

ANÁLISIS DE LA CALIDAD DE AGUAS
EN EMBALSES MEDIANTE SIMULACIÓN CON EEMS



TRABAJO FIN DE GRADO
CURSO 2023-2024

AUTOR

FERNANDO SÁNCHEZ GARCÍA

MARCOS PADILLA ALONSO

DIRECTOR

JOSÉ LUIS RISCO MARTÍN

GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

TÍTULO EN ESPAÑOL **ANÁLISIS DE LA CALIDAD DE AGUAS
EN EMBALSES MEDIANTE SIMULACIÓN CON EEMS**
TITLE IN ENGLISH **ANALYSIS OF WATER QUALITY IN
RESERVOIRS THROUGH SIMULATION WITH EEMS**

TRABAJO DE FIN DE GRADO EN INGENIERÍA INFORMÁTICA

AUTOR

**FERNANDO SÁNCHEZ GARCÍA
MARCOS PADILLA ALONSO**

DIRECTOR

JOSÉ LUIS RISCO MARTÍN

CONVOCATORIA: JUNIO 2024

GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

27 DE JUNIO DE 2024

DEDICATORIA

A todos los familiares y amigos de ambos participantes, y al tutor del TFG José Luis por su compromiso.

AGRADECIMIENTOS

A José Luis nuestro tutor, por el tiempo empleado en ayudarnos en lo que fuese necesario. A nuestros familiares por el apoyo brindado durante el desarrollo del proyecto. Y a la Universidad Complutense de Madrid, por el soporte ofrecido durante estos años.

RESUMEN

Análisis de la calidad de aguas en embalses mediante simulación con EEMS.

El presente trabajo desarrolla un modelo mediante el software de EEMS con el objetivo de analizar la calidad del agua del embalse de Santillana con el propósito de poder anticiparse a la aparición de toxinas. Este trabajo servirá de aportación para el proyecto de investigación de DEVS-BLOOM.

Palabras clave

Modelo, afloración, monitorizar, EEMS, Grid+, malla, cianobacterias, embalse, USV.

ABSTRACT

Analysis of water quality in reservoirs through simulation with EEMS.

The present work develops a model using EEMS software with the objective of analyzing the water quality of the Santillana reservoir with the purpose of being able to anticipate the appearance of toxins. This work will serve as a contribution to the DEVS-BLOOM research project.

Keywords

Model, bloom, monitoring, EEMS, Grid+, mesh, model, cyanobacteria, reservoir, USV.

ÍNDICE DE CONTENIDOS

Capítulo 1 - Introducción	1
1.1 Motivación	1
1.2 Objetivos	2
1.3 Plan de trabajo	3
1.4 Estructura del documento	6
1.5 Separación del trabajo	6
Capítulo 2 - Contexto del TFG	8
2.1 Proyecto principal	8
2.2 Conexión entre EEMS y DEVS-BLOOM	14
2.3 Variables de la simulación	14
Capítulo 3 - Desarrollo del trabajo	20
3.1 Descarga de las herramientas y licencia	20
3.2 Proceso de mallado con la herramienta Grid +	22
3.3 Proceso de modelado con EEMS	24
3.4 Ejecución y rendimiento	27
Capítulo 4 - Casos de uso y planes de contingencia	32
1. Proceso de mallado en el embalse de Santillana	32
1.1 Problema - Limitación en la exportación de archivos en Grid+	33
1.1 Solución - Extracción de los diferentes formatos de archivo disponibles	33
2. Proceso de Modelado en el embalse de Santillana	34
2.1 Problema - Limitación en la importación de archivos gráficos de Grid+	34
2.1.1 Solución - Cambio de muestra a lagos más pequeños	34
2.2 Problema - Limitación en el número de celdas de la malla	34
2.2.1 Solución - Disminución de celdas	34
2.3 Problema - Incompatibilidad de importación de modelos en EEMS	35
2.3.1 Solución - Descarte de EEMS. Creación de archivo de simulación desde el inicio. Uso de archivos KML en simulaciones	35
2.4 Problema - Limitaciones en la inserción de datos geoespaciales en variables de simulación	37
2.4.1 Solución - Descarte de la inserción de datos. Trasposición de lago Washington a la ubicación del embalse de Santillana	37
1. Reemplazar archivo Washington	37
2. Integración de variables geoespaciales en un archivo nuevo	37

2.5 Problema - Incoherencia en las dimensiones de las variables	38
2.5.1 Solución - Uso de MATLAB	38
2.5.2 Solución - Tratar la diferencia en las dimensiones de las variables	39
3. Proceso de recreación de la simulación del embalse de Santillana	39
3.1 Problema - Fallos en la representación 3D	40
3.1.1 Solución - Redimensión de variables relevantes para la representación 3D	40
3.1.2 Solución - Creación de una nueva malla	41
4.1 Diagnóstico de las limitaciones y obstáculos	42
Capítulo 5 - Ejecución del framework completo	44
Capítulo 6 - Conclusión y trabajo futuro	48
6.1 Revisión de los objetivos	48
6.2 Revisiones finales y trabajo futuro	49
Capítulo 7 - Conclusions and future work	51
7.1 Review of objectives	51
7.2 Final revision and future work	52
Capítulo 8 - Contribuciones Personales	54
Marcos Padilla Alonso	54
Fernando Sánchez García	54

ÍNDICE DE FIGURAS

Figura 1: Esquema 1 proyecto SMART-BLOOMS	10
Figura 2: Esquema 2 proyecto SMART-BLOOMS	11
Figura 3: Predicción de algas	12
Figura 4: Scenary Configuration Editor	13
Figura 5: Aplicación de monitorización del USB	13
Figura 6: Licencia de instalación	21
Figura 7: Autenticación EEMS	21
Figura 8 : License Manager	22
Figura 9: Autenticación de clave	22
Figura 10: Extracción de malla	24
Figura 11: Time Zone	25
Figura 12: Condiciones Iniciales	25
Figura 13: Ventana Boundary Conditions	26
Figura 14: Ventana Timing	26
Figura 15: Malla de 34080 celdas del embalse de Santillana	33
Figura 16: Formatos de exportación permitidos	33
Figura 17: Error 'Overflow' de EEMS	33
Figura 18: Malla del lago del Campillo	34
Figura 19: Malla <300 celdas del embalse de Santillana	35
Figura 20: Modelo de simulación ofrecido por EMS	36
Figura 21: Salida del script de corrección del lago Washington	36
Figura 22: Variables 'lon' y 'lat' de Santillana .nc	38
Figura 23: Variables 'lon' y 'lat' de Washington .nc	38
Figura 24: Variables de Santillana .nc en MATLAB	39
Figura 25: Salida del script de corrección de Santillana	39

Figura 26: Representación 3D 1	40
Figura 27: Representación 3D 2	40
Figura 28: Representación 3D 3	40
Figura 29: Acumulación de celdas	40
Figura 30: Salida del script de corrección de Santillana 1450 celdas	41
Figura 31: Representación 3D 1	42
Figura 32: Representación 3D 2	42
Figura 33-Representación variables 1	45
Figura 34-Representación variables 2	46
Figura 35-Representación variables 3	46
Figura 36-Representación variables 4	47
Figura 37-Representación variables 5	47

ÍNDICE DE TABLAS

Tabla 1. Variables archivo netCDF4	19
Tabla 2. Dispositivo HP Pavilion Laptop 16-a0	28
Tabla 3. Dispositivo Hp laptop 14s	28

Capítulo 1 - Introducción

Este Trabajo de Fin de Grado (TFG), tiene como propósito la evaluación del agua del embalse de Santillana, a través de simulaciones realizadas con el software de EEMS. El objetivo es obtener una medición de la cantidad de algas, nitratos, oxígenos, etc., en diferentes etapas del año. Con esto, se busca la anticipación a la aparición de toxinas, lo cual afectaría posteriormente en el uso recreativo de estas aguas. También se utilizará Python para el manejo de datos. Por último, el resultado de los datos de la simulación serán contrastados con datos reales. Estos datos reales son enviados a tiempo real por un drone acuático, el cual está dotado con una serie de sensores que se encargan de recoger distintas cualidades y parámetros del agua por donde pasa.

Este TFG es una contribución al proyecto de investigación SMART-BLOOMS, el cual, busca mejorar la gestión de los afloramientos de cianobacterias (HCB) a través del empleo de tecnologías de Edge Computing y Digital Twins.

1.1 Motivación

Un buen suministro de agua potable es necesario para el bienestar de las comunidades. Las floraciones de algas y cianobacterias nocivas (HCB¹), que se producen en aguas continentales y marítimas, suponen una amenaza para los entornos naturales al producir toxinas que afectan a la salud humana y animal (Vincent WF.)[\[3\]](#). El método tradicional de evaluar las floraciones de algas o FAN, consistía en una persona en su barca recogiendo muestras del agua en diferentes puntos del lago para su posterior estudio y, ocasionalmente, mediante instrumentos automáticos que adquieren información de lugares fijos. Estos procedimientos no proporcionan datos con la resolución espacial y temporal deseable para prever la formación de FAN. El tiempo que suponía recolectar, llevar las muestras para estudio y las conclusiones del mismo, se reflejaba a su vez en cambios en el nacimiento y desarrollo de estas algas quitándole validez al estudio.

Por lo tanto, se necesitan renovar herramientas y tecnologías para detectar, analizar y responder eficazmente a las FAN que amenazan la calidad del agua, con mejor calidad en los resultados. Es esencial hoy en día, cuando el suministro mundial de agua está sometido a una enorme presión debido al cambio climático, la sobreexplotación y la contaminación.

¹ Del inglés Harmful Cyanobacterial Blooms

El fenómeno del calentamiento global y el cambio climático ha emergido como un desafío ambiental global de grandes proporciones. El calentamiento global, la sobreexplotación y la contaminación que ello supone, es un gran desafío al que nos enfrentamos en estos últimos años. Entre los muchos campos que afecta directamente, el nivel del agua de los lagos de España es de lo que más se ha visto comprometido recientemente, en el pasado 2022-2023. A esto se le suman las pocas precipitaciones y la gran demanda de este recurso natural. El agua, es un bien necesario que escasea, y la amenaza que supone el desarrollo de estas algas para la calidad del agua, ha suscitado la necesidad de mejorar las estrategias de cuidado y prevención de la proliferación de estas. El desarrollo de estas algas de una manera descontrolada afecta directamente varias zonas del lago que abastecen de agua a pueblos, comprometiendo altamente el proceso de depuración de agua.

Es un peligro para la sanidad, por tanto este sistema nos ayudará a mitigar su efecto prediciendo ese avance de algas y cómo afectará ese avance al agua a los pueblos colindantes. Se mostrará el lago de una manera visual con el fin de poder ver el resultado de una manera gráfica y que las autoridades y los responsables locales de cada zona puedan tomar decisiones en base a ello.

1.2 Objetivos

El objetivo general de este Trabajo de Fin de Grado (TFG) es desarrollar y aplicar un modelo de simulación de eventos discretos utilizando el software [EEMS](#) (EE Modeling System), para analizar la calidad del agua en los embalses de Guajaraz y Santillana, así como en la laguna del Campillo. Este modelo permitirá simular la dinámica de las cianobacterias y otros parámetros relevantes como nitratos y oxígeno, con el fin de predecir la aparición de floraciones algales nocivas y su impacto potencial en el uso recreativo y de consumo del agua. El TFG se enmarca dentro del proyecto de investigación SMART-BLOOMS, que busca mejorar la gestión de las floraciones de cianobacterias mediante la adopción de tecnologías de Edge Computing y Digital Twins.

Los objetivos específicos son los siguientes:

1. Desarrollo de un modelo de simulación para embalses y lagunas:

- Diseñar y construir un modelo computacional en EEMS que represente con precisión la dinámica de las cianobacterias y otros parámetros de calidad del agua en los cuerpos de agua seleccionados.
- Integrar el modelo con programación en Python para la automatización de procesos y análisis de datos.

2. Generación y análisis de datos sintéticos:

- Utilizar el modelo de simulación para generar datos sintéticos que reflejen el comportamiento de las cianobacterias y otros parámetros de calidad del agua bajo diferentes condiciones ambientales y temporales.
- Aplicar técnicas estadísticas para el análisis de los datos sintéticos generados, con el objetivo de identificar patrones y tendencias.

3. Contribución al proyecto SMART-BLOOMS:

- Aportar al proyecto SMART-BLOOMS un caso de estudio detallado sobre la aplicabilidad de la simulación de eventos discretos en la gestión de la calidad del agua y la predicción de afloramiento de cianobacterias.
- Explorar la integración del modelo de simulación desarrollado en el TFG con las tecnologías de Edge Computing y en DEVS-BLOOM propuestas en el proyecto SMART-BLOOMS.

4. Difusión de resultados y conocimientos:

- Documentar cuidadosamente el proceso de modelado y simulación, así como los resultados obtenidos, para su posterior utilización por los miembros del proyecto.
- Preparar material multimedia para apoyar la documentación escrita sobre modelado y simulación de eventos discretos aplicados a la gestión de recursos hídricos.

Al alcanzar estos objetivos, el TFG contribuirá significativamente al campo de la gestión sostenible del agua en embalses y lagunas, ofreciendo herramientas valiosas para la toma de decisiones informadas en el ámbito de la salud pública y la conservación ecológica.

1.3 Plan de trabajo

Aquí describiremos estructuradamente los pasos que hemos seguido para alcanzar los objetivos que hemos planteado en secciones anteriores.

Introducción

Vamos a presentar brevemente el plan de trabajo que se ha seguido para poder llevar a cabo el desarrollo del proyecto.

Se han tenido una serie de reuniones periódicas (cada dos semanas) con el tutor del TFG para ponernos en contexto del trabajo dentro del proyecto en el que se encuentra. Una vez realizada esta parte, las reuniones se empezaron a realizar también con compañeros del TFG de 3D para poder coordinarnos de manera más eficiente. Además de estas reuniones con

el tutor, hemos tenido reuniones conjuntas de manera externa con los miembros del TFG de 3D, para la solución de algunos problemas surgidos durante el desarrollo de ambos proyectos.

El trabajo dentro de nuestro proyecto se ha realizado de manera paralela entre ambos alumnos, centrándonos ambos en las mismas tareas. Se ha llevado a cabo una metodología iterativa en la que íbamos refinando y mejorando cada parte ya mencionada. Ha habido una comunicación activa durante el proceso que se ha llevado a cabo mediante videoconferencias y mensajería escrita.

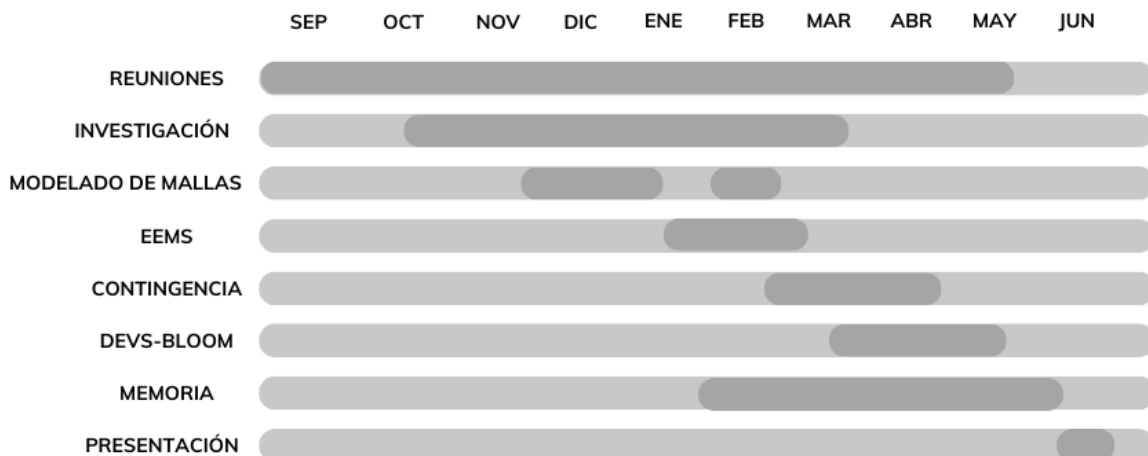
Metodología

Para lograr los objetivos que ya hemos mencionado hemos seguido un proceso de investigación mediante internet, accediendo a documentación y vídeos de ejemplo disponibles en la página de [EEMS](#).

Para poder realizar un modelo de simulación, primero diseñamos la malla desde el software de Grid+. En base a esta malla y a través del software de EEMS, generamos una serie de datos sintéticos del embalse de Santillana, para su posterior análisis. Como resultado tenemos un archivo de simulación, con datos precisos del embalse apto para ser utilizado por el software del proyecto denominado DEVS-BLOOM. Este software da como resultado un conjunto de archivos csv que representan algunas predicciones como el movimiento de los drones acuáticos, la dinámica de las floraciones de cianobacterias entre otras cosas.

Por último, se ha documentado cuidadosamente todo este proceso de forma más detallada en esta memoria.

Cronograma



Recursos necesarios

- Herramienta Grid+, para realizar el modelado del embalse.
- Herramienta de EEMS, para generar un modelo sobre la malla obtenida, obteniendo un archivo netCDF4.
- Python, para el desarrollo de scripts para análisis de datos, consumos, representación de datos, etc.
- Un entorno en el que poder ejecutar código Python para poder ejecutar el software de DEVS-BLOOM.

Riesgos y contingencias

Uno de los mayores riesgos que puedes enfrentar en un proyecto de este tipo es la falta de la licencia del software de EEMS. Para ello, pusimos en marcha un plan de contingencia que se basa en un desplazamiento de un modelo disponible en la demo de EEMS como el lago de Washington, a la posición de nuestro embalse, adaptando las variables y las dimensiones necesarias, para poder llevar a cabo esta transición y cumplir con nuestro propósito. Esto se puede ver de forma más detallada en el [Capítulo 4](#).

