estos dispositivos operan midiendo el tiempo de tránsito de ondas ultrasónicas emitidas entre transductores ubicados aguas arriba y aguas abajo del flujo, donde la diferencia de tiempos permite inferir la velocidad del fluido y, en consecuencia, el caudal [9]. Este principio de funcionamiento se ilustra en la Figura 2.3.



*Fig. 2.3:* Principio de medición de caudal mediante sensores ultrasónicos por tiempo de tránsito, donde la diferencia entre las señales emitidas aguas arriba y aguas abajo permite estimar la velocidad del flujo.

A pesar de estas ventajas, los sensores ultrasónicos presentan desventajas importantes para su uso en entornos residenciales. Su instalación requiere una configuración inicial precisa, ya que los transductores deben ubicarse bajo ángulos y separaciones específicas para garantizar una correcta propagación de la señal. Además, estos sistemas son altamente sensibles al tipo de tubería (PVC, PPR, cobre) y al espesor del material, lo que obliga a realizar procesos de calibración cuidadosos y, en algunos casos, periódicos. Estas limitaciones complican su uso en viviendas y reducen su confiabilidad en escenarios reales, especialmente cuando se busca una solución de bajo costo y fácil instalación [9].

### Sistemas basados en visión por computador

Dentro de los métodos no obstructivos, uno de los enfoques más relevantes para entornos residenciales es el uso de visión por computador para digitalizar la lectura de medidores

analógicos. En vez de reemplazar el medidor, se instala una cámara frente al display y se aplica procesamiento de imágenes para extraer los dígitos.

En [16] se presenta un sistema de lectura automática de medidores de agua que utiliza una cámara Raspberry Pi y técnicas de reconocimiento óptico de caracteres (OCR) para identificar los dígitos del medidor y enviar los datos a la nube. De manera similar, el trabajo descrito en [17] emplea un módulo ESP32-CAM junto con una carcasa impresa en 3D para capturar imágenes estables del medidor, segmentar la región de interés y reconocer los dígitos mediante visión por computador, proponiendo una solución de bajo costo y adaptable a escenarios reales.

Además, se han explorado modelos de detección como YOLOv4 para identificar las zonas del medidor que contienen información numérica, logrando altos niveles de precisión (mAP superiores al 97 %) incluso con diferentes niveles de iluminación y orientación [18]. Este tipo de soluciones suelen seguir un flujo de tres etapas:

1. Detección de la región de interés.
2. Corrección geométrica o de perspectiva para alinear la imagen.
3. Reconocimiento de dígitos mediante OCR o redes convolucionales.

Este tipo de enfoques es especialmente útil en un contexto en donde predominan los medidores analógicos y donde el reemplazo masivo por medidores inteligentes no es factible a corto plazo. La visión por computador permite extender la funcionalidad de los medidores actuales de forma no invasiva, con un costo relativamente bajo y sin modificar la infraestructura hidráulica.

### **2.2 Técnicas de desagregación del consumo**

La desagregación del consumo de agua, también conocida como Non-Intrusive Water Disaggregation (NIWD), es un área de investigación que busca identificar el uso asociado a cada dispositivo del hogar a partir del consumo total registrado por un único medidor [45]. Al igual que en NILM para energía [36], el objetivo es inferir el uso de dispositivos a partir de patrones característicos de caudal, duración, frecuencia y volumen.

Diversos trabajos han abordado este problema con diferentes métodos. Por ejemplo [23] se presenta una herramienta de desagregación semisupervisada que utiliza un conjunto optimizado de características extraídas de cada evento, tales como el caudal promedio, caudal máximo (peak flow rate), caudal modal, duración del evento y volumen total consumido. Mediante el algoritmo Boruta, los autores identifican que las variables de mayor relevancia para la clasificación son principalmente el volumen, la duración y el caudal modal.

Una vez seleccionadas las características, el trabajo evalúa diversos modelos supervisados, incluyendo QDA, MLP, AdaBoost, Random Forest, Gaussian Naive Bayes, CART, GPC y SVM. Los clasificadores Random Forest, CART y QDA alcanzan las mayores precisiones promedio, superando el 90 %, mientras que métodos como AdaBoost presentan rendimientos más bajos. Estos resultados confirman que, cuando las ventanas de consumo están correctamente segmentadas y las características son lo suficientemente discriminativas, la desagregación puede identificar correctamente artefactos como ducha, inodoro, lavamanos, grifos, bañera, entre otros [23].

Otro de los trabajos lleva a cabo el aprendizaje profundo, en [25] se analizan técnicas de desagregación utilizando arquitecturas como SimpleGRU, WGRU, SAED y NFED. Estos modelos fueron entrenados utilizando el dataset WEUSEDTO, logrando identificar dispositivos como duchas, bidet, lavamanos, llaves de cocina y lavadoras. En el trabajo se aprecia que la similitud de ciertos dispositivos representa un desafío significativo, pero demuestra que las redes neuronales recurrentes y los modelos híbridos pueden capturar de forma efectiva la secuencia temporal del caudal.

El estudio de [26] demuestra que incluso con datos de medidores inteligentes de baja resolución (intervalos de 15 a 60 minutos) es posible obtener información útil sobre los patrones generales de uso del agua. Para ello, los autores entrenan un modelo de *stacking* basado solo en el volumen y el tiempo, utilizando más de cinco millones de eventos previamente medidos a alta resolución. Si bien este enfoque no permite identificar eventos individuales ni realizar desagregación detallada por vivienda, sí entrega estimaciones confiables de la distribución de usos cuando se analizan grupos de hogares y periodos más largos, lo que lo convierte en una herramienta válida para estudios poblacionales y análisis de demanda.

Estas investigaciones refuerzan que la desagregación del consumo es un problema com-

plejo debido a la similitud hidráulica entre dispositivos, la superposición de eventos y la variabilidad del comportamiento humano. No obstante, los avances en técnicas semisupervisadas, aprendizaje automático y modelos de Deep Learning han permitido desarrollar herramientas cada vez más robustas para estimar el consumo desagregado utilizando solo la señal total del medidor [23, 25, 26].

Además de los trabajos previamente revisados, es importante considerar avances provenientes de la desagregación no intrusiva de cargas eléctricas (NILM), ya que este problema es conceptualmente similar a la desagregación del consumo de agua (NIWD). En NILM, la potencia activa agregada medida por un único contador puede representarse como la suma de la potencia activa de cada carga, más el ruido de medición [38]:

$$p(t) = \sum_{k=1}^K p_k(t) + n(t), \quad (2.1)$$

donde  $p(t)$  es la potencia activa agregada,  $p_k(t)$  es la potencia de la carga  $k$ ,  $K$  es el número total de cargas y  $n(t)$  representa ruido de medición. En la práctica, el número de cargas de interés suele ser menor que el total, por lo que el consumo de las cargas restantes puede agruparse como una perturbación adicional [38]:

$$p(t) = \sum_{k=1}^M p_k(t) + v(t), \quad (2.2)$$

donde  $M \leq K$  es el número de cargas de interés y  $v(t)$  agrupa el consumo de las cargas restantes junto con el ruido de medición.

En el trabajo presentado [32] utilizan z-score el cual indica cuántas desviaciones estándar está un punto de datos específico por encima o por debajo de la media de un conjunto de datos, además utilizan extracción de características, y un clasificador basado en XGBoost. Los resultados que obtuvieron mostraron un rendimiento superior de este último (90 % de exactitud, 92 % de *precision*, 92 % de *recall* y 91 % de *F1*), superando a modelos como *Decision Tree* (84 % de exactitud) y *Logistic Regression* (80 %). Para evaluar dichos modelos se emplean métricas estándar como:

**la exactitud**, indica qué proporción del total de predicciones fue correcta. Resume el desempeño global del modelo, considerando tanto aciertos en la clase positiva como en la

negativa.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN},$$

**la precision**, que mide cuántas de las instancias que el modelo clasificó como positivas realmente lo eran. Evalúa qué tan “precisas” son las predicciones positivas, evitando falsos positivos.

$$\text{Precision} = \frac{TP}{TP + FP},$$

**el recall**, el cual indica qué porcentaje de los casos positivos reales fue correctamente identificado por el modelo. Evalúa la capacidad de “detectar” todos los positivos, evitando falsos negativos.

$$\text{Recall} = \frac{TP}{TP + FN},$$

y el **puntaje F1**, el cual combina **precision** y **recall** en una sola métrica. Es útil cuando existe desequilibrio entre clases o cuando es importante equilibrar falsos positivos y falsos negativos.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

En estas métricas, los términos utilizados se definen como:

- **TP (True Positives)**: casos positivos correctamente identificados por el modelo.
- **TN (True Negatives)**: casos negativos correctamente identificados por el modelo.
- **FP (False Positives)**: casos que fueron clasificados como positivos sin serlo, es decir, “falsas alarmas”.
- **FN (False Negatives)**: casos positivos que el modelo no logró detectar, es decir, “detecciones perdidas”.

Los resultados obtenidos en [32] demuestran que la selección de características influyen directamente en el rendimiento del modelo. Además estas técnicas es posible aplicarlas al

contexto del agua, y se evidencia que la extracción de características son fundamentales para lograr desagregaciones confiables.

### 2.3 Datasets para desagregación

El desarrollo de modelos de desagregación depende de la disponibilidad de datasets que capturen el comportamiento real del consumo en hogares. Sin embargo, en las investigaciones se evidencia que todavía existe una limitación en la cantidad y diversidad de conjuntos de datos disponibles para investigación.

Uno de los primeros datasets es el trabajo de [22], donde se presenta un sistema IoT para el monitoreo y recopilación del consumo residencial mediante sensores de flujo y un microcontrolador. Este estudio generó un dataset real correspondiente al uso de agua de un único hogar en Italia, con una única persona durante un periodo prolongado de tiempo. Aunque el dataset es limitado en tamaño, este conjunto de datos permitió caracterizar el comportamiento individual del consumo y constituyó una base experimental para evaluar técnicas de segmentación y análisis temporal.

Posteriormente, en [24] se realiza una búsqueda exhaustiva de los datasets disponibles. Los autores clasifican los conjuntos de datos en categorías como mediciones a nivel de hogar, mediciones por punto de consumo y registros con distintos niveles de granularidad temporal. Su estudio destaca la falta de estandarización y la escasez de datos abiertos para la comunidad. Asimismo, identifican que la mayoría de los datasets disponibles corresponden a estudios piloto en regiones específicas, lo que dificulta el entrenamiento de modelos generalizables.

Otra manera de obtener datasets es la generación de datos sintéticos. En [27] se propone un modelo estocástico para simular series temporales de consumo residencial. En esta investigación se utiliza información que fue recopilada de distintas localidades de Estados Unidos para generar consumo de agua de forma realista, considerando horarios típicos de consumo, duración de eventos y probabilidades de activación de artefactos. Este tipo de datasets es útil cuando no se dispone de registros reales de alta resolución o cuando se requiere un volumen considerable de datos para entrenar modelos complejos de aprendizaje automático.

Por lo tanto la evidencia muestra que, aunque existen algunos datasets relevantes, aún hay

importantes desafíos como la escasez de datos abiertos, baja diversidad geográfica, falta de estándares y limitaciones en la resolución temporal. Estos factores motivan la recolección, simulación y anotación de datos que permitan avanzar en el estado del arte del consumo desagregado.

### 2.3.1 Resumen comparativo de datasets

Con el fin de sintetizar las principales características de los datasets presentes en la investigación realizada sobre desagregación del consumo de agua, y facilitar su comparación, en la Tabla ?? se presenta un resumen de diversos conjuntos de datos. En particular, se consideran aspectos clave como la frecuencia de muestreo, el tipo de variables disponibles, la cantidad de viviendas monitoreadas, el tipo de datos y el nivel de acceso.

Dataset	Origen	Viviendas	Frecuencia	Accesibilidad y características
WEUSEDTO	Italia	1	1 segundo	Dataset real, público, con eventos de consumo etiquetados por artefacto. Adecuado para validación experimental de modelos de desagregación.
STREaM	EE.UU. (simulado)	Múltiples (simulado)	10 segundos (configurable)	Dataset sintético generado mediante modelos estocásticos calibrados con datos reales. Permite simular escenarios variados y controlados.
Datasets propietarios (Ej. Pecan Street)	EE.UU.	Múltiples	Alta	Acceso restringido o pagado. Datos no completamente disponibles para investigación abierta, lo que limita su reproducibilidad y uso académico.

Tab. 2.2: Resumen comparativo de datasets para desagregación del consumo de agua

Tal como se observa en la Tabla ??, una parte significativa de los datasets utilizados en investigación sobre consumo residencial de agua no se encuentra disponible de forma pública. En muchos casos, estos conjuntos de datos corresponden a infraestructuras de *smart water*

*metering* operadas por empresas sanitarias o empresas privadas, donde las restricciones de privacidad y uso de información personal impiden su libre acceso y el acceso a documentación detallada.

Si bien existen iniciativas como Pecan Street <sup>2</sup>, estas requieren licencias de pago o presentan limitaciones de compatibilidad con estudios académicos, lo que dificulta su utilización en contextos de investigación reproducible.

En este escenario, los datasets **WEUSEDTO** y **STREaM** son alternativas viables y complementarias. El primero permite trabajar con datos reales de alta resolución temporal, mientras que el segundo facilita la generación de escenarios controlados y escalables mediante datos sintéticos. Por estas razones, ambos conjuntos de datos son seleccionados como base para el desarrollo y validación de los modelos de desagregación propuestos en el presente trabajo.

### 2.4 Métodos de detección de anomalías

La detección de anomalías en el consumo, ya sea energético o hídrico, es un problema clave para identificar comportamientos inadecuados, fallas en dispositivos o posibles fugas. En la investigación se han utilizado diferentes métodos, que van desde los estadísticos basados en percentile hasta modelos avanzados de aprendizaje automático y técnicas basadas en lógica difusa, tal como muestran diversos trabajos recientes [28, 30, 31, 29].

#### 2.4.1 Métodos estadísticos basados en percentiles

Los métodos tradicionales de detección de anomalías utilizan umbrales que son obtenidos directamente a partir de los datos históricos de consumo. A través del cálculo de percentiles (por ejemplo, P90, P95 o P99), es posible estimar el rango esperado del consumo y marcar como anomalías aquellos valores que exceden dichos límites [30].

---

<sup>2</sup> <https://www.pecanstreet.org/dataport/>

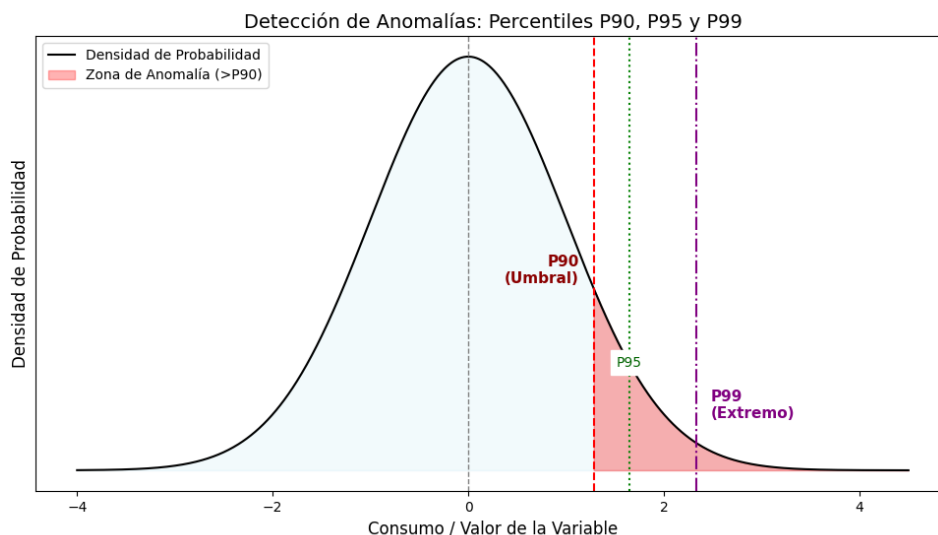


Fig. 2.4: Representación conceptual de una distribución de probabilidad del consumo y la ubicación de distintos percentiles (P90, P95 y P99). Los percentiles elevados permiten definir distintos niveles de umbral para la detección de consumos atípicos o extremos.

Tal como se ilustra en la Figura 2.4, los percentiles elevados corresponden a eventos de baja probabilidad dentro de la distribución del consumo. Los percentiles como el P90, P95 y P99 permiten definir distintos niveles de severidad, donde valores cercanos a P99 representan eventos extremos. Este criterio permite establecer umbrales estadísticos de forma directa y sin requerir entrenamiento de modelos complejos.

Este enfoque es simple, interpretable y computacionalmente liviano, por lo que suele emplearse como línea base en sistemas de monitoreo en tiempo real. Por estas razones, diversos estudios en contextos IoT lo consideran un paso inicial antes de aplicar métodos más avanzados de detección de anomalías [30].

#### 2.4.2 Métodos basados en aprendizaje automático

En los últimos años han surgido técnicas basadas en aprendizaje automático que buscan aprender los patrones normales de consumo directamente a partir de los datos, sin requerir umbrales fijos ni supuestos estadísticos estrictos. Estos enfoques son especialmente útiles en el IoT, donde los patrones de consumo pueden tener alta variabilidad y cambios dinámicos en el tiempo.

Uno de los métodos más utilizados es *Isolation Forest*, el cual es un algoritmo no supervisado que detecta anomalías a partir del principio de aislamiento. A diferencia de otros enfoques, este método no muestra explícitamente el comportamiento normal, sino que identifica como anómalas aquellas observaciones que pueden ser separadas del resto de los datos mediante un menor número de particiones aleatorias. En la Figura 2.5 se presenta un esquema conceptual del funcionamiento de este método, tomado de una fuente especializada<sup>3</sup>. Debido a su eficiencia computacional y a que no requiere asumir ninguna distribución específica de los datos, *Isolation Forest* ha sido ampliamente adoptado en entornos IoT.

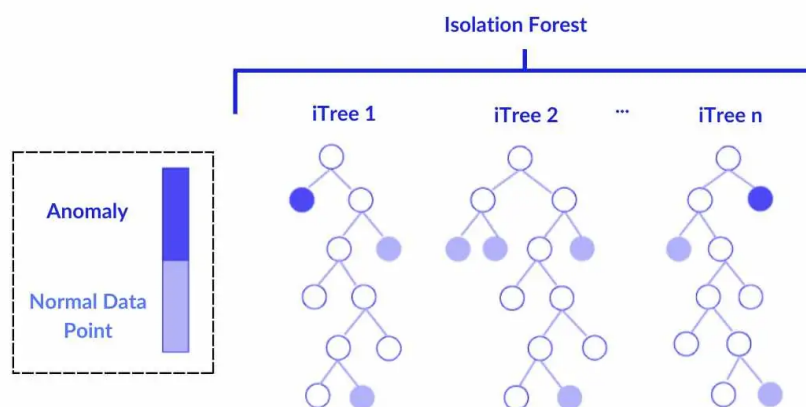


Fig. 2.5: Esquema conceptual del funcionamiento de Isolation Forest, donde las observaciones anómalas son aisladas con un menor número de particiones en comparación con los datos normales.

Como se observa en la Figura 2.5, las anomalías tienden a presentar menores profundidades promedio en los árboles de aislamiento.

Por otra parte, los métodos basados en *one-class Support Vector Machine* (one-class SVM) buscan aprender una frontera que delimite el conjunto de comportamientos considerados normales en el espacio de características. La Figura 2.6 ilustra este principio geométrico mediante un ejemplo bidimensional, adaptado de la literatura técnica<sup>4</sup>. En este enfoque, el modelo es entrenado únicamente con datos normales, y aquellas observaciones que quedan fuera de la frontera aprendida son clasificadas como anómalas.

<sup>3</sup> <https://spotintelligence.com/2024/05/21/isolation-forest/>

<sup>4</sup> <https://www.mdpi.com/2076-3417/13/3/1734>

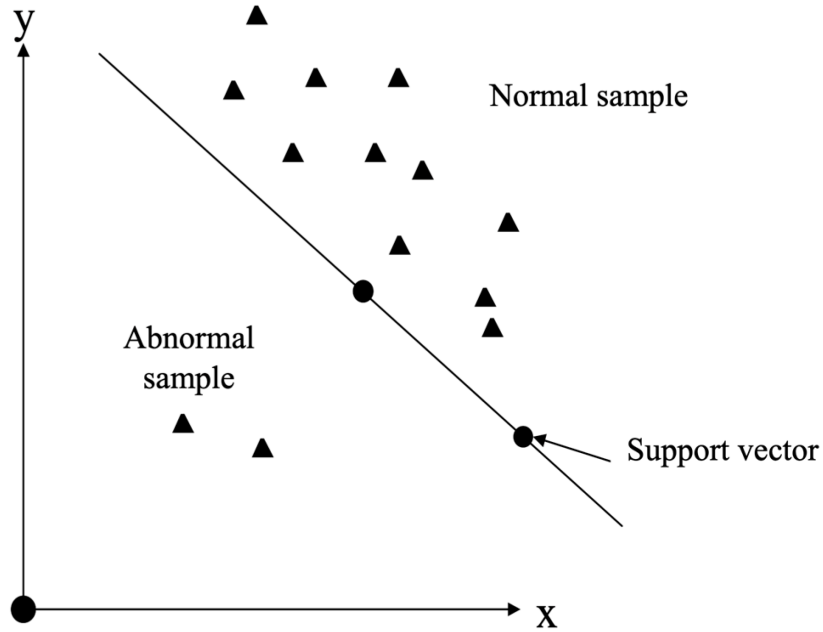


Fig. 2.6: Representación conceptual de un modelo one-class SVM, donde una frontera delimita el comportamiento normal y los puntos fuera de esta son considerados anomalías.

La Figura 2.6 muestra el principio geométrico de separación utilizado por los métodos one-class SVM.

También han surgido enfoques más avanzados orientados al modelamiento temporal del consumo. En [31], los autores proponen un modelo basado en redes neuronales recurrentes LSTM combinado con regresión de cuantiles (*Deep Quantile Regression*), el cual permite estimar directamente intervalos de predicción asociados a distintos niveles de probabilidad, sin asumir una distribución específica del error.

Sea  $y_t$  el valor observado del consumo en el instante  $t$  y  $\hat{y}_t^{(\alpha)}$  la predicción del cuantil  $\alpha \in (0, 1)$  generada por el modelo LSTM. El entrenamiento se realiza minimizando la función de pérdida de regresión de cuantiles, definida como:

$$\mathcal{L}_\alpha(\xi_t) = \begin{cases} \alpha \xi_t, & \text{si } \xi_t \geq 0 \\ (\alpha - 1) \xi_t, & \text{si } \xi_t < 0 \end{cases} \quad (2.3)$$

donde  $\xi_t = y_t - \hat{y}_t^{(\alpha)}$  corresponde al error de predicción para el cuantil  $\alpha$ . En la práctica, el modelo es entrenado para estimar múltiples cuantiles, típicamente un cuantil inferior y uno superior. A partir de estas predicciones se define el intervalo de cuantiles como:

$$QI_t = \hat{y}_t^{(\alpha_u)} - \hat{y}_t^{(\alpha_l)} \quad (2.4)$$

donde  $\alpha_l$  y  $\alpha_u$  representan los cuantiles inferior y superior, respectivamente. La amplitud de este intervalo actúa como una medida de incertidumbre del modelo y se utiliza como *score de anomalía*, de modo que valores elevados de  $QI_t$  indican comportamientos atípicos o no observados durante el entrenamiento. Los resultados reportados en [31] muestran que este enfoque presenta mayor robustez frente a cambios abruptos en las series temporales y reduce la generación de falsas alarmas en comparación con métodos basados en umbrales fijos.

Finalmente, además del desarrollo de modelos, resulta relevante considerar el contexto de despliegue de estas técnicas. En [28] se presenta una arquitectura que integra modelos livianos de aprendizaje automático directamente en medidores inteligentes, destacando la importancia de combinar procesamiento local (*edge computing*) con análisis en la nube. Este enfoque permite reducir latencia, consumo energético y dependencia de la conectividad, aspectos críticos en sistemas de monitoreo IoT a gran escala.

### 2.4.3 Métodos basados en predicción futura

Otra línea de investigación relevante para la detección de anomalías en el consumo de agua corresponde a los enfoques basados en predicción futura (*forecasting-based anomaly detection*). En estos métodos, se entrena un modelo para estimar el valor esperado del consumo en instantes futuros y las anomalías se identifican a partir de desviaciones significativas entre el consumo observado y el consumo predicho [43].

Para cuantificar esta desviación de manera formal, la literatura propone el uso de estadísticos como  $\chi^2$ , el cual mide la discrepancia normalizada entre los valores observados y los esperados:

$$\chi^2 = \sum_{i=1}^n \frac{(X_i - E_i)^2}{E_i} \quad (2.5)$$

donde  $X_i$  representa el valor observado y  $E_i$  el valor esperado derivado del modelo [43]. Un evento es considerado anómalo cuando la magnitud de este estadístico excede un umbral predefinido, el cual puede establecerse mediante criterios estadísticos clásicos.

Estos enfoques han sido implementados principalmente utilizando modelos estadísticos clásicos, como ARIMA. Sin embargo, la literatura indica que este tipo de modelos presenta limitaciones para capturar la no linealidad y la variabilidad inherente al consumo de agua en entornos residenciales [44].

No obstante, en entornos residenciales reales el consumo de agua suele presentar una alta variabilidad asociada al comportamiento humano, lo que puede afectar la estabilidad de los modelos predictivos y aumentar la generación de falsas alarmas. Es por esto que en las investigaciones se sugiere complementar estos métodos con enfoques no supervisados o híbridos cuando no se dispone de históricos extensos [45].

### **2.4.4 Lógica difusa para evaluación de severidad**

Más allá de clasificar un evento como “normal” o “anómalo”, en sistemas de monitoreo es indispensable cuantificar el nivel de severidad de cada anomalía detectada. En escenarios reales, los patrones de uso tienen variaciones graduales que no pueden ser correctamente representadas mediante umbrales que sean estrictos, ya que estos tienden a generar falsas alarmas ante desviaciones que pueden ser naturales del comportamiento del usuario. Es por esto que la lógica difusa surge como una alternativa adecuada, ya que permite representar la incertidumbre y la gradualidad inherente a los datos mediante grados de pertenencia, en lugar de decisiones binarias estrictas [39, 40].

### **Fundamentos y representación difusa**

La lógica difusa extiende la lógica clásica permitiendo que una variable pertenezca parcialmente a uno o más conjuntos difusos, con grados de pertenencia definidos en el intervalo continuo  $[0, 1]$ . Esta característica resulta especialmente adecuada para modelar fenómenos imprecisos o variables altamente dependientes del contexto, como el consumo de agua en entornos residenciales y la detección de fugas en redes de distribución [40]. La representación de estos conjuntos se realiza mediante funciones de membresía, las cuales traducen variables numéricas, como caudal, duración o volumen consumido, en conceptos lingüísticos interpretables tales como “bajo”, “medio” o “alto”.

En aplicaciones prácticas de monitoreo, se emplean comúnmente funciones de membresía

triangulares, trapezoidales o gaussianas, debido a su simplicidad computacional y su capacidad para representar transiciones suaves entre estados [41]. La selección del tipo de función depende del equilibrio entre precisión y complejidad, siendo las funciones gaussianas especialmente útiles en sistemas donde se busca evitar cambios abruptos en la evaluación del estado del sistema.

### **Inferencia difusa y evaluación de severidad**

A partir de las funciones de membresía, la lógica difusa añade reglas lingüísticas del tipo *IF-THEN*, las cuales permiten combinar múltiples variables de entrada y replicar el razonamiento humano a través de reglas claras y transparentes. Estas reglas definen relaciones del tipo: “SI el caudal es alto Y la duración es prolongada, ENTONCES la severidad del evento es alta”, lo cual permite evaluar escenarios complejos y brindar explicaciones semánticas sin recurrir a modelos estrictamente deterministas [42].

El resultado del proceso de inferencia corresponde a una salida difusa que representa el grado de severidad del evento. Para obtener un valor cuantitativo final, este resultado es transformado mediante un proceso de defuzzificación, siendo el método del centroide (Center of Gravity) uno de los más utilizados en las investigaciones recientes por su capacidad para integrar la contribución de todas las reglas activadas [41]. El valor obtenido puede ser posteriormente categorizado en niveles de severidad como “leve”, “moderada” o “severa”, lo que facilita la priorización de alertas y la toma de decisiones. Este enfoque permite tolerar variaciones habituales del consumo sin confundirlas con fallas críticas, reduciendo la generación de falsas alarmas en sistemas de monitoreo dinámicos [29].

