



**Universidad**  
Zaragoza

## Trabajo Fin de Grado

Sistema de Información Web para la gestión y visualización de cuadros de mandos de indicadores medioambientales georreferenciados para centros educativos.

A Web Information System for the management and display of scorecards containing georeferenced environmental indicators for educational purposes.

Autor

Íñigo Úriz Gil

Directora

Raquel Trillo Lado



## Resumen

El desarrollo sostenible y la protección del medioambiente están a la orden del día. Numerosos movimientos sociales organizan con frecuencia manifestaciones a nivel mundial en su defensa. Incluso en el plano político se llevan a cabo cumbres en las que se debate qué se puede hacer para proteger nuestro futuro. Esto, unido a la digitalización de las aulas, en las que con frecuencia se utilizan dispositivos electrónicos, crea un marco donde es necesario incidir en aspectos medioambientales en la educación obligatoria. Este documento describe el desarrollo de un sistema de información web para la visualización de cuadros de mandos con indicadores medioambientales georreferenciados en el ámbito educativo.

En este trabajo, se ha desarrollado una aplicación web que permite el refuerzo del aprendizaje de conceptos de asignaturas como matemáticas o tecnología de manera más dinámica, logrando una mayor motivación del alumnado y mayor efectividad en el estudio. Para ello, el profesorado de dichas asignaturas puede crear y compartir con el alumnado mapas con información medioambiental de una localización concreta. Cuando el alumnado accede a ellos, se le presenta un cuadro de mandos en el que se muestran los sensores medioambientales georreferenciados instalados en la localización escogida. Así, si el usuario selecciona uno de ellos, se muestran gráficas de los valores recogidos para el contaminante seleccionado a lo largo del tiempo. Además, el profesorado puede plantear cuestionarios y ejercicios relacionados con cada uno de los mapas a los estudiantes. Una vez realizados estos, el alumno puede almacenar notas y comentarios sobre, por ejemplo, explicaciones ofrecidas por el profesor en el aula, que se pueden recuperar después en su domicilio.

Se han alcanzado los objetivos planteados en la propuesta de este Trabajo Fin de Grado. No obstante, debido a la crisis sanitaria, no ha sido viable la realización de pruebas en un instituto con los estudiantes en el aula.

## Abstract

Sustainable development and environmental protection are the order of the day. Numerous social movements often organize worldwide demonstrations in their defense. Even at the political level, summits are held to discuss what can be done to protect our future. This, together with the digitization of classrooms, where electronic devices are increasingly used, creates a framework where environmental issues need to be addressed in compulsory education. This document describes the development of a web-based information system for the display of dashboards with georeferenced environmental indicators in education.

In this project, a web application has been developed that allows the reinforcement of the learning of concepts of subjects such as mathematics or technology in a more dynamic way, achieving a greater motivation of the students and a greater effectiveness in the study. To this end, teachers of these subjects can create and share with students maps with environmental information of a specific location. When students access them, they are presented with a control panel showing the georeferenced environmental sensors installed in the chosen location. Thus, if the user selects one of them, graphs of the values collected for the selected pollutant are shown over time. In addition, teachers can pose questionnaires and exercises related to each of the maps to students. Once these have been completed, the student can store notes and comments on, for example, explanations given by the teacher in the classroom, which can later be retrieved at home.

The objectives set out in the proposal for this End of Degree Project have been achieved. However, due to the health crisis, it has not been feasible to carry out tests in a high school with students in the classroom.

# Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Estudio del estado del arte</b>	<b>3</b>
<b>3. Análisis de requisitos</b>	<b>9</b>
3.1. Requisitos del sistema . . . . .	9
3.1.1. Requisitos funcionales . . . . .	9
3.1.2. Requisitos no funcionales . . . . .	11
3.2. Casos de uso de la aplicación . . . . .	12
<b>4. Diseño del sistema de información</b>	<b>14</b>
4.1. Diseño de la base de datos . . . . .	14
4.2. Diseño de la aplicación . . . . .	18
4.2.1. Arquitectura del sistema . . . . .	18
4.2.2. Diseño de la interfaz de usuario . . . . .	19
<b>5. Implementación y desarrollo</b>	<b>22</b>
5.1. Tecnología utilizada . . . . .	22
5.2. Decisiones de desarrollo e implementación . . . . .	23
5.3. Despliegue del sistema . . . . .	25
<b>6. Análisis de riesgos y mantenimiento de la aplicación</b>	<b>27</b>
6.1. Plan de análisis y gestión de riesgos . . . . .	27
6.2. Plan de mantenimiento . . . . .	30
<b>7. Plan de viabilidad y negocio de la aplicación</b>	<b>33</b>
7.1. Licencia de distribución de la aplicación . . . . .	35
7.2. Plan de marketing . . . . .	37
<b>8. Conclusiones</b>	<b>41</b>
8.1. Trabajo futuro . . . . .	42
8.2. Colaboraciones . . . . .	42
<b>A. Glosario</b>	<b>44</b>
<b>B. Estudio del estado del arte</b>	<b>45</b>
B.1. Frameworks de desarrollo web . . . . .	45
B.2. Sistemas gestores de bases de datos . . . . .	49
B.3. Sistemas de visualización y formato de datos métricos . . . . .	53
B.4. Servidores de Sistemas de Información Geográficos . . . . .	57
B.5. Integración de soluciones existentes . . . . .	60
<b>C. Casos de uso del sistema</b>	<b>63</b>
C.1. Diagrama de casos de uso . . . . .	63
C.2. Descripción de escenarios . . . . .	64
<b>D. Prototipado wireframe</b>	<b>70</b>

<b>E. Gestión del proyecto</b>	<b>77</b>
E.1. Esfuerzos realizados . . . . .	77
E.2. Presupuesto . . . . .	79
<b>F. Guía de instalación y despliegue</b>	<b>82</b>
F.1. Requisitos de instalación . . . . .	82
F.2. Pasos a seguir . . . . .	82
F.3. Fichero de configuración para Nginx . . . . .	84
<b>G. Amenazas, vulnerabilidades y salvaguardas identificadas en el plan de análisis y gestión de riesgos</b>	<b>86</b>
<b>H. Tecnología utilizada</b>	<b>88</b>

## Índice de figuras

1. Página principal de Zooniverse . . . . .	3
2. Página principal de Project Copernicus . . . . .	4
3. Modelo E/R . . . . .	16
4. Modelo relacional . . . . .	17
5. Arquitectura del sistema . . . . .	18
6. Diagrama de secuencias - Dashboard . . . . .	23
7. Diagrama de secuencias - Crear un nanosensor . . . . .	24
8. Diagrama de secuencias - Activar una capa del <i>dashboard</i> . . . . .	24
9. Vista de despliegue . . . . .	25
10. Tabla para el cálculo de la probabilidad . . . . .	28
11. Tabla para el cálculo del impacto . . . . .	29
12. Tabla para el cálculo del riesgo . . . . .	29
13. <i>Cashflows</i> provisionales para la organización . . . . .	33
14. Cuenta de resultados . . . . .	34
15. Gráfico de evolución de los resultados . . . . .	34
16. Balance . . . . .	35
17. Matriz de factores del análisis DAFO . . . . .	38
18. Estrategias a adoptar (1) . . . . .	39
19. Estrategias a adoptar (2) . . . . .	40
20. Gráfico de distribución del tiempo en las tareas realizadas . . . . .	42
21. Tabla comparativa de software GIS . . . . .	59
22. Tabla comparativa de resultados WFS . . . . .	60
23. Diagrama de casos de uso . . . . .	63
24. Login de usuarios . . . . .	70
25. Vista de administración de centro - Gestión de usuarios . . . . .	70
26. Vista de administración de centro - Modificación de datos personales . . . . .	71
27. Vista de docente - Gestión de histórico y mapas . . . . .	71
28. Vista de docente - Gestión de nanosensores propios . . . . .	72
29. Popup de modificación de información . . . . .	72
30. Vista de docente - Modificación de datos personales . . . . .	73
31. Vista de docente - Cuadro de mandos . . . . .	73
32. Vista de docente - Gestión de quiz en el mapa . . . . .	74

33.	Vista de estudiante - Consulta de histórico y de mapas . . . . .	74
34.	Vista de estudiante - Modificación de datos personales . . . . .	75
35.	Vista de estudiante - Cuadro de mandos . . . . .	75
36.	Vista de estudiante - Quiz y notas en el mapa . . . . .	76
37.	Distribución de esfuerzos (1) . . . . .	77
38.	Distribución de esfuerzos (2) . . . . .	78
39.	Distribución de esfuerzos (3) . . . . .	79

## Índice de cuadros

1.	Requisitos funcionales de la aplicación . . . . .	11
2.	Requisitos no funcionales de la aplicación . . . . .	12
3.	Inventario de activos de la organización . . . . .	28
4.	Permisos aceptados por las licencias . . . . .	36
5.	Condiciones impuestas por las licencias . . . . .	36
6.	Limitaciones de cada una de las licencias . . . . .	36
7.	Comparativa de frameworks de desarrollo web . . . . .	48
8.	Comparativa de SGBD . . . . .	52
9.	Comparativa de sistemas de visualización y formato de datos métricos .	56

# 1. Introducción

En la actualidad, la informática está presente en la mayor parte de los contextos y resulta imprescindible para el funcionamiento de la sociedad tal y como la conocemos hoy en día. Ya sea para comunicarnos con otros, en el ámbito laboral o incluso en nuestro tiempo de ocio, es habitual que se utilicen ordenadores, smartphones o tablets. El aula no es una excepción. Con la introducción de las pizarras digitales, poco a poco, el aprendizaje tradicional mediante pizarra y tiza se complementa con el uso de herramientas informáticas. Es aquí donde se enmarca el proyecto desarrollado.

El producto final de este proyecto es una herramienta solvente, completa, que se adapta a las necesidades y requerimientos planteados por los actores involucrados en su uso: profesorado, alumnado y personal de administración y servicios de los centros de educación obligatoria. Además, el proyecto tiene un gran potencial de evolución y mejorar, permitiéndose la adición de módulos que ofrezcan nuevas funcionalidades y posibilidades. Por ello, se ha apostado por el lanzamiento del sistema desarrollado como proyecto de código abierto (OpenSource), el cual se ha denominado **Stics** y se encuentra disponible en: <https://gitlab.com/Cildo98/stics>.

## Motivación y contexto

La creciente preocupación social por el medio ambiente y su conservación es innegable. Desde hace algunos años esta problemática ha tomado relevancia mundial con movimientos como *Fridays For Future* [1], encabezado por la activista sueca Greta Thunberg, quien ha llegado a intervenir en la Cumbre del Clima [2] junto a mandatarios de múltiples países.

Tratándose este de un tema de actualidad, el principal objetivo del presente *TFG* (Trabajo Final de Grado) es el de acercar a los centros educativos de todos los niveles el aprendizaje sobre cuestiones medioambientales mediante su introducción en el aula en asignaturas de ámbito científico tales como matemáticas o tecnología industrial. En concreto, el trabajo abordado ha supuesto el desarrollo de una aplicación web con la que se trabaja en el aula y donde se aplican los conceptos teóricos aprendidos en un ámbito diferente, el medio ambiente.

## Objetivos

Al hacer uso de este sistema, el profesorado puede crear sus propias colecciones de mapas de datos y gráficos relacionados con aspectos medioambientales que compartirá con su alumnado. De este modo, al trabajar con estos, se permite el refuerzo del aprendizaje de, por ejemplo, estadística, pero también del cálculo de derivadas. Asimismo, al formar parte el presente *TFG* de un proyecto más amplio, se ofrece la posibilidad de que el alumnado pueda registrar nuevos sensores medioambientales en el sistema que haya podido desarrollar en asignaturas de carácter práctico, como por ejemplo, tecnología.