1.4 Estructura del documento

El documento se divide en cinco capítulos. El capítulo [1](#) introduce el proyecto SMART-BLOOMS y cual es su cometido en el beneficio de la calidad del agua. A su vez, contiene un listado de objetivos que se buscan cumplir con la realización de este TFG.

En el capítulo [2](#) se profundiza más sobre el proyecto en el que participamos al realizar este TFG. Es necesario situarnos correctamente en el contexto y entender el orden del flujo de procesos que contiene SMART-BLOOMS para entender nuestro cometido en este proyecto. Detallamos la importancia de la automatización en el proceso de extracción de algas frente a su método manual y el impacto que tienen estas algas sobre la salud de la humanidad. También indagamos en la conexión de nuestro TFG con el proyecto. De qué forma vamos a proporcionarle una entrada a SMART-BLOOMS con la herramienta EEMS para generar una simulación subordinada a esta eliminación de algas. Por último exponemos el tipo de archivo y las variables usadas para facilitar la comprensión de un archivo netCDF4, usado en este TFG.

Seguidamente, en el capítulo [3](#) se delimita la metodología de trabajo usada para crear un modelo de simulación con EEMS. Su correcto uso, guiado por secciones, con el fin de determinar un marco de trabajo en el cual se trabajará. Por otro lado se estudian las técnicas que se destinan para el análisis del rendimiento y la ejecución del sistema completo de modelado de simulación, estudiando su aplicabilidad en los dispositivos usados.

Posteriormente, en el capítulo 4 seguimos los pasos del anterior capítulo extrapolados al embalse de Santillana. Explicamos caminos alternativos que no vimos obligados a explorar debido a ciertas limitaciones que nos encontramos en nuestro proceso de modelado de la simulación. Estos problemas vienen causados por una falta de licencia completa del sistema EEMS. También incluye una breve conclusión extraída del estudio realizado de la simulación sin el uso de EEMS.

En el capítulo 5, se recogen los resultados de la ejecución de nuestro archivo de simulación, y se lleva a cabo la visualización 2D de los resultados obtenidos tras dicha ejecución.

Por último, en el capítulo 6 se revisan los objetivos marcados al comienzo del trabajo, y se resumen las principales conclusiones extraídas de este estudio. A su vez, se proponen también en diferentes vías de estudio que permitan continuar con la investigación de modelos sin necesidad de sistemas de modelado.

1.5 Separación del trabajo

En este punto explicaremos qué plan de trabajo hemos seguido para desarrollar el TFG. Después de la presentación de nuestro tutor acerca del proyecto global al que pertenece este TFG, nos proporcionó una serie de documentos relacionados. Entre ellos, se incluían varios programas de Python, el documento oficial del proyecto global SMART-BLOOM y un enlace a la página oficial de EEMS.

No hemos tenido ninguna división de trabajo concreta que podamos demostrar. Los dos estuvimos intentando bajo sus propios medios explotar al máximo las funciones que nos ofrecía la versión demo de EEMS. En cuanto a la creación de las diversas mallas usadas en el trabajo, al ser un trabajo laborioso cada uno avanzaba desde el punto que lo había dejado el anterior compañero. Fernando se encargó de los programas y la obtención del rendimiento y el coste de ejecución tanto de EEMS como de Grid+. Marcos se encargó de modificar e intentar ajustar los archivos de simulación con tal de que funcionase dentro de la representación 3D del TFG de nuestros compañeros. Los dos invertimos tiempo en investigar formas de suplir la acción de EEMS en nuestro proyecto y en crear programas en Python, Matlab que imite la salida correcta. Posteriormente está explicado en nuestro [plan de contingencia](#).

Cada dos semanas se han realizado reuniones con nuestro tutor del TFG José Luis Risco Martín para observar nuestro avance y guiarnos en cada paso y decisión en el TFG. Como posteriormente explicaremos, tuvimos problemas en la realización de simulaciones, por lo que fueron necesarias estas reuniones para reubicar el enfoque que le hemos dado a este TFG. También mantuvimos una conexión constante con nuestros compañeros del TFG de la

representación 3D del embalse. Les mandamos nuestros borradores de las simulaciones de los embalses para comprobar la corrección de nuestras muestras y a su vez de su programa de representación 3D. Fué crucial para entender correctamente cuál era el comportamiento de ciertas variables en la simulación y su cometido dentro de la simulación 3D.

Capítulo 2 - Contexto del TFG

2.1 Proyecto principal

En las aguas tanto marinas como interiores, se producen las ya mencionadas FAN, las cuales producen toxinas que afectan de forma directa a la vida humana, representando una amenaza para el medio ambiente. La exposición a estas cianobacterias de las que hablamos, puede tener graves efectos en la salud humana y animal, lo que se podría ver reflejado en graves pérdidas económicas que afectasen a las comunidades locales.

Estas afloraciones suelen producirse en una variedad de ecosistemas de agua dulce, como embalses, lagos y ríos. (Vincent WF.)[\[3\]](#) Durante la última década, se ha detectado que su intensidad y frecuencia de aparición han aumentado a nivel mundial, lo que supone un serio problema. Esto se debe principalmente a la vulnerabilidad actual de los recursos hídricos, el calentamiento global, el crecimiento demográfico, etc. De ahí que las autoridades sanitarias y medioambientales necesiten mejores datos de rápida disponibilidad para aplicar oportunamente herramientas digitales de decisión que les ayuden a minimizar los riesgos asociados a la exposición al HCB. Uno de los ejemplos más sonados sobre una de estas afloraciones es el del Toledo (Ohio, EE.UU), el cual supuso el suministro de agua no potable a más de 400.000 personas durante varios días. (Schmale DG, Ault AP, Saad W et al.)[\[1\]](#)

Este proyecto surge debido a las arcaicas técnicas utilizadas para controlar las calidades del agua de los embalses en España. Ya que estas técnicas no tienen el alcance suficiente, lo que se verá reflejado en una mayor seguridad en el control de calidad del agua, aspecto muy relevante en la salud pública del país. Estas técnicas se basaban en la recolección de muestras de aguas de forma manual por parte de personal especializado, el cual se desplazaba en barco por el embalse que se deseaba analizar y tomaba muestras en puntos estratégicos, para poder analizarlas posteriormente en laboratorios. En algunos casos, estas muestras eran tomadas por instrumentos automáticos que se encontraban colocados de manera fija en el propio embalse, los cuales estaban situados de forma estratégica, basándose en los lugares donde era más probable la aparición de las ya mencionadas FAN. El problema, es que estas técnicas los datos que nos proporcionan no nos permiten obtener una resolución tanto espacial como temporal para poder predecir posteriormente la aparición de nuevas afloraciones. El retraso que sufren los resultados analíticos y la limitada capacidad para interpretar el escenario actual, reduce la capacidad de reacción (predicción, prevención y mitigación) de las autoridades responsables de la distribución de agua potable y sus usos recreativos(Meriluoto J, Spoof L and Codd GA)[\[4\]](#). Por lo tanto, es esencial crear nuevos sistemas económicos de monitoreo y detección temprana que puedan prever y anticipar la

densidad de células de cianobacterias o algas. Estos sistemas deben ser capaces de predecir cuándo y dónde se forman las FAN y producen toxinas. Esto ayudaría a los gestores y autoridades del agua a orientar sus políticas y proteger la salud pública mediante la implementación de sistemas de alerta temprana eficaces. (Risco-Martín, Esteban, Chacón)[\[2\]](#)

Por todo lo mencionado anteriormente, surge la idea del proyecto de SMART-BLOOM, cuyo objetivo es permitir el monitoreo en tiempo real de por ejemplo un embalse, para llevar a cabo la predicción de riesgos de una nueva afloración. Proporciona un enfoque factible para garantizar una detección de alto rendimiento de peligros a través de la implementación de tecnologías Digital Twins en un entorno dinámico. El uso de Digital Twins sigue la tendencia actual hacia una transición digital. Los sistemas complejos se desarrollan primero en un plano virtual y progresivamente van siendo reemplazados por elementos del mundo real. La transición digital propuesta por SMART BLOOMS, crea una réplica de los recursos hídricos para optimizar el despliegue de elementos de análisis de HCB. Como pueden ser los diferentes sensores, cámaras ,vehículos no tripulados incluso el método de extracción tradicional. Digital Twins se alimenta inicialmente de datos históricos y posteriormente por datos reales recogidos en la capa edge, que veremos posteriormente. (Risco-Martín, Esteban, Chacón)[\[2\]](#)

El proyecto divide el sistema de gestión de HCB's en tres capas clásicas de IoT: edge,cloud y fog.

La *capa edge* incluye todos los dispositivos conectados a Internet y que generan datos. En nuestro proyecto se incluyen dispositivos fronterizos entre la capa fog, como sensores ambientales, cámaras colocadas en posiciones estacionarias y USV (del inglés Unmanned Surface Vehicle). Reúne y procesa datos de manera local y los envía a la siguiente capa para su posterior procesamiento.(Risco-Martín, Esteban, Chacón)[\[2\]](#)

La *capa fog* es una capa intermedia entre la capa edge y la capa cloud. Esta capa incluye dispositivos informáticos y capacidades de almacenamiento para realizar procesamiento básico y análisis de datos. Es la responsable de procesar datos a tiempo real y resumir los datos que necesitan ser mandados a la capa cloud para su posterior análisis.(Risco-Martín, Esteban, Chacón)[\[2\]](#)

La *capa cloud* incluye servidores en la nube y centros de datos que pueden almacenar y procesar grandes cantidades de datos. La capa cloud actúa en tareas complejas de análisis de datos y aprendizaje automático que requieren una importante potencia informática y gran capacidad de almacenamiento.(Risco-Martín, Esteban, Chacón)[\[2\]](#)

La masa de agua es monitorizada en la capa edge por un conjunto de sensores, incluidos los que se encuentran a bordo de embarcaciones automatizadas, denominadas vehículos de superficie acuática no tripulados o del inglés Unmanned Surface Vehicles (USV),

que envían datos continuamente al servidor en el terreno más cercano GCS del inglés Ground Control Station en la capa fog. (Risco-Martín, Segundo Esteban, Chacón)[2]

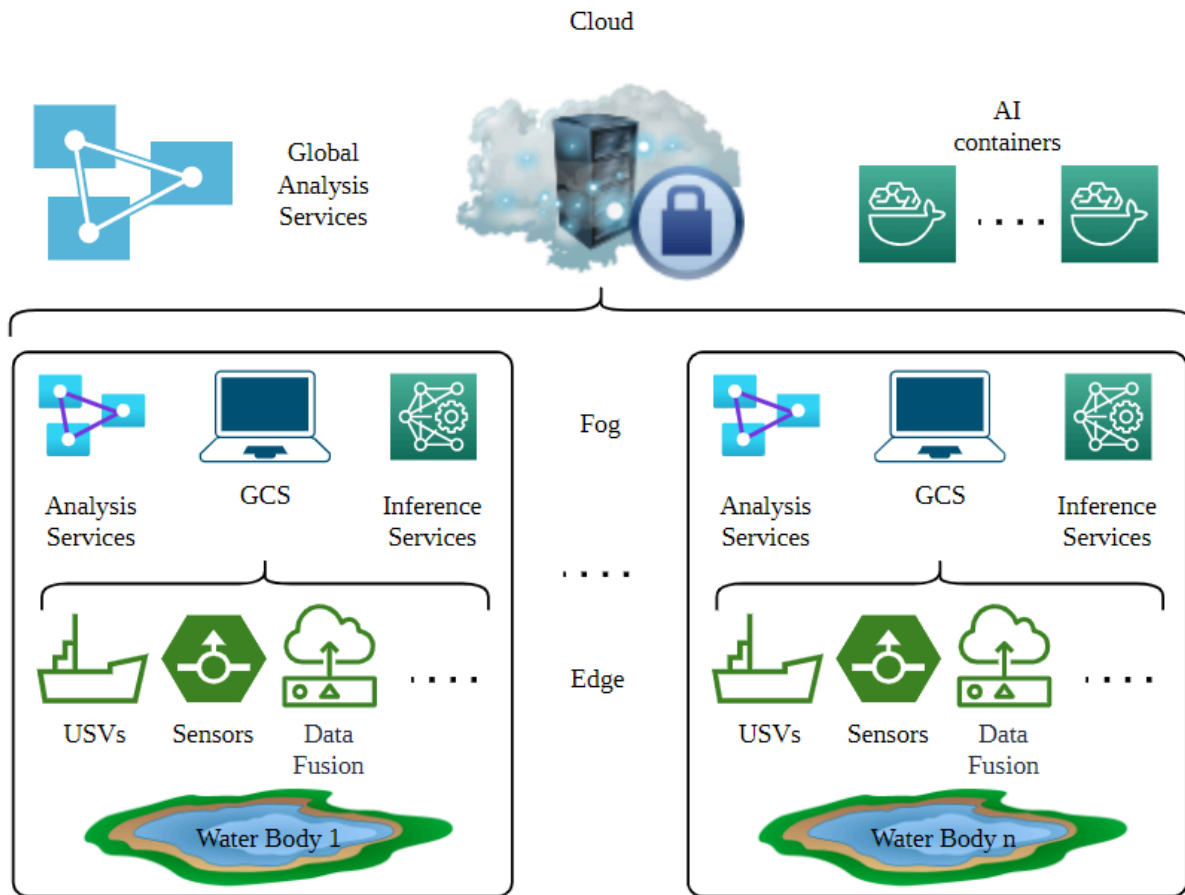


Figura 1: Esquema 1 proyecto SMART-BLOOMS

Aquí, hay programas que desempeñan la función de sensor virtual, que imita el comportamiento del sensor real, introduciendo los correspondientes retrasos, ruido, errores de saturación, envejecimiento, etc.. Allí, los expertos pueden analizar los datos, ejecutar algunos modelos, pruebas o planificar las trayectorias de los USV y contrastar información. Puede predecir la aparición de la floración y guiar automáticamente a los USV a la zona de interés, o tomar mediciones (Se puede ver en la unidad USV's planner out en la [Figura 1](#). El marco soporta escalabilidad horizontal, pudiendo agregar más cuerpos de agua con su debido soporte en la cloud, donde las autoridades pueden comparar diferentes informes y tomar decisiones de alto nivel. (Risco-Martín, Esteban, Chacón)[2]

La función que desempeñamos en este proyecto es crear el escenario de monitorización para DEVS-BLOOM. Es relevante porque nos muestra como DEVS-BLOOM es

usado para predecir las siguientes localizaciones de las HAB y automáticamente controlar el USV para verificar su aparición. En este caso aporta un dataset de datos sintéticos de todo el cuerpo de agua en el que se va a desarrollar posteriormente las simulaciones. Por consiguiente tiene que ser un escenario exacto.

Va a suponer la entrada de datos que incluyen la velocidad del agua, temperatura del agua, densidades de oxígeno, nitratos y concentraciones de alga al sistema de sensores ya mencionados. Nuestro escenario, tiene en la edge layer un USV que debe monitorear el agua y transmitir datos a la fog layer y la cloud layer. El USV, está equipado con varios sensores y unidades (ver sensores en la unidad USV en la [Figura 1](#)). Algunos de ellos toman datos del cuerpo de agua para monitorear continuamente el estado del florecimiento de las algas y alimentan el modelo de inferencia.

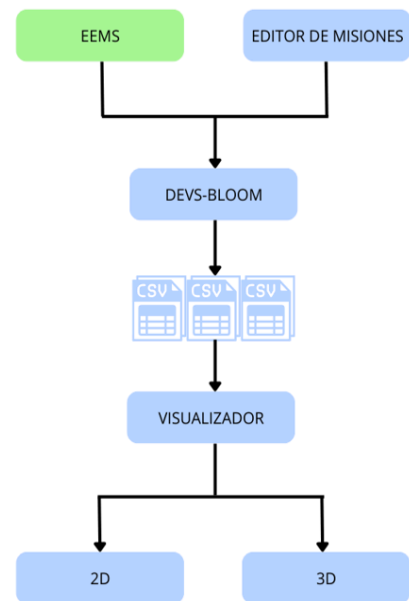


Figura 2: Esquema 2 proyecto SMART-BLOOMS

- **Sensor de temperatura:** se encarga de medir la temperatura del agua.
- **Unidad de energía:** incluye paneles solares, cargadores y baterías encargadas de recargar las baterías del barco cuando recibe radiación solar.
- **Medidor del flujo de agua:** mide la velocidad y la dirección del agua con respecto del barco.
- **Unidad de posición:** nos ayuda a saber la posición y la velocidad del barco.
- **Sonda de oxígeno disuelto:** encargada de medir la densidad del oxígeno disuelto en el agua. Si hay altos niveles de oxígeno, puede haber una floración de algas que desprende oxígeno por fotosíntesis.
- **Sonda de nitrógeno:** mide la densidad del nitrógeno disuelto nitratos en el agua. El nitrato es el principal alimento de las algas por lo tanto, nos da una señal para predecir el crecimiento de la floración.

