
Ingeniería basada en modelos para la gestión de afloraciones de cianobacterias



UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INGENIERÍA INFORMÁTICA

Trabajo de Fin de Máster del Máster en Ingeniería Informática

Curso 2022/23

Model driven engineering for the management of cyanobacteria blooms

Convocatoria: Febrero 2023

Calificación: 10

Autora:

BEATRIZ HERGUEDAS PINEDO

Directores:

JOSÉ LUIS RISCO MARTÍN

EVA BESADA PORTAS

Madrid, 10 de febrero de 2023

Resumen

En la actualidad, la escasez de agua es un tema que está a la orden del día ya que el abastecimiento está condicionado por el cambio climático, la sobreexplotación y la contaminación. Además, a estos factores condicionantes hemos de añadir las afloraciones de algas y cianobacterias (HABs, del inglés, *Harmful Algal and Cyanobacterial Blooms*), que al ocurrir en todo tipo de masas de agua y conllevar la producción de toxinas nocivas para la salud humana y de los animales, tienen un gran impacto ambiental y económico.

Hasta ahora, estas afloraciones han sido evaluadas principalmente realizando una recolección manual y un posterior análisis de las muestras de agua, o con la ayuda ocasional de ciertos instrumentos automáticos que adquieren información de ubicaciones fijas, como podrían ser algunas boyas de muestreo periódico que se encuentran ancladas en lagos y embalses.

Sin embargo, aunque lo ideal es disponer de sistemas de alerta temprana para la detección de cianobacterias, los procedimientos utilizados hasta el momento no suelen proporcionar datos con la consistencia espacial y temporal suficiente para poder anticipar la formación de HABs, por lo que los sistemas de alerta temprana existentes no tienen las prestaciones necesarias para predecir de forma adecuada y con la anticipación suficiente la evolución de cianobacterias.

Por todo esto, es necesario desarrollar nuevas técnicas y herramientas que combinen recogidas de datos más exhaustivas con simulaciones numéricas de los HABs para detectar, caracterizar y poder predecir y responder eficientemente a las afloraciones que amenazan la calidad del agua y la salud de los seres vivos. Por ello, en este trabajo se propone implementar una solución integral para la monitorización y gestión en tiempo real de las afloraciones de cianobacterias, respaldada por una infraestructura de ingeniería de sistemas basada en modelos (MBSE, del inglés, *Model-Based Systems Engineering*) para entornos dinámicos.

Palabras clave

Monitorización de cianobacterias, Afloraciones de Algas Nocivas (HABs), Modelado y Simulación (M&S), Sistemas de Alerta Temprana (EWS), Especificación de Sistemas de Eventos Discretos (DEVS).

Abstract

Nowadays, water scarcity is an issue that is the order of the day as water supply is conditioned by climate change, overexploitation and pollution. In addition to these conditioning factors, Harmful Algal and Cyanobacterial Blooms (HABs), which occur in all types of water bodies and produce toxins that are harmful to human and animal health, have a major environmental and economic impact.

So far, these upwellings have been assessed mainly by manual collection and subsequent analysis of water samples, or with the occasional help of certain automatic instruments that acquire information from fixed locations, such as some periodic sampling buoys that are anchored in lakes and reservoirs.

However, although the ideal is to have early warning systems for the detection of cyanobacteria, the procedures used so far do not usually provide data with sufficient spatial and temporal consistency to anticipate the formation of HABs, so that existing early warning systems do not have the necessary performance to adequately predict the evolution of cyanobacteria sufficiently in advance.

Therefore, it is a need to develop new techniques and tools that combine more comprehensive data collection with numerical simulations of HABs to detect, characterize, predict and efficiently respond to upwelling that threaten water quality and the health of living organisms. Therefore, in this work we propose to implement a comprehensive solution for real-time monitoring and management of cyanobacterial blooms, supported by a Model-Based Systems Engineering (MBSE) infrastructure for dynamic environments.

Keywords

Cyanobacteria monitoring, Harmful Algal and Cyanobacteria Blooms (HABs), Modeling and Simulation (M&S), Early Warning Systems (EWS), Discrete Event System Specification (DEVS).

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Contexto	4
1.3. Objetivos	5
1.4. Estructura del documento	6
2. Estado del arte	8
2.1. Cianobacterias	8
2.1.1. Morfología y alimentación	9
2.1.2. Comportamiento	10
2.1.3. Impacto ambiental, económico y sanitario	11
2.1.4. Tratamiento actual	12
2.2. Sistemas de detección y alerta temprana	12
2.3. DEVS	15
3. Arquitectura del sistema	19
3.1. Imagen general	19
3.2. Arquitectura en DEVS	21
3.2.1. Capa <i>Edge</i>	22
3.2.2. Capa <i>Fog</i>	25
3.2.3. Capa <i>Cloud</i>	26
4. Implementación de los subsistemas de predicción y análisis	28
4.1. Predicción del <i>bloom</i>	28
4.2. Análisis de datos	31
4.2.1. Informes de la capa <i>Fog</i>	31
4.2.2. Informes de la capa <i>Cloud</i>	44
4.3. Servicio de entrenamiento	48
5. Caso de uso	51
5.1. Descripción del escenario	51
5.2. Simulación	51
5.3. Análisis de datos	53
5.4. Entrenamiento	60
6. Conclusiones	62
6.1. Revisión de objetivos	62
6.2. Trabajo futuro	63

Anexos	65
A. Introduction	66
A.1. Motivation	66
A.2. Context	69
A.3. Objectives	70
A.4. Document structure	71
B. Conclusions	72
B.1. Review of objectives	72
B.2. Future work	73
Bibliografía	75

Índice de figuras

1.1.	Afloración de cianobacterias en la superficie de un río (fuente: [8])	2
1.2.	Extracción manual de muestras para el análisis del agua (fuente: [12])	3
1.3.	Boya de muestreo periódico del agua (fuente: [14])	4
2.1.	Cianobacterias observadas al microscopio (fuentes [22, 23])	9
2.2.	Distintas configuraciones de cianobacterias en el espacio (fuente [36])	11
2.3.	Representación gráfica de un sistema	15
2.4.	Representación del juego de Ping Pong en DEVS	16
2.5.	Representación de un modelo acoplado	17
3.1.	Esquema general del marco propuesto	20
3.2.	Arquitectura del sistema en DEVS	21
3.3.	Arquitectura de la capa <i>Edge</i> en DEVS	22
3.4.	Ejemplos de eventos transmitidos por el USV de la capa <i>Edge</i> al GCS de la capa <i>Fog</i>	24
3.5.	Arquitectura de la capa <i>Fog</i> en DEVS	25
3.6.	Arquitectura de la capa <i>Cloud</i> en DEVS	27
4.1.	Gráfico 1 del informe de la capa <i>Fog</i>	33
4.2.	Gráfico 2 del informe de la capa <i>Fog</i>	34
4.3.	Gráfico 3 del informe de la capa <i>Fog</i>	35
4.4.	Gráfico 4 del informe de la capa <i>Fog</i>	36
4.5.	Gráficos 5 y 6 del informe de la capa <i>Fog</i>	37
4.6.	Gráfico 7 del informe de la capa <i>Fog</i>	38
4.7.	Gráfico 8 del informe de la capa <i>Fog</i>	38
4.8.	Gráfico 9 del informe de la capa <i>Fog</i>	39
4.9.	Gráfico 10 del informe de la capa <i>Fog</i>	39
4.10.	Gráfico 11 del informe de la capa <i>Fog</i>	40
4.11.	Gráfico 12 del informe de la capa <i>Fog</i>	40
4.12.	Gráfico 13 del informe de la capa <i>Fog</i>	41
4.13.	Gráfico 14 del informe de la capa <i>Fog</i>	41
4.14.	Gráficos 15 y 16 del informe de la capa <i>Fog</i>	42
4.15.	Gráfico 17 del informe de la capa <i>Fog</i>	42
4.16.	Gráfico 18 del informe de la capa <i>Fog</i>	43
4.17.	Gráficos 19 y 20 del informe de la capa <i>Fog</i>	43
4.18.	Gráfico 21 del informe de la capa <i>Fog</i>	44
4.19.	Gráfico 22 del informe de la capa <i>Fog</i>	44
4.20.	Mapa de calor generado por el informe de <i>Cloud</i>	47
4.21.	Zoom sobre el mapa de calor del informe	48

4.22. Representación de la técnica de descenso de gradiente (fuente: [83])	49
5.1. Lago de Washington en Washington, Estados Unidos	52
5.2. Marco de simulación del seguimiento de HABs	54
5.3. Medidas de los sensores embarcados a lo largo de la simulación	56
5.4. Información del <i>bloom</i> inferido durante la simulación	56
5.5. Información de los parámetros del USV durante la simulación	57
5.6. Error cometido en la predicción de HABs	57
5.7. Mapa de calor de las zonas con más probabilidad de encontrar HABs	58
5.8. Ampliación del mapa de calor con más probabilidad de encontrar HABs	59
5.9. Segunda ampliación del mapa de calor con más probabilidad de encontrar HABs	59
5.10. Error en la densidad del <i>bloom</i> predicho tras el entrenamiento	60
5.11. Diferencias en la predicción del <i>bloom</i> antes y después del entrenamiento	61
A.1. Cyanobacteria bloom on the surface of a river (source: [8])	67
A.2. Manual extraction of samples for water analysis (source: [12])	68
A.3. Periodic water sampling buoy (source: [14]).	69

Índice de tablas

4.1. Tabla del informe de la capa <i>Cloud</i>	46
5.1. Tabla con cálculos sobre los datos de la simulación	58
5.2. Número de HABs detectados en la simulación	58

Capítulo 1

Introducción

En este capítulo se explica la motivación que ha impulsado la investigación llevada a cabo en el presente trabajo, así como los objetivos a cumplir al finalizar el mismo y la estructura del documento.

1.1. Motivación

La escasez de agua afecta hoy en día a alrededor de 4 mil millones de personas (i.e. dos tercios de la población mundial), problema que se ve agravado durante por lo menos un mes al año [1].

Cuando un territorio extrae al menos el 25 % de sus recursos renovables de agua dulce, se dice que sufre estrés hídrico [1]. En la actualidad, 2.300 millones de personas viven en países con este problema, de los cuales 733 millones habitan en lugares con estrés hídrico alto o crítico. Además, 3.200 millones de personas residen en zonas agrícolas con escasez de agua de alta a muy alta, de las cuales 1.200 millones tienen graves limitaciones de agua.

A todo lo anterior debemos añadir que más de 1.300 millones de personas no tienen acceso a agua potable, incumpliendo así el derecho a un acceso asequible a agua salubre establecido por el Comité de Derechos Económicos, Sociales y Culturales de las Naciones Unidas [2].

El problema de escasez de agua va estrechamente ligado al término *calidad del agua* [3], que hace referencia a las características químicas, físicas y biológicas del agua dependiendo del uso que se le va a dar. De hecho, para que el agua sea salubre, debe cumplir ciertos estándares de calidad, y para determinar si dichos estándares se satisfacen, se miden y analizan determinadas características como su temperatura, su contenido mineral y la cantidad de bacterias que tiene.

En la actualidad, la calidad del agua es un tema de gran relevancia por muchos motivos [4]. En primer lugar, la población mundial ha aumentado de forma considerable en los últimos años, lo que implica una mayor necesidad de agua para garantizar el abastecimiento global. En segundo lugar, el cambio climático que sufre el planeta desde hace algunas décadas provoca grandes períodos de sequía, con una disminución de la frecuencia y cantidad de lluvia que afecta de manera significativa a las reservas hídricas. Por último,

la sobreexplotación y el exceso de contaminación que sufren los recursos de agua tanto en las áreas urbanas como en las rurales, aumenta la importancia de cumplir los estándares de calidad y que el garantizarlos sea cada vez más difícil.

Por un lado, en los grandes núcleos urbanos encontramos problemas de uso excesivo y contaminación del agua debido al gran volumen de residuos generados, incluyendo el alcantarillado y los vertidos de las zonas industriales que, en gran cantidad, van a parar a los recursos hídricos [5]. Por otro lado, en las áreas rurales encontramos numerosas explotaciones agrícolas y ganaderas que también requieren una gran cantidad de agua, y además utilizan abonos, fertilizantes y pesticidas que pueden acabar filtrados en ella. Todo esto produce cambios en la composición del agua por la introducción de sustancias o microorganismos no deseados que pueden llegar a convertirse en un riesgo para la salud.

Uno de los microorganismos que pueden acabar invadiendo las reservas de agua y poniendo en riesgo nuestra salud son las cianobacterias. Estas bacterias, también llamadas algas verdeazuladas por su color más habitual [6, 7], son organismos microscópicos unicelulares que se encuentran naturalmente en todo tipo de aguas, ya sea agua dulce, marina o salobre (agua dulce y salada combinada). Realizan la fotosíntesis, utilizando la luz solar para alimentarse de los nutrientes presentes en el agua. Por ello, en ambientes cálidos y ricos en fósforo y nitrógeno, las cianobacterias pueden multiplicarse rápidamente, creando afloraciones que se extienden por toda la masa de agua y que pueden volverse visibles cuando emergen a la superficie. Es entonces cuando se aprecia su color verdeazulado, como puede observarse en la figura 1.1.



Figura 1.1: Afloración de cianobacterias en la superficie de un río (fuente: [8])

Estos organismos están presentes en todos los ambientes y juegan un papel importante como productores primarios en todos ellos, por lo que son buenos y necesarios en la naturaleza. Sin embargo, cuando se multiplican de manera exagerada y sin control, forman afloraciones que son dañinas para la salud de las personas, los animales o el medio ambiente al bloquear la luz solar que otros organismos necesitan para vivir, robarles el oxígeno y

los nutrientes, y producir metabolitos secundarios de elevada toxicidad denominados de forma genérica *cianotoxinas* [9].

Por tanto, nadar o realizar actividades en aguas que contienen afloraciones de cianobacterias, beber agua que proviene de un lago o embalse que tiene una proliferación de ellas, o comer pescado infectado de las toxinas que producen es tremendamente nocivo tanto para la salud humana como animal [10]. En los seres humanos, estas toxinas pueden provocar dolor abdominal, dificultad para respirar, daños hepáticos y renales irreversibles, problemas en el sistema nervioso y crecimiento de tumores.

Por este motivo, no son pocos los esfuerzos invertidos en detectar estos organismos en los embalses y lagos. El procedimiento más habitual para ello es tomar manualmente muestras de agua para analizarlas posteriormente y así poder determinar si las cianotoxinas están contaminando el agua [11]. Esto conlleva un gran esfuerzo por parte del personal involucrado, ya que la extracción manual de muestras y su posterior análisis tiene un alto coste en tiempo y dinero, además de no resultar del todo efectivo debido a la gran cantidad de superficie que se necesita abarcar para poder llevar a cabo un estudio exhaustivo. En la figura 1.2 puede observarse el método manual de recolección de muestras para el análisis de la calidad del agua.

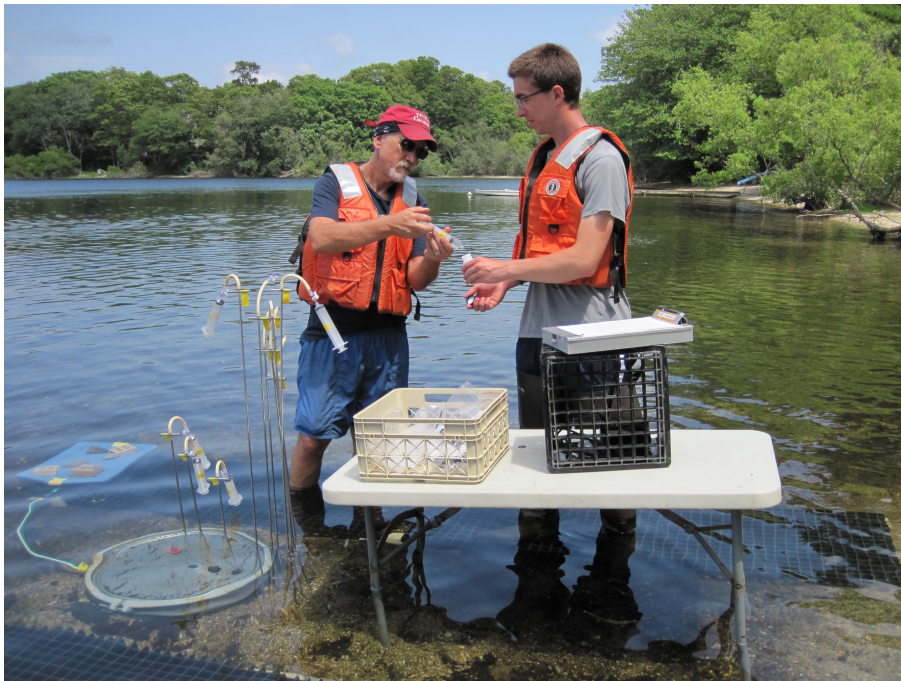


Figura 1.2: Extracción manual de muestras para el análisis del agua (fuente: [12])

Además, las restricciones de recursos financieros y de personal reducen la recolección manual a los momentos del año en que la aparición de afloraciones de cianobacterias es más probable, llevándose a cabo en pocos puntos geográficos y con frecuencias mínimas. El retraso que sufren los resultados analíticos y la limitada capacidad para interpretar el escenario actual disminuye la posibilidad de predicción, prevención y mitigación de HABs por parte de las autoridades responsables de la distribución de agua potable y sus usos recreativos [13].

Este proceso de extracción de muestras cuenta ocasionalmente con ayuda de ciertos instrumentos automáticos que adquieren información del estado del agua en determinados puntos estratégicos. Un buen ejemplo de este tipo de instrumentos son las boyas de muestreo periódico [14] que se encuentran ancladas en diferentes puntos de lagos y embalses, y que cuentan con sensores que, de forma regular, toman medidas sobre diferentes parámetros relacionados con la calidad del agua aunque, claro está, siempre en el mismo punto. Una de estas boyas puede verse en la figura 1.3 junto con los datos que monitoriza.

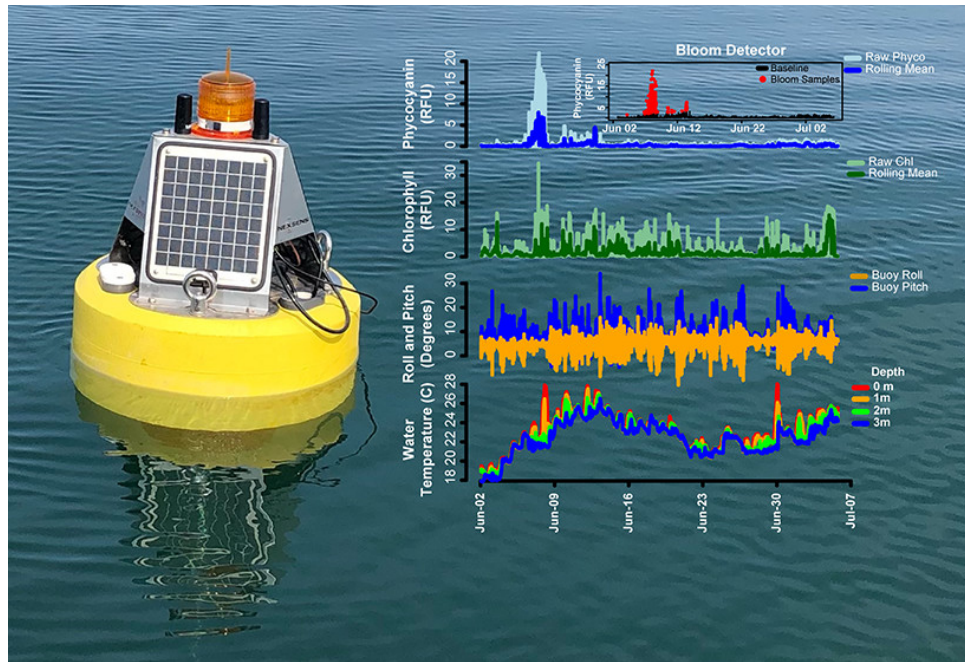


Figura 1.3: Boya de muestreo periódico del agua (fuente: [14])

1.2. Contexto

Para dar una solución integral al problema de la monitorización y gestión de las cianobacterias conviene ir un paso más allá, y no solo detectar las afloraciones de estos microorganismos con mayor frecuencia y resolución, sino también predecir su evolución espacio temporal. Para esto último existen a día de hoy algunos sistemas de alerta temprana [15] que se basan en la simulación de modelos numéricos, como los que implementan herramientas comerciales como MIKE [16] o EEMS [17]. Sin embargo, el uso exclusivo de instrumentos de simulación no suele ser suficiente, ya que la complejidad de los procesos biológicos y de transporte de las cianobacterias dificulta la correcta sintonización de los numerosos parámetros que existen en estos modelos comerciales. Además, introduce incertidumbres relevantes en las predicciones de los modelos.

Por ello se propone una solución integral, es decir, una solución que no solo implemente los modelos numéricos sino que también incorpore todo el ecosistema de aparataje para realizar mediciones que complementen y alimenten las predicciones hechas por los modelos numéricos con datos reales. Esta idea se está desarrollando en el Proyecto Sinérgico de I+D en Nuevas y Emergentes Áreas Científicas de la Comunidad de Madrid, llamado *Hacia un sistema Integral para la Alerta y GESTión de BLOOMs de cianobacterias en aguas continentales, IA-GES-BLOOM-CM*, en el que se está elaborando un sistema

complejo que sugiere una alternativa para mejorar la toma de muestras, creando nuevos modelos, sintonizándolos y acercando los resultados a los gestores, todo ello siguiendo una arquitectura basada en IoT (del inglés, *Internet of Things*) que involucra elementos de distinta índole, como son modelos, sensores, barcos y gestores.

Debido a la complejidad que implica un sistema de dichas características, es conveniente crear una infraestructura en la que se realice un gemelo digital de todo el sistema, con herramientas sistemáticas de modelado y simulación. En este contexto es muy útil el formalismo DEVS (del inglés, *Discrete Event System Specification*, es decir, Especificación de Sistemas de Eventos Discretos) [18, 19], ya que proporciona un marco de modelado y simulación para eventos discretos sólido para poder desarrollar sistemas tan sofisticados como el propuesto. En esta línea se está desarrollando también en la Universidad Complutense de Madrid otro proyecto llamado *Más allá del uso de tecnologías digitales en blooms de cianobacterias: gestión inteligente de cianobacterias mediante el uso de gemelos digitales y computación en el borde (SMART-BLOOMS)*.

Este TFM se centra por tanto en colaborar en la construcción de la arquitectura global del sistema, y en la implementación de algunos servicios que forman parte de la compleja infraestructura descrita. Tanto el diseño de la arquitectura como su implementación se realizan en base al formalismo DEVS. Para ello se sigue una organización basada en IoT. Todo esto nos permite garantizar de manera eficiente un sistema no solo de detección sino también de predicción y monitorización de afloraciones de cianobacterias, combatiendo así los riesgos que estos microorganismos generan para la salud y el medio ambiente.

1.3. Objetivos

Se han definido una serie de objetivos para este trabajo, que se enumeran a continuación. Dichos objetivos se pueden categorizar como generales o específicos. Los objetivos generales están orientados al desarrollo de metas a gran escala, mientras que los específicos profundizan más en las diferentes cuestiones que se abordan.

- [O1] Profundizar en el conocimiento de las cianobacterias, los sistemas de predicción y el formalismo DEVS.
 - [O1.1] Estudiar las afloraciones de cianobacterias, su origen y su impacto en la calidad del agua y en la salud de los seres vivos.
 - [O1.2] Revisar los sistemas de alerta temprana para afloraciones de cianobacterias existentes.
 - [O1.3] Profundizar en el conocimiento del formalismo DEVS y su funcionamiento.
- [O2] Participar en la creación de la arquitectura en DEVS de un sistema integral para la monitorización y gestión de HABs.
 - [O2.1] Idear un módulo de modelado y simulación que permita predecir la aparición de cianobacterias.
 - [O2.2] Implementar un servicio de análisis de datos que permita a sus usuarios validar y visualizar los resultados obtenidos con cada simulación realizada.
 - [O2.3] Desarrollar un protocolo de entrenamiento en la nube que ayude a la definición de modelos predictivos para la detección temprana de las afloraciones.

- [O3] Integrar y verificar los módulos desarrollados en la arquitectura DEVS general.
 - [O3.1] Integrar en la arquitectura global el servicio de predicción de HABs desarrollado.
 - [O3.2] Añadir al sistema el servicio de análisis de datos implementado.
 - [O3.3] Incluir en el marco del sistema el servicio de entrenamiento ideado.
- [O4] Comprobar el funcionamiento del sistema a través de un caso de uso.
 - [O4.1] Ejemplificar el funcionamiento del sistema completo con una simulación de un lago real.
 - [O4.2] Probar el servicio de predicción de HABs y comparar la predicción con datos reales.
 - [O4.3] Activar el servicio de análisis de datos y comprobar que los informes generados son correctos.
 - [O4.4] Utilizar el servicio de entrenamiento y comprobar que optimiza la predicción.

1.4. Estructura del documento

El presente documento se encuentra dividido en seis capítulos a través de los cuales se va describiendo el desarrollo del trabajo llevado a cabo. En esta sección se realiza un breve resumen del contenido de cada uno de estos capítulos.

El capítulo 1 introduce la problemática de las afloraciones de cianobacterias en los embalses y lagos, y expone el contexto en el que ha surgido el trabajo y cuál es su motivación. A su vez, incluye la lista de objetivos generales y específicos que se se propone alcanzar al finalizar la investigación.

Seguidamente, en el capítulo 2 se profundiza en el estudio de las cianobacterias, la causa de las afloraciones y el impacto tanto ambiental como económico y sanitario de las mismas. También se realiza un breve estado del arte sobre los sistemas de alerta temprana para detectar, monitorizar y gestionar la presencia de las toxinas nocivas producidas por *blooms* de cianobacterias. Por último, se presentan las características más relevantes del formalismo DEVS, su estructura y funcionamiento.

En el capítulo 3 se detalla la arquitectura del sistema propuesto para gestionar de manera eficiente las afloraciones de cianobacterias. La infraestructura, desarrollada en DEVS e integrada en un entorno IoT que se apoya en el uso de gemelos digitales, consta de tres capas que se explican de forma que sea fácil visualizar la imagen general del marco propuesto, que tiene como objetivos facilitar la predicción, agilizar la detección de HABs, y mejorar la gestión de los problemas que acarrearán en tiempo real.

Posteriormente, en el capítulo 4, se profundiza en el desarrollo de tres servicios que forman parte de la arquitectura presentada en el capítulo 3. Estos servicios son el modelo de predicción, que permite simular el comportamiento de afloraciones en tiempo real; el servicio de análisis de datos, que aporta gráficas para el estudio del escenario de simulación; y el sistema de entrenamiento en la nube, que permite re-entrenar el modelo de predicción para optimizar los resultados. Estos servicios se corresponden con el subsistema de predicción y análisis de datos de la arquitectura propuesta.

El trabajo continúa en el capítulo 5, donde se ejemplifica todo lo expuesto y desarrollado en los capítulos 3 y 4 con un caso de uso que consiste en una simulación llevada a cabo en el lago de Washington (Estados Unidos) usando datos sintéticos del lago extraídos de la herramienta EEMS. En la ejecución se puede observar el funcionamiento del sistema completo, lo que nos permite estudiarlo y prestar especial interés a los modelos de predicción, análisis de datos y entrenamiento.

En el capítulo 6 se revisan los objetivos detallados al inicio del documento y se plantean posibles vías de investigación de cara a un trabajo futuro relacionado con el ámbito del modelado, predicción y gestión de las cianobacterias y sus problemas derivados para la salud pública y el medio ambiente.

Por último, el documento contiene dos anexos que consisten en la traducción al inglés de los capítulos 1 y 6, que son, respectivamente, la Introducción y las Conclusiones.

Capítulo 2

Estado del arte

En este capítulo daremos las nociones básicas de tres conceptos fundamentales para el desarrollo del trabajo así como las soluciones existentes al problema propuesto. En primer lugar, estudiaremos las cianobacterias en profundidad, así como su impacto en el medio ambiente y la vida humana. Posteriormente exploraremos los sistemas de alerta temprana existentes en la actualidad y su funcionamiento. Por último, hablaremos sobre DEVS, introduciendo de forma general los conceptos de sistema y modelos de eventos discretos para continuar con la exposición de las características más relevantes del formalismo elegido y de los simuladores de eventos discretos que usaremos para modelar nuestra propuesta.

2.1. Cianobacterias

Las cianobacterias, cuyo nombre proviene del griego antiguo y hace referencia a su color más habitual, son un tipo de bacterias que viven en entornos acuosos, obtienen energía a través de la fotosíntesis y producen metabolitos tóxicos secundarios. También son conocidas como algas verdeazuladas [6, 7], aunque no suelen clasificarse científicamente como algas, y existen algunas especies con coloraciones diferentes. Es común encontrarlas en agregados coloniales que pueden adoptar formas de filamentos, láminas o incluso esferas huecas [20], tal y como puede observarse en la figura 2.1. Cabe destacar que estos organismos a menudo dominan las capas superiores de los tapetes microbianos encontrados en todo tipo de ambientes, incluyendo algunos extremos como aguas hipersalinas, aguas termales, desiertos o regiones polares [21].