El hecho de introducir en el aprendizaje una herramienta y una temática diferentes

a las habituales, puede lograr una mayor implicación por parte del alumnado en el trabajo diario y que, por tanto, los conocimientos queden bien afianzados, al despertar un mayor interés en los estudiantes. Además, esto contribuye a crear enlaces entre las diferentes materias y asignaturas, de modo que el alumnado deje de considerar cada asignatura como algo aislado.

## Metodología

El trabajo desarrollado ha supuesto la realización de labores de investigación y colaboración estrecha con los actores implicados en el uso de la herramienta pero no en el desarrollo de esta, es decir, equipos docentes y alumnado, para obtener una imagen clara y fidedigna de cuáles son las necesidades y carencias a suplir. Para ello, se han llevado a cabo reuniones de tipo “tormenta de ideas” [3] en las que se obtuvieron las ideas básicas de partida. Además, se han llevado a cabo presentaciones parciales del proyecto a lo largo de su desarrollo, especialmente en las primeras etapas, para tratar de identificar lo antes posibles problemas, fallos, y/o malentendidos que pudiesen conducir a errores en el uso de la herramienta.

Esta idea de mejora continua de la calidad del producto se enmarca en la metodología basada en ciclos PDCA (Plan-Do-Check-Act). Esta ha sido escogida ya que permite una mejora de la competitividad del producto mediante la mejora continua de la calidad de este, reduciendo así los costes, optimizando la productividad y, por tanto, la rentabilidad.

A lo largo del presente documento, se presentan las distintas fases que se abordan en el proceso del producto desarrollado. En primer lugar, se presenta el estudio del estado del arte inicial, en el que se analizan cuáles son las opciones principales existentes en el mercado para el propósito deseado, escogiéndose aquellas más adecuadas. Posteriormente, se describe la fase de análisis de requisitos (tanto funcionales como no funcionales) requeridos para el sistema y se identifican los principales casos de uso. La fase de diseño del sistema de información, donde se comenta el diseño de la base de datos realizado, así como el propio diseño de la aplicación, presentándose la arquitectura del sistema y los prototipos *wireframe* realizados se describe en el capítulo 4. Tras esto, en implementación y desarrollo, se comentan aspectos técnicos más detallados, así como algunas decisiones tomadas durante el desarrollo y el despliegue del sistema. A continuación, se presenta un plan de análisis y gestión de riesgos, así como otro de mantenimiento y de viabilidad y negocio de la aplicación. Por último, se incluyen las principales conclusiones y líneas de trabajo futuro.



## 2. Estudio del estado del arte

En cualquier proyecto, el estudio del estado del arte es la primera fase para conocer qué aplicaciones existen ya en el mercado con un propósito similar a la del proyecto que se plantea. Si estas son numerosas, quizás se opte por el abandono del proyecto, o su modificación para distinguirse del resto de sistemas disponibles. En cambio, si no se encuentran productos que suplan las carencias identificadas, el proyecto puede continuar y ser lanzado como algo nuevo e innovador. Además, esta fase permite conocer a los productos competidores y planificar estrategias que diferencien al producto de los que ya existen.

Tras la realización de una amplia búsqueda para identificar productos existentes que cubriesen los requisitos establecidos, no se ha encontrado un sistema que corresponda exactamente con un sistema que cubra las necesidades de la comunidad de usuarios a la que está dirigida este proyecto (centros de enseñanza obligatoria). Únicamente se han encontrado aplicaciones que, de un modo u otro, se acercan a aspectos que componen la idea de la nueva plataforma que se plantea. Las principales herramientas y sistemas identificados se describen a continuación.

### Estudio de aplicaciones de propósito similar

Una de estas es **Zooniverse** [4], tal y como se recoge en su propio sitio web, permite participar en una investigación de vanguardia en muchos campos de las ciencias, las humanidades, etc. En la aplicación, se proponen pequeñas actividades al usuario, como clasificación de imágenes o identificación de elementos en estas, con las que contribuyen a los proyectos de investigación que se plantean, a la vez que sirven como aprendizaje. Sin embargo, no se permite trabajar con información georreferenciada ni con series temporales, lo cual resulta indispensable como requisito para el sistema a desarrollar.

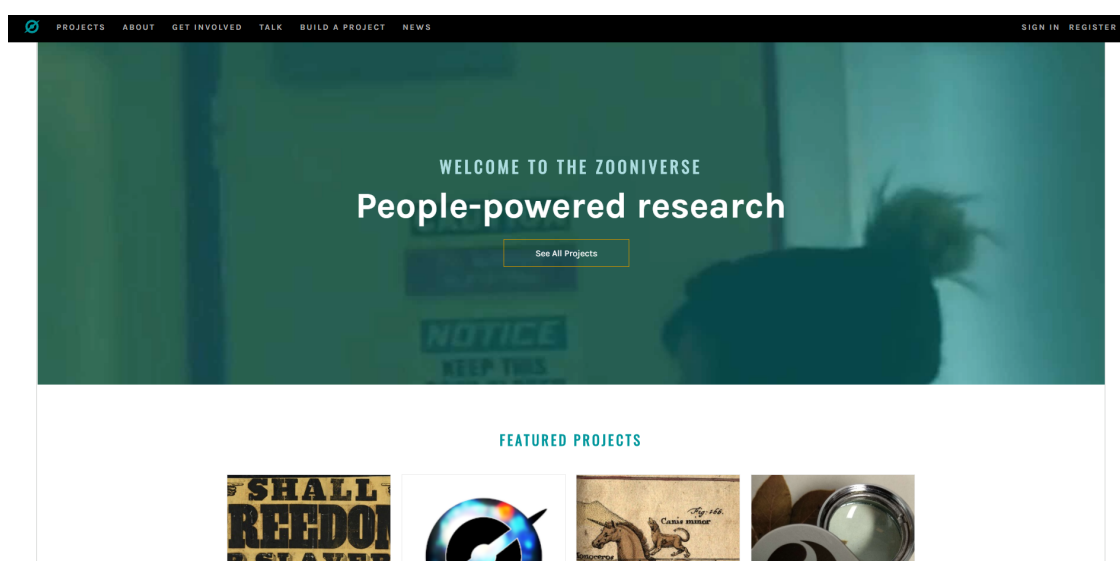


Figura 1: Página principal de Zooniverse

Por otro lado, el **Project Copernicus** [5], es un desarrollo europeo para divulgación de información atmosférica, marina, terrestre, acerca del cambio climático, entre otros. En este proyecto se trabaja con datos similares a los que se desea gestionar en el sistema propuesto. Copernicus proporciona una amplia cantidad de datos e información pero no permite la adaptación y particularización de esta al contexto de enseñanza-aprendizaje en el aula. De nuevo, por tanto, no es un producto que se corresponda plenamente con la idea concebida.

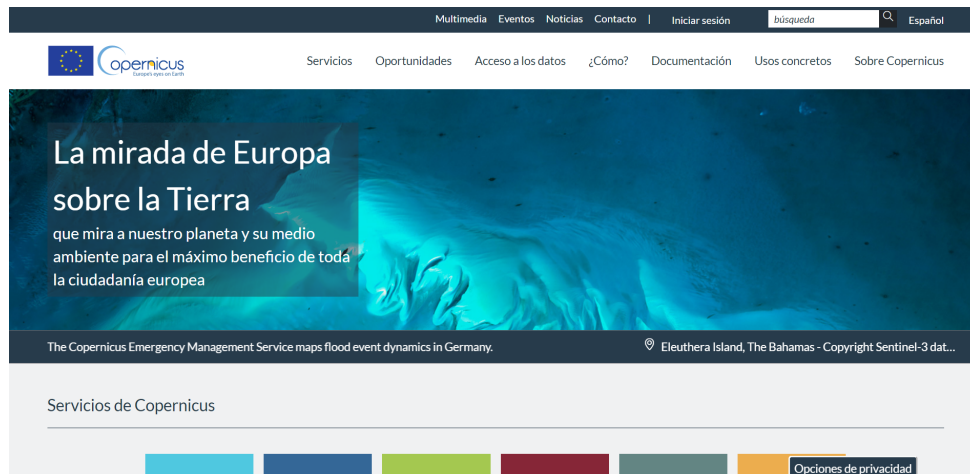


Figura 2: Página principal de Project Copernicus

Una vez detectada, por tanto, la no existencia de productos que puedan adaptarse para satisfacer las necesidades planteadas, se apuesta por el desarrollo de una aplicación. Por ello, en primer lugar se identifican los componentes claves de la que va a estar formada: un sistema gestor de bases de datos y una aplicación web. A continuación, se analizan las tecnologías disponibles para el desarrollo de estos componentes, estudiando las soluciones existentes que puedan cumplir con la función solicitada con el objetivo de escoger la que mejor se ajuste a las necesidades y restricciones impuestas por las características del proyecto. A continuación, se muestra un resumen de los diferentes análisis realizados, que se encuentran descritos de manera pormenorizada en el anexo B.

## Tecnologías y *frameworks* de desarrollo de aplicaciones Web

Un *framework* de desarrollo web es una herramienta que está diseñada para apoyar el desarrollo de aplicaciones web, incluidos los servicios web, los recursos web y las API (*Application Programming Interface*) web. Los frameworks web proporcionan una forma estándar de construir y desplegar aplicaciones web en la World Wide Web [6] y tienen como principal objetivo automatizar las actividades comunes realizadas en el desarrollo de la web. Por ejemplo, muchos marcos web proporcionan bibliotecas para el acceso a bases de datos, marcos de plantillas y gestión de sesiones, y a menudo promueven la reutilización del código.

Por ello, los *frameworks* de desarrollo web en la actualidad se han convertido en una

pieza clave para agilizar el desarrollo. Existe una gran variedad de *frameworks* orientados a diferentes tecnologías Web y lenguajes de programación como por ejemplo JavaScript (JS), Python, Ruby o Java. Cada uno cuenta con ciertas ventajas y más o menos inconvenientes por lo que es imprescindible llevar a cabo un estudio de todos ellos para así poder escoger el más acorde con las necesidades del proyecto a desarrollar. Los *frameworks* escogidos para ser analizados pormenorizadamente han sido Express.js [7], Django [8], Ruby On Rails [9] y Spring [10]. Esta preselección se ha llevado a cabo en base a criterios de variedad y popularidad, para poder valorar los aspectos positivos que proporcionan cada una de las herramientas.

Los criterios que han primado a la hora de la elección final han sido la facilidad de uso, la disponibilidad de documentación suficiente para poder recurrir a ella en caso de duda sobre el funcionamiento del software, la disponibilidad de *plugins* que faciliten el desarrollo porque ya implementen elementos necesarios en el proyecto y, por último, la capacidad de escalar, ya que si el proyecto crece y cuenta con gran aceptación, el tráfico entrante crecería de forma notable. En base a esto, la elección final ha sido Django, por su facilidad de uso, además del conocimiento previo de la herramienta, la gran cantidad de documentación disponible en la red, además de la comunidad de desarrolladores de que dispone, la disponibilidad de múltiples *plugins* fácilmente acoplables al proyecto y la gran capacidad de escalar. En el caso de Express.js, es una herramienta de gran calidad pero puede resultar compleja de usar, además de que hubiera sido necesario aprender también otro software para la parte *frontend*.

En cuanto a Ruby On Rails, del análisis realizado se desprende que es una herramienta compleja de utilizar y que su capacidad de escalar no es buena, por lo que ha sido descartada. Por último, Spring ha sido, de nuevo, descartada por la dificultad de aprendizaje de la herramienta.

## Sistemas Gestores de Bases de Datos

De forma oficial, Django ofrece soporte a múltiples SGBD (Sistemas Gestores de Bases de Datos) que son intercambiables uno por otro en todo momento en función de las necesidades o nueva problemática que se pueda plantear. Concretamente, los SGBD que se soportan según la documentación oficial vigente de Django son los siguientes:

- **PostgreSQL** [11]. Sistema Gestor de Bases de Datos OpenSource con gran capacidad de gestión en entornos de alta concurrencia y variedad de tipos nativos.
- **MySQL** [12]. Sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general (GPL)/Licencia comercial por Oracle Corporation.
- **MariaDB** [13]. Sistema de gestión de bases de datos derivado de MySQL con licencia GPL que introduce dos motores de almacenamiento nuevos. Este SGBD surge a raíz de la compra de Sun Microsystems por Oracle.
- **Oracle** [14]. Oracle Database es un sistema de gestión de base de datos desarrollado por Oracle Corporation. Su dominio en el mercado de servidores empresariales había sido casi total hasta Microsoft SQL Server y de la oferta de otros RDBMS con licencia libre como PostgreSQL o MySQL.