Creamos una simulación precisa del lago y exportamos archivo de la simulación para obtener a través de él, datos de salida. (Risco-Martín, Esteban, Chacón) [\[2\]](#)

Los datos de salida de la simulación son necesarios para acceder y gestionar el sistema desde el punto de vista de un usuario, u operador. La manipulación de datos se simplifica notoriamente al utilizarse archivos fuente en formato csv, los cuales son mencionados en el

[Capítulo 5](#). Permite configurar las condiciones iniciales de la simulación, obteniendo así resultados basados en las condiciones iniciales según el criterio profesional de un técnico que vaya a utilizar el sistema. Teniendo en cuenta estas condiciones iniciales, se establecería a su vez una posición inicial de USV posteriormente. Esto se denomina iniciar una misión. El proceso de iniciar misiones empieza desde un estudio de las predicciones. El proyecto tiene creada una interfaz que recibe los csv y representa de una forma amigable y concisa el comportamiento de las simulaciones reflejadas en los archivos. Como primer paso, se atenderá el comportamiento de las algas en el mapa de predicción, semejante al mostrado a continuación. (Risco-Martín, Esteban, Andrade) [\[5\]](#)

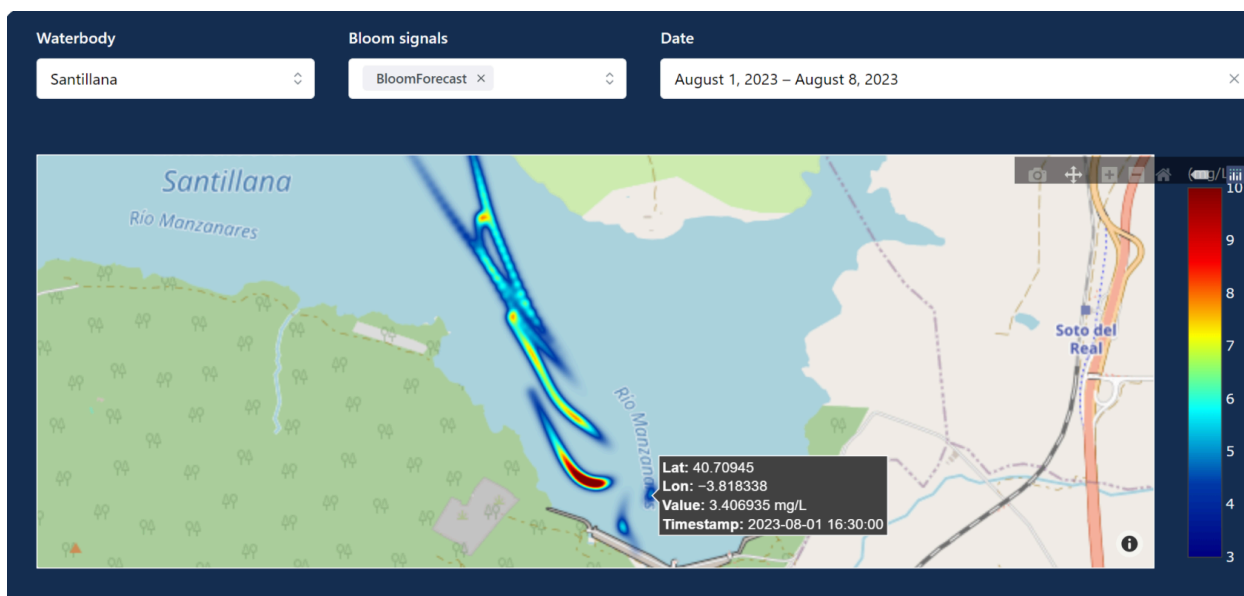


Figura 3: Predicción de algas

Se puede observar en el ejemplo que durante ese periodo, algunos días presentan altas densidades de algas, que además se desplazan a otras zonas. Por lo tanto, el especialista debe activar el sistema de vigilancia. En este punto se incluye el barco en las coordenadas de máxima concentración para que gestione las FAN de ese lugar ya estudiado. Tenemos ya el escenario de actuación del USV. Los usuarios pueden establecer el despliegue estratégico de los equipos de vigilancia, garantizar una cobertura completa ellos. Existe otro editor, se puede ver en la [Figura 4](#), donde se desarrolla el funcionamiento del USV. El editor facilita la creación de escenarios complejos al permitir a los usuarios definir las áreas de trabajo de los USV. Proporciona flexibilidad para adaptarse a las condiciones ambientales cambiantes o incorporar nuevos elementos al escenario. (Risco-Martín, Esteban, Andrade) [\[5\]](#)

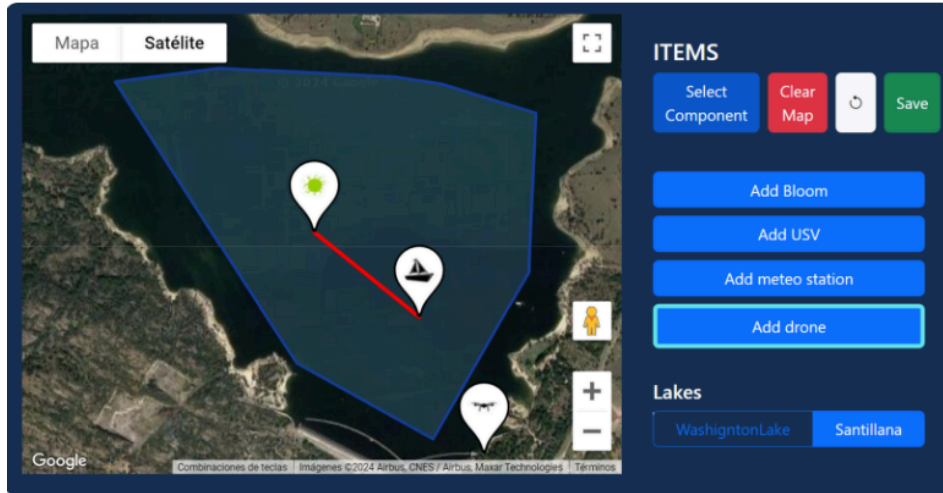


Figura 4: Scenary Configuration Editor (SCE)

Al activarse el sistema, el USV persigue las FAN. La aplicación recoge todas las trayectorias descritas por los objetos USV, mostrando la variación entre las trayectorias reales y las de referencia suministradas por la simulación DEVS-BLOOM. Contiene una ventana deslizante para ver su avance a lo largo del tiempo. (Risco-Martín, Esteban, Andrade) [5]

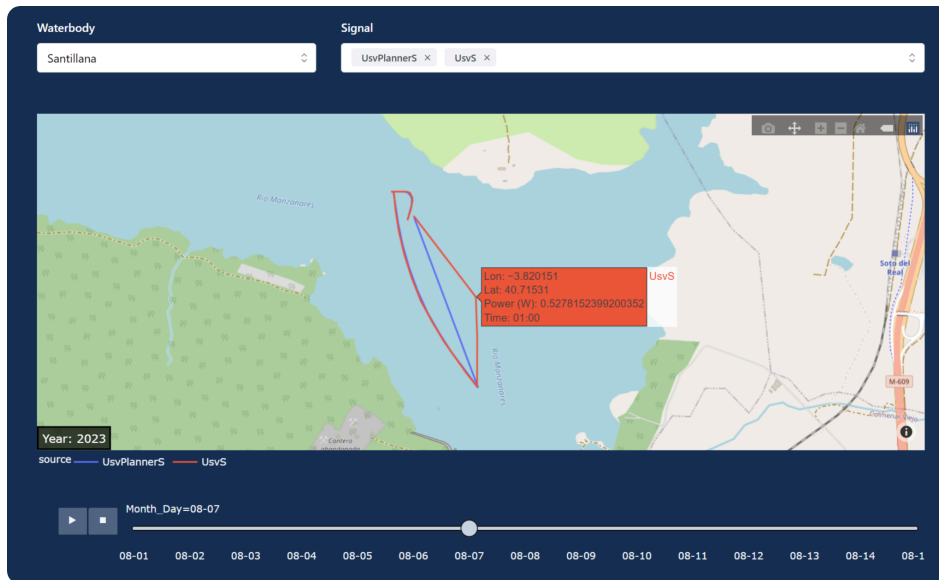


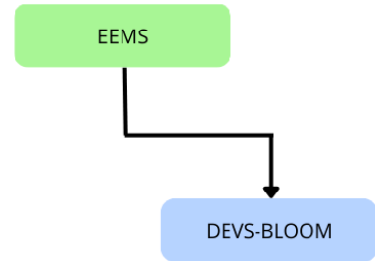
Figura 5: Aplicación de monitorización del USV

Se recogen los resultados de los datos obtenidos, y se analizan sacando gráficas comparativas y guardando los niveles de la concentración de las FAN en la interfaz. Se registran y se ven cuando han superado el nivel máximo de concentración definido, junto a qué

situación y en qué lugar. Esto resultará en avisos a las autoridades o planes de depuración en esas zonas . (Risco-Martín, Esteban, Andrade) [5]

2.2 Conexión entre EEMS y DEVS-BLOOM

Una vez puestos en contexto sobre el proyecto, vamos a usar la simulación obtenida como una entrada a un sistema. La herramienta que se emplea es EEMS. Se le describe como paquete completo de modelado para hidrodinámica, transporte de sedimentos, destino y transporte de productos químicos, y eutrofización. Gracias al modelo simulado del lago de Santillana obtenido por el trabajo realizado en EEMS, tenemos todas las características, variables y factores que afectan al comportamiento de las algas y del agua impresos en un modelo. El modelo de EEMS nos devuelve un fichero [netCDF](#) .



Este fichero será la entrada para el programa de Python del DEVS-BLOOM. Dicho software recibe esta simulación y guarda como salida varios .csv con información relevante. Recaltar entre ellos un csv llamado "BloomModel.csv" que incluye por donde predice que va a haber floración de algas, con su posición, su tamaño y su concentración. "UsvModel.csv" que incluye la posición del barco junto a el porcentaje de batería que va persiguiendo las algas por el lago en formato .csv. De esta forma obtenemos resultados numéricos de dos simulaciones completadas, una de el lago y otra en función de la otra simulación. Con estos datos se podrá representar de manera visual en 3D estas simulaciones en otro TFG en paralelo al nuestro.

2.3 Variables de la simulación

El resultado de la simulación obtenida del lago Santillana, es un archivo netCDF4. Los archivos con la extensión nc generalmente se refieren a archivos de datos Net Common Data Format (netCDF). Este formato se suele utilizar para almacenar grandes conjuntos de datos multidimensionales. En nuestro caso, esta característica de este tipo de archivos nos conviene. Guardamos la simulación en una matriz con la forma y dimensión del lago el cual cada celda de la matriz representa una porción de agua del lago con las características propias de esa porción. Estos datos están estructurados en arrays lo que facilita el trabajo con ellos en programas como Python y R. Para entender un poco más este tipo de archivos vamos a realizar una explicación básica de cómo se organizan estos. Un archivo netCDF contiene un conjunto de dimensiones, como puede ser la altitud, longitud, tiempo etc. Estas sirven para organizar los datos dentro del archivo. Por otro lado, tenemos las variables, que son los datos en sí y que se

almacenan en el archivo. Cada variable va a tener asociada a ella una o más dimensiones. También debemos tener en cuenta que las variables a su vez tienen asociadas atributos, que nos proporcionan más información de estas (unidades de los datos, nombre largo de la variable, etc.). Esta organización de los archivos nos proporciona una característica muy interesante la cual nos permite filtrar datos específicos pudiendo ignorar otros, y nos facilita la visualización de momentos temporales o características de la variable. Vamos a pasar a comentar la estructura del archivo "Washington-1m-2008-09_UGRID.nc". Para entender completamente las variables utilizadas en el archivo con extensión nc, necesitamos definir las dimensiones correspondientes a cada una.

– 'tau' (En el archivo, 'time'): Tiempo actual del momento-

– 'row': Fila de la malla.

– 'col': Columna de la malla. Al coger el par de dimensiones row-col, creamos una variable llamada CELL.

– 'NODE': Capa de la malla.

– 'FOUR': Identificador del vértice de cada celda (1-4).

– 'ÁLG': Identificador del Algae.

Nombre	Nombre completo	Descripción	Dimensiones	Unidades
Nv	-	Nodos que rodean el elemento. Posición de los vértices (4) que encuadran el nodo al que pertenece	Sin dimensiones	Sin unidades
lon	Longitude	Nodal longitude con coordenadas	'[row,col, NODE]'	Grados este
lat	Latitude	Nodal latitude con coordenadas	'[row,col, NODE]'	Grados norte
lonc	zonal longitude	Zonal longitude con coordenadas	'[row,col]'	Grados este
latc	zonal latitude	Zonal latitude con coordenadas	'[row,col]'	Grados norte
lon_bnds	Longitude boundaries	Límite de longitudes con coordenadas	'[row,col,cnr]'	Grados este

lat_bnds	Latitude boundaries	Límite de latitud con coordenadas	'[row,col,cnr]'	Grados norte
layers	Layer	Nº de capas verticales con coordenadas	'[row,col]'	Sin unidades
bottom_layer	Bottom_layer	Índice de capas inferiores con coordenadas	'[row,col]'	Sin unidades
sigma	Sigma	Coordenada sigma del océano en el punto medio de la capa	'[row,col, KC]'	Sigma
layer_thickness	layer_thickness	Grosor de la capa de agua.	'[NODE, KC]'	Fracción
time	Time	Con coordenadas	'[Time]'	Días desde el día referenciada
bottom	Bottom bathymetry	Índice de capas inferiores	'[row,col]'	m
CUV	CUV	Rotación de la celda	'[NODE, FOUR]'	Sin unidades
RSSBC	RSSBC	Ponderación del borde de velocidad.		Sin unidades
SEDDIA	SEDDIA	Tamaño del grano de los sedimentos	'[FOUR, LCM]'	Micrómetros
BELV	BELV	Elevación del fondo.	'[time,CELL]'	Sin unidades
HP	HP	Profundidad del agua	'[Time, NODE (latc lonc)]'	m

WSEL	Water surface elevation	Elevación de la superficie del agua (time dependent)	'[Time, NODE (latc lonc)]'	Grados este
U	Eastward Water Velocity	Velocidad del agua hacia el este (time dependent)	'[TIME,KC, NODE]'	m/s
V	Northward Water Velocity	Velocidad del agua hacia el norte (time dependent)	'[TIME,KC, NODE]'	m/s
W	Upward Water Velocity	Velocidad del agua del viento ascendente (time dependent)	'[TIME,KC, NODE]'	m/s
delt	Delt	Time Step de la ejecución	'[TIME]'	Time
tota_shear	Tota_shear	Esfuerzo al que está sometido el lecho	'[TIME,NODE]'	N/m2
RPOC	RPOC	Partículas refractarias de carbono orgánico	'[TIME, KC,NODE]'	mg/L
LPOC	LPOC	Partículas lábiles de carbono orgánico	'[TIME, KC, NODE]'	mg/L
DOC	DOC	Partículas disueltas de carbono orgánico	'[TIME, KC, NODE]'	mg/L
RPOP	RPOP	Partículas refractarias de fósforo	'[TIME, KC, NODE]'	mg/L
LPOP	LPOP	Partículas lábiles de fósforo	'[TIME, KC, NODE]'	mg/L
DOP	DOP	Partículas disueltas de fósforo	'[TIME, KC, NODE]'	mg/L
P4D	P4D	Fosfato total	'[TIME, KC, NODE]'	mg/L
RPON	RPON	Partículas refractarias de nitrógeno orgánico	'[TIME, KC,NODE]'	mg/L
LPON	LPON	Partículas lábiles de nitrógeno orgánico	'[TIME, KC, NODE]'	mg/L
DON	DON	Partículas disueltas de nitrógeno orgánico	'[TIME, KC, NODE]'	mg/L

NHX	NHX	Partículas disueltas de nitrógeno amoniacal	'[TIME, KC, NODE]'	mg/L
ÑOX	ÑOX	Partículas disueltas de nitrato de nitrógeno	'[TIME, KC, NODE]'	mg/L
SUU	SUU	Partículas de sílice biogénico	'[TIME, KC, NODE]'	mg/L
SAA	SAA	Partículas disueltas de sílice	'[TIME, KC, NODE]'	mg/L
COD	COD	Demanda de oxígeno químico	'[TIME, KC, NODE]'	mg/L
DOX	DOX	Oxígeno químico disuelto	'[TIME, KC, NODE]'	mg/L
TAM	Tam	Total de metales activos	'[TIME, KC, NODE]'	mg/L
FCB	FCB	Bacterias coliformes provenientes de materia fecal	'[TIME, KC, NODE]'	mg/L
ALG	Alga	Algas	'[ALG, TIME, KC, NODE]'	mg/L
Wx	Eastward Wind Velocity	Velocidad del viento hacia el este con coordenadas (time dependent).	'[TIME, NODE]'	m/s
Wy	Northward Wind Velocity	Velocidad del viento hacia el norte con coordenadas (time dependent)	'[TIME, NODE]'	m/s
Vx	Eastward flow velocity	Velocidad del flujo del viento hacia el este con coordenadas (time dependent)	'[TIME, NODE]'	m/s
Vy	Northward flow velocity	Velocidad del flujo del viento hacia el norte con coordenadas (time dependent)	'[TIME, NODE]'	m/s
Vz	Upward flow velocity	Velocidad del flujo del viento ascendente con coordenadas (time dependent)	'[TIME, NODE]'	m/s
Temp	Temperature	Temperatura del agua	'[TIME, KC, NODE]'	Gr° celsius

air_pressure	air_pressure	Presión barométrica del aire	'[TIME, NODE]'	hPa
--------------	--------------	------------------------------	----------------	-----

Tabla 1. Variables archivo netCDF4

Capítulo 3 - Desarrollo del trabajo

Aquí se describe el plan de trabajo a seguir para la consecución de los objetivos descritos en el apartado anterior. Principalmente nos enfocaremos en el uso del simulador EEMS y sus configuraciones para poder obtener una simulación precisa. [EEMS](#) es un motor de modelado de aguas superficiales que incluye componentes hidrodinámicos, contaminantes de sedimentos diseñado para simular sistemas acuáticos en una, dos y tres dimensiones. Ofrece una herramienta de generación de cuadrículas en modelos (Grid+) con la que podemos dar valor matricial a cada celda y poder ser más precisos en las simulaciones en 2D y 3D.

A la hora de empezar a modelar con este sistema, el orden de nuestras interacciones con el mismo se ha visto afectado por que es un sistema con una suscripción de pago. Dado el sobrecoste de la licencia (13000 €) hemos trabajado con la versión demos de EEMS. Se ha intentado obtener una licencia para el proyecto SMART-BLOOMS, pero desde octubre de 2023 hasta mediados de mayo de 2024 ha sido imposible su obtención. Hasta que no conseguimos la versión completa, nuestro modelaje ha sido dificultado por la falta de funcionalidades que nos aportaba la versión de muestra de este sistema.