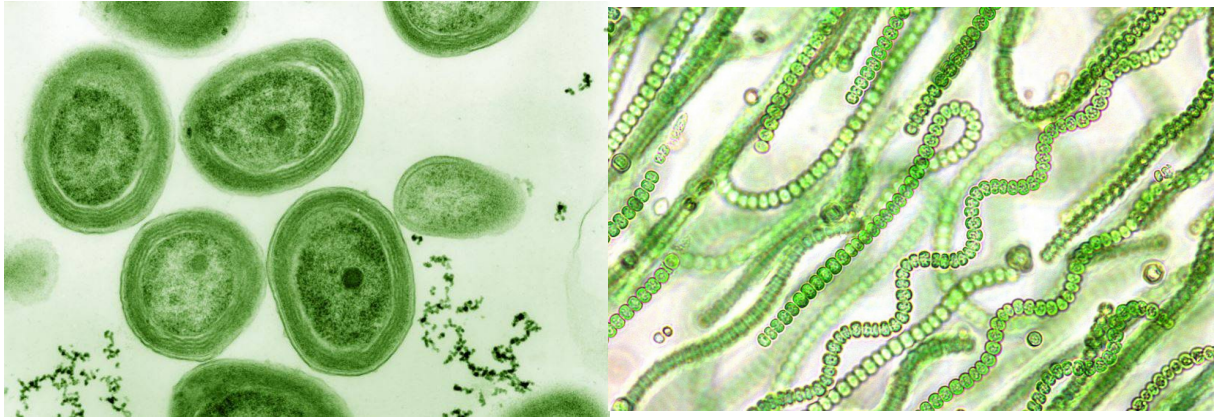


Figura 2.1: Cianobacterias observadas al microscopio (fuentes [22, 23])

Registros fósiles que datan de hace 3.500 millones de años demuestran que estos organismos se encuentran entre los más antiguos de la Tierra [24]. Desde su aparición, las cianobacterias han sido actores esenciales en el desarrollo de los ecosistemas del planeta. Además, son los principales contribuyentes a los ciclos biogeoquímicos globales en la actualidad [25]. Concretamente, las cianobacterias planctónicas son un componente fundamental de las cadenas alimenticias acuáticas y son importantes cooperantes en los flujos globales de carbono y nitrógeno. Sin embargo, algunas cianobacterias forman afloraciones nocivas, provocando la interrupción de los ciclos de los ecosistemas acuáticos y la intoxicación de las diversas especies de la vida silvestre y humana por la producción de poderosas toxinas llamadas cianotoxinas [9], entre las que se encuentran las microcistinas y las saxitoxinas. La toxicidad de alguna de estas sustancias secundarias es tan elevada, que las afloraciones de cianobacterias representan una grave amenaza para los entornos acuáticos y la salud pública, cuya frecuencia y magnitud está aumentando a nivel mundial [26].

2.1.1. Morfología y alimentación

Las cianobacterias tienen una morfología variable, desde formas unicelulares y filamentosas hasta formas coloniales [25]. Las formas filamentosas presentan una diferenciación celular funcional, como los *heterocistos* (para la fijación de nitrógeno), los *aquinetos* (células en estado de reposo) y las *hormogonias* (filamentos móviles reproductivos que se alejan de la biomasa principal) [27]. Estos, junto con las conexiones intercelulares que poseen, se consideran los primeros signos de pluricelularidad [28].

Este grupo de bacterias fotosintéticas ha evolucionado para adaptarse a condiciones ambientales de bajo oxígeno. Por ello, algunas de estas bacterias han desarrollado la posibilidad de fijar nitrógeno atmosférico de forma anaeróbica convirtiéndolo en amoníaco, nitritos o nitratos, que pueden ser absorbidos por las plantas y convertidos en proteínas y ácidos nucleicos. Estas especies fijadoras de nitrógeno viven en una amplia variedad de suelos húmedos y agua, ya sea de forma independiente o en una relación simbiótica con plantas u hongos formadores de líquenes [29].

Además, las cianobacterias realizan la fotosíntesis y la respiración celular, dos procesos opuestos, en una misma membrana. En la fotosíntesis [30] las cianobacterias utilizan la energía de la luz para sintetizar compuestos orgánicos a partir de dióxido de carbono,

usando de forma general agua como donante de electrones y produciendo oxígeno. La respiración [31] es el proceso contrario, es decir, convierte los carbohidratos en dióxido de carbono.

2.1.2. Comportamiento

Algunas cianobacterias, incluso las unicelulares, muestran comportamientos colectivos sorprendentes y forman colonias agrupándose en esteras o tapetes microbianos que pueden flotar en el agua y desempeñar funciones ecológicas importantes. Sin embargo, a veces estas colonias pueden generar metabolitos secundarios tóxicos dañinos para el medio ambiente, agotar el oxígeno del agua o reducir la penetración de luz solar, comprometiendo con ello el desarrollo de otras especies [32].

Las cianobacterias tienen requisitos estrictos de luz. Muy poca luz puede dar como resultado una producción de energía insuficiente y hacer que las células recurran a la respiración heterótrofa. Sin embargo, demasiada luz puede inhibir las funciones celulares, disminuir la eficiencia de la fotosíntesis y causar daño por decoloración. La radiación ultravioleta es especialmente letal para las cianobacterias, y en algunos casos los niveles de radiación solar en la superficie terrestre son significativamente perjudiciales para estos microorganismos [20, 21, 33].

Debido a los fuertes requisitos de luz que precisan, las cianobacterias que viven en esteras microbianas a menudo migran vertical y horizontalmente dentro de la estera para encontrar el lugar que proporcione el nivel adecuado de luz para realizar la fotosíntesis. Por ejemplo, se han encontrado cianobacterias en las esteras bentónicas hipersalinas de Guerrero Negro, México, que migran hacia las capas inferiores durante el día para escapar de la intensa luz solar y luego ascienden a la superficie al anochecer [34]. Sin embargo, otra población de cianobacterias encontrada en esteras hipersalinas en Camargue, Francia, migran a la capa superior de la estera durante el día y se distribuyen homogéneamente a través de la estera durante la noche [35]. Así, estas especies son capaces de controlar su posición vertical y, al mismo tiempo, están sujetas a movimientos horizontales provocados por las corrientes de agua y la acción del viento. Por lo tanto, su distribución espacial puede cambiar en pocas horas, creando diferentes configuraciones de posición, tal y como puede verse en la figura 2.2. En ella, los puntos 1 y 2 representan afloraciones superficiales; los puntos 3, 5 y 6 su distribución a lo largo de la columna de agua; el punto 7 reproduce afloraciones debajo del hielo; el punto 4 representa *blooms* metalimnéticos¹ y el punto 8 simboliza tapetes bentónicos² de cianobacterias. Además, la línea horizontal discontinua indica el límite inferior de la zona eufótica, que corresponde a la capa que recibe el 1% de la luz que incide sobre la superficie del lago, mientras que A y B indican la ubicación de tuberías de extracción para el tratamiento y suministro de agua potable.

¹Que están entre dos hábitats.

²Las algas que se hallan fijas al fondo marino son bentónicas, a diferencia de las que se encuentran flotando en el agua, que se denominan planctónicas.

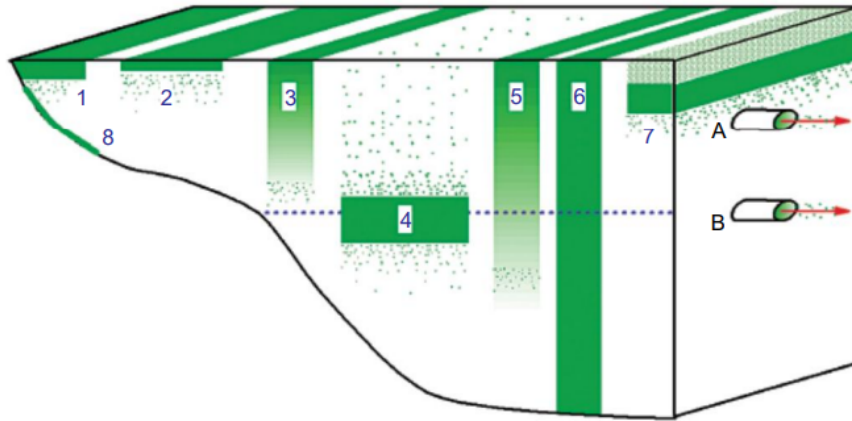


Figura 2.2: Distintas configuraciones de cianobacterias en el espacio (fuente [36])

Todos estos comportamientos dificultan la obtención de muestras representativas y la evaluación del desarrollo de los riesgos relacionados con la presencia de la cianotoxinas para los habitantes y usuarios del agua [36]. Además, dificultan enormemente la elaboración de modelos numéricos de su dinámica.

2.1.3. Impacto ambiental, económico y sanitario

Las cianobacterias se pueden encontrar en casi todos los hábitats terrestres y acuáticos, ya sean océanos, agua dulce, suelo húmedo, rocas humedecidas, rocas antárticas e incluso dentro de piedras y conchas [37]. Algunas son endosimbiontes en líquenes, plantas, protistas o esponjas proporcionando energía al huésped. Otras incluso viven en el pelaje de los perezosos, proporcionándoles una forma de camuflaje. Las que se encuentran en medios acuáticos (tanto marinos como de agua dulce) son conocidas por sus extensas afloraciones, habitualmente con apariencia de pintura azul verdosa o de espuma.

El crecimiento de cianobacterias se ve favorecido en lagos y embalses, donde las aguas son tranquilas y tienen poco movimiento, y es más difícil que se dé en lugares con corrientes agitadas o turbulentas [38]. Además, su crecimiento también se ve favorecido con temperaturas altas, lo que beneficia el desarrollo de metabolitos potencialmente tóxicos. Por tanto, el cambio climático unido a otros factores como la eutrofización, la estratificación vertical, aporte de nutrientes (principalmente nitrógeno) y el aumento del dióxido de carbono atmosférico están provocando que las cianobacterias dominen cada vez más los ecosistemas acuáticos [39]. Esto repercute negativamente en el agua potable, la pesca, la irrigación y el uso recreativo del agua [26]. También daña gravemente el ecosistema acuático al causar la asfixia y muerte de peces y plantas, y comprometer la seguridad de animales y seres humanos.

Hay numerosos ejemplos de los importantes problemas en la calidad del agua causados por cianobacterias en todo el mundo. Algunos de ellos son el lago Taihu [40] en China, los lagos Erie [41] y Okeechobee [42] en Estados Unidos, el lago Victoria [43] en África (entre Kenia, Uganda y Tanzania) y el mar Báltico [44, 45, 46].

El problema conlleva además un alto coste económico en muchos sectores de negocios, como pueden ser el sector agrícola, la pesca, las plantas de tratamiento de agua, la indus-

tria del turismo, los servicios recreativos y los precios de los bienes inmuebles en las zonas costeras.

En cuanto al impacto en la salud de las personas, como ya sabemos, algunas cianobacterias pueden producir neurotoxinas, capaces de dañar el sistema nervioso; hepatotoxinas, que tienen la capacidad de dañar el hígado; y citotoxinas y endotoxinas, capaces de dañar la piel. Además, pueden causar daño renal, dolor abdominal, dificultad para respirar y posibilidad de crecimiento tumoral [9]. Por último, algunos estudios sugieren que la exposición significativa a altos niveles de toxinas producidas por las cianobacterias, como la toxina BMAA (*beta-N-methylamino-L-alanina*), puede encontrarse entre las causas de la esclerosis lateral amiotrófica (ELA) [47, 48].

Además, el peligro no se limita al entorno acuático más cercano, ya que se ha observado material extracelular de HABs de agua dulce en el agua y la atmósfera de lugares mucho más allá de los límites de la masa de agua donde surgieron.

2.1.4. Tratamiento actual

Existen varios métodos para tratar las cianobacterias. El tratamiento químico [49] es el más utilizado, aunque también es el más nocivo para el medio ambiente. Consiste en utilizar hipoclorito de calcio, sulfato de cobre, cupricida y simazina, que causan la muerte de las células de las cianobacterias. Sin embargo, la intervención química no resuelve el problema central, ya que las bacterias pueden reaparecer después del tratamiento, y la muerte de las cianobacterias no evita la liberación de las toxinas que se encuentran en su interior.

Otra opción es reducir la cantidad de nutrientes que entran en los sistemas de agua [50]. Esto requiere cambios radicales en las actividades urbanas, agrícolas e industriales que producen la contaminación por nutrientes. Además, debido a la variedad de origen de estos nutrientes, este método es muy difícil de llevar a cabo tan rápida y eficazmente como es necesario.

También existe una tecnología de ultrasonidos [50] que crea una barrera de sonido en las capas superiores del agua impidiendo que las cianobacterias alcancen la superficie, y por tanto evitando que su crecimiento continúe. Además, para mantener el equilibrio ecológico, este sistema no elimina completamente las cianobacterias, sino que las reduce hasta en un 90%. Aunque esta tecnología parece ser un método eficiente en el control de afloraciones, se encuentra todavía en fases de investigación.

2.2. Sistemas de detección y alerta temprana

Por todo lo explicado sobre los HABs, es fundamental implementar sistemas de alerta temprana (EWS, del inglés, *Early-Warning Systems*) que recopilen información sobre el estado del agua, identifiquen la presencia de niveles inadecuados de cianobacterias en la masa de agua, agilicen la determinación de cuándo y dónde se formarán los HABs, y faciliten su gestión por parte de los diferentes responsables (p.e. administraciones públicas o empresas de suministro) de la calidad del agua.

Tradicionalmente, se ha utilizado la determinación microscópica basada en las características morfológicas de las algas para el seguimiento de cianobacterias, eligiendo el lugar y

la profundidad de muestreo de forma visual, y realizando la extracción de muestras de forma manual. Sin embargo, esto requiere un equipo de gran tamaño, así como un usuario experto con el microscopio y el resto del aparataje. Además, este método no es capaz de diferenciar las cianobacterias productoras de toxinas de las no productoras debido a que pueden tener una morfología similar y a que las cianobacterias de la misma especie pueden producir o no toxinas. [51, 52]. Por lo tanto, han surgido técnicas con diferentes enfoques para desarrollar una herramienta de monitorización que analice y prediga la concentración de cianobacterias en el agua.

En la actualidad existen algunas alternativas a estos sistemas manuales de detección, monitorización, predicción y alerta temprana de cianobacterias. Debido a la importancia del problema, varios países están desarrollando EWS apoyados por sistemas predictivos [53] y respaldados por un conjunto de *modelos base* que describen el comportamiento de diferentes elementos, como la dinámica del agua, el crecimiento de las cianobacterias, sus desplazamientos verticales, etc. Estos modelos se integran en entornos de predicción, como veremos más adelante. A continuación, se enumeran algunos de los que se encuentran actualmente en uso.

Los modelos más utilizados para el modelado ecológico [54] de los HABs son los basados en datos, como las Redes Neuronales Artificiales [55, 56, 57], las Redes Bayesianas [58, 59] y la Programación Genética [60]. Todos ellos son capaces de proporcionar información sobre los fenómenos modelados desde un punto de vista estadístico, que es útil para hacer predicciones. Algunos de estos modelos intentan alcanzar predicciones precisas a través de lo que llamamos *modelos base* o están integrados en herramientas de *software* más genéricas como EEMS [17]. Las redes neuronales son ampliamente utilizadas en este ámbito debido a su poder predictivo y su capacidad para mapear la relación no lineal entre las variables del ecosistema [61]. En concreto, el algoritmo *backpropagation* y el perceptrón multicapa han sido los más aplicados para predecir la aparición de afloraciones de cianobacterias [62], consiguiendo hacerlo con incluso tres semanas de antelación.

También existen sistemas de predicción fundamentados en modelos matemáticos, como un Modelos Eulerianos [63], Modelos Lagrangianos [64], Modelos Basados en Reglas [65] o Modelos Probabilísticos [66, 67]. Todos ellos modelan las afloraciones de cianobacterias en base a ciertos factores como la temperatura del agua, el pH, la salinidad, el fósforo, el nitrógeno amoniacal y el oxígeno disuelto, para poder así identificar los puntos probables de crecimiento de estos organismos. Los Modelos Eulerianos destacan por estar compuestos por ecuaciones diferenciales que simulan dinámicas comunitarias en espacios, y los Modelos Lagrangianos por introducir la posibilidad de agregar diferentes clases de partículas con propiedades individualizadas, aunque lamentablemente, realizar simulaciones lagrangianas con una gran cantidad de partículas tiene un coste computacional elevado.

Por último, también existen modelos acuáticos mecanicistas basados en el conocimiento de las especies y de sus respuestas a los estímulos del ecosistema, como la disponibilidad de nutrientes o la estratificación térmica [68]. Estos modelos pueden resultar desafiantes debido al conocimiento incompleto que habitualmente se tiene de muchos de sus parámetros, pero también pueden ser más apropiados para realizar predicciones futuras basados en estadísticas. Hay estudios que hablan de una correlación positiva entre la fluorescencia de ficocianina y la biomasa de cianobacterias en el agua [69], observando cambios rápidos en la fluorescencia cuando el viento mueve concentraciones altas de cianobacterias, por lo

que podrían ser utilizados para desarrollar EWS más eficaces. A raíz de las conclusiones de dichos estudios, ha surgido un sistema portátil de medición de fluorescencia [70] que permite la detección de cianobacterias aprovechando los fotodiodos comerciales y un pequeño suministro de energía, y que transfiere posteriormente las lecturas de fluorescencia de las cianobacterias a un servidor de forma inalámbrica. Sin embargo, a día de hoy este método tiene más aplicaciones en la detección de afloraciones que en su predicción [71].

Todos los modelos comentados en este apartado suelen estar integrados dentro de herramientas *software* más genéricas que cuentan con interfaces gráficas de usuario (GUIs, del inglés, *Graphical User Interfaces*). Algunas de las más destacables son EEMS, MIKE y Delft3D. Más concretamente, EEMS [17] es una GUI que proporciona una amplia gama de herramientas de procesamiento previo y posterior para ayudar a desarrollar, calibrar y analizar modelos hidrodinámicos, de contaminantes de sedimentos y de eutrofización. MIKE Powered by DHI [16] es una gama de productos *software* que permiten analizar, modelar y simular con precisión diferentes tipos de problemas en entornos acuáticos. Por último, Delft3D [72] es un conjunto de herramientas *software* de código abierto que facilita el modelado de diferentes subsistemas hidrodinámicos, morfodinámicos, de ondas, de calidad del agua o basados en partículas.

Algunos ejemplos de sistemas relacionados con este trabajo son el Sistema Automático de Información de la Calidad del Agua de España [73], que consiste en una red de estaciones automáticas desplegadas en lugares críticos del sistema hidrográfico para obtener mediciones y proporcionar información genérica sobre la calidad del agua (no necesariamente sobre el estado y la evolución de cianobacterias) en tiempo real; la Red de monitorización de la Calidad del Agua del Centro de Investigación Ambiental del Sudeste [74], propiedad de la Universidad Internacional de Florida, que se enfoca en la monitorización costera del sur de la península de Florida e incluye algunas estaciones de medición automática; y la Red Nacional de Monitorización de la Calidad del Agua del Servicio Geológico de los Estados Unidos [75], que combina fuentes de datos y técnicas para monitorizar las aguas interiores de los Estados Unidos. Sin embargo, todos ellos tienen limitaciones, dado que sus estaciones de monitorización automática suelen ser inmóviles, se centran principalmente en el seguimiento de los datos medidos, no abordan la posterior predicción y gestión de los mismos, y su objetivo primordial no está centrado en las particularidades de las cianobacterias. En otras palabras, la información que obtienen carece de la resolución espacial necesaria para monitorizar, predecir y gestionar afloraciones de cianobacterias.

Así, teniendo en cuenta las ventajas y desventajas de los diferentes tipos de modelos explicados en esta sección, la propuesta de este trabajo es combinar los resultados proporcionados por varios modelos para hacer una representación precisa del estado de la masa de agua, permitiendo que el sistema responda rápidamente contra amenazas potenciales y brindando al operador las herramientas necesarias para manejar el problema. Se propone crear un sistema integrado que permita evaluar tanto modelos-base como elementos automáticos de tratamiento de cianobacterias, tales como el uso de USVs, boyas, etc. Para ello utilizaremos el formalismo DEVS por las facilidades que aporta al modelado de sistemas complejos. En la sección que viene a continuación hablaremos detalladamente de este formalismo.

2.3. DEVS

Comenzaremos esta sección introduciendo el concepto de *sistema*, definido por Wymore [76] como un conjunto compuesto por una entrada, un proceso que depende de un estado y que puede alterar la entrada, y una salida que sería la respuesta dada por dicho proceso a la entrada en función del estado del sistema. La figura 2.3 mostrada a continuación ilustra visualmente este concepto.

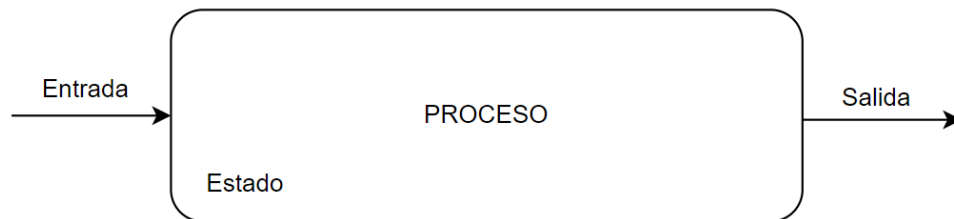


Figura 2.3: Representación gráfica de un sistema

Para tratar sistemas complejos surgen los formalismos que recogen el modelado y simulación de sistemas (M&S, del inglés, *Modeling and Simulation*), y para estudiar aquellos sistemas dinámicos cuya variación depende de eventos discretos que acontecen en el tiempo, aparece la *Simulación de Eventos Discretos* (DES, del inglés, *Discrete Event Simulation*), que se encuentran soportados por el análisis de las secuencias de eventos que tienen lugar en momentos determinados y que definen el cambio de estado del sistema.

Dentro del ámbito de los DES, en 1975 Bernard P. Zeigler formula el formalismo de Especificación de Sistemas de Eventos Discretos (DEVS, del inglés, *Discrete Event System Specification*), que se define como una subclase de los sistemas de Wymore que captura las características relevantes para los modelos de simulación orientados a eventos [19]. Más concretamente, DEVS especifica qué es un modelo, qué debe o no contener el mismo y cómo debe comportarse. Este paradigma es además único y universal para los modelos de sistemas de eventos discretos, lo que implica que cualquier sistema que acepte eventos temporales como entradas es equivalente a uno en DEVS (o lo que es lo mismo, su estructura y comportamiento pueden describirse usando este formalismo).

Por otro lado, desde la óptica del formalismo de DEVS, los sistemas se pueden dividir en modelos básicos o atómicos que definen el comportamiento de un sistema, y modelos compuestos o acoplados, que especifican la estructura del mismo [18].

Más concretamente, los modelos atómicos en DEVS procesan la entrada de eventos basándose en el estado actual y su condición, generando eventos de salida y desembocando en transiciones a otros estados.

Así, puede definirse un **modelo atómico** como la tupla

$$A = \langle X; Y; S; \delta_{ext}; \delta_{int}; \delta_{con}; \lambda; ta \rangle$$

donde X , Y , S , δ_{ext} , δ_{int} , δ_{con} , λ y ta se corresponden los siguientes conjuntos y funciones de DEVS:

- X representa el conjunto de entradas al sistema.

- Y representa el conjunto de salidas.
- S representa el conjunto de estados secuenciales, siendo

$$Q = (s, e), \quad s \in S, \quad 0 \leq e \leq ta(s)$$

el conjunto de estados, donde e es el tiempo transcurrido desde la última transición.

- δ_{ext} representa la función de transición externa. Se ejecuta automáticamente cuando un evento externo llega a uno de los puertos de entrada, cambiando el estado si es necesario.
- δ_{int} representa la función de transición interna. Se ejecuta automáticamente cuando un evento externo llega a uno de los puertos de entrada, cambiando el estado si es necesario.
- δ_{con} representa la función de confluencia. Esta transición decide el siguiente estado en casos de colisión entre eventos externos y eventos internos.
- λ representa la función de salida. Cuando el tiempo transcurrido desde la última función de salida es igual a la función de avance del tiempo, entonces λ es ejecutada automáticamente.
- ta representa la función de avance temporal.

A modo de ejemplo, la figura 2.4 muestra cómo puede construirse un modelo atómico en DEVS que capture el comportamiento del juego del Ping Pong. El juego se describe a partir de dos jugadores A y B que tienen, cada uno, dos posibles estados, *esperar* o *enviar*. El estado *esperar* tiene una duración de tiempo indefinida, y encontrándose en ese estado, una vez el jugador recibe la pelota a través del evento *?recibir*, son necesarios 0,1 segundos para poder devolverla a través del evento *!enviar*. Podemos resumir de manera sencilla la estructura del juego de la siguiente forma: el jugador A emite un evento de salida que se transmite al jugador B y viceversa.

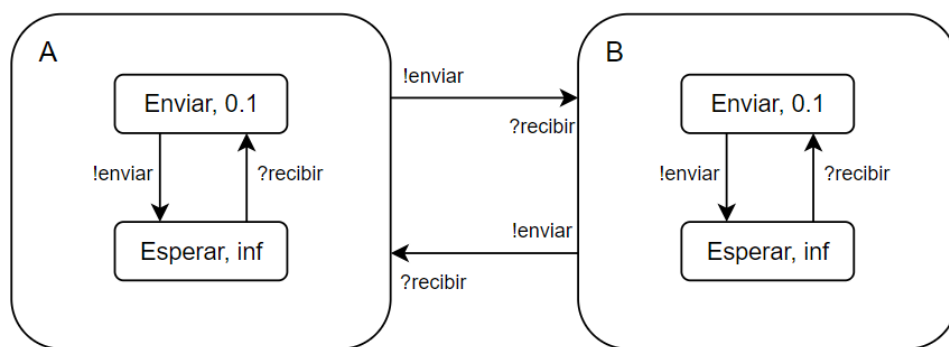


Figura 2.4: Representación del juego de Ping Pong en DEVS

Para este juego es posible entonces proporcionar explícitamente los siguientes valores para los conjuntos y funciones que definen el modelo atómico:

- $X = \{?recibir\}$
- $Y = \{!enviar\}$

- $s_0 = (enviar, 0.1)$
- $S = \{(d, \sigma) : d \in \{esperar, enviar\}, \sigma \in \mathbb{T}^\infty = [0, \infty]\}$
- $\delta_{ext}(((esperar, \sigma), t_e), ?recibir) = (enviar, 0.1)$
- $\delta_{int}(enviar, \sigma) = (esperar, \infty)$
- $\delta_{int}(esperar, \sigma) = (enviar, 0.1)$
- $\lambda(enviar, \sigma) = !enviar$
- $\lambda(esperar, \sigma) = \emptyset$
- $ta(s) = \sigma, \forall s \in S$

De forma análoga a como lo hemos hecho con los modelos atómicos, podemos definir un **modelo acoplado** como la tupla

$$M = \langle X; Y; C; EIC; EOC; IC \rangle$$

donde X, Y, C, EIC, EOC, IC se corresponden con los conjuntos y relaciones de DEVS especificados a continuación:

- X representa el conjunto de entradas al modelo.
- Y representa el conjunto de salidas.
- C representa el conjunto de modelos atómicos/acoplados que lo constituyen.
- EIC representa el conjunto que define la relación de acoplamiento externa, es decir qué entradas externas al modelo acoplado M se relacionan con entradas de modelos de C .
- EOC representa la relación de acoplamiento externa, es decir la relación entre las salidas de los modelos componentes de C y las salidas de M .
- IC representa la relación de acoplamiento interna, es decir las relaciones de salida de los modelos componentes $c_i \in C$ con las salidas de $c_j \in C$ para $i \neq j$.

A modo de ejemplo, la figura 2.5 representa la estructura de un modelo acoplado sencillo y sus interconexiones detalladas.

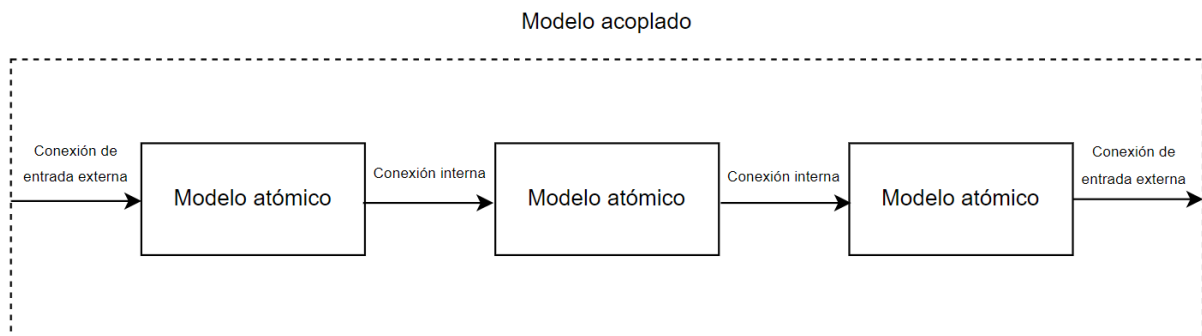


Figura 2.5: Representación de un modelo acoplado

Notemos que cada componente, ya sea atómico o acoplado, puede tener puertos de entrada y salida, y estos se pueden conectar entre ellos. Así, tanto los modelos atómicos como los acoplados pueden conectarse y agregarse unos con otros. [18].

DEVS es apropiado para la definición e implementación de sistemas complejos como el que se aborda en este TFM. Entre sus múltiples ventajas destacan la integridad, la verificabilidad, la extensibilidad y la mantenibilidad. Otra de las ventajas de DEVS que explotaremos ampliamente en los dos siguientes capítulos es que permite definir formalmente tanto la estructura (o arquitectura) de un sistema como su comportamiento (o implementación).

Capítulo 3

Arquitectura del sistema

En este capítulo introduciremos la arquitectura del modelo propuesto como solución integral para la monitorización y gestión de cianobacterias, que forma parte del Proyecto Sinérgico de la Comunidad de Madrid, titulado *Hacia un sistema Integral para la Alerta y GESTión de BLOOMs de cianobacterias en aguas continentales, IA-GES-BLOOM-CM*. El proyecto está siendo desarrollado por el Grupo de Investigación de Ingeniería de Sistemas, Control, Automática y Robótica (ISCAR) de la Universidad Complutense de Madrid (UCM) en colaboración con un equipo de investigación de la Facultad de Ciencias de la Universidad Autónoma de Madrid (UAM). Como podrá observarse en este capítulo, *IA-GES-BLOOM-CM* es un proyecto grande, que involucra varias ramas de conocimiento y que abarca un gran espectro de tareas a desarrollar. Por ello, en este capítulo daremos una imagen general del sistema de modelado y simulación y de su arquitectura, en lo que se ha colaborado activamente en este trabajo. En el capítulo 4 daremos detalles de la implementación del subsistema de predicción y análisis de datos, desarrollado íntegramente en este TFM.

3.1. Imagen general

La visión general del proyecto queda reflejada mediante un modelo holístico e integrador definido siguiendo una metodología MBSE, que incluye no solo el uso de *modelos base*, sino también la infraestructura del sistema de alerta temprana (EWS). La figura 3.1 muestra la concepción del marco de simulación, siguiendo una arquitectura basada en internet de las cosas (IoT, del inglés, *Internet of Things*) y mediante el uso de gemelos digitales (DT, del inglés, *Digital Twin*).