- **SQLite** [15]. SQLite es un sistema de gestión de bases de datos relacional compatible con las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad, en castellano), contenida en una relativamente pequeña biblioteca escrita en C. SQLite es un proyecto de dominio público.

Los SGBD soportados, salvo SQLite, son software solvente en la mayor parte de situaciones, lo cual incluye aplicaciones con un cierto volumen de tráfico generado que puede ser el caso del proyecto actual en función de la evolución y expansión de este. Es por esto por lo que todos han de ser analizados en profundidad para, de este modo, realizar la elección que mejor se ajuste a los requisitos planteados para el sistema, así como a las restricciones de tipo presupuestario que vienen derivadas del hecho de tratarse de un Trabajo Final de Grado, que no cuenta con fuente de ingresos alguna.

SQLite es la única que no se va a proceder a analizar con detalle ya que queda, sin lugar a dudas, fuera de cualquier posibilidad de elección por su lentitud en la ejecución de operaciones en la base de datos.

En esta ocasión, los parámetros analizados para realizar la elección del SGBD han sido principalmente el coste de la herramienta, la escalabilidad, la capacidad de realizar copias de seguridad, la disponibilidad de *plugins* y el trabajo previo con la herramienta, que hace que sea más fácil realizar un buen trabajo desde el principio. Además, el gestor debe tener soporte para almacenamiento y gestión de datos georreferenciados, dado que este tipo de datos son relevantes en el contexto del proyecto.

En base a esto, el sistema gestor de bases de datos escogido para la base de datos principal del proyecto es PostgreSQL por no contar con coste derivado de la utilización de la herramienta, su capacidad en materia de copias de seguridad y de *plugins* disponibles, así como el haber trabajado previamente y, por tanto, conocer la herramienta. Oracle ha sido descartado por su coste inasumible. Por otro lado, MySQL ha sido descartado por haber sido adquirido por Oracle, mencionando algunos artículos la posibilidad de que Oracle abandonara el proyecto en algún momento, además de por poder provocar su uso problemas de integridad en entornos de alta concurrencia en la modificación. Por último, MariaDB ha sido descartada ya que no se había trabajado previamente con ella.

## Sistemas de visualización y formato de datos métricos

Los sistemas de visualización y formato de datos métricos son software específico que, en mayor o menor medida, permiten la conexión con diversas fuentes de datos, incluidas en algunos casos bases de datos de series de tiempo para, posteriormente, crear cuadros de mandos con distintos tipos de gráficos que permitan el análisis rápido y práctico de los datos recogidos. Esto permite, por ejemplo, obtener una radiografía de la evolución de una empresa para así adoptar las medidas necesarias para la consecución de los objetivos empresariales planteados.

En este proyecto, su uso se concreta en su integración en el sitio web de forma que los usuarios de este puedan visualizar y trabajar con los datos medioambientales recogidos en los diversos sensores de contaminantes instalados en las ciudades.

A continuación, se procede al análisis de varios productos existentes en el mercado para este propósito, de forma que se pueda escoger el que mejor satisfaga las necesidades detectadas en el proyecto. En concreto, se trata de los siguientes:

- **Grafana** [16]. Software libre basado en licencia de Apache que permite la visualización y el formato de datos métricos. Permite crear cuadros de mando y gráficos a partir de múltiples fuentes, incluidas bases de datos de series de tiempo.
- **Graphite** [17]. Software OpenSource desarrollado por Orbitz Worldwide, Inc. que monitoriza y genera gráficos de datos numéricos temporales como el rendimiento de computadores.
- **Chart.js** [18]. Biblioteca JavaScript gratuita de código abierto para la visualización de datos, que admite 8 tipos de gráficos.
- **Netdata** [19]. Herramienta para visualizar y monitorizar métricas en tiempo real, optimizada para acumular todo tipo de datos, como uso de CPU, actividad de disco, consultas SQL, visitas a un sitio web, etc.

Para el presente análisis se han tenido en cuenta parámetros como los tipos de gráficos disponibles, el coste de utilización de la herramienta, la disponibilidad de *plugins* y de definición de alarmas, así como si la herramienta cuenta con casos de éxito de indudable reputación. Finalmente, el software escogido ha sido Grafana por contar con multitud de tipos de gráficos, ser gratuita, disponer de *plugins* para añadir funcionalidades, la posibilidad de definición de alarmas y contar con numerosos casos de éxito en empresas grandes en las que se ha implantado la herramienta. En cuanto al resto de soluciones analizadas, Graphite ha sido descartada por no contar con la posibilidad de definición de alarmas, mientras que Chart.js cuenta con un catálogo de gráficos disponibles bastante limitado. Por último, no fue posible encontrar casos de éxito con Netdata, lo cual hace dudar de la verdadera utilidad de la herramienta.

## Servidores de Sistemas de Información Geográficos

Blakemore, en 1986, definió un Sistema de Información Geográfico, GIS por sus siglas en inglés, como “paquetes de software que integran el almacenamiento, manipulación, análisis, modelado y mapeado de información espacial digital”. Burrough, por su parte, definió GIS como “conjuntos de herramientas potentes para la recolección, transformación y muestra de datos espaciales” [20]. Las aplicaciones GIS permiten al usuario definir consultas interactivas, analizar la información espacial recuperada, editarla y compartir los resultados, lo cual resulta fundamental en el proyecto planteado. De ahí que se hayan analizado este tipo de sistemas.

Los productos analizados en este caso han sido tres: gvSIG [21], ArcGIS Server [22] y GeoServer [23]. Para este análisis, los criterios que han primado han sido, como en análisis previos, los casos de éxito en el uso de la herramienta, así como los protocolos soportados y estudios sobre la velocidad y el tamaño del fichero final en la realización de peticiones bajo WFS (*Web Feature Service*) al servidor de Sistemas de Información Geográficos.

gvSig fue descartado por tratarse de una herramienta de ámbito regional, de la Comunidad Valenciana, de la cual no se han identificado más casos de éxito que el propio desarrollo en el ámbito de la comunidad autónoma. Entre ArcGIS server y Geoserver ha primado el mayor número de protocolos de entre los definidos por el OGC (Open Geospatial Consortium) [24] soportados por este último, además del estudio comentado con anterioridad, en el que los datos devueltos por ArcGIS Server son mucho más grandes y conllevan mucho más tiempo de procesamiento que los proporcionados por GeoServer. Por todo ello, además del coste asociado al uso de ArcGIS, en el cual no se incurre si se utiliza GeoServer, es esta última la herramienta escogida para el proyecto.

Para la visualización e interacción con la información geográfica servida a través de GeoServer se han considerado diferentes librerías JavaScript de código abierto. En concreto, se han considerado: OpenLayers [25] y Leaflet [26]. Ambas tecnologías hacen posible la gestión de mapas en web así como la superposición de capas de datos provenientes de, por ejemplo, GeoServer, haciendo uso para ello del formato GeoJSON. En este caso, la elección ha sido Leaflet por su mayor facilidad de aprendizaje, la claridad de su documentación y la gran cantidad de *plugins* para añadir funcionalidad, lo cual lo hace óptimo para un proyecto cuyo propósito es general, sin necesidades demasiado concretas.

## Interoperabilidad con soluciones existentes

Uno de los principios básicos en el desarrollo de software es el de no reinventar lo que ya está hecho, sino aprender a usarlo e integrarlo en el proyecto. Con este fin, se realizó una amplia búsqueda de soluciones en distintos ámbitos, que, de un modo u otro pudieran ser útiles en el trabajo. De dicha investigación se extrajo la conclusión de que podría ser interesante la integración de algunos aspectos desarrollados en **EINAmbiente**, proyecto realizado en las prácticas de Sistemas de Información I, por ejemplo, el desarrollo de cuestionarios por parte del profesorado para su realización por el alumnado que se encuentre a su cargo, además de la posibilidad de que el alumnado almacene notas y comentarios en un mapa concreto para su posterior consulta.

En segundo lugar, se estudió la integración de la solución **Building Blocks**, desarrollada por la Comisión Europea para facilitar el uso de herramientas a nivel europeo por cualquier ciudadano de la Unión. No obstante, fue descartado ya que es requisito indispensable para disponer de acceso al sistema ser funcionario público de la Administración de un país miembro. Por último, se barajó la posibilidad de integración del sistema con plataformas de gestión académica ya implantadas en los centros, como **SIGAD**. No obstante, al no disponer de una API pública utilizable por un usuario corriente, dicha integración quedó reducida a trabajar con un formato de fichero de datos lo más similar posible al extraído de dichas plataformas de gestión.

## 3. Análisis de requisitos

El análisis de requisitos es el proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, de hardware o de software, así como el proceso de estudio y refinamiento de dichos requisitos. Por su parte, un requisito es una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado, así como las condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificación. En esta sección se van a abordar por tanto, en primer lugar, la identificación de los requisitos del sistema, tanto funcionales como no funcionales, para posteriormente identificar y describir pormenorizadamente los casos de uso del sistema.

### 3.1. Requisitos del sistema

A continuación se introducen los requisitos funcionales y no funcionales emanados de las necesidades detectadas durante las diferentes reuniones mantenidas con una representación del público objetivo.

#### 3.1.1. Requisitos funcionales

Código	Descripción
RF1	El sistema debe permitir el registro correcto de los usuarios según su rol en este.
RF1.1	Se distinguirán dos tipos básicos de usuarios: el profesorado y el alumnado.
RF1.2	Además, se define un tipo especial de usuario: la administración del centro. La funcionalidad requerida para este usuario es la de asignar nuevo alumnado y profesorado a sus correspondientes grupos en el sistema.
RF1.3	El sistema permitirá la modificación de los datos personales del usuario, tanto en el caso de profesorado como alumnado.
RF1.3.1	El sistema habilitará un mecanismo mediante el cual el usuario pueda ejercer sus derechos de acceso, rectificación, cancelación y oposición a sus datos.
RF2	El sistema debe integrar un mecanismo de modificación en caso de contraseña olvidada.
RF3	El sistema debe permitir al profesorado la definición de cuadros de mandos.
RF3.1	El profesorado seleccionará el grupo o alumnado concreto que tiene acceso al nuevo cuadro de mandos.
RF3.2	El sistema debe permitir al profesorado la elección de la localización del mapa con los indicadores medioambientales a mostrar, el tipo de estos y el nombre del mapa que se define.
RF3.2.1	Los posibles tipos de indicadores a mostrar serán: sensores oficiales y nanosensores contruidos por los usuarios del sistema.
RF4	El sistema mostrará un mapa donde se encontrarán superpuestos los indicadores seleccionados.