3.1 Descarga de las herramientas y licencia

Lo primero que debemos realizar para poder llevar a cabo un modelado de un lago, embalse, río o cualquier cuerpo de agua; es descargar las herramientas necesarias para llevar a cabo este proceso. Para ello, habrá que acceder a la página oficial de EEMS y dirigirse a la parte inferior de esta hasta ver un botón que diga 'Probar EEMS'. Después de pulsar el botón, será redirigido a otra ventana donde se explicarán las limitaciones que tiene la demo gratuita de este software. Al final de esta página, verá otro botón con el texto de 'Descargar'. Una vez se haya pulsado dicho botón, se abrirá otra ventana, desde la que se podrá ver las distintas opciones de paquetes a descargar. Una vez elegida la opción que más se adapte a nuestras necesidades, se descargará un archivo comprimido en nuestro ordenador.

Después de realizar los pasos anteriores habrá que descomprimir dicho archivo y ejecutar el instalador de EEMS, que será el archivo con la extensión exe; el cual ejecutará el asistente de instalación de EEMS, como se muestra en la [Figura 6](#).



Figura 6 : Licencia de instalación

A continuación, habrá que aceptar los términos de uso de EEMS; rellenar un nombre de usuario y de entidad; seleccionar la carpeta destino donde serán instaladas las distintas herramientas de EEMS y los modelos de ejemplo. Posteriormente elegiremos el tipo de opción de configuración, en este caso habrá que elegir entre 'Completa' o 'Personalizada' y una vez terminado esto habrá que pulsar en el botón de finalizar.

La primera vez que abramos EEMS se mostrará una ventana de registro, como la de la [Figura 7](#), mediante la cual tendrá que ser rellenada con sus datos.

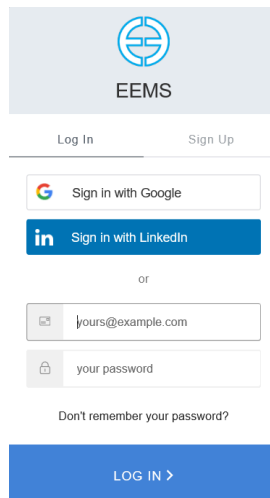


Figura 7 : Autenticación EEMS

Una vez se haya hecho el 'login' ya estaremos viendo la interfaz de EEMS. Lo primero que hará será dirigirse a la pestaña de 'Help' situada en la barra de herramientas y desde ahí hacer click en la ventana de 'License Management', como se muestra en la [Figura 8](#).

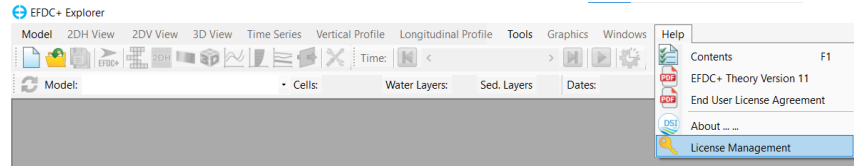


Figura 8: License Manager

Se abrirá otra pestaña, en la cual podremos observar una opción llamada 'Demo Mode License', que para empezar, será la que habrá que activar. En caso de que más tarde queramos activar una licencia de pago, desde esta misma ventana tras pinchar en el botón 'Logout' aparecerá otra ventana con la forma de la [Figura 9](#).



Figura 9 : Autenticación de clave

Desde aquí, habrá que pulsar en el botón de 'Full', lo que abrirá un formulario de activación. Desde este formulario se seleccionará la opción de 'Activar con código de licencia' y una vez introducido el código de licencia y pulsado el botón de 'Activar', EEMS activará de forma automática la versión de la licencia que hay detectado en base al código.

3.2 Proceso de mallado con la herramienta Grid +

Lo primero que se realizará, será abrir la aplicación en nuestro ordenador. Una vez realizado esto, se observará que esta ofrece una imagen de un satélite donde mediante un movimiento manual por medio del ratón se podrá recorrer cualquier punto del planeta tierra.

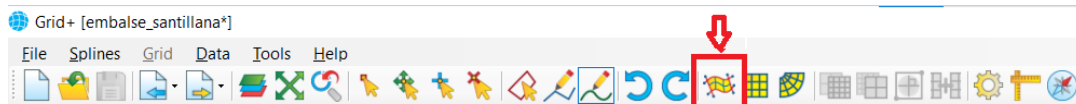
La primera fase para realizar un mallado con esta herramienta, va a ser situarse en el lugar donde quiere ser realizado. Para ello, es recomendable buscar previamente las coordenadas del lugar en Internet, pues esto ayudará a que el lugar sea encontrado de forma

más rápida. Otra cosa a tener en cuenta, es que se pueden observar las coordenadas sobre las que se encuentra el puntero del ratón en la parte inferior derecha de la pantalla.

Una vez se tenga situado el lugar que se desea mallar, habrá que realizar 'splines'. Para ello, habrá que pulsar el botón marcado en la siguiente figura, y con este realizar un espacio de cuadrilátero adaptado a la forma del lugar que deseamos mallar.



Una vez hayan sido cerrados los cuatro lados del cuadrilátero se nos activará una opción que permitirá crear una malla que esté comprendida dentro del cuadrilátero.



Una vez pulsado este botón, se abrirá una ventana, desde la que se indicará el número de celdas que se desean en la dirección 'I', y el número de celdas que se desean en la dirección 'J', obteniendo como resultado una malla de tamaño I x J.

La idea para obtener una malla lo más refinada posible, es ir creando numerosas mallas de pequeño tamaño, lo que nos llevará al siguiente paso, la unión de dos mallas.

Para llevar a cabo la unión de dos mallas, se deben considerar una serie de aspectos para evitar errores. Por el lateral coincidente de la malla, el número de celdas de cada una debe de ser el mismo, ya que si no se mostrará un error al intentar unir las dos mallas. Una vez se haya tenido esto en cuenta, cuando las dos mallas que se desean unir, las cuales deben estar juntas, se habilitará un botón, el cual se muestra en la siguiente figura, que al ser pulsado realizará la unión de manera automática.



Otro botón bastante utilizado es el de 'Refine Grid', el cual se muestra en la siguiente imagen. Este botón, es bastante útil para evitar el error que se ha mencionado anteriormente, cuando se intentan unir dos mallas de distintos tamaños. Cuando este botón es pulsado se muestra una ventana en la cual se puede elegir el grado de refinamiento de las celdas, tanto en la dirección 'I' como en la 'J'. Si por ejemplo se elige un grado de refinamiento de 2 en la dirección 'I', las celdas de la dirección 'I' se dividirán a la mitad, pasando a ser el doble (en caso de que fuese una malla de 4x4, pasaría a ser una de 8x4).



Tras conseguir todo el terreno deseado cubierto por una malla única, llega el último paso, que consiste en la exportación de esta para poder utilizarla posteriormente en EEMS y en base a ella crear un modelo de simulación.

Para esto, habrá que dar click derecho sobre el nombre de la malla que se quiere exportar. Los nombres tanto de los 'splines', como de las mallas aparecen a la izquierda de la pantalla, en una lista. Una vez se haya hecho click, se mostrará un desplegable, donde se tendrá que pinchar en la opción de 'Export...' y seleccionar el formato deseado para exportar la malla, como se muestra en la [Figura 10 : Extracción de malla](#).

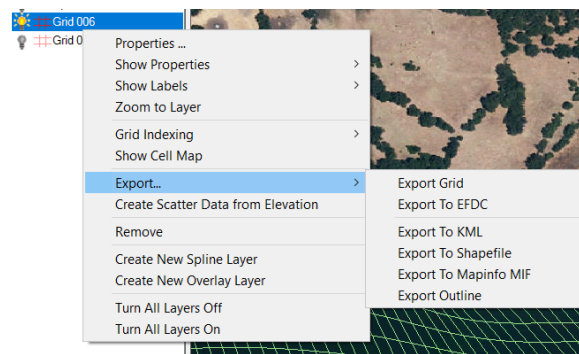


Figura 10 : Extracción de malla

3.3 Proceso de modelado con EEMS

El siguiente paso a dar vendría de la mano del EEMS, para diseñar y ejecutar un modelo de simulación sobre la malla obtenida anteriormente.

Lo primero de todo, sería abrir la aplicación de EEMS. Una vez abierta esta, habrá que crear un nuevo modelo. Esto se puede realizar pinchando en el botón que tiene una hoja en blanco o desde la pestaña de 'Model' y dentro de esta 'New Model'. (Ambas opciones se pueden observar en la [Figura 8](#)).

Una vez creado el nuevo modelo, se abrirá un nueva ventana, donde se tendrá que seguir una de estos dos caminos para cargar la malla obtenida anteriormente.

Uno de los caminos, sería importar la malla completa generada en el Grid+. Para esto, habría que pulsar la opción de 'Import Grid from Files' y luego dar al botón de browse e importar un malla que esté en formato cvl o gpc.

El otro camino sería generar un 'Uniform Grid' con las dimensiones necesarias, en el cual se tendrían que ajustar parámetros como el tamaño de la celda, el número de celdas, etc. y sobre esta malla uniforme creada, importar encima el contorno de la malla creada anteriormente en formato p2d. Para ello, hay que dirigirse a la parte situada debajo de la configuración de los parámetros de la malla y en la casilla de 'Bounding Polygons' hacer click derecho e importar el archivo p2d. Después de esto, habría que establecer la zona horaria y pulsar el botón de 'Remove Dry', como se muestra en la siguiente imagen [Figura: Time Zone](#).

Para finalizar este paso, habría que pulsar en el botón de 'Ok' y ya tendremos la malla cargada en un modelo de EEMS.

El siguiente paso a realizar, será el establecimiento de las condiciones iniciales de nuestro modelo. Para ello, en la ventana de 'Model Control', la cual se creó al cargar la malla en el modelo, habrá que dirigirse a la pestaña de 'Initial Conditions'. Dentro de esta pestaña podremos asignar algunas condiciones como la batimetría, que se puede cargar con archivo .dat; la rugosidad del fondo o la temperatura del agua, como se muestra en la siguiente imagen [Figura: Condiciones Iniciales](#).

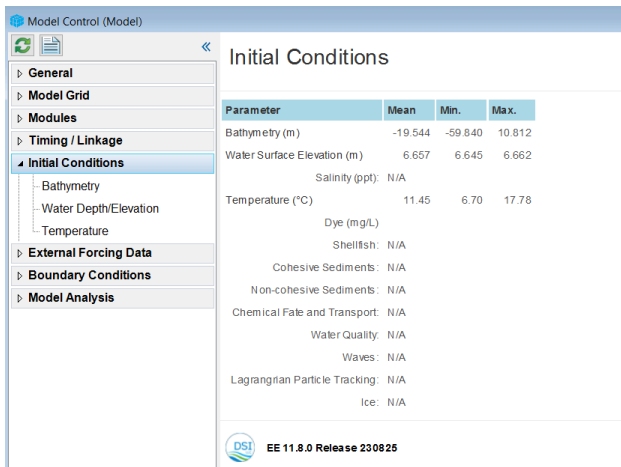


Figura 12: Condiciones Iniciales

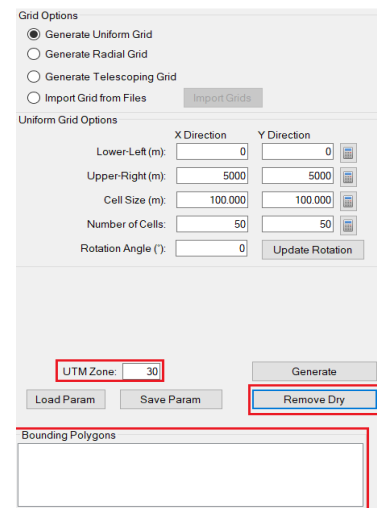


Figura 11 : Time Zone

Una vez definidas las condiciones iniciales del modelo, se tendrán que definir las condiciones de contorno del mismo. Dentro de este apartado, lo más importante es definir los flujos de entrada y salida de agua. Se realiza mediante el establecimiento de series temporales las cuales se asignan a la celda o celdas que se correspondan con el flujo que se quiere definir. Para realizar esta configuración hay que dirigirse a la pestaña de 'Boundary Conditions', como se muestra en la [Figura 13](#).

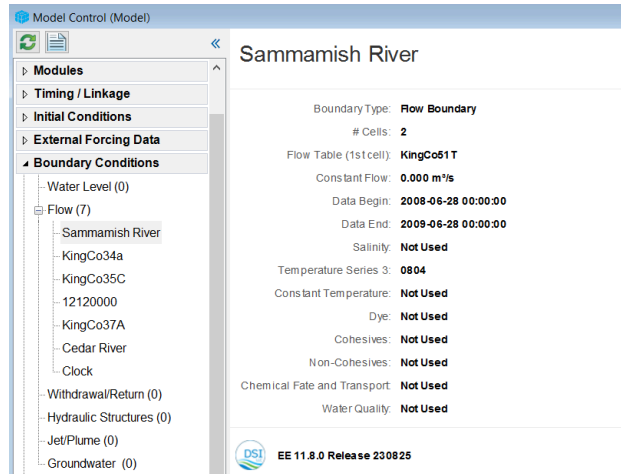


Figura 13 : Ventana Boundary Conditions

Una vez completados estos pasos, se habrá completado prácticamente el modelo. El siguiente paso que se deberá realizar es la configuración del tiempo de simulación del modelo. Para ello, habrá que dirigirse a la ventana de 'Timing/Linkage' y dentro de esta pulsar en la pestaña de 'Timing', como se muestra en la [Figura 14](#), y dentro de esta se tendrá que definir la fecha de inicio y de fin de la simulación, teniendo en cuenta las series temporales definidas anteriormente en las condiciones de entorno.

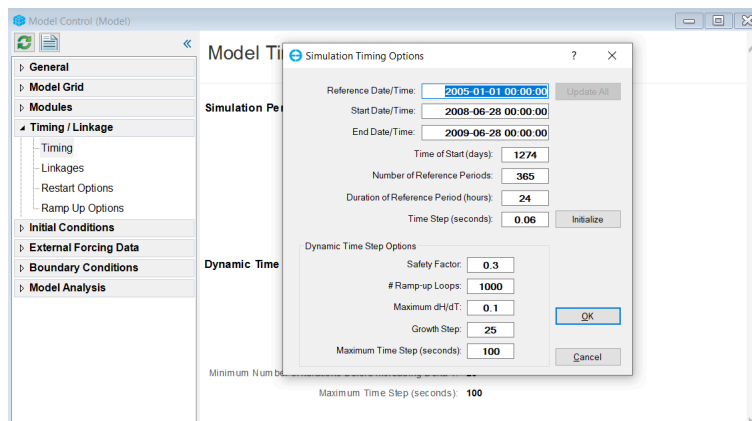
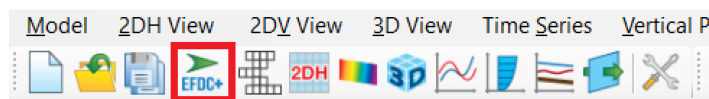


Figura 14: Ventana Timing

Después desde la ventana de 'Model Grid' pinchando en 'Grid', habrá que ingresar las capas verticales.

Para terminar, se pasará a llevar a cabo la ejecución del modelo. Para esto, se tendrá que pulsar en el botón de la siguiente imagen, el cual nos abrirá otra pestaña.



En esta nueva pestaña, podremos configurar algunos parámetros de la ejecución del modelo como el número de hilos de procesamiento que se van a utilizar, el intervalo de salida de los resultados, etc. Tras ajustar dichos parámetros, se pulsará el botón de “Run EFDC+” y el modelo comenzará a ejecutarse en una ventana aparte.

Para finalizar esta parte, una vez ejecutado el modelo, podremos ver los resultados de la simulación, para esto, deberemos pulsar el botón de “Refresh Output”. Los resultados de la simulación podrán ser visualizados desde la pestaña de Vista 2DH, 2DV o 3D.

3.4 Ejecución y rendimiento

En esta sección se mostrarán los resultados tanto del porcentaje del uso de la CPU como el porcentaje de uso de la memoria obtenidos, mientras se ejecutan las distintas herramientas utilizadas en el proyecto mencionadas con anterioridad.

Para llevar a cabo este procedimiento de manera más cómoda se ha desarrollado un script en Python con el cual se ha logrado automatizar el Administrador de tareas de Windows. Así, se ha podido extraer la información requerida de este, seleccionando el nombre del proceso del que se necesitaba extraer la información. Además, desde el script se puede controlar el periodo de tiempo durante el cuál se recopila información, así como el tiempo de intervalo de medición.

Por último, cabe recalcar que esta información que se obtiene es volcada en fichero de tipo csv para su posterior visualización gráfica mediante el uso de la librería Matplotlib.