El marco de modelado creado para simular y operar este sistema recibe el nombre de *DEVS-BLOOM* y ha sido diseñado para que algunos componentes virtuales se construyan como un gemelo digital y puedan ser reemplazados directamente por sus homólogos reales, siguiendo los principios de la metodología MBSE. El sistema tiene por objetivo la monitorización en tiempo real y la predicción de afloraciones de cianobacterias. Su desarrollo se ha realizado a través de DEVS [77], el conocido formalismo de modelado y simulación.

Este marco está dividido en las tres capas clásicas de IoT: *Edge*, *Fog* y *Cloud*, como puede observarse en la figura 3.1. Además, *DEVS-BLOOM* se alimenta con datos que pueden

provenir de un cuerpo de agua real o de una base de datos que, a su vez, puede almacenar datos reales o sintéticos. Esto permite que el sistema funcione en modo virtual, real o híbrido en función de si los datos provienen exclusivamente del cuerpo de agua real, de la base de datos, o de ambos.

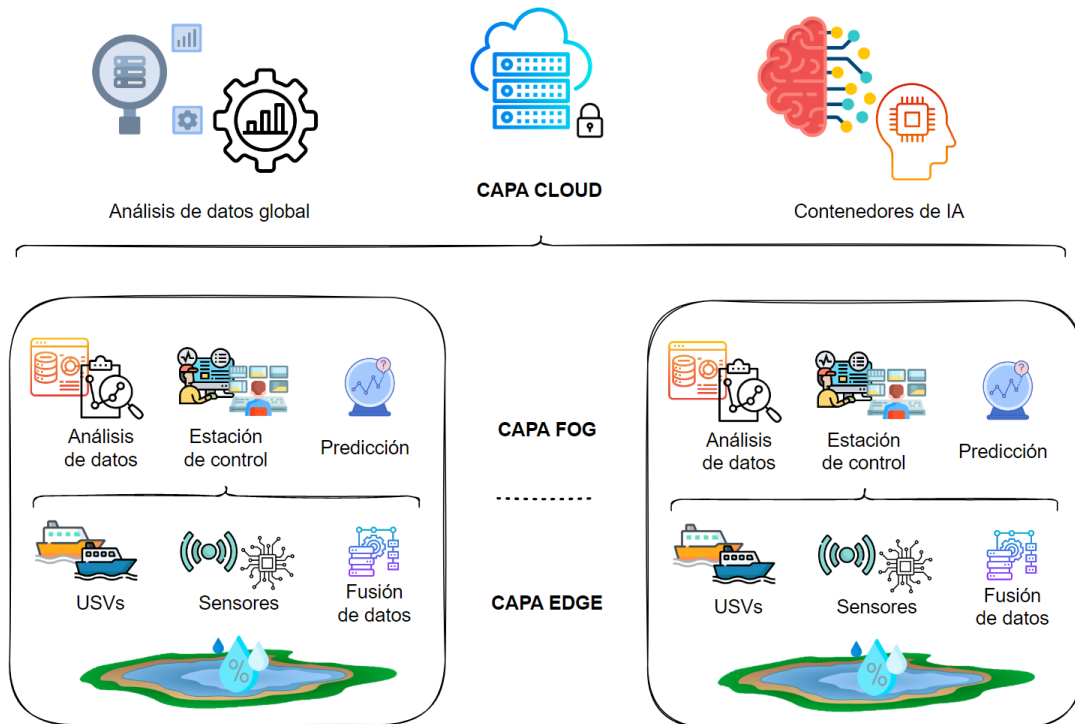


Figura 3.1: Esquema general del marco propuesto

Así, las masas de agua son monitorizadas en la capa *Edge* por un conjunto de sensores, incluidos a bordo de embarcaciones automatizadas, denominados en adelante USVs (del inglés, *Unmanned Surface Vehicle*), que continuamente envían datos a la estación de control de tierra, denominada en adelante GCS (del inglés, *Ground Control Station*), ubicada en la capa *Fog*. En esta capa intermedia los expertos del dominio pueden analizar los datos, ejecutar algunos modelos, hacer pruebas o planificar las trayectorias de los USVs, usando algunos servicios de predicción, análisis de datos, etc.

El marco admite escalabilidad horizontal, permitiendo agregar más cuerpos de agua y más USVs con el apoyo de una capa *Cloud*, donde las autoridades responsables de la gestión de las masas de agua pueden estudiar diferentes informes y tomar decisiones de alto nivel, apoyados de nuevo en servicios de análisis de datos, entrenamiento del modelo de predicción, etc.

Hasta donde se sabe, esta investigación es la primera relacionada con el desarrollo de soluciones integradas basadas en modelos para la gestión de cianobacterias, ya que combina el uso de *modelos base* con la ayuda de modelos de la infraestructura como sensores y USVs.

En las secciones que vienen a continuación se describen con más detalle los elementos de la figura 3.1 desde una perspectiva más técnica.

3.2. Arquitectura en DEVS

Como ya comentamos en la sección 2.3, DEVS es un formalismo modular y jerárquico para modelar sistemas de eventos discretos basados en la teoría de conjuntos [77], que proporciona un marco para el modelado de información y que aporta numerosas ventajas al análisis y diseño de sistemas complejos, como son la integridad, la verificabilidad, la extensibilidad y la mantenibilidad.

DEVS-BLOOM se implementa y ejecuta mediante xDEVS, un simulador DEVS multi-plataforma [78] que ofrece un buen equilibrio entre facilidad de uso y rendimiento. En particular, utiliza la rama xDEVS/Python de dicho simulador. De esta forma, el marco del sistema puede usar tiempo virtual o real y puede ejecutar simulaciones secuenciales o paralelas.

Mientras que la figura 3.1 ilustra la imagen general de la arquitectura de *DEVS-BLOOM*, la figura 3.2 muestra su estructura en DEVS. Como puede observarse, el sistema cuenta con un modelo acoplado en la capa *Edge* (*USV*), un modelo acoplado en la capa *Fog* (*FOG_SERVER*) y un modelo acoplado en la capa *Cloud* (*CLOUD*).

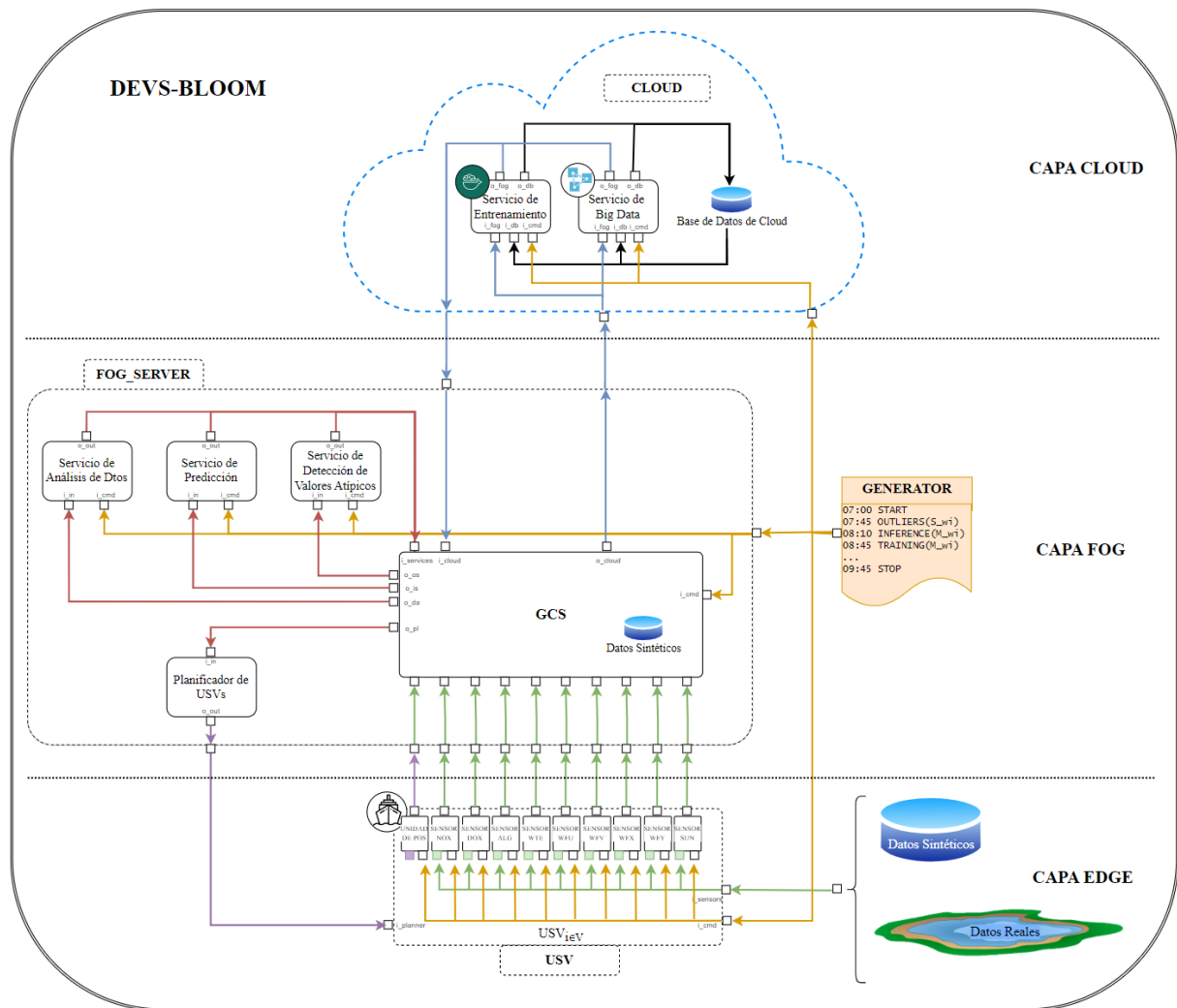


Figura 3.2: Arquitectura del sistema en DEVS

Además, en la figura 3.2 puede observarse un modelo atómico singular etiquetado como *Generator*, que se encuentra representado en la capa *Fog* y que consiste en una fuente que lee de un archivo de texto todos los eventos que serán inyectados en el proceso de simulación a través de su puerto de salida. Cada instrucción en el archivo de simulación representa un evento de entrada compuesto por una marca de tiempo que indica el instante virtual en el que se activará ese evento, el tipo de comando asociado a él y los argumentos que necesita. Como resultado, este archivo replica el conjunto de eventos externos que podrían ocurrir en un escenario del mundo real.

Como se muestra en la figura 3.2, el fichero asociado al modelo atómico *Generator* siempre empieza y termina con las instrucciones de inicialización (**START**) y finalización (**STOP**) del experimento de simulación que, como sus nombres indican, lanzan y paran, respectivamente, la simulación. Este modelo atómico es el encargado de enviar las instrucciones de la simulación a las tres capas del sistema. La mayoría de los servicios se activan con la instrucción de **START**, pero algunos como la detección de valores atípicos o la predicción de HABs pueden activarse posteriormente. De este archivo de simulación cabe destacar que es un elemento virtual puro sin homólogo exacto en el mundo real, y que no pertenece a ninguna de las capas del sistema, aunque por comodidad se haya representado en la capa *Fog*. En las siguientes secciones, describimos el resto de los componentes incluidos en el sistema en función de la capa en la que se ubiquen.

3.2.1. Capa *Edge*

La capa *Edge* contiene el modelo *USV*, que a su vez encapsula los modelos atómicos de los sensores embarcados. En la figura 3.2 no pueden apreciarse los detalles, por lo que presentamos en la figura 3.3 el mismo esquema de la capa *Edge* ampliado.

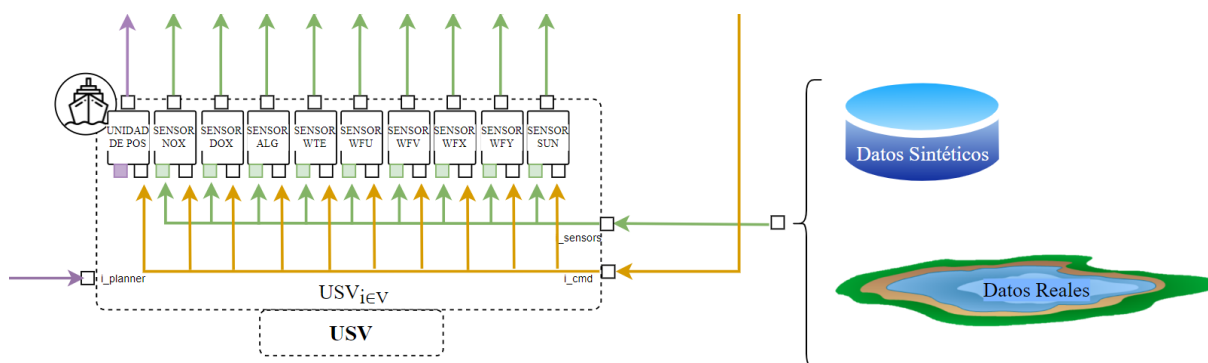


Figura 3.3: Arquitectura de la capa *Edge* en DEVS

Por un lado, los sensores miden los diferentes parámetros que afectan al crecimiento de cianobacterias, como pueden ser la concentración de nitrógeno y de oxígeno, o la temperatura y velocidad del agua. Los sensores reales están colocados dentro de los USVs, lo que permite ir tomando medidas en diferentes ubicaciones de la masa de agua. Por ello, sus gemelos digitales se han ubicado también como modelos atómicos dentro del modelo acoplado *USV*. Esto permite que los sensores lean los datos de una base de datos sintética, o bien directamente de la masa de agua real, tal y como puede observarse en la figura 3.3. Los sensores virtuales imitan el comportamiento del sensor real, introduciendo los

correspondientes retrasos, ruido, errores de saturación, envejecimiento, etc. Todos estos parámetros opcionales se definen a través de un archivo de configuración. Como la mayoría de los componentes del marco de modelado, los sensores son modelos pasivos, es decir, se despiertan cuando el USV recibe un evento **START** del archivo de simulación por el puerto i_{cmd} .

Concretamente, el sistema cuenta con los siguientes nueve sensores creados como DT y una unidad de posicionamiento que calcula las coordenadas del USV:

- Sensor de nitrógeno (etiquetado en la figura 3.2 como *NOX*).
- Sensor de oxígeno disuelto (etiquetado como *DOX*).
- Sensor de densidad de cianobacterias (etiquetado como *ALG*).
- Sensor de temperatura del agua (etiquetado como *WTE*).
- Sensor de velocidad del agua dirección este/oeste (etiquetado como *WFU*).
- Sensor de velocidad del agua dirección norte/sur (etiquetado como *WFV*).
- Sensor de velocidad del viento dirección este/oeste (etiquetado como *WFX*).
- Sensor de velocidad del viento dirección norte/sur (etiquetado como *WFY*).
- Sensor de radiación solar (etiquetado como *SUN*).
- Unidad de cálculo de posición y batería (etiquetado como *Unidad de pos.*)

Además, cada uno de los gemelos digitales anteriores tiene un elemento del aparataje real que le corresponde y que forma parte del USV en la realidad. Estos son:

- Sonda de nitrógeno, que mide la densidad de nitratos disueltos en el agua, ya que es el principal alimento de las cianobacterias.
- Sonda de oxígeno disuelto, que se encarga de medir la densidad de oxígeno en el agua. Como las algas producen oxígeno por fotosíntesis, donde hay oxígeno en abundancia es probable que haya HABS.
- Sensor de ficocianina, que detecta la presencia de cianobacterias y facilita la medición de su densidad.
- Sensor de temperatura para medir los grados a los que se encuentra la masa de agua en el punto de medición.
- Caudalímetro, que mide la velocidad y la dirección del agua con respecto al USV.
- Anemómetro y veleta para medir la velocidad y la dirección del viento con respecto al barco.
- Sensor de radiación solar, que mide los niveles de radiación en el punto en el que se toma la medida.
- Unidad de posicionamiento inercial, que mide la posición y velocidad del USV.
- Unidad de potencia, que incluye paneles solares, cargadores y condensadores encargados de recargar las baterías de la embarcación cuando recibe radiación solar.

Estos dos últimos, encargados de la posición y la batería del USV, no tienen un sensor asociado como gemelo digital, sino un modelo atómico, ya que en la simulación virtual ambos parámetros se calculan matemáticamente y no requieren que su información sea consultada en una base de datos.

Cabe destacar que *DEVS-BLOOM* puede manejar un conjunto de estos USVs, que reciben comunicaciones de la capa *Fog* o de otro USV, ya sea para evitar colisiones o para compartir información entre ellos. La figura 3.3 solo ilustra uno de los USVs que puede haber en el sistema. El modelo de cada uno de ellos tiene una serie de puertos de salida por los que transmite la información recopilada por sus sensores o la información del estado del propio USV. Es decir, por los puertos de salida el USV transmite información relacionada con variables de estado de la masa de agua que son útiles para la predicción del *bloom* (como la temperatura del agua o la cantidad de oxígeno disuelto) e información relacionada con el estado del USV (como el estado de la batería o la posición). El modelo acoplado *USV* cuenta además con puertos de entrada por los que recibe las instrucciones del *Generator* y las órdenes del planificador, que se encuentra en la capa *Fog* y se encarga de definir la ruta del USV.

Así, cada gemelo digital de la capa *Edge* transmite eventos que siguen una estructura predefinida y genérica que encapsula las medidas, comandos o cualquier otra información relevante. Esa estructura de evento genérica consta de un **source** y un **target** que identifican respectivamente el origen y el destino del evento, un **timestamp** con la hora real del evento, y un conjunto de pares clave-valor con el identificador de cada dato y su correspondiente medida. Un ejemplo de eventos enviados desde un USV a la capa *Fog* se muestra la figura 3.4. En la primera línea de dicha figura se puede observar el **timestamp** acompañado del **source**, que es el USV, y del **target**, que es el modelo atómico GCS de la capa *Fog* encargado del control del sistema. A continuación se observan los pares clave-valor que el modelo acoplado *USV* envía al *GCS*. En el caso de la imagen estos pares clave-valor son los datos recopilados por los sensores del USV.

```
-----
USV->GCS: DateTime: 2008-09-06 23:30:00
-----
USV->GCS: Data: {'usv_power': 0.9502270597267416, 'usv_lon': -122.21788464013156, 'usv_lat': 47.50659272073676, 'SensorsOn': True}
Sensor: WFX: DateTime: 2008-09-06 23:30:03
Sensor: WFX, Value = -1.141825795173645
Sensor: WFY: DateTime: 2008-09-06 23:30:03
Sensor: WFY, Value = 0.15544059872627258
Sensor: WTE: DateTime: 2008-09-06 23:30:04
Sensor: WTE, Value = 20.49156010245863
Sensor: WFU: DateTime: 2008-09-06 23:30:04
Sensor: WFU, Value = -0.002890722340503991
Sensor: WfV: DateTime: 2008-09-06 23:30:04
Sensor: WfV, Value = -0.0032700625547429536
Sensor: DOX: DateTime: 2008-09-06 23:30:05
Sensor: DOX, Value = 19.15842122329674
Sensor: NOX: DateTime: 2008-09-06 23:30:06
Sensor: NOX, Value = 0.0006827225978144596
Sensor: ALG: DateTime: 2008-09-06 23:30:07
Sensor: ALG, Value = 5.400799030150721
Sensor: SUN: DateTime: 2008-09-06 23:30:09
Sensor: SUN, Value = 0.0
```

Figura 3.4: Ejemplos de eventos transmitidos por el USV de la capa *Edge* al GCS de la capa *Fog*

3.2.2. Capa *Fog*

La capa *Fog* se simula a través del modelo acoplado *FOG_SERVER*. Aquí, los operadores y expertos en el ámbito de los *blooms* de cianobacterias en el entorno acuático analizan los datos y toman decisiones. Como muestra la figura 3.5, el modelo acoplado *FOG_SERVER* tiene varios puertos de entrada para recibir los eventos enviados por los componentes de la capa *Edge* y de la capa *Cloud*, así como las instrucciones del Generador. Además, tiene varios puertos de salida que envían datos a la capa *Cloud* o a la capa *Edge*.

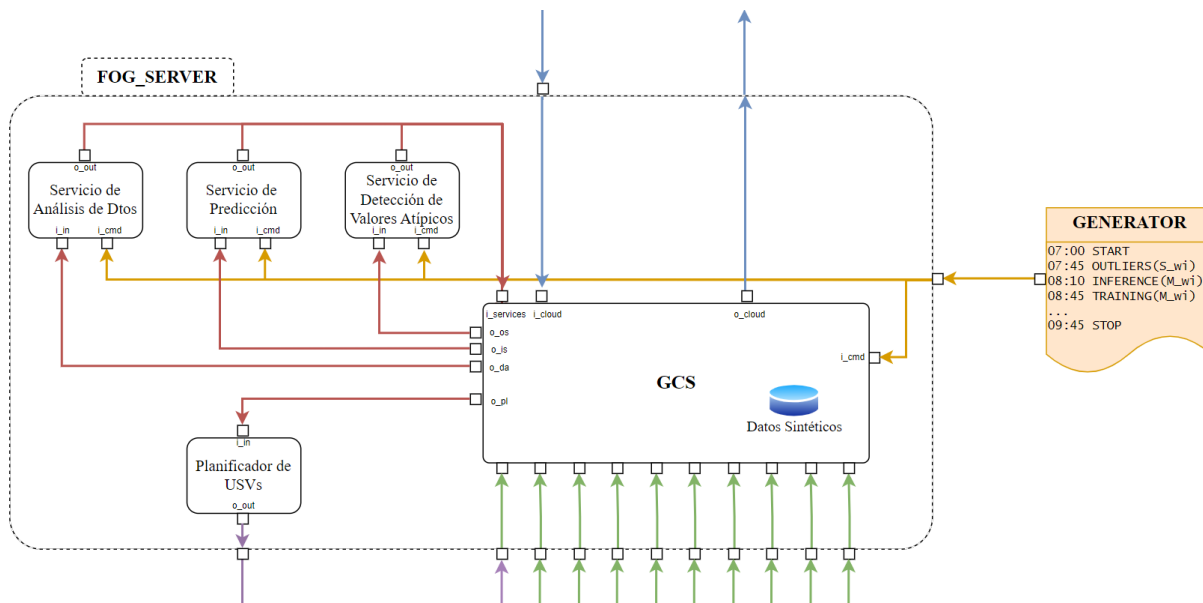


Figura 3.5: Arquitectura de la capa *Fog* en DEVS

Este modelo acoplado contiene en su interior una serie de modelos atómicos. Su modelo principal es el GCS, ya que representa el núcleo de la infraestructura informática de la estación de control de tierra. Por lo general, se despliega sobre una estación de trabajo estática o un ordenador portátil conectado a la red local. El GCS recibe comandos de simulación del archivo de simulación, que le indican cuándo comenzar a leer datos, ejecutar un servicio sobre los modelos predictivos, calcular la trayectoria de un USV, etc. Cuando comienza la simulación, se reciben los datos de los sensores a través de los puertos de entrada provenientes de la capa *Edge* y se almacenan en la base de datos local. Estos datos se envían al modelo atómico *CLOUD* a intervalos predefinidos. Por otro lado, cuando se recibe una solicitud de servicio desde el archivo de simulación, se propaga al modelo atómico correspondiente a través del puerto asignado. Los datos reales o predichos también se almacenan en la base de datos local y se envían regularmente al modelo atómico *CLOUD*.

El modelo acoplado *FOG_SERVER* contiene, además del GCS, un conjunto de modelos atómicos encargados de ejecutar diferentes servicios. Todos ellos reciben comandos por el puerto de entrada i_{in} y envían los resultados a través de los puertos de salida o_{out} . Los resultados se envían de regreso a los modelos GCS o USV, en función del servicio llevado a cabo. Actualmente hay implementados 4 servicios: uno, etiquetado en la figura 3.5 como *Servicio de Detección de Valores Atípicos*, para detectar y corregir valores atípicos de la distribución de la predicción; otro, etiquetado como *Servicio de Predicción*; para realizar

predicciones y calcular la probabilidad de formación y ubicación de cianobacterias en la masa de agua; un tercer servicio, denominado *Servicio de Análisis de Datos*, para analizar la información contenida en la base de datos y generar informes; y un último servicio, etiquetado como *Planificador de USVs*, responsable de calcular las rutas que deben seguir los USVs. Más concretamente, este último servicio utiliza los resultados de los cálculos realizados por el servicio de predicción del estado de las cianobacterias para obtener los puntos de paso (*waypoints*) que debe visitar y las trayectorias que deben seguir las embarcaciones. La forma de proceder depende del objetivo final del planificador, que por ejemplo puede devolver como próximo *waypoint* la zona esperable de máxima concentración de cianobacterias u obtener aquella trayectoria que maximiza la probabilidad de detectarlas haciendo que la embarcación recorra/utilice la mínima distancia/tiempo para monitorizar el estado del embalse [79].

3.2.3. Capa *Cloud*

El modelo atómico *CLOUD* se encuentra en etapas iniciales de desarrollo. Este modelo recibe todos los datos de diferentes cuerpos de agua y los almacena en la base de datos. Al igual que en el modelo acoplado de la capa *Fog*, el modelo atómico *Cloud* puede ejecutar diferentes servicios, entre los que se incluyen el uso de técnicas de análisis de *big data* sobre todos los datos almacenados en la capa *Cloud*, o la ejecución de servicios de entrenamiento de los modelos de predicción ubicados en la capa *Fog*. En cualquier caso, estas acciones son desencadenadas por el archivo de la capa *Fog* que contiene las instrucciones de la simulación.

Además, cabe destacar que la capa *Cloud* está capacitada para gestionar varias masas de agua de forma simultánea. En ese caso, el marco de modelado y simulación *DEVS_BLOOM* solo debe replicar, para cada masa de agua, los modelos correspondientes de las capas inferiores de la infraestructura de IoT.

En la figura 3.6 puede observarse que, como hemos comentado anteriormente, la capa *Cloud* recibe datos del modelo acoplado *FOG_SERVER* para poder realizar los servicios de entrenamiento y *big data*. Actualmente, estos servicios se ejecutan en una máquina local, aunque está previsto que en el futuro se ejecuten en contenedores distribuidos en la nube para resolver el problema de la escalabilidad.

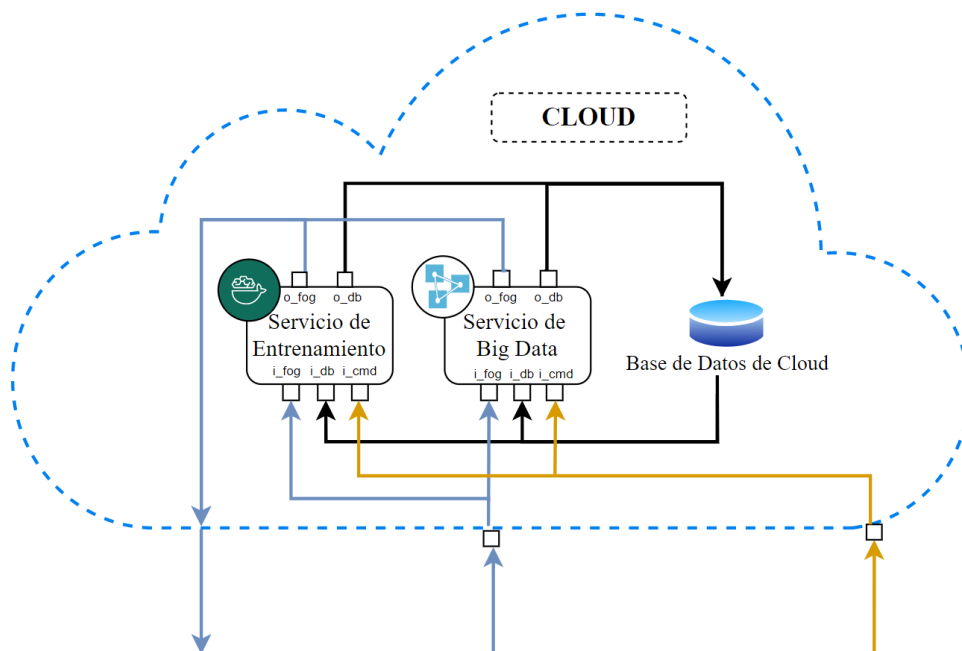


Figura 3.6: Arquitectura de la capa *Cloud* en DEVS

Para terminar el capítulo, cabe destacar que, como se ha mencionado al inicio, todo este marco de modelado y simulación está siendo desarrollado por diferentes investigadores del grupo de investigación ISCAR, con los que he colaborado de forma global en la puesta a punto, refinamiento y depuración de diferentes partes y aspectos del sistema.

Además, en el siguiente capítulo (4) se detallarán aquellos modelos relacionados con los servicios de predicción, análisis de datos y entrenamiento, cuya funcionalidad actual es fruto de las labores específicas desarrolladas durante este Trabajo Final de Máster.

Lo implementado en este trabajo puede verse en el repositorio [80] de GitHub. Para ello, basta con descargar la versión etiquetada como *tfm-beatriz* o clonarla mediante línea de comandos:

```
git clone --depth 1 --branch tfm-beatriz
git@github.com:iscar-ucm/devs-bloom-public.git
```

y posteriormente seguir las instrucciones del *README* para resolver las dependencias.

Capítulo 4

Implementación de los subsistemas de predicción y análisis

En este capítulo describiremos y profundizaremos en aquellos modelos de la arquitectura DEVS desarrollada en el proyecto DEVS-BLOOM que se encuentran directamente relacionados con la predicción del estado de los *blooms* de cianobacterias y con la generación de informes de datos. Por ello, se comenzará hablando del servicio de la localización del afloramiento de las cianobacterias. Posteriormente describiremos el servicio de análisis de los datos relativos al *bloom*. Finalizaremos explicando el servicio de entrenamiento del modelo predictivo.

4.1. Predicción del *bloom*

La función principal del subsistema de predicción es calcular las probabilidades de formación y localización de las floraciones en cada masa de agua. Este modelo base representa la densidad y la posición de las floraciones en función de determinados factores, como la cantidad de nitrógeno y oxígeno disueltos o la velocidad del agua. Para ello, se utiliza el siguiente sistema de ecuaciones diferenciales:

$$\frac{dr(t)}{dt} = k_{growth} \cdot u(t) - k_{decline} \cdot (r(t) - r(0)) \quad (4.1)$$

$$\frac{dx(t)}{dt} = k_{speed} \cdot v_x(t) \quad (4.2)$$

$$\frac{dy(t)}{dt} = k_{speed} \cdot v_y(t) \quad (4.3)$$

En particular, en la ecuación (4.1), r representa la densidad de la floración, mientras que u es la concentración de nutrientes presentes en el agua. Además, en las ecuaciones (4.2) y (4.3), x e y son las coordenadas de la posición de la floración a una altura determinada, mientras que v_x y v_y son la velocidad del agua en las mismas coordenadas. En cuanto a las constantes de esas ecuaciones, k_{growth} es la constante de crecimiento del *bloom*, $k_{decline}$ es la constante de decrecimiento, y k_{speed} representa el porcentaje de la velocidad del agua que se transfiere a las cianobacterias.