RF4.1	Tras clicar en un indicador concreto, se mostrará información temporal detallada sobre este en los distintos gráficos que se ofrecen al usuario.
RF4.2	El profesorado debe poder elegir los tipos de gráficos que desea hacer visibles de entre los disponibles.
RF4.2.1	Los gráficos disponibles serán los siguientes: gráfico de evolución temporal del contaminante en la estación, gráfico de evolución temporal media del contaminante y último valor registrado.
RF5	El sistema debe permitir al profesorado la creación de cuestionarios para su realización por el alumnado.
RF5.1	El sistema debe permitir al profesorado la elección del tipo de pregunta a formular.
RF5.1.1	Los tipos de pregunta posibles son los siguientes: pregunta de respuesta textual, pregunta de respuesta única y pregunta de respuesta múltiple.
RF5.2	El sistema debe permitir al profesorado la impresión de los cuestionarios preparados para su realización en el aula o en un ámbito externo a la aplicación.
RF5.3	El sistema debe permitir al profesorado la impresión de la corrección de los cuestionarios preparados.
RF6	El sistema debe permitir al alumnado la visualización de los cuadros de mandos generados por el profesorado, además de ofrecer la posibilidad de interactuar con ellos.
RF6.1	La interacción con los cuadros de mandos será variada y determinada por el profesorado, que podrá incluir preguntas o permitir que el alumnado guarde notas o comentarios asociados al mapa concreto que está visualizando.
RF7	El sistema debe permitir al profesorado la modificación de los cuadros de mandos que ha creado.
RF7.1	El profesorado podrá modificar los datos que caracterizan a un cuadro de mandos: localización y nombre de este y tipos de sensores a mostrar.
RF8	El sistema debe permitir al profesorado la eliminación de los cuadros de mandos que haya definido con anterioridad.
RF9	El sistema debe permitir al profesorado dar de alta nanosensores creados, por ejemplo, en el aula por el alumnado en el marco de asignaturas como Tecnología Industrial.
RF9.1	Los elementos que caracterizan a un nanosensor son: sus coordenadas geográficas, el modelo y el estado en que se encuentra.
RF9.1.1	Adicionalmente se podrá añadir una breve descripción de este si se considera necesario.
RF9.1.2	El punto donde se ubica el nuevo nanosensor debe poder ser escogido mediante un mapa que se muestre al usuario o mediante la introducción manual de sus coordenadas.
RF10	El profesorado debe poder modificar todos los elementos que caracterizan a un nanosensor en todo momento.
RF11	El profesorado debe poder dar de baja un nanosensor en todo momento.
RF12	El sistema debe permitir a todos los usuarios la consulta de información histórica mediante la elección del curso escolar en que estén interesados.



RF12.1	En la vista de consulta de histórico, la administración del centro no podrá realizar cambios más allá de la eliminación de alumnado y profesorado, siendo esta una funcionalidad principalmente de consulta de información.
RF13	El usuario encargado de la gestión del centro deberá poder llevar a cabo tareas de gestión de alumnado de forma masiva.
RF13.1	Las operaciones disponibles serán las de crear cuentas de estudiante y asignar alumnado a grupos, para lo cual se utilizará como fuente de datos ficheros de texto de formato .csv.

Cuadro 1: Requisitos funcionales de la aplicación

### 3.1.2. Requisitos no funcionales

Código	Descripción
RNF1	Para garantizar la accesibilidad del sitio web a todos los usuarios, a la hora de mostrar la información detallada de los indicadores, el formato escogido será el adecuado para su comprensión sin dificultad por parte del alumnado.
RNF1.1	Se prestará especial atención al juego de colores utilizado para ello, de forma que resulte lo más accesible y coherente posible, al mismo tiempo que deberá mantener la estética y el estilo de diseño que se utilice en el resto de la aplicación.
RNF2	El sitio deberá ser fácil de usar, de modo que su uso no suponga una carga de trabajo notable para el profesorado.
RNF3	El registro en el sitio se llevará a cabo mediante usuario y contraseña para todos los tipos de usuario.
RNF3.1	El registro del centro educativo y del personal administrativo en el sistema se llevará a cabo de forma manual por parte del personal de mantenimiento de la aplicación una vez se firme el contrato correspondiente por ambas partes.
RNF4	La información de carácter personal se almacenará en el sistema de forma segura para cumplir con lo estipulado en el RGPD (Reglamento General de Protección de Datos) [27] en materia de protección de datos.
RNF4.1	Se mostrará un elemento visible en el sitio web avisando del uso de “cookies” por parte del sistema, que el usuario deberá aceptar para continuar la navegación.
RNF4.2	La navegación en el sitio web se llevará a cabo en todo caso utilizando HTTPS (Protocolo seguro de transferencia de hipertexto) [28], de forma que el intercambio de información se lleve a cabo siempre de forma segura.
RNF4.3	Se deberá desarrollar una sección en el sitio web en la que se informe de la licencia bajo la cual se distribuye el producto, así como de la política de privacidad que sigue el mismo para que el usuario esté informado en todo momento.
RNF5	El sitio web será accesible en múltiples idiomas, para alinearse con la estrategia de educación bilingüe presente en un número creciente de centros educativos.

RNF6	La aplicación web podrá ser instalada en un centro concreto, haciendo uso de los recursos de los que este dispone. No obstante, se ofrecerá también en forma de una plataforma SaaS ( <i>Software as a Service</i> ) [29], para lo cual se prevee su despliegue en la nube.
RNF6.1	La base de datos del sistema deberá ser planteada teniendo en cuenta esta característica.
RNF7	El sitio web deberá incluir una sección a modo de publicidad y marketing del sistema en la que se informará de las principales características y ventajas de este.
RNF8	Con el fin de reducir la curva de aprendizaje del sistema para sus usuarios, se crearán una serie de videotutoriales enlazados desde el sitio web en donde se explican los principales casos de uso de este.

Cuadro 2: Requisitos no funcionales de la aplicación

## 3.2. Casos de uso de la aplicación

Como se ha comentado con anterioridad, se han identificado tres tipos de usuario del sistema: el alumnado, el profesorado y la administración de centro. A continuación, se ofrece un listado de los casos de uso principales del sistema. Para consultar el diagrama de casos de uso del proyecto ver Anexo C, así como la descripción de escenarios asociada a los casos de uso aquí descritos. Los principales casos de uso identificados, agrupados por tipos de usuario, son:

### ■ Alumnado

- Creación de notas en un mapa por parte del alumnado.
- Visualización de la información medioambiental de un sensor.

### ■ Profesorado

- Creación de mapa.
- Modificación de mapa.
- Compartir mapa.
- Creación de nanosensor.
- Modificación de nanosensor.
- Definición de una nueva pregunta en el cuestionario de un mapa.
- Permitir al alumnado la visualización de un nuevo gráfico en un mapa.
- Impresión de cuestionario.

### ■ Administración del centro

- Modificación del estado del estudiante.
- Adición de alumnado a un grupo.
- Carga masiva de alumnado a grupos.
- Creación de un nuevo grupo.

Por último el sistema debe permitir, al alumnado y profesorado de los centros educativos, el registro en el sistema. Asimismo, todos los usuarios de la aplicación deben poder iniciar sesión en esta, recuperar su contraseña en caso de haberla olvidado, modificar sus datos personales y consultar información histórica asociada a ellos.

## 4. Diseño del sistema de información

El Glosario de Términos de Calidad e Ingeniería del Software de la Asociación Española de Control de Calidad [30] define la fase de diseño del software como “el proceso de definición de la arquitectura software: componentes, módulos, interfaces, procedimientos de prueba y datos de un sistema que se crean para satisfacer unos requisitos especificados”. Esta sección por tanto establece cómo se van a materializar los requisitos planteados en la sección anterior. En primer lugar, se va a abordar el diseño de la capa de persistencia y gestión de datos mediante un modelo entidad/relación. Posteriormente, se describen los puntos clave del diseño del producto mediante la descripción de la arquitectura del sistema y la inclusión de diagramas de clases.

### 4.1. Diseño de la base de datos

El diseño de una base de datos es el proceso por el cual se lleva a cabo la organización de los datos según un modelo de base de datos. El diseñador determina qué datos deben ser almacenados y cómo se interrelacionan los elementos de datos. El sistema de gestión de la base de datos gestiona los datos en consecuencia. El diseño de la base de datos implica la clasificación de los datos y la identificación de sus interrelaciones.

Este es un proceso clave en el desarrollo del proyecto ya que de él depende la operatividad del sistema, la complejidad del desarrollo de la capa de lógica de negocio de la aplicación, así como la eficiencia del sistema en su conjunto cuando se reciben peticiones que requieren de la recuperación de información de la base de datos. Además, un buen diseño de la base de datos mejorará su mantenibilidad y facilitará su modificación en base, por ejemplo, a nuevas funcionalidades que se deseen ofrecer a los usuarios.

Por todo ello, en primer lugar, se ha desarrollado un modelo conceptual mediante el uso del modelo Entidad/Relación (E/R), el cual se puede observar en la figura 3. Durante el proceso de definición del modelo se han aplicado las técnicas necesarias para obtener las características comentadas con anterioridad, que, a continuación, se pasa a comentar en detalle:

- Los diferentes tipos de usuario han sido modelados utilizando para ello la herencia entre entidades. De este modo, se desarrolla un modelo más coherente, evitando duplicidades y, sobre todo, facilitando en gran medida la ampliación del sistema mediante la inclusión de nuevos tipos de usuarios, que podrían ser definidos sin realizar modificación alguna en el resto.
- Este mismo principio ha sido aplicado en el caso de los diferentes tipos de preguntas planteadas. Así pues, nuevamente, si se deciden enriquecer los cuestionarios del sitio web con nuevos tipos de preguntas, bastaría con hacer que la nueva entidad herede de la entidad preexistente Pregunta.
- Se ha definido una relación ternaria entre Grupo, Curso y Usuario para la gestión del histórico de cursos académicos y el acceso al contenido desarrollado durante los mismos.

- Se ha definido una herencia entre `AbstractUser` y `Usuario` que, a priori, podía haberse evitado fusionándolas en una sola. No obstante, `AbstractUser` es una entidad propia del *framework* escogido, es decir, Django, que incluye multitud de atributos listos para utilizar y que además incorpora mecanismos de seguridad como por ejemplo, el almacenamiento seguro de los datos de usuarios. Como esto es un aspecto crítico que ya ha sido resuelto por el *framework*, se ha decidido mantenerla y utilizarla.
- Se ha definido otra entidad, `Dashboard`, con dos atributos e inconexa con el resto de entidades y relaciones. Esta entidad es necesaria para controlar la existencia de cuadros de mandos en Grafana de los diferentes contaminantes en cada una de las estaciones de medición de contaminación ambiental.