Características de los dispositivos usados

Dispositivo 1: HP Pavilion Gaming Laptop 16-a0

Características:

Componentes	Descripciones
-------------	---------------

Procesador	Intel Core I7-10750H @ 2.60GHz, 2592 MHz, 6 procesadores principales, 12 procesadores lógicos
Memoria RAM	Instalada 16 GB (15,8 GB usable)
Tarjeta gráfica	NVIDIA GeForce GTX 1650 Ti
Sistema Operativo	Windows 10 Pro 64 bits

Tabla 2: Dispositivo HP Pavilion Gaming Laptop 16-a0

Dispositivo 2: Hp laptop 14s

Características:

Componentes	Descripciones
Procesador	11th Gen Intel Core I7-1165G7 @ 2.80GHz, 2701 MHz, 4 procesadores principales, 8 procesadores lógicos.
Memoria RAM	8,00 GB (7,65 GB usable)
Tarjeta gráfica	Intel Iris Xe Graphics
Sistema Operativo	Microsoft Windows 11 Home

Tabla 3: Dispositivo Hp laptop 14s

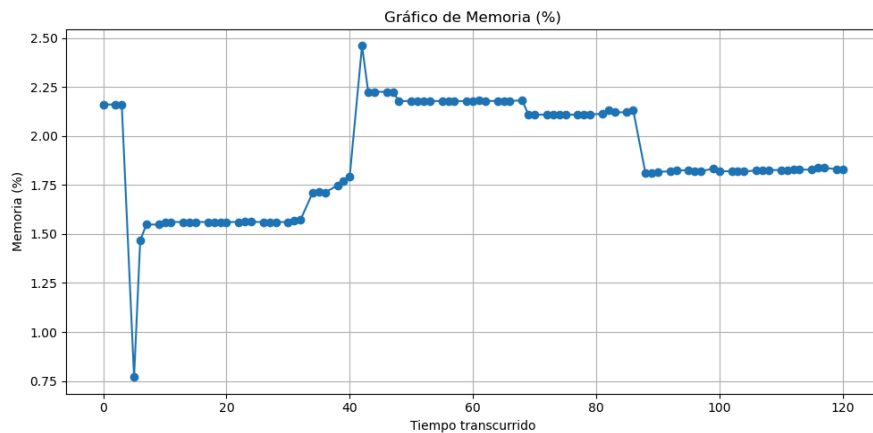
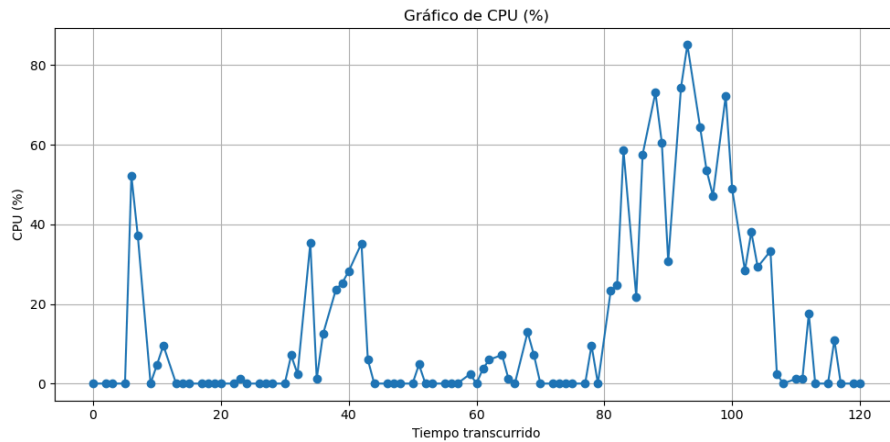
Monitorización del Grid+

Para mayor entendimiento de las gráficas que se van a mostrar a continuación, debemos de tener en cuenta lo que se ha realizado en este software durante la automatización del Administrador de tareas. Lo primero que se debe saber es que la medición que se ha realizado ha tenido una duración de 120 segundos. Los primeros 10-15 segundos se han destinado a abrir el software Grid+. Posteriormente, se ha mantenido la aplicación en reposo hasta el segundo 30 aproximadamente. Después de esto, se ha procedido a cargar un

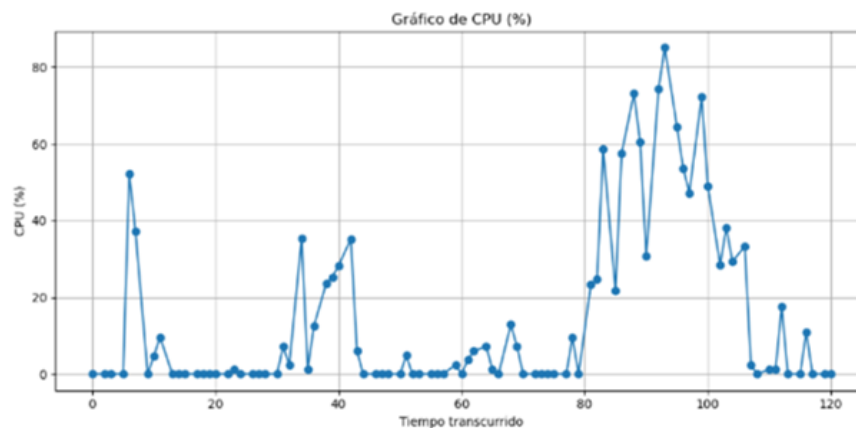
proyecto ya creado anteriormente, lo que ha ocupado hasta el segundo 45 aproximadamente.

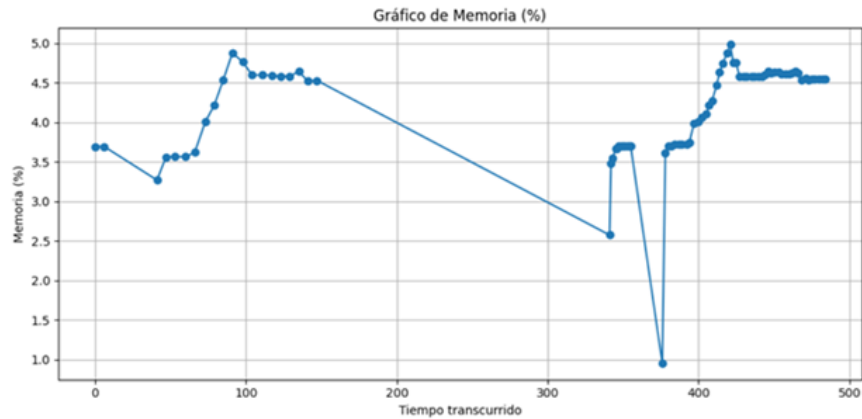
De nuevo, hemos vuelto a dejar la aplicación en reposo hasta que se ha cumplido el minuto. Una vez pasado este tiempo, hemos procedido a borrar el grid que abarcaba todo el pantano, lo que ha llevado unos 5 segundos. Hemos vuelto a dejar la aplicación en reposo hasta el minuto y 20 segundos aproximadamente. Para finalizar hemos creado un grid de 20 por 5, acción que nos ha ocupado hasta el minuto y 50 segundos aproximadamente. Los últimos 10 segundos la aplicación se ha mantenido en reposo.

Dispositivo 1



Dispositivo 2



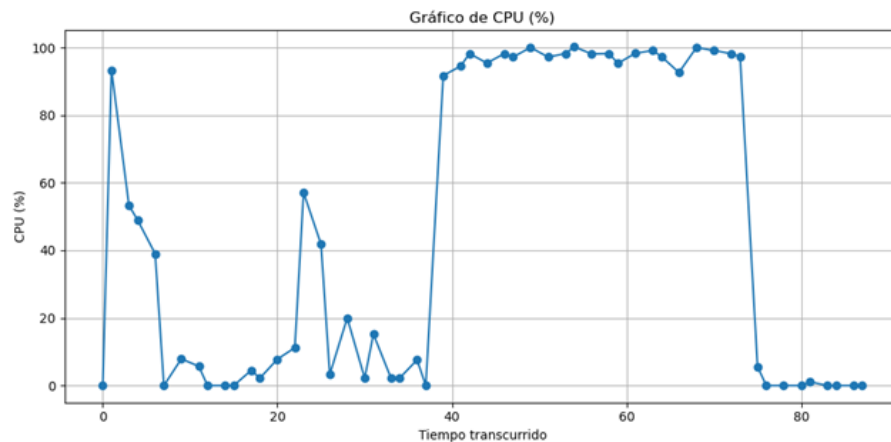


Monitorización del EE11.8

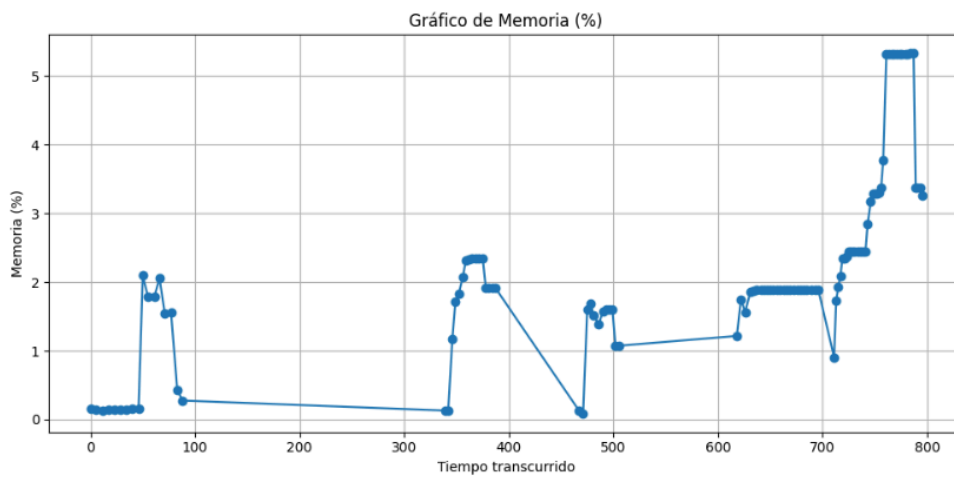
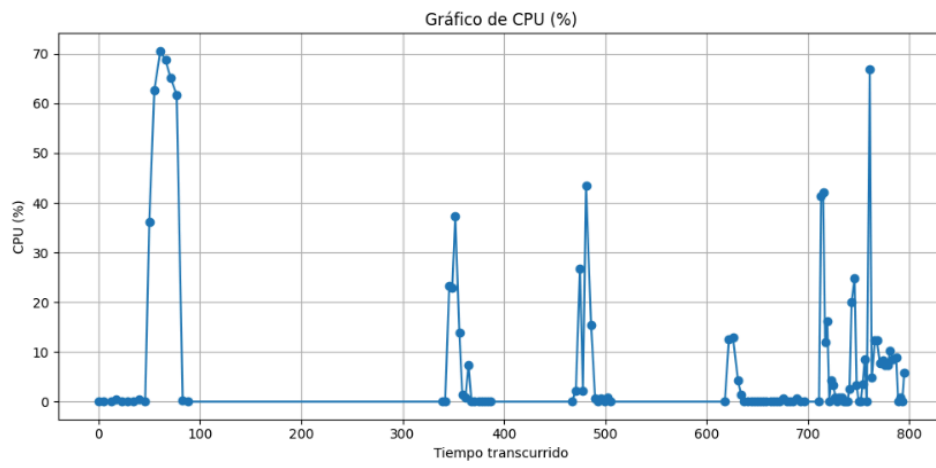
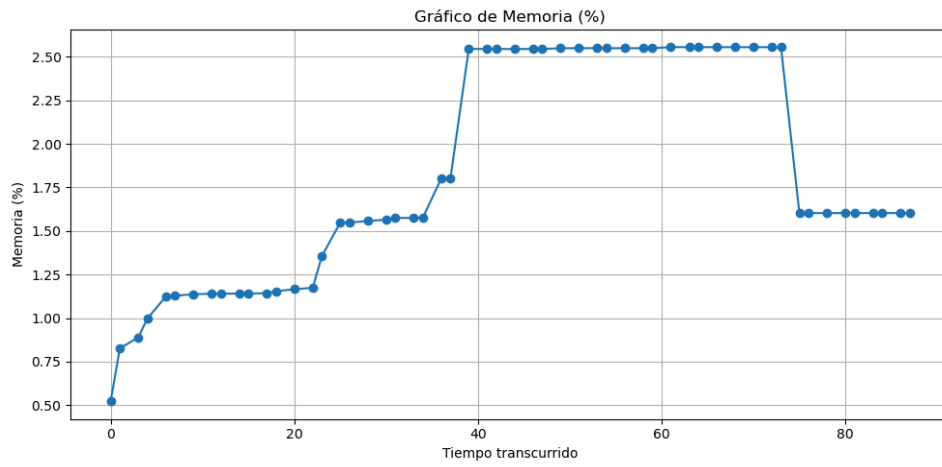
Esta monitorización también se ha llevado a cabo mediante la automatización del Administrador de tareas mediante un script de Python. Este script es ligeramente diferente al anterior, pues al ejecutar el modelo de EEMS, este se ejecuta en otro proceso distinto al de la herramienta.

Para un mayor entendimiento de las gráficas mostradas a continuación se procede a explicar las acciones realizadas durante el tiempo de medición del administrador de tareas. Inicialmente se han dedicado los primeros 15 segundos a abrir la herramienta EE11.8. Posteriormente, para observar mejor en las gráficas las variaciones en estas, se ha esperado hasta el segundo 20 de la medición sin hacer nada. Después, desde el segundo 20 al 30 de la medición hemos cargado un modelo ya existente de EEMS. Más tarde, por la razón explicada anteriormente, hemos estado 5 segundos en un estado de inactividad. Tras esto, desde el segundo 35 hasta el minuto y 15 segundos hemos estado ejecutando el modelo cargado anteriormente. Para finalizar, en los últimos 15 segundos se ha vuelto a realizar una pausa hasta la finalización de la medición.

Dispositivo 1



Dispositivo 2



Capítulo 4 - Casos de uso y planes de contingencia

A continuación hablaremos sobre el desarrollo con la versión demo y el proceso de mallado y de simulación del embalse de Santillana más los planes de contingencia. Vamos a seguir los procedimientos explicados en el [Capítulo 3](#), esta vez aplicado a nuestra muestra que es el lago de Santillana.

Como ya hemos mencionado con anterioridad, nos hemos ido encontrando ciertas limitaciones con el sistema de modelado. El sistema principal con el que se crea el modelo de simulación no ha podido ser usado debido a la falta de la licencia. En esta sección también explicaremos el proceso que hemos seguido para poder construir una simulación, reflejaremos los problemas a los que nos hemos enfrentado y los caminos alternativos que hemos seleccionado para enfrentarnos a estos. De esta manera, en esta sección quedará reflejado nuestro proceso mental mediante el cual hemos tomado ciertas decisiones, y esperamos que esto pueda servir a futuro para posibles proyectos a modo de inspiración o quizás a modo de un primer punto de partida.

1. Proceso de mallado en el embalse de Santillana

Como punto de partida debemos acotar el cuerpo de agua, en nuestro caso el embalse de Santillana. Para ello se ha utilizado la herramienta de Grid+. Con ella, empezamos a hacer un contorno curvilíneo a lo largo de toda la cuenca del embalse de Santillana. Posteriormente y cerciorándonos de que el contorno está perfectamente cerrado, procedimos a separar en lago en celdas quedando el resultado de la [Figura 15: Malla de 34080 celdas del embalse de Santillana](#). El resultado es de una malla de 34080 celdas. Una vez terminamos con la malla, empezamos con el proceso de exportación de esta para poder crear un modelo sobre ella en EEMS usando la versión DEMO. Fue aquí cuando nos encontramos con el primer problema.



Figura 15: Malla de 34080 celdas del embalse de Santillana

1.1 Problema - Limitación en la exportación de archivos en Grid+

El paso siguiendo el orden es exportar el Grid en formato EFDC, pero el programa nos incapacitaba su extracción, limitando solo a 4 formatos que desconocíamos. Después de esto, decidimos proseguir con los formatos de exportación que nos dejaba la versión actual del programa visibles en [Figura 16: Formatos de exportación permitidos](#).

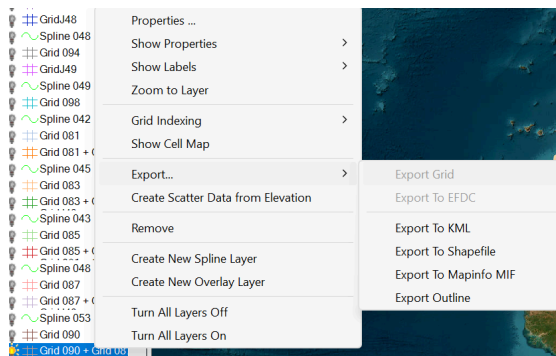


Figura 16: Formatos de exportación permitidos

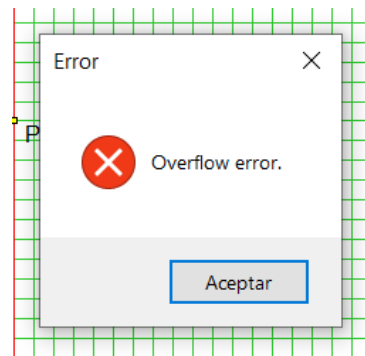


Figura 17: Error 'Overflow' de EEMS

1.1 Solución - Extracción de los diferentes formatos de archivo disponibles

Tras varias pruebas exportando la malla en diferentes formatos como kml, gpc o formatos exportación del proyecto de la malla en Grid+ como gpp y no poder continuar con el proceso, exportamos el contorno del embalse en formato p2d (Export Outline).

2. Proceso de Modelado en el embalse de Santillana

El siguiente paso es importar las mallas obtenidas en el [paso anterior](#) . Podemos seguir los dos caminos para crear un nuevo modelo del embalse: La creación de un nuevo modelo generando un [Uniform Grid]. O importar un proyecto de Grid completo. Esto se explica de forma más detenida en la siguiente [sección](#).

2.1 Problema - Limitación en la importación de archivos gráficos de Grid+

Como se puede ver en la [Figura 16](#), la opción de exportar a ECDF estaba limitada por la DEMO, no podíamos usar la opción de importar una malla de nuestros archivos. Como consecuencia, decidimos usar el camino del uniform grid. La idea se basaba en empezar a crear el modelo sin una malla por debajo, crear dentro del propio EEMS un modelo vacío y acotarlo con el archivo p2d que la versión aceptaba su importación. Cuando realizamos las pruebas con los tamaños de las celdas y del contorno p2d necesarios siempre obtuvimos el mismo error de Overflow que se puede ver en la [Figura 17](#).

2.1.1 Solución - Cambio de muestra a lagos más pequeños

Realizamos otro intento más al pensar en usar un lago que abarque menos agua, como la laguna del Campillo. Al tener que simular menos metros cúbicos y menos diferencia de latitud/longitud, quizás iba a proporcionarnos algún modelo, pero el fallo seguía saltando .