La concentración de nutrientes u se calcula del siguiente modo:

$$u(t) = k_{photo} \cdot photo(t) + k_{breath} \cdot breath(t), \quad (4.4)$$

donde *breath* representa la respiración y *photo* representa la fotosíntesis, y k_{breath} y k_{photo} son las constantes que miden el porcentaje de respiración y fotosíntesis que afectan efectivamente a la cantidad de nutrientes disueltos en el agua.

Por tanto, sustituyendo en la ecuación (4.1) la ecuación anterior obtenemos la expresión:

$$\frac{dr(t)}{dt} = k_{growth} \cdot (k_{photo} \cdot photo + k_{breath} \cdot breath) - k_{decline} \cdot (r(t) - r(0)) \quad (4.5)$$

Por último cabe explicar que *breath* y *photo* se calculan dinámicamente, para cada paso de tiempo h , mediante las siguientes expresiones en función de los datos obtenidos por los sensores de nitrógeno, oxígeno disuelto y radiación solar:

$$\begin{aligned} breath &= nitrate \cdot dissolved_oxygen \\ photo &= nitrate \cdot solar_radiation, \end{aligned} \quad (4.6)$$

ya que las cianobacterias necesitan nitratos y oxígeno para realizar la respiración, y precisan de nitratos y energía solar para realizar la fotosíntesis.

La ecuación (4.5) modela la evolución de la densidad de la floración en función de 4 constantes: k_{growth} , $k_{decline}$, k_{breath} y k_{photo} , que pueden traducirse en tres parámetros que dotan al sistema de tres grados de libertad, como puede verse en la siguiente expresión donde $k_1 = k_{growth} \cdot k_{photo}$, $k_2 = k_{growth} \cdot k_{breath}$ y $k_3 = k_{decline}$:

$$\frac{dr(t)}{dt} = k_1 \cdot photo(t) + k_2 \cdot breath(t) - k_3 \cdot (r(t) - r(0)) \quad (4.7)$$

El valor de todas las constantes del modelo se obtiene entrenando el sistema, que partirá de unos valores dados para ellas. Estos valores se ajustan cada vez que el error entre el valor real y el valor simulado supera un umbral determinado.

Inicialmente las ecuaciones diferenciales propuestas se resolvían utilizando el método de Euler. Este es un procedimiento mono-paso para la aproximación numérica de la solución a un cierto problema de valor inicial del tipo:

$$\begin{cases} x'(t) = f(t, x(t)), & t \in [t_0, T] \\ x(t_0) = \xi_0 \end{cases} \quad (4.8)$$

donde $\xi_0 \in \mathbb{R}^n$, $t_0 \in \mathbb{R}$, $T \in \mathbb{R}$ y f es una función continua y lipschitziana en la segunda variable con constante de Lipschitz $L > 0$, lo que garantiza que el problema de valor inicial planteado tiene existencia y unicidad de solución.

Una vez planteado el problema y para un $N \in \mathbb{N}$ dado, se divide el intervalo $[t_0, T]$ en N partes iguales por medio de puntos equiespaciados de la siguiente manera:

$$t_i = t_0 + ih \quad \text{para } i = 0, 1, \dots, N$$

donde

$$h = \frac{T - t_0}{N}.$$

Por lo tanto, teniendo en cuenta que

$$x'(t) = \lim_{h \rightarrow 0} \frac{x(t+h) - x(t)}{h} \in \mathbb{R}^n$$

para valores de h suficientemente pequeños se verifica que

$$x'(t) \simeq \frac{x(t+h) - x(t)}{h}.$$

Sustituyendo en la expresión anterior la función $f(\cdot)$ del problema de valor inicial en cada t_i por $x'(t_i) = f(t_i, x(t_i))$ se obtiene

$$\frac{x(t_{i+1}) - x(t_i)}{h} \simeq f(t_i, x(t_i)).$$

A continuación, podemos reescribir la ecuación anterior como

$$\frac{x_{i+1} - x_i}{h} = f(t_i, x_i) \tag{4.9}$$

donde x_i es una aproximación de $x(t_i)$, que será mejor cuanto menor sea el valor del paso h (o el número de pasos N realizados).

Finalmente, despejando x_{i+1} de la ecuación (4.9) obtenemos el esquema numérico del método del Método de Euler:

$$\begin{aligned} \text{Paso 1: } & x_0 \simeq \xi_0 \\ \text{Paso 2: } & x_{i+1} = x_i + h \cdot f(t_i, x_i) \quad \text{para } i = 0, 1, \dots, N \end{aligned}$$

Utilizando por tanto este método, se calculaban las aproximaciones de Euler de las 3 ecuaciones propuestas. Comenzamos aproximando la ecuación (4.1) mediante la ecuación (4.9) para:

$$\frac{r_{i+1} - r_i}{h} = k_{growth} \cdot u - k_{decline} \cdot r_i$$

Despejando r_{i+1} obtenemos

$$r_{i+1} = r_i + h \cdot k_{growth} \cdot u - h \cdot k_{decline} \cdot r_i \tag{4.10}$$

Realizando un proceso análogo para las ecuaciones (4.2) y (4.3):

$$\begin{cases} \frac{x_{i+1} - x_i}{h} = k_{speed} \cdot v_x \\ \frac{y_{i+1} - y_i}{h} = k_{speed} \cdot v_y \end{cases}$$

Despejando x_{i+1} y y_{i+1} concluimos que:

$$x_{i+1} = x_i + h \cdot k_{speed} \cdot u_x \quad (4.11)$$

$$y_{i+1} = y_i + h \cdot k_{speed} \cdot u_y \quad (4.12)$$

Sin embargo, tras utilizar otros métodos para resolver las ecuaciones diferenciales y comparar los resultados, se decidió utilizar para este propósito la función *odeint* de la biblioteca LSODE. Esta biblioteca utiliza un método predictor-corrector de Adams que combina una variante explícita con otra implícita. La variante explícita predice una aproximación, mientras que la implícita la corrige.

Para finalizar, destacaremos que el modelo base descrito en esta sección se encapsula en un modelo atómico. Su propósito es verificar y validar todo el EWS y después mejorar los modelos base manteniendo las mismas entradas y salidas en los componentes DEVS. En este caso, la transición interna incluye el modelo de ecuaciones diferenciales. El modelo atómico DEVs que encapsula las ecuaciones anteriores recibe los datos transmitidos por los sensores relativos a las variables u , v_x y v_y . Una vez calculada la densidad de la floración con la ecuación (4.1), si la densidad es suficiente para considerar que hay afloramiento masivo, se calcula su nueva posición con las ecuaciones (4.2) y (4.3). En caso contrario, la posición se restablece a la posición de búsqueda inicial.

4.2. Análisis de datos

El subsistema de análisis de datos se ubica en la capa *Fog* y en la capa *Cloud* y tiene como cometido analizar la información almacenada en la base de datos y generar informes. Actualmente hay implementados dos tipos de informes, los de la capa *Fog* y los de la capa *Cloud*. En las siguientes secciones profundizaremos un poco más en cada uno de ellos.

4.2.1. Informes de la capa *Fog*

Los informes de la capa *Fog* consisten en diferentes gráficas que ayudan a visualizar todo lo ocurrido durante la simulación. Por ahora, el informe consta de 22 gráficas, 6 de ellos dinámicos, desarrollados utilizando la librería *matplotlib* de Python.

Cada informe de esta capa se divide en cinco grandes secciones: una inicial con gráficas dinámicas para que el experto de dominio pueda hacerse una idea visual de la simulación. Una segunda sección donde se muestran gráficas con los datos de la evolución de las medidas de los sensores. Una tercera donde ocurre lo mismo que en la segunda sección pero con las variables del *bloom*. Una cuarta similar a la de las dos secciones previas con la evolución de las variables del USV y una quinta para la evolución del modelo predictivo. Hagamos, pues, un recorrido por cada una de estas secciones para entender cómo se han diseñado las gráficas y cuál es su aportación al marco de modelado y simulación *DEVSBLOOM*.

Gráficas dinámicas

El primero de los gráficos que aparece en el informe de la capa *Fog* es un mapa animado que muestra detalladamente el proceso de simulación. En él se puede observar en colores más claros la cantidad de *bloom* real que hay en una masa de agua (que se obtiene de los valores almacenados en una base de datos relacionada con una simulación realizada con el software EEMS) en contraposición con el color oscuro del agua. Sobre esta información, se muestra con un círculo rojo la cantidad y el tamaño del HAB predicho por el modelo explicado en la sección anterior. Finalmente puede observarse también, representado con un punto negro sobre el mapa, la posición del USV en cada momento. Por lo tanto, esta gráfica nos ayuda a hacernos una idea visual de lo que está teniendo lugar en la simulación, y nos permite observar si la predicción es correcta, y si la trayectoria del USV es la adecuada.

La gráfica anterior se actualiza en cada *frame* de tiempo, de forma que el operador o el experto pueda visualizar de forma simultánea el mapa de color que representa el *bloom* real, el círculo del *bloom* predicho y la localización del barco. El código principal utilizado para realizar esta representación gráfica es el siguiente:

```
# GRÁFICA 1
# -----
# Mapa de color del bloom real
norm = plt.Normalize(0, 10)
alg = np.array(list(map(norm, algae[frame][1][lyr][:])))
colors = cmap(alg)

# Posición del bloom predicho
circle = Circle((v_bloom_lon[frame], v_bloom_lat[frame]), radius = 0.001 *
    math.sqrt(v_bloom_size[frame]), fill = False, color = "red")
ax.add_patch(circle)

# Posición del USV
ax.plot(v_usv_lon[frame], v_usv_lat[frame], color = "black", marker = 'o',
    markersize = 4)
```

Un ejemplo de un *frame* del gráfico resultante se muestra en la figura 4.1, mientras que un ejemplo de una animación completa puede verse en el vídeo [81]. En dicha figura puede observarse el lago completo, en el que las concentraciones más elevadas de algas se encuentran en las zonas con colores amarillos, situadas en los extremos del lago. También puede verse el punto negro que representa a la embarcación autónoma y el círculo rojo que representa el *bloom* predicho.

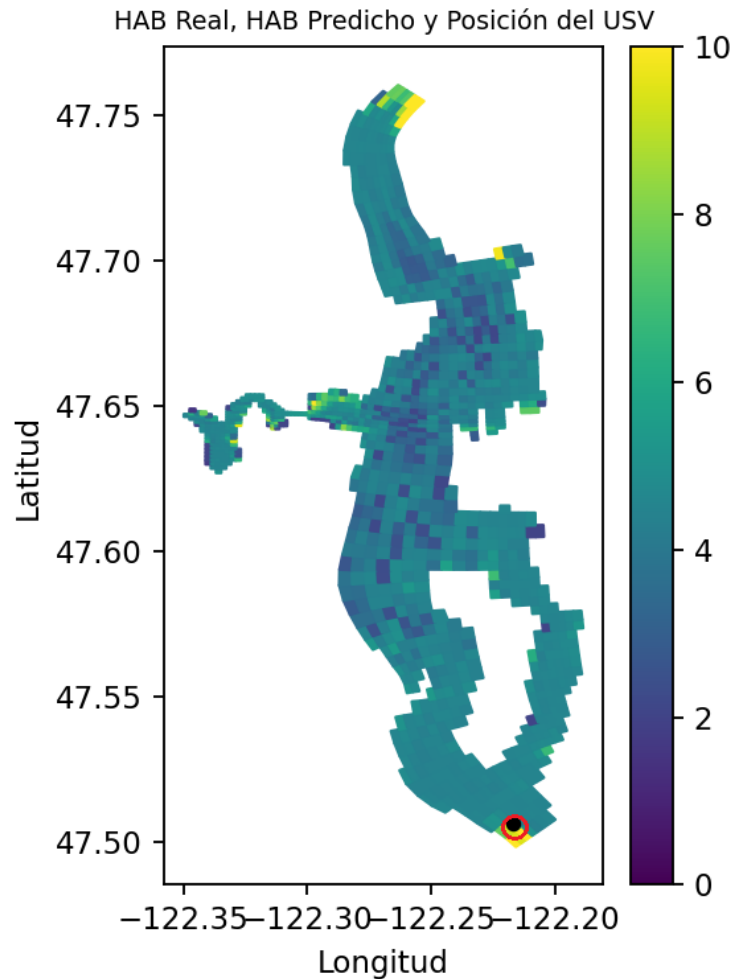


Figura 4.1: Gráfico 1 del informe de la capa *Fog*

El segundo de los gráficos del informe es un mapa animado similar al anterior que muestra en colores más claros la cantidad de oxígeno disuelto en el agua con datos sacados de EEMS. La cantidad de oxígeno disuelta en las diferentes zonas de la masa de agua aporta información muy importante para la predicción, pues las cianobacterias liberan oxígeno al hacer la fotosíntesis, por lo que las zonas con mayor oxígeno del mapa son zonas donde es más probable que haya un *bloom*. El código central del mapa animado es el mostrado a continuación:

```
# GRÁFICA 2
# -----
norm = plt.Normalize(0, 25)
disox = np.array(list(map(norm, dissolved_oxygen[frame][lyr][:])))
colors = cmap(disox)
```

Un ejemplo de un *frame* del gráfico final se muestra en la figura 4.2, mientras que la animación completa puede verse en el vídeo [81] junto al resto de animaciones del informe.

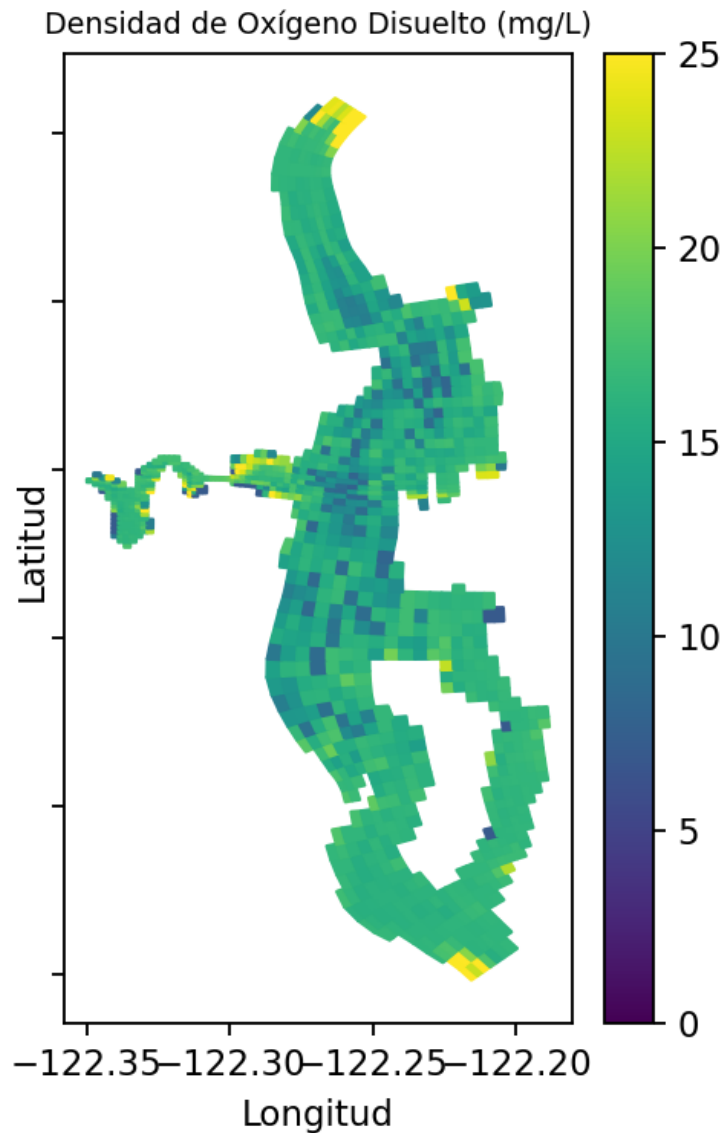


Figura 4.2: Gráfico 2 del informe de la capa *Fog*

El tercer gráfico del informe de la capa *Fog* es un mapa dinámico similar a los anteriores que muestra la cantidad de nitrógeno disuelto en el agua con datos sacados de EEMS. Este es un dato fundamental para realizar la predicción, pues los nitratos son el alimento principal del *bloom*, por lo que es más probable encontrarlos donde la cantidad de este nutriente es más elevada. El código principal de este gráfico se muestra a continuación:

```
# GRÁFICA 3
# -----
norm = plt.Normalize(0, 0.2)
nit = np.array(list(map(norm, nitrogen[frame][lyr][:])))
colors = cmap(nit)
```

La figura 4.3 muestra un ejemplo de un *frame* de este tipo de gráfico, aunque, como en los dos casos anteriores, la animación completa puede verse en el vídeo [81].

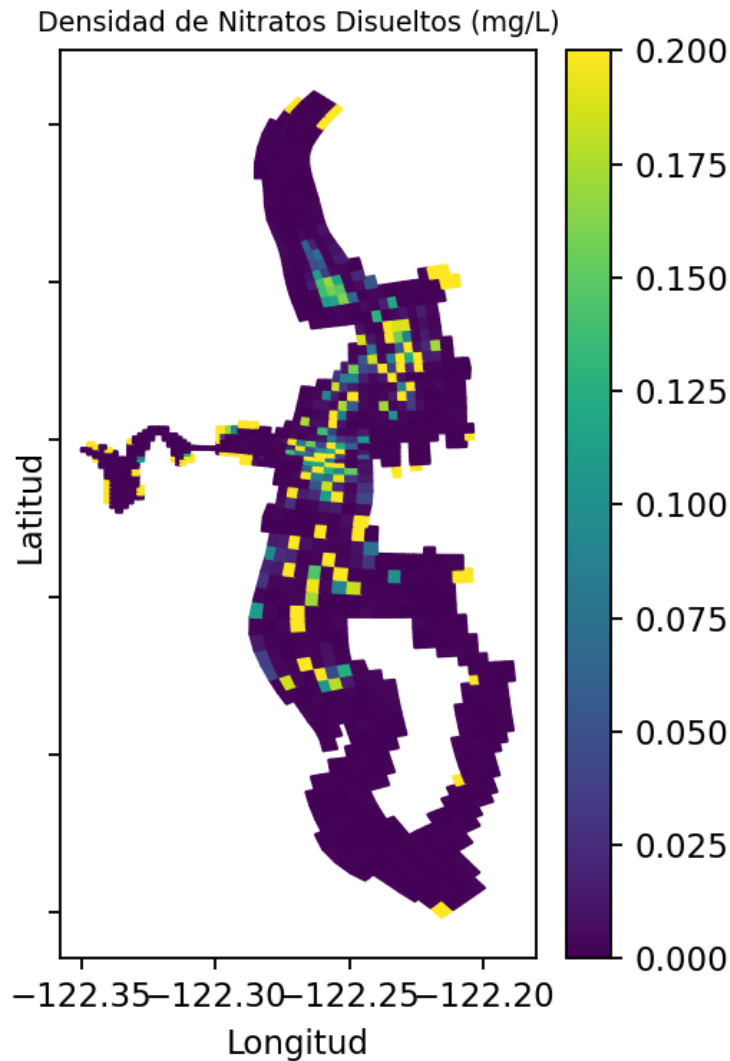


Figura 4.3: Gráfico 3 del informe de la capa *Fog*

La cuarta gráfica del informe representa la temperatura del agua en la posición en la que se encuentra el USV en cada instante de tiempo, la radiación solar en la misma posición y la batería del USV, que se recarga con la energía solar. Esta gráfica nos ayuda a visualizar la correlación entre estas tres variables, pues la temperatura del agua y la batería del barco dependen de la irradiancia del sol. El código utilizado para generar esta gráfica se muestra a continuación:

```
# GRÁFICA 4
# -----
# Radiación solar
ax.plot(hours, sun, color = "blue", label = "Radiación Solar (A/h)")
ax.plot(hours[frame], sun[frame], marker = 'o', color = "blue")

# Temperatura del agua
ax2.plot(hours[initFrame : frame + 1], water_temp[lyr][myip][initFrame : frame
+ 1], color = "red", label = "Temperatura del Agua (°C)")
ax2.plot(hours[frame], water_temp[lyr][myip][frame], color = "red", marker =
"o", markersize = 4)
```

Batería eléctrica del USV

```
ax.plot(hours[initFrame : frame + 1], v_usv_power[initFrame : frame + 1],
        color = "black", label = "Batería Eléctrica del USV (A/h) [0, 1]")
ax.plot(hours[frame], v_usv_power[frame], color = "black", marker = 'o',
        markersize = 4)
```

La figura 4.4 muestra la representación descrita. La radiación solar se muestra en azul. La temperatura del agua se muestra en rojo y la batería del USV en negro. Como las anteriores, la animación completa puede verse en el vídeo [81]. De esta representación cabe destacar que las curvas de la radiación solar se pintan todas a la vez y luego se va moviendo dinámicamente un punto azul que indica el nivel de radiación en ese momento. Se ha representado de esta manera para tener una referencia fija que ayude a la visualización.

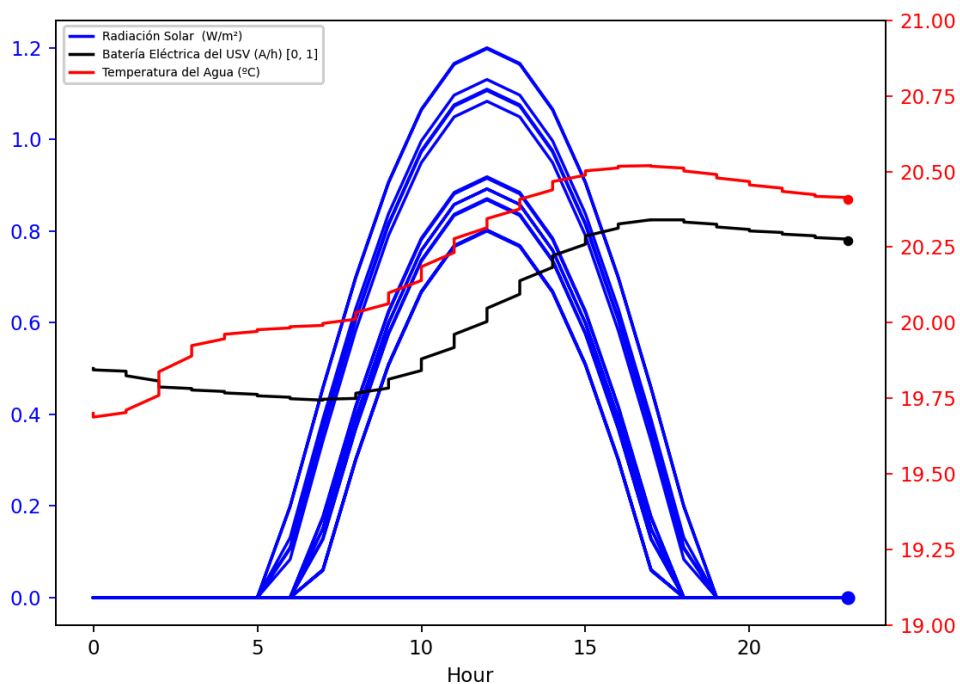


Figura 4.4: Gráfico 4 del informe de la capa *Fog*

Las gráficas 5 y 6 del informe muestran dos mapas animados con las representaciones respectivas de la velocidad del agua y del viento en forma de vectores que varían su módulo y dirección. Actualmente en el modelo de predicción solo se utiliza la velocidad del agua, pero la del viento es un parámetro a tener en cuenta y que en un futuro se podría usar también para las predicciones del estado de las cianobacterias, por lo que hemos incluido en los informes una gráfica dinámica que lo representa. Del código de estas dos gráficas que se incluye a continuación cabe destacar que los vectores se consiguen representar con la función *quiver*.

GRÁFICA 5

Velocidad del agua

```
ax.quiver(zonal_lon, zonal_lat, water_speed_x[frame][lyr][:] / 10,
         water_speed_y[frame][lyr][:] / 10, color = 'b')
```

GRÁFICA 6

Velocidad del viento

```
ax.quiver(zonal_lon, zonal_lat, wind_speed_x[frame][:] / 10,
         wind_speed_y[frame][:] / 10, color = 'b')
```

Un ejemplo de estas gráficas se muestra en la figura 4.5, en las que puede observarse que la velocidad del viento es más constante en toda la masa de agua que la velocidad de las corrientes, que es más intensa en algunas zonas.

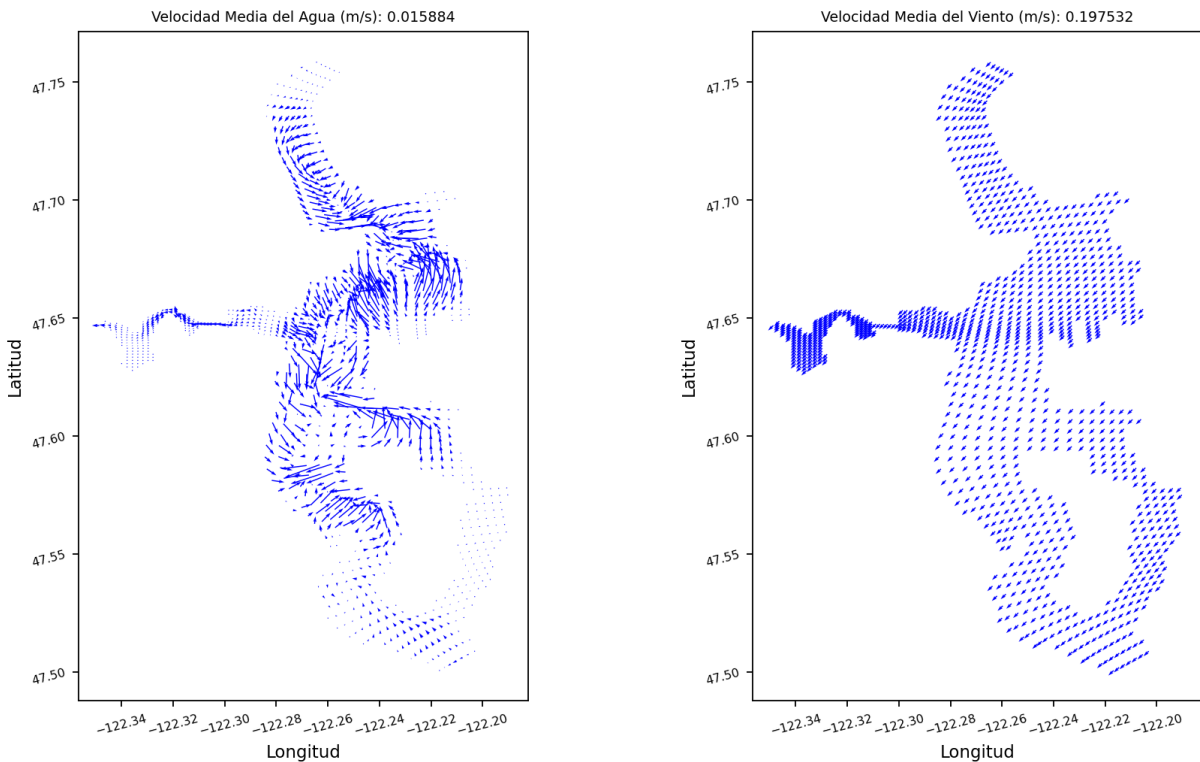


Figura 4.5: Gráficos 5 y 6 del informe de la capa *Fog*

Mediciones de los sensores

De la séptima gráfica en adelante nos encontramos con representaciones en las que se muestra en el eje de ordenadas el valor de una variable determinada en cada instante de tiempo de la simulación, que es representado en el eje de abscisas. Esto permite visualizar la evolución temporal de diferentes variables de la simulación.

La gráfica número 7 muestra la radiación solar registrada durante el tiempo que dura la simulación. La radiación solar es un parámetro importante a la hora de predecir el *bloom*, pues influye en la realización de fotosíntesis de las cianobacterias. Por ello, a mayor luminosidad solar, mayor probabilidad de encontrar HABs. El código de dicha gráfica se presenta a continuación:

GRÁFICA 7

```
ax.plot(date[ini_frame : end_frame], sun[ini_frame : end_frame], color =  
"blue")
```

Un ejemplo de esta representación se muestra en la figura 4.6. En ella se observa que hay días con más intensidad lumínica y otros días que la meteorología es menos favorable a la formación de HABS al haber menos radiación solar.

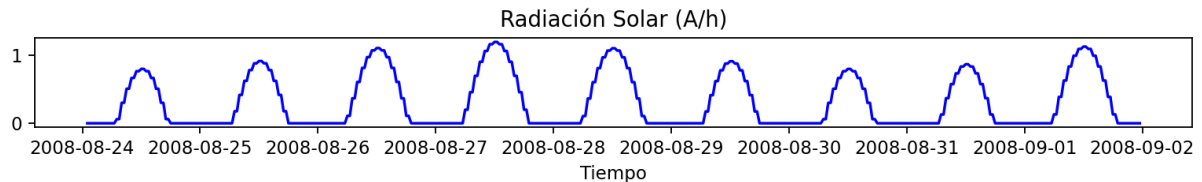


Figura 4.6: Gráfico 7 del informe de la capa *Fog*

La gráfica número 8 es semejante a la anterior, pero muestra la temperatura del agua en lugar de la luminosidad. La temperatura del agua favorece también la aparición de *blooms*, por lo que es importante su medición y su representación para el estudio de su comportamiento. A continuación se incluye el código correspondiente.

```
# GRÁFICA 8  
# -----  
ax.plot(date[ini_frame : end_frame], v_water_temp[ini_frame : end_frame],  
color = "blue")
```

La figura 4.7 muestra un ejemplo de la gráfica de la temperatura del agua en función de los días que transcurren en la simulación. Pueden observarse en él fluctuaciones claras entre las horas de sol y las de oscuridad, pues la radiación, también influye en este parámetro, tal y como se veía claramente en la gráfica 4.4.