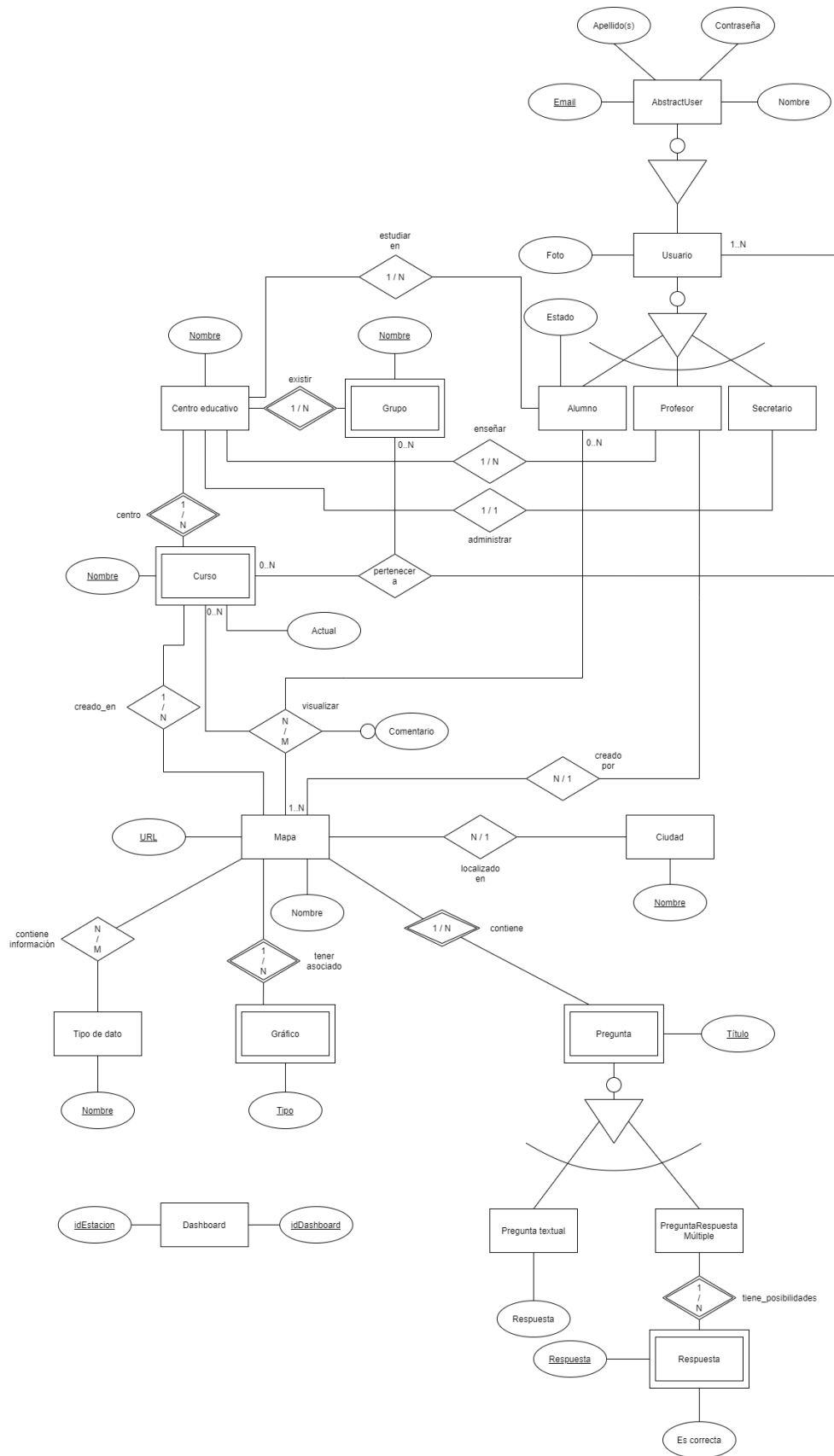


Figura 3: Modelo E/R

Una vez obtenido el modelo entidad/relación de la base de datos anterior, el siguiente paso en todo proceso de diseño de una base de datos es la traducción de este modelo conceptual a un modelo lógico que facilite su implementación. En este caso se ha apostado por el modelo relacional, dado que es ampliamente conocido y usado y se ajusta a las necesidades del sistema. El modelo relacional para el presente proyecto se muestra a continuación (ver figura 4).

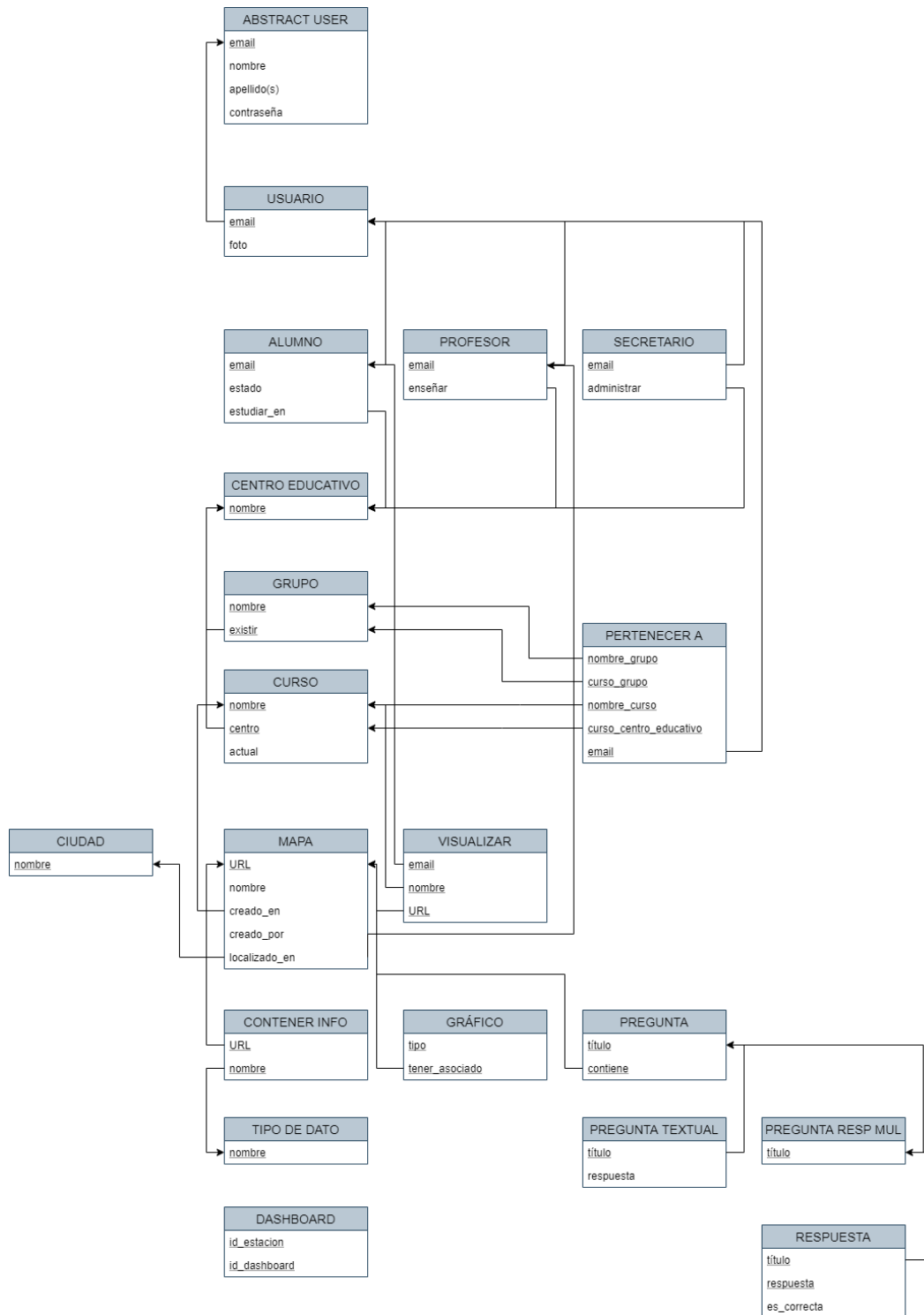


Figura 4: Modelo relacional

Para verificar la buena traducción a modelo relacional realizada, fue preciso comprobar la normalización del modelo. En primer lugar, el modelo se encontraba ya en primera forma normal debido a la inexistencia de atributos multivaluados. Asimismo, al no existir dependencia alguna, dada una tabla concreta, entre atributos, se verificaba el cumplimiento de la segunda y tercera forma normal.

## 4.2. Diseño de la aplicación

Como ya se ha comentado, la fase de diseño del software es una de las fases principales en el proceso de desarrollo. Si se realiza un buen diseño en el *backend* del sistema, por ejemplo, en la base de datos y la lógica de la aplicación, se facilitará el desarrollo de una aplicación solvente y el cumplimiento de los requisitos no funcionales establecidos en la fase de análisis. Si, por el contrario, el diseño está mal planteado, el software será lento y contendrá errores que harán que el usuario final no se sienta cómodo y seguro en él y lo abandone.

Esto mismo ocurre a nivel de diseño de la capa de vista de la aplicación. Si al acceder al sistema el usuario encuentra una interfaz simple, que comprende con facilidad, que no plantea problemas de usabilidad y que presenta una estética actual, moderna y sin errores, habrá más posibilidades de retener al usuario en nuestro sistema y evitar que opte por soluciones alternativas.

Por todo ello, se ha apostado por un diseño modular basado en patrones de diseño [31] y la creación de prototipos de la interfaz, que se han validado con los usuarios finales. El diseño y prototipos realizados se describen a continuación.

### 4.2.1. Arquitectura del sistema

El sistema desarrollado está compuesto por 4 componentes principales que hacen uso de dos bases de datos para la gestión persistente de estos (ver figura 5).

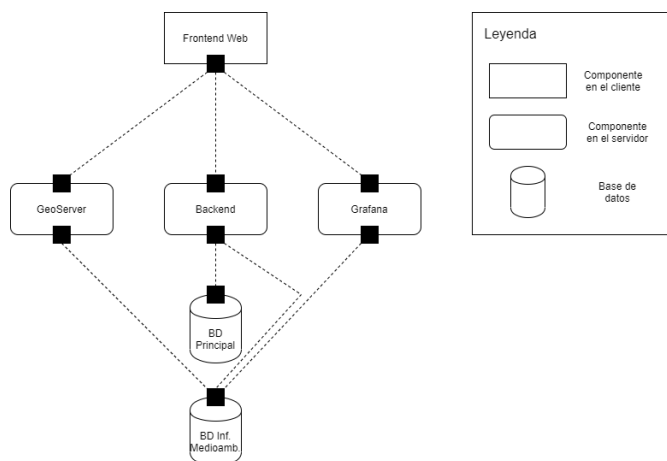


Figura 5: Arquitectura del sistema



Además del *frontend* y *backend* implementados, se hace uso del servidor GeoServer para proporcionar servicios OGC sobre datos georreferenciados almacenados en la base de datos de información medioambiental del sistema y Grafana para la gestión de gráficos e imágenes. En concreto, se identifican los siguientes elementos:

- **Backend.** Por un lado se encuentra el *backend*, que es el elemento clave del sistema. Este interactúa con la base de datos principal para recuperar la información relativa a usuarios y mapas que hayan creado. Además interactúa con otra base de datos que almacena los datos provenientes de los sensores medioambientales para la definición y consulta de nanosensores creados por los usuarios de la aplicación.
- **Base de datos principal.** Como ya se ha comentado, esta base de datos almacena la información de los usuarios del sitio web y la generada fruto de su interacción con el mismo, ya sean mapas, grupos, centros educativos, etc.
- **Base de datos de información medioambiental.** Es una base de datos pre-existente, implementada en PostgreSQL con el *plugin* PostGIS [32] para funcionalidad geoespacial.
- **GeoServer.** Servidor de código abierto con el que el navegador de la máquina cliente establece comunicación utilizando estándares abiertos para recuperar las capas de datos de sensores oficiales y nanosensores.
- **Grafana.** Herramienta de software libre para visualizar datos de series temporales. El *frontend* del sistema establece comunicación con él para recuperar los gráficos referentes a los valores de contaminación registrados para el sensor y contaminante seleccionados por el usuario.

La arquitectura del backend posee tres capas y sigue esencialmente un patrón de tipo MVC (Modelo-Vista-Controlador) para facilitar la creación de nuevas interfaces y la modificación de la lógica de negocio de la aplicación:

- **Modelo.** Almacena la información que debe gestionarse de forma persistente y gestiona el acceso y actualización de la misma. Envía a la vista la información solicitada en cada momento para que sea mostrada al usuario.
- **Vista.** Presenta el modelo en un formato adecuado para la interacción con el usuario.
- **Controlador.** Responde a eventos e invoca peticiones al modelo cuando se hace alguna solicitud que requiera acceso a la información persistente.

#### 4.2.2. Diseño de la interfaz de usuario

La experiencia de usuario es aquello que una persona percibe al interactuar con un producto o servicio. Se dice que se obtiene una buena experiencia de usuario o UX, por sus siglas en inglés (*User Experience*), al enfocarse en el diseño de productos útiles y usables. Esto es un aspecto clave para la satisfacción del usuario con este.

Este concepto se aplica indudablemente en el diseño de interfaces de aplicaciones web,

y, por tanto, en el proyecto actual. Es por esto que el diseño web no es simplemente la implementación de una interfaz, ya que intervienen aspectos como el diseño gráfico web, diseño de interfaz y experiencia de usuario, con elementos como la navegabilidad, interactividad, usabilidad, etc.

En el proceso de desarrollo de la interfaz del sitio web hay numerosos aspectos que se han tenido en cuenta. El primero de ellos es cubrir las necesidades de los usuarios. Es necesario comprender lo que los usuarios buscan en un diseño, para lo cual es imprescindible implicarlos desde las fases más tempranas del desarrollo.