### **2.5 Síntesis del estado del arte**

La revisión del estado del arte muestra que el monitoreo del consumo de agua ha avanzado gracias a tecnologías IoT, sensores de flujo y plataformas en la nube, pero que la mayoría de las soluciones comerciales y académicas siguen siendo obstructivas (sensores instalados en la tubería) o dependen del recambio completo a medidores inteligentes. Si bien estos enfoques permiten una medición de alta resolución y la detección de fugas, estos tienen grandes

inconvenientes tanto en costo como la complejidad de instalación, que en contextos como el chileno, dificultan poder incorporar esto de manera masiva. Métodos no obstructivos como los sensores ultrasónicos y las soluciones basadas en visión por computador demuestran ser una alternativa viable para reutilizar los medidores tradicionales existentes, pero la mayoría de los trabajos se concentran en la digitalización de la lectura y no en el análisis avanzado del consumo.

Además, la NIWD y la detección de anomalías han incorporado técnicas de aprendizaje automático capaces de identificar patrones de uso por dispositivo y eventos anómalos a partir del consumo total del medidor. Sin embargo, todo esto viene acompañado de pocos datasets que sean reales y estén abiertos al público, los cuales tienen baja diversidad geográfica y limitaciones en la resolución temporal, lo que complica la generalización de los modelos. Si bien en la literatura actual se ofrecen herramientas que van desde métodos estadísticos y algoritmos de aprendizaje automático como Isolation Forest o SVM, hasta la lógica difusa para evaluar severidad, la integración en conjunto de estas técnicas en sistemas de bajo costo y no intrusivos sigue siendo escasa. Actualmente, existe una brecha en la aplicación de estos modelos en escenarios residenciales reales que se adapten al contexto local. El enfoque de este proyecto es contribuir con un sistema que integra medición no obstructiva basada en visión por computadora, desagregación del consumo y detección de anomalías con clasificación flexible de severidad, orientada específicamente a las necesidades de los hogares chilenos.

La Tabla 2.3 resume las características principales de los trabajos orientados a la desagregación del consumo. En general, las investigaciones se centran en la clasificación por dispositivo mediante modelos de aprendizaje automático y en el uso de características derivadas del caudal y la duración. Sin embargo, ninguno de ellos incorpora captura no obstructiva mediante visión por computador, ni contempla la integración conjunta entre desagregación y detección de anomalías dentro de un mismo sistema. Estos vacíos motivan la contribución del presente trabajo, el cual propone una solución que es no obstructiva.

Por otra parte, cuando se consideran trabajos enfocados específicamente en la detección de anomalías en el consumo, la situación es diferente. En la misma Tabla 2.3 se observa que, si bien existen trabajos que abordan la detección de anomalías en consumo, la combinación de un enfoque no obstructivo para el monitoreo de agua, junto con Isolation Forest, lógica difusa

## 2. ESTADO DEL ARTE

de severidad e integración en un pipeline IoT orientado a usuarios hogares, sigue siendo un vacío presente en la literatura, lo que refuerza el aporte de la presente memoria.

<b>Característica</b>	<b>Attallah (2020)</b>	<b>Gkalinikis (2022)</b>	<b>Gerasimov (2025)</b>	<b>Esta memoria (2025)</b>
<b>Bloque 1: Desagregación del consumo</b>				
Clasificación por dispositivo	X	X		X
Extracción de características	X		X	X
Modelos de Machine learning	X	X	X	X
Métricas Accuracy, Precision, Recall, F1	X		X	X
Solución no obstructiva de medición				X
comparación entre datasets reales y sintéticos				X

<b>Característica</b>	<b>Okada (2022)</b>	<b>Qaddoori (2023)</b>	<b>Tambuwal &amp; Neagu (2021)</b>	<b>Fuzzy (Scidirect)</b>	<b>Esta memoria (2025)</b>
<b>Bloque 2: Detección de anomalías</b>					
Dominio hídrico (agua)	X	X		X	X
Uso en contexto IoT / medidor	X	X		X	X
Detección de anomalías	X	X	X	X	X
Métodos estadísticos (percentiles)	X				X
Isolation Forest	X				X
Lógica difusa para severidad				X	X
Clasificación de severidad de eventos				X	X
Integración desagregación + anomalías					X

Tab. 2.3: Comparación de características entre trabajos de desagregación, detección de anomalías y la presente memoria.

# CAPÍTULO 3: DESAGREGACIÓN DEL CONSUMO DE AGUA

---

## 3.1 Diseño de los módulos principales

Antes de tratar en detalle los modelos y técnicas de desagregación del consumo de agua, es necesario presentar una visión general de la arquitectura del sistema propuesto. Dicho sistema ha sido diseñado siguiendo un enfoque por capas, lo que permite organizar sus componentes de acuerdo con su funcionalidad y facilitar tanto su comprensión como su escalabilidad.

Tal como se muestra en el modelo de capas presentado en el Anexo 5.2, la arquitectura se compone de una capa física de sensado, una capa de comunicación IoT, una capa de procesamiento y servicios desplegada en la nube, y una capa de aplicación orientada a la visualización de información y la interacción con el usuario. Teniendo esto en mente, el presente trabajo se centra principalmente en la capa de procesamiento, donde se desarrollan los módulos de análisis de datos, en particular aquellos asociados a la desagregación del consumo y la detección de anomalías.

Sobre esta base, el sistema se estructura de manera modular, lo cual permite separar las funciones principales en cuatro grandes grupos:

- Módulo de sensado.
- Desagregación del consumo total.
- Detección de anomalías.
- Aplicación web

La Figura 3.1 muestra una vista general de como es el flujo de datos entre estos componentes, permitiendo entender como la información capturada en el medidor es transformada progresivamente hasta llegar a la interfaz de consulta del usuario.

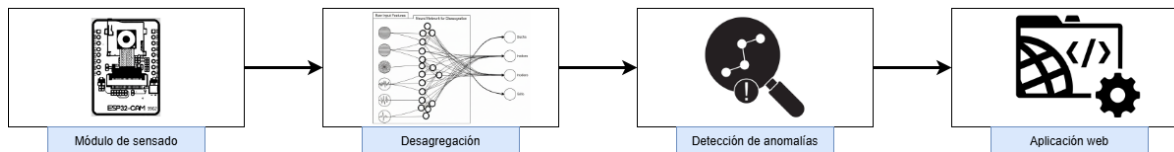


Fig. 3.1: Módulos por capas

#### 3.1.1 Alcance de la implementación realizada

Este capítulo y el siguiente describen la implementación de los componentes desarrollados en este proyecto y su integración dentro del sistema global de monitoreo del consumo de agua previamente presentado. Si bien el sistema contempla diversos módulos que conforman una solución completa de monitoreo, el alcance del presente trabajo se definió de manera explícita sobre aquellos componentes que constituyen el núcleo analítico del sistema.

En el presente informe se aborda el diseño, la implementación e integración de los módulos asociados al análisis de datos, los cuales permiten transformar las mediciones de consumo en información de alto valor para el usuario final. Estos módulos corresponden a:

- La construcción e implementación del **módulo de desagregación de consumos**, basado en técnicas de extracción de características y clasificación mediante modelos de aprendizaje automático.
- El desarrollo del **módulo de detección de anomalías**, complementado con un sistema de lógica difusa para la clasificación de severidad.
- La integración de ambos módulos dentro del pipeline de procesamiento y su despliegue en la infraestructura de AWS.

Estos módulos son el aporte de este proyecto y representan las capacidades analíticas del sistema, los cuales permiten transformar lecturas en crudo en información útil sobre hábitos de consumo y posibles situaciones anómalas.

## 3.2 Diseño del módulo de desagregación

El módulo de desagregación tiene como propósito identificar a qué tipo de dispositivo del hogar corresponde cada ventana de consumo registrada. Este proceso se inspira en los enfo-

ques de desagregación no intrusiva del consumo eléctrico (NILM), ampliamente utilizados en el ámbito energético y revisados en el estado del arte, los cuales han sido adaptados en este trabajo al contexto del consumo de agua, considerando las particularidades que tiene el uso de agua, tales como la variabilidad del caudal y la duración de los eventos de uso [36, 45].

Este módulo clasifica los periodos de uso en categorías como ducha, lavamanos, lavaplatos, inodoro u otros dispositivos, permitiendo obtener información más detallada del comportamiento del consumo. Su funcionamiento se aprecia en la Figura 3.2.

La entrada del módulo corresponde a las lecturas en crudo almacenadas en la base de datos. En la etapa de preprocesamiento, estas lecturas son agrupadas con el fin de formar ventanas de consumo que reflejan periodos continuos de uso de agua.

Luego, para cada una de estas ventanas se realiza la extracción de características, las cuales permiten distinguir patrones de uso asociados a diferentes dispositivos del hogar.

Finalmente, un algoritmo de clasificación utiliza estas características para determinar a qué categoría de dispositivo corresponde cada ventana de consumo. Los resultados del proceso de desagregación se almacenan en la base de datos del sistema, quedando disponibles tanto para su visualización como para otras etapas del sistema, como la detección de anomalías.

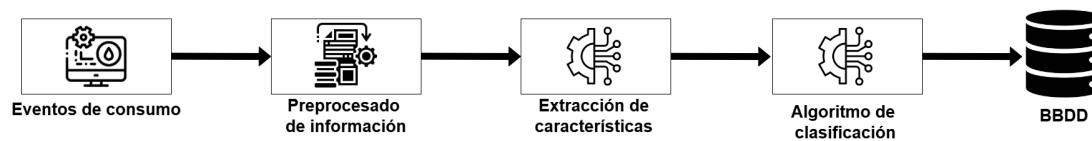


Fig. 3.2: Módulo de desagregación

### 3.3 Preparación y construcción del dataset

La construcción de los dataset utilizados para entrenar y evaluar los modelos de desagregación y detección de anomalías se realizó a partir de dos fuentes principales de datos: el dataset **WEUSEDTO** (Water End Use Dataset and Tools) y el modelo **STREaM** (Stochastic Residential Water End Use Model) para generación de datos. Ambos se presentan en el Capítulo de Estado del Arte, pero en esta sección se hace uso de estos en la implementación.

El dataset **WEUSEDTO**, descrito en trabajos como [23, 25], corresponde a mediciones

reales de consumo de agua por dispositivo registradas con alta resolución temporal, con un intervalo de muestreo de **1 segundo**. Por otra parte, **STREaM** permite generar series temporales sintéticas de consumo residencial a partir de modelos estocásticos, pudiendo simular patrones de uso para distintos tipos de dispositivos, en este caso, las series se generan con una resolución temporal mínima de **10 segundos** entre muestras [27].

En este capítulo se realiza el preprocesamiento, limpieza y unificación de los datos, con el objetivo de obtener datasets estructurados que puedan ser utilizados para el entrenamiento de los modelos de clasificación y de detección de anomalías.

#### 3.3.1 Dataset 1: WEUSEDTO

##### Descripción de la fuente

El dataset **WEUSEDTO** tiene mediciones de consumo de agua a nivel de dispositivo, las cuales tienen una resolución temporal de un segundo. Cada dispositivo del hogar se encuentra en un archivo independiente, dentro de los cuales los que se utilizaron fueron :

- `feed_Kitchenfaucet.MYD.csv`: correspondientes al lavaplatos.
- `feed_Shower.MYD.csv`: correspondientes a la ducha.
- `feed_Washbasin.MYD.csv`: correspondientes al lavamanos.
- `feed_Washingmachine.MYD.csv`: correspondientes a la lavadora.
- `feed.Toilet.csv`: correspondientes al inodoro.
- `feed_Dishwasher30.MYD.csv`: correspondientes al lavavajillas.

Cada uno de estos archivos contiene las mediciones de caudal asociadas al artefacto correspondiente. Sin embargo, la estructura de las columnas no es la misma por cada uno de estos dispositivos, en algunos casos solo se tiene tiempo y flujo (por ejemplo, ducha o lavamanos), mientras que en otros se incluye explícitamente un tiempo de inicio, flujo y tiempo de término (como el inodoro o el lavavajillas). Además, los nombres de las columnas y la forma en que se separan los datos varían entre archivos, con lo cual fue necesario un proceso de normalización.

El objetivo del preprocesamiento fue transformar estos datos en eventos de consumo, donde cada evento es un intervalo continuo de uso de agua, el contiene datos como el tiempo de inicio, tiempo de término, duración, flujo promedio y volumen total consumido.

#### **Preprocesamiento y limpieza**

Como los archivos originales tenían diferencias en el formato (separadores, presencia o ausencia de cabeceras y nombres de columnas), el primer paso consistió en estandarizar su estructura. Para ello se implementaron las siguientes etapas:

- **Detección del separador:** se inspecciona la primera línea de cada archivo para identificar que tipo de seaparador hay, si es que es coma, tabulación u otro carácter. En base a esto, se configura el parámetro `sep` al momento de cargar el archivo con `pandas`.
- **Normalización de nombres de columnas:** se creó una función que transforma los nombres originales a nombres estándar:
  - `Time`: tiempo de medición o inicio.
  - `Flow`: caudal instantáneo medido.
  - `End_time`: tiempo de término del evento (cuando está disponible).

De este modo, independientemente del formato original, las columnas quedan unificadas para su procesamiento posterior.

- **Conversión de tipos y ordenación:** tanto `Time` como `Flow` (y `End_time`, cuando existe) se convierten a valores numéricos, descartando filas que no puedan interpretarse correctamente. Luego, los registros se ordenan por `Time` para respetar la secuencia temporal de los datos.
- **Filtrado de mediciones sin flujo:** para la construcción de eventos de consumo solo se consideran aquellas muestras donde el caudal es estrictamente positivo ( $Flow > 0$ ), descartando periodos sin uso.
- **Manejo inicial de `End_time`:** en los dispositivos que ya incluyen explícitamente el tiempo de término, esta columna se conserva después de la normalización. En cambio,

en los archivos que solo tienen de la serie `Time-Flow`, la columna `End_time` no se construye aún en esta etapa, sino que se genera posteriormente durante la agrupación de muestras en eventos continuos, definiendo el `End_time` como la última marca temporal de cada grupo.

#### Unificación de formatos y estructura común

Una vez normalizados los datos, para el análisis posterior se definieron las siguientes columnas:

- `Time`: instante de inicio del evento.
- `End_time`: instante de término del evento.
- `Duration_s`: duración del evento en segundos.
- `Flow_Lmin`: caudal promedio del evento en litros por minuto.
- `Litros_totales`: volumen total consumido durante el evento, en litros.

Cabe destacar que la columna **End\_time** se maneja dependiendo de la información que exista en cada archivo. En los dispositivos que ya tiene esta columna, simplemente se conserva después de la normalización y se utiliza para el cálculo de la duración. En cambio, en los archivos que solo contienen **Time** y **Flow**, la columna **End\_time** aún no se construye en esta etapa, sino que se genera después, específicamente durante la agrupación en eventos, la cual se agrupa desde que comienza un flujo positivo hasta que llega la última medición con flujo positivo, en donde se toma el tiempo de la última medición como el parámetro **End\_time**.

*Archivos sin **End\_time** (agrupación en eventos):* En los dispositivos donde solo se tiene de **Time** y **Flow**, es necesario agrupar las mediciones en eventos de uso. Para lo cual se utilizó un umbral de separación temporal específico para cada dispositivo (por ejemplo, 1 minuto para lavaplatos, 1 minuto para ducha, 3 minutos para lavadora, etc). Si la diferencia de tiempo entre dos mediciones consecutivas supera dicho umbral, se considera que se inicia un nuevo evento.

### 3. DESAGREGACIÓN DEL CONSUMO DE AGUA

---

Sea  $t_i$  la secuencia de tiempos con flujo positivo para un dispositivo dado. La diferencia entre mediciones se calcula como:

$$\Delta t_i = t_i - t_{i-1}.$$

Cuando  $\Delta t_i$  supera el umbral definido para el dispositivo o se trata de la primera muestra, se inicia un nuevo evento. Para cada evento se obtiene:

$$\text{Time} = t_{\text{inicio}}, \quad \text{End\_time} = t_{\text{término}},$$

$$\text{Duration\_s} = \text{End\_time} - \text{Time}.$$

El caudal instantáneo almacenado de la columna **Flow** tiene una unidad de medida en mililitros por segundo (mL/s). Para utilizar una unidad más interpretable, se convierte a litros por minuto (L/min) mediante:

$$\text{Flow\_Lmin} = \text{Flow\_mL/s} \cdot 0,06,$$

ya que 1 mL/s equivale a 0,06 L/min.

El caudal promedio del evento se calcula como el promedio de las mediciones del evento, y para el volumen total consumido se hace una aproximación mediante una sumatoria del caudal por el intervalo temporal entre mediciones:

$$V_{\text{ml}} \approx \sum_i \text{Flow}_i \cdot \Delta t_i,$$

donde  $\text{Flow}_i$  está en mL/s y  $\Delta t_i$  en segundos. Finalmente, el volumen se convierte a litros mediante:

$$\text{Litros\_totales} = \frac{V_{\text{ml}}}{1000}.$$

Asimismo, se aplicó un filtro por volumen mínimo para cada tipo de dispositivo con el fin de descartar mediciones de un valor irrelevante.

Archivos con **End\_time** (cálculo directo de eventos): En los dispositivos que ya incluyen **Time**, **Flow** y **End\_time**, el proceso es más directo. La duración del evento se calcula como:

$$\text{Duration}_s = \text{End\_time} - \text{Time},$$

forzando un valor mínimo de 1 segundo en casos donde la diferencia es cero o negativa debido a inconsistencias en los datos originales.

El caudal se convierte nuevamente desde mL/s a L/min mediante:

$$\text{Flow\_Lmin} = \text{Flow\_mL/s} \cdot 0,06,$$

y el volumen total se obtiene mediante:

$$\text{Litros\_totales} = \frac{\text{Flow\_Lmin} \cdot \text{Duration}_s}{60}.$$

De esta forma, tanto los dispositivos que requieren agrupación de mediciones como aquellos que ya incluyen tiempos de inicio y término quedan bajo una misma estructura. El resultado es un conjunto homogéneo de eventos de consumo, con información temporal y volumétrica consistente, listo para ser utilizado en las etapas posteriores de desagregación y detección de anomalías.

*Asignación de la columna **Dispositivo** y unificación final:* Una vez listos los eventos de cada dispositivo por separado, se genera un archivo por dispositivo con la estructura mencionada anteriormente. Pero para preparar el dataset que los modelos de desagregación utilizarán, estos archivos se deben combinar en un único dataset unificado. Para esto se agrega una columna adicional la cual tiene el nombre de **Dispositivo**, que identifica el tipo de artefacto al que corresponde cada evento (por ejemplo, **ducha**, **lavamanos**, **lavaplatos**, **inodoro**). Esta columna es indispensable para los algoritmos de clasificación, ya que es la etiqueta objetivo (*target*) para el entrenamiento supervisado.

El dataset unificado se obtiene concatenando los archivos procesados individualmente y ordenándolos según la columna **Time**. El resultado es un único archivo que contiene todos los eventos del dataset **WEUSEDTO**, para su uso en las etapas posteriores de entrenamiento.

#### 3.3.2 Dataset 2: STREaM

##### Descripción de la fuente

El segundo conjunto de datos utilizado en este trabajo corresponde al modelo **STREaM**, el cual genera datos de consumo de agua a partir de un modelo estocástico a nivel de dispositivo, el cual se trabaja en MATLAB. A diferencia del dataset **WEUSEDTO**, que proviene de mediciones reales de un único hogar, en este caso se simula el consumo de un hogar de tamaño configurable, lo cual permite controlar el número de integrantes en un hogar, los dispositivos presentes y el horizonte temporal de la simulación.

Para este trabajo se configuró el modelo con un tamaño de hogar de cuatro personas:

$$\text{param.HHsize} = 4,$$

y se seleccionaron únicamente artefactos de consumo estándar (no de alta eficiencia), marcando como presentes los siguientes dispositivos:

- `StToilet` (inodoro estándar),
- `StShower` (ducha estándar),
- `StFaucet` (grifos del hogar: lavamanos, lavaplatos, etc.),
- `StClothesWasher` (lavadora estándar),
- `StDishwasher` (lavavajillas estándar),

y desactivando los dispositivos de alta eficiencia. Así, el escenario que se está simulando permite replicar el uso de un hogar normal con un conjunto de diversos dispositivos, sin incorporar aún medidas de alta eficiencia.

El horizonte temporal de la simulación se fijó en:

$$\text{param.H} = 90 \text{ días},$$

lo que equivale aproximadamente a tres meses de consumo. La resolución temporal se configuró mediante:

`param.ts = 1,`

donde cada unidad de tiempo corresponde a un intervalo de 10 segundos. Por lo tanto, los datos generados para cada dispositivo contiene un valor de consumo cada 10 segundos a lo largo de los 90 días simulados.

El modelo **STREaM** devuelve los resultados en un archivo `.mat` que contiene una estructura denominada `outputTrajectory`, en la cual se almacenan los datos de cada uno de los dispositivos definidos (`StToilet`, `StShower`, `StFaucet`, `StClothesWasher`, `StDishwasher`), además de una serie `TOTAL` que realiza una suma del consumo si es que se utilizó más de un dispositivo en el mismo instante.

#### **Preprocesamiento y limpieza**

El preprocesamiento del dataset **STREaM** consta de dos etapas principales

1. La conversión del archivo `.mat` a un formato `csv`.
2. La normalización de las series para obtener eventos de consumo con la estructura utilizada en el resto del trabajo.

*Conversión desde **.mat** a CSV:* Primero, se carga el archivo `outputTrajectory.mat` generado por MATLAB y se extraen los datos de interés. Para ello se utilizó Python con la función `loadmat` de `scipy`, y con esto se accedió a la estructura del archivo `outputTrajectory` y se convirtió a un diccionario. Desde este archivo se extraen los datos correspondientes a:

- `StToilet`
- `StShower`
- `StFaucet`
- `StClothesWasher`
- `StDishwasher`

- TOTAL

Cada uno de los datos obtenidos se transforman a un vector unidimensional y se organiza en un `DataFrame` utilizando `pandas`, donde cada columna representa un dispositivo y cada fila corresponde a un paso de tiempo de 10 segundos. El índice del `DataFrame` se denomina `time_step` y representa el número de paso dentro del horizonte de simulación. Finalmente, este `DataFrame` se guarda en un archivo CSV intermedio (`outputTrajectory.all\_series.csv`), que sirve como base para el procesamiento posterior.

En este paso se verifican los siguientes aspectos:

- Todas las columnas numéricas se convierten a tipo `float` para facilitar los cálculos.
- Se mantiene explícitamente la columna `time_step` para poder reconstruir más adelante los eventos.
- Los datos de TOTAL se conservan como referencia, aunque la construcción de eventos se realiza a nivel de dispositivo.

*Normalización de las series y preparación para la construcción de eventos:* A partir del archivo `outputTrajectory.all\_series.csv`, se realiza un segundo preprocesamiento en el cual se transforman los datos de consumo en eventos discretos. Para esto, se consideran únicamente las columnas asociadas a dispositivos (es decir, todas excepto `time_step` y TOTAL) y se trabaja con una resolución temporal de:

$$\text{RESOLUCION\_S} = 10 \text{ segundos.}$$

Los valores de cada serie se interpretan bajo el modo de unidades:

```
UNITS_MODE = "L_per_step"
```

es decir, cada muestra representa un volumen consumido en litros dentro del intervalo de 10 segundos correspondiente. Se define además un umbral pequeño `EPS` para ignorar variaciones muy cercanas a cero y evitar datos que puedan ser ruido.

Antes de construir los eventos, las columnas de números se convierten a tipo `float`, para mantener la consistencia en las operaciones de suma y promedio. Con esto, cada dato de consumo por dispositivo queda listo para ser unido en un evento de consumo.

#### Unificación de formatos y estructura común

Al igual que en el caso de **WEUSEDTO**, el objetivo final es obtener un conjunto de eventos de consumo con una estructura estándar que pueda ser utilizada directamente por los modelos de desagregación y detección de anomalías. Para ello se utilizó la misma estructura que en el dataset **WEUSEDTO** que son las siguientes columnas:

- **Time**: instante de inicio del evento.
- **End\_time**: instante de término del evento.
- **Duration\_s**: duración del evento en segundos.
- **Flow\_Lmin**: caudal promedio del evento en litros por minuto.
- **Litros\_totales**: volumen total consumido durante el evento, en litros.
- **Dispositivo**: etiqueta del artefacto asociado al evento (como **ducha, inodoro, lavadora, lavavajillas, grifo**).

*Construcción de eventos a partir de los datos de consumo:* Para cada columna de dispositivo se recorren los datos de consumo, identificando conjuntos consecutivos de valores positivos.

El criterio es el siguiente:

- Cuando el valor de consumo pasa un umbral que es considerado como reposo, a un valor superior a este umbral, se considera el inicio de un evento y se registra el índice de inicio  $i_{\text{inicio}}$ .
- Mientras los valores se mantengan por sobre el umbral, el evento continúa.
- Cuando el valor cae nuevamente por debajo o igual al umbral, se considera que el evento termina en el índice  $i_{\text{término}}$ .

La duración de cada evento se obtiene multiplicando el número de muestras del tramo por la resolución temporal de 10 segundos:

$$\text{Duration\_s} = N_{\text{muestras}} \cdot \text{RESOLUCION\_S},$$

donde  $N_{\text{muestras}}$  es la cantidad de datos de medición entre  $i_{\text{inicio}}$  e  $i_{\text{término}}$  (incluyendo ambas), y  $\text{RESOLUCION\_S} = 10$  s.

Dado que en el modo "L\_per\_step" cada muestra representa directamente un volumen en litros consumido durante el intervalo de 10 segundos correspondiente, el volumen total de cada evento se calcula como:

$$\text{Litros\_totales} = \sum_{i=i_{\text{inicio}}}^{i_{\text{término}}} v_i,$$

donde  $v_i$  es el valor de consumo en el paso  $i$ .

El caudal promedio del evento se calcula dividiendo el volumen total por la duración del evento expresada en minutos:

$$\text{Flow\_Lmin} = \frac{\text{Litros\_totales}}{\text{Duration\_s}/60}.$$

*Convertir los pasos del simulador en tiempos reales y etiquetado de dispositivos:* Las columnas `Time` y `End_time` se reconstruyen a partir del índice de muestra y de la resolución temporal. Si se denota por  $k$  el índice de inicio del evento y por  $k'$  el índice final (en términos de pasos de 10 segundos), los tiempos asociados en segundos se calculan como:

$$t_{\text{inicio}} = k \cdot \text{RESOLUCION\_S}, \quad t_{\text{término}} = (k' + 1) \cdot \text{RESOLUCION\_S}.$$

Estos instantes se convierten luego tiempo real con formato de fecha y hora legible mediante una función que suma  $t_{\text{inicio}}$  y  $t_{\text{término}}$  (en segundos) a una fecha base configurada, por ejemplo:

$$\text{FECHA\_BASE} = \text{"2025-01-01"},$$

de tal forma que **Time** y **End\_time** queden con el formato "YYYY-MM-DD HH:MM:SS".

Para la columna **Dispositivo**, se realiza un mapeo desde los nombres originales del mode-

lo (StToilet, StShower, StFaucet, etc.) a etiquetas en español más descriptivas, tales como inodoro, ducha, grifo, lavadora o lavavajillas. Este mapeo se aplica al final cuando ya se generaron todos los eventos.

*Dataset final unificado de STREaM:* Una vez contruidos los eventos para cada dispositivo, todos ellos se combinan en un único DataFrame, se ordenan por la columna Time y se seleccionan únicamente las seis columnas definidas en la estructura común:

{Time, End\_time, Duration\_s, Flow\_Lmin, Litros\_totales, Dispositivo}.

El resultado es un dataset con eventos de consumo, generado a partir del modelo STREaM, que tiene la misma estructura que el dataset construido desde **WEUSEDTO**. Esto permite utilizar ambos conjuntos de datos de manera equivalente en el entrenamiento y al momento de evaluar los modelos de desagregación y en la posterior detección de anomalías.

#### **3.4 Implementación del módulo de desagregación de consumos**

La desagregación del consumo de agua consiste en asignar a cada evento detectado el dispositivo que provocó dicho consumo, permitiendo diferenciar entre ducha, grifo, lavadora, lavavajillas e inodoro. Este módulo es fundamental para el sistema, ya que la salida alimenta directamente al módulo de detección de anomalías y, por lo tanto, cualquier mejora en su precisión es beneficiosa para una detección posterior.

Para su implementación se evaluaron múltiples modelos de clasificación, distintos conjuntos de datos y dos enfoques de etiquetado los cuales son *single-label* y *multi-label*.