Figura 18: Malla del lago del Campillo

2.2 Problema - Limitación en el número de celdas de la malla

Tras investigar en la página del [EEMS](#), indicaba que para crear un modelo a partir de una malla en la versión demo esta no podría exceder las 300 celdas.

2.2.1 Solución - Disminución de celdas

La alternativa que contemplamos, era crear el modelo de simulación bajo una malla de menos de 300 celdas. Esta malla iba a carecer de efectividad. Íbamos a recrear un escenario muy grande, de casi 87,0 Hm³ según el [Instituto de Estadística](#) con una prueba muy escueta.



Figura 19: Malla <300 celdas del embalse de Santillana

2.3 Problema - Incompatibilidad de importación de modelos en EEMS

Una vez realizado dicho proceso, nos encontrábamos con el mismo problema que en las mallas ya creadas con anterioridad y de tamaño superior a 300 celdas. En este caso se cerraba la aplicación sin dar aviso. Al seguir investigando dentro de las restricciones que tenía la versión de prueba, llegamos a la conclusión que no podíamos importar ninguna malla ni ningún proyecto ni modelo de batimetría nuevo. Finalmente, al ver que no había manera de usar el EEMS, decidimos buscar nuevas alternativas.

2.3.1 Solución - Descarte de EEMS. Creación de archivo de simulación desde el inicio. Uso de archivos KML en simulaciones

La idea principal era rellenar de datos de manera manual hasta conseguir un archivo válido de simulación, imitando el funcionamiento que nos debería de ofrecer EEMS. Para ello había que informarse sobre cuál era el formato final de la simulación. En este caso eran los archivos [NETCDF](#).

Nos propusimos la idea de hacer algo con modelos existentes. EEMS recoge varios [modelos existentes](#) para su uso libre. La idea que nos ha permitido sacar la simulación del embalse de Santillana, está basada en el caso de prueba del Lago Washington.

Para entender un poco más este proceso, vamos a explicar los pasos que hemos ido realizando. Inicialmente, exportamos la malla creada en Grid+ a formato kml. Era el único formato de los que teníamos que tenía características relevantes que podíamos llegar a extraer desde Python. Mediante un script, conseguimos sacar las coordenadas de las distintas celdas de la malla. Esto se debe a que el kml tiene un campo para cada punto con el siguiente formato, <coordinates> longitud, latitud, altitud (opcional) </coordinates>. Para corregir e ir viendo el avance del programa, hicimos una función para pintar una array de puntos en base a unas coordenadas. Corroboramos que el código funciona al pasarle el netCDF4 del lago Washington el cual fue dibujado satisfactoriamente en la [Figura 21](#). Al obtener un archivo nuevo, podríamos verificar la validez de sus coordenadas con este código.

Nuestra primera idea era crear un archivo netCDF desde 0. Crear las variables necesarias para que funcione, fijándonos en el netCDF4 de Washington. Teníamos las variables de latitud y longitud, solo había que añadir el resto de variables. Rápidamente nos dimos cuenta que era muy complicada esta idea.

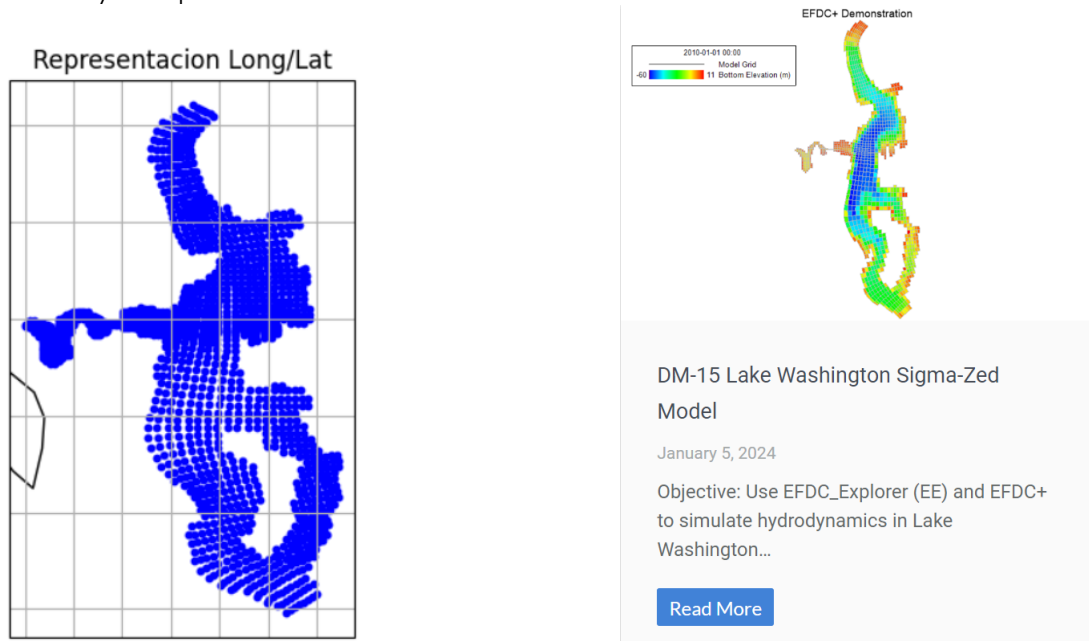


Figura 21: Salida del script de corrección del lago Washington Figura 20: Modelo de simulación ofrecido por EEMS

2.4 Problema - Limitaciones en la inserción de datos geospaciales en variables de simulación

Existen variables en las que había que insertar datos que tuvieran un sentido dentro de la simulación, y no podían ser extraídos de una forma alternativa (como puede ser la latitud de un lago, la profundidad...). Por ejemplo la variable 'nv' es una matriz 4x[número de celdas] en el que cada columna hace referencia a una celda de agua del lago y cada fila de esa columna sitúa en un eje cartesiano los vértices para encuadrar y colocar esa celda en el mapa. Esas coordenadas resultaban imposibles de sacar de una manera sintética. Sumado a esto, el archivo kml empezado a usar en la [Solución 2.3.1](#) es un formato de archivo que se utiliza para mostrar datos geográficos en un navegador terrestre. Fueron imprescindibles para la extracción de la latitud y longitud, pero no contenían ninguna otra variable extraíble desde Python.

2.4.1 Solución - Descarte de la inserción de datos. Trasposición de lago Washington a la ubicación del embalse de Santillana

Según lo visto anteriormente, queríamos explotar al máximo la oportunidad que nos brindó el contorno y la malla de Grid+ obtenidos en [el proceso de mallado de Santillana](#) y apoyarnos en las variables del netCDF4 ya creado de Washington usado en la [Solución 2.3.1](#).

Nuestra forma de pensar fue utilizar variables de un archivo correctamente funcional, e intentar llevarla a nuestro terreno. Hacer una "calca" del embalse de Santillana dentro de una porción del lago Washington. Esta idea fue la más sólida hasta el momento y fue convergiendo en nuestro resultado más correcto como veremos ahora.

Quisimos trasplantar las variables de Washington más importantes para la simulación a nuestro nc. Intentamos varios métodos:

1. Reemplazar archivo Washington

El archivo Washington totalmente funcional fue renombrado y la idea era usar las diferentes librerías de Python para sustituir las anteriores variables geográficas de Washington por las nuevas de Santillana obtenidas en [Solución 2.3.1](#).

2. Integración de variables geospaciales en un archivo nuevo

El intento fue crear un archivo netCDF4 y meter las variables de Washington funcionales sumadas a la inserción de las variables de latitud y longitud de Santillana obtenidas en [Solución 2.3.1](#).

2.5 Problema - Incoherencia en las dimensiones de las variables

En ambos métodos nos daba un fallo 'Out of bounds'. Y es que las dimensiones de las variables del lago Washington eran diferentes. Las librerías de Python de NETCDF4 al igual que los archivos netCDF4 funcionan correctamente si todo tiene unas dimensiones coherentes e iguales entre sus variables. Posteriormente veremos que ese problema de dimensiones se extendía a todo el archivo.

```
[40.7189360601654, 40.7188643172768, 40.7189004942963, 40.7189933347106  
[-3.869586109173, -3.869598478638, -3.869494288815, -3.869477561173, -3  
Tam longitud: 34080  
Tam latitud: 34080
```

Figura 22: Variables 'lon' y 'lat' de Santillana.nc

```
[-122.34986 -122.34985 -122.34756 ... -122.1885 -122.18888  
-122.189255]  
[47.646275 47.647198 47.645306 ... 47.573578 47.576538 47.579494]  
Tam longitud: 1448  
Tam latitud: 1448
```

Figura 23: Variables 'lon' y 'lat' de Washington.nc

2.5.1 Solución - Uso de MATLAB

En este punto la herramienta adecuada a usar fue MATLAB. MATLAB está optimizado para operar con matrices. Aprovechando que nuestras variables del archivo netCDF4 tienen forma de matriz, la inserción de variables de un archivo a otro iba a ser directa ([Figura 24: Variables de Santillana.nc en MATLAB](#)).

Al llevar a cabo la copia de las variables con MATLAB no obtuvimos ningún error. No teníamos las variables extraídas de Python en [Solución 2.3.1](#) ([Figura 22: Variables 'lon' y 'lat' de Santillana.nc](#)) pero queríamos de alguna forma conseguirlas. Restamos la diferencia de latitudes y longitudes que se llevaban todas las celdas de los dos lagos. Al fijarnos en el archivo netCDF4 obtenido nos dimos cuenta que el script que usábamos solo copiaba bien algunas celdas de nuestro embalse y en el resto apuntaba a latitudes que no conformaban la forma de Santillana, sino a porciones de tierra. Aparentemente esta solución iba bien encaminada pero descubrimos este error gracias al script de corrección creado en [Solución 2.3.1](#). Por eso al corregirlo, se nos pintó nuevamente el lago de Washington, cuando debería haber aparecido

el lago de Santillana. Había que buscar unas coordenadas donde al hacer la resta no hubiese ningún punto de tierra, y era muy laborioso.

2.5.2 Solución - Tratar la diferencia en las dimensiones de las variables

Después de un estudio de las variables, decidimos pintar el tamaño de la dimensión NODE explicado anteriormente "[Variables de la simulación](#)". Una vez hecho esto, nos dimos cuenta que el tamaño de esta dimensión no había sido redimensionado y que estaba manteniendo el tamaño del Lago Washington. La ejecución de un código en Python consiguió la redimensión de NODE.

filename	'a2.nc'
lat	34080x1 dou...
latc	1183x1 single
lon	34080x1 dou...
lonc	1183x1 single
nv	4x1183 uint32
temp	1183x55x721...
time	721x1 double
wind_x	1183x721 do...

Figura 24: Variables de Santillana.nc en MATLAB

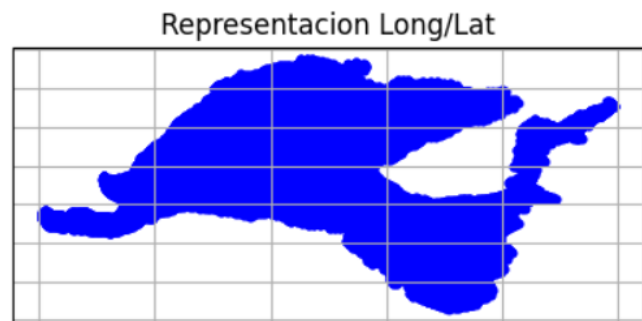


Figura 25: Salida del script de corrección de Santillana

Por consiguiente, el script de Python ya sustituía las variables de 'lat' y 'lon' correctamente. Gracias al script de corrección en Python comentado anteriormente, podemos ver en la [Figura 25: Salida del script de corrección de Santillana](#) que se nos pintaba exactamente igual el lago. En esa imagen estaba la el contorno del embalse Santillana sobre el modelo de Washington. Conservaba todas sus dimensiones y variables.

3. Proceso de recreación de la simulación del embalse de Santillana

El siguiente paso y el más tedioso era recrear la simulación. Las variables, pese a su correcta dimensionalización de la variable NODE, tienen diferente número de celdas. Como podemos ver en la [Figura 24: Variables de Santillana.nc en MATLAB](#) los tamaños de las matrices de 'lat' y 'latc' no concuerdan. Desconocíamos el funcionamiento de la simulación, pero hicimos una serie de modificaciones para acomodar las variables para la representación en 3D del lago y la simulación en DEVS-BLOOM.

Nos apoyamos mucho en nuestros compañeros del TFG de la representación del lago en 3D. Les mandamos la primera maqueta del lago, pero como previmos era errónea y

representaba el lago Washington. Las diferencias de las dimensiones de las variables eran las detonantes.

3.1 Problema - Fallos en la representación 3D

Teníamos un problema ya que como hemos podido comprobar con las pruebas anteriores, hay gran dificultad en redimensionar las variables, ese problema se extrapola a la representación 3D. Con la redimensión de NODE solo bastaba para subsanar la representación en 2D.

3.1.1 Solución - Redimensión de variables relevantes para la representación 3D

Contando con que las dimensiones que diferían eran 'lat' y 'lon' y vimos anteriormente que no se podían redimensionar. Decidimos redimensionar variables que aparentemente fuesen necesarias para la representación. Nos decantamos por las variables 'latc' y 'lonc'.

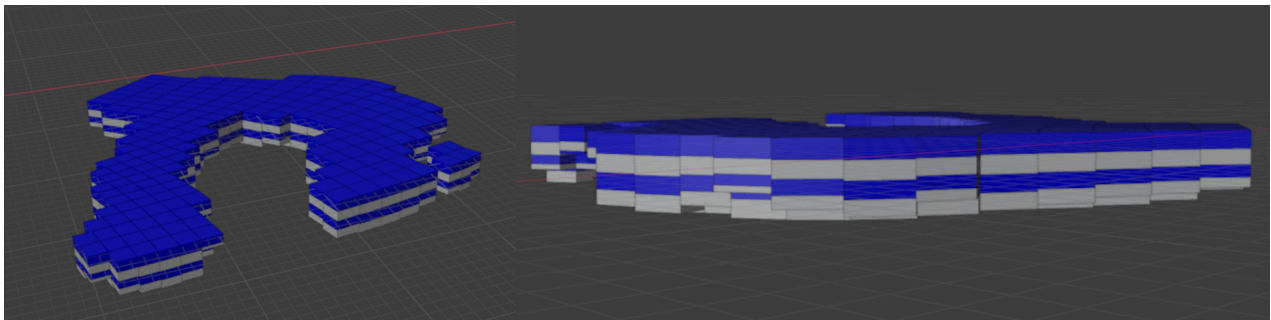


Figura 26: Representación 3D 1

Figura 27: Representación 3D 2

Podemos ver una representación algo acertada y asemejada a una porción del lago de Santillana. Pero el resto de dimensiones creaban incoherencias, como se puede ver en la imagen. Alteran el tamaño de las celdas y hace que se superpongan unas encima de otras dando lugar a esa forma.

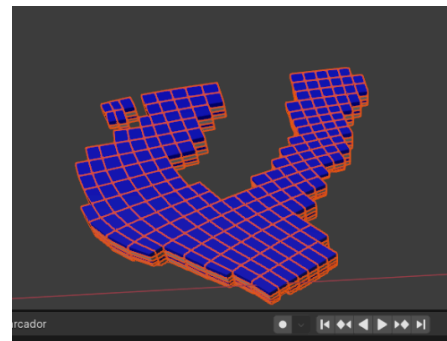
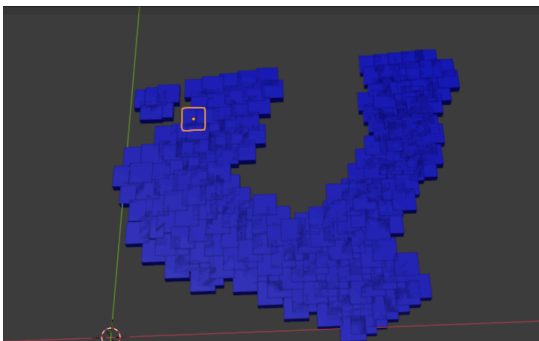


Figura 28: Representación 3D 3

Por último y tras algunas reuniones con los compañeros poniendo en común los conocimientos adquiridos por ambos TFGs, llegamos a una conclusión. El problema lo causaba la variable 'nv'.

3.1.2 Solución - Creación de una nueva malla

Viendo que no podíamos redimensionar de una manera sintética ninguna variable, por última opción decidimos crear una malla nueva. Al crear una malla nueva más pequeña, como hemos visto en [Solución 2.3.1](#), extraeríamos de su kml unas variables 'lat' y 'lon' con menos elementos. Teníamos que llegar a una malla con una forma exacta cuyas dimensiones de los variables se acercasen lo máximo posible a las dimensiones que trae el archivo de Washington como pudimos ver en la [Figura 30: Salida del script de corrección de Santillana 1450 celdas](#), en este caso tamaño 1448. Por consiguiente no habría que realizar ningún cambio en ninguna variable ni ninguna dimensión. Tras una meticulosa creación de una nueva malla y al reutilizar los programas creados y obtuvimos un archivo aparentemente bastante sólido.