Además, es necesaria una jerarquía clara en la organización del contenido en el sitio web y mantener la consistencia a lo largo de la aplicación. Los usuarios se familiarizan con un nuevo producto mucho más rápido si este incluye elementos que le recuerden a otros productos que ya conoce o si la disposición de los items le recuerda a los de otras webs. Por otro lado, es importante asegurar la accesibilidad del sitio web ya que este ha de ser usable por tanta gente como sea posible, lo cual incluye a personas con discapacidad.

Otros conceptos importantes que se han aplicado en el diseño realizado es, por ejemplo, utilizar un lenguaje simple y correcto, comprensible con facilidad por el usuario. Asimismo, es necesario hacer sentir al usuario que mantiene el control sobre sus acciones en todo momento mediante la inclusión de iconos o botones por ejemplo de tipo “Cancelar” o “Descartar cambios”.

Para lograr estos principios, como se ha comentado con anterioridad, ha sido necesario implicar a los usuarios desde el principio del proceso de diseño. Además, es relevante destacar que se ha incluido a todos los tipos de usuarios, de forma que pudiesen evaluar cada uno los elementos de la interfaz correspondientes a su rol, ya que cada uno presenta características distintas. A continuación se presenta la técnica utilizada para el prototipado rápido de la aplicación en sus primeras fases.

## Prototipado wireframe

Un *wireframe* de un sitio web es una guía visual que representa el esqueleto de un sitio web. Los *wireframes* se crean con el propósito de ordenar los elementos para lograr mejor un propósito particular. El *wireframe* representa el diseño de página o la disposición del contenido del sitio web, incluidos los elementos de la interfaz y los sistemas de navegación, y cómo funcionan juntos. El *wireframe* suele carecer de estilo tipográfico, color o gráficos, ya que el enfoque principal radica en la funcionalidad, el comportamiento y la prioridad del contenido. Los *wireframes* pueden ser dibujos a lápiz o bocetos en una pizarra, o pueden ser producidos mediante una amplia gama de aplicaciones de software libre o comercial. Un *wireframe* se centra en:

- Las funciones disponibles.
- La prioridad de la información y las funciones.
- Las reglas para mostrar la información.
- El efecto de los elementos en la pantalla.

Además, con la realización de un prototipo wireframe, se logra una versión primitiva del sitio web que ya puede ser testada por usuarios para, de este modo, poder corregir errores de diseño en una fase muy temprana en la que el impacto sobre la planificación sea casi inexistente. En este proyecto, se ha realizado el prototipado wireframe utilizando para ello la herramienta Balsamiq Cloud [33]. Los prototipos de las diferentes pantallas se muestran en el anexo D.

## 5. Implementación y desarrollo

La implementación del software es el proceso en el cual los desarrolladores programan el código necesario para llevar a cabo los diseños planificados con anterioridad y para que su funcionalidad responda a lo planteado en la fase de análisis de requisitos. En esta fase toma especial relevancia la metodología de desarrollo de software escogida para el proyecto, en este caso, el ciclo de Deming, o, también conocido como ciclo PDCA.

En cada ciclo se busca obtener *feedback* del cliente, de modo que se pueda acometer la fase *Act* de forma útil y planificar la siguiente fase del desarrollo. Se ha puesto especial énfasis en realizar pruebas del software en partes pequeñas, para, de este modo, asegurar su buen funcionamiento como paso previo a su integración con el resto del software.

### 5.1. Tecnología utilizada

En las primeras fases del desarrollo del presente proyecto se realizaron comparativas en componentes clave del software que se podría escoger para su realización, por ejemplo, el framework de desarrollo web, la base de datos, el software de visualización y formato de datos métricos y, por último, el servidor de Sistemas de Información Geográficos. No obstante, hay elementos de menor importancia, o, en otras palabras, de menor impacto en el producto final, que, sin embargo, lo enriquecen y aportan valor añadido a este. En muchos casos, se trata de extensiones o módulos del software principal escogido. A continuación, se procede a nombrar estos elementos ya que han sido también objeto de estudio para saber si era conveniente su utilización, encontrándose el análisis realizado en el anexo H.

- PostgreSQL - *Unaccent*. *Unaccent* es una extensión de PostgreSQL que permite consultar datos en la BD independientemente de tildes y otras grafías.
- OpenLayers o Leaflet. Se tratan de librerías JavaScript que permiten gestionar la creación de mapas y la superposición de datos en ellos.
- WFS. Protocolo definido por el *OGC* que ofrece una interfaz de comunicación que permite interactuar con los objetos geográficos servidos por el estándar WFS.
- Multilenguaje en Django. El framework Django ofrece de forma nativa capacidad de multilenguaje para una aplicación web. Actualmente, el sitio desarrollado se ofrece en castellano, inglés, francés e italiano.
- PostgreSQL - TimescaleDB. Base de datos OpenSource construída para analizar datos de series temporales con la potencia y practicidad de SQL.
- Git - Software de control de versiones<sup>1</sup>.

---

<sup>1</sup>El software se encuentra disponible en el repositorio público: <https://gitlab.com/Cildo98/stics> bajo licencia GNU GPL v3.

## 5.2. Decisiones de desarrollo e implementación

A continuación, se muestran las principales decisiones que han sido adoptadas durante la implementación del sistema. El primer bloque se refiere a cuestiones técnicas, mientras que el segundo son principios estéticos que han regido a la hora de implementar la interfaz.

Durante el desarrollo del proceso de implementación del proyecto, se han sucedido algunos inconvenientes o necesidades que han provocado la necesidad de toma de determinaciones en cuanto a cómo diseñar la secuencia de acciones para implementar una funcionalidad. Es el caso, por ejemplo, de la secuencia de acciones para mostrar los gráficos de un contaminante en la estación seleccionada por el usuario a partir de Grafana. Tras su uso, la API de este producto se ha revelado como algo simplista, que limita la capacidad de acción del desarrollador, o, al menos, dificulta su tarea.

El problema en el proyecto ha sido que no existe en la API un recurso dedicado expresamente ni a la creación de *dashboards* ni de paneles de *dashboard*, sino que si se desea generar contenido dinámicamente mediante el uso de la API de Grafana, este ha de ser en base a contenido ya creado. Esto se ha traducido en la definición de un *dashboard* de ejemplo utilizando para ello la interfaz gráfica de Grafana, el cual es recuperado cuando se detecta que el *dashboard* de un contaminante en una estación no ha sido creado con anterioridad; se modifica para su adaptación a los datos del contaminante deseado, siendo este último el que se envía a Grafana para su almacenamiento, mostrándose asimismo al cliente de la aplicación. Este *workflow* viene definido por el siguiente diagrama de secuencias.

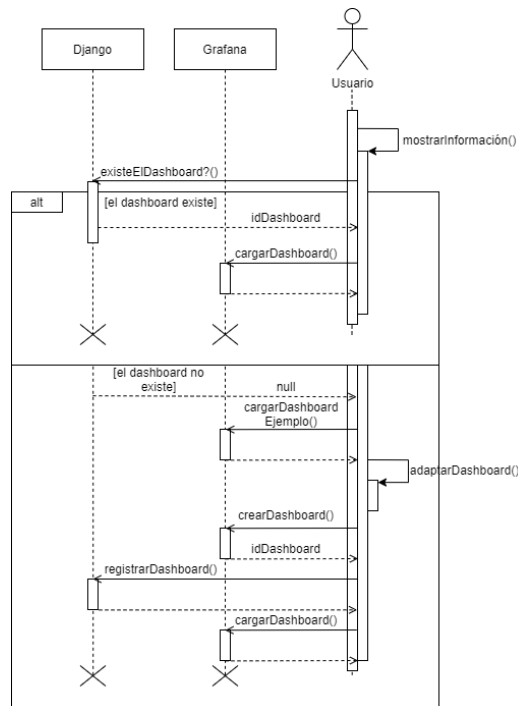


Figura 6: Diagrama de secuencias - Dashboard

A continuación se incluye otro diagrama de secuencias para ilustrar la interacción con la base de datos TimescaleDB.

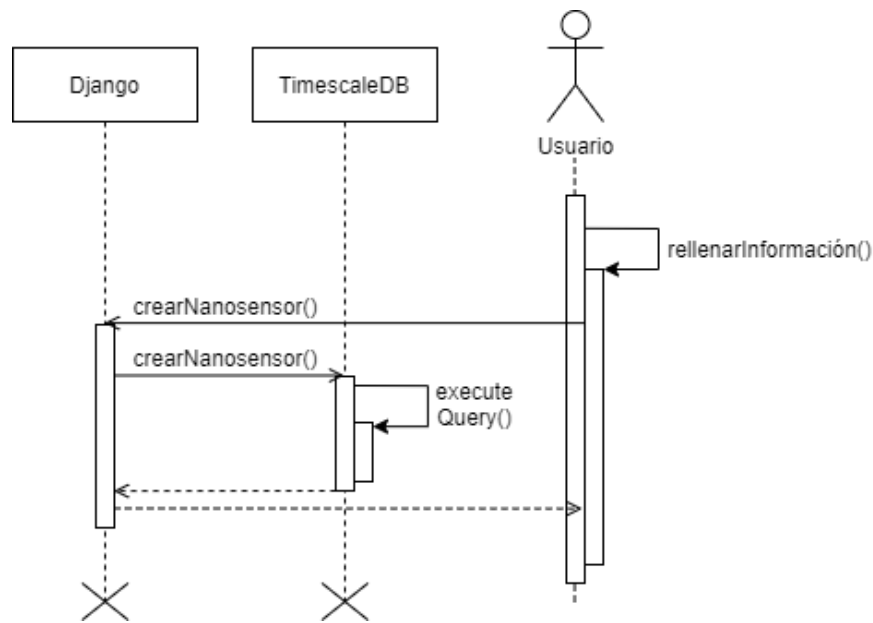


Figura 7: Diagrama de secuencias - Crear un nanosensor

Por último, a continuación se incluye un diagrama de secuencias para ilustrar la secuencia de acciones en la interacción con GeoServer para la carga de una capa de datos en el mapa del cuadro de mandos.

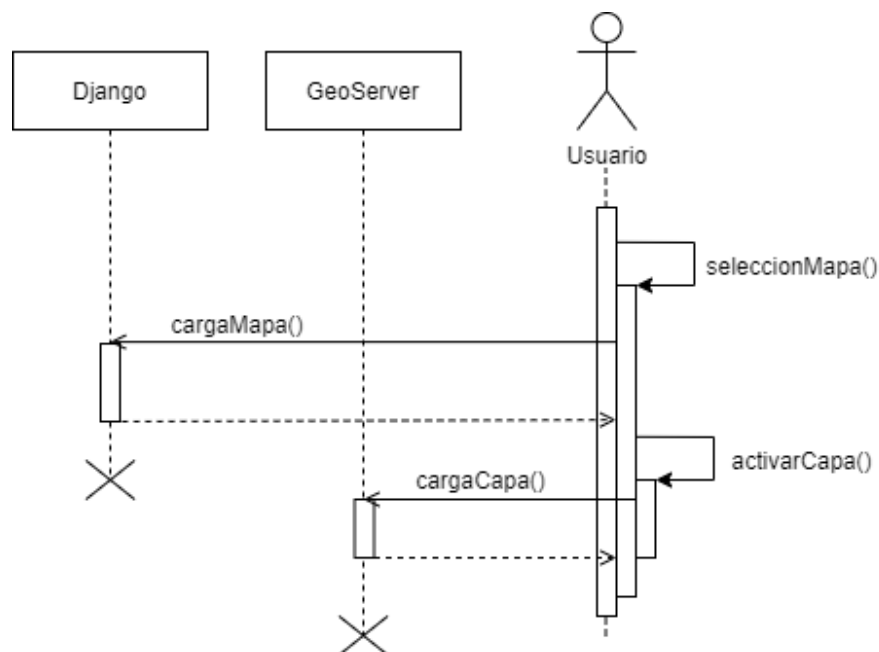


Figura 8: Diagrama de secuencias - Activar una capa del *dashboard*

## Patrón MTV

El uso de Django como *framework* de desarrollo web hace que el *backend* del sistema siga una variante del patrón MVC en su arquitectura. El *framework* Django ofrece una implementación del patrón Modelo-Vista-Controlador que utiliza la terminología “Templates” para “Views” y “Views” para “Controller”. Las “templates” se relacionan con la vista en el patrón MVC ya que se refieren a la capa de presentación y esencialmente gestiona el contenido a mostrar y cómo mostrarlo al usuario.