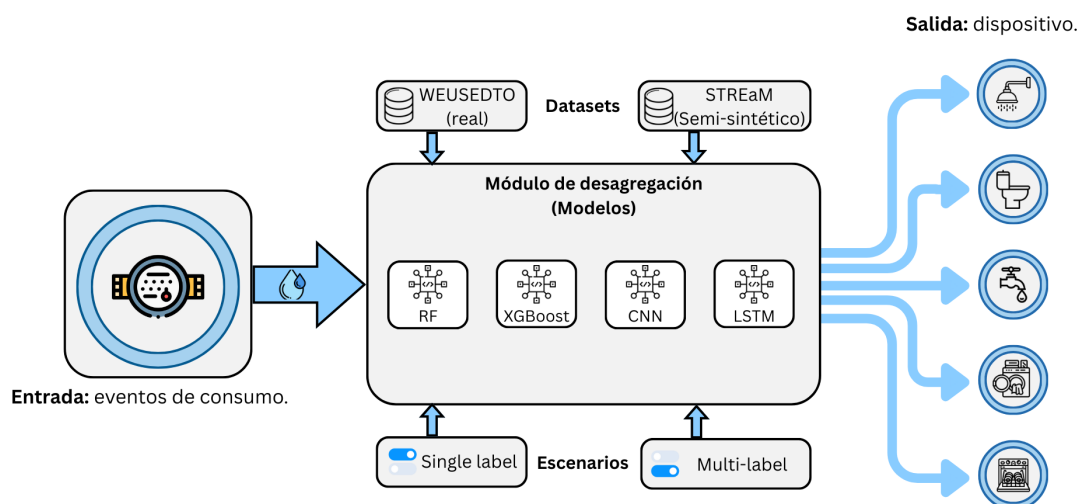


Fig. 3.3: Visión general del módulo de desagregación de consumos. A partir de los eventos de consumo detectados, se emplean distintos modelos de clasificación entrenados con datasets reales y semi-sintéticos, considerando escenarios single-label y multi-label, para identificar el dispositivo responsable del consumo.

La Figura 3.3 muestra el funcionamiento general del módulo de desagregación propuesto.

Los eventos de consumo son la entrada del sistema, los cuales son procesados por distintos modelos de Machine Learning y Deep Learning, entrenados a partir de los datasets WEUSEDTO y STREaM, generando como salida el dispositivo o conjunto de dispositivos asociados a cada evento.

#### 3.4.1 Modelos de clasificación evaluados

Para el módulo de desagregación se evaluaron cuatro modelos de Machine Learning, los cuales son:

- Random Forest:** Modelo de aprendizaje supervisado basado en un conjunto de árboles de decisión, el cual permite obtener decisiones más robustas y menos sensibles al ruido presente en datos reales. Este tipo de modelo ha sido ampliamente utilizado en tareas de clasificación sobre variables derivadas de señales temporales, como duración, volumen y caudal [23].

- **XGBoost**: Algoritmo de *gradient boosting* que construye modelos de forma iterativa, corrigiendo los errores de estimaciones previas. Su capacidad para modelar relaciones no lineales complejas lo hace adecuado para trabajar con características asociadas a eventos de consumo, como caudal máximo, duración y volumen [32].
- **Convolutional Neural Network (CNN)**: Arquitectura de redes neuronales profundas diseñada para extraer patrones locales en los datos. En el contexto de señales de consumo, las CNN permiten identificar formas características y cambios abruptos dentro de los eventos de uso de agua [25, 25].
- **Long Short-Term Memory (LSTM)**: Tipo de red neuronal recurrente orientada al análisis de secuencias temporales, capaz de modelar dependencias de largo plazo. Este enfoque resulta especialmente útil para capturar la evolución temporal del consumo dentro de un evento completo [31].

Inicialmente, los modelos se entrenaron con **single-label**, donde cada evento tiene una única etiqueta. Sin embargo, en escenarios reales, especialmente en hogares con más de una persona, es común que ocurran consumos en simultáneo. Es por esto que también se evaluó un enfoque **multi-label**, en el cual un mismo evento puede contener a más de un dispositivo.

#### 3.4.2 Proceso de entrenamiento

El proceso de entrenamiento consiste en tres etapas:

1. Preprocesamiento.
2. Configuración de modelos.
3. Definición de esquemas de evaluación.

#### Preprocesamiento

Los datos de ambos datasets fueron procesados para obtener una forma de tabla, en donde se tienen variables como la duración del evento, los litros consumidos, el caudal promedio, la hora en que ocurrió y otras métricas derivadas.

Dependiendo del modelo, se aplicó normalización MinMax o estandarización para asegurar que todas las variables tuvieran escalas comparables. Además, para el escenario multi-label se utilizó una codificación binaria que permite indicar si hay uso en simultáneo de varios dispositivos en un mismo evento.

#### Configuración de parámetros de entrenamiento

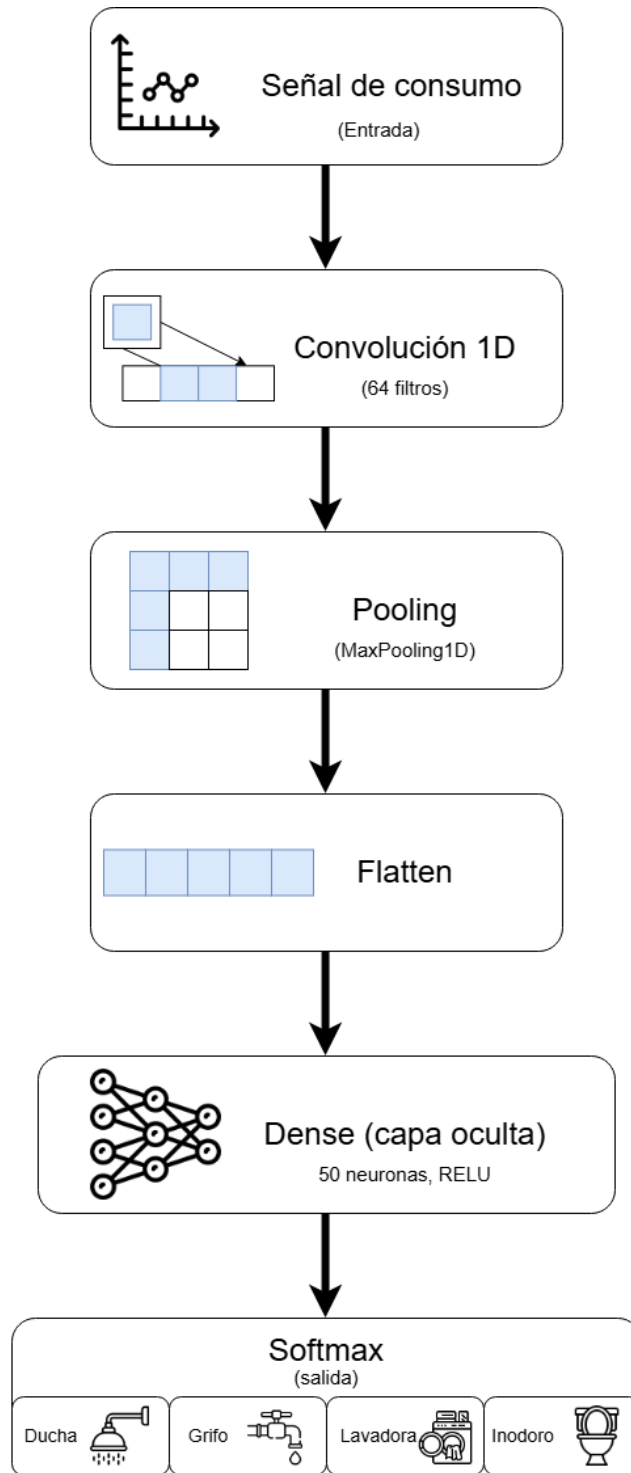
En el caso de los modelos Random Forest y XGBoost, se utilizaron configuraciones base, fijando únicamente los parámetros más relevantes para asegurar reproducibilidad y estabilidad en los resultados. No se realizó una optimización exhaustiva de hiperparámetros, dado que el objetivo principal de este trabajo es la comparación entre distintos enfoques de modelado para la desagregación del consumo de agua.

Modelo	Parámetro	Valor utilizado
Random Forest	Número de árboles ( <i>n_estimators</i> )	100
	Semilla aleatoria ( <i>random_state</i> )	42
	Paralelización ( <i>n_jobs</i> )	Todos los núcleos
XGBoost	Función objetivo ( <i>objective</i> )	multi:softprob
	Métrica de evaluación ( <i>eval_metric</i> )	mlogloss
	Semilla aleatoria ( <i>random_state</i> )	42

Tab. 3.1: Configuración de parámetros utilizados en los modelos basados en árboles

#### Arquitectura de modelos basados en Deep Learning

A continuación se describen las arquitecturas empleadas para los modelos basados en Deep Learning, las cuales permiten capturar patrones locales y dependencias temporales presentes en los eventos de consumo.



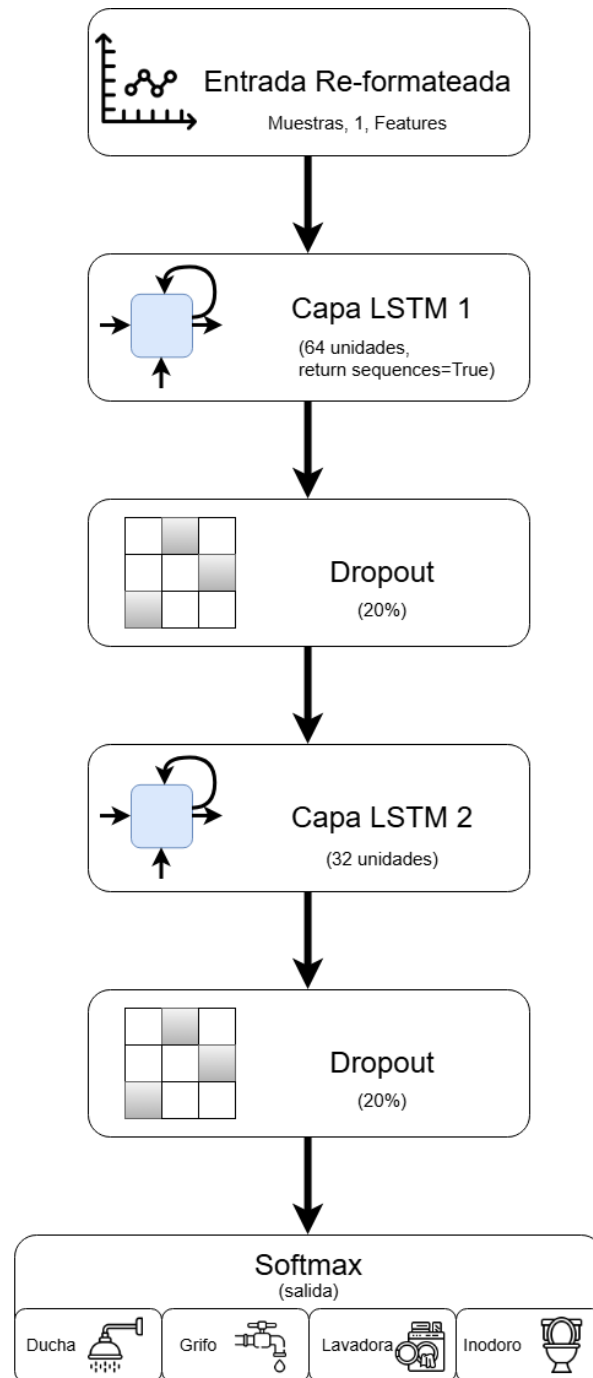
Entrenamiento: 20 epochs, batch size = 32, optimizador : adam

Fig. 3.4: Arquitectura del modelo CNN utilizado para la desagregación del consumo.

La Figura 3.4 muestra la arquitectura general del modelo CNN empleado. La señal de

### 3. DESAGREGACIÓN DEL CONSUMO DE AGUA

consumo es procesada mediante capas de convolución unidimensional que permiten extraer patrones característicos del caudal, los cuales son posteriormente transformados y clasificados mediante capas densas para determinar el dispositivo asociado a cada evento.



Entrenamiento: 20 epochs, batch size = 32, optimizador : adam

Fig. 3.5: Arquitectura del modelo LSTM utilizado para la desagregación del consumo.

Por otro lado, la Figura 3.5 presenta la arquitectura del modelo LSTM empleado en este trabajo.

#### 3.4.3 Métricas de evaluación y comparación de modelos

Para comparar el rendimiento de los modelos se emplearon métricas comúnmente utilizadas en problemas de clasificación multiclase y multilabel.

- **Accuracy:** Valor que nos indica predicciones correctas.
- **Precision por clase:** qué tan precisas son las predicciones de cada dispositivo.
- **Recall por clase:** capacidad del modelo para detectar realmente los eventos de cada dispositivo.
- **F1-score:** promedio entre precisión y recall, tanto por clase como macro-promediado.
- **Matriz de confusión:** permite visualizar de forma detallada los aciertos y errores por categoría.

### 3.5 Experimentos y análisis

En este apartado se verán los experimentos desarrollados para evaluar el desempeño del módulo de desagregación de consumos, considerando tanto el comportamiento de los modelos en el entrenamiento como la capacidad de generalización frente a distintos tipos de datos que los modelos tienen.

El objetivo principal de esta etapa es comparar cada tipo dataset sobre los modelos para medir el rendimiento, evaluar distintos parámetros en el proceso de clasificación y analizar qué tan bien se comportan los modelos cuando se evalúan sobre datos que son distintos al utilizado durante el entrenamiento. Además, se aplican estrategias de evaluación cruzada para medir la robustez de las soluciones propuestas y se justifica la selección final del modelo que se integró en el sistema desplegado en AWS.

#### 3.5.1 Diseño experimental

El proceso de desagregación del consumo de agua tiene un diseño experimental el cual está enfocado en evaluar diferentes modelos de clasificación bajo condiciones que sean controladas y comparables. Este diseño considera la definición de los conjuntos de datos utilizados, las características extraídas, los modelos seleccionados y las configuraciones de entrenamiento. El objetivo principal es analizar el desempeño de cada modelo tanto en escenarios reales como sintéticos, así como estudiar su capacidad de generalización.

#### Conjuntos de datos utilizados y sus características

Para los experimentos se utilizaron las dos fuentes principales de datos:

- **WEUSEDTO**: dataset real con lecturas de caudal a resolución de 1 segundo, etiquetado manualmente por dispositivo. Incluye registros entre febrero de 2019 y noviembre de 2020.
- **STREAM**: dataset semi-sintético con resolución de 10 segundos, construido a partir de patrones típicos de uso.

Tanto el dataset WEUSEDTO como el de STREaM, tienen patrones de tiempo característicos, como pulsos cuadrados durante los eventos, lo cual permite la extracción de características como duración, caudal promedio y volumen total. La Figura 3.6 muestra un ejemplo de este comportamiento.

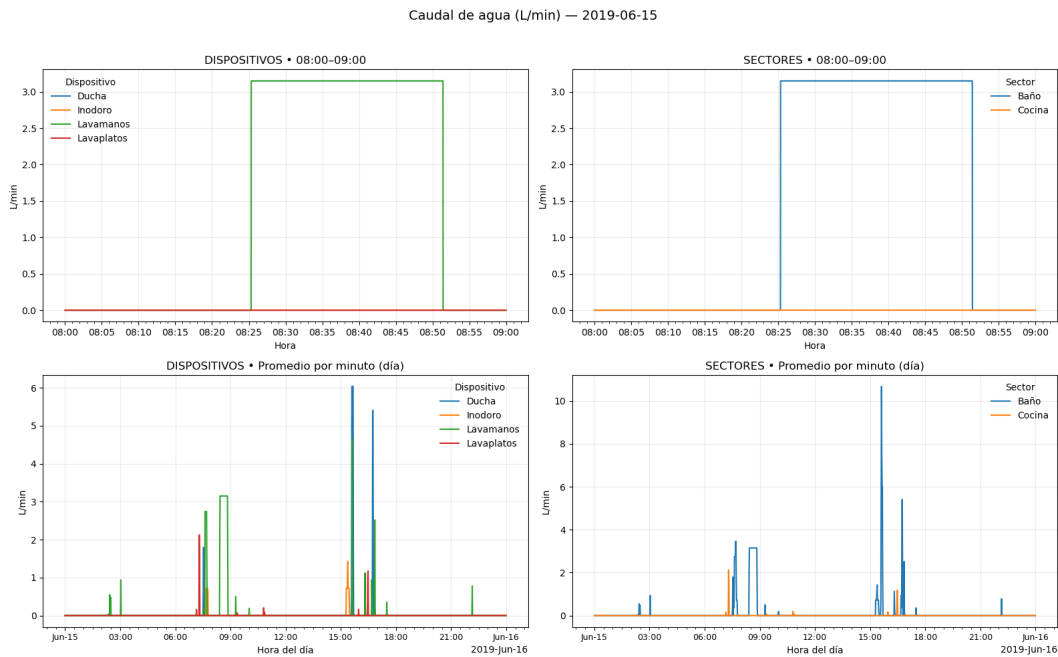


Fig. 3.6: Ejemplo de pulso cuadrado típico en el dataset WEUSEDTO.

#### Visualización exploratoria

Antes del entrenamiento de los modelos, se llevo a cabo la visualización de los consumos, para comprender el comportamiento del consumo y escoger de mejor forma las características. Entre las visualizaciones generadas se tienen:

- **Consumo mensual por dispositivo** (Figura 3.7);
- **Consumo agregado por hora del día** (Figura 3.8);
- **Pulse shape** o pulso cuadrado en un día típico (Figura 3.6).

Estas gráficas permiten entender la forma del consumo durante el tiempo y validan las características seleccionadas.

### 3. DESAGREGACIÓN DEL CONSUMO DE AGUA

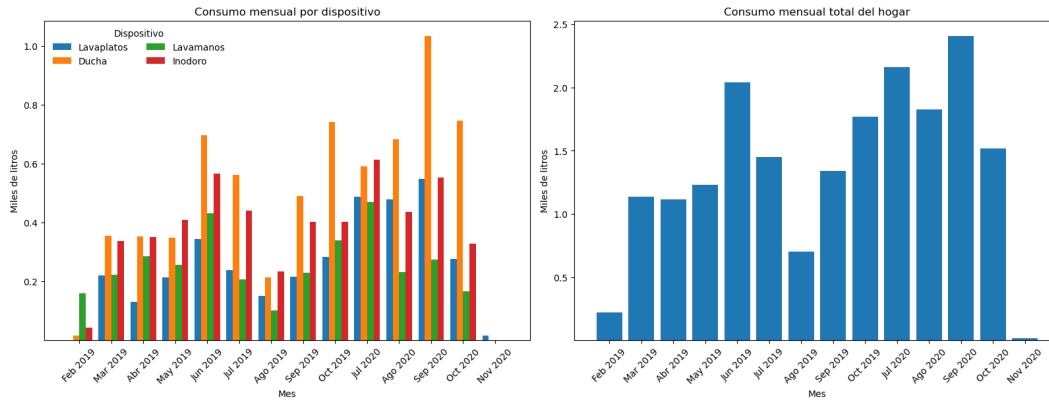


Fig. 3.7: Consumo mensual por dispositivo en WEUSEDTO.

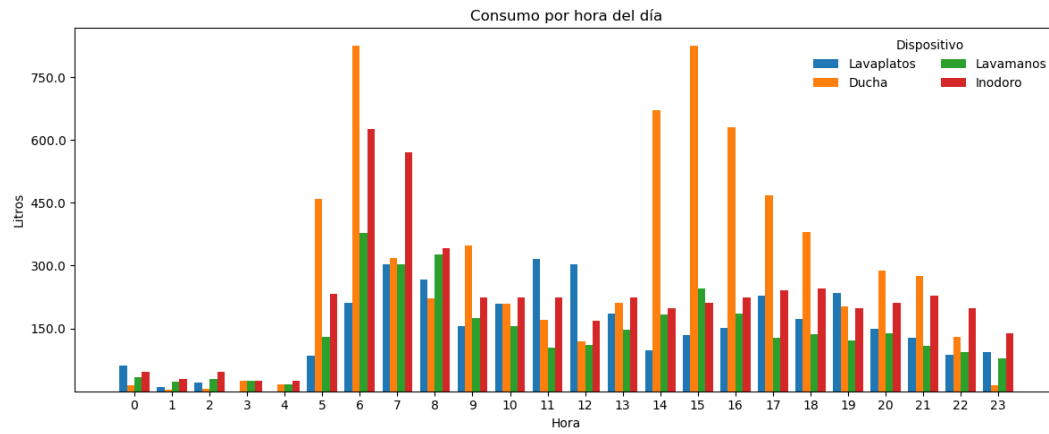


Fig. 3.8: Distribución del consumo según la hora del día en WEUSEDTO.

### 3. DESAGREGACIÓN DEL CONSUMO DE AGUA

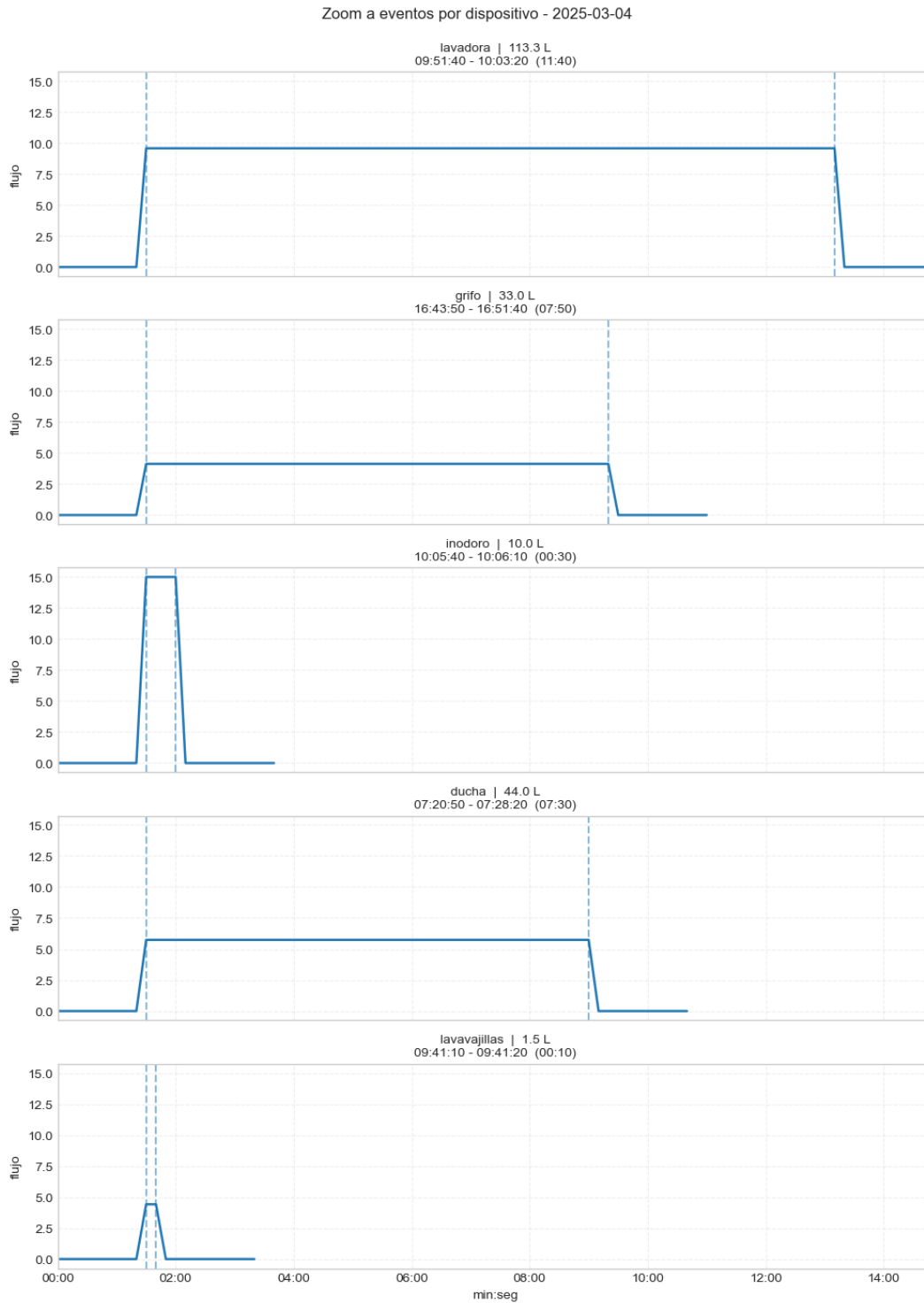


Fig. 3.9: Ejemplo de eventos individuales por dispositivo en el dataset STREaM. Todos los subgráficos comparten un eje temporal común (eje X), mientras que el eje Y representa el caudal instantáneo. Se observan diferencias claras en duración, intensidad del caudal y forma del pulso para lavadora, grifo, inodoro, ducha y lavavajillas.

La Figura 3.9 permite ver el consumo de forma detallada de los eventos para cada dispositivo. Estos patrones son especialmente útiles para entender el comportamiento de la señal:

- **Lavadora:** presenta pulsos largos y de caudal elevado, característicos de ciclos de lavado.
- **Grifo:** mantiene un flujo moderado y estable, con duraciones más breves.
- **Inodoro:** Tiene un pulso muy corto y de alta intensidad.
- **Ducha:** Tiene un flujo de mediana intensidad y larga duración.
- **Lavavajillas:** Los eventos son breves y de bajo volumen.

Esta figura permite ver las diferencias de consumo por dispositivo, mostrando las formas típicas de cada evento, lo cual refuerza la idea de la elección de características utilizadas en el entrenamiento de los modelos supervisados.

#### **Extracción de características**

El diseño experimental se basó inicialmente en un conjunto pequeño de características que representan la forma general de cada evento:

- Duración en segundos.
- Caudal promedio en litros por minuto.
- Volumen total consumido (litros).

Posteriormente, el conjunto de características se amplió incorporando información temporal:

- Hora del día de inicio del evento.
- Día de la semana.
- Variable binaria indicando si corresponde a fin de semana.

El objetivo de esta incorporación de características fue evaluar si el contexto temporal permite mejorar las predicciones de las clases o si introduce ruido para algunos modelos sensibles a la distribución de datos, como LSTM o CNN.

#### 3.5.2 Comparación entre datasets reales y sintéticos

La comparación entre los conjuntos de datos WEUSEDTO (real) y STREaM (sintético) permite probar el desempeño de los modelos cuando se entrenan con diferentes señales. Esto es fundamental para determinar qué tan bien generalizan los modelos ante diferencias en la forma temporal del flujo, la resolución de muestreo y el nivel de ruido que puede haber en los datos.

Para esta comparación se realizaron experimentos con 2 configuraciones principales:

1. Entrenamiento y prueba sobre STREaM.
2. Entrenamiento y prueba sobre WEUSEDTO.

En ambos casos se evaluaron cuatro modelos Random Forest, XGBoost, CNN 1D y LSTM, estos modelos se probarán tanto con el conjunto reducido de características (duración, caudal promedio y litros totales) como la versión extendida con información de tiempo (hora del día, día de la semana y variable de fin de semana).

#### Single-label con tres características

Para comenzar se evaluó el escenario más simple, donde cada evento corresponde a un único dispositivo (escenario *single-label*) y sólo se utilizan tres características. Los resultados muestran una diferencia clara en el rendimiento entre ambos datasets, siendo superior el desempeño obtenido con el dataset STREaM, como se muestra en la Figura 3.10.

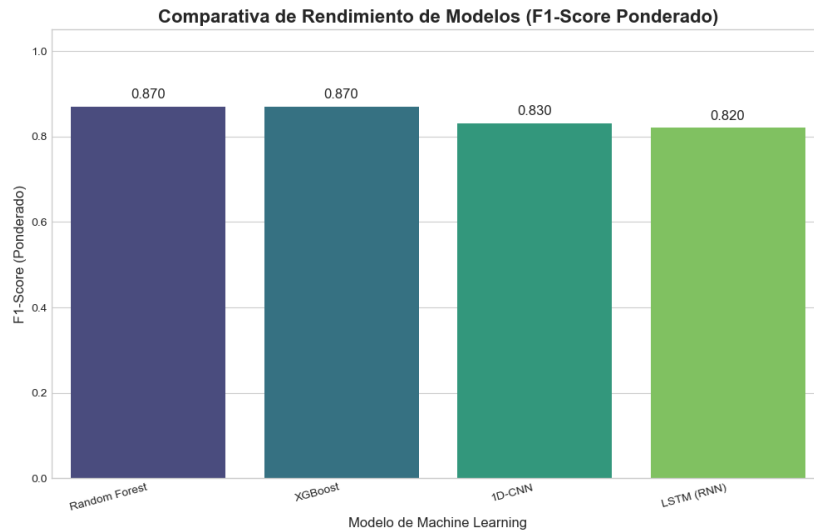


Fig. 3.10: Rendimiento de modelos en STREaM bajo configuración single-label con tres características.