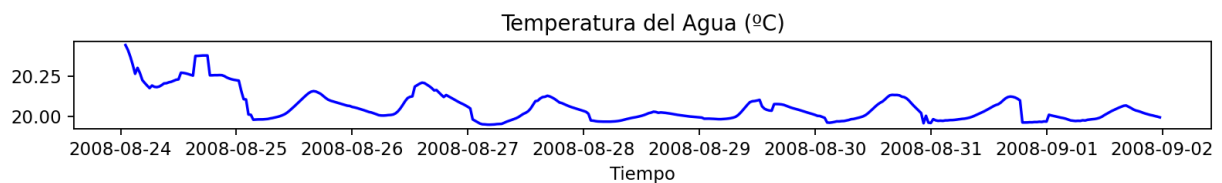


Figura 4.7: Gráfico 8 del informe de la capa *Fog*

La novena gráfica del informe muestra la cantidad de nitratos medida por el sensor a lo largo de la simulación. Como hemos comentado anteriormente, los nitratos representan el alimento principal de las cianobacterias y es interesante estudiar las medidas obtenidas por el sensor de la embarcación a lo largo de la simulación. El código de esta representación es el siguiente:

```
# GRÁFICA 9  
# -----
```

```
ax.plot(date[ini_frame : end_frame], v_nitrate[ini_frame : end_frame], color =  
"blue")
```

Un ejemplo de gráfica de la cantidad de nitratos medida es el representado en la figura 4.8.

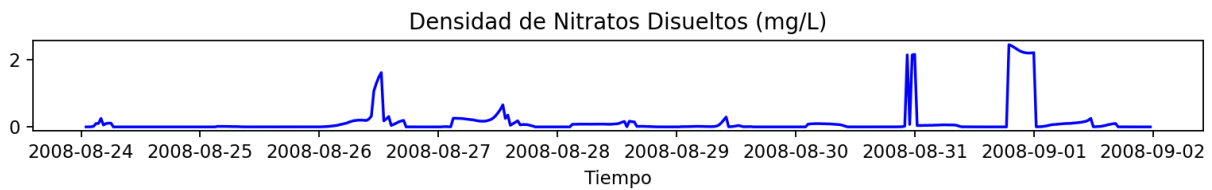


Figura 4.8: Gráfico 9 del informe de la capa *Fog*

La gráfica 10 es semejante a la anterior, pero muestra las mediciones de oxígeno disuelto en vez de las de nitratos. Como se ha comentado con anterioridad, el oxígeno puede ser signo de que hay HABs por la zona. El código es el siguiente:

```
# GRÁFICA 10  
# -----  
ax.plot(date[ini_frame : end_frame], v_oxygen[ini_frame : end_frame], color =  
"blue")
```

La figura 4.9 muestra cómo es la representación de las mediciones de oxígeno disuelto en la masa de agua.

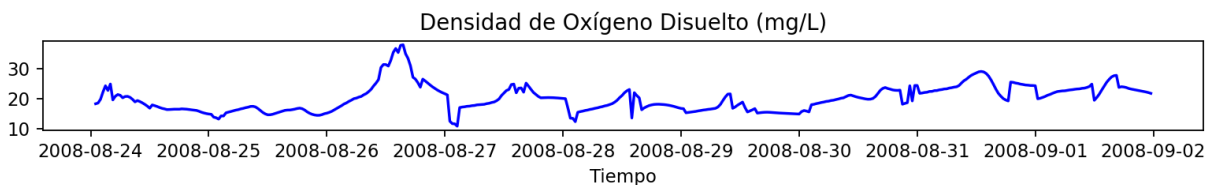


Figura 4.9: Gráfico 10 del informe de la capa *Fog*

En el gráfico número 11 se representan las mediciones del módulo de la velocidad del agua a lo largo de la simulación. El código puede verse a continuación:

```
# GRÁFICA 11  
# -----  
ax.plot(date[ini_frame : end_frame], v_water_speed[ini_frame : end_frame],  
color = "blue")
```

Un ejemplo de esta representación puede observarse en la figura 4.10.



Figura 4.10: Gráfico 11 del informe de la capa *Fog*

Como último gráfico de medición de los sensores embarcados, la gráfica 12 representa el módulo de la velocidad del viento que, como hemos comentado con anterioridad, puede ser útil para predicciones futuras. El código es el que sigue:

```
# GRÁFICA 12
# -----
ax.plot(date[ini_frame : end_frame], v_wind_speed[ini_frame : end_frame],
        color = "blue")
```

En la figura 4.11 se observa un ejemplo de dicha representación.

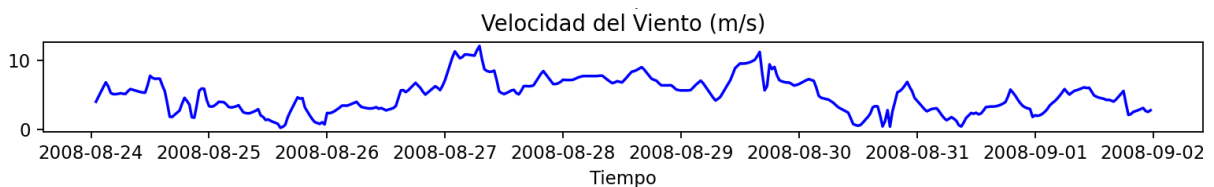


Figura 4.11: Gráfico 12 del informe de la capa *Fog*

Información del *bloom* predicho

En cuanto al *bloom* predicho, hay una serie de valores que interesa representar. Uno de ellos es si hay o no una afloración relevante a lo largo del tiempo. Por eso la gráfica 12 muestra el valor de una variable *booleana* que indica si ha habido detección de *bloom* a lo largo de la simulación. Es interesante de cara al análisis de datos el número de veces que se detecta una afloración de algas nocivas. El código de esta gráfica se muestra a continuación:

```
# GRÁFICA 13
# -----
ax.plot(date[ini_frame : end_frame], v_bloom_detection[ini_frame : end_frame],
        color = "blue")
```

La figura 4.11 muestra las diferentes detecciones de *bloom* a lo largo del tiempo de simulación.

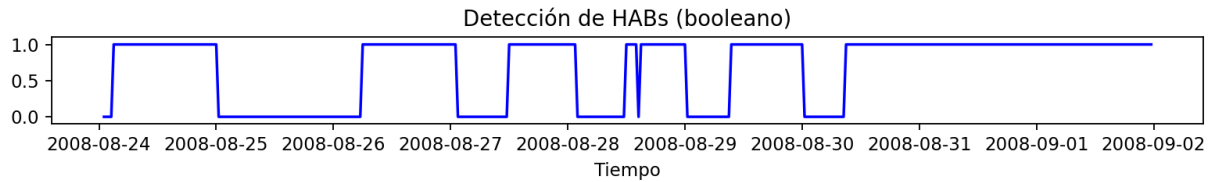


Figura 4.12: Gráfico 13 del informe de la capa *Fog*

Otro parámetro que nos interesa medir es la densidad del *bloom* predicho. Esto se representa en la gráfica 13 del informe, para la que se ha programado el código siguiente:

```
# GRÁFICA 14
# -----
ax.plot(date[ini_frame : end_frame], v_bloom_size[ini_frame : end_frame],
        label = "HAB Predicho", color = "blue")
```

La figura 4.13 muestra un ejemplo de esta gráfica. Esta medida, comparada con la concentración real de *bloom* resulta interesante porque nos ayuda a hacernos una idea visual del error de predicción, error que será posteriormente calculado y utilizado para re-entrenar el modelo de predicción, de lo cual hablaremos en la siguiente sección 4.3.

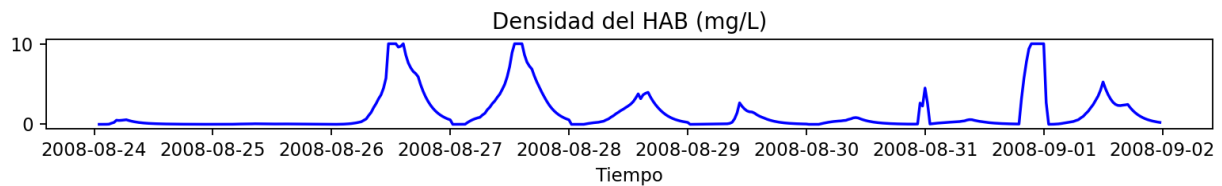


Figura 4.13: Gráfico 14 del informe de la capa *Fog*

Las dos siguientes gráficas del informe de la capa *Fog* muestran la variación en la longitud y latitud (respectivamente) del *bloom* predicho. Esto nos sirve para hacernos una idea del desplazamiento tanto en el eje *x* como en el eje *y* del HAB predicho. El código utilizado para obtenerlas se muestra a continuación:

```
# GRÁFICA 15
# -----
ax.plot(date[ini_frame : end_frame], v_bloom_lon[ini_frame : end_frame], color
        = "blue")

# GRÁFICA 16
# -----
ax.plot(date[ini_frame : end_frame], v_bloom_lat[ini_frame : end_frame], color
        = "blue")
```

En la figura 4.14 se ve representada esta información.

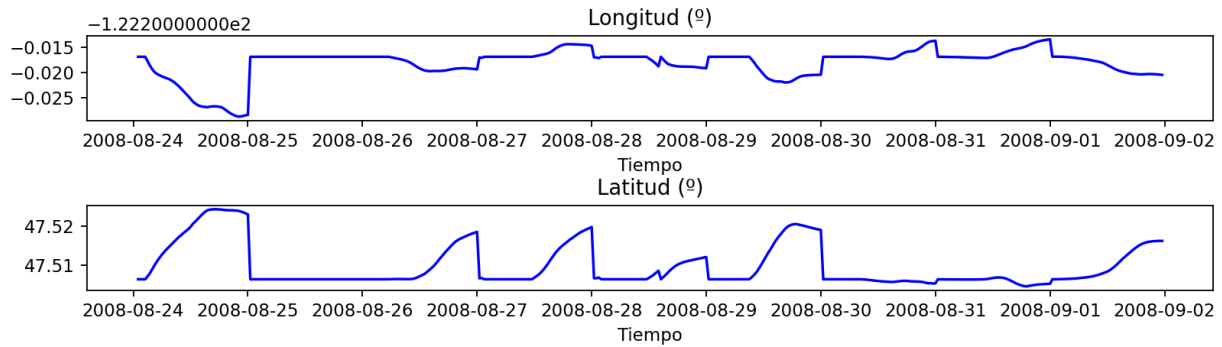


Figura 4.14: Gráficos 15 y 16 del informe de la capa *Fog*

Mediciones de las variables del USV

Las últimas gráficas del informe hacen referencia a los datos de la embarcación. La primera de las mismas, que es la gráfica número 17 del informe, representa el nivel de batería del USV a lo largo del tiempo de simulación en relación con los niveles de radiación solar. Esto nos permite apreciar, una vez más, la relación entre estos dos parámetros. El código se presenta a continuación:

```
# GRÁFICA 17
# -----
ax.plot(date[ini_frame : end_frame], v_usv_power[ini_frame : end_frame], label
        = "Batería del USV", color = "blue")
ax.plot(date[ini_frame : end_frame], sun[ini_frame : end_frame], label =
        "Sun", color = "Radiación Solar")
```

La figura 4.15 muestra un ejemplo de la representación gráfica descrita.

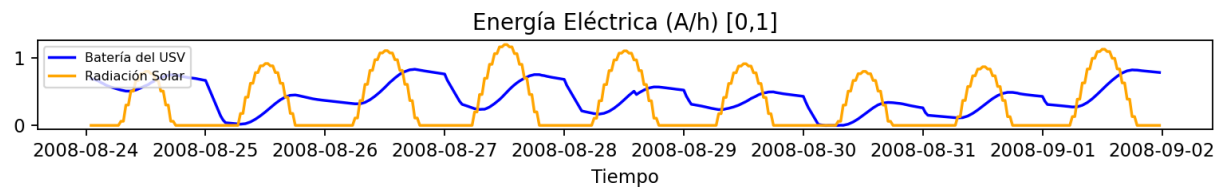


Figura 4.15: Gráfico 17 del informe de la capa *Fog*

La segunda gráfica del USV plasma el módulo de la velocidad del mismo a lo largo de las misiones que realiza durante la simulación. Esta velocidad se calcula a partir de la distancia que recorre y del tiempo que tarda en recorrerla. El código, una vez más, se muestra a continuación:

```
# GRÁFICA 18
# -----
ax.plot(date[ini_frame : end_frame], speed[ini_frame : end_frame], color =
        "blue")
```

En la figura 4.16 se muestra un ejemplo de la representación de la velocidad del USV.

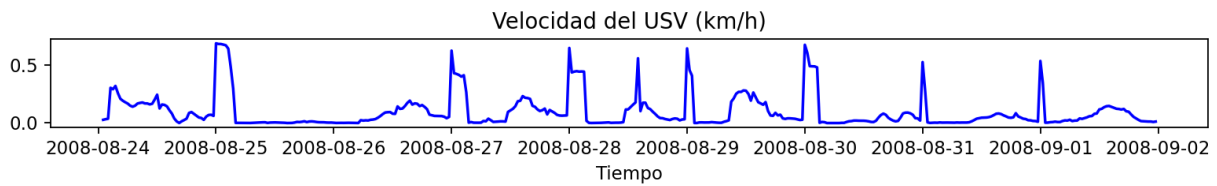


Figura 4.16: Gráfico 18 del informe de la capa *Fog*

Por último, se muestran en el informe dos gráficas análogas a las de la figura 4.14, pero representando esta vez la variación en la longitud y latitud de la posición del USV. El código implementado para obtenerlas es el siguiente:

```
# GRÁFICA 19
# -----
ax.plot(date[ini_frame : end_frame], v_usv_lon[ini_frame : end_frame], color =
        "blue")

# GRÁFICA 20
# -----
ax.plot(date[ini_frame : end_frame], v_usv_lat[ini_frame : end_frame], color =
        "blue")
```

En la figura 4.17 puede verse un ejemplo de su representación.

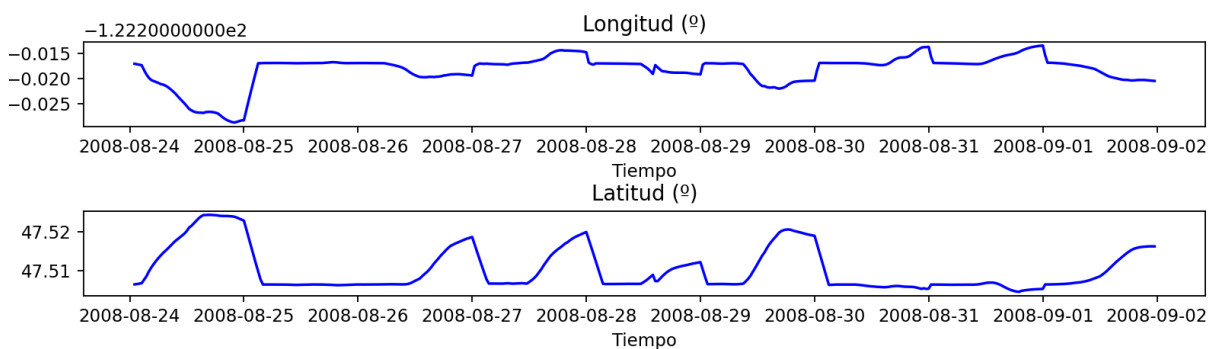


Figura 4.17: Gráficos 19 y 20 del informe de la capa *Fog*

Evolución de la función de error

La última sección del informe está dedicada a representar el error cometido por el modelo de predicción respecto a la situación real del *bloom* de cara a poder discernir si es necesario re-entrenar el modelo. Por este motivo, la gráfica número 21 muestra enfrentadas la densidad real del *bloom* (obtenida de EEMS) en la localización de la embarcación y el valor predicho por nuestro modelo en la misma localización. El código utilizado para generarla es el siguiente:

```
# GRÁFICA 21
```

```
# -----
ax.plot(date[ini_frame:end_frame], v_bloom_size[ini_frame:end_frame], label =
        "HAB Predicho")
ax.plot(date[ini_frame:end_frame], v_algae[ini_frame:end_frame], label = "HAB
        Real", color = "purple")
```

En la figura 4.18 puede verse un ejemplo de esta representación en la que se nota claramente la discrepancia entre los valores de ambas curvas a lo largo del tiempo.

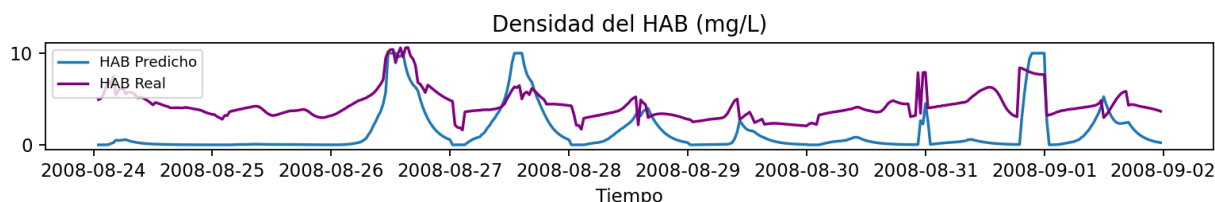


Figura 4.18: Gráfico 21 del informe de la capa *Fog*

Por último, en la gráfica número 22 se representa en color rojo y con un trazo continuo el error calculado como la diferencia de las dos curvas mostradas en la gráfico anterior (representado en la figura 4.18) y en color azul y un trazo discontinuo la media aritmética de dicho error. El código utilizado para representar ambas curvas es el siguiente:

```
# GRÁFICA 22
# -----
ax.plot(date[ini_frame:end_frame], error[ini_frame:end_frame], color = "red",
        label = "Error")
ax.plot(date[ini_frame:end_frame], error_mean[ini_frame:end_frame], '--',
        label = f"Media aritmética del error = {'%.2f'%error_mean_value}")
```

Y finalmente, un ejemplo de este tipo de gráfica puede ser observado en la figura 4.19.

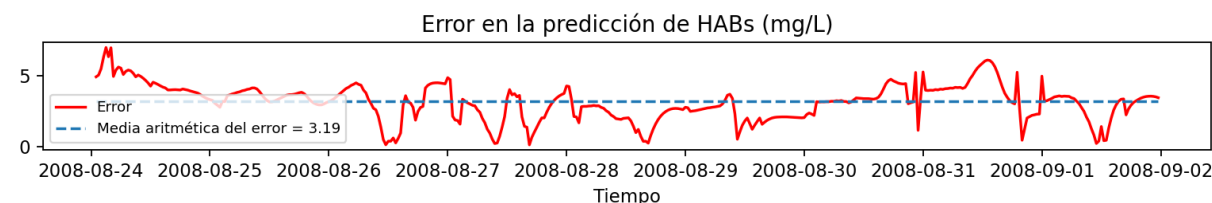


Figura 4.19: Gráfico 22 del informe de la capa *Fog*

4.2.2. Informes de la capa *Cloud*

Los informes de la capa *Cloud* se encuentran en fase preliminar, por lo que son más sencillos que los de la capa *Fog*. Estos consisten en una tabla en la que se muestran los resultados de algunos cálculos que se realizan sobre la información almacenada en la base de datos global y un mapa de calor que muestra los lugares donde se han encontrado HABs

tras haber realizado una simulación. Dichos lugares serán denominados *calientes* porque existe una probabilidad significativamente alta de que las formaciones de cianobacterias vuelvan a aparecer en ellos.

Más concretamente, la primera parte del informe de la capa *Cloud* consiste en una tabla que recoge los resultados obtenidos al realizar los siguientes cálculos sobre diferentes variables de la base de datos:

- Recuento total, que es equivalente al cardinal del número de elementos del conjunto. El concepto matemático es el siguiente:

$$|X| = n \leftrightarrow \exists f : \mathbb{N}_n \mapsto X$$

donde f es una función biyectiva y \mathbb{N}_n es el conjunto de los n primeros números naturales.

- Media aritmética, que se calcula, tal y como se muestra a continuación, como la suma de todos los valores pertenecientes a un conjunto, dividida entre el número total de sumandos.

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

donde μ es la media aritmética, x_i son los datos de la muestra y N es el número de datos en la muestra.

- Desviación típica, que representa la variación de los datos con respecto a la media, y que se calcula de la siguiente forma:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

donde σ es la desviación típica, x_i son los datos de la muestra, μ es la media aritmética y N es el número de datos en la muestra.

- Valor mínimo, es decir, el valor más pequeño del conjunto:

$$\text{mín}(X) = a \leftrightarrow a \leq x_i \quad \forall x_i \in X$$

- Percentil 25, que indica el valor que deja a la izquierda de la distribución el 25 % de los datos de un conjunto por ser inferiores a él. Su fórmula es:

$$P_{25} = \left(\frac{N + 1}{100} \right) \times 25$$

donde N es el número total de datos del conjunto.

- Percentil 50, que indica el valor que deja a la izquierda de la distribución el 50 % de los datos de un conjunto por ser inferiores a él. Su fórmula es:

$$P_{50} = \left(\frac{N + 1}{100} \right) \times 50$$

donde N es el número total de datos del conjunto.

- Percentil 75, que indica el valor que deja a la izquierda de la distribución el 75 % de los datos de un conjunto por ser inferiores a él. Su fórmula es:

$$P75 = \left(\frac{N + 1}{100} \right) \times 75$$

donde N es el número total de datos del conjunto.

- Valor máximo, es decir, el valor más grande del conjunto:

$$\text{máx}(X) = a \leftrightarrow a \geq x_i \quad \forall x_i \in X$$

Los datos sobre los que se aplican estos cálculos actualmente son el nivel de la batería eléctrica del barco, la longitud y latitud sobre los que se ha situado la embarcación a lo largo de la misión, el error de la posición de la embarcación (también por separado, respectivamente, longitud y latitud) respecto a la localización de la predicción, el valor de la radiación solar, la velocidad del agua (por separado, respectivamente, sobre los ejes x e y), la temperatura del agua, la densidad del HAB predicho y la posición del mismo (por separado, respectivamente, su longitud y latitud).

El cálculo de estos valores es de gran ayuda para analizar de forma global los datos de una simulación, y para conocer el error cometido en la predicción y analizar la necesidad de re-entrenar el modelo predictivo desarrollado para estimar el estado del *bloom* de cianobacterias.

Visualmente, la tabla del informe es similar a la representada en el cuadro 5.1 que observamos a continuación, aunque es probable que el número y contenido de las columnas varíe en función de las necesidades de la simulación llevada a cabo.

	Batería Eléctrica del USV (A/h)	Radiación Solar (A/h)	Temperatura del Agua (°C)	Densidad del HAB (mg/L)
Total	720.000	720.000	720.000	720.000
Media Aritmética	0.919	0.013	20.169	0.752
Desviación Típica	0.102	0.015	0.370	1.720
Mínimo	0.445	0.000	19.357	0.009
Percentil 25	0.863	0.000	19.905	0.069
Percentil 50	0.952	0.000	20.262	0.327
Percentil 75	1.000	0.028	20.516	0.599
Máximo	1.000	0.040	20.585	10.000

Tabla 4.1: Tabla del informe de la capa *Cloud*

Para poder realizar estos cálculos es necesario que durante la simulación todos los datos de la misma se guardan en un fichero tipo *.csv*, que es leído antes de generar los datos del informe *Cloud* da cada simulación. Para ello utilizamos la función *describe()* de la librería *pandas*. El código con el que se genera la tabla anterior puede verse a continuación:

```
df = pd.read_csv(base_folder + "/FogServer.InferenceService.csv")
report1 = df.describe()
```

Por último y tal como ya hemos explicado anteriormente, el informe de *Cloud* recoge un mapa de calor que muestra las zonas *calientes*. Para representarlo se ha utilizado la función *density_mapbox* de la librería *plotly* de Python. Esta es una librería que permite crear gráficos interactivos como diagramas de líneas, diagramas de dispersión, gráficos de área, gráficos de barras, etc.

El código utilizado para generar dicho mapa en formato *html* es el siguiente:

```
fig = px.density_mapbox(report3, lat='latitude', zoom=12, lon='longitude',  
                        z='density', mapbox_style="stamen-terrain")  
fig.write_html(self.base_folder + "/cloud-report3.html")
```

Y un ejemplo de este mapa de calor se muestra en las figuras 4.20 y 4.21, utilizando colores más claros para las zonas donde es más probable localizar los *blooms* de cianobacterias.

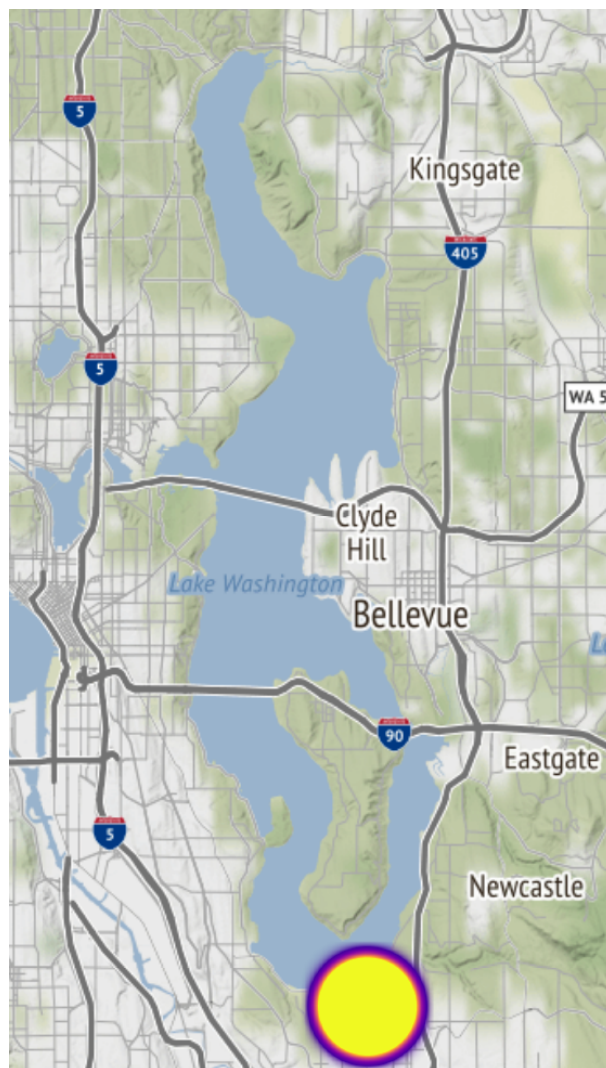


Figura 4.20: Mapa de calor generado por el informe de *Cloud*

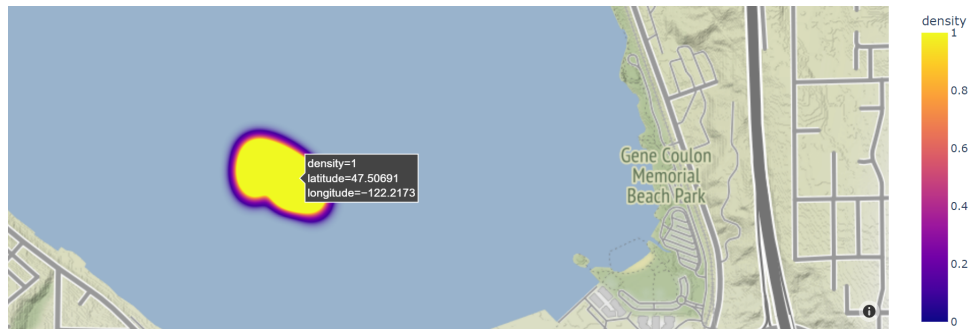


Figura 4.21: Zoom sobre el mapa de calor del informe

4.3. Servicio de entrenamiento

El modelo del servicio de entrenamiento se ubica en la capa *Cloud*, y su función es re-entrenar el modelo de predicción del *bloom* ubicado en la capa *Fog* para poder mejorar el resultados de las nuevas predicciones. En concreto, lo que queremos es empezar entrenando la ecuación (4.7) propuesta al principio del capítulo, para ajustar las constantes que forman parte de ella de manera que los resultados de la predicción se ajusten a la situación real del HAB. Esto se consigue minimizando el error, es decir, la distancia entre nuestra predicción y los datos reales del HAB obtenidos de EEMS estimada del *bloom*, es decir, en la posición calculada por las ecuaciones (4.2) y (4.3).

Repasando la sección 4.1, podemos recordar que la ecuación 4.7 modela la evolución de densidad de HAB en función de 3 constantes, k_1 , k_2 y k_3 definidas en un primer momento de forma manual a los siguientes valores por defecto:

$k_1 = 1$
 $k_2 = 1$
 $k_3 = 1$

Lo que buscamos en el servicio de entrenamiento es, por tanto, reajustar estas constantes de forma que se minimice el error en el estado del *bloom* cuando este aumente sustancialmente. Para ello se utilizó inicialmente el método del descenso de gradiente [82, 83], que es un algoritmo utilizado para encontrar los valores de los parámetros de una función que minimizan una función de coste. Este método es ampliamente utilizado en el ámbito del aprendizaje automático.

De forma general, el proceso del gradiente descendiente consiste en inicializar los parámetros de la función con valores aleatorios, calcular el gradiente (o derivada parcial) de la función de coste con respecto a cada parámetro, y actualizar los valores de los parámetros en la dirección del gradiente si se busca maximizar la función objetivo, o en dirección opuesta al gradiente si lo que se busca es minimizar dicha función. Además, el ajuste de los valores de los parámetros se hace de forma incremental utilizando un factor adicional constante. Este proceso se repite varias veces hasta que la función de coste (que es el error de predicción en este caso) alcance un valor mínimo o máximo (en función de lo que se busque) o se hayan llevado a cabo un número específico de iteraciones.

Formalmente, si queremos minimizar la función objetivo, empezando en un punto aleatorio x_0 , y avanzando una distancia positiva α en la dirección indicada por el gradiente negativo, el nuevo valor de x_1 será de la forma:

$$x_1 = x_0 - \alpha \nabla f(x_0)$$

Generalizando la fórmula anterior para las diferentes repeticiones, se obtiene la siguiente expresión que se debe aplicar durante las iteraciones del servicio de aprendizaje:

$$x_{n+1} = x_n - \alpha \nabla f(x_n) \quad (4.13)$$

donde x_{n+1} es el valor siguiente que estamos calculando, x_n es el último valor obtenido, α es la distancia a recorrer en la dirección indicada, y $\nabla f(x_i)$ es el gradiente de la función objetivo (error en nuestro caso) en el último valor obtenido. Así, a partir de un valor aleatorio inicial se va mejorando la solución hasta encontrar un mínimo local. La figura 4.22 ilustra este proceso de forma sencilla.