Además, la adopción del patrón MTV por el framework de desarrollo web Django ofrece las siguientes ventajas:

- Desarrollo rápido. Como esta arquitectura separa los elementos en distintos componentes, es fácil de paralelizar el trabajo en diferentes aspectos de la misma aplicación entre varios desarrolladores.
- Facilidad de modificación. Este patrón permite modificar un componente sin necesidad de modificación alguna en el resto. Esto hace más simple el mantenimiento y mejora del sistema.

### 5.3. Despliegue del sistema

La vista de despliegue en arquitectura software es utilizada para proporcionar una base para comprender la distribución física del sistema en uno o varios nodos. A continuación, se incluye esta vista aplicada en el proyecto desarrollado.

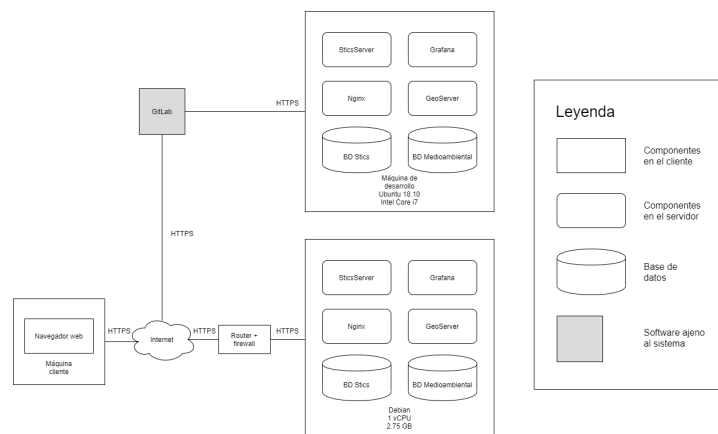


Figura 9: Vista de despliegue

En la figura anterior se puede observar los elementos que componen el sistema en detalle. Por un lado, se encuentra la máquina cliente, cuyo único requisito es que disponga de un navegador web y que, lógicamente, esté conectada a Internet. Por otro lado, se presentan dos máquinas.

Una de ellas corresponde al equipo donde se ha realizado el desarrollo completo del

proyecto, cuyas especificaciones técnicas son las que se incluyen al pie. De esta máquina, los cambios que se realizan a la aplicación pasan a GitLab utilizando para ello en todo momento comunicación por HTTPS.

La segunda de estas máquinas corresponde a una máquina en *cloud* con las especificaciones técnicas mínimas para que el sistema pueda funcionar con normalidad. Debe notarse que estas especificaciones son válidas con un número bajo de clientes accediendo de forma concurrente ya que, de otro modo, sería necesario el despliegue en una máquina con mayor potencia. Esta máquina está protegida con un *firewall* que también viene indicado en la vista de despliegue. Este está configurado de tal forma que sólo están abiertos los puertos imprescindibles para el buen funcionamiento del sistema. Por último, el almacenamiento mínimo disponible en la máquina debe ser 30 GB para asegurar nuevamente el correcto funcionamiento.

Para optimizar tiempo y recursos en la instalación del producto en el CPD (Centro de Procesamiento de Datos) [34] de un nuevo centro educativo, se ha desarrollado una guía de despliegue del sistema, consultable en el anexo F.



## 6. Análisis de riesgos y mantenimiento de la aplicación

El análisis de riesgo es el estudio de las causas de las posibles amenazas y probables eventos no deseados y los daños y consecuencias que éstas puedan producir. El INCIBE, Instituto Nacional de Ciberseguridad [35], recomienda, si se considera que las herramientas tecnológicas y la información manejada son de gran valor, poner en práctica un Plan Director de Seguridad.

Un PDS se define, de nuevo según el INCIBE [36], como la definición y priorización de un conjunto de proyectos en materia de seguridad de la información, dirigido a reducir los riesgos a los que está expuesta la organización hasta unos niveles aceptables a partir de un análisis de la situación inicial. Aplicado a la organización actual, el plan se expone a lo largo de la presente sección, centrándose en un plan de análisis y gestión de riesgos, así como un plan de mantenimiento.

### 6.1. Plan de análisis y gestión de riesgos

El siguiente plan de análisis y gestión de riesgos cubre, en primer lugar, la designación de los activos importantes. Además, se identifican las principales amenazas y se realiza una evaluación del riesgo asociado a cada una de ellas. Por último, se proponen estrategias de reducción de este [37].

#### Activos

A continuación, se ha realizado un inventario de activos más importantes del sistema desarrollado, que se muestran en la siguiente tabla.

ID	Nombre	Descripción	Responsable	Tipo	Ubicación	Crítico
Sv1	Servidor tipo 1	Servidor de centro educativo	Responsable de Informática del centro educativo	Servidor (Físico)	CPD centro educativo	Sí
Sv2	Servidor tipo 2	Servidor cloud	Director de la organización	Servidor (Virtual)	CPD externo	Sí
Rd1	Red	Infraestructura de red del centro educativo	Responsable de Informática del centro educativo	Red	CPD Centro educativo	Sí
Rd2	Red	Infraestructura de red del proveedor cloud	Director de la organización	Red	CPD externo	Sí
Od1	Ordenador	Ordenador de desarrollo de software	Director de la organización	Ordenador	Oficina de la organización	No

Dt1	Datos	Datos provenientes de las operaciones de la organización	Director de la organización	Datos (Virtual)	CPD externo y de centro educativo	Sí
-----	-------	--	-----------------------------	-----------------	-----------------------------------	----

Cuadro 3: Inventario de activos de la organización

## Amenazas

A continuación, se adjunta un listado categorizado de las amenazas más relevantes que se han de tener en cuenta en el proceso de análisis y gestión de riesgos de la organización. Las amenazas que se consideran son aplicables únicamente en el ámbito del centro educativo, ya que los proveedores en la nube poseen su propio plan de análisis y gestión de riesgos con las medidas que hayan decidido adoptar. Es posible encontrar un listado detallado de estas, así como de las principales vulnerabilidades y salvaguardas identificadas, en el anexo G.

- N.1/I.1 - Fuego en la sala de servidores.
- N.2 - Inundación en la sala de servidores.
- I.5 - Avería de origen físico/lógico en el servidor donde se aloja el sistema.
- E.1 - Errores de los usuarios en el uso de software.
- E.21 - Errores de la administración en el mantenimiento / actualización de software.

## Evaluación del riesgo

Tras la realización de las fases anteriores, se debe calcular el riesgo. Para cada par activo-amenaza, se estima la probabilidad de que la amenaza se materialice y el impacto sobre el negocio que esto produciría. El cálculo de riesgo se puede realizar usando tanto criterios cuantitativos como cualitativos utilizando las siguientes tablas.

Cualitativo	Cuantitativo	Descripción
Baja	1	La amenaza se materializa a lo sumo una vez cada año.
Media	2	La amenaza se materializa a lo sumo una vez cada mes.
Alta	3	La amenaza se materializa a lo sumo una vez cada semana.

Figura 10: Tabla para el cálculo de la probabilidad

En base a esto, la probabilidad de ocurrencia de las amenazas catalogadas es la siguiente:

- N.1/I.1 - Probabilidad baja.
- N.2 - Probabilidad baja.
- I.5 - Probabilidad media.
- E.1 - Probabilidad media.
- E.21 - Probabilidad baja.

Cualitativo	Cuantitativo	Descripción
Bajo	1	El daño derivado de la materialización de la amenaza no tiene consecuencias relevantes para la organización.
Medio	2	El daño derivado de la materialización de la amenaza tiene consecuencias reseñables para la organización.
Alto	3	El daño derivado de la materialización de la amenaza tiene consecuencias graves reseñables para la organización.

Figura 11: Tabla para el cálculo del impacto

En base a esto, el impacto derivado de la materialización de las amenazas catalogadas es el siguiente:

- N.1/I.1 - Impacto alto.
- N.2 - Impacto medio.
- I.5 - Impacto medio.
- E.1 - Impacto medio.
- E.21 - Impacto alto.

Al haber sido valorados tanto la probabilidad de ocurrencia del riesgo como el impacto que la materialización de este tendría sobre la organización en base a un análisis cualitativo, la valoración final se rige por la tabla que se encuentra a continuación.

		IMPACTO		
		Bajo	Medio	Alto
PROBABILIDAD	Baja	Muy bajo	Bajo	Medio
	Media	Bajo	Medio	Alto
	Alta	Medio	Alto	Muy alto

Figura 12: Tabla para el cálculo del riesgo

Así pues, el riesgo final de cada una de las amenazas es el siguiente:

- **N.1/I.1** - Riesgo medio.
- **N.2** - Riesgo bajo.
- **I.5** - Riesgo medio.
- **E.1** - Riesgo medio.
- **E.21** - Riesgo medio.

## Tratamiento del riesgo

Una vez calculado el riesgo, se deben tratar aquellos riesgos que superen un límite establecido, en este caso, se tratarán los riesgos mayores o iguales que medio.

- **N.1/I.1** - Como la probabilidad de ocurrencia de un incendio en la sala de servidores del centro educativo es muy baja, se decide asumir el riesgo ya que, de otro modo, las medidas a adoptar en base a instalación de protección ignífuga tendrían un coste demasiado alto, inasumible.
- **I.5** - En este caso, la medida a adoptar es el establecimiento de planes de vigilancia tecnológica que permitan estar al corriente cuanto antes de posible software erróneo instalado en el sistema para su corrección.
- **E.1** - Para este riesgo, la determinación adoptada es la de llevar a cabo las modificaciones necesarias en el software para capturar y tratar todos los errores posibles del usuario, de forma que nunca puedan afectar a la disponibilidad del sistema.
- **E.21** - Por último, para la mitigación del riesgo E.21 se ha decidido el establecimiento de protocolos de actuación para las labores de mantenimiento que incluyan fases de aseguramiento del buen funcionamiento del riesgo.

## 6.2. Plan de mantenimiento

Con el desarrollo de un plan de mantenimiento se reducen las averías y problemas imprevistos, viéndose toda la organización beneficiada, ya que se anticipan los problemas que puedan llegar a surgir.

### Objetivos del plan de mantenimiento

Los principales objetivos identificados que se quieren alcanzar con el presente plan son:

- Ofrecer una disponibilidad del sistema de un 99 %.
- Reducir los fallos que se encuentra el usuario.
- Mejorar la eficiencia de la mano de obra.

- Ofrecer un sistema que utilice las últimas versiones estables disponibles del software para aprovechar todo su potencial.

El mantenimiento no se realizará nunca en caso de presentar impagos el centro educativo interesado.

## Presupuesto

La distribución del presupuesto en mantenimiento que se desea alcanzar con el desarrollo del plan es de 80 % en mantenimiento preventivo y únicamente un 20 % en corrección de averías. Se planifica un presupuesto para mantenimiento de 50 € mensuales, lo cual corresponde a 600 € en el total del ejercicio.

## Elementos a controlar

Esta sección es diferente si el centro educativo interesado adquirió el software o si, por el contrario, paga por mensualidades. En el caso de adquisición del software y de su despliegue en el centro, los elementos a controlar serán los siguientes:

- Estado del servidor.
- Estado de la sala en la que se encuentra el servidor.
- Limpieza de la máquina y su entorno.
- Estado de la infraestructura de red.
- Versión del software de *backend*.
- Versión del software de GeoServer, Grafana, Nginx y PostgreSQL.
- Versión del sistema operativo.