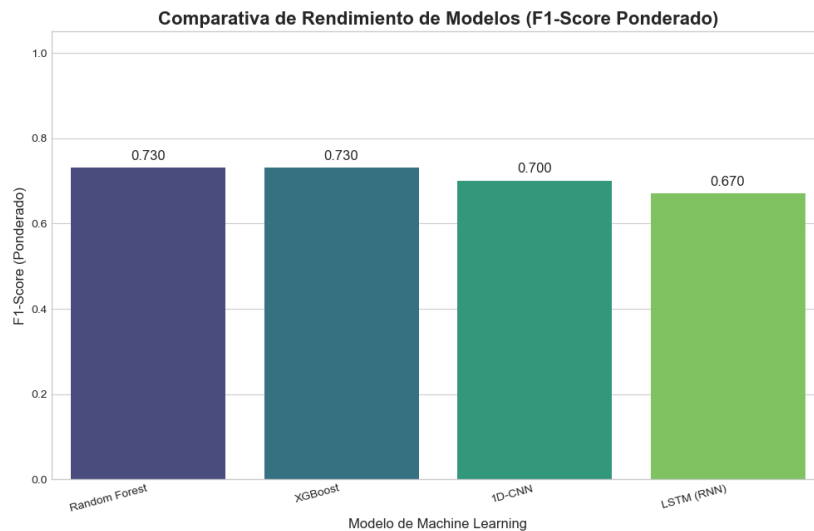


Fig. 3.11: Rendimiento de modelos en WEUSEDTO bajo configuración single-label con tres características.

Los resultados obtenidos muestran que los modelos basados en árboles, en particular Random Forest y XGBoost, alcanzan los valores más altos de F1-score ponderado en ambos datasets. Esta métrica es especialmente relevante, ya que entrega una medida equilibrada entre *precision* y *recall*, permitiendo evaluar simultáneamente la capacidad del modelo para clasificar correctamente los eventos y para no omitir casos relevantes. En términos prácticos, un F1-score elevado indica un buen compromiso entre falsos positivos y falsos negativos,

lo cual es deseable en un sistema de monitoreo orientado a identificar correctamente los dispositivos en uso.

En el caso del dataset STREaM, los valores cercanos a 0.87 obtenidos por Random Forest y XGBoost reflejan un desempeño alto y consistente, indicando modelos confiables para la desagregación bajo el escenario single-label. Los resultados en torno a 0.83 - 0.82 obtenidos por CNN y LSTM, si bien son inferiores, siguen representando un rendimiento adecuado, aunque con una mayor probabilidad de errores en la clasificación de algunos dispositivos.

Modelo	Accuracy	Precision	Recall	F1-score
Random Forest	0.87	0.87	0.87	0.87
XGBoost	0.88	0.88	0.88	0.87
LSTM (RNN)	0.84	0.81	0.84	0.82
1D-CNN	0.85	0.81	0.85	0.83

Tab. 3.2: Métricas de desempeño para el escenario single-label con tres características utilizando el dataset STREaM.

Por otra parte, en el dataset WEUSEDTO se observa una disminución general del rendimiento, con valores de F1-score en el rango de 0.73 para Random Forest y XGBoost, los cuales pueden considerarse razonables dadas las condiciones reales y la mayor variabilidad presente en los datos. Los valores obtenidos por CNN (0.70) y LSTM (0.67) evidencian un desempeño más limitado, lo que sugiere una mayor dificultad para capturar patrones discriminativos a partir de un número reducido de características en un escenario realista.

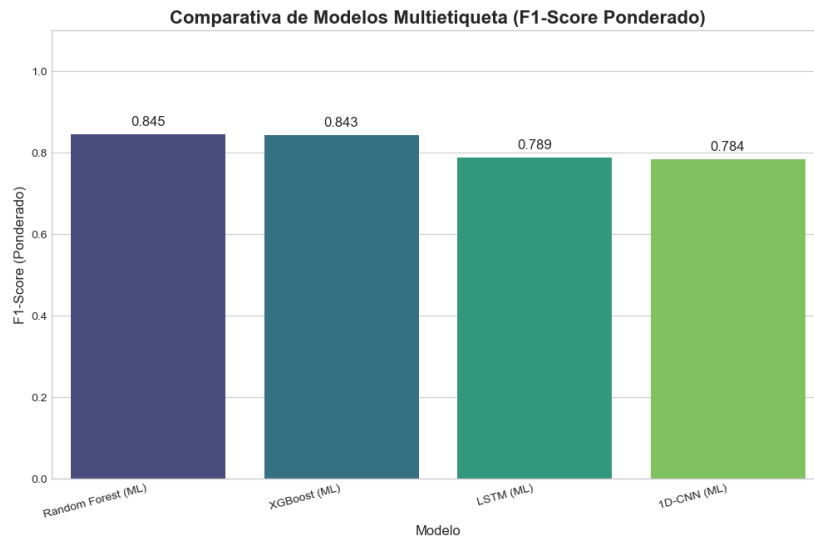
Modelo	Accuracy	Precision	Recall	F1-score
Random Forest	0.73	0.73	0.73	0.73
XGBoost	0.73	0.73	0.73	0.73
LSTM (RNN)	0.67	0.67	0.67	0.67
1D-CNN	0.70	0.70	0.70	0.70

Tab. 3.3: Métricas de desempeño para el escenario single-label con tres características utilizando el dataset WEUSEDTO.

### Multi-label con tres características

Cuando se evalúa la posibilidad de que múltiples dispositivos estén activos en un mismo evento (escenario *multi-label*), el problema de desagregación se vuelve considerablemente

más complejo. En este caso, un mismo intervalo de consumo puede asociarse simultáneamente a más de un dispositivo, lo que incrementa la superposición de patrones y dificulta la separación de las contribuciones individuales. Este aumento de complejidad se refleja en una disminución general del F1-score en comparación con el escenario single-label.



*Fig. 3.12:* Comparativa de modelos multilabel en STREaM utilizando tres características.

Tal como se observa en la Figura 3.12, en el dataset STREaM los modelos basados en árboles mantienen el mejor desempeño bajo el escenario multi-label. Random Forest y XGBoost alcanzan valores de F1-score ponderado de 0.845 y 0.843, respectivamente, lo que evidencia una buena capacidad para manejar la superposición de eventos y la coexistencia de múltiples dispositivos activos.

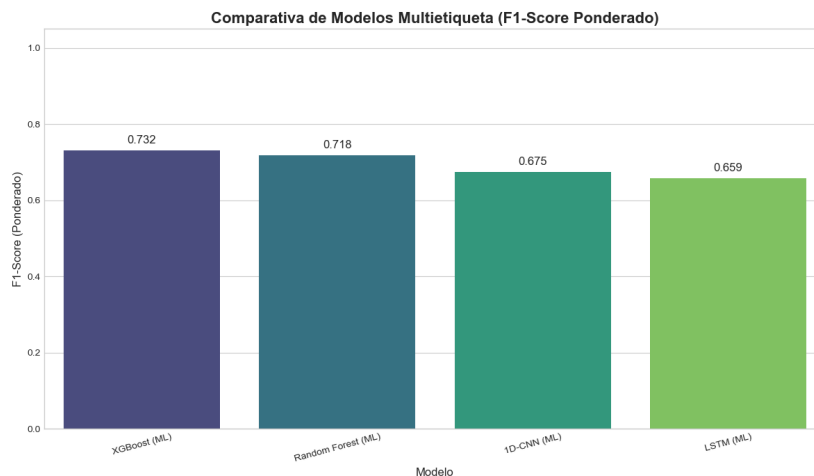


Fig. 3.13: Comparativa de modelos multilabel en WEUSEDTO utilizando tres características.

En la Figura 3.13 se aprecia una disminución adicional del rendimiento para todos los modelos al utilizar el dataset WEUSEDTO, lo cual es consistente con la naturaleza real de los datos, al poseer mayor variabilidad, ruido de medición y desbalance entre clases. En este escenario, XGBoost obtiene el mejor desempeño con un F1-score ponderado de 0.732, seguido por Random Forest con un valor de 0.718.

Por su parte, los modelos LSTM y 1D-CNN presentan los valores más bajos de F1-score ponderado, con resultados de 0.659 y 0.675, respectivamente, lo que evidencia una mayor dificultad para capturar patrones discriminativos cuando múltiples dispositivos se encuentran activos simultáneamente en escenarios reales.

Modelo	Accuracy	Precision	Recall	F1-score
Random Forest (ML)	0.818	0.867	0.828	0.845
XGBoost (ML)	0.809	0.864	0.828	0.843
LSTM (ML)	0.773	0.807	0.778	0.789
1D-CNN	0.768	0.801	0.774	0.784

Tab. 3.4: Métricas de desempeño para el escenario multi-label con tres características utilizando el dataset STREaM.

Modelo	Accuracy	Precision	Recall	F1-score
Random Forest (ML)	0.66	0.73	0.71	0.718
XGBoost (ML)	0.66	0.74	0.73	0.732
LSTM (ML)	0.61	0.70	0.65	0.659
1D-CNN	0.60	0.69	0.66	0.675

Tab. 3.5: Métricas de desempeño para el escenario multi-label con tres características utilizando el dataset WEUSEDTO.

Estos resultados confirman que los modelos basados en árboles presentan una mayor robustez frente a la complejidad introducida por el escenario multi-label, mientras que los modelos de aprendizaje profundo requieren mayor cantidad de datos y estrategias adicionales de preprocesamiento para mantener un desempeño competitivo.

### Características extendidas en configuración multilabel

La inclusión de información temporal mejora la clasificación en modelos como XGBoost y Random Forest, sin embargo podemos apreciar que introduce ruido en LSTM y CNN debido a la inclusión de estas características de tiempo.

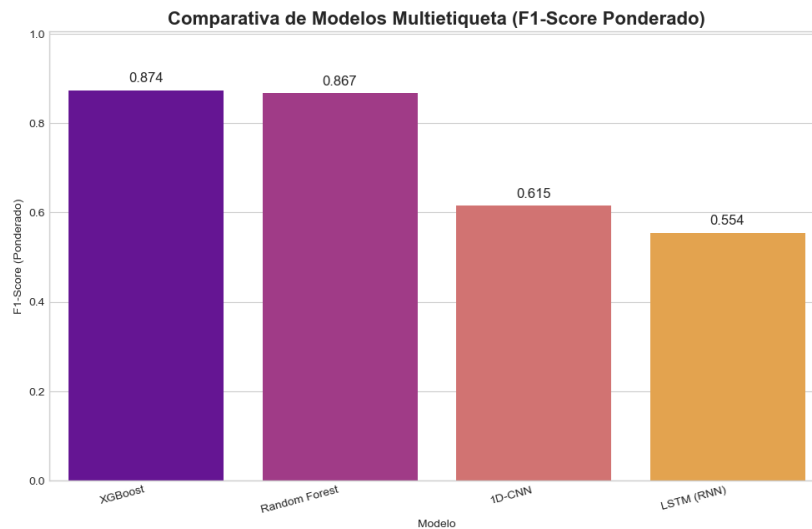


Fig. 3.14: Desempeño de modelos multilabel con características extendidas en STREaM.

### 3. DESAGREGACIÓN DEL CONSUMO DE AGUA

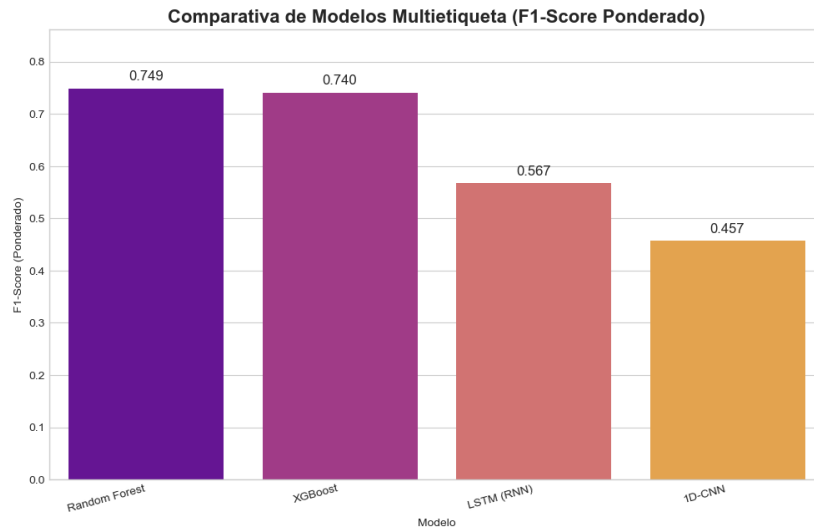


Fig. 3.15: Desempeño de modelos multilabel con características extendidas en WEUSEDTO.

Como los modelos de aprendizaje profundo son los más afectados por el aumento de dimensionalidad cuando existe un desbalance de clases, se procedió a aplicar técnicas adicionales de preprocesamiento.

#### Ajustes con oversampling y balanceo de clases

Con el objetivo de reducir el desbalance, se aplicaron técnicas de oversampling y asignación de pesos por clase. Esto permitió mejorar el rendimiento de LSTM y CNN.

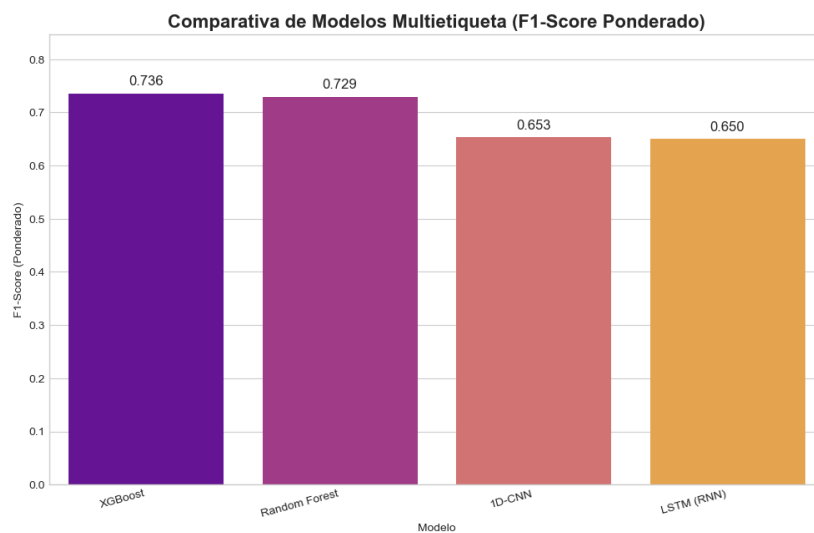


Fig. 3.16: Resultados en WEUSEDTO multilabel con características extendidas y balanceo de clases.

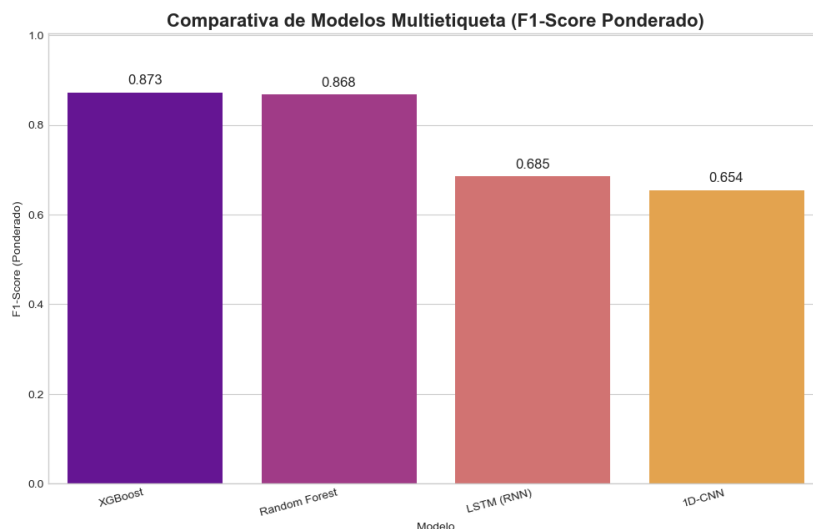


Fig. 3.17: Resultados en STREaM multilabel con características extendidas y balanceo de clases.

Estos ajustes reducen la diferencia que en los resultados que existen entre los modelos de clasificación y redes neuronales, aunque Random Forest y XGBoost continúan siendo los más consistentes en todos los escenarios.

#### Discusión global de resultados

El análisis de los distintos escenarios evaluados permite extraer conclusiones relevantes sobre el comportamiento de los modelos de clasificación y sobre la naturaleza del problema de desagregación del consumo de agua. En particular, el mejor desempeño observado en Random Forest y XGBoost frente a LSTM y CNN es un resultado esperable y consistente con las características de los datos analizados.

El consumo de agua residencial se caracteriza principalmente por eventos de corta duración, asociados a aperturas y cierres de dispositivos, con patrones simples y poco dependientes de relaciones temporales de largo plazo. En este escenario, la información más relevante se encuentra bien representada mediante características físicas agregadas, como la duración del evento, el caudal promedio, los valores máximos de flujo y el volumen total consumido. Las características empleadas corresponden a variables físicas directamente relacionadas con el flujo de agua durante cada evento, lo que permite representar el comportamiento de cada dispositivo de manera interpretable. Este tipo de representación es especialmente adecuada para modelos basados en árboles, los cuales pueden utilizar de forma eficiente estas variables

para separar las distintas clases de dispositivos.

Por el contrario, los modelos de aprendizaje profundo, como LSTM y CNN, suelen mostrar su mayor potencial cuando se tienen grandes volúmenes de datos, señales continuas con alta resolución temporal y dependencias temporales más complejas. En los experimentos realizados, la cantidad limitada de datos, el desbalance entre clases y la naturaleza agregada de los eventos dificultan que estos modelos puedan aprender representaciones más robustas, lo que se refleja en un menor desempeño, especialmente en escenarios multi-label.

En este sentido, el uso de características físicas bien diseñadas es más efectivo que el uso directo de la señal cruda del caudal, permitiendo capturar de manera más clara los patrones característicos de cada dispositivo. El mejor desempeño de Random Forest y XGBoost, reflejado en valores superiores de F1-score ponderado, debe interpretarse como una consecuencia natural de la adecuación de estos modelos al tipo de datos y al problema abordado, y no como una limitación de los enfoques de aprendizaje profundo.

Finalmente, estos resultados demuestran la importancia de seleccionar modelos que sean acordes al contexto de aplicación. Para el caso de la desagregación del consumo de agua en entornos residenciales, los modelos basados en árboles, combinados con características físicas relevantes, brindan una solución robusta, interpretable y consistente, especialmente en escenarios reales donde los datos suelen ser escasos, ruidosos y desbalanceados.

#### **3.5.3 Impacto de variaciones de características**

A diferencia de un ajuste tradicional de hiperparámetros, el principal factor que modificó el rendimiento de los modelos en este estudio fue la selección y ampliación del conjunto de características utilizadas para representar cada evento. En lugar de optimizar parámetros internos de los modelos, el análisis se centró en cómo distintos niveles de información afectaban la capacidad de clasificación.

##### **Tres características base**

El primer conjunto de experimentos utilizó únicamente tres variables agregadas:

- Duración del evento (s).

- Caudal promedio (L/min).
- Litros totales consumidos.

Estas características resumen la forma general de cada evento y permiten comparar directamente el rendimiento entre modelos tabulares y modelos basados en secuencias. Los resultados mostraron que:

- Random Forest y XGBoost obtienen un desempeño más estable debido a que estas variables capturan patrones suficientemente diferenciados entre dispositivos.
- LSTM y CNN presentan limitaciones al no disponer de la señal temporal completa, lo cual reduce la ventaja de estas arquitecturas.

#### **Características extendidas**

Posteriormente, se evaluó un segundo diseño de características que incorpora información temporal:

- Hora del día de inicio del evento.
- Día de la semana.
- Indicador binario de fin de semana.

El objetivo fue determinar si el contexto temporal aporta información útil a la clasificación. Los resultados evidenciaron que:

- La incorporación de estas características produce mejoras moderadas en los modelos de Random Forest y XGBoost.
- En los modelos de CNN y LSTM afectó de forma negativa, ya que estas características introducen ruido al ser modelos sensibles a la distribución temporal. y generan mucho desabalance en las predicciones de las clases

#### Uso de oversampling y pesos de clase

Debido a la presencia de desbalance entre dispositivos, particularmente en el dataset real, se evaluó el impacto de dos estrategias:

- **Oversampling de clases minoritarias,**
- **Asignación de pesos por clase según el error.**

Los resultados mostraron que:

- LSTM y CNN presentan mejoras más significativas al balancear la distribución, recuperando parte del rendimiento perdido.
- Random Forest y XGBoost mantienen su estabilidad, aunque con mejoras más leves.

En conjunto, estos resultados muestran que el rendimiento de los modelos depende más de las características seleccionadas y del balance de datos. Esto es consistente con investigaciones previas que muestran la sensibilidad de los modelos a los datos de entrada.

#### Síntesis del impacto de las características

El análisis de las distintas configuraciones de características y de las estrategias de balanceo refuerza las conclusiones obtenidas previamente. Aun considerando información temporal adicional y técnicas de oversampling, los modelos basados en árboles, en particular Random Forest y XGBoost, mantienen un desempeño superior frente a LSTM y CNN.

Este resultado indica que, en el problema de desagregación del consumo de agua, el rendimiento está dominado principalmente por la representación de los eventos más que por la complejidad del modelo. Con un conjunto reducido de características físicas agregadas, como duración, caudal promedio y volumen total, los modelos basados en arboles muestran un comportamiento más estable, mientras que las arquitecturas secuenciales pierden su ventaja al no disponer de la señal temporal completa.

Si bien el balanceo de clases permite mejorar parcialmente el desempeño de los modelos de aprendizaje profundo, los métodos basados en árboles conservan una mayor robustez global en todos los escenarios evaluados.

#### 3.5.4 Evaluación cruzada entre datasets

La evaluación cruzada entre datasets tiene como objetivo medir la capacidad de generalización de los modelos cuando se entrenan en un conjunto de datos y se evalúan en otro distinto, proveniente de un entorno diferente. Este enfoque no corresponde a la validación cruzada clásica utilizada para ajuste de hiperparámetros, sino a una de transferencia entre datasets, lo que permite analizar la robustez de los modelos frente a cambios en la distribución de los datos. Esta prueba es bastante relevante en este trabajo, ya que los datasets utilizados presentan diferencias en resolución temporal, comportamiento de los pulsos de consumo y distribución de eventos.

Para garantizar la consistencia entre ambos conjuntos de datos, en el caso de WEUSEDTO se unificaron las clases *lavaplatos* y *lavamanos* dentro de la categoría “grifo”. Esto para tener las mismas clases que el dataset de STREaM, para así tener comparaciones válidas en los experimentos de transferencia.

En esta etapa se realizaron pruebas bajo el enfoque *single-label*, utilizando únicamente tres características agregadas: duración del evento, caudal promedio y volumen total. Esto debido a que son las características que describen de mejor manera el comportamiento de consumo y permiten analizar la sensibilidad de los modelos frente a cambios en la distribución de los datos.

#### Escenarios evaluados

Se evaluaron dos configuraciones principales:

1. **STREaM** → **WEUSEDTO**: los modelos se entrenaron con datos sintéticos y se evaluaron sobre datos reales.
2. **WEUSEDTO** → **STREaM**: los modelos se entrenaron con datos reales y se evaluaron sobre datos sintéticos.

Estos escenarios permiten analizar qué tan dependientes son los modelos de la naturaleza del dataset y si los patrones aprendidos son transferibles entre contextos.

#### **Reducción de clases para mejorar la transferencia**

Los experimentos iniciales utilizaron todas las clases presentes en los datasets (ducha, grifo, inodoro, lavadora y lavavajillas). Sin embargo, debido a la baja cantidad de datos de algunos dispositivos como la lavadora y el lavavajillas, los modelos no lograban la generalización esperada.

Por esta razón, los experimentos se repitieron considerando únicamente las tres clases más representativas:

- Ducha.
- Grifo.
- Inodoro.

Esta reducción permitió evaluar con mayor precisión el efecto de los cambios de dominio y obtener métricas más estables en los escenarios de prueba.

#### **Resultados de la evaluación cruzada**

Los resultados se presentan mediante matrices de confusión y métricas por clase (*precision*, *recall* y *F1-score*).

1. *WEUSEDTO* → *STREaM* En este escenario, los modelos entrenados con datos reales tienden a sobreajustarse a los patrones específicos del dataset, lo que se aprecia en una disminución de la precisión al enfrentarse a los datos sintéticos del dataset *STREaM*.

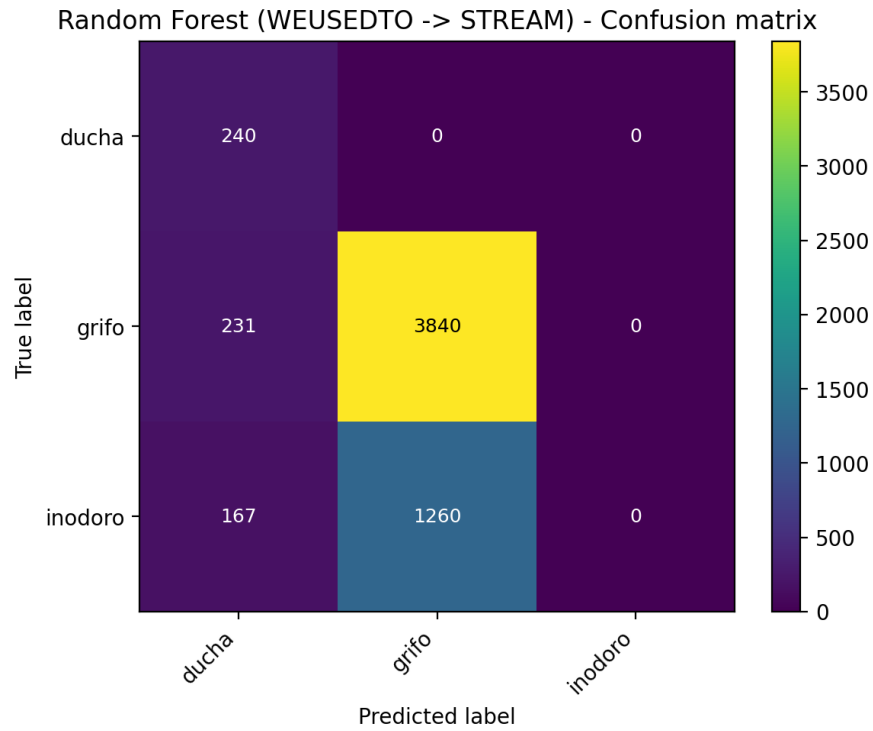


Fig. 3.18: Matriz de confusión para Random Forest entrenado en WEUSEDTO y evaluado en STREaM.

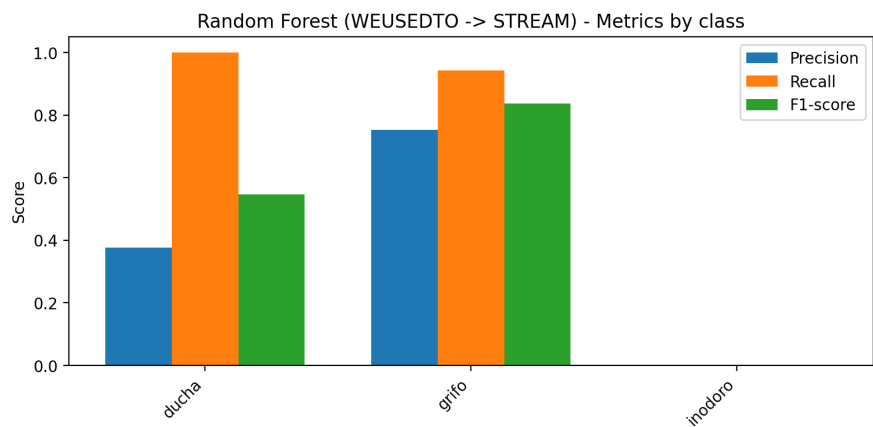


Fig. 3.19: Métricas por clase para Random Forest (WEUSEDTO → STREaM).

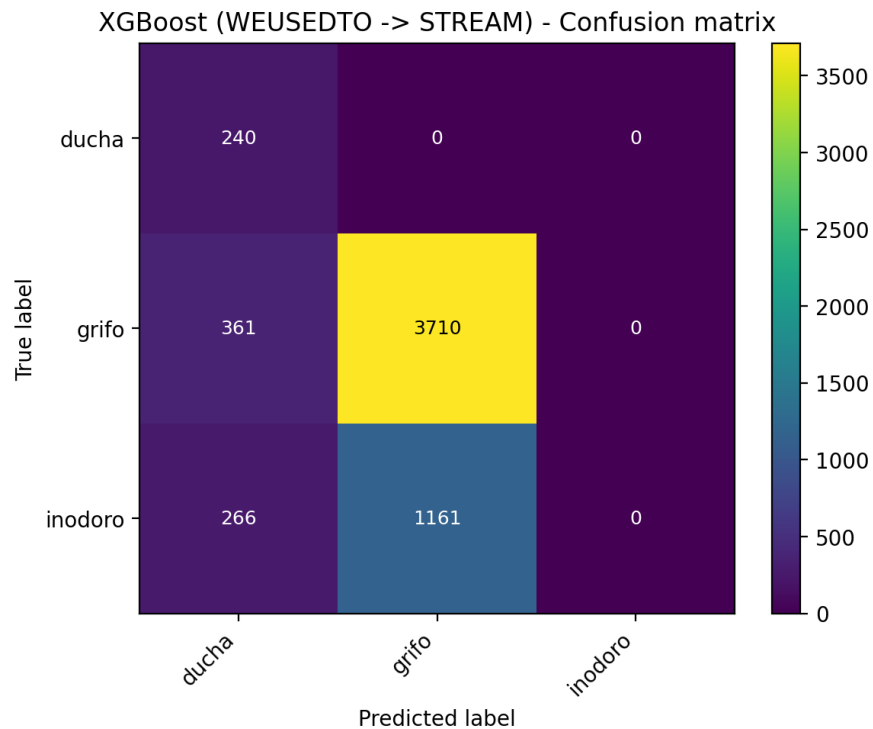


Fig. 3.20: Matriz de confusión para XGBoost entrenado en WEUSEDTO y evaluado en STREaM.