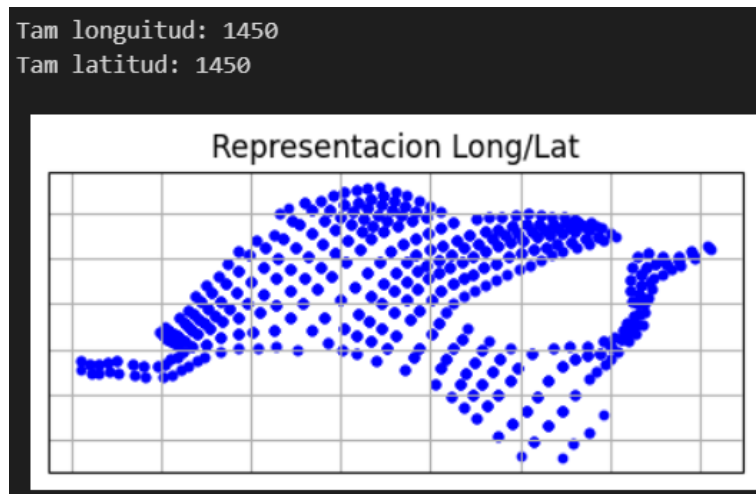


Figura 30: Salida del script de corrección de Santillana 1450 celdas

Contenía un número de celdas pequeño para evitar solapamientos a la hora de la representación, el tamaño de las variables era 1450, es decir 2 más que el archivo referencia [Figura 22: Variables 'lon' y 'lat' de Santillana.nc](#). Por último las latitudes y longitudes eran

correctas y se representaba la silueta del lago correctamente, pero el resultado no fue satisfactorio.

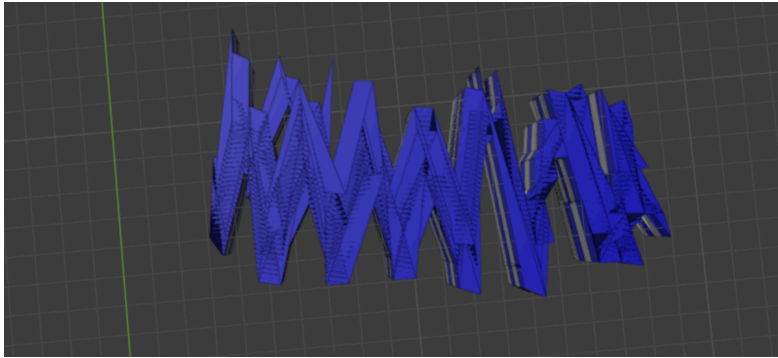


Figura 31: Representación 3D

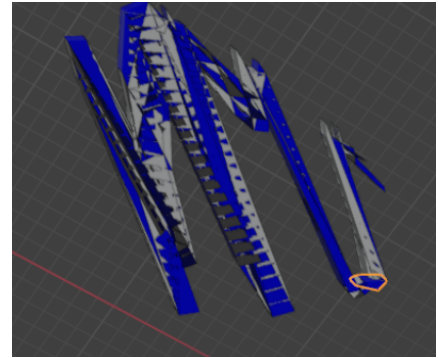


Figura 32: Representación 3D 2

4.1 Diagnóstico de las limitaciones y obstáculos

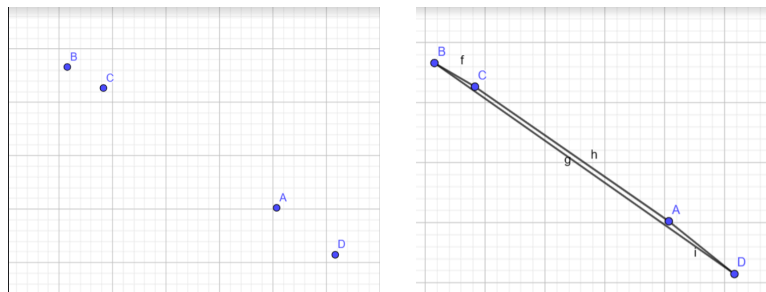
Nuestro cometido en este TFG era crear un archivo que se pudiese simular sobre él y a su vez se pudiese representar en 2D y 3D, por ello los conocimientos que hemos adquirido era para solventar esos 2 objetivos. La escasez de referencias o ejemplos en este preciso tema, nos ha hecho hacer un estudio intenso sobre el funcionamiento de cada variable y su comportamiento dentro de esa representación y esa simulación. Hemos podido comprobar qué variables son indispensables para la representación, y cuales son menos relevantes, hasta el punto que se pueden reemplazar.

Partiendo de la base que nuestra idea de simulación, parte de otra simulación, podemos asegurar que no es posible la representación 3D hasta el momento. Hay variables que son características y exclusivas de cada simulación, lo que supone que la copia no es una opción. El método o el programa que está creando el archivo netCDF4 sigue un patrón o un algoritmo que forma las variables específicamente para esa ubicación concreta. Sin el conocimiento de ese algoritmo o sin saber cómo se extraen esos datos para esas variables, es imposible la representación de un netCDF4 mediante la copia de otro, ya que podemos asegurar que es exclusivo.

Hablamos de la variable 'nv'. Esta variable guarda una matriz $4 \times X$, donde X es el número de celdas que va a tener la representación, como demuestra la [Figura 24: Variables de Santillana.nc en MATLAB](#). Como ya sabemos, en Washington eran 1448 celdas. Esta variable contiene los 4 vértices que conforman el cuadrado que forma la celda. Al situarlo en un eje cartesiano se irían formando cuadrados correctamente situados en él. Las columnas de la

matriz son las 1448 celdas y las filas, cada uno de los 4 vértices de cada celda. Esa sería una disposición lógica pero el caso es que la matriz no está ordenada de esa forma. Al traspasar esa disposición de los vértices a otro archivo deja de tener coherencia. Por poner un ejemplo, la celda con 'idx'=0 para 'X' se encuentra en el norte y para 'Y' se encuentra en el este. Al volcar los datos de 'X' → 'Y', la 'idy'=0 pasa a estar en el norte, cuando 'idy' le pertenece estar en el este. Esto extrapolado a cada celda y a cada vértice de cada celda, produce una incoherencia en la representación.

Para entender este comportamiento empecé a estudiar la disposición de los vértices. Utilizando un programa de graficación como geogebra, visualicé la disposición de los vértices en la primera celda de Santillana.nc y este es el resultado. Viendo esto cobra sentido la imagen de arriba.



Capítulo 5 - Ejecución del framework completo

Una vez obtenido el archivo de tipo netCDF4 del embalse de Santillana, llega la parte de simular este en el software de DEVS-BLOOMS.

Para llevar a cabo esta parte del proceso, el tutor al principio del desarrollo del trabajo nos entregó el código de dicho software desarrollado en Python. Para entender mejor este punto, se describirán a continuación algunos aspectos del mismo.

Este software recibe como entrada, el archivo de tipo netCDF4, el cual se ha obtenido anteriormente del software EEMS. En el caso del proporcionado por el tutor estaba diseñado para realizar una simulación del Lago Washington. Este software como salida, guarda el movimiento del barco que va persiguiendo las algas, en formato csv.

La simulación consiste en este software consiste en que se sitúa el barco en una posición inicial que nosotros determinemos mediante unas coordenadas de latitud y longitud. Dicho barco está equipado con una serie de sensores que van midiendo una serie de parámetros que nos interesan, como la concentración de algas, oxígeno, nitratos, la batimetría, etc. aunque realmente lo que hacen es recoger la información que se ha guardado previamente por EEMS, en la ubicación en la que se encuentra el barco.

Después de esto, se añade un modelo de Bloom, que básicamente es una predicción sobre donde se cree que va a aparecer el afloramiento. Esto se hace en base a los datos que están aportando los sensores mencionados anteriormente.

Como resultado de la ejecución obtenemos una serie de archivos csv. Uno de ellos es el archivo 'BloomModel.csv' el cual contiene una predicción de donde el modelo piensa que va a generarse un bloom, y este lo modela con una mancha en forma de círculo de un radio determinado. Cuando este radio es lo suficientemente grande o la concentración de oxígeno es suficiente se predice como si hubiese un bloom, y entonces el barco se desplazará hacia allí.

Dentro de estos csv, obtenemos otros con información de cada sensor, lo que se recoge directamente de EEMS como ya hemos mencionado anteriormente. Vamos a pasar a mencionar y explicar brevemente cada uno:

- **SimSenA.csv:** cianobacterias predichas por EEMS
- **SimSenB.csv:** profundidad del lago
- **SimSenN.csv:** concentración de nitratos
- **SimSenO.csv:** concentración de oxígeno
- **SimSenS.csv:** irradiación solar
- **SimSenT.csv:** temperatura

- **SimSenU.csv**: velocidad del viento
- **SimSenV.csv**
- **SimSenX.csv**
- **SimSenY.csv**

Luego, también tenemos el archivo 'UsvModel.csv', que guarda algunos datos como la posición del barco en cada momento, esto lo hace mediante su longitud y latitud, y la energía de la batería de esta, normalizada a 1.

Una vez obtenidos estos archivos, se pueden visualizar gráficamente en [2D](#) o en 3D. De esta segunda parte se están encargando otros compañeros de la facultad en su TFG.

A continuación, vamos a mostrar alguna imágenes de la visualización 2D de una simulación del embalse de Santillana, la primera muestra la predicción del modelo de la aparición de blooms, y la segunda el movimiento del barco en la persecución del bloom:

A continuación, se muestran unas gráficas obtenidas con Python que representan información de diferentes variables del archivo netCDF4.

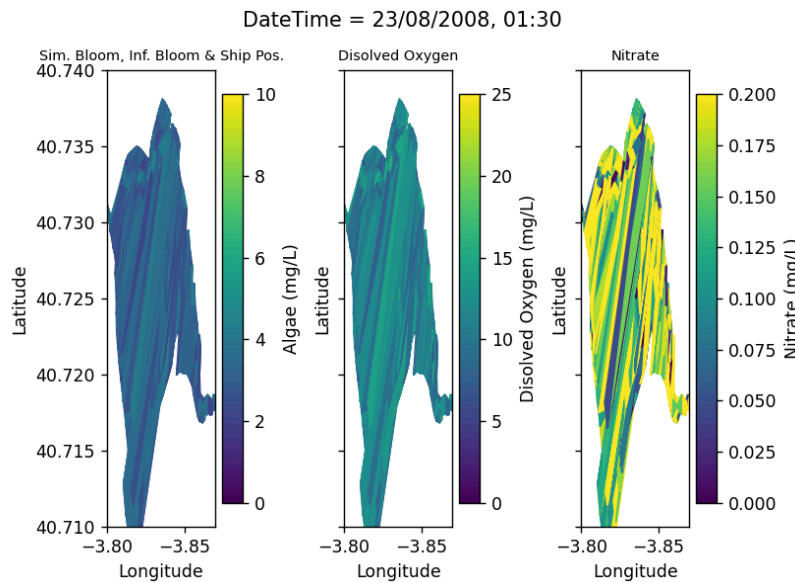


Figura 33-Representación variables 1

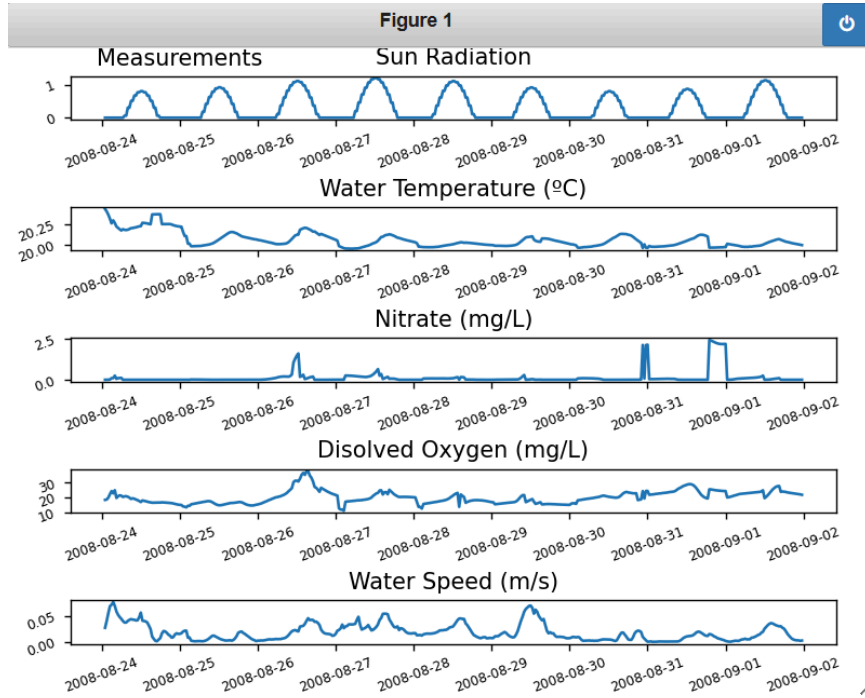


Figura 34 - Representación variables 2

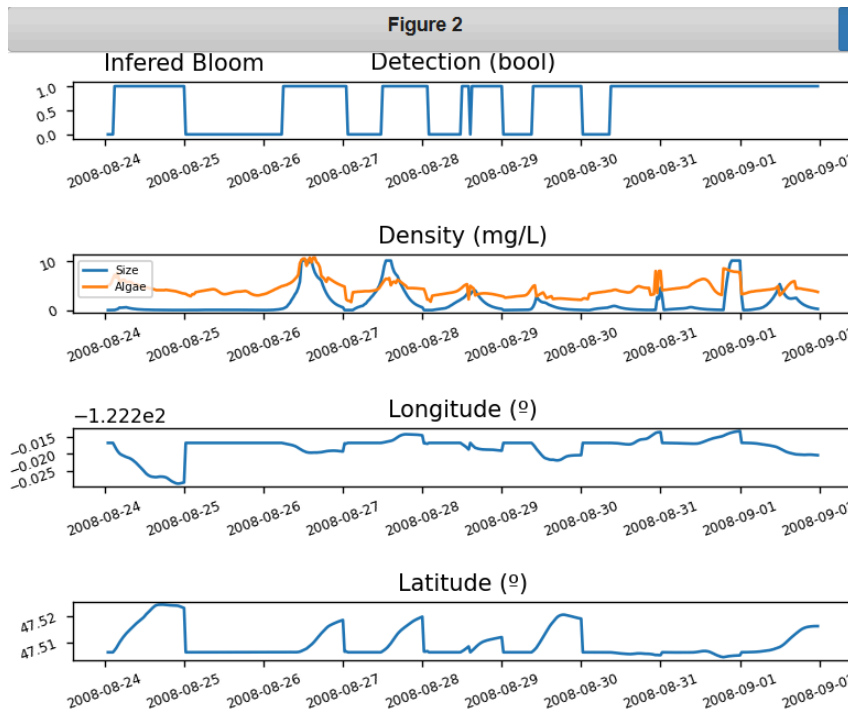


Figura 35-Representación variables 3

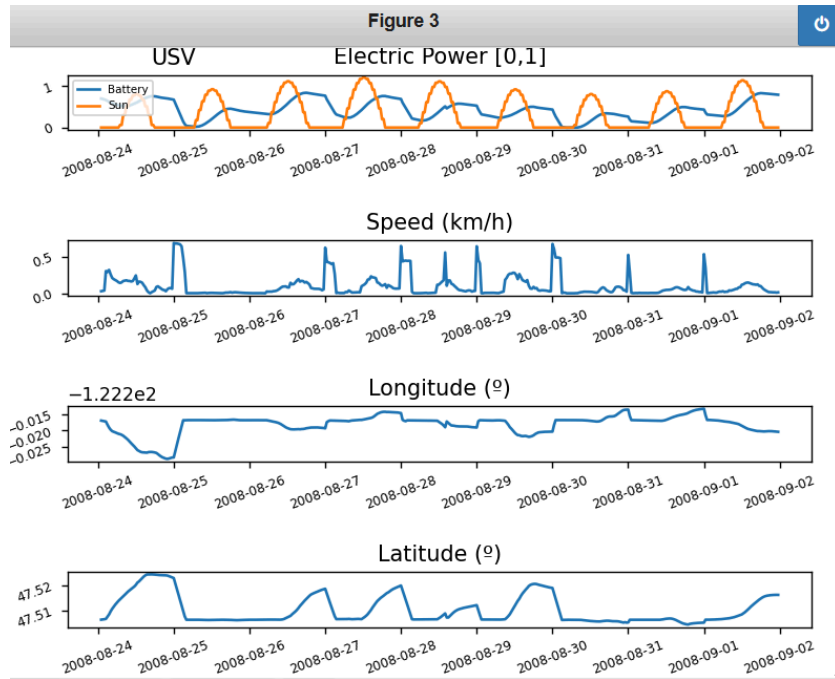


Figura 36-Representación variables 4

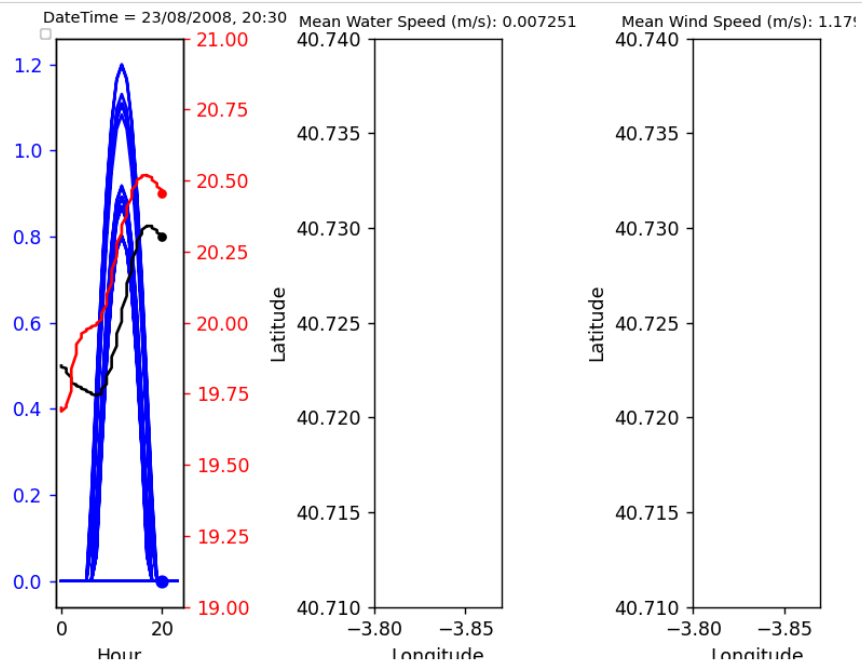


Figura 37-Representación variables 5

Capítulo 6 - Conclusión y trabajo futuro

En este capítulo se presenta una revisión de los objetivos establecidos en la sección [1.2](#), junto con un resumen general de las principales conclusiones obtenidas de este estudio, a modo de consideraciones finales. Además, se incluyen propuestas de investigaciones futuras para dar continuidad a este trabajo.