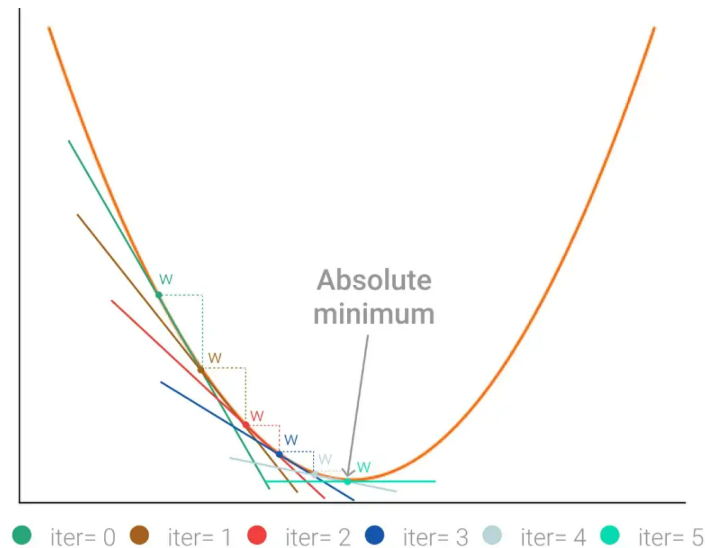


Figura 4.22: Representación de la técnica de descenso de gradiente (fuente: [83])

Para aplicar este método a nuestro problema de aprendizaje solo necesitábamos definir la función *objetivo*, también llamada *función de coste*. Para ello, hemos seleccionado la función del error cuadrático medio (MSE, del inglés, *Mean Square Error*), que compara el valor predicho con el valor conocido, calculando el error entre ellos de la siguiente forma:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

donde n es el número de datos de entrenamiento, y_i es el valor real para el i -ésimo dato de entrenamiento e \hat{y}_i es el valor predicho para el i -ésimo dato de entrenamiento.

Así, nuestra función de coste se definió como el error cuadrático medio entre la ecuación (4.5) que predice la densidad del *bloom* y los valores reales obtenidos por EEMS. La función de coste será por tanto la siguiente:

$$f(k_1, k_2, k_3) = \frac{1}{n} \sum_{i=1}^n (y_i - (k_1 \cdot photo + k_2 \cdot breath - k_3 \cdot (r_i - r_0)))^2 \quad (4.14)$$

donde n es el número de datos de entrenamiento, y_i es el valor real para el i -ésimo dato de entrenamiento, r_i es el i -ésimo valor de nutrientes predicho, h es el paso de tiempo, y *photo* y *breath* son las variables calculadas mediante la ecuación (4.6) en función de la cantidad de nitratos, el oxígeno disuelto y la radiación solar.

El código utilizado para calcular el error cuadrático medio fue el siguiente:

```
MSE = np.square(np.subtract(v_bloom_predicted, v_real_bloom)).mean()
```

Sin embargo, este método no convergía en algunos casos, por lo que se decidió utilizar la función *curve_fit* de la librería *scipy*, que utiliza el modelo de regresión por mínimos cuadrados no lineales (NLS, del inglés, *Nonlinear Least Squares*) para ajustar el modelo predictivo a los datos. Es decir, dada una aproximación numérica a la ecuación diferencial evaluada en el instante de tiempo t_i , y un valor inicial \mathbf{k}_0 de $\mathbf{k} = [k_1, k_2, k_3]$, el conjunto de parámetros a ajustar se estiman minimizando la siguiente expresión:

$$\hat{\mathbf{k}} = \min \sum_{i=1}^N [r_i - \hat{r}(t_i, \mathbf{k}, r_0)]^2 \quad (4.15)$$

donde $\hat{\mathbf{k}}$ representa los parámetros a estimar, r_i son los datos reales de densidad de floración en el instante t_i , y $\hat{r}(t_i, \mathbf{k}, r_0)$ representa la predicción para el instante t_i [84]. Este método sí converge, como podrá observarse en el caso de uso expuesto en el capítulo 5.

Por último cabe explicar que el sistema de entrenamiento no está encapsulado en un modelo atómico. En su lugar, se simula el modelo acoplado EWS replicando la misión actual, y si el error supera un umbral dado, se lanza el servicio de entrenamiento de la siguiente manera:

```
# Inicializa la simulación DEVS con el modelo EWS
Coordinator coordinator(ews_model)

# Ejecuta la simulación y obtén el error
coordinator.simulate()
error = ews_model.get_predictive_error()

# Ajustar los parámetros cuando error > umbral
if (error > threshold):
    ews_model.adjust_predictive_model()
```

Capítulo 5

Caso de uso

Esta sección presenta un escenario de simulación que pone a prueba la arquitectura e implementación del *software* desarrollado. En ella se muestran todos los resultados de la simulación de forma gráfica y visual para que el lector pueda ser partícipe del funcionamiento práctico de *DEVS-BLOOM*.

5.1. Descripción del escenario

El cuerpo de agua seleccionado para la simulación es el lago Washington, situado en el estado de Washington de EEUU y cuya localización y forma superficial puede observarse en la figura 5.1. Más concretamente, nos centraremos en el área sur de dicho lago, delimitada en la misma figura por un rectángulo rojo.

En este caso, todo el conjunto de datos de la masa de agua es sintético y se genera con la herramienta EEMS [17], que incorpora un modelo preciso del lago y que permite la simulación de algas configurables con la mayoría de las propiedades de las cianobacterias.

De esta forma, cuando sea necesario cada uno de los sensores embarcados en el USV leerá el dato correspondiente del fichero generado por EEMS y lo transmitirá a la capa *Fog*. Estas lecturas de datos y sus correspondientes transmisiones a la capa *Fog* tienen lugar cada 30 minutos del día simulado. Además, el USV enviará también a la capa *Fog* las coordenadas de su posición y el nivel restante de su batería. Esto nos permitirá monitorizar el estado de la afloración de cianobacterias y alimentar el modelo de predicción.

5.2. Simulación

La simulación utilizada tiene lugar entre el día 23 de agosto de 2008 a las 00 : 00 horas y el día 7 de septiembre de 2008 a la misma hora en la zona sur del lago de Washington, en Estados Unidos. Estos son datos antiguos porque se corresponden a una simulación proporcionada como ejemplo con la herramienta EEMS. En total, se ha elegido una simulación de 18 días, cuyas características serán también pautadas por las instrucciones almacenadas en el fichero que gestiona el modelo atómico *Generator*, cuya parte principal podemos observar a continuación:

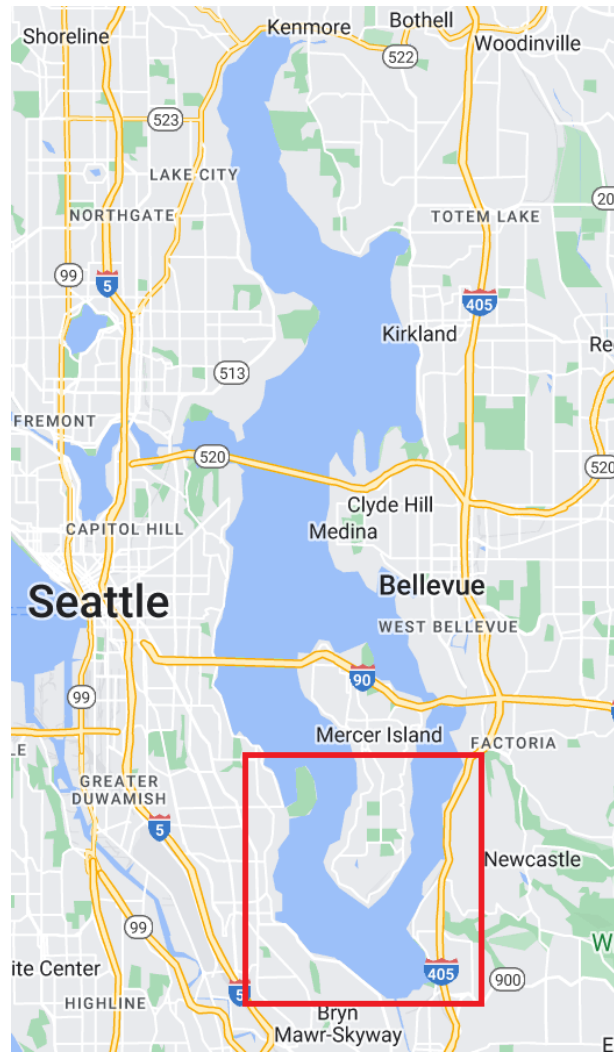


Figura 5.1: Lago de Washington en Washington, Estados Unidos

```
DATETIME;COMMAND;ARGUMENTS
2008-08-23 00:00:00;CMD_START_SIM;
2008-09-07 00:00:00;CMD_SAVE_DATA;
2008-09-07 00:00:00;CMD_FOG_REPORT;
2008-09-07 00:00:00;CMD_STOP_SIM;
2008-09-07 00:00:00;CMD_CLOUD_REPORT;
```

Así, el 23 de agosto de 2008 comienza la simulación con la instrucción `CMD_START_SIM`, que reciben todos los modelos conectados al Generador. Durante la simulación, cada 30 minutos virtuales los sensores realizan mediciones y envían los datos al GCS. Este, a su vez, envía la información al servicio de predicción detallado en la sección 4.1, que dispone del modelo predictivo basado en ecuaciones en diferencias descritas en la misma sección. El servicio de predicción utilizando los datos recibidos de los sensores, es decir, la velocidad del agua, las concentraciones de oxígeno y nitratos, y la radiación solar, predice la aparición y el desplazamiento de HABs. Luego, envía el resultado de la predicción de vuelta al CGS, quien se la transmite al planificador para que en función de la posición y el tamaño del HAB predicho genere los puntos de paso obligatorio del USV, a quien

se los manda una vez calculados. Para cerrar el bucle, el USV navega hasta el punto de paso obligatorio (en la simulación lo hace virtualmente) y vuelve a tomar las medidas correspondientes en esa nueva posición, enviándoselas de nuevo a la capa *Fog* y lanzando la repetición de todo el proceso descrito en este párrafo.

Un apunte importante sobre el modelo de predicción es que inicializa la localización del *bloom* al comienzo del día en la zona sur del lago, junto a un canal que desemboca en una zona poco profunda. Estas zonas son conocidas como incubadoras por brindar las condiciones ideales para la formación de afloraciones y acumulaciones de nitratos. Por lo tanto, es muy probable que la afloración nazca en esta zona, y a continuación crezca con la radiación solar, se mueva con las corrientes de agua y se disperse por el resto del lago.

Justo antes de terminar la simulación, el *Generator* envía las instrucciones `CMD_SAVE_DATA` para que el GCS envíe toda la información a la capa *Cloud* y `CMD_FOG_REPORT` para que se genere el informe de la capa *Fog*. Finalmente, se termina la simulación con el comando `CMD_STOP_SIM`, mandando la instrucción `CMD_CLOUD_REPORT` para que el servicio de análisis de datos genere el informe de la capa *Cloud*. Con los datos de este informe, se evalúa la posibilidad de reajustar los parámetros del modelo predictivo con el servicio de entrenamiento para mejorar su precisión y adaptarlo a las peculiaridades de cada masa de agua.

5.3. Análisis de datos

El informe de la capa *Fog* muestra todos los detalles de la simulación, generando un vídeo (que puede encontrarse en [81]) en el que se incluyen 6 gráficas dinámicas que ilustran la simulación de manera muy exhaustiva y visual. La figura 5.2 es una imagen tomada de un *frame* de dicho vídeo. En ella, pueden observarse las seis gráficas, cuyo contenido explicaremos a continuación.

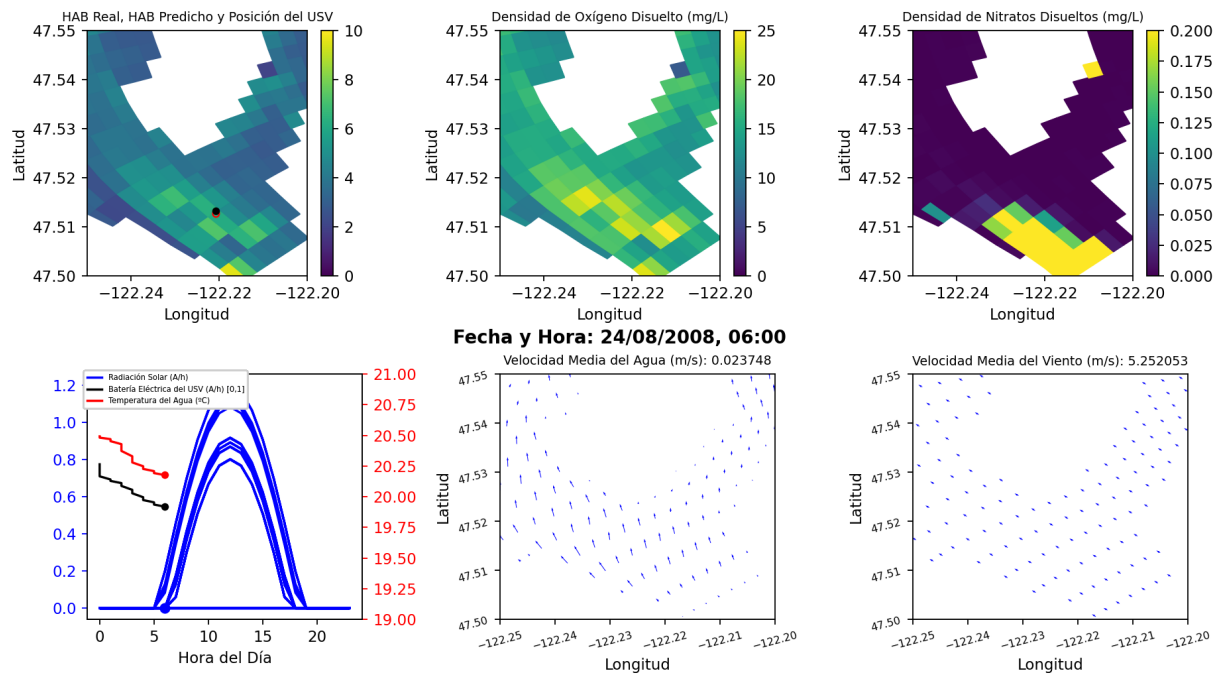


Figura 5.2: Marco de simulación del seguimiento de HABs

El gráfico superior izquierdo muestra el mapa de densidad de cianobacterias en la capa superficial, marca con un círculo rojo sobre el mapa la posición y el tamaño de la afloración predicha, e indica con un punto negro sobre la misma gráfica la posición actual del USV. El mapa que representa la densidad de HABs se ilustra con datos tomados directamente del simulador EEMS para validar que el modelo de predicción está anticipando correctamente la dinámica del *bloom*. Así, es fácil observar que el círculo rojo va moviéndose a posiciones con colores más claros, lo que indica que contienen más *bloom*, y el círculo negro va siguiendo al rojo haciendo caso a las órdenes enviadas por el planificador.

El gráfico central superior muestra el mapa completo de la densidad de oxígeno disuelto en la capa superficial. El USV toma esta medida en su posición y se la envía a la capa *Fog*, donde es utilizada por el modelo de predicción para decidir si hay afloración o no, basándose en que, como las cianobacterias en la fotosíntesis producen mucho oxígeno, donde más oxígeno haya habrá más probabilidad de que haya también un HAB.

El gráfico superior derecho muestra el mapa equivalente al anterior con la densidad de nitrato en la superficie. El modelo de predicción utiliza la medida igualmente tomada por el USV del modelo EEMS para estimar el crecimiento de la afloración, ya que los nitratos son el principal alimento del *bloom*, por lo que cuanto más nitratos hay en la masa de agua más probabilidad de que el tamaño de la afloración aumente.

El gráfico inferior izquierdo muestra tres señales en función de la hora del día: la radiación solar, representada en color azul en la gráfica y medida por el sensor de irradiancia; la temperatura del agua, representada en color rojo y medida por uno de los sensores del USV; y el nivel de batería del barco, representada en color negro en la gráfica. Con esta gráfica podemos comprobar que la batería del barco se recarga con la radiación solar, y que la temperatura es lógicamente más elevada en las horas en las que hay más radiación solar.

Esto afecta también al crecimiento del *bloom*, y es utilizado por el modelo de predicción para predecir su formación.

Los dos gráficos inferiores restantes muestran el mapa con la dirección y velocidad media del agua (centro) y del viento (derecha) en la capa superficial. El barco mide esta señal en su posición y la transmite a la capa *Fog* para que el modelo de predicción pueda estimar el desplazamiento de la afloración y en consecuencia el planificador pueda calcular el próximo punto de medida del USV.

Como se mencionó anteriormente, todos los datos utilizados en esta simulación son sintéticos. En consecuencia, todos los sensores funcionan en modo virtual. Cuando un sensor debe tomar una medida, busca en la base de datos del archivo EEMS, modifica la señal según sus características técnicas y genera un mensaje con el valor de la señal como los mostrados en la figura 3.4. La capa *Fog* recibe estas señales para realizar diferentes cálculos como la predicción del modelo y la planificación de la trayectoria del USV, y al finalizar la simulación envía la información a la capa *Cloud*.

Además, el informe de la capa *Fog* genera una serie de gráficos que representan la evolución de diferentes datos utilizados por el modelo de predicción para pronosticar la formación de HABs a lo largo de toda la simulación, que dura en total 18 días.

Así, en la figura 5.3 podemos observar 6 gráficas que muestran las diferentes medidas a lo largo del tiempo de todos los sensores de la embarcación, a excepción del de densidad de HAB, cuya representación se encuentra más adelante. Más concretamente, las gráficas de la figura 5.3 muestran la radiación solar, la temperatura del agua, la densidad de nitratos disueltos, la densidad de oxígeno disuelto, el módulo de la velocidad del agua y el módulo de la velocidad del viento, todos ellos obtenidos a lo largo de la simulación desde la localización en la que se encontraba en cada momento el USV.

En la figura 5.4 se observan 4 gráficas con información sobre el *bloom* inferido. La primera de ellas muestra el valor de un *booleano* que representa si se ha detectado afloramiento o no. En ella puede verse que se ha detectado un total de 11 veces a lo largo de la simulación. Además, se puede observar en la segunda gráfica representada la concentración de cianobacterias medida a lo largo del tiempo. Las dos últimas gráficas representan, respectivamente, la longitud y latitud de la posición del *bloom* predicho.

Estudiando los valores representados en las dos figuras que acabamos de mencionar, 5.3 y 5.4 podemos entender la correlación entre la densidad de nitratos y de oxígeno disueltos en el agua (gráficas 3 y 4 de la figura 5.3) y la detección de *bloom* (gráfica 1 de la figura 5.4). Es claro por lo observado en ellas que a mayor concentración de nitratos y oxígeno disuelto en el agua, más detección de afloraciones de bloom.

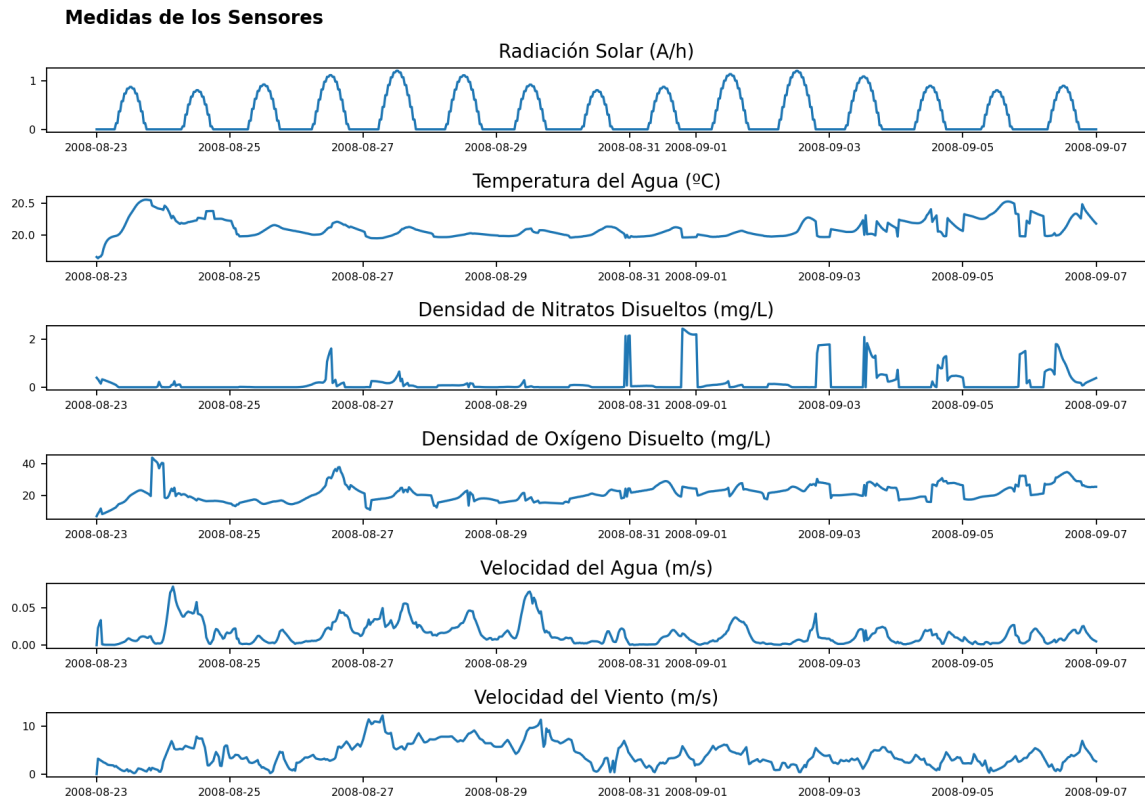


Figura 5.3: Medidas de los sensores embarcados a lo largo de la simulación

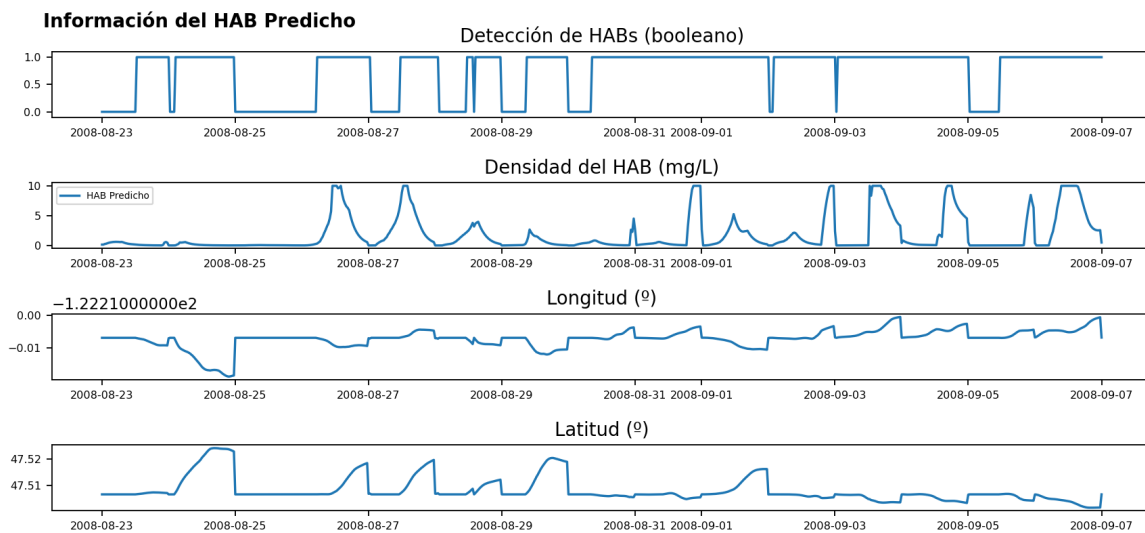


Figura 5.4: Información del *bloom* inferido durante la simulación

La figura 5.5 muestra las variables más relevantes del estado del modelo acoplado *USV*. El primer gráfico muestra el nivel de carga de la batería y la radiación solar. Aquí se puede ver claramente la recarga de batería llevada a cabo a partir de la energía que aporta el sol. La segunda gráfica muestra el módulo de la velocidad de la embarcación, calculado en base al espacio que recorre en cada iteración de tiempo. Los gráficos tercero y cuarto muestran la posición (longitud y latitud) de la embarcación a lo largo del tiempo que dura la simulación. Con toda esta información puede observarse por ejemplo que el día 30 de

agosto el USV se queda sin batería porque ha estado rastreando afloraciones en puntos distantes durante cuatro días consecutivos.

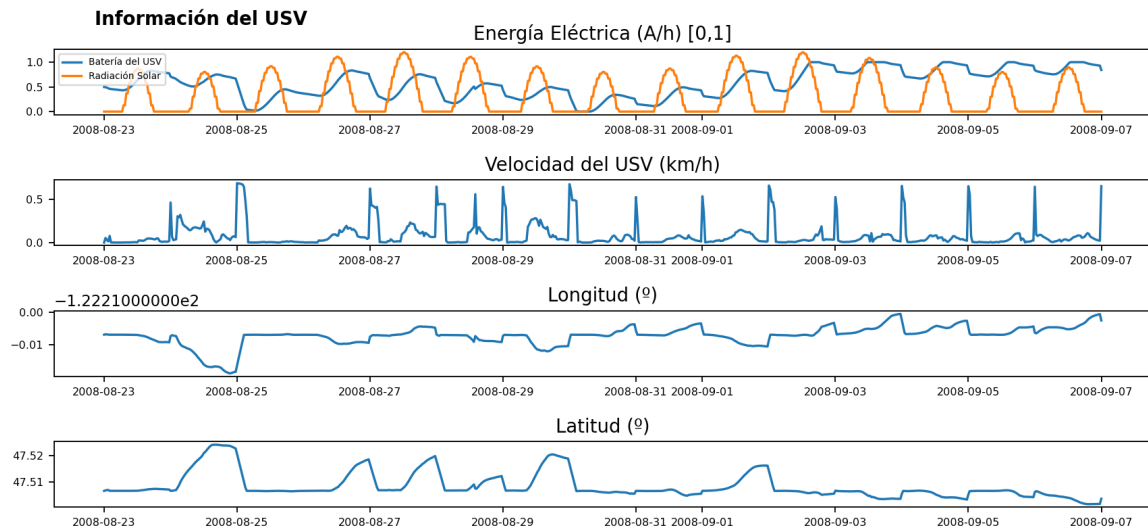


Figura 5.5: Información de los parámetros del USV durante la simulación

Finalmente, podemos observar en la primera gráfica de la figura 5.6 los valores de la densidad del *bloom* real y de su predicción. Esta gráfica está relacionada con la segunda, que muestra la diferencia entre ambos valores mediante la línea continua, y la media aritmética del error cometido entre ambas mediante la línea discontinua. Esta información es necesaria para reentrenar los parámetros del modelo predictivo.

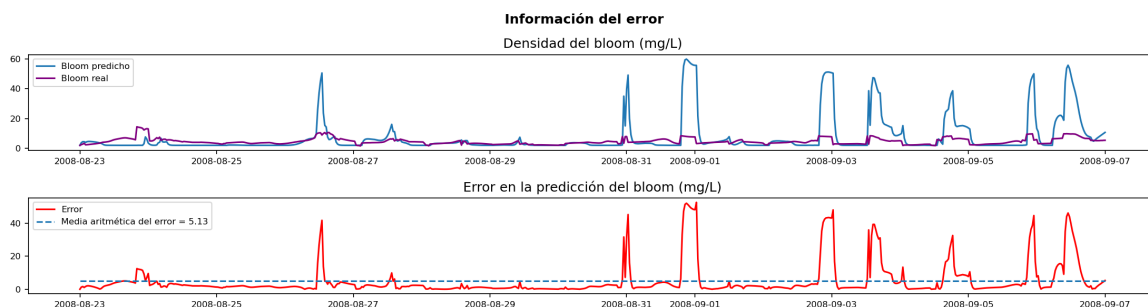


Figura 5.6: Error cometido en la predicción de HABs

Por último se generan los informes de la capa *Cloud*, que crean una tabla que resume las características más relevantes de diferentes variables de la simulación, como son la batería del USV, la radiación solar, la temperatura del agua y el tamaño del *bloom*. Para ello se realizan diferentes operaciones sobre los datos, como la media aritmética, el mínimo, el máximo, etc. La tabla correspondiente a los valores de la simulación recogida en este TFM se puede ver en el cuadro 5.1.

	Batería Eléctrica del USV (A/h)	Radiación Solar (A/h)	Temperatura del Agua (°C)	Densidad del HAB (mg/L)
Total	720.000	720.000	720.000	720.000
Media Aritmética	0.919	0.013	20.169	0.752
Desviación Típica	0.102	0.015	0.370	1.720
Mínimo	0.445	0.000	19.357	0.009
Percentil 25	0.863	0.000	19.905	0.069
Percentil 50	0.952	0.000	20.262	0.327
Percentil 75	1.000	0.028	20.516	0.599
Máximo	1.000	0.040	20.585	10.000

Tabla 5.1: Tabla con cálculos sobre los datos de la simulación

Por otro lado, el informe hace un recuento del número total de HABs detectados a lo largo de toda la simulación, de cara también al análisis de datos. En la simulación llevada a cabo para este trabajo, el recuento total de afloraciones a lo largo de los 18 días transcurridos ha sido de un total de 11, aproximadamente uno cada dos días, como podemos ver en la información representada en el cuadro 5.2.

Simulación 1	
Número de HABs detectados	11

Tabla 5.2: Número de HABs detectados en la simulación

También se genera un mapa de calor interactivo que muestra las zonas más probables en las que encontrar HABs en base a las mediciones realizadas durante la simulación. Este mapa, que muestra con colores más claros las zonas más *calientes*, se puede observar en la figura 5.7.