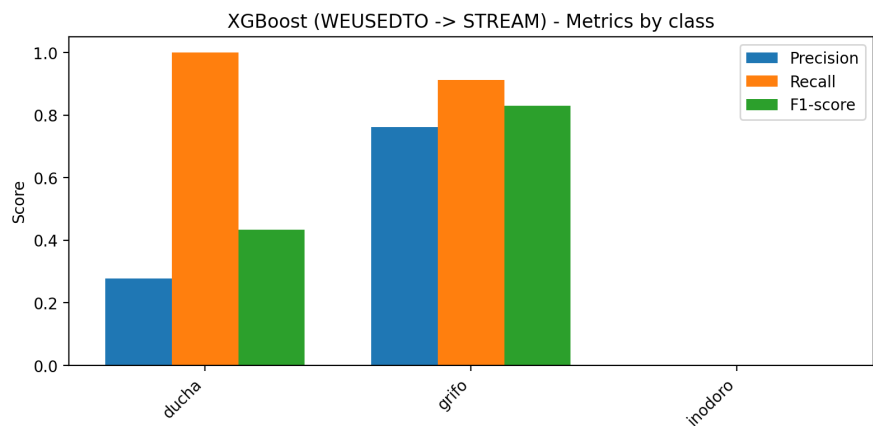


Fig. 3.21: Métricas por clase para XGBoost (WEUSEDTO → STREaM).

2. *STREaM* → *WEUSEDTO* Por otro lado, los modelos entrenados con datos sintéticos tienen una transferencia más estable hacia datos reales. Esto es debido a que *STREaM* cubre un rango más amplio de patrones, lo cual permite que los modelos aprendan de mejor forma.

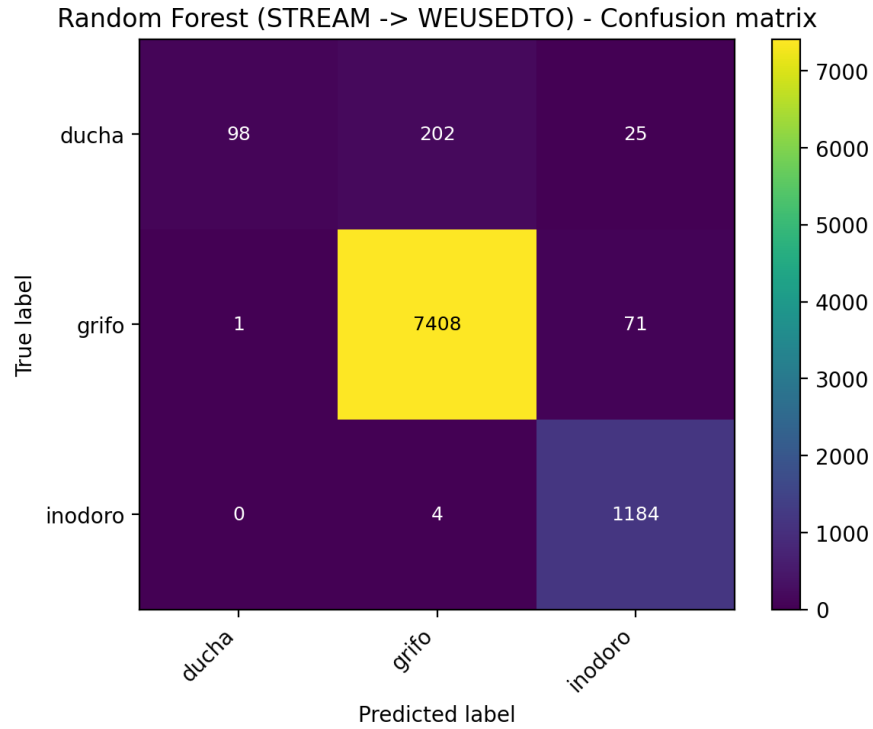


Fig. 3.22: Matriz de confusión para Random Forest entrenado en STREAM y evaluado en WEUSEDTO.

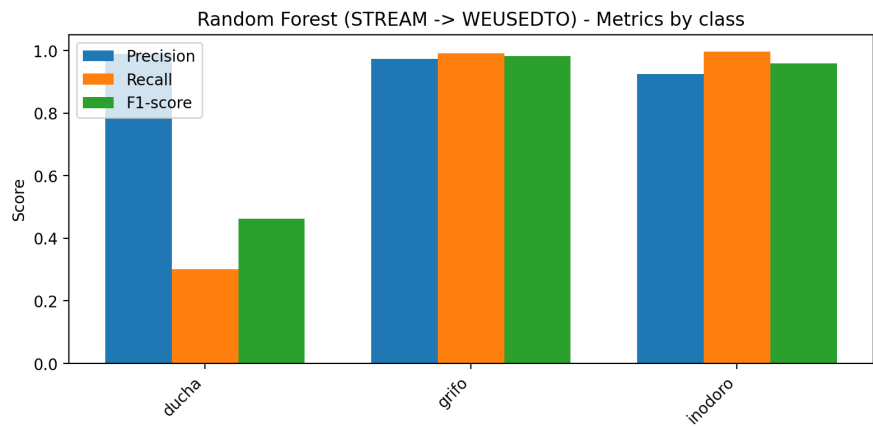


Fig. 3.23: Métricas por clase para Random Forest (STREAM → WEUSEDTO).

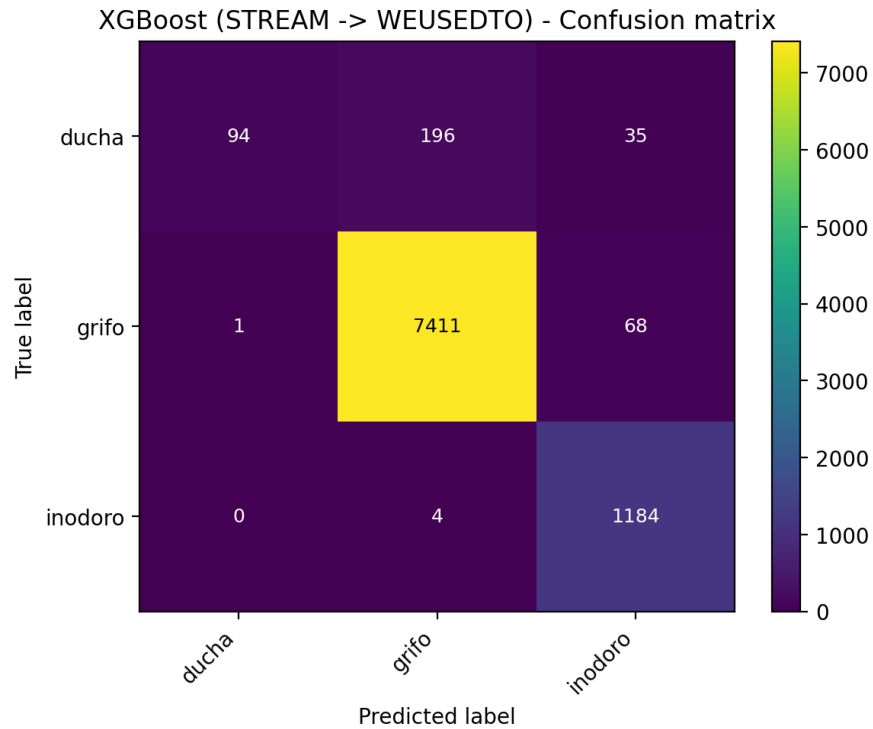


Fig. 3.24: Matriz de confusión para XGBoost entrenado en STREaM y evaluado en WEUSEDTO.

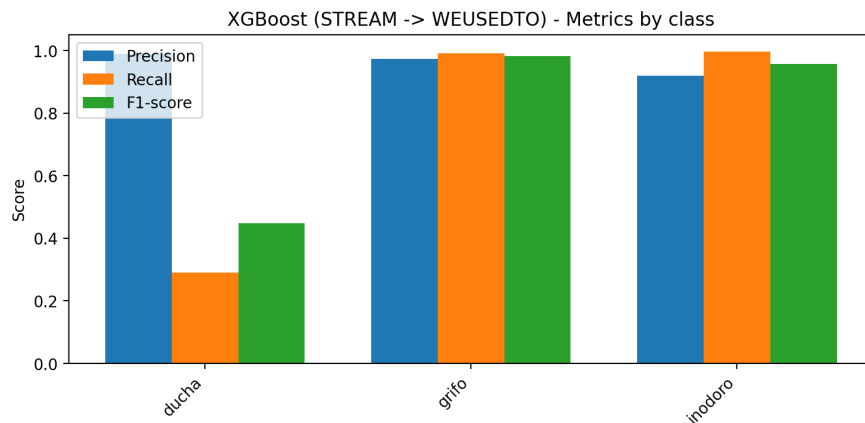


Fig. 3.25: Métricas por clase para XGBoost (STREaM → WEUSEDTO).

### Discusión de resultados

Los experimentos permiten identificar tres conclusiones principales:

1. La transferencia **STREaM → WEUSEDTO** es más estable que **WEUSEDTO → STREaM**. Esto indica que el dataset sintético posee patrones más generales, mientras que el dataset real tiene variaciones particulares del mismo hogar.

2. **La clase grifo domina la predicción en ambos sentidos.** Debido a su alta frecuencia y amplitud de valores, los modelos tienden a favorecer esta clase, lo cual se aprecia en las matrices de confusión.
3. **Reducir las clases mejora la capacidad de generalización.** Al trabajar únicamente con ducha, grifo e inodoro, se obtiene un rendimiento considerablemente superior, especialmente en escenarios donde el modelo debe adaptarse a un dataset distinto del utilizado para entrenar.

#### 3.5.5 Selección final del modelo

A partir de todos los experimentos realizados, tanto en escenarios *single-label* como en la evaluación cruzada entre datasets, se evidenció que el modelo que presenta el comportamiento más estable y consistente es **XGBoost**. Su rendimiento fue superior o comparable al de Random Forest en la mayoría de los casos, y mostró una menor variabilidad frente a cambios en el conjunto de entrenamiento y en la estructura de clases.

Un elemento clave para esta decisión fue la capacidad de generalización demostrada en los experimentos de transferencia *STREaM*  $\rightarrow$  *WEUSEDTO*, donde XGBoost mantuvo métricas más equilibradas entre clases y teniendo menor sobreajuste. Esto demuestra que el modelo captura de manera más robusta los patrones de los eventos de consumo.

Dado que el objetivo del sistema es operar en entornos reales con condiciones variables, se priorizó un modelo que ofreciera estabilidad ante cambios en los datos. En este sentido, el dataset **STREaM** se seleccionó como la base final de entrenamiento debido a su mayor cobertura de variabilidad y a que permite generar eventos con características controladas, lo cual favorece la capacidad del modelo para adaptarse a distintos hogares.

Además, XGBoost tiene otras ventajas como el tiempo de inferencia, el cual es menor a los modelos de aprendizaje profundo, su implementación es compatible con entornos ligeros y su integración AWS es más eficiente. Por ello, el modelo entrenado con STREaM y basado en **XGBoost** fue finalmente seleccionado para incorporarse al módulo de desagregación del sistema.

#### 3.5.6 Discusión de limitaciones y posibles mejoras

Si bien el modelo seleccionado tiene un rendimiento adecuado y estable, existen limitaciones. Una primera limitación corresponde a la naturaleza del dataset real (WEUSEDTO), que proviene de un único hogar con un solo integrante. Esto dificulta la obtención de un modelo capaz de generalizar.

Por otra parte, las características como duración, caudal promedio y volumen total solo muestran una descripción resumida del evento, lo que puede provocar confusiones entre dispositivos que tienen patrones similares. Si bien la incorporación de variables temporales mejora parcialmente esto, aún existe espacio de mejora.

Finalmente, la elección de XGBoost, si bien es efectiva, limita la capacidad de capturar dependencias temporales más complejas que podrían estar presentes en dispositivos como duchas o lavadoras. Una posible mejora futura es incorporar modelos híbridos que combinen lo mejor de cada modelo, o incluso explorar arquitecturas como Transformers adaptadas a series temporales.

Todas estas limitaciones abren oportunidades para avanzar hacia modelos más robustos, adaptables y capaces de operar en entornos más diversos, lo que es fundamental para una plataforma IoT desplegada a escala.

Adicionalmente, los resultados de la evaluación cruzada entre datasets permiten observar que la transferencia desde STREaM hacia WEUSEDTO es más estable que en el sentido inverso. Esto sugiere que el dataset sintético captura patrones de consumo más generales, mientras que el dataset real se encuentra fuertemente influenciado por variaciones específicas de un único hogar. Asimismo, se evidencia que la clase *grifo* tiende a dominar las predicciones en ambos escenarios de transferencia, debido a su alta frecuencia y a su amplio rango de valores, lo que induce a los modelos a favorecer esta categoría, tal como se refleja en las matrices de confusión.

Por otra parte, la reducción del número de clases a ducha, grifo e inodoro mejora de forma significativa la capacidad de generalización de los modelos, especialmente cuando se evalúan sobre un dataset distinto al utilizado durante el entrenamiento. En conjunto, estos resultados indican que la estabilidad y la capacidad de generalización dependen mayormente de la es-

estructura de los datos y de la definición de las clases que de la complejidad del modelo. Esto respalda la selección de XGBoost como modelo final, ya que presenta un comportamiento más consistente y una menor variabilidad frente a cambios en el conjunto de entrenamiento y en la estructura de clases, con un rendimiento superior o comparable al de Random Forest en la mayoría de los escenarios evaluados.

### 3.6 Integración y validación del módulo de desagregación en escenario real

La Figura 3.26 resume la arquitectura de integración del módulo de desagregación en AWS, destacando los componentes principales y el flujo de datos desde la obtención de lecturas hasta la visualización por el usuario.

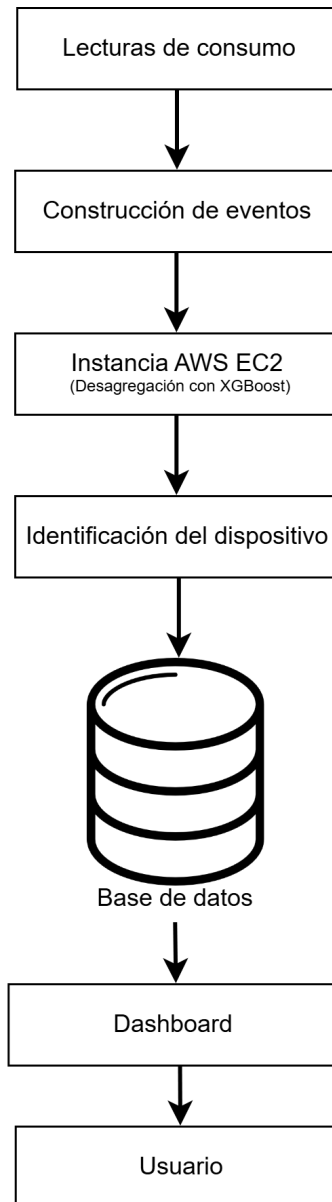


Fig. 3.26: Arquitectura de integración del módulo de desagregación en AWS y flujo de datos hacia el backend y el dashboard.

#### 3.6.1 Despliegue del módulo de desagregación en AWS

Con el objetivo de validar el funcionamiento del módulo de desagregación en un entorno real, este fue integrado dentro de la infraestructura desplegada en una instancia EC2 de Amazon Web Services (AWS). El sistema opera de manera completamente automatizada mediante un servicio y un temporizador *systemd*, ejecutándose de forma periódica sin intervención manual.

El despliegue permite procesar lecturas reales de consumo, construir eventos de uso, aplicar el modelo de clasificación entrenado y enviar los resultados al backend de la plataforma IoT, replicando condiciones de operación reales.

#### 3.6.2 Flujo operativo del pipeline de desagregación

Cada hora, el temporizador activa el servicio `water-pipeline.service`, el cual ejecuta un script maestro que coordina secuencialmente los distintos módulos del sistema. En el contexto de la desagregación, el flujo considera las siguientes etapas:

1. Obtención de lecturas de caudal desde la API del backend, las cuales se almacenan localmente en formato JSONL.
2. Unificación de lecturas y construcción de eventos de consumo a partir de periodos continuos de flujo.
3. Extracción de características físicas agregadas, tales como duración del evento, caudal promedio y volumen total.
4. Ejecución del modelo de clasificación (XGBoost) para determinar los dispositivos asociados a cada evento.
5. Envío de los eventos desagregados al backend para su almacenamiento y posterior visualización.

#### 3.6.3 Validación del módulo en entorno productivo

La Figura 3.27 muestra un ejemplo de las lecturas crudas de consumo recibidas por la instancia EC2, correspondientes a datos reales provenientes del medidor de agua. Estas lecturas

constituyen la entrada directa del módulo de desagregación.

```
ubuntu@ip-172-31-17-121:~/Monitoreo_consumo_agua/data$ cat lecturas_actual.json
{"time": "2025-10-23T17:28:00", "data": 15126.999916729354}
{"time": "2025-10-23T17:29:00", "data": 15130.41027271989}
{"time": "2025-10-23T17:30:00", "data": 15138.744946006844}
{"time": "2025-10-23T17:31:00", "data": 15147.911515131731}
{"time": "2025-10-23T17:32:00", "data": 15152.78817760499}
{"time": "2025-10-23T17:33:00", "data": 15152.946251356278}
{"time": "2025-10-23T17:34:00", "data": 15153.54416326044}
{"time": "2025-10-23T17:35:00", "data": 15162.343250334316}
{"time": "2025-10-23T17:36:00", "data": 15169.660442493225}
{"time": "2025-10-23T17:37:00", "data": 15172.53808769055}
{"time": "2025-10-23T17:38:00", "data": 15178.766854721214}
{"time": "2025-10-23T17:39:00", "data": 15184.49341619973}
{"time": "2025-10-23T17:40:00", "data": 15185.227860424866}
{"time": "2025-10-23T17:41:00", "data": 15190.688847698573}
{"time": "2025-10-23T17:42:00", "data": 15192.435888262647}
{"time": "2025-10-23T17:43:00", "data": 15198.377214423874}
{"time": "2025-10-23T17:44:00", "data": 15201.228597849129}
{"time": "2025-10-23T17:45:00", "data": 15205.648622167862}
{"time": "2025-10-23T17:46:00", "data": 15207.826906504948}
{"time": "2025-10-23T17:47:00", "data": 15208.367452793302}
{"time": "2025-10-23T17:48:00", "data": 15208.723893431477}
{"time": "2025-10-23T17:49:00", "data": 15215.657051276698}
{"time": "2025-10-23T17:50:00", "data": 15224.355229174922}
{"time": "2025-10-23T17:51:00", "data": 15226.613458022253}
{"time": "2025-10-23T17:52:00", "data": 15229.74358178084}
```

Fig. 3.27: Lecturas reales de caudal recibidas y almacenadas en la instancia EC2.

A partir de estas lecturas, el sistema construye eventos de consumo y aplica el modelo de clasificación entrenado. En la Figura 3.28 se observa un ejemplo del archivo generado con los eventos desagregados, donde se incluyen variables físicas del evento y la predicción de los dispositivos asociados.

```
ubuntu@ip-172-31-17-121:~/Monitoreo_consumo_agua/data_out$ cat eventos_enviados.json
[
  {
    "homeId": 1,
    "time": 1761239940,
    "endTime": 1761244080,
    "duration": 4140.0,
    "flow": 4.586104237288595,
    "totalLiters": 316.44119237291306,
    "predRF": "ducha,inodoro,lavadora",
    "probRF": 0.0
  }
]
```

Fig. 3.28: Eventos de consumo generados y predicciones del módulo de desagregación.

Finalmente, los eventos desagregados son enviados de forma automática al backend de la plataforma, quedando almacenados en la base de datos para su posterior análisis y visualización. La Figura 3.29 muestra el registro del envío exitoso de los eventos, confirmando la correcta integración del módulo dentro del flujo operacional del sistema.

### 3. DESAGREGACIÓN DEL CONSUMO DE AGUA

```
2025-10-24T00:29:44:00:00 @RABT
[INFO] POST http://3.90.200.123:5009/api/data/getData body={'begin': '2025-10-23T23:29:44.434Z', 'end': '2025-10-24T00:29:44.434Z', 'homeId': 1}
[ROTATE] lecturas_actual.jsonl -> lecturas_previo.jsonl
[OK] Guardado en /home/ubuntu/Monitoreo_consumo_agua/data/lecturas_actual.jsonl (30 files)
/usr/lib/python3/dist-packages/xgboost/core.py:160: UserWarning: [00:29:45] WARNING: ./src/gbm/../common/error_msg.h:80: If you are loading a serialized model (like pickle in Python, RDB in R) or configuration generated by an older version of XGBoost, please export the model by calling 'booster.save_model' from that version first, then load it back in current version. See:
https://xgboost.readthedocs.io/en/stable/tutorials/saving_model.html
for more details about differences between saving model and serializing.
warnings.warn(msg, UserWarning)
[INFO] Lecturas unificadas desde previo * actual
[OK] Eventos guardados en /home/ubuntu/Monitoreo_consumo_agua/data_out/eventos_unificados.csv
[WARN] scaler.pkl no encontrado. Prediciendo sin escalar.
[OK] Predicciones MSB guardadas en /home/ubuntu/Monitoreo_consumo_agua/data_out/eventos_unificados_pred_rf.csv
[OK] JSON (10 enviado) en /home/ubuntu/Monitoreo_consumo_agua/data_out/eventos_enviados.json
[POST] Envío: OK=1 FAIL=0
```

Fig. 3.29: Registro del envío de eventos desagregados al backend de la plataforma IoT.

Finalmente, los resultados generados por el módulo de desagregación son visualizados en el dashboard de la plataforma, permitiendo al usuario final analizar su consumo de agua tanto de forma agregada como desagregada por dispositivo. La Figura 3.30 muestra un ejemplo de esta visualización, donde se presenta el consumo total del hogar y el desglose temporal asociado a los dispositivos detectados por el sistema.



Fig. 3.30: Visualización del consumo total y desagregado por dispositivo en el dashboard de la plataforma, a partir de datos reales procesados por el módulo de desagregación.

Esta visualización confirma que el módulo de desagregación entrega información coherente y consistente con el comportamiento observado en las lecturas reales, facilitando la

### 3. DESAGREGACIÓN DEL CONSUMO DE AGUA

---

interpretación del consumo y permitiendo identificar patrones de uso de manera intuitiva. De esta forma, se valida no solo el funcionamiento técnico del módulo, sino también su utilidad práctica dentro de la plataforma IoT.

# CAPÍTULO 4: DETECCIÓN DE ANOMALÍAS EN EL CONSUMO

## 4.1 Diseño del módulo de detección de anomalías

El módulo de detección de anomalías tiene como objetivo identificar eventos de consumo que se desvían del comportamiento habitual del hogar. Como entrada recibe los eventos desagregados provenientes del módulo anterior, es decir, ventanas de consumo ya asociadas a un tipo de dispositivo. Su uso se indica en la Figura 4.1

A estos eventos se le aplica un preprocesamiento inicial y luego son evaluados mediante un algoritmo que determina si el comportamiento observado es normal o anómalo. Si el evento no es anómalo, se clasifica simplemente como un evento normal. En caso contrario, el sistema aplica una clasificación de severidad mediante lógica difusa, asignando niveles como baja, media o alta severidad.

Finalmente, los eventos detectados como anómalos y su correspondiente nivel de severidad se almacenan en la base de datos para su posterior visualización y notificación al usuario.

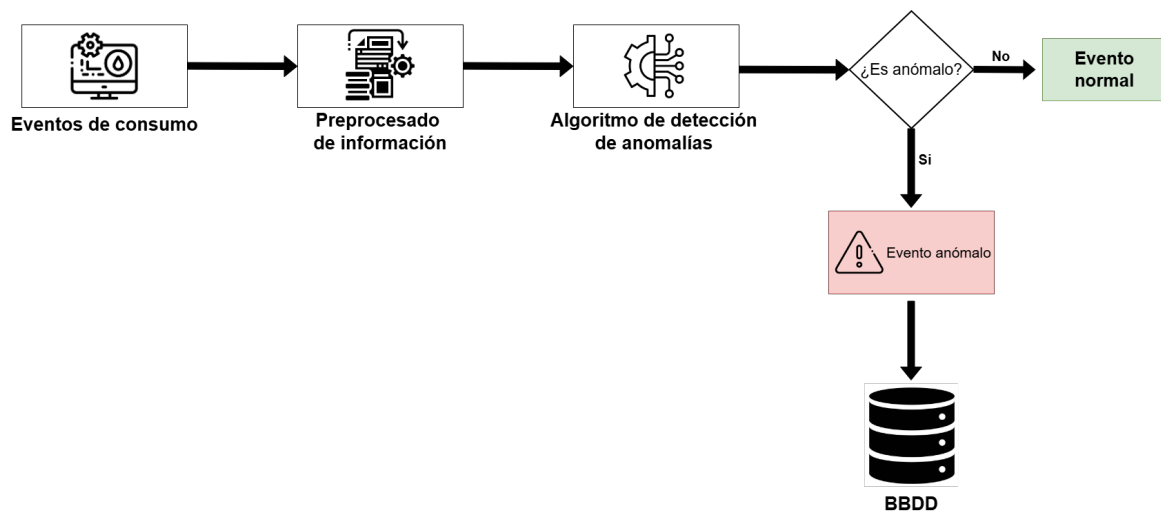


Fig. 4.1: Módulo de detección de anomalías

## 4.2 Implementación del módulo de detección de anomalías

Para la implementación del módulo de detección de anomalías se probaron 2 métodos dentro de los cuales uno de estos se basa en métodos tradicionales de análisis de datos y en

el otro se utilizan técnicas de Machine Learning

En esta investigación, se define como **evento anómalo** a cualquier consumo que cumpla con alguna de las siguientes condiciones:

- **Anomalía de Duración:** Un evento de consumo que se extiende por un periodo de tiempo excesivamente prolongado en comparación a lo normal (posible fuga o dispositivo abierto).
- **Anomalía de Litros:** Un evento cuya cantidad de litros supera significativamente los rangos normales esperados para un determinado dispositivo.

##### 4.2.1 Métodos iniciales sin Machine Learning y enfoques explorados

Para comenzar se evaluaron enfoques de análisis estadístico basados en reglas y umbrales, los cuales permitían detectar eventos anómalos sin necesidad de entrenamiento. Una de las estrategias fue utilizar percentiles (por ejemplo, P90 y P95) como límites superiores de duración o volumen de agua consumido para marcar posibles anomalías en cada dispositivo del hogar.

Sin embargo, antes de aplicar reglas basadas en percentiles, es fundamental entender las distribuciones reales del consumo por dispositivo. Para ello se analizaron las distribuciones de *litros totales* utilizando el dataset unificado de STREaM, obteniéndose los histogramas que se muestran en la Figura 4.2.

Con el fin de evaluar la capacidad de los métodos explorados para detectar comportamientos anómalos, se introdujeron de manera controlada eventos atípicos (*outliers*) en el conjunto de datos. Estos outliers corresponden a consumos artificialmente prolongados o con volúmenes significativamente superiores a los valores habituales, simulando situaciones como fugas, dispositivos abiertos por error o usos anómalos. Esto permitió analizar la respuesta de los distintos enfoques de detección frente a eventos anómalos conocidos.