6.1 Revisión de los objetivos

A lo largo de todo el documento, se han abordado los diferentes objetivos establecidos al inicio del trabajo. En esta sección, relacionaremos los objetivos con los capítulos, secciones y subsecciones correspondientes. De este modo, veremos cómo se han ido logrando todas las metas propuestas.

1.Desarrollo de un modelo de simulación para embalses y lagunas:

Hemos construido un modelo computacional gracias a Grid+ y usando programas mayoritariamente en Python y MATLAB supliendo la acción de EEMS. Pese a que tiene por debajo variables y cualidades de otro lago, es capaz de representar la dinámica de las cianobacterias.

2.Generación y análisis de datos sintéticos:

Podemos usar el modelo de simulación para crear diversos csv gracias a DEVS-BLOOM. Entre estos archivos se encuentra la dinámica de la cianobacterias, los movimientos del USV de acuerdo a la predicción de la floración de las cianobacterias en función de las condiciones establecidas por nuestro modelo.

3.Contribución al proyecto SMART-BLOOMS:

La contribución a SMART-BLOOMS no ha sido la esperada. Sin embargo, hemos estudiado a fondo la integración del modelo de simulación. Dadas nuestras condiciones, hemos contribuido a la exploración de otros modos de integrar y crear modelos sin el sistema predefinido. Explorando formas alternativas de modelaje y representación útiles en un futuro. Conociendo cuales son las virtudes y las desventajas, al igual que determinando los límites de estas nuevas formas de modelaje y abriendo caminos para las posibles representaciones futuras en 3D.

4.Difusión de resultados y conocimientos:

Hemos documentado minuciosamente los procesos necesarios para crear una simulación. Al encontrarnos el inconveniente de la falta de licencia, hemos adaptado el

documento hacia todo el espectro de usuarios. Desde los que tienen licencia del sistema, como los que disponen sólo de la versión de prueba, hasta usuarios que no disponen del sistema. Describiendo los pasos a seguir desde el proceso de autenticación de licencia, siguiendo por el proceso de mallado hasta la creación del modelo para toda la gama de usuarios antes mencionados.

6.2 Revisiones finales y trabajo futuro

A partir de lo visto en el [capítulo 4](#), podemos concluir en una simulación funcional para su uso en predicciones de FAN y apta para iniciar misiones de USV para el análisis y gestión de las algas. Como vimos en el apartado [5](#), la ejecución resulta en unos csv determinados que describen el movimiento de las algas y a su vez el paso del USV sobre la zona de mayor abundamiento de afloración. Todo ello apto para ser representado gráficamente en una interfaz amigable.

A pesar de todo lo intentado, no podemos concluir un trabajo exacto como acordamos en algunos objetivos. El alto coste de la licencia de uso ha dificultado nuestro desarrollo con EEMS. Dadas las condiciones limitantes que hemos tenido a la hora de crear los modelos por el sistema predilecto para ello, las características del agua conseguidas para la simulación no son exactamente las cualidades del agua de Santillana. Lo podemos denominar como una investigación de un método alternativo para obtener simulaciones reales de comportamientos en volúmenes de agua descrita paso a paso en el [capítulo 4](#). Consiguiendo hasta la fecha una técnica basada en la transposición de modelos desde uno ya formado, aplicando la forma geográfica y las coordenadas del nuevo cuerpo de agua a modelar. Desarrollamos pruebas, intentado con diversos métodos, utilizando varios lenguajes de programación.

Como consecuencia, hemos conseguido que el modelo sea representable además de ejecutable, lo que en primera instancia resultaba ser algo complicado. Al mismo tiempo esta versión alternativa propuesta no es capaz de ser representada en futuras representaciones 3D, debido a la naturaleza de las variables de simulación que hemos comentado en el apartado [4.1](#).

Finalmente, conociendo el EEMS en sus diferentes versiones, tanto la versión demo como con la versión completa, hemos aunado los conocimientos y experiencia en un documento. Reflejamos en él a modo de guía los pasos a seguir en un modelaje, describiendo procedimientos, posibles errores a tener en cuenta y el rendimiento y coste de CPU y memoria que supone, todo recogido en el [capítulo 2](#).

Teniendo en cuenta las conclusiones previamente presentadas, es importante resaltar que los resultados y conclusiones obtenidas representan solo un paso adicional en el proceso de simulación de modelos. Requiere de un estudio continuado impulsando nuevas ideas y alternativas para que siga en la dirección correcta. Con este documento ayudamos en la ergonomía del uso de sistemas de simulado con licencia y plantamos una posible semilla en la simulación sin sistemas de simulado.

Se puede aumentar el rango de usuarios dispuestos a crear simulaciones y se les puede facilitar el proceso, pero aún queda mucho por estudiar. Se necesita explotar la compatibilidad vista en este documento de modelos ya creados, con otros lenguajes de programación. Hay que ser capaces de adaptar modelos replicados y que pierdan todas las características del anterior. Que sean aptos para ser representados en 3D sin ninguna excepción, o buscar una extensión de archivos que sea compatible con simulaciones y flexible en su manipulación. Siempre hay una versión de mejora y de avance y más aún si supone un progreso en la calidad de la salud de los humanos.

Capítulo 7 - Conclusions and future work

This chapter shows a review of the objectives established in section 1.2, with a general summary of the main conclusions obtained from this study, as final considerations. In addition, proposals for future research are included to give continuity to this work.

7.1 Review of objectives

Throughout the document, the different objectives established at the beginning of the work have been addressed. In this section, we will relate the objectives to the corresponding chapters, sections and subsections. In this way, we will see how all the proposed goals have been achieved.

1. Development of a simulation model for reservoirs and lagoons:

We have built a computational model thanks to Grid and using programs mostly in Python and MATLAB replacing the action of EEMS. Although it has below, and other lake qualities, it is able to represent the dynamics of cyanobacteria.

2. Generación y análisis de datos sintéticos:

We have built a computational model thanks to Grid and using programs mostly in Python and MATLAB replacing the action of EEMS. Although it has below, and other lake qualities, it is able to represent the dynamics of cyanobacteria.

3. Contribution to the SMART-BLOOMS project:

The contribution to SMART-BLOOMS has not been as expected. However, we have thoroughly studied the integration of the simulation model. Given our conditions, we have contributed to the exploration of other ways to integrate and create models without the predefined system. Exploring alternative ways of modeling and representation useful in the future. Knowing what are the virtues and disadvantages, as well as determining the limits of these new forms of modeling and opening ways for possible future 3D representations.

4. Spreading of results and knowledge:

We have thoroughly documented the processes required to create a simulation. When we encountered the inconvenience of the lack of a license, we have adapted the document to the whole spectrum of users. From those who have a system license,

such as those who only have the trial version, to users who do not have the system. Describing the steps to follow from the license authentication process, through the meshing process to the creation of the model for the whole range of users mentioned above.

7.2 Final revision and future work

From what we saw in chapter 4, we can conclude in a functional simulation for the use in FAN predictions and suitable for initiating USV missions for algae analysis and management. As we saw in section 5, the execution ends with a certain csv that describes the movement of algae and in turn the passage of the USV over the area of greatest abundance of upwelling. All this can be represented graphically by a friendly interface.

We can't conclude an accurate work as we agreed on some objectives. The high cost of the license has made our development with EEMS impossible. Having into account the limiting conditions we have had when creating the models with the suitable system, the water characteristics obtained for the simulation have not exactly the qualities of Santillana. We can call it an investigation of an alternative method to obtain real simulations of behavior in volumes of water described step by step in chapter 4. Achieving a technique based on the transposition of models. From one already formed, applying the geographic shape and coordinates of the new body of water to be modeled. We developed tests, tried different methods, using various programming languages.

As a consequence, we have managed to make a representable model as well as executable, which at first seems to be complicated. At the same time this alternative version is not able to be represented in future 3D representations, due to the nature of the simulation variables that we have discussed in section 4.1.

Finally, knowing the different EEMS versions, both the demo version and the full version, we have combined our knowledge and experience in a document. The steps you have to take in a modeling, describing procedures, possible errors to take into account and the performance and cost of CPU and memory involved, are reflected in it as a guide, all collected in chapter 2. Taking into account the conclusions previously presented, it is important to emphasize that the results and conclusions obtained represent only an additional step in the model simulation process. It requires a continued study, promoting new ideas and alternatives in order to continue in the right direction. With this paper we help in the ergonomics of using licensed simulation systems and plant a seed in simulation without simulation systems. The range of users willing to create simulations can be increased and the process can be made easier for them, but much remains to be studied. We need to exploit the compatibility seen in this paper of models already

created, with other programming languages. It is necessary to be able to adapt replicated models that lose all the characteristics of the previous one. They must be able to be represented in 3D without any exception, or look for a file extension that is compatible with simulations and flexible in its manipulation. There is always a version of improvement and advancement and even more so if it means an improvement in the quality of human health.

Capítulo 8 - Contribuciones Personales

Al tratarse de un trabajo conjunto, cada participante va a describir el trabajo realizado y su aportación a este proyecto.

Marcos Padilla Alonso

A continuación vamos a describir cada uno su aportación en este TFG . Pese a que se ha desarrollado el trabajo de forma conjunta. Cada uno partía desde el punto que lo había dejado el otro. Nos centraremos en especificar más detalladamente que ha hecho cada uno.

En las primeras semanas, me estuve formando viendo videos de ejemplos de uso del sistema EEMS accesibles en la página oficial. Intentando crear proyectos y creando mallas básicas basándome en las guías. Leyendo documentos y foros donde los usuarios escriben soluciones a sus problemas. Es un sistema difícil de entender sin ningún tipo de documentación, por lo que me llevó tiempo familiarizarme con el entorno. Dado esta falta de información nos llevó a estudiar casos para crear mapas con Google Earth Pro entre otros programas de representaciones geográficas.

Como hemos mencionado anteriormente, nos encontramos con el problema de la licencia del sistema. La naturaleza del TFG nos ha conducido a estar constantemente cambiando y corrigiendo códigos. No podemos asegurar un trabajo individual por parte de nadie en el código ya que es el fruto de aportaciones de ambos participantes. Estuve investigando sobre formas de recrear el funcionamiento de EEMS. Cree un el archivo con Python para introducir variables a un .nc desde 0, pero ciertas variables no se podían modificar su dimensión. Intente alterar las variables del archivo de Washington totalmente funcional sustituyendo las variables lat y lon de Santillana consiguiendo el mismo error anterior con Python.

Me encargué del proceso de transposición del lago de Washington al embalse de Santillana desde Matlab restando las variables de los diferentes archivos. En ambos lenguajes, nos surgieron varios problemas ya mencionados en anteriores capítulos. Me encargue de investigar, apoyándome también con Matlab para averiguar las posibles detonantes del error.

Una vez conseguido el script de corrección mencionado anteriormente, se nos abrieron muchas puertas. Desarrollé un código Python donde se aunaban los dos lenguajes. Se ejecutaba el script de corrección y activaba el proceso de trasposición de variables de Matlab, que dió como resultado el modelo actual más acertado.

Me encargué de mantener comunicación con los del TFG 3D y entender su proyecto, para adaptar nuestros códigos y nuestras simulaciones a su trabajo y conseguir la representación. Realizamos varias llamadas e intercambiamos varios mensajes escritos por correo comparando resultados. Me encaje de la alternativa de la malla alterando las variables $latc$ y $lonc$. Me encargué de explotar este script jugando con varios archivos con extensión nc y ajustando las dimensiones para resolver el problema de la incoherencia de las variables. Esto supone la creación de diversas mallas con varios tamaños de celdas hasta dar con la más exacta.

Me he encargado de estructurar correctamente la memoria, creación de los diversos índices como los de imágenes, tablas y de contenido, asegurándome de su correcto formato y claridad de disposición de estos en el proyecto. También dejando bien referenciadas las URL, enlaces a figuras tablas y páginas web visitadas para la correcta comprensión del TFG. He empleado bastante tiempo en estudiar el proyecto SMART-BLOOM y enterarme debidamente del contexto en el que gira entorno nuestro TFG. De esta manera, conseguimos expresar correctamente a los lectores el funcionamiento de SMART-BLOOM y la ubicación de nuestro TFG dentro de este proyecto.

He asistido a las reuniones acordadas con el tutor al igual que con mi compañero. Y he establecido ciertas reuniones con otro compañero del TFG 3D. Gracias a esta conexión, pude desarrollar la teoría descrita en el diagnóstico de las limitaciones, en el [capítulo 4.1](#). Lo demostré con otras herramientas de graficación como Geogebra e indagando en el contenido de las variables que afectan a este diagnóstico con Matlab y las salidas de Python.

Fernando Sánchez García

Ahora se va a contar el trabajo que ha realizado Fernando en el desarrollo del TFG. Aunque como se ha mencionado anteriormente el trabajo se ha desarrollado conjuntamente entre ambos, en este apartado se contará de forma más específica el trabajo desarrollado por Fernando.

La primera tarea que realizó Fernando fue durante las primeras semanas llevar a cabo un proceso de formación en las herramientas que se iban a utilizar posteriormente para desarrollar el modelo. Esta formación se llevó a cabo mediante la búsqueda de información en internet, en su amplia medida, en la página de EEMS, donde se encontró una pestaña de recursos en la cual hay videos subidos de ejemplo y documentación de uso.

Después de esto y una vez instaladas las herramientas ya mencionadas, se modelaron una serie de mallas de distintos tamaños, con más y menos celdas, tanto del embalse de Santillana como de la laguna del Campillo. Esta última al final no fue utilizada.

El modelado de las distintas mallas se realizó debido a una serie de problemas ocurridos durante el desarrollo del proceso como se explica en el punto de contingencias .

También, se intentó construir un modelo de simulación en el software de EEMS, pero debido a la falta de la licencia no se consiguió lograr por las limitaciones de la versión demo.

Durante este proceso de intento de creación del modelo, se llevaron a cabo varios procesos distintos para intentar conseguirlo, en los cuales Fernando estuvo involucrado, como la exportación de las mallas creadas anteriormente en distintos formatos, con el fin de poder importarla de alguna manera en la herramienta EEMS.

A parte de esto, también ha desarrollado una serie de códigos los cuales han servido para poder obtener un resultado óptimo del TFG, como son los que vamos a explicar a continuación.

Un cuadernillo en lenguaje Python llamado 'info_archivo_nc.ipynb', el cual se ha utilizado para sacar información sobre el archivo netCDF4 que posteriormente se usó en el

software de DEVS-BLOOM. Este cuadernillo utiliza la librería xarray, la cual se suele usar en el manejo y análisis de datos multidimensionales, como es nuestro caso.

Otro cuadernillo, también desarrollado en lenguaje Python, llamado 'modificacion_washington.ipynb', el cual se ha utilizado para llevar a cabo el plan de contingencia [referenciar]. Este cuadernillo se encarga de extraer el array de longitud y latitud del embalse de santillana de la malla creada en Grid+, tras esto y tras una conversión del tamaño de las variables del Lago Washington, el array de este archivo netCDF4 previamente obtenido de la página de EEMS, sustituye las variables necesarias en este por las obtenidas del archivo del embalse de Santillana, realizando así la traslación a modo de contingencia por la falta de la licencia.

También, desarrolló otro cuadernillo llamado 'automatizacion_consumos.ipynb'. Este se encarga de la automatización del administrador de tareas de nuestro ordenador. Recoge el porcentaje de memoria y de CPU, en un intervalo de tiempo que podemos especificar y lo va volcando en un archivo de tipo csv.

El cuadernillo para dibujar las gráficas de consumos, llamado 'dibujar_graficas.ipynb', también fue desarrollado por Fernando. Este basa su código en la librería de Python [Matplotlib](#) y representa gráficamente los archivos csv proporcionados por el código anterior.

El cuadernillo de 'Bloom_sim_bea.ipynb', fue modificado por Fernando para adaptarlo al archivo netCDF4 del embalse de Santillana, y así poder visualizar gráficamente mediante este variables y predicciones, como se explica y se muestra en el [capítulo 5](#).

A todo esto se le suma el estudio detallado del proyecto de SMART-BLOOM en el que se encuentra este TFG, al igual que es estudio del software DEVS-BLOOM para poder entenderlo y ejecutarlo.

A parte de lo ya mencionado, ha participado en el proceso de escritura de esta memoria, trabajo que al igual que el anterior se ha desarrollado de forma conjunta entre ambos integrantes.

Por último, también ha participado en las reuniones periódicas que se han desarrollado de manera virtual, salvo la primera toma de contacto con el tutor, la cual se realizó de manera presencial.

BIBLIOGRAFÍA

- [1] Schmale DG, Ault AP, Saad W et al., Perspectives on Harmful Algal Blooms (HABs) and the cyberbiosecurity of freshwater systems. *Frontiers in Bioengineering and Biotechnology* 2019;128.
- [2] Jose L. Risco-Martín, Segundo Esteban, Jesús Chacón et al ., *Simulation-driven engineering for the management of harmful algal and cyanobacterial blooms*. DOI: 10.1177/ToBeAssignedwww.sagepub.com/.
- [3] Vincent WF. Cyanobacteria. *Encyclopedia of Inland Waters* 2009; 3: 226–232.
- [4] Meriluoto J, Spoof L and Codd GA. *Handbook of cyanobacterial monitoring and cyanotoxin analysis*. John Wiley & Sons, 2017.
- [5] Jesús Chacón, Giordy A. Andrade, Jose L. Risco-Martín et al., *A Bleeding Edge Web Application for Early Detection of Cyanobacterial Blooms*.