En el caso de la utilización del software mediante el pago de mensualidades, las únicas secciones concernidas son las tres últimas, ya que el resto de elementos están ya previstos dentro del plan de actuación del proveedor de infraestructura en la nube.

Asimismo, se deberá llevar un control sobre los mantenimientos previos realizados. Esto ayudará a saber qué fue revisado, quién llevó a cabo dicha revisión, en qué fechas y cuáles fueron los problemas detectados.

## Prevención de problemas

Para la prevención de problemas en términos de pérdida de datos o de configuraciones se prevé la realización periódica de *backups*, o copias de seguridad. En concreto, dado que el servicio ofrecido no es crítico, ni tampoco el problema derivado de una pérdida de datos en el mismo, se establece la realización de una copia de seguridad diferencial anual, además de copias de seguridad incrementales con periodicidad mensual. De este modo, se asegura una recuperación óptima en caso de fallo, a la vez que no se generan problemas de almacenamiento. Por último, las copias de seguridad se almacenarán en un soporte distinto a la máquina donde se aloja el servidor que contiene el sistema.

## Gestión de problemas

Cuando se detecte que el sistema ha dejado funcionar, o funciona ofreciendo una calidad de servicio anormal el problema deberá ser abordado y subsanado a la mayor brevedad posible.

En el caso de que el sistema haya dejado de funcionar por completo, se revisarán los registros pertinentes para obtener información adicional sobre la causa del problema, obrándose en consecuencia. Se realizará una copia de los registros para que quede constancia del problema acaecido y para poder solucionar con mayor rapidez el mismo problema si vuelve a aparecer. En caso de que el sistema siga funcionando de forma anormal, se emplearán herramientas de monitorización que permitan conocer si el problema es del sistema o ajeno a él. Se tomarán las determinaciones pertinentes. Si resulta oportuno, se revisarán los registros de las herramientas que componen el sistema.

Para la reanudación del servicio, se lanzará una vez de forma manual todo el sistema y se llevarán a cabo las verificaciones pertinentes para comprobar el funcionamiento del sistema de acuerdo a sus especificaciones, lo cual incluye la calidad del servicio. Si las pruebas son superadas, el sistema volverá a operar en segundo plano con normalidad.

## Obligaciones legales

El principal marco normativo del mantenimiento en España viene dado por:

- Ley 31/1995 de Prevención de Riesgos Laborales [38].
- Real Decreto 1215/1997 de Equipos de trabajo [39].
- Real Decreto 486/1997 de Lugares de Trabajo [40].

## Periodicidad del mantenimiento

La periodicidad de realización del mantenimiento es, por defecto y como mínimo bianual, sin perjuicio de condiciones particulares que se pudiesen negociar con el centro educativo interesado.

## 7. Plan de viabilidad y negocio de la aplicación

Un plan de viabilidad analiza las posibilidades de éxito de una empresa, examinando el producto o servicio en el mercado, calculando costes empresariales, precios de productos y proyecciones de ventas. La realización de estos cálculos es necesaria para considerar la puesta en marcha del proyecto empresarial. A continuación, se incluyen los principales elementos del plan, es decir, los flujos de caja provisionales, la cuenta de resultados y el balance en el periodo de tiempo de 2021 - 2025.

Como inversiones (activos) iniciales se ha considerado únicamente la adquisición de equipos informáticos para el ejercicio de la actividad profesional, valorados en 3000 €, y amortizables en 5 años a razón de 600 € por año. En cuanto a las provisiones de tesorería iniciales se han considerado 600 € en gastos legales de constitución, además de 2000 € en gastos de lanzamiento y 3000 € en fondo de reserva para imprevistos. Esto da un reparto de 35 % de inversión inicial en activos, frente a un 65 % en provisiones de tesorería.

La financiación inicial para los gastos detallados anteriormente se distribuye en un 70 % de fondos propios, frente a un 30 % en préstamos de terceros, lo cual da una aportación de 6000 € por parte del socio del proyecto y 2600 € en préstamos de terceros, a devolver en 5 años al 5 % de interés.

SticsApp		Cash Flow Previsional					2
CASH FLOW		2021	2022	2023	2024	2025	
<b>Saldo al inicio</b>		5.600					
<i>Aportaciones s. (capital) +</i>		0	0	0	0	0	
<i>Préstamos ingresados +</i>		0	0	0	0	0	
<i>Amortizaciones +</i>		600	600	600	600	600	
<i>Proveedores (crédito) +</i>		0	0	0	0	0	
<i>Resultados</i>		-2.384	-1.230	213	2.038	4.371	
<i>Inversiones -</i>		0	0	0	0	0	
<i>Préstamos amortizados -</i>		471	494	519	545	572	
<i>Clientes (crédito) -</i>		1.060	186	234	296	380	
<i>Dividendos -</i>		0	0	149	1.426	3.059	
<b>Saldo neto de cada ejercicio</b>		2.286	-1.311	-89	370	960	
<b>Saldo acumulado al final</b>		<b>2.286</b>	<b>975</b>	<b>887</b>	<b>1.257</b>	<b>2.217</b>	

Figura 13: *Cashflows* provisionales para la organización

SticsApp		Previsión de Pérdidas y Ganancias					1
RESULTADOS		2021	2022	2023	2024	2025	
<b>Ventas</b>		<b>6.446</b>	<b>7.580</b>	<b>9.002</b>	<b>10.805</b>	<b>13.114</b>	
<i>Coste de Ventas</i>		0	0	0	0	0	
<b>Margen Bruto</b>		<b>6.446</b>	<b>7.580</b>	<b>9.002</b>	<b>10.805</b>	<b>13.114</b>	
<i>Gastos de Personal</i>		0	0	0	0	0	
<i>Gastos Corrientes</i>		8.100	8.104	8.107	8.111	8.115	
<b>EBITDA</b>		<b>-1.654</b>	<b>-524</b>	<b>894</b>	<b>2.694</b>	<b>4.999</b>	
<i>Amortizaciones</i>		600	600	600	600	600	
<b>B.A.I.I. - EBIT</b>		<b>-2.254</b>	<b>-1.124</b>	<b>294</b>	<b>2.094</b>	<b>4.399</b>	
<i>Gastos financieros (intereses)</i>		130	106	82	56	29	
<b>Resultado Bruto - BAI</b>		<b>-2.384</b>	<b>-1.230</b>	<b>213</b>	<b>2.038</b>	<b>4.371</b>	
<i>Impuesto sociedades</i>	0,0%	0	0	0	0	0	
<b>RESULTADO Neto</b>		<b>-2.384</b>	<b>-1.230</b>	<b>213</b>	<b>2.038</b>	<b>4.371</b>	
<i>% S/Ventas</i>		-36,98%	-16,23%	2,36%	18,86%	33,33%	

Figura 14: Cuenta de resultados

De las dos figuras anteriores, se extrae que las cuentas de pérdidas y ganancias para el proyecto en el horizonte 2021 - 2025 comienzan a ser positivas en el transcurso del año 2023, con 213 €.

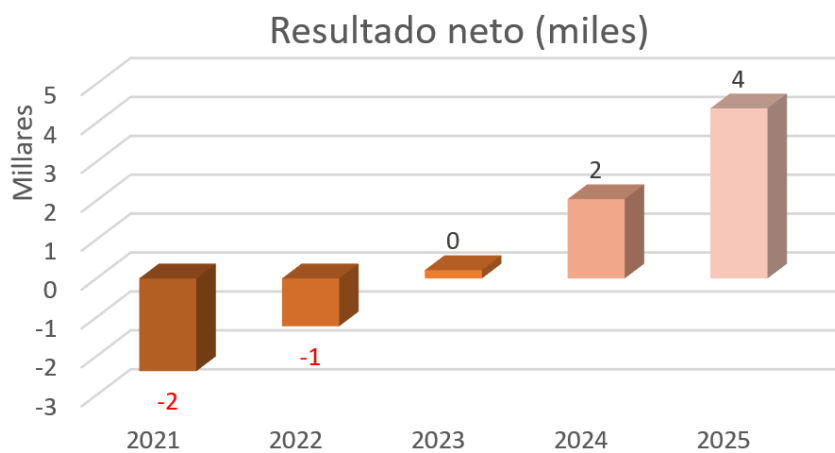


Figura 15: Gráfico de evolución de los resultados



SticsApp	<b>Balances Previsionales</b>					3
<b>BALANCE PREVISIONAL</b>	<b>2021</b>	<b>2022</b>	<b>2023</b>	<b>2024</b>	<b>2025</b>	
<b>ACTIVO NO CORRIENTE</b>	<b>2.400</b>	<b>1.800</b>	<b>1.200</b>	<b>600</b>	<b>0</b>	
<i>Inmovilizado</i>	3.000	3.000	3.000	3.000	3.000	
<i>Amortizaciones</i>	600	1.200	1.800	2.400	3.000	
<b>ACTIVO CORRIENTE</b>	<b>3.346</b>	<b>2.221</b>	<b>2.366</b>	<b>3.033</b>	<b>4.372</b>	
<i>Existencias</i>	0	0	0	0	0	
<i>Realizable</i>	1.060	1.246	1.480	1.776	2.156	
<i>Disponibile</i>	2.286	975	887	1.257	2.217	
<b>TOTAL ACTIVO</b>	<b>5.746</b>	<b>4.021</b>	<b>3.566</b>	<b>3.633</b>	<b>4.372</b>	
<b>PATRIMONIO NETO</b>	<b>3.616</b>	<b>2.386</b>	<b>2.450</b>	<b>3.061</b>	<b>4.372</b>	
<i>Capital</i>	6.000	6.000	6.000	6.000	6.000	
<i>Reservas</i>	0	0	64	675	1.986	
<i>Pérdidas</i>	-2.384	-3.614	-3.614	-3.614	-3.614	
<b>PASIVO NO CORRIENTE</b>	<b>2.129</b>	<b>1.635</b>	<b>1.117</b>	<b>572</b>	<b>0</b>	
<i>Deudas entidades de crédito</i>	2.129	1.635	1.117	572	0	
<b>TOTAL NO CORRIENTE</b>	<b>5.746</b>	<b>4.021</b>	<b>3.566</b>	<b>3.633</b>	<b>4.372</b>	
<b>PASIVO CORRIENTE</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	
<i>Proveedores</i>	0	0	0	0	0	
<i>Cashflow (-)</i>	0	0	0	0	0	
<b>TOTAL Pat. NETO y PASIVO</b>	<b>5.746</b>	<b>4.021</b>	<b>3.566</b>	<b>3.633</b>	<b>4.372</b>	
<i>Fondo de Maniobra</i>	3.346	2.221	2.366	3.033	4.372	
	2.129	1.635	1.117	572	0	

Figura 16: Balance

Los modelos de presupuestos para los centros educativos en función de la modalidad de uso del sistema escogida, esto es, despliegue en CPD del centro o utilización del software mediante SaaS desplegado en la nube, se encuentran disponibles en el anexo E. El cálculo realizado supone que el coste derivado del proceso de desarrollo de software quede cubierto con la adopción del sistema por parte de 15 centros educativos.

## 7.1. Licencia de distribución de la aplicación

Una licencia, según la Open Source Initiative, es un contrato entre el desarrollador de un software sometido a propiedad intelectual y a derechos de autor y el usuario, en el cual se definen con precisión los derechos y deberes de ambas partes. Es el desarrollador, o aquél a quien éste haya cedido los derechos de explotación, quien elige la licencia según la cual distribuye el software. Es por tanto necesario en el marco de desarrollo de este proyecto el análisis de las distintas licencias de software existentes y sus características particulares para así escoger la que más se ajuste a los intereses y voluntades del desarrollador.

En primer lugar, se descartan licencias de software de tipo privativo. Es decir, se opta por la elección de una licencia de software libre, bajo la cual la comunidad puede contribuir a la mejora y ampliación de la plataforma para así obtener un producto cada

vez mejor y que satisfaga las necesidades del usuario de mejor manera. A continuación se incluyen varias tablas que