#### 4. DETECCIÓN DE ANOMALÍAS EN EL CONSUMO

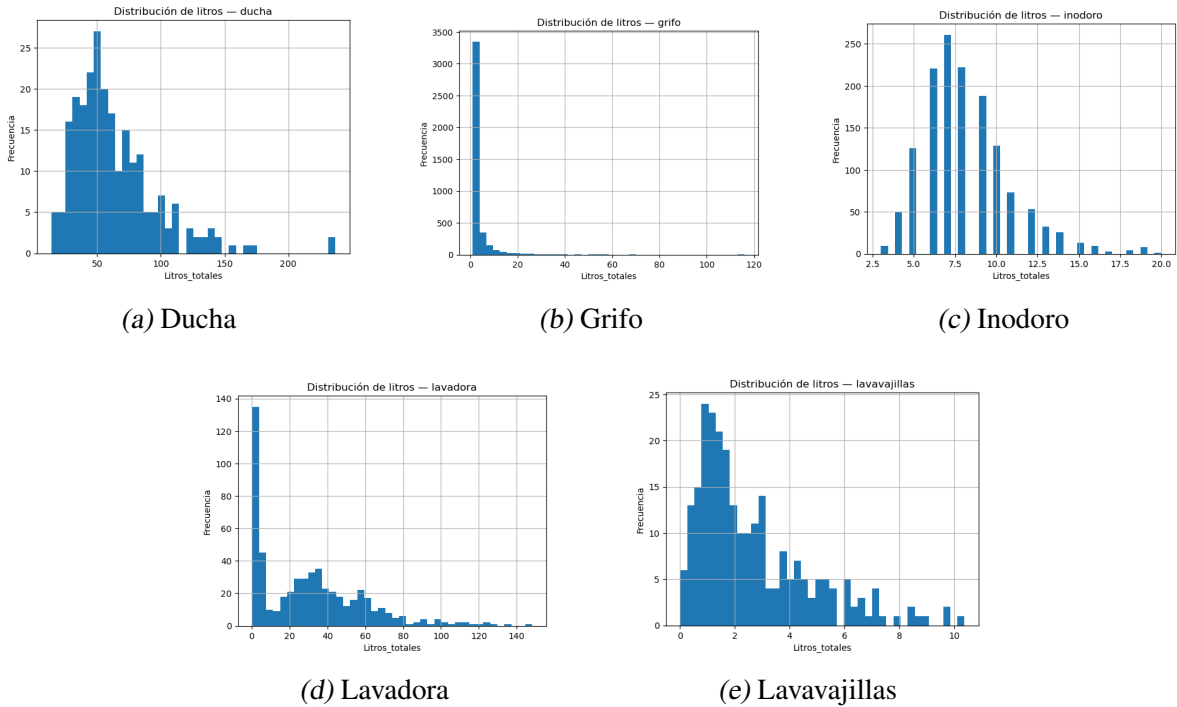


Fig. 4.2: Distribución empírica del volumen total consumido por dispositivo en el dataset unificado STREAM.

A partir de los histogramas se observa que ninguna de las distribuciones presenta un comportamiento cercano a una distribución normal. Por el contrario, se evidencian asimetrías marcadas y una alta variabilidad. En dispositivos como el grifo o la lavadora se observa una gran concentración de eventos de bajo consumo, acompañada de valores significativamente más altos, lo que refleja un comportamiento disperso. Esta variabilidad es propia del uso del agua en un entorno residencial real, donde los consumos no siguen patrones estrictos y pueden ocurrir en distintos momentos del día con intensidades muy variables.

El enfoque basado en percentiles consiste en definir umbrales superiores para cada dispositivo a partir de la distribución empírica de una variable de interés, como la duración del evento o el volumen total consumido. Un evento se considera anómalo si su valor excede un percentil previamente definido, por ejemplo P90 o P95, calculado a partir de los datos históricos del dispositivo.

**Algorithm 1** Detección de anomalías basada en percentiles

---

```
1: for cada dispositivo  $d$  do  
2:   Calcular el percentil  $P_k$  de la variable  $x$  (duración o volumen)  
3: end for  
4: for cada evento  $e$  del dispositivo  $d$  do  
5:   if  $x_e > P_k(d)$  then  
6:     Marcar evento como anómalo  
7:   else  
8:     Marcar evento como normal  
9:   end if  
10: end for
```

---

A pesar de su simplicidad, el uso de umbrales basados en percentiles elevados presenta limitaciones importantes en este contexto. En distribuciones altamente asimétricas y con colas largas, como las observadas en la Figura 4.2, los valores extremos no necesariamente representan comportamientos anómalos, sino que forman parte del comportamiento habitual de ciertos dispositivos. En estos casos, percentiles altos como P90 o P95 incluyen eventos legítimos de consumo, lo que provoca que el método clasifique erróneamente una fracción significativa de eventos normales como anómalos, incrementando el número de falsos positivos.

En particular, de los 6550 eventos analizados, el método basado en percentiles clasificó 609 eventos como anómalos, lo que corresponde aproximadamente al 10 % del total. Este valor resulta excesivo considerando que el conjunto de datos representa el consumo típico de un hogar bajo condiciones normales. Con lo cual, el módulo generaba alertas en situaciones no anómalas, afectando la utilidad práctica del sistema y aumentando el riesgo de saturar al usuario con notificaciones innecesarias.

Para complementar este análisis se presentan las tablas por dispositivo, que muestran el número de eventos y valores típicos de consumo:

Tab. 4.1: Cantidad de eventos por dispositivo en el dataset unificado STREaM.

Hogar	Dispositivo	Eventos
home_default	ducha	240
home_default	grifo	4071
home_default	inodoro	1427
home_default	lavadora	558
home_default	lavavajillas	253

Tab. 4.2: Estadísticas descriptivas por dispositivo.

Hogar	Dispositivo	Eventos	Litros mediana	Litros P90	Duración mediana (s)	Flow mediano
home_default	ducha	240	55.00	105.20	465	7.28
home_default	grifo	4071	1.00	6.00	20	6.00
home_default	inodoro	1427	8.00	12.00	60	8.40
home_default	lavadora	558	27.30	66.46	160	9.31
home_default	lavavajillas	253	1.93	5.52	30	4.50

Los resultados permiten concluir que los métodos tradicionales basados en umbrales y percentiles no son adecuados para este tipo de problemas. La falta de normalidad en las distribuciones y la alta variabilidad del consumo provocan muchos falsos positivos, lo que hace necesario buscar un enfoque más robusto y que se pueda adaptar de mejor forma. Esto justifica la transición hacia técnicas de Machine Learning orientadas a la detección de anomalías no supervisadas, lo cual se abordada en la siguiente sección.

#### 4.2.2 Implementación con Isolation Forest

Debido a que los métodos basados en percentiles generaban un número excesivo de falsos positivos (cercano al 10 % del total de eventos), se optó por implementar un modelo de detección de anomalías no supervisado basado en **Isolation Forest**. Este enfoque permite identificar eventos anómalos a partir de la facilidad con la que ciertos consumos pueden ser aislados del resto de los datos, sin requerir supuestos previos sobre la distribución de las variables<sup>1</sup>.

El algoritmo funciona construyendo un conjunto de árboles binarios aleatorios que particionan recursivamente el espacio de características. Los eventos que requieren un menor número de particiones para quedar aislados presentan una menor longitud de camino promedio dentro de los árboles y son clasificados como anómalos. Una ventaja relevante de este enfoque es la posibilidad de controlar explícitamente la proporción esperada de anomalías mediante el parámetro *contamination*.

En este trabajo, el modelo fue entrenado utilizando el dataset STREaM, fijando el parámetro de contaminación en un 2 %, valor considerablemente menor al porcentaje de eventos marcados como anómalos por el método basado en percentiles. Esta configuración permitió

<sup>1</sup> <https://datascientest.com/es/isolation-forest>

obtener un detector más estable y acorde al comportamiento esperado del consumo de agua en un entorno residencial.

El entrenamiento del modelo se realizó utilizando únicamente dos características por evento: **duration\_min** y **totalLiters**, las cuales están directamente relacionadas con la definición de anomalía adoptada en este estudio, asociada a consumos excesivamente prolongados o con volúmenes anormalmente altos. El script encargado del entrenamiento y persistencia del modelo es `train_isolation_forest.py`, mientras que la fase de inferencia se implementa en `inference_isolation_forest.py`.

La Figura 4.3 muestra un esquema general del proceso de detección de anomalías mediante Isolation Forest, desde la entrada de los eventos desagregados hasta la clasificación final de cada evento como normal o anómalo.

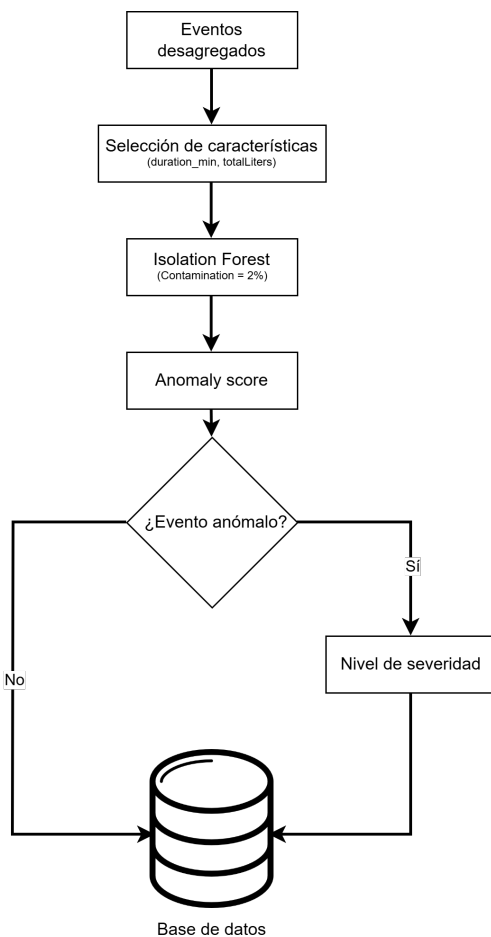


Fig. 4.3: Esquema general del proceso de detección de anomalías mediante Isolation Forest, donde los eventos desagregados son evaluados en función de su duración y volumen total para ser clasificados como normales o anómalos antes de su envío al módulo de severidad.

El modelo se entrena por cada dispositivo, con lo cual se genera un diccionario de modelos independientes para ducha, grifo, lavadora, inodoro y lavavajillas. Cada uno incluye su propio escalador y su propio bosque de aislamiento, además de sus medianas de referencia (*baselines*) almacenadas en `normal_baselines.json`. De esta manera se consigue que el detector aprenda que regiones son normales para cada dispositivo.

La Figura 4.4 muestra el resultado del modelo para el caso del grifo. Cada punto corresponde a un evento representado en el plano duración–litros, en escala logarítmica, y se indican tres elementos clave:

- **Región verde:** zona estimada como comportamiento normal por el modelo.
- **Región roja:** área que el modelo identifica como anomalía.
- **Estrella verde:** punto que representa la mediana de los eventos normales utilizados como baseline.

Este gráfico corresponde, además, a un “zoom” conceptual del comportamiento observado en el gráfico de evolución del caudal de la Figura 4.8, dado que los eventos anómalos se reflejan ahí como rectángulos marcados en color rojo.

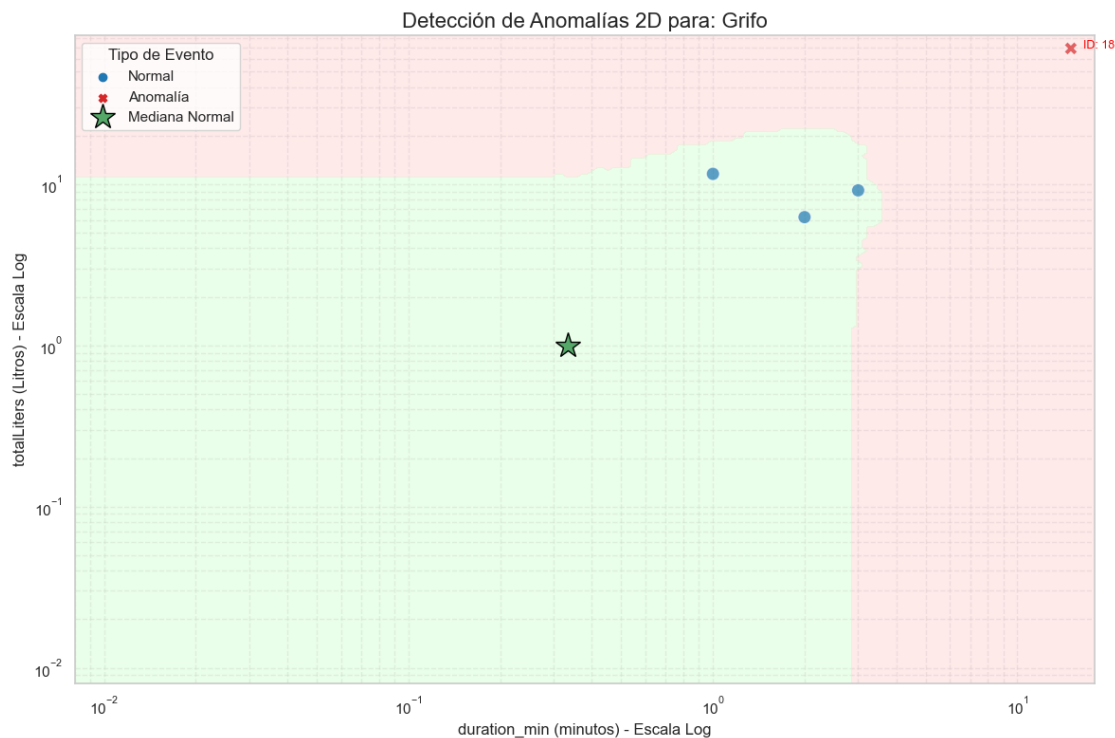


Fig. 4.4: Detección de anomalías en el espacio duración–litros utilizando Isolation Forest (dispositivo: grifo).

### 4.2.3 Complemento con lógica difusa para severidad

### 4.2.4 Complemento con lógica difusa para evaluación de severidad

Si bien Isolation Forest permite distinguir entre eventos normales y anómalos, su salida corresponde a un *score* numérico que no resulta directamente interpretable para el usuario final. Con el objetivo de traducir esta información en una evaluación comprensible del impacto del evento, se incorporó un módulo de lógica difusa para la clasificación de severidad.

La lógica difusa es una extensión de la lógica clásica que permite representar conceptos imprecisos mediante grados de pertenencia en el intervalo continuo  $[0, 1]$ , en lugar de decisiones binarias estrictas. Este enfoque resulta especialmente adecuado para sistemas de monitoreo, donde las variaciones del consumo suelen ser graduales y dependientes del contexto, y donde el uso de umbrales rígidos puede generar un número elevado de falsas alarmas [39, 40].

En el sistema propuesto, la lógica difusa recibe como entrada el *anomaly score* generado por Isolation Forest, junto con variables físicas asociadas al evento, tales como la duración y

el volumen total consumido. Estas variables son transformadas mediante funciones de membresía en conceptos lingüísticos como *bajo*, *medio* y *alto*. Posteriormente, un conjunto de reglas difusas del tipo *IF-THEN* permite evaluar el nivel de severidad del evento, replicando un razonamiento similar al humano [42].

El proceso de inferencia produce una salida difusa que es convertida en un valor final mediante un método de defuzzificación, categorizando el evento en uno de los siguientes niveles de severidad:

- **LOW.**
- **MEDIUM.**
- **HIGH.**

La Figura 4.5 presenta un esquema general del funcionamiento del módulo de lógica difusa, indicando sus entradas y salidas dentro del proceso de detección de anomalías. Cabe destacar que una descripción detallada de los fundamentos teóricos de la lógica difusa y de sus aplicaciones en sistemas de monitoreo se presenta en el Estado del Arte de este trabajo.

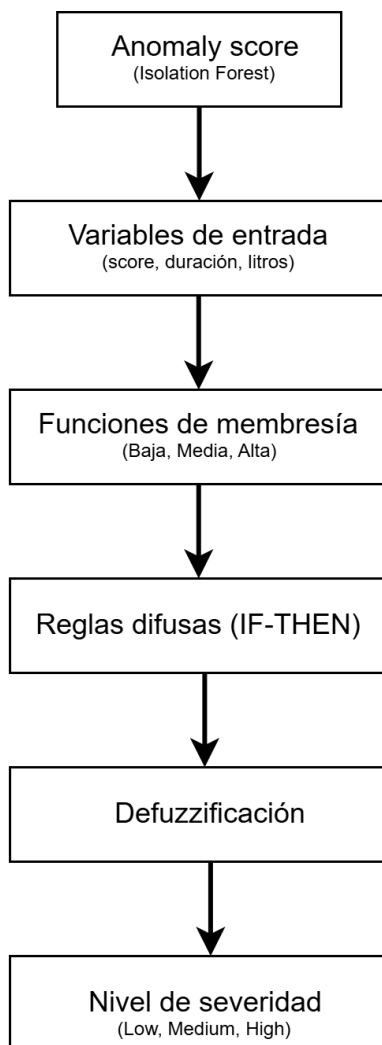


Fig. 4.5: Esquema del módulo de lógica difusa para la evaluación de severidad de eventos anómalos.

Este sistema está completamente implementado en `inference\_isolation\_forest.py`.

El sistema utiliza funciones de membresía triangulares tanto para la variable de entrada (score de anomalía generado por Isolation Forest) como para la variable de salida (nivel de severidad). Los parámetros de estas funciones no se definieron de manera arbitraria, sino que fueron ajustados empíricamente a partir de la distribución del *anomaly score* obtenida durante la etapa de entrenamiento y validación del modelo.

En particular, el rango de valores del score observado en los datos muestra que los eventos más anómalos se concentran en la cola izquierda de la distribución, con valores cercanos a  $-0,30$ , mientras que los eventos normales presentan valores próximos a 0. Sobre esta base, se

definieron los conjuntos difusos *HIGH*, *MEDIUM* y *LOW*, de modo que las transiciones entre niveles de severidad reflejen de forma gradual la separación inducida por Isolation Forest, tal como se ilustra en la Figura 4.6.

Las funciones de pertenencia para la variable de entrada `score_anomalia` se muestran en la Figura 4.6. Se observa cómo las regiones *LOW*, *MEDIUM* y *HIGH* se solapan suavemente, logrando así, que un mismo evento tenga grados de pertenencia parciales a varias categorías, lo cual da una mayor robustez frente a pequeñas variaciones del score.

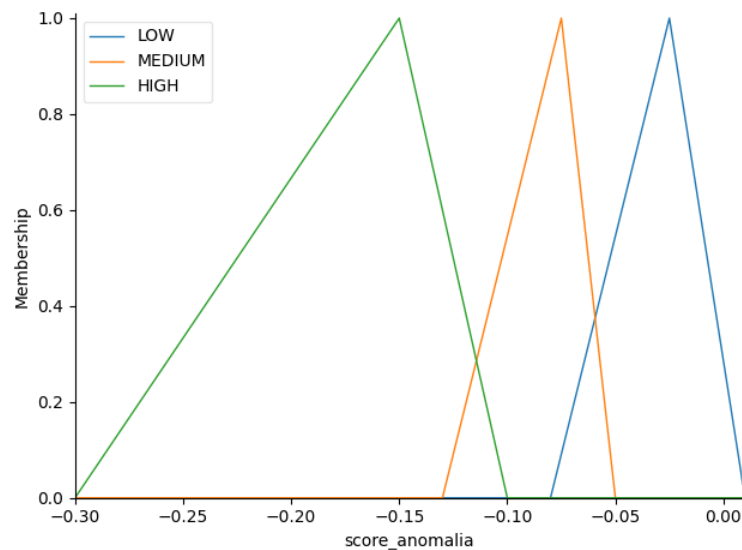


Fig. 4.6: Funciones de membresía triangulares para la entrada `score_anomalia`.

La elección de funciones de membresía triangulares se realizó principalmente por su simplicidad e interpretabilidad. Este tipo de funciones permite definir transiciones suaves entre los distintos niveles de severidad sin introducir una complejidad innecesaria en el sistema. A diferencia de otras alternativas, como las funciones gaussianas o trapezoidales, las funciones triangulares facilitan el ajuste de los rangos de severidad a partir del comportamiento observado en los datos y permiten una relación directa y comprensible entre el *anomaly score* y el nivel de severidad asignado. De esta forma, el sistema mantiene una evaluación gradual de las anomalías, evitando decisiones abruptas y por ende hay una interpretación más intuitiva de los resultados.

La variable de salida `severidad_out`, definida en el rango 0–100, también utiliza funciones triangulares (Figura 4.7). Mediante las reglas difusas:

IF score es LOW THEN severidad es LOW

IF score es MEDIUM THEN severidad es MEDIUM

IF score es HIGH THEN severidad es HIGH

se produce un valor numérico que posteriormente se discretiza en las categorías finales de severidad.

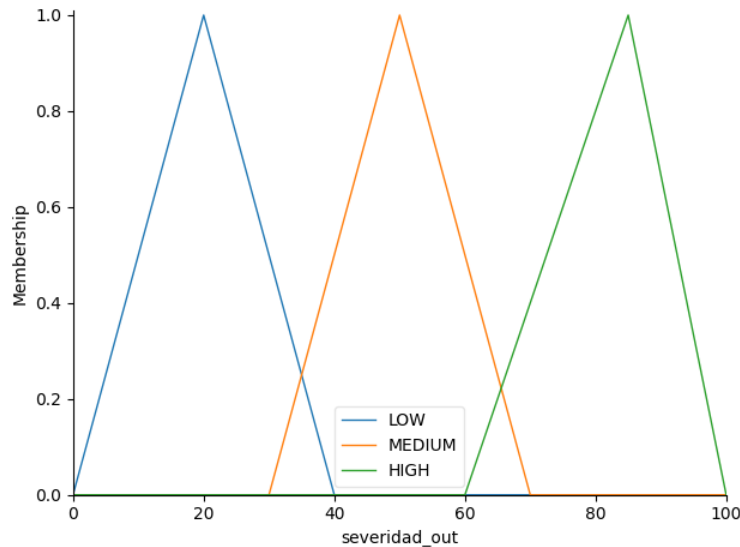


Fig. 4.7: Funciones de membresía triangulares para la salida `severidad_out`.

En la Figura 4.8 se muestran los eventos detectados durante un periodo de tiempo, donde cada rectángulo representa un evento de consumo agregado. Los rectángulos en color amarillo corresponden a eventos clasificados como normales, mientras que los rectángulos en color rojo representan eventos que el modelo Isolation Forest etiquetó como anómalos. El cálculo de severidad (LOW, MEDIUM, HIGH) se realiza de forma interna mediante la lógica difusa y se refleja en el contenido de la alerta enviada al usuario.

## 4. DETECCIÓN DE ANOMALÍAS EN EL CONSUMO

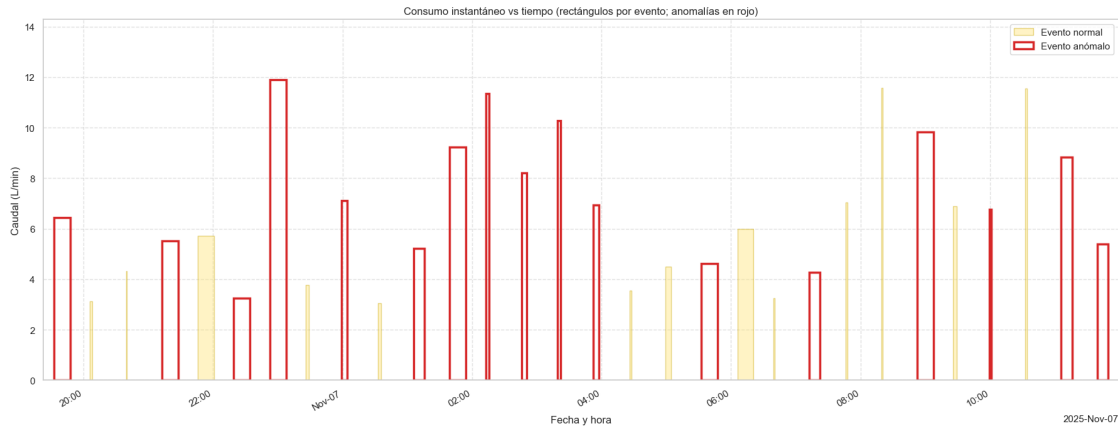


Fig. 4.8: Consumo instantáneo en el tiempo. Cada rectángulo representa un evento de consumo; los eventos en rojo son clasificados como anómalos y los amarillos como normales.

También, la Figura 4.9 corresponde a la reconstrucción del consumo del día en donde vemos que si hay un valor anómalo, este es reemplazados por el valor del baselines para el determinado dispositivo. Así es posible visualizar el impacto real de las anomalías sobre el consumo diario total.

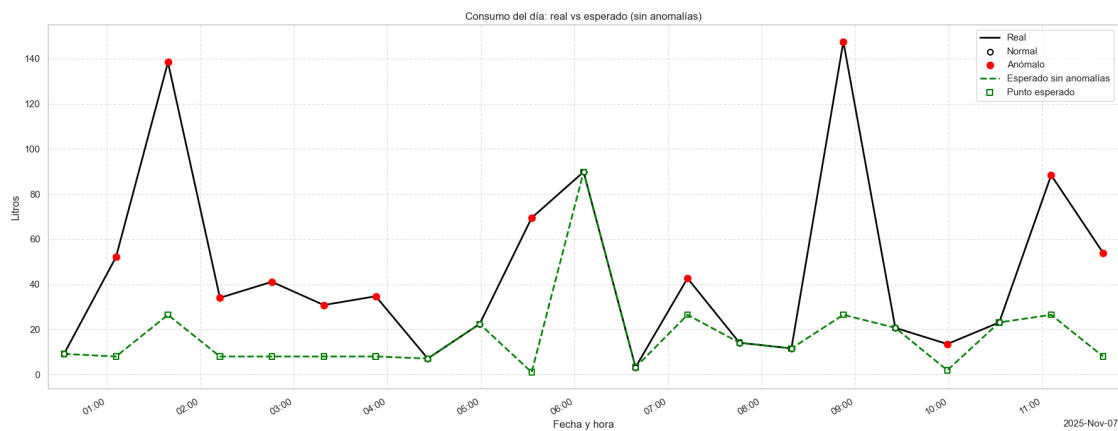


Fig. 4.9: Comparación entre consumo real y consumo esperado sin anomalías.

En conjunto, el modelo Isolation Forest y la lógica difusa permiten no solo identificar eventos atípicos, sino también clasificarlos según su severidad.

### 4.3 Integración y despliegue del módulo de anomalías en AWS

El módulo de detección de anomalías se integró dentro de un pipeline automatizado ejecutado en una instancia EC2 de AWS. Este opera mediante un servicio y un temporizador

*systemd*, lo que permite la ejecución programada y continua sin intervención manual.

La Figura 4.10 muestra un esquema general de la arquitectura del sistema y del flujo de datos, desde la adquisición de lecturas de consumo hasta la visualización de los eventos anómalos por parte del usuario.

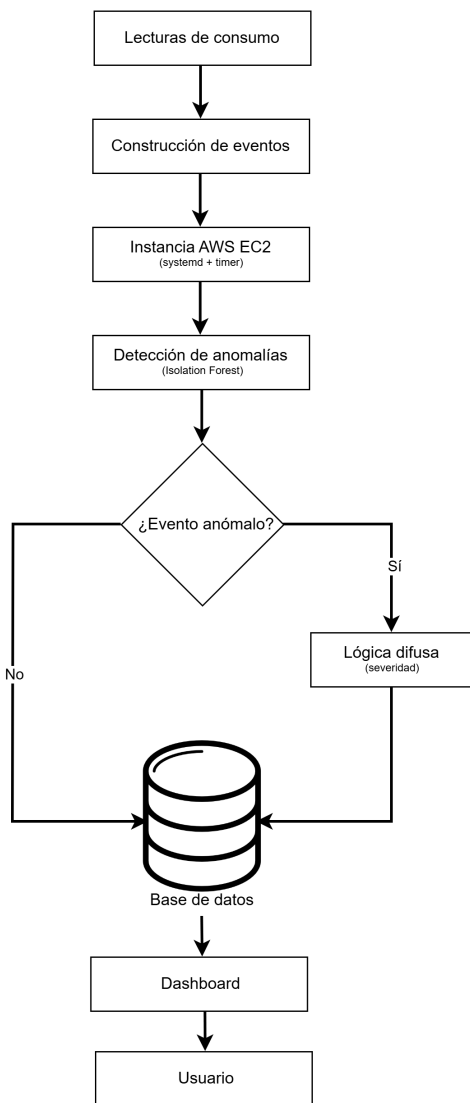


Fig. 4.10: Esquema de la arquitectura y flujo de datos del módulo de detección de anomalías desplegado en AWS.

#### Flujo general de operación

Cada hora, el temporizador activa el servicio `water-pipeline.service`, el cual ejecuta el script “maestro” `run_pipeline.sh`. Este script coordina secuencialmente los

siguientes módulos:

1. **getData.py**: obtiene desde la API del backend las lecturas nuevas de caudal.
2. **send\_predictions.py**: procesa las lecturas, unifica ventanas, construye eventos y envía al servidor los consumos desagregados.
3. **send\_alerts.py**: ejecuta el módulo de detección de anomalías, genera alertas y las envía al backend.