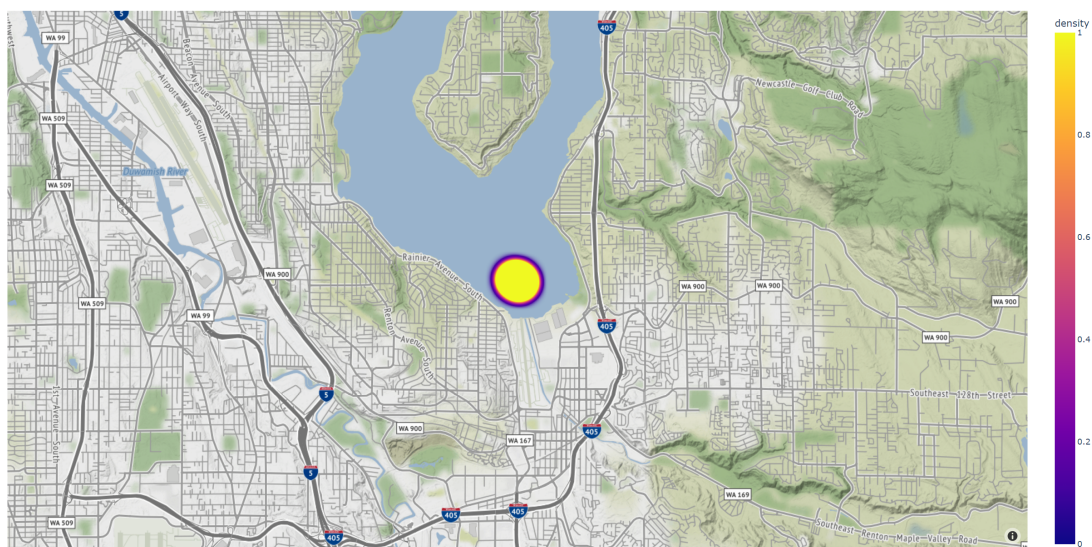


Figura 5.7: Mapa de calor de las zonas con más probabilidad de encontrar HABs

Según vamos haciendo un zoom sobre el mapa vemos con más claridad la distribución de las zonas más probables en las que puede aparecer HAB. Para ilustrar este hecho, los resultados de dos niveles de *zoom* diferentes son los representados en las figuras 5.8 y 5.9.

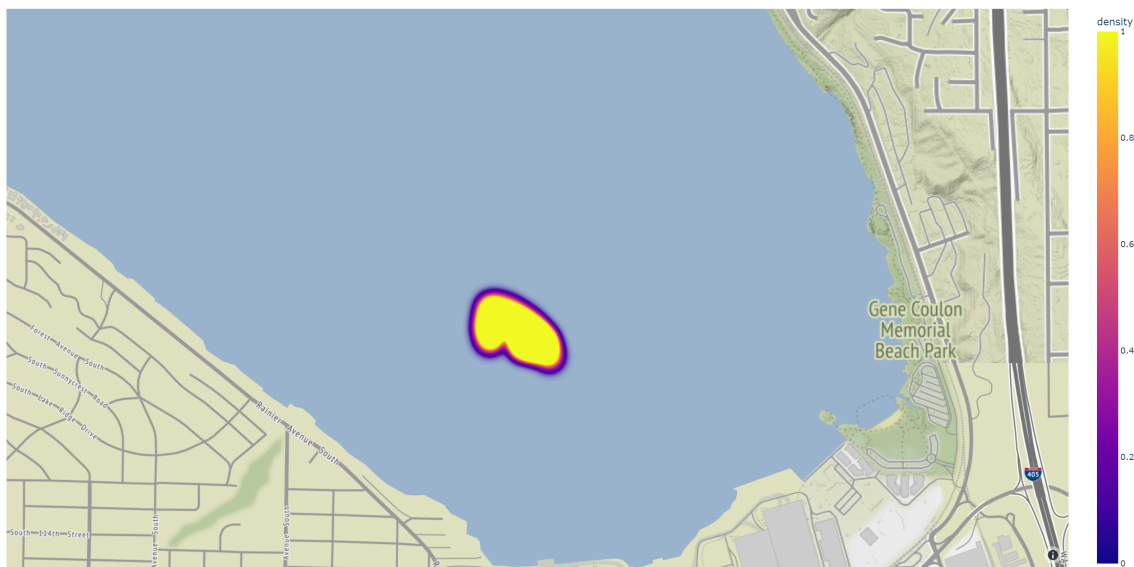


Figura 5.8: Ampliación del mapa de calor con más probabilidad de encontrar HABs



Figura 5.9: Segunda ampliación del mapa de calor con más probabilidad de encontrar HABs

5.4. Entrenamiento

Una vez finalizada la simulación y ejecutado el modelo de análisis de datos, podemos ver el error de predicción en la sección final del informe de la capa Fog. Esta primera simulación se ha realizado con los valores por defecto de las constantes, que son $k_1 = 1$, $k_2 = 1$ y $k_3 = 1$.

El resultado de dicha simulación y la diferencia entre los datos reales y los previstos para la densidad de floración ya se mostraron en la figura 5.6. El error medio aritmético de esta simulación resultó ser de 5,13, lo que es elevado. Por tanto, se activa el servicio de entrenamiento, que inicia el método de mínimos cuadrados no lineales para minimizar el error. Cuando finaliza el servicio de entrenamiento, devuelve los nuevos valores de los parámetros 3 del modelo base de densidad, que en este caso son $k_1 = 1,02424907$, $k_2 = 0,35783188$ y $k_3 = 2,2402504$, y que se sustituyen por los antiguos en las ecuaciones diferenciales. Así, se vuelve a lanzar una nueva simulación con un modelo predictivo más preciso, como puede verse en la figura 5.10, que representa la nueva diferencia entre lo predicho.

k_1 = 1,02424907
 k_2 = 0,35783188
 k_3 = 2,2402504

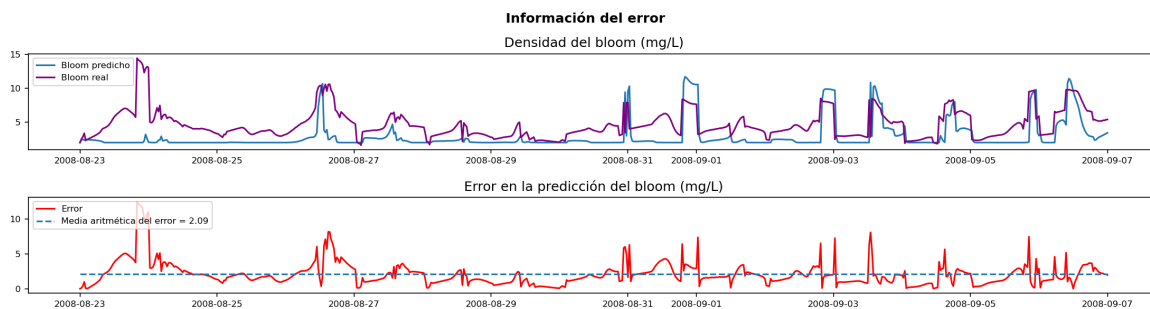


Figura 5.10: Error en la densidad del *bloom* predicho tras el entrenamiento

La figura muestra las predicciones antes y después del entrenamiento, comparadas con los datos reales de densidad de floración. La línea de densidad de los modelos ajustados sigue mejor la dinámica de los datos que la línea de la predicción sin ajustar. Además, el error medio aritmético entre la predicción y los datos reales ha disminuido de 5,13 a 2,09 con respecto a la ejecución anterior, lo que implica que el error se ha reducido en más de la mitad. Sin embargo, cabe mencionar que aunque el modelo de predicción de densidad aún no sigue la misma dinámica que los datos reales, ya es capaz de detectar los picos de la dinámica.

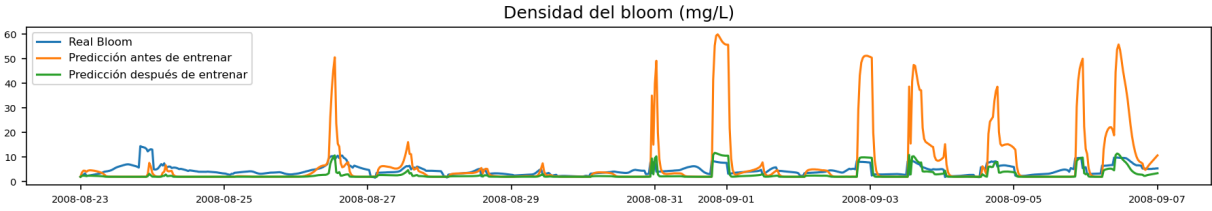


Figura 5.11: Diferencias en la predicción del *bloom* antes y después del entrenamiento

Capítulo 6

Conclusiones

En este capítulo se revisan los objetivos establecidos en la sección 1.3 del primer capítulo y proporciona un resumen general de las principales conclusiones obtenidas durante la realización de este TFM. Al final del capítulo, también se presentan algunas sugerencias de trabajos de investigación futuros que deberían realizarse a continuación de este TFM.

6.1. Revisión de objetivos

A lo largo de este documento se han abordado los objetivos establecidos al comienzo del trabajo. En esta sección, se revisará cómo se han logrado cada una de las metas establecidas, relacionando el trabajo desarrollado con los objetivos propuestos.

Profundizar en el conocimiento de las cianobacterias, los sistemas de predicción y el formalismo DEVS

El primero de los objetivos generales [O1] marcaba como meta realizar un estudio sobre el ámbito de las cianobacterias, su origen y su impacto en el medio ambiente, sobre la situación actual de los sistemas de alerta temprana y sobre el formalismo DEVS. Precisamente, en el capítulo 2 se realiza el estudio del estado del arte, y con ello queda cubierto este objetivo general. De forma más detallada, en la sección 2.1 se realizó un análisis exhaustivo de las cianobacterias, su morfología, alimentación, comportamiento e impacto tanto ambiental como económico y sanitario, cumpliendo así el objetivo específico [O1.1]. Además, en la sección 2.2 se hizo un recorrido por los sistemas de alerta temprana existentes en la literatura, cumpliendo así con el objetivo específico [O1.2]. Por último, en la sección 2.3 se profundiza en el estudio del formalismo DEVS y sus aplicaciones, obedeciendo así al objetivo específico [O1.3].

Participar en la creación de la arquitectura en DEVS de un sistema para la monitorización de HABs

El objetivo general [O2] proponía la participación en la creación de un sistema integral en DEVS para la monitorización y predicción de afloraciones de cianobacterias. Este objetivo fue completado en los capítulos 3 y 4, en los que se explica las labores realizadas en este TFM para ampliar la funcionalidad de la arquitectura global del sistema. Los subobjetivos [O2.1], [O2.2] y [O2.3] que proponen la implementación del subsistema de predicción y

análisis de datos y la creación de un modelo de entrenamiento se logran a lo largo de estos dos capítulos, aunque de forma más específica el trabajo realizado en cada uno de estos servicios se detalla respectivamente en las secciones 4.1, 4.2 y 4.3 del capítulo 4.

Integrar y verificar los módulos desarrollados en la arquitectura DEVS

En cuanto al objetivo [O3], que consistía en integrar los sub-modelos desarrollados en la arquitectura, de nuevo se satisface a lo largo de los capítulos 3 y 4, en los cuales se ven cumplidos uno a uno los objetivos específicos [O3.1], [O3.2] y [O3.3].

Comprobar el funcionamiento del sistema a través de un caso de uso

Por último, el objetivo general [O4] que proponía la comprobación del funcionamiento del sistema a través de un caso de uso, se logra en el capítulo 5 de la memoria de este TFM. En dicho capítulo se realizó una simulación completa, en la que se ha confirmado el correcto funcionamiento de todos los servicios, y más específicamente de los servicios de predicción, análisis de datos y entrenamiento, cumpliendo de esta forma los subobjetivos [O4.1], [O4.2], [O4.3] y [O4.4].

6.2. Trabajo futuro

Al analizar los diferentes capítulos de este documento, se puede determinar que se han cumplido con éxito los objetivos propuestos. Sin embargo, es posible continuar mejorando el sistema DEVS-BLOOM, motivo por el que a continuación se presentarán diferentes posibles líneas de investigación futura en esta área.

Al ser *IA-GES-BLOOM-CM* un proyecto tan complejo y que engloba tantos elementos dentro de su infraestructura, se abren numerosos caminos de cara a un trabajo futuro para mejorar el sistema. De hecho, hay algunas funcionalidades del proyecto que se encuentran aún en fases poco avanzadas del proceso de desarrollo.

En la capa *Edge*, cabría lanzar simulaciones que tuviesen más de un USV, lo cual implicaría también una línea de investigación futura en la capa *Fog*, para poder planificar varias trayectorias dentro de la misma masa de agua y optimizar los resultados.

Dentro de la capa *Fog*, el planificador de trayectorias del USV y el servicio de detección de valores atípicos ofrecen muchas vías de investigación futura. Por ejemplo, la optimización de la trayectoria del barco teniendo en cuenta los puntos en los que debe medir es sin duda un problema que puede abordarse desde diferentes puntos de vista.

En cuanto al servicio de predicción, se pueden plantear diferentes líneas de investigación. En primer lugar, podrían idearse unas ecuaciones más sofisticadas para modelar el comportamiento del *bloom* aplicando también la dinámica de fluidos al transporte de nutrientes y/o teniendo en cuenta factores adicionales como la densidad y la viscosidad de la masa de agua en la que se está realizando la simulación. También podría modelarse el comportamiento de las cianobacterias con redes neuronales del tipo LSTM (del inglés, *Long Short-Term Memory*). O utilizando las ecuaciones diferenciales originalmente propuestas, podrían probarse diferentes métodos numéricos de integración, estudiando también el coste computacional asociado a cada uno de los mismos. Por último, podría modelarse también el movimiento vertical de las cianobacterias.

En lo referente a los informes de la capa *Fog*, generados por el servicio de análisis de datos, podrían presentarse más gráficas en distintos formatos y comparativas de unos parámetros con otros para que el análisis de las correlaciones que existen entre ellos fuera más sencilla de visualizar para el usuario. Además, deberían incluirse representaciones en 3 dimensiones para poder observar también el desplazamiento vertical de las cianobacterias. Asimismo, sería muy interesante que fuera el propio usuario quién pudiera decidir qué información quiere ver en el informe y de qué manera desea verla mediante la personalización de informes.

Por último, la capa *Cloud* se encuentra todavía en fases tempranas del proceso de implementación. Actualmente los servicios que pertenecen a esta capa se ejecutan localmente, pero la idea es que se haga de manera distribuida de forma que la capa *Cloud* pueda gestionar información de diferentes masas de agua simultáneamente. En cuanto a los servicios que se ubican en ella, por un lado el de análisis de *big data* está aún en fases tempranas de su desarrollo y tiene mucho potencial de cara a comprender el comportamiento y la tendencia de los datos. Por otro lado, el servicio de entrenamiento basado en el método del descenso de gradiente, puede mejorarse añadiendo más iteraciones comenzando desde diferentes puntos iniciales, ya que cuantas más iteraciones se lleven a cabo más probabilidad hay de encontrar el óptimo global de la función de error, reduciendo las posibilidades de que el ajuste de los parámetros de los modelos quede *atrapado* en un óptimo local.

En cuanto a los informes generados por la capa *Cloud*, de cara al futuro pueden incluirse gráficos basados en las estadísticas de cada masa de agua (gracias al servicio de *big data*), comparaciones de resultados entre una simulación y otra, etc. Además, se podrían calcular también otro tipo de métricas para cada conjunto de datos con el objetivo de evaluar el rendimiento del simulador.

De cara a la realización de nuevas simulaciones, sería de mucha utilidad modelar uno o varios embalses españoles de características diferentes para poder ejecutar las simulaciones con datos más actuales y sobre zonas más cercanas, y comprobar así el funcionamiento del sistema en otras masas de agua virtuales.

Y como última línea de investigación futura, habría que trabajar en el análisis del funcionamiento del sistema en un caso real, con datos reales de los sensores, y comprobar de esta forma la utilidad real del sistema para predecir y gestionar *blooms* de cianobacterias en diferentes entornos acuáticos. Actualmente se están probando prototipos de USV y sensores cuyo funcionamiento se incluirá en *DEVS-BLOOM* para que este funcione combinando elementos virtuales y reales como verdaderos gemelos digitales.

Anexos

Anexo A

Introduction

This chapter explains the motivation that has driven the research carried out in this work, as well as the objectives to be achieved at the end of it and the structure of the document.

A.1. Motivation

Water scarcity today affects about 4 billion people (i.e. two-thirds of the world's population), a problem that is exacerbated for at least one month a year [1].

When a territory extracts at least 25 % of its renewable freshwater resources, it is said to suffer from water stress [1]. Currently, 2.3 billion people live in countries with this problem, 733 million of whom live in places with high or critical water stress. In addition, 3.2 billion people reside in agricultural areas with high to very high water scarcity, of which 1.2 billion are severely water-stressed.

In addition, more than 1.3 billion people do not have access to safe drinking water, in violation of the right to affordable access to safe drinking water established by the United Nations Committee on Economic, Social and Cultural Rights [2].

The problem of water scarcity is closely linked to the term *water quality*. [3], which refers to the chemical, physical and biological characteristics of water depending on its intended use. In fact, for water to be safe, it must meet certain quality standards, and to determine whether these standards are met, certain characteristics such as its temperature, mineral content and the amount of bacteria it contains are measured and analyzed.

Today, water quality is an issue of great importance for many reasons [4]. Firstly, the world's population has increased considerably in recent years, which implies a greater need for water to ensure the global supply. Secondly, the climate change that the planet has been undergoing for several decades is causing long periods of drought, with a decrease in the frequency and quantity of rainfall that significantly affects water reserves. Finally, overexploitation and overpollution of water resources in both urban and rural areas increase the importance of meeting quality standards, which are becoming increasingly difficult to guarantee.

On the one hand, in large urban centers we find problems of excessive use and water pollution due to the large volume of waste generated, including sewage and discharges from industrial areas that, in large quantities, end up in water resources [5]. On the other

hand, in rural areas we find numerous agricultural and livestock farms that also require a large amount of water, and also use fertilizers and pesticides that can end up filtered into it. All this produces changes in the composition of the water due to the introduction of unwanted substances or microorganisms that can become a health risk.

One of the microorganisms that can end up invading water reserves and putting our health at risk are cyanobacteria. These bacteria, also called blue-green algae because of their more common color [6, 7], are microscopic unicellular organisms that are found naturally in all types of water, whether fresh, marine or brackish (fresh and salt water combined). They perform photosynthesis, using sunlight to feed on nutrients present in the water. Therefore, in warm environments rich in phosphorus and nitrogen, cyanobacteria can multiply rapidly, creating blooms that spread throughout the body of water and can become visible when they emerge to the surface. This is when their blue-green color becomes apparent, as can be seen in the figure A.1.



Figura A.1: Cyanobacteria bloom on the surface of a river (source: [8])

These organisms are present in all environments and play an important role as primary producers in all environments, so they are good and necessary in nature. However, when they multiply in an exaggerated and uncontrolled manner, they form blooms that are harmful to the health of humans, animals or the environment by blocking the sunlight that other organisms need to live, robbing them of oxygen and nutrients, and producing highly toxic secondary metabolites generically called *cyanotoxins* [9].

Therefore, swimming or engaging in activities in waters containing cyanobacterial blooms, drinking water that comes from a lake or reservoir that has a proliferation of them, or eating fish infected with the toxins they produce is tremendously harmful to both human and animal health [10]. In humans, these toxins can cause abdominal pain, difficulty breathing, irreversible liver and kidney damage, nervous system problems and tumor growth.

For this reason, considerable effort has been invested in detecting these organisms in reservoirs and lakes. The most common procedure for this is to manually take water samples for later analysis to determine whether cyanotoxins are contaminating the water [11]. This entails a great effort on the part of the personnel involved, since the manual extraction of samples and their subsequent analysis has a high cost in time and money, in addition to not being entirely effective due to the large amount of surface area that needs to be covered in order to carry out an exhaustive study. The manual method of collecting samples for water quality analysis can be seen in figure A.2.



Figura A.2: Manual extraction of samples for water analysis (source: [12])

In addition, financial and personnel resource constraints reduce manual collection to times of the year when the occurrence of cyanobacterial blooms is most likely, taking place at few geographic locations and with minimal frequencies. The delay suffered by analytical results and the limited ability to interpret the current scenario diminishes the capacity for prediction, prevention and mitigation of HABs by the authorities responsible for drinking water distribution and its recreational uses [13].

This sampling process is occasionally aided by certain automatic instruments that acquire water status information at certain strategic points. A good example of this type of instruments are the periodic sampling buoys [14] that are anchored in different points of lakes and reservoirs, and that have sensors that, on a regular basis, take measurements on different parameters related to water quality, although, of course, always at the same point. One of these buoys can be seen in figure A.3.

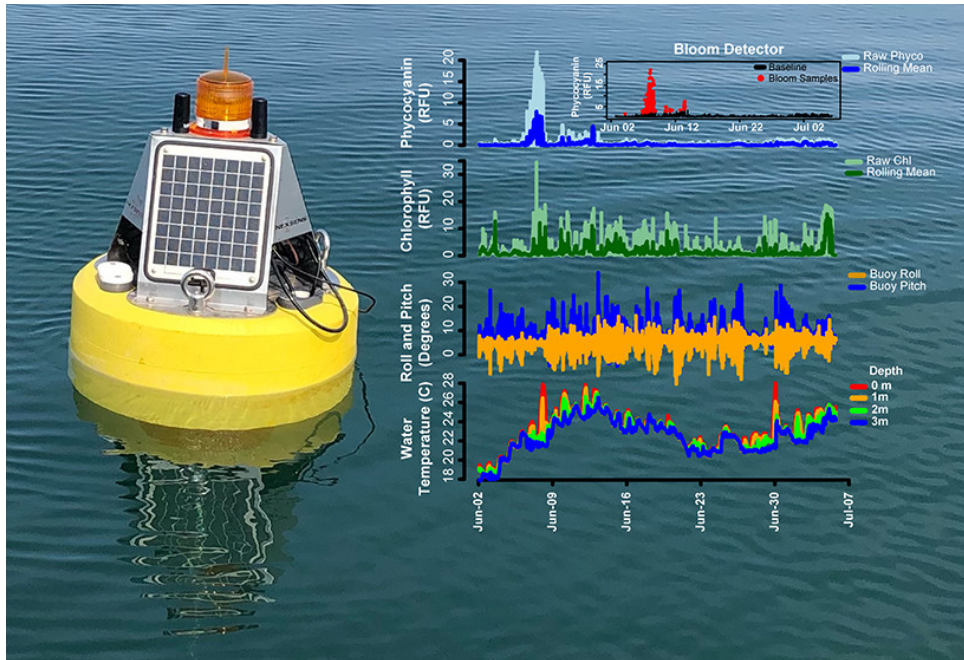


Figura A.3: Periodic water sampling buoy (source: [14]).

A.2. Context

To provide a comprehensive solution to the problem of monitoring and management of cyanobacteria, it is necessary to go a step further, and not only detect the outcrops of these microorganisms with greater frequency and resolution, but also predict their spatiotemporal evolution. For the latter, there are currently some early warning systems [15] based on the simulation of numerical models, such as those implemented by commercial tools like MIKE ([16]) or EEMS ([17]). However, the exclusive use of simulation tools is often not sufficient, as the complexity of the biological and transport processes of cyanobacteria makes it difficult to correctly tune the numerous parameters that exist in these commercial models. Moreover, it introduces relevant uncertainties in model predictions.

Therefore, an integral solution is proposed, i.e., a solution that not only implements the numerical models but also incorporates the whole ecosystem of apparatus to perform measurements that complement and feed the predictions made by the numerical models with real data. This idea is being developed in the Synergic Project in New and Emerging Scientific Areas of the Community of Madrid, called *Hacia un sistema Integral para la Alerta y GESTión de BLOOMs de cianobacterias en aguas continentales, IA-GES-BLOOM-CM*, in which a complex system is being developed that suggests an alternative to improve sampling, creating new models, tuning them and bringing the results closer to the managers, all following an architecture based on IoT (Internet of Things) that involves elements of different nature, such as models, sensors, ships and managers.

Due to the complexity involved in a system of such characteristics, it is convenient to create an infrastructure in which a digital twin of the whole system is realized, with systematic modeling and simulation tools. In this context, the DEVS (Discrete Event System Specification) formalism [18, 19] is very useful, since it provides a solid modeling and simulation framework for discrete events to be able to develop such sophisticated

systems as the proposed one. In this line, another project called *Más allá del uso de tecnologías digitales en blooms de cianobacterias: gestión inteligente de cianobacterias mediante el uso de gemelos digitales y computación en el borde (SMART-BLOOMS)* is also being developed at the Complutense University of Madrid.

This TFM is therefore focused on the collaboration in the construction of the overall architecture of the system, and in the implementation of some services that are part of the complex infrastructure described. Both the architecture design and its implementation are based on the DEVS formalism. For this purpose, an IoT-based organization is followed. All this allows us to efficiently guarantee a system not only for detection but also for prediction and monitoring of cyanobacteria outbreaks, thus combating the risks that these microorganisms generate for health and the environment.

A.3. Objectives

A series of objectives have been defined for this work, which are listed below. These objectives can be categorized as general or specific. The general objectives are oriented towards the development of large-scale goals, while the specific ones go into more depth on the different issues being addressed.

- [O1] To deepen the knowledge of cyanobacteria, prediction systems and DEVS formalism.
 - [O1.1] To study cyanobacterial blooms, their origin and their impact on water quality and the health of living organisms.
 - [O1.2] Review existing early warning systems for cyanobacterial blooms.
 - [O1.3] To deepen the knowledge of the DEVS formalism and its operation.
- [O2] To participate in the creation of the DEVS architecture of an integral system for the monitoring and management of HABs.
 - [O2.1] Devise a modeling and simulation module to predict the occurrence of cyanobacteria.
 - [O2.2] Implement a data analysis service that allows its users to validate and visualize the results obtained with each simulation performed.
 - [O2.3] Develop a training protocol in the cloud that helps in the definition of predictive models for the early detection of outcrops.
- [O3] Integrate and verify the developed modules into the overall DEVS architecture.
 - [O3.1] Integrate the developed HABs prediction service into the overall architecture.
 - [O3.2] Add the implemented data analysis service to the system.
 - [O3.3] To include in the system framework the devised training service.
- [O4] Test the operation of the system through a use case.
 - [O4.1] Exemplify the operation of the complete system with a simulation of a real lake.
 - [O4.2] Test the HABs prediction service and compare the prediction with real data.

[O4.3] Activate the data analysis service and check that the generated reports are correct.

[O4.4] Use the training service and check that it optimizes the prediction.

A.4. Document structure

This document is divided into six chapters through which the development of the work carried out is described. This section provides a brief summary of the contents of each of these chapters.

Chapter 1 introduces the problem of cyanobacterial blooms in reservoirs and lakes, and explains the context in which the work has arisen and its motivation. It also includes the list of general and specific objectives to be achieved at the end of the research.

Next, in chapter 2 the study of cyanobacteria, the cause of the outcrops and their environmental, economic and health impact are discussed in depth. A brief state of the art on early warning systems to detect, monitor and manage the presence of harmful toxins produced by cyanobacterial blooms is also presented. Finally, the most relevant features of the DEVS formalism, its structure and operation are presented.

Chapter 3 details the architecture of the proposed system to efficiently manage cyanobacterial blooms. The infrastructure, developed in DEVS and integrated in an IoT environment that relies on the use of digital twins, consists of three layers that are explained in a way that makes it easy to visualize the overall picture of the proposed framework, which aims to facilitate prediction, speed up the detection of HABs, and improve the management of the problems they entail in real time.

Subsequently, in chapter 4, we go into the development of three services that are part of the architecture presented in chapter 3. These services are the prediction model, which allows to simulate the behavior of outcrops in real time; the data analysis service, which provides graphs for the study of the simulation scenario; and the cloud training system, which allows to retrain the prediction model to optimize the results. These services correspond to the prediction and data analysis subsystem of the proposed architecture.

The work continues in chapter 5, where everything presented and developed in chapters 3 and 4 is exemplified with a use case consisting of a simulation carried out in Lake Washington (United States) using synthetic lake data extracted from the EEMS tool. In the execution, the operation of the complete system can be observed, allowing us to study it and to pay special attention to the prediction model, the data analysis service and the training service.

In chapter 6 the objectives detailed at the beginning of the document are reviewed and possible avenues of research for future work related to the field of modeling, prediction and management of cyanobacteria and their derived problems for public health and the environment are proposed.

Finally, the document contains two annexes consisting of the English translation of the chapters 1 and 6, which are, respectively, the Introduction and the Conclusions.

Anexo B

Conclusions

This chapter reviews the objectives stated in the 1.3 section of the first chapter and provides a general summary of the main conclusions obtained during the completion of this TFM. At the end of the chapter, some suggestions for future research work that should follow this TFM are also presented.

B.1. Review of objectives

Throughout this paper, the objectives stated at the beginning of the paper have been addressed. In this section, we will review how each of the established goals have been achieved, relating the work developed with the proposed objectives.

Deepening the knowledge of cyanobacteria, prediction systems and the DEVS formalism

The first of the general objectives [O1] set as a goal to carry out a study on the field of cyanobacteria, their origin and their impact on the environment, on the current situation of early warning systems and on the DEVS formalism. Precisely, in chapter 2 the state of the art is studied, thus covering this general objective. In more detail, in section 2.1 an exhaustive analysis of cyanobacteria, their morphology, feeding, behavior and environmental, economic and sanitary impact was carried out, thus fulfilling the specific objective [O1.1]. In addition, in the section 2.2 a tour of the existing early warning systems in the literature was made, thus fulfilling the specific objective [O1.2]. Finally, in the 2.3 section, the study of the DEVS formalism and its applications is deepened, thus obeying the specific objective [O1.3].

To participate in the creation of the DEVS architecture of a system for HABs monitoring

The general objective [O2] proposed the participation in the creation of an integral system in DEVS for the monitoring and prediction of cyanobacterial blooms. This objective was completed in chapters 3 and 4, which explain the work done in this TFM to extend the functionality of the overall system architecture. The sub-objectives [O2.1], [O2.2] and [O2.3] that propose the implementation of the data prediction and analysis subsystem and the creation of a training model are achieved throughout these two chapters, although

more specifically the work done in each of these services is detailed respectively in the sections 4.1, 4.2 y 4.3 of chapter 4.

Integrating and verifying the modules developed in the DEVS architecture

As for objective [O3], which consisted of integrating the sub-models developed in the architecture, it is again satisfied throughout chapters 3 and 3, in which the specific objectives [O3.1], [O3.2] and [O3.3] are fulfilled one by one.

Testing the system operation through a use case

Finally, the general objective [O4], which proposed the testing of the system operation through a use case, is achieved in the chapter 5 of the memory of this TFM. In this chapter, a complete simulation was carried out, in which the correct operation of all the services, and more specifically of the prediction, data analysis and training services, has been confirmed, thus fulfilling the sub-objectives [O4.1], [O4.2], [O4.3] and [O4.4].

B.2. Future work

By analyzing the different chapters of this document, it can be determined that the proposed objectives have been successfully met. However, it is possible to continue improving the DEVS-BLOOM system, which is why different possible lines of future research in this area will be presented below.