La integración del módulo de anomalías ocurre completamente dentro del archivo `send\_alerts.py`, el cual tiene tres fases:

1. **Ejecución de la inferencia**: En esta etapa se ejecuta Isolation Forest, la lógica difusa y la construcción del archivo `alerts.jsonl` junto al payload de visualización `viz_payload.json`.
2. **Lectura y envío de alertas**: el script se asegura que el archivo de alertas exista, carga todas las líneas del JSONL y realiza un envío individual a la ruta `/api/events/pushNotification` mediante solicitudes HTTP POST.
3. **Envío del payload de visualización**: finalmente, se envía el archivo `viz\_payload.json` al endpoint `/api/events/pushVis`, el cual contiene toda la información necesaria para que el frontend grafique los eventos desagregados y la detección de anomalías.

Este diseño permite separar la lógica de inferencia del proceso de comunicación con el backend. Además, el sistema guarda de forma local archivos JSON y logs diarios, lo que permite ver el consumo y las alertas emitidas.

### Automatización mediante systemd

La automatización del pipeline se basa en los archivos `water-pipeline.service` y `water-pipeline.timer`, instalados en el sistema. Esto garantiza:

- Ejecución puntual cada hora.

- Reinicio limpio tras cambios en los scripts.
- Aislamiento y monitoreo del proceso mediante `journalctl`.
- Control de fallos con límites de tiempo (`RuntimeMaxSec`).

Gracias a esta arquitectura, el módulo de detección de anomalías queda completamente integrado en el flujo operacional de la plataforma IoT, generando alertas y enviándolas al backend de manera automatizada, sin necesidad de supervisión manual.

### Formato y contenido del payload enviado

Los archivos generados en la fase de inferencia incluyen:

- `alerts.jsonl`: colección de alertas detectadas, una por línea.
- `viz_payload.json`: estructura unificada con eventos, baselines, predicciones y severidades, utilizadas para la visualización.

El script `send_alerts.py` se encarga de enviar ambos archivos al backend.

En caso de que alguno de los archivos no exista por ejemplo, si no hubo anomalías detectadas el sistema lo inicializa de forma vacía, evitando errores en el pipeline.

Todo lo anterior permite asegurar que el módulo de anomalías opere como un componente robusto y automatizado dentro de la infraestructura completa del proyecto.

#### 4.3.1 Validación del sistema en escenario real

El sistema propuesto fue validado en un entorno operativo real mediante su despliegue en una instancia EC2 de Amazon Web Services. En este escenario, el pipeline de procesamiento se ejecuta de forma automática y periódica, permitiendo recibir lecturas reales del medidor, construir eventos de consumo, aplicar la desagregación y ejecutar el módulo de detección de anomalías sin intervención manual.

En primer lugar, se verificó el correcto funcionamiento técnico del flujo completo de procesamiento. La Figura 4.11 muestra un ejemplo de la salida generada por el sistema durante su ejecución en la instancia EC2, donde se observan los eventos procesados junto con sus

#### 4. DETECCIÓN DE ANOMALÍAS EN EL CONSUMO

atributos principales, tales como duración, volumen total, caudal, predicción de anomalía y severidad asociada. Esta evidencia confirma que el sistema es capaz de operar de forma estable en producción, generando y enviando correctamente los resultados al backend.

```
    "totalLiters": 2.329492249220093,\n    "flow": 3.480603373842314,\n    "hour": 14,\n    "anomaly_prediction": 1,\n    "anomaly_score": 0.0,\n    "severity": null,\n    "fuzzy_severity_score": null\n  },\n  {\n    "eventId": "1762785194-2",\n    "deviceId": "lavadora",\n    "timestamp": 1762785194,\n    "timestampIso": "2025-11-10T14:33:14Z",\n    "eventDuration": 0.95,\n    "duration_min": 0.95,\n    "totalLiters": 0.3833505166694522,\n    "flow": 0.402354693962093,\n    "hour": 14,\n    "anomaly_prediction": 1,\n    "anomaly_score": 0.0,\n    "severity": null,\n    "fuzzy_severity_score": null\n  },\n  {\n    "eventId": "1762785245-3",\n    "deviceId": "bañat",\n    "timestamp": 1762785245,\n    "timestampIso": "2025-11-10T14:34:05Z",\n    "eventDuration": 0.95,\n    "duration_min": 0.95,\n    "totalLiters": 0.3833505166694522,\n    "flow": 0.402354693962093,\n    "hour": 14,\n    "anomaly_prediction": 1,\n    "anomaly_score": 0.0,\n    "severity": null,\n    "fuzzy_severity_score": null\n  }\n],\n  "totalLiters": 2.329492249220093,\n  "flow": 3.480603373842314,\n  "hour": 14,\n  "anomaly_prediction": 1,\n  "anomaly_score": 0.0,\n  "severity": null,\n  "fuzzy_severity_score": null\n}
```

Fig. 4.11: Salida generada por el pipeline de procesamiento en la instancia EC2, mostrando eventos desagregados y resultados del módulo de detección de anomalías durante la operación real del sistema.

Posteriormente, se analizó el comportamiento funcional del sistema a partir de la visualización de los resultados generados. La Figura 4.12 presenta la evolución temporal del caudal instantáneo junto con los eventos detectados durante un período de operación real, donde no se registran anomalías. Esta visualización permite comprobar que el sistema no genera falsas alertas ante consumos normales y que el comportamiento observado es consistente con el uso cotidiano del agua en un entorno residencial.

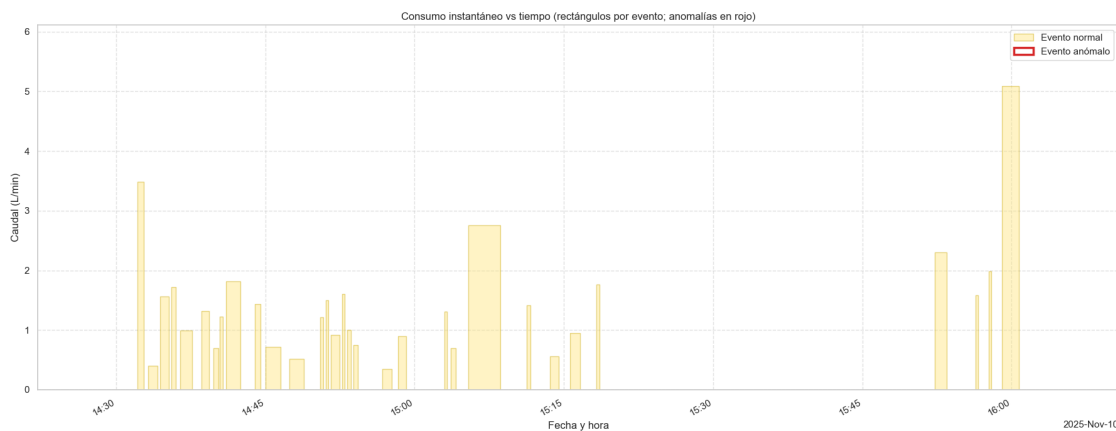


Fig. 4.12: Visualización del consumo instantáneo y eventos procesados durante un período de operación real del sistema, donde no se detectan anomalías.

En conjunto, estas validaciones demuestran que el sistema no solo funciona correctamente a nivel técnico dentro de la infraestructura AWS, sino que además presenta un comportamiento estable y coherente al enfrentarse a datos reales de consumo. Esto confirma la viabilidad del enfoque propuesto para su aplicación en escenarios residenciales reales y sienta una base sólida para futuras extensiones del sistema.

### 4.4 Discusión de ventajas, limitaciones y posibles mejoras

A pesar de que el módulo de detección de anomalías basado en Isolation Forest y lógica difusa logran reducir la cantidad de falsos positivos en comparación a los métodos tradicionales, es importante reconocer las limitaciones actuales del sistema.

#### Limitaciones del modelo

Un primer aspecto a considerar es que el modelo Isolation Forest se entrena utilizando únicamente dos características por evento *duration\_min* y *totalLiters*. Si bien estos atributos capturan el patrón general de consumo, no incorporan información como día de la semana o la hora del día. La ausencia de estas variables limita la capacidad del modelo para poder diferenciar entre consumos que no son normales pero que si son legítimos y comportamientos efectivamente anómalos en el hogar.

De esta misma manera, el modelo asume que el conjunto de entrenamiento proveniente del dataset STREaM representa fielmente el comportamiento normal de un hogar. Si bien es un dataset bueno para caracterizar patrones generales, puede que este no refleje los usos de ciertos usuarios. Esto implica que no siempre se ajustan perfectamente al ambiente real.

Otra limitación se relaciona con la interpretación del *score* generado por Isolation Forest. Este valor corresponde a una medida numérica de la rareza relativa de un evento respecto al conjunto de datos utilizado durante el entrenamiento, sin una interpretación física ni probabilística directa. Si bien este *score* es adecuado para discriminar entre eventos normales y anómalos, su significado no resulta intuitivo para el usuario final. En este contexto, la lógica difusa cumple un rol complementario al traducir dicho valor en niveles de severidad más comprensibles, aunque la definición de estos niveles sigue dependiendo de umbrales establecidos

manualmente.

#### **Limitaciones del flujo operativo**

En el entorno de producción, el sistema corre cada hora mediante un servicio *systemd* y un temporizador asociado. Esto implica que las anomalías se detectan únicamente después de que ocurren los eventos, por lo que el sistema no opera en tiempo real. Aunque esto es suficiente para usuarios de hogares residenciales, este delay puede ser crítico en escenarios donde se necesite reacción inmediata ante fugas importantes.

#### **Ventajas del enfoque propuesto**

A pesar de las limitaciones identificadas, el enfoque basado en Isolation Forest complementado con lógica difusa presenta una serie de ventajas relevantes para su aplicación en escenarios reales de monitoreo de consumo de agua.

En primer lugar, el uso de un modelo no supervisado permite detectar comportamientos anómalos sin requerir datos etiquetados, los cuales son difíciles de obtener en contextos reales y costosos de generar a gran escala. Esto hace que el sistema sea fácilmente adaptable a nuevos hogares sin necesidad de un proceso de entrenamiento manual previo.

En segundo lugar, el modelo presenta un bajo costo computacional, tanto en la etapa de entrenamiento como en la inferencia. Al utilizar un conjunto reducido de características agregadas por evento, el sistema puede operar de manera estable en infraestructuras de bajo costo, como instancias en la nube de recursos limitados, sin afectar los tiempos de respuesta.

Otra ventaja importante es la interpretabilidad del enfoque. A diferencia de modelos más complejos de aprendizaje profundo, Isolation Forest permite comprender de manera intuitiva el criterio de detección basado en la rareza de los eventos. Esta interpretabilidad se ve reforzada por la incorporación de la lógica difusa, la cual traduce el *score* numérico del modelo en niveles de severidad comprensibles para el usuario final.

Finalmente, la arquitectura modular del sistema facilita su extensión y mejora progresiva. Tanto el modelo de detección de anomalías como el módulo de evaluación de severidad pueden ser reemplazados o ajustados de forma independiente, permitiendo incorporar nuevas variables, técnicas de aprendizaje incremental o inferencia en tiempo real sin necesidad de

rediseñar completamente el sistema.

#### **Posibles mejoras**

Entre las principales oportunidades de mejora destacan:

- **Incorporación de más variables:** incluir atributos como hora del día, estación del año o consumo promedio para mejorar la distinción entre eventos normales y anómalos.
- **Entrenamiento incremental:** adaptar el algoritmo de Isolation Forest para retroalimentarse de manera periódica con datos reales del hogar, ajustando así sus valores de normalidad y disminuyendo falsos positivos residuales.
- **Inferencia en tiempo real:** integrar detección temprana directamente desde la lectura de caudal, permitiendo alertas inmediatas.
- **Mejorar la interpretación para el usuario:** Por ejemplo señalando qué tan lejos se encuentra el evento de los baselines almacenados.

Todas estas mejoras permitirían hacer que el sistema sea más preciso, adaptativo y pueda operar en tiempo real, incrementando el valor para el usuario final.

# CAPÍTULO 5: CONCLUSIONES Y TRABAJOS FUTUROS

---

## 5.1 Conclusiones

Esta investigación permitió estudiar, implementar y validar en un escenario real un sistema de desagregación del consumo de agua y detección de anomalías, utilizando dos fuentes de datos con orígenes diferentes, el dataset real WEUSEDTO y el dataset semi-sintético STREaM. A partir de múltiples experimentos y de su integración en una plataforma IoT desplegada en AWS, se identificaron los principales factores que determinan la capacidad de clasificación, la generalización de los modelos y su desempeño bajo condiciones operacionales reales.

### Conclusiones sobre la desagregación del consumo

Los resultados permiten destacar:

1. **Influencia directa del dataset en la generalización.** STREaM mostró mayor estabilidad y generalización, mientras que WEUSEDTO mostró variabilidad, lo cual es propio de un entorno real, pero esto afectó su capacidad para generalizar en otros contextos.
2. **Las tres características principales son suficientes.** Duración, caudal promedio y volumen total entregaron modelos robustos.
3. **XGBoost se consolida como el modelo más estable.** Mostró el mejor equilibrio entre precisión, consistencia y capacidad de generalización, superando a Random Forest, CNN y LSTM en la mayoría de los escenarios.
4. **Reducir clases mejora la generalización.** Al trabajar solo con ducha, grifo e inodoro se obtuvo un rendimiento mucho más estable, evidenciando que clases con baja ocurrencia dificultan el aprendizaje.
5. **Limitaciones del aprendizaje multi-label.** Incluso con oversampling y ponderación de clases, los modelos de aprendizaje profundo tuvieron menor estabilidad que XGBoost y Random Forest.

### **Conclusiones sobre la detección de anomalías**

La combinación de **Isolation Forest** y **lógica difusa** permitió construir un módulo capaz de:

- Detectar desviaciones respecto al comportamiento normal.
- Asignar niveles de severidad interpretables.
- Adaptarse a distintos patrones de consumo.

Este enfoque resultó adecuado para escenarios con ruido, variabilidad y datos incompletos.

### **Conclusiones sobre la validación en escenario real**

La integración del módulo de desagregación dentro de una infraestructura IoT desplegada en AWS permitió validar su funcionamiento bajo condiciones reales. Los resultados demostraron que el sistema es capaz de procesar lecturas reales de consumo, construir eventos, aplicar el modelo de clasificación y enviar los resultados al backend de forma automática y continua, sin intervención manual.

La validación en entorno productivo confirmó que el desempeño observado en los experimentos offline se mantiene de forma general en un escenario real, particularmente en el caso del modelo XGBoost, el cual mostró estabilidad frente a ruido de medición y desbalance entre clases. El uso de características físicas agregadas permitió además un funcionamiento eficiente, con bajo costo computacional y tiempos de respuesta compatibles con un monitoreo periódico.

No obstante, durante la operación real se observaron confusiones recurrentes entre ciertos dispositivos como lavadora y lavavajillas, lo que motivó la reducción del conjunto de clases a ducha, inodoro y grifo (incluyendo lavamanos y lavaplatos). Esta decisión permitió mejorar la estabilidad de las predicciones y reducir errores de clasificación en el entorno real.

Estas confusiones pueden atribuirse, en parte, a diferencias entre los patrones de consumo presentes en los datasets utilizados para el entrenamiento y los patrones reales observados en el contexto chileno. Mientras que STREaM representa un entorno residencial típico de

Estados Unidos y WEUSEDTO corresponde a mediciones realizadas en Italia, los hábitos de uso, caudales y configuraciones sanitarias difieren de los observados localmente, lo que afecta la capacidad del modelo para generalizar a ciertos dispositivos específicos.

En conjunto, estos resultados evidencian que la solución propuesta es viable para su uso en escenarios reales de monitoreo residencial, pero también destacan la importancia de contar con datos locales representativos y una definición adecuada de clases para lograr una desagregación más fina. Aun así, el sistema constituye una base sólida y funcional para futuras extensiones y despliegues a mayor escala.

### 5.2 Trabajos Futuros

A partir de los resultados obtenidos se hay diversas oportunidades de mejora:

- **Ampliar y diversificar los datos reales** para mejorar la robustez de los modelos.
- **Desarrollar mejores modelos multi-label**, explorando arquitecturas modernas como Transformers.
- **Aplicar técnicas de domain adaptation** que reduzcan la brecha entre datos sintéticos y reales.
- **Extender el módulo de anomalías** mediante enfoques híbridos.
- **Validar los modelos en tiempo real** para estudiar su desempeño en condiciones operacionales.

## Visión general del sistema

La Figura 5.1 presenta una visión general de la solución propuesta para el monitoreo inteligente del consumo de agua en un hogar. El sistema está basado en una solución no obstructiva, la cual implementa visión por computadora, donde una cámara instalada frente al medidor analógico captura imágenes de forma periódica para digitalizar el valor del uso del agua. Dicha información es enviada mediante un dispositivo IoT a través del protocolo MQTT hacia la plataforma en la nube.

Una vez en la infraestructura alojada en AWS, los datos son procesados para reconstruir el consumo total en el tiempo, aplicar técnicas de desagregación que permitan inferir el uso por cada dispositivo hídrico y, posteriormente, ejecutar la detección de anomalías la cual está orientada a identificar situaciones como consumos inusuales o fugas. Los resultados se almacenan en una base de datos PostgreSQL y son mostrados al usuario final a través de una aplicación web, donde es posible visualizar su consumo histórico tanto total como por cada dispositivo hídrico, además de recibir alertas en caso de un consumo anómalo.

En conjunto, este sistema integra captura de datos, comunicación IoT, análisis en la nube y herramientas de visualización, constituyendo una arquitectura modular que facilita su despliegue y futura ampliación.

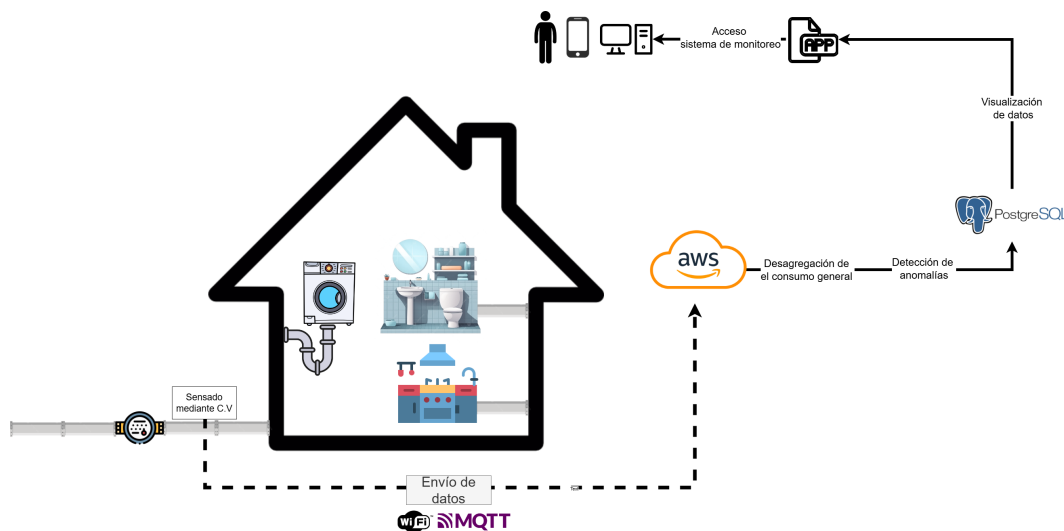


Fig. 5.1: Propuesta de solución

## Modelo de capas

El sistema propuesto se estructura mediante un modelo de capas que permite organizar sus componentes según el tipo de funcionalidad. La Figura 5.2 muestra una vista general de esta arquitectura, donde se distinguen la capa física encargada del sensado, la capa de comunicación IoT, la capa de procesamiento y servicios desplegada en la nube, y finalmente la capa de aplicación orientada a la visualización de datos y la interacción con el usuario. En las subsecciones siguientes se describe en detalle cada capa dentro del sistema.

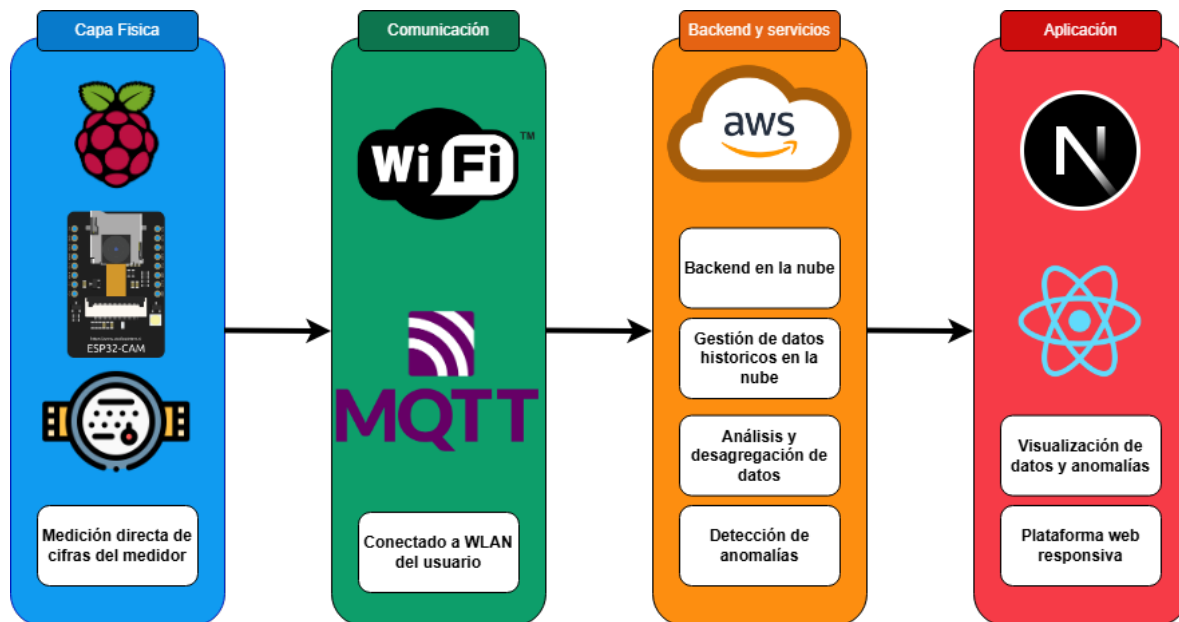


Fig. 5.2: Modelo de capas

### Capa Física

La capa física corresponde al nivel más bajo de la arquitectura y se encarga de la captura de información del medidor de agua. En esta capa están los dispositivos electrónicos y los elementos de sensado para obtener las lecturas en formato digital, sus principales componentes son:

- **ESP32-CAM:** dispositivo encargado de tomar fotografías del medidor de agua. Su función principal es capturar de forma continua la lectura visual del medidor y enviar estas imágenes a la Raspberry Pi para su procesamiento posterior.

- **Raspberry Pi:** equipo que recibe las imágenes enviadas por la ESP32-CAM y realiza el procesamiento local, incluyendo la inferencia del modelo de visión por computador para obtener la lectura digital. Una vez procesados los datos, la Raspberry Pi se comunica con el sistema externo y envía la información hacia la nube mediante el protocolo MQTT.
- **Medidor de agua analógico:** instrumento físico cuya lectura se obtiene a partir de las imágenes capturadas. El sistema transforma esta lectura visual en datos digitales de consumo.

Esta capa constituye la base operativa del sistema, proporcionando la información en bruto que permite los análisis posteriores.

### Capa de comunicación

La capa de comunicación se encarga de enviar los datos procesados desde el dispositivo instalado en el hogar hacia la plataforma en la nube, sus principales elementos son:

- **Conectividad Wi-Fi:** permite que el sistema se conecte a la red del hogar y pueda enviar la información capturada.
- **Protocolo MQTT:** permite el envío de datos para transmitir las lecturas del medidor desde la Raspberry Pi hacia la nube.

### Capa de procesamiento y análisis

En esta capa se realiza el procesamiento más importantes del sistema, aquí es donde se maneja el historial de consumos, se analizan los datos recibidos y se detectan posibles eventos anómalos. Estas funciones se ejecutan en la nube, lo que permite contar con mayor capacidad de almacenamiento y un funcionamiento continuo.

- **Backend en la nube** recibe la información enviada desde la Raspberry Pi y la almacena para su uso posterior. Aquí se mantiene el registro completo de consumos y eventos, lo que permite revisar el comportamiento del usuario a lo largo del tiempo.

- **Módulo de desagregación:** se encarga de analizar los datos y ver a que artefacto del hogar corresponde el consumo como la ducha, el lavaplatos o el inodoro.
- **Módulo de detección de anomalías:** identifica eventos que sean inusuales, como consumos excesivos o posibles fugas, comparando el comportamiento actual con lo que normalmente ocurre en el hogar.

### Capa de servicios y visualización

La capa de servicios y visualización es la encargada de mostrar al usuario final la información ya procesada, facilitando la interpretación de los datos y la toma de decisiones sobre su consumo. Se implementa mediante una aplicación web responsiva que muestra gráficos tanto de consumo históricos como por dispositivo, además de mostrar alertas en caso de un consumo anómalo. Sus elementos son:

- **Visualización de datos y anomalías:** Gráficos que muestran el consumo diario, histórico, desagregado y los eventos detectados por el sistema.
- **Plataforma web responsiva:** interfaz accesible desde dispositivos móviles y computadores, desarrollada con frameworks como Next.js y React, que permite consultar datos y recibir alertas en tiempo real.

### Diagrama de contexto

El diagrama de contexto en la Figura 5.3 muestra una visión general de cómo se relacionan los distintos componentes del sistema de monitoreo de consumo de agua. En ella se identifican los actores principales, los módulos funcionales y el flujo de información entre ellos.

En primer lugar, el proceso comienza con el dispositivo de sensado basado en visión por computadora, encargado de capturar la lectura del medidor y enviar los datos obtenidos hacia la plataforma IoT. Sobre esta información, el sistema ejecuta los módulos de desagregación del consumo y luego la detección de anomalías, los cuales permiten indicar el uso

por dispositivo y detectar si existió un comportamiento anómalo. Cuando se identifican eventos anómalos, estos son gestionados por el módulo de notificaciones para ser informados al usuario final.

La plataforma IoT centraliza todas estas operaciones, almacena los datos históricos en la base de datos en la nube y muestra la información procesada para que pueda ser visualizada mediante dispositivos de usuario tales como teléfonos móviles o computadores. Finalmente, los usuarios del hogar pueden visualizar su consumo histórico y actual, así como recibir alertas en caso de fugas o consumos inusuales.

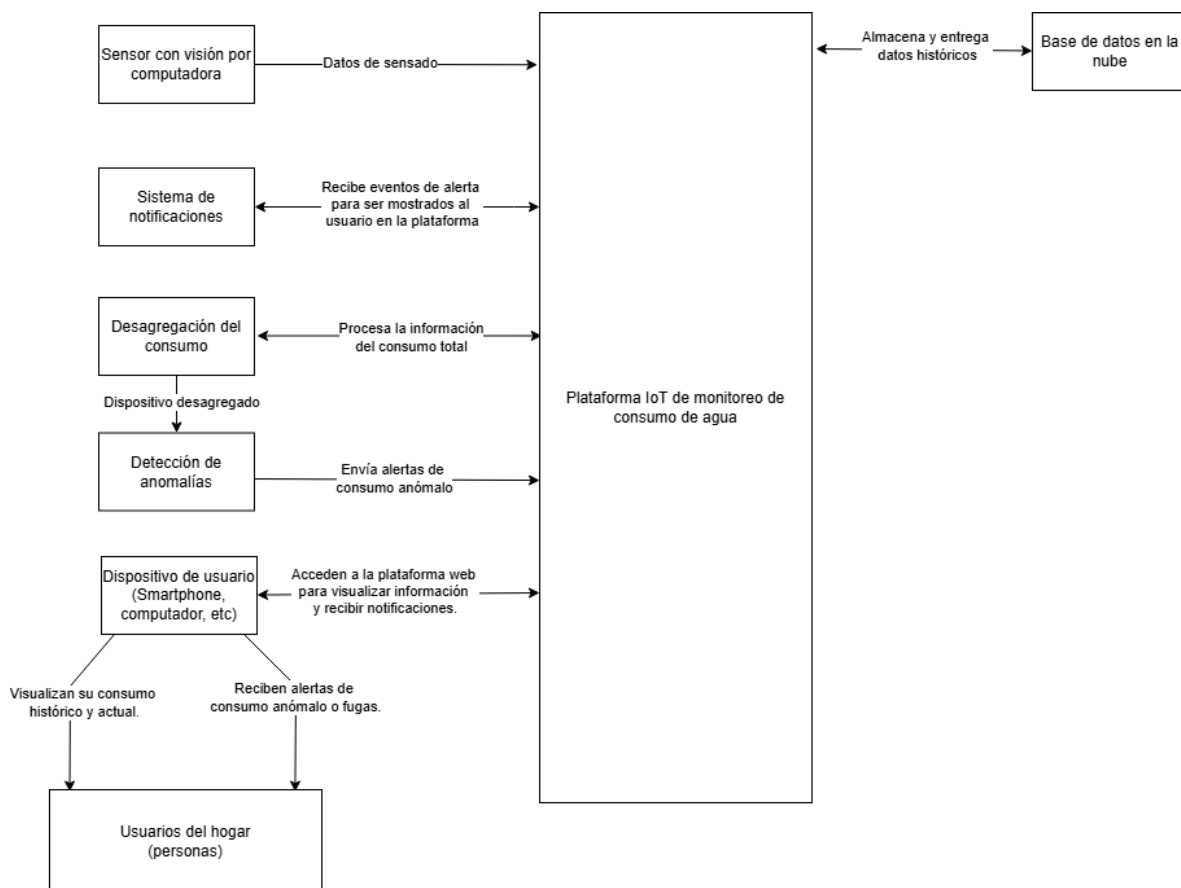


Fig. 5.3: Diagrama de contexto

## Diseño de los módulos principales

Como se mencionó en el capítulo 3 el sistema se estructura de manera modular, lo cual permite separar las funciones principales en cuatro grandes grupos:

- Módulo de sensado.
- Desagregación del consumo total.
- Detección de anomalías.
- Aplicación web

En particular, ahora se explicarán como funcionan el resto de módulos tanto el de sensado como el de aplicación web presentados en la figura 3.1.

### **Diseño del módulo de sensado**

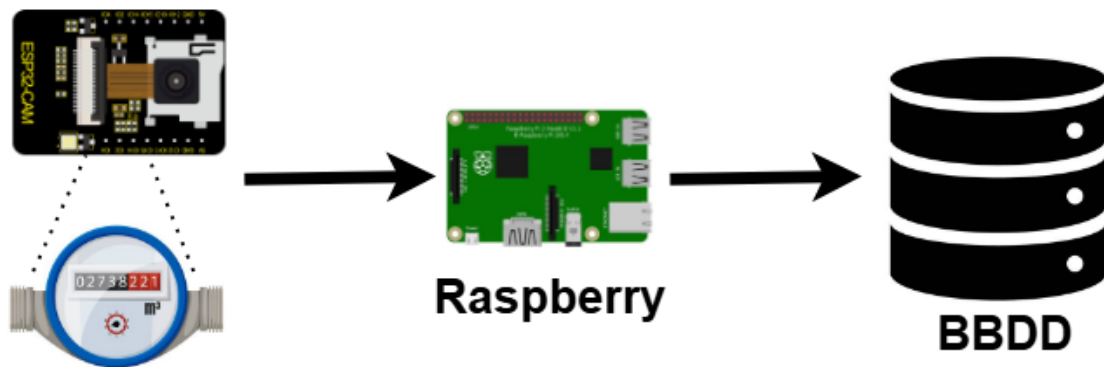
El módulo de sensado que se ve en la Figura 5.4 tiene el propósito de obtener la información del medidor de agua mediante un enfoque no obstructivo basado en visión por computadora. Este módulo tiene dos componentes principales, los cuales son la ESP32-CAM y la Raspberry Pi, estos actúan de forma simultánea para adquirir y preparar los datos que serán utilizados por los siguientes módulos del sistema.

La ESP32-CAM cumple la función de registrar imágenes del medidor de forma periódica. Su salida corresponde a una secuencia de imágenes que representa la lectura visual del consumo.

La Raspberry Pi funciona como unidad de procesamiento local dentro del módulo. Su responsabilidad es recibir las imágenes provenientes de la ESP32-CAM y transformarlas en un formato adecuado para su almacenamiento y posterior análisis.

La salida final del módulo de sensado corresponde a un conjunto de datos que se envían hacia la base de datos del sistema para permitir las etapas posteriores de desagregación y detección de anomalías.

## ESP32 CAM



*Fig. 5.4: Módulo de sensor*

### Diseño del módulo de aplicación web

El módulo de aplicación web es el encargado de que los usuarios puedan acceder a la información ya procesada por el sistema y logren visualizar de forma clara su consumo de agua. Como se muestra en la Figura 5.5, este módulo cuenta con un backend, el cual es el encargado de gestionar los datos y un frontend orientado a la presentación y navegación del usuario.

El backend, implementado con ASP.NET Core, se encarga de consultar la base de datos, obtener los registros necesarios y permitir el acceso al frontend. Aquí se construyen las funcionalidades principales, como la visualización del consumo total, el consumo desagregado por tipo de dispositivo y el sistema de notificaciones relacionado con los eventos anómalos.

El frontend, desarrollado con Next.js y React, permite a los usuarios revisar su consumo tanto histórico como actual, además de permitir la recepción de notificaciones cuando se detectan situaciones anómalas.

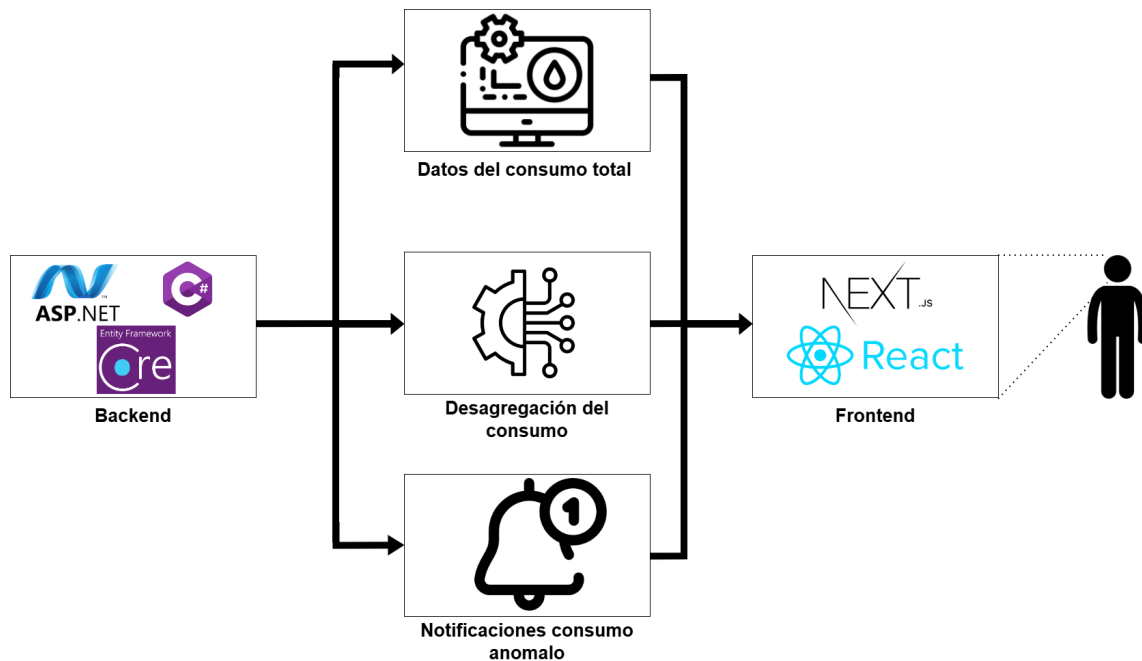


Fig. 5.5: Módulo de aplicación web

## Interacción entre componentes

### Flujo de datos

La Figura 5.6 muestra el flujo de información dentro del sistema. El proceso comienza en el módulo de sensado, el cual envía lecturas del medidor a la base de datos cada un minuto. Estas lecturas se van acumulando y sirven como base para las etapas posteriores.

Cada una hora, el módulo de desagregación recopila la última hora de datos, forma las ventanas de consumo y clasifica cada evento asociando un tipo de dispositivo. Los resultados se guardan nuevamente en la base de datos y, al mismo tiempo, se envían al módulo de detección de anomalías.

A partir de esta información, el módulo de detección de anomalías evalúa si existe algún comportamiento anómalo y, en caso de detectarlo, registra el evento y su severidad en la base de datos. Finalmente, la aplicación web utiliza toda esta información para mostrar al usuario su consumo total, el consumo por dispositivo y las alertas generadas.

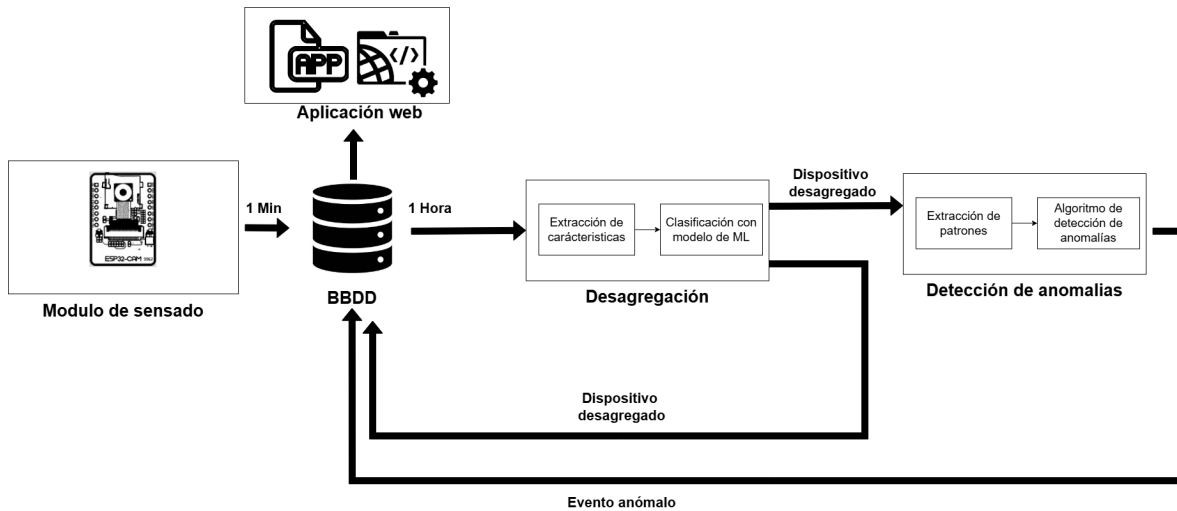


Fig. 5.6: Flujo de los módulos

### Dependencias entre módulos

Como se observa en la Figura 5.6, cada módulo depende de la información generada por el anterior. El sensado entrega las lecturas que permiten la desagregación, y esta última proporciona los eventos que utiliza el módulo de detección de anomalías.

Además la aplicación web depende de todos los resultados almacenados en la base de datos para mostrar al usuario su consumo y las alertas generadas. Por lo tanto cada módulo aporta información esencial para que el sistema funcione de forma completa.

## Diseño de interfaces

### Modelo de navegación

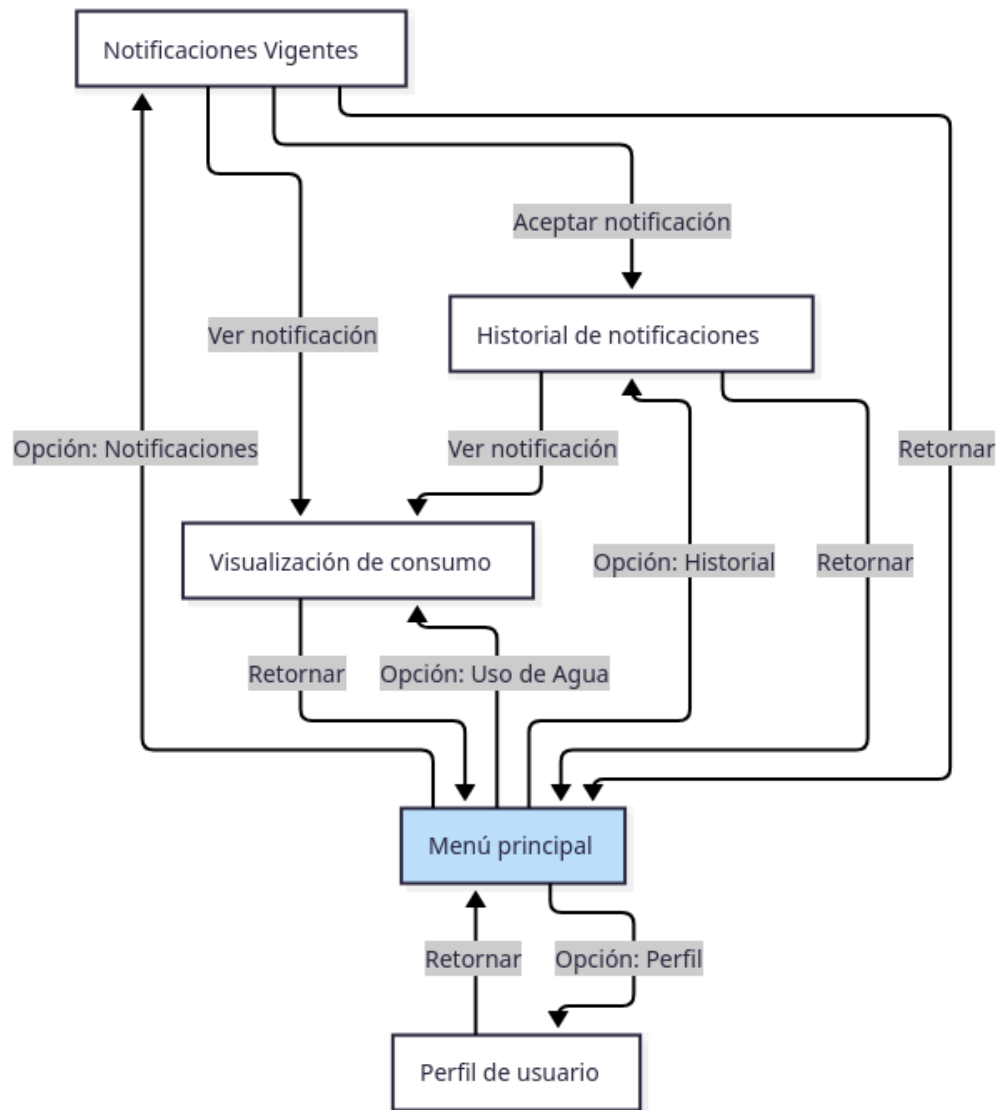


Fig. 5.7: Modelo de navegación

El acceso a la plataforma se puede hacer desde Smartphone o PC mediante una plataforma web. Antes de poder ingresar a los servicios internos, se requiere iniciar sesión.

Al iniciar sesión, se entra a un *landing page* con accesos a las visualizaciones de consumo de agua, notificaciones vigentes e históricas, y administración de perfil.

La visualización de consumo de agua provee opciones para observar el consumo en una

ventana de horas, día, semana, mes o año, dependiendo del historial disponible. Además, se provee feedback con las notificaciones vigentes, mostrándolas dentro de (o acompañando a) la visualización.

El menú de notificaciones permite ver las notificaciones vigentes (no revisadas por el usuario), con opciones para aceptar (ignorarla y pasarla al histórico de notificaciones) o bien verla, que permite mirar información adicional de la notificación.

### Diseño de interfaces usuarias

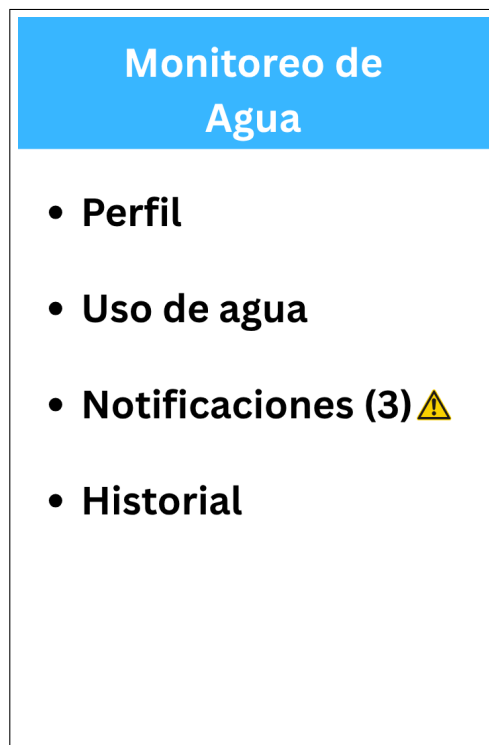


Fig. 5.8: Página principal

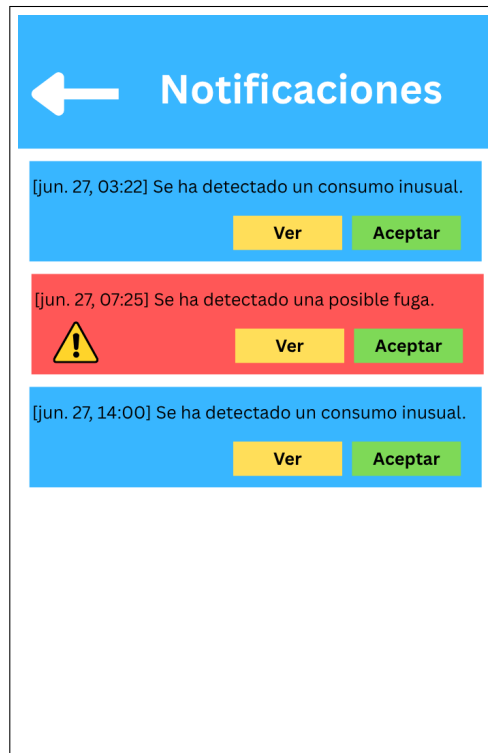


Fig. 5.9: Listado de notificaciones



Fig. 5.10: Visualización de consumo de agua

# BIBLIOGRAFÍA

---

- [1] Superintendencia de Servicios Sanitarios (SISS), “Informe de Gestión del Sector Sanitario 2022”, Gobierno de Chile, Santiago, Chile, Rep. Téc., 2023. <https://www.siss.gob.cl>
- [2] R. Chao, P.-Y. Ting y K.-T. Chuang, “Online Water Usage Monitoring under Anomalous Interference in Residential Households”, en Proceedings of the International Conference on Technologies and Applications of Artificial Intelligence (TAAI), IEEE, Taiwán, 2022.
- [3] J.-E. Fonseca-Revelo, J.-C. Fonseca-Revelo y A. Cerón-Correa, “Desarrollo de un sistema para el monitoreo de recursos hídricos en el hogar”, *Revista Científica* 2024. <https://doi.org/10.14483/23448350.21594>
- [4] Juan Wang and Jakobus E. van Zyl and Le Wen and Yongbo Li and Shuai Che, “The impact of smart meter programmes on household water consumption: evidence from New Zealand”, en Journal of Behavioral and Experimental Economics, 2025. <https://www.sciencedirect.com/science/article/pii/S2214804325000771>
- [5] F. H. Alves, J. Silva, P. Firme, R. Caldeirinha and R. Oliveira, "Design Analysis of Smart Water Meters: An Open Architecture Approach", en Proceedings of the 13th International Conference on Sensor Networks (SENSORNETS 2024), SciTePress, 2024. <https://www.scitepress.org/Papers/2024/125484/125484.pdf>
- [6] OECD, “Water Consumption and Sustainable Water Resources Management”, OECD Publishing, Paris, Francia, Rep. Téc., 1998. <https://doi.org/10.1787/9789264162648-en>
- [7] UNAD, “Diseñar un prototipo de medición de consumo de agua en entornos residenciales mediante el uso de un sistema bajo el estándar IEEE 802.15.4, soportado por un sensor de caudal y registro de información en la nube para el municipio de San Juan de Pasto”, Universidad Nacional Abierta y a Distancia, Colombia, 2022. <https://repository.unad.edu.co/handle/10596/54029>

- [8] Seeed Studio, “YF-S201 Hall Effect Water Flow Sensor Datasheet”, Shenzhen, China, 2020. <https://components101.com/sensors/yf-s201-water-flow-measurement-sensor>
- [9] D. Pinchao, “Diseño y acondicionamiento del sistema de control del sensor ultrasónico para la obtención de información del flujo y de la velocidad superficial”, Escuela Politécnica Nacional, Quito, Ecuador, Tesis, 2021. <https://bibdigital.epn.edu.ec/bitstream/15000/23267/1/CD%2012686.pdf>
- [10] E. T. de Camargo, J. K. S. Tagliari y A. L. G. Azzolin, “Low-Cost Water Quality Sensors for IoT: A Systematic Review”, en *\*Sensors\**, vol. 23, no. 9, 2023. <https://pubmed.ncbi.nlm.nih.gov/articles/PMC10181703/>
- [11] Arad Group, “Smart Water Meters”, Tel Aviv, Israel, Rep. Téc., 2024. <https://www.arad.co.il/>
- [12] H. S. Mamede, R. Moura, P. Pinho y M. Luís, “GraphSmart: A Method for Green and Accurate IoT Water Monitoring”, en *\*Proceedings of the ACM International Conference on Systems for Energy-Efficient Computing and Communication\**, 2023. <https://dl.acm.org/doi/10.1145/3695769>
- [13] M. A. M. Sadeeq y R. Ibrahim, “IoT and ICT based Smart Water Management, Monitoring and Controlling System: A Review”, en *\*Asian Journal of Research in Computer Science\**, 2021. <https://www.academia.edu/48755180/>
- [14] M. A. Khan, F. Ullah, H. Almogren y M. Guizani, “LoRaWAN-based Smart Water Management IoT Applications: A Review”, en *\*International Journal of Smart Sensor Technologies\**, 2025. <https://www.tandfonline.com/doi/full/10.1080/24751839.2025.2458889>
- [15] S. Kumar, L. R. Ryunus, P. Chauhan y R. Venugopal, “IoT Innovations in Sustainable Water and Wastewater Management and Monitoring”, en *\*IEEE Access\**, 2024. <https://ieeexplore.ieee.org/document/10506924>

- [16] M. Madhuri Latha, G. R. Raju y B. Swathi, “Automated Water Meter Reading through Image Recognition”, en *\*International Journal for Research in Applied Science and Engineering Technology\**, vol. 12, no. 5, pp. 1001–1006, 2024. <https://www.ijraset.com/research-paper/automated-water-meter-reading-through-image-recognition>
- [17] AMC SS2024 – Embedded Systems Lab, “Smart Water Metering using ESP32-CAM and Image Recognition”, EOLab, Alemania, 2024. [https://student-wiki.eolab.de/doku.php?id=amc:ss2024:smart\\_water\\_metering:start](https://student-wiki.eolab.de/doku.php?id=amc:ss2024:smart_water_metering:start)
- [18] C. C. Jain, Q. Li y K. Mori, “Smart Reading System for Water Meter Using YOLOv4 and Deep Learning”, en *\*Journal of Advanced Computational Intelligence and Intelligent Informatics\**, vol. 28, no. 1, pp. 57–63, 2024. [https://www.jstage.jst.go.jp/article/jaciii/28/1/28\\_206/\\_article/-char/en](https://www.jstage.jst.go.jp/article/jaciii/28/1/28_206/_article/-char/en)
- [19] The Things Network, “LoRaWAN Documentation”, Amsterdam, Países Bajos, 2025. <https://www.thethingsnetwork.org/docs/lorawan/>
- [20] HiveMQ, “MQTT Essentials – A Lightweight IoT Protocol”, Landshut, Alemania, 2025. <https://www.hivemq.com/mqtt-essentials/>
- [21] J. Holler, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand y D. Boyle, “From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence”, Academic Press, Waltham, MA, EE.UU., 2014.
- [22] Anna Di Mauro, Armando Di Nardo, Giovanni F. Santonastaso y Salvatore Venticinque, “An IoT System for Monitoring and Data Collection of Residential Water End-Use Consumption”, 2019.
- [23] N. A. Attallah, J. S. Horsburgh y C. J. B. Pacheco, “An Open-Source, Semisupervised Water End-Use Disaggregation and Classification Tool”, *ASCE Journal of Water Resources Management*. <https://doi.org/10.1061/JWRMD5>
- [24] A. di Mauro, A. Cominola, A. Castelletti y A. di Nardo, “Urban Water Consumption at

- Multiple Spatial and Temporal Scales: A Review of Existing Datasets”, *Water*, vol. 13, n.º 1, 2021. <https://doi.org/10.3390/w13010036>
- [25] N. V. Gkalinikis y D. Vrakas, “Efficient Deep Learning Techniques for Water Disaggregation”, *SyNERGY MED 2022*. <https://doi.org/10.1109/SyNERGYMED55767.2022.9941424>
- [26] K. A. Nguyen, R. Stewart y H. Zhang, “Transforming Residential Water End Use Analysis: Insights from Low-Resolution Smart Meter Data”, *Water Research*, vol. 278, 2025. <https://doi.org/10.1016/j.watres.2025.123344>
- [27] A. Cominola, M. Giuliani, A. Castelletti, A. Abdallah y D. E. Rosenberg, “Developing a Stochastic Simulation Model for the Generation of Residential Water End-Use Demand Time Series”, *Proceedings of IEMSS 2016*.
- [28] A. A. Qaddoori and N. S. Ali, “A lightweight machine learning architecture for anomaly detection in smart meters”, *IET Wireless Sensor Systems*, 2023. <https://doi.org/10.1049/wss2.12054>
- [29] ScienceDirect, “Fuzzy Logic System”, *Engineering Topics*. <https://www.sciencedirect.com/topics/engineering/fuzzy-logic-system>
- [30] M. Okada et al., “Intelligent Monitoring of Water Consumption in Urban Buildings Using Computing and FOG-SWM”, en *Anais do XXII Simpósio Brasileiro de Computação Ubíqua e Pervasiva*, SBC, 2022. <https://sol.sbc.org.br/index.php/courb/article/download/35260/35050>
- [31] B. Tambuwal and D. Neagu, “Deep Quantile Regression for Unsupervised Anomaly Detection in Time-Series”, *SN Computer Science*, 2021. <https://www.researchgate.net/publication/354948827>
- [32] G. Gerasimov, I. Kamyshev, S. M. Hoosh, and E. Gryazina, “Toward Explainable NILM: Real-Time Event-Based NILM Framework for High-Frequency Data”, *IEEE PowerTech*, 2025. <https://ieeexplore.ieee.org/document/11180424>

- [33] T. C. Britton, R. A. Stewart and O. P. R. O'Halloran, "Smart metering: enabler for rapid and effective post meter leakage identification and water loss management", 2013 <https://www.sciencedirect.com/science/article/abs/pii/S0959652613003272>
- [34] A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, "Internet of Things for Smart Cities", <https://www.mdpi.com/1424-8220/18/9/2796>
- [35] A. Cominola, M. Giuliani, D. Piga, A. Castelletti, and A. E. Rizzoli, "Benefits and challenges of non-intrusive water end-use monitoring and disaggregation: A systematic review", *Environmental Modelling & Software*, 2015. <https://doi.org/10.1016/j.envsoft.2015.05.006>
- [36] A. Cominola and M. Giuliani and D. Piga and A. Castelletti and A.E. Rizzoli, "A Hybrid Signature-based Iterative Disaggregation algorithm for Non-Intrusive Load Monitoring", *Applied Energy*, 2017. <https://www.sciencedirect.com/science/article/abs/pii/S030626191631488X>
- [37] D. Cruz-Rangel, C. Ocampo-Martinez, and J. Diaz-Rozo, "Online non-intrusive load monitoring: A review", *Energy Nexus*, 2025. <https://doi.org/10.1016/j.nexus.2024.100348>
- [38] Giulia Tanoni and Emanuele Principi and Stefano Squartini, "Non-Intrusive Load Monitoring in industrial settings: A systematic review", *Renewable and Sustainable Energy Reviews*, 2024. <https://www.sciencedirect.com/science/article/pii/S1364032124004295>
- [39] O. Al-Tameemi, J. Zhang, and M. Al-Tameemi, "Interactive Fuzzy Logic Interface for Enhanced Real-Time Water Quality Index Monitoring", *Algorithms*, vol. 18, no. 9, 2025. <https://doi.org/10.3390/a18090591>
- [40] A. Indriani, Mullah, and A. G. Putrada, "Water Pipe Leakage Detection System based on Fuzzy Logic and Sensor Network", *International Journal of Engineering Trends and*

- Technology*, vol. 73, no. 4, pp. 111–116, 2025. <https://doi.org/10.14445/2315381/IJETT-V73I4P111>
- [41] A. Ramos, F. Moxca, and V. Sousa, “Evaluating the Impact of Membership Functions and Defuzzification Methods in a Fuzzy System”, *Applied Sciences*, vol. 15, no. 4, 2025. <https://doi.org/10.3390/app15041934>
- [42] P. De Marinis, A. Candelieri, and F. Archetti, “Explainable Fuzzy GNNs for Leak Detection in Water Distribution Networks”, *arXiv preprint arXiv:2601.03062*, 2026. <https://arxiv.org/abs/2601.03062>
- [43] V. Chandola, A. Banerjee y V. Kumar, “Anomaly detection: A survey”, en *ACM Computing Surveys (CSUR) 2009*. <https://dl.acm.org/doi/10.1145/1541880.1541882>
- [44] A. Candelieri, “Clustering and support vector regression for water demand forecasting and anomaly detection”, 2017. <https://www.mdpi.com/2073-4441/9/3/224>
- [45] A. Cominola, M. Giuliani, D. Piga, A. Castelletti y A.-E. Rizzoli, “Benefits and challenges of high resolution water smart metering: A survey”, en *Environmental Modelling & Software*, 2015. <https://www.sciencedirect.com/science/article/abs/pii/S1364815215300177>