As *IA-GES-BLOOM-CM* is such a complex project and encompasses so many elements within its infrastructure, there are many avenues for future work to improve the system. In fact, there are some functionalities of the project that are still in the early stages of the development process.

In the Edge layer, it would be possible to launch simulations with more than one USV, which would also imply a future line of research in the Fog layer, to be able to plan several trajectories within the same body of water and optimize the results.

Within the Fog layer, the USV trajectory planner and outlier detection service offer many avenues for future research. For example, the optimization of the ship's trajectory taking into account the points at which it should measure is certainly a problem that can be approached from different points of view.

As for the prediction service, different lines of research can be envisaged. Firstly, more sophisticated equations could be devised to model the behavior of the bloom by also applying fluid dynamics to nutrient transport and/or taking into account additional factors such as the density and viscosity of the water body in which the simulation is being carried out. The behavior of cyanobacteria could also be modeled with neural networks of the LSTM (Long Short-Term Memory) type. Or, using the originally proposed differential equations, different numerical integration methods could be tested, studying also the computational cost associated with each of them. Finally, the vertical movement of cyanobacteria could also be modeled.

Regarding the reports of the fog layer, generated by the data analysis service, more graphs in different formats and comparisons of some parameters with others could be presented so that the analysis of the correlations that exist between them would be easier for the

user to visualize. In addition, representations in 3 dimensions should be included so that the vertical displacement of the cyanobacteria can also be observed. Also, it would be very interesting if the user himself could decide what information he wants to see in the report and in what way he wants to see it through report customization.

Finally, the Cloud layer is still in the early stages of the implementation process. Currently the services that belong to this layer are executed locally, but the idea is that this will be done in a distributed manner so that the Cloud layer can manage information from different water bodies simultaneously. As for the services that are located in it, on the one hand, the big data analysis service is still in early stages of development and has a lot of potential for understanding the behavior and trend of the data. On the other hand, the training service based on the gradient descent method can be improved by adding more iterations starting from different starting points, since the more iterations carried out, the more likely it is to find the global optimum of the error function, reducing the chances that the fit of the model parameters is trapped in a local optimum.

As for the reports generated by the Cloud layer, going forward, graphs based on the statistics of each water body (thanks to the big data service), comparisons of results between one simulation and another, etc. can be included. In addition, other types of metrics could also be calculated for each dataset in order to evaluate the performance of the simulator.

With a view to carrying out new simulations, it would be very useful to model one or more Spanish reservoirs with different characteristics in order to run the simulations with more current data and on closer areas, and thus check the performance of the system in other virtual bodies of water.

And as a last line of future research, it would be necessary to work on the analysis of the operation of the system in a real case, with real data from the sensors, and thus check the real usefulness of the system for predicting and managing cyanobacteria blooms in different aquatic environments. USV and sensor prototypes are currently being tested and their operation will be included in *DEVS-BLOOM* to make it work by combining virtual and real elements as real digital twins.

Bibliografía

- [1] U. Nations. «Summary Progress Update 2021: SDG 6 — water and sanitation for all». En: *UN Water* (2021). URL: https://web.archive.org/web/20221221065046/https://www.unwater.org/sites/default/files/app/uploads/2021/12/SDG-6-Summary-Progress-Update-2021_Version-July-2021a.pdf.
- [2] U. Nations. «Human Right to Water and Sanitation». En: *UN Water Decade Programme on Advocacy and Communication* (2002). URL: <https://web.archive.org/web/20221229183508/https://www.acnur.org/fileadmin/Documentos/BDL/2012/8789.pdf>.
- [3] G. E. Cordy. «A Primer on Water Quality». En: *USGS* (mar. de 2001). URL: <https://web.archive.org/web/20221229184248/https://pubs.usgs.gov/fs/fs-027-01/pdf/FS-027-01.pdf>.
- [4] D. L. Johnson, S. H. Ambrose, T. J. Bassett *et al.* «Meanings of Environmental Terms». En: *Journal of Environmental Quality* (mayo de 1997). DOI: 10.2134/jeq1997.00472425002600030002x. URL: <https://doi.org/10.2134/jeq1997.00472425002600030002x>.
- [5] M. Denchak. «Water Pollution: Everything You Need to Know». En: *NRDC* (abr. de 2022). URL: <https://web.archive.org/web/20221229185709/https://www.nrdc.org/stories/water-pollution-everything-you-need-know>.
- [6] R. P. Sinha y D.-P. Häder. «UV-protectants in cyanobacteria». En: *Plant Science* 174.3 (2008), págs. 278-289. ISSN: 0168-9452. DOI: <https://doi.org/10.1016/j.plantsci.2007.12.004>. URL: <https://www.sciencedirect.com/science/article/pii/S0168945207003202>.
- [7] B. U. of California. «Cyanobacteria: Life History and Ecology». En: *Museum of Paleontology* (2012). URL: <https://web.archive.org/web/20120919015239/http://www.ucmp.berkeley.edu/bacteria/cyanolh.html>.
- [8] A. Shutterstock. «Cianobacterias en el Río de La Plata». En: *PERFIL* (2022). URL: <https://web.archive.org/web/20221229193109/https://www.perfil.com/noticias/salud/emiten-una-alerta-por-cianobacterias-en-balnearios-bonaerenses-y-piden-que-la-gente-no-se-meta-al-agua.phtml>.
- [9] L. Bláha, P. Babica y B. Maršálek. «Toxins produced in cyanobacterial water blooms – toxicity and risks». En: *Interdisciplinary Toxicology* (2009). DOI: 10.2478/v10102-009-0006-2. URL: <https://doi.org/10.2478/v10102-009-0006-2>.
- [10] H. W. Paerl y T. G. Otten. «Harmful Cyanobacterial Blooms: Causes, Consequences, and Controls». En: *Microbial Ecology* (2013). DOI: 10.1007/s00248-012-0159-y. URL: <https://doi.org/10.1007/s00248-012-0159-y>.
- [11] U. S. E. P. Agency. «Detection Methods for Cyanotoxins». En: *United States Environmental Protection Agency* (2022). URL: <https://web.archive.org/web/>

- 20221229213542/<https://www.epa.gov/ground-water-and-drinking-water/detection-methods-cyanotoxins>.
- [12] D. R. LeBlanc. «USGS Scientists Collecting Water-Quality Samples from Shallow Groundwater under Ashumet Pond». En: *USGS* (jul. de 2014). URL: <https://web.archive.org/web/20221229212954/https://www.usgs.gov/media/images/usgs-scientists-collecting-water-quality-samples-shallow-groundwater-under-ashumet>.
- [13] J. Meriluoto, L. Spoof y G. A. Codd. *Handbook of cyanobacterial monitoring and cyanotoxin analysis*. John Wiley & Sons, 2017.
- [14] T. R. Miller, W. Tarpey, J. Nuese *et al.* «Real-Time Monitoring of Cyanobacterial Harmful Algal Blooms with the Panther Buoy». En: *ACS EST Water* (mayo de 2022). DOI: 10.1021/acsestwater.2c00072. URL: <https://doi.org/10.1021/acsestwater.2c00072>.
- [15] H. Almuhtaram, F. A. Kibuye, S. Ajjampur *et al.* «State of knowledge on early warning tools for cyanobacteria detection». En: *Ecological Indicators* 133 (2021). ISSN: 1470-160X. DOI: <https://doi.org/10.1016/j.ecolind.2021.108442>. URL: <https://www.sciencedirect.com/science/article/pii/S1470160X21011079>.
- [16] DHI. *MIKE ECO Lab*. (Visited on 11/2022). 2022. URL: <https://www.mikepoweredbydhi.com/products/mike-eco-lab>.
- [17] DSI. *EE Modeling System*. <https://www.eemodelingsystem.com>. Ene. de 2022. URL: <https://www.eemodelingsystem.com/user-center/downloads>.
- [18] J. L. Risco-Martín, S. Mittal, J. C. F. Jiménez *et al.* «Reconsidering Performance of DEVS Modeling and Simulation Environments Using the DEVStone Benchmark». En: *SAGE* (2017).
- [19] B. Zeigler. *Theory of modeling and simulation : discrete event and iterative system computational foundations*. London, United Kingdom: Academic Press, an imprint of Elsevier, 2019. ISBN: 978-0-12-813370-5.
- [20] C. Tamulonis, M. Postma y J. Kaandorp. «Modeling Filamentous Cyanobacteria Reveals the Advantages of Long and Fast Trichomes for Optimizing Light Exposure». En: *Plos One* (jul. de 2011). DOI: 10.1371/journal.pone.0022084. URL: <https://doi.org/10.1371/journal.pone.0022084>.
- [21] B. A. Whitton. *Ecology of Cyanobacteria II: Their Diversity in Space and Time*. Durham, United Kingdom: Springer, 2012. ISBN: 978-94-007-3855-3.
- [22] C. Lab. «Cianobacterias marinas del género Prochlorococcus». En: *Sinc* (feb. de 2020). URL: <https://web.archive.org/web/20221231104110/https://www.agenciasinc.es/Noticias/No-solo-de-fotosintesis-viven-las-cianobacterias-marinas>.
- [23] S. Blog. «Cyanobacteria contribute to the early atmosphere». En: *Labroots* (nov. de 2015). URL: <https://web.archive.org/web/20221231104448/https://www.labroots.com/trending/microbiology/1963/fresh-air-courtesy-of-cyanobacteria>.
- [24] J. W. Schopf y B. M. Packer. «Early Archean (3.3-Billion to 3.5-Billion-Year-Old) Microfossils from Warrawoona Group, Australia». En: *Science* 237.4810 (jul. de 1987). DOI: 10.1126/science.11539686. URL: <https://doi.org/10.1126/science.11539686>.
- [25] A. Aguilera, M. Klemenčič, D. J. Sueldo *et al.* «Cell Death in Cyanobacteria: Current Understanding and Recommendations for a Consensus on Its Nomenclature». En:

- Frontiers in Microbiology* (mar. de 2021). DOI: 10.3389/fmicb.2021.631654. URL: <https://doi.org/10.3389/fmicb.2021.631654>.
- [26] J. Huisman, G. A. Codd, H. W. Paerl *et al.* «Cyanobacterial blooms». En: *Nature Reviews Microbiology* (jun. de 2018). DOI: 10.1038/s41579-018-0040-1. URL: <https://doi.org/10.1038/s41579-018-0040-1>.
- [27] D. Claessen, D. E. Rozen, O. P. Kuipers *et al.* «Bacterial solutions to multicellularity: a tale of biofilms, filaments and fruiting bodies». En: *Nature Reviews Microbiology* (ene. de 2014). DOI: 10.1038/nrmicro3178. URL: <https://doi.org/10.1038/nrmicro3178>.
- [28] D. J. Nürnberg, V. Mariscal, J. Parker *et al.* «Branching and intercellular communication in the Section V cyanobacterium *Mastigocladus laminosus*, a complex multicellular prokaryote». En: *Wiley* (ene. de 2014). DOI: 10.1111/mmi.12506. URL: <https://doi.org/10.1111/mmi.12506>.
- [29] P. Fay. «Oxygen relations of nitrogen fixation in cyanobacteria». En: *ASM* (jun. de 1992). DOI: 10.1128/mr.56.2.340-373.1992. URL: <https://doi.org/10.1128/mr.56.2.340-373.1992>.
- [30] C. A. Kerfeld, S. Heinhorst y G. C. Cannon. «Bacterial Microcompartments». En: *Annual Reviews* (oct. de 2010). DOI: 10.1146/annurev.micro.112408.134211. URL: <https://doi.org/10.1146/annurev.micro.112408.134211>.
- [31] A. Nagarajan y H. B. Pakrasi. «Membrane-Bound Protein Complexes for Photosynthesis and Respiration in Cyanobacteria». En: *Wiley* (jun. de 2016). DOI: 10.1002/9780470015902.a0001670.pub2. URL: <https://doi.org/10.1002/9780470015902.a0001670.pub2>.
- [32] C. W. Mullineaux y A. Wilde. «Bacterial Blooms: The social life of cyanobacteria». En: *eLife* (jun. de 2021). DOI: 10.7554/eLife.70327. URL: <https://doi.org/10.7554/eLife.70327>.
- [33] V. A. Donkor, D. H. Amewor y D.-P. Häder. «Effects of tropical solar radiation on the motility of filamentous cyanobacteria». En: *FEMS Microbiology Ecology* 12 (jun. de 1993). DOI: 10.1111/j.1574-6941.1993.tb00026.x. URL: <https://doi.org/10.1111/j.1574-6941.1993.tb00026.x>.
- [34] A. Fourçans, A. Solé, E. Diestra *et al.* «Vertical migration of phototrophic bacterial populations in a hypersaline microbial mat from Salins-de-Giraud (Camargue, France)». En: *FEMS Microbiology Ecology* 57.3 (sep. de 2006), págs. 367-377. ISSN: 0168-6496. DOI: <https://doi.org/10.1111/j.1574-6941.2006.00124.x>. eprint: <https://academic.oup.com/femsec/article-pdf/57/3/367/19528592/57-3-367.pdf>.
- [35] L. L. Richardson y R. W. Castenholz. «Diel Vertical Movements of the Cyanobacterium *Oscillatoria terebriformis* in a Sulfide-Rich Hot Spring Microbial Mat». En: *Applied and Environmental Microbiology* 53.9 (1987), págs. 2142-2150. DOI: <https://journals.asm.org/doi/abs/10.1128/aem.53.9.2142-2150.1987>. eprint: <https://journals.asm.org/doi/pdf/10.1128/aem.53.9.2142-2150.1987>.
- [36] N. Salmaso, C. Bernard, J.-F. Humbert *et al.* «Basic Guide to Detection and Monitoring of Potentially Toxic Cyanobacteria». En: *Handbook of Cyanobacterial Monitoring and Cyanotoxin Analysis*. John Wiley Sons, Ltd, 2016. Cap. 6, págs. 46-69. ISBN: 9781119068761. DOI: <https://doi.org/10.1002/9781119068761.ch6>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119068761>.

- ch6. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119068761.ch6>.
- [37] A. De Los Ríos, M. Grube, L. G. Sancho *et al.* «Ultrastructural and genetic characteristics of endolithic cyanobacterial biofilms colonizing Antarctic granite rocks». En: *FEMS Microbiology Ecology* 59.2 (feb. de 2007), págs. 386-395. ISSN: 0168-6496. DOI: 10.1111/j.1574-6941.2006.00256.x. eprint: <https://academic.oup.com/femsec/article-pdf/59/2/386/18103173/59-2-386.pdf>. URL: <https://doi.org/10.1111/j.1574-6941.2006.00256.x>.
- [38] K. D. Jöhnk, J. Huisman, J. Sharples *et al.* «Summer heatwaves promote blooms of harmful cyanobacteria». En: *Wiley* 12 (nov. de 2007). DOI: 10.1111/j.1365-2486.2007.01510.x. URL: <https://doi.org/10.1111/j.1365-2486.2007.01510.x>.
- [39] H. W. Paerl y V. J. Paul. «Climate change: Links to global expansion of harmful cyanobacteria». En: *Water Research* 46.5 (2012). Cyanobacteria: Impacts of climate change on occurrence, toxicity and water quality management, págs. 1349-1363. ISSN: 0043-1354. DOI: <https://doi.org/10.1016/j.watres.2011.08.002>. URL: <https://www.sciencedirect.com/science/article/pii/S0043135411004386>.
- [40] K. Shi, Y. Zhang, Y. Zhou *et al.* «Long-term MODIS observations of cyanobacterial dynamics in Lake Taihu: Responses to nutrient enrichment and meteorological factors». En: *Scientific Reports* 7 (ene. de 2017). DOI: 10.1038/srep40326. URL: <https://doi.org/10.1038/srep40326>.
- [41] A. M. Michalak, E. J. Anderson, D. Beletsky *et al.* «Record-setting algal bloom in Lake Erie caused by agricultural and meteorological trends consistent with expected future conditions». En: *Proceedings of the National Academy of Sciences* 110.16 (2013), págs. 6448-6452. DOI: 10.1073/pnas.1216006110. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.1216006110>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1216006110>.
- [42] J. S. Metcalf, S. A. Banack, J. T. Powell *et al.* «Public health responses to toxic cyanobacterial blooms: perspectives from the 2016 Florida event». En: *Water Policy* 20.5 (jul. de 2018), págs. 919-932. ISSN: 1366-7017. DOI: 10.2166/wp.2018.012. eprint: <https://iwaponline.com/wp/article-pdf/20/5/919/492626/020050919.pdf>. URL: <https://doi.org/10.2166/wp.2018.012>.
- [43] L. Sitoki, R. Kurmayer y E. Rott. «Spatial variation of phytoplankton composition, biovolume, and resulting microcystin concentrations in the Nyanza Gulf (Lake Victoria, Kenya)». En: *Hydrobiologia* (mar. de 2012). DOI: 10.1007/s10750-012-1062-8. URL: <https://doi.org/10.1007/s10750-012-1062-8>.
- [44] T. S. Bianchi, E. Engelhaupt, P. Westman *et al.* «Cyanobacterial blooms in the Baltic Sea: Natural or human-induced?» En: *Limnology and Oceanography* 45.3 (2000), págs. 716-726. DOI: <https://doi.org/10.4319/lo.2000.45.3.0716>. eprint: <https://aslopubs.onlinelibrary.wiley.com/doi/pdf/10.4319/lo.2000.45.3.0716>. URL: <https://aslopubs.onlinelibrary.wiley.com/doi/abs/10.4319/lo.2000.45.3.0716>.
- [45] H. Ploug, B. Adam, N. Musat *et al.* «Carbon, nitrogen and O₂ fluxes associated with the cyanobacterium *Nodularia spumigena* in the Baltic Sea». En: *The ISME Journal* 5 (mar. de 2011). DOI: 10.1038/ismej.2011.20. URL: <https://doi.org/10.1038/ismej.2011.20>.
- [46] M. Kahru y R. Elmgren. «Multidecadal time series of satellite-detected accumulations of cyanobacteria in the Baltic Sea». En: *Biogeosciences* 11 (jul. de 2014). DOI: 10.5194/bg-11-3619-2014. URL: <https://doi.org/10.5194/bg-11-3619-2014>.

- [47] P. A. Cox, R. Richer, J. S. Metcalf *et al.* «Cyanobacteria and BMAA exposure from desert dust: A possible link to sporadic ALS among Gulf War veterans». En: *Amyotrophic Lateral Sclerosis* 10.sup2 (2009). PMID: 19929742, págs. 109-117. DOI: 10.3109/17482960903286066. URL: <https://doi.org/10.3109/17482960903286066>.
- [48] T. A. Caller, J. W. Doolin, J. F. Haney *et al.* «A cluster of amyotrophic lateral sclerosis in New Hampshire: A possible role for toxic cyanobacteria blooms». En: *Amyotrophic Lateral Sclerosis* 10.sup2 (2009). PMID: 19929741, págs. 101-108. DOI: 10.3109/17482960903278485. URL: <https://doi.org/10.3109/17482960903278485>.
- [49] D. Main. «Toxic Algal Blooms». En: *Farmnote* (2004). URL: https://web.archive.org/web/20141201235311/http://archive.agric.wa.gov.au/objtwr/imported_assets/content/lwe/water/watq/fn052_2004.pdf.
- [50] L. Sonic. *Floraciones de Cianobacterias: causas, riesgos y tratamiento*. URL: <https://web.archive.org/web/20230103222347/https://www.lgsonic.com/es/cianobacterias/>.
- [51] J. Gregor, B. Maršálek y H. Šípková. «Detection and estimation of potentially toxic cyanobacteria in raw water at the drinking water treatment plant by in vivo fluorescence method». En: *Water Research* 41.1 (2007), págs. 228-234. ISSN: 0043-1354. DOI: <https://doi.org/10.1016/j.watres.2006.08.011>. URL: <https://www.sciencedirect.com/science/article/pii/S0043135406004751>.
- [52] A. J. Ouellette y S. W. Wilhelm. «Toxic cyanobacteria: the evolving molecular toolbox». En: *Frontiers in Ecology and the Environment* 1.7 (2003), págs. 359-366. DOI: [https://doi.org/10.1890/1540-9295\(2003\)001\[0359:TCTEMT\]2.0.CO;2](https://doi.org/10.1890/1540-9295(2003)001[0359:TCTEMT]2.0.CO;2). URL: <https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1890/1540-9295%5C%282003%5C%29001%5C%5B0359%5C%3ATCTEMT%5C%5D2.0.CO%5C%3B2>.
- [53] Y. Wu, J. Zhang, Z. Hou *et al.* «Seasonal Dynamics of Algal Net Primary Production in Response to Phosphorus Input in a Mesotrophic Subtropical Plateau Lake, Southwestern China». En: *Water* 14.5 (2022).
- [54] F. Recknagel, M. French, P. Harkonen *et al.* «Artificial neural network approach for modelling and prediction of algal blooms». En: *Ecological Modelling* 96.1 (1997), págs. 11-28. ISSN: 0304-3800. DOI: [https://doi.org/10.1016/S0304-3800\(96\)00049-X](https://doi.org/10.1016/S0304-3800(96)00049-X). URL: <https://www.sciencedirect.com/science/article/pii/S030438009600049X>.
- [55] B. Wei, N. Sugiura y T. Maekawa. «Use of artificial neural network in the prediction of algal blooms». En: *Water Research* 35.8 (2001), págs. 2022-2028. ISSN: 0043-1354. DOI: [https://doi.org/10.1016/S0043-1354\(00\)00464-4](https://doi.org/10.1016/S0043-1354(00)00464-4). URL: <https://www.sciencedirect.com/science/article/pii/S0043135400004644>.
- [56] H.-S. Yi, S. Park, K.-G. An *et al.* «Algal Bloom Prediction Using Extreme Learning Machine Models at Artificial Weirs in the Nakdong River, Korea». En: *International Journal of Environmental Research and Public Health* 15.10 (2018). ISSN: 1660-4601. DOI: 10.3390/ijerph15102078. URL: <https://www.mdpi.com/1660-4601/15/10/2078>.
- [57] M. Jang, D. Seo, J. Kim *et al.* «Spatiotemporal Algal Bloom Prediction of Geum River, Korea Using the Deep Learning Models in Company with the EFDC Model». En: *Proceedings of the 2020 Summer Simulation Conference (SummerSim)*. 2020. ISBN: 9781713814290.
- [58] S. J. Moe, R.-M. Couture, S. Haande *et al.* «Predicting Lake Quality for the Next Generation: Impacts of Catchment Management and Climatic Factors in a Proba-

- bilistic Model Framework». En: *Water* 11.9 (2019). ISSN: 2073-4441. DOI: 10.3390/w11091767. URL: <https://www.mdpi.com/2073-4441/11/9/1767>.
- [59] A. Rigosi, P. Hanson, D. P. Hamilton *et al.* «Determining the probability of cyanobacterial blooms: the application of Bayesian networks in multiple lake systems». En: *Ecological Applications* 25.1 (2015), págs. 186-199. DOI: <https://doi.org/10.1890/13-1677.1>. eprint: <https://esajournals.onlinelibrary.wiley.com/doi/pdf/10.1890/13-1677.1>. URL: <https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1890/13-1677.1>.
- [60] C. Sivapragasam, N. Muttill, S. Muthukumar *et al.* «Prediction of algal blooms using genetic programming». En: *Marine Pollution Bulletin* 60.10 (2010), págs. 1849-1855. ISSN: 0025-326X. DOI: 10.1016/j.marpolbul.2010.05.020. URL: <https://www.sciencedirect.com/science/article/pii/S0025326X10002213>.
- [61] H. Wilson y F. Recknagel. «Towards a generic artificial neural network model for dynamic predictions of algal abundance in freshwater lakes». En: *Ecological Modelling* 146.1 (2001), págs. 69-84. ISSN: 0304-3800. DOI: [https://doi.org/10.1016/S0304-3800\(01\)00297-6](https://doi.org/10.1016/S0304-3800(01)00297-6). URL: <https://www.sciencedirect.com/science/article/pii/S0304380001002976>.
- [62] H.-M. Oh, C.-Y. Ahn, J.-W. Lee *et al.* «Community patterning and identification of predominant factors in algal bloom in Daechung Reservoir (Korea) using artificial neural networks». En: *Ecological Modelling* 203.1 (2007). Special Issue on Ecological Informatics: Biologically-Inspired Machine Learning, págs. 109-118. ISSN: 0304-3800. DOI: <https://doi.org/10.1016/j.ecolmodel.2006.04.030>. URL: <https://www.sciencedirect.com/science/article/pii/S030438000600562X>.
- [63] B. Vinçon-Leite y C. Casenave. «Modelling eutrophication in lake ecosystems: a review». En: *Science of the Total Environment* 651 (2019), págs. 2985-3001.
- [64] E. Van Sebille, S. M. Griffies, R. Abernathey *et al.* «Lagrangian ocean analysis: Fundamentals and practices». En: *Ocean Modelling* 121 (2018), págs. 49-75.
- [65] D.-K. Kim, H. Cao, K.-S. Jeong *et al.* «Predictive function and rules for population dynamics of *Microcystis aeruginosa* in the regulated Nakdong River (South Korea), discovered by evolutionary algorithms». En: *Ecological Modelling* 203.1 (2007). Special Issue on Ecological Informatics: Biologically-Inspired Machine Learning, págs. 147-156. ISSN: 0304-3800. DOI: <https://doi.org/10.1016/j.ecolmodel.2006.03.040>. URL: <https://www.sciencedirect.com/science/article/pii/S0304380006005667>.
- [66] C. Zhao, N. Shao, S. Yang *et al.* «Predicting cyanobacteria bloom occurrence in lakes and reservoirs before blooms occur». En: *Science of The Total Environment* 670 (2019), págs. 837-848. ISSN: 0048-9697. DOI: <https://doi.org/10.1016/j.scitotenv.2019.03.161>. URL: <https://www.sciencedirect.com/science/article/pii/S004896971931143X>.
- [67] L. Håkanson, A. C. Bryhn y J. K. Hytteborn. «On the issue of limiting nutrient and predictions of cyanobacteria in aquatic systems». En: *Science of The Total Environment* 379.1 (2007), págs. 89-108. ISSN: 0048-9697. DOI: <https://doi.org/10.1016/j.scitotenv.2007.03.009>. URL: <https://www.sciencedirect.com/science/article/pii/S0048969707003087>.
- [68] B. Z. Rousso, E. Bertone, R. Stewart *et al.* «A systematic literature review of forecasting and predictive models for cyanobacteria blooms in freshwater lakes». En: *Water Research* 182 (2020), pág. 115959.

- [69] K. Izydorczyk, M. Tarczynska, T. Jurczak *et al.* «Measurement of phycocyanin fluorescences as an online early warning system for cyanobacteria in reservoir intake water». En: *Environmental Toxicology* 20.4 (2005), págs. 425-430. DOI: <https://doi.org/10.1002/tox.20128>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/tox.20128>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/tox.20128>.
- [70] S.-H. Kim, Y. He, E.-H. Lee *et al.* «Portable Fluorometer for Cyanobacteria Detection». En: *IEEE Sensors Journal* 17.8 (2017), págs. 2377-2384. DOI: 10.1109/JSEN.2017.2672993.
- [71] M. Beutler, K. Wiltshire, M. Arp *et al.* «A reduced model of the fluorescence from the cyanobacterial photosynthetic apparatus designed for the in situ detection of cyanobacteria». En: *Biochimica et Biophysica Acta (BBA) - Bioenergetics* 1604.1 (2003), págs. 33-46. ISSN: 0005-2728. DOI: [https://doi.org/10.1016/S0005-2728\(03\)00022-7](https://doi.org/10.1016/S0005-2728(03)00022-7). URL: <https://www.sciencedirect.com/science/article/pii/S0005272803000227>.
- [72] Deltares. *Delft3D 4*. (Visited on 11/2022). 2022. URL: <https://oss.deltares.nl/web/delft3d>.
- [73] A. Serramiá. «SAIH/SAICA: automatic hydrological information system and automatic water quality information system in the Spanish watersheds». es. En: *Tecno Ambiente (España)* 139 (2004), págs. 15-18.
- [74] Southeast Environmental Research Center. *SERC Water Quality Monitoring Network*. (Visited on 11/2022). 2022. URL: <http://serc.fiu.edu/wqmnetwork/>.
- [75] M. L. Riskin y C. J. Lee. *U.S. Geological Survey National Water Quality Monitoring Network*. English. Inf. téc. Reston, VA: USGS National Water Quality Monitoring Network, 2021, pág. 2. DOI: 10.3133/fs20213019.
- [76] A. W. Wymore. *A mathematical theory of systems engineering: The elements*. New York, United States: John Wiley y Sons, 1967.
- [77] B. P. Zeigler, A. Muzy y E. Kofman. *Theory of modeling and simulation: discrete event & iterative system computational foundations*. Academic press, 2018.
- [78] J. L. Risco-Martín, K. Henares, S. Mittal *et al.* «A Unified Cloud-Enabled Discrete Event Parallel and Distributed Simulation Architecture». En: *Simulation Modelling Practice and Theory* 118 (2022).
- [79] G. Carazo-Barbero, E. Besada-Portas, J. M. Giron-Sierra *et al.* «EA-based ASV Trajectory Planner for Pollution Detection in Lentic Waters». En: *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*. 2021.
- [80] ISCAR. *DEVS BLOOM Simulation*. 2023. URL: <https://github.com/iscar-ucm/devs-bloom-public>.
- [81] B. Herguedas. *DEVS BLOOM Simulation*. 2023. URL: <https://archive.org/details/details/devs-bloom-simulation>.
- [82] S. Ruder. «An overview of gradient descent optimization algorithms». En: (2017). DOI: 10.48550/arXiv.1609.04747. URL: <https://doi.org/10.48550/arXiv.1609.04747>.
- [83] C. Bento. *Stochastic Gradient Descent explained in real life*. 2021. URL: <https://web.archive.org/web/20230115070158/https://towardsdatascience.com/stochastic-gradient-descent-explained-in-real-life-predicting-your-pizzas-cooking-time-b7639d5e6a32?gi=d7ad10210a94>.

- [84] M. Carey y J. Ramsay. «Fast stable parameter estimation for linear dynamical systems». En: *Computational Statistics & Data Analysis* 156 (2021), pág. 107124. ISSN: 0167-9473. DOI: <https://doi.org/10.1016/j.csda.2020.107124>. URL: <https://www.sciencedirect.com/science/article/pii/S0167947320302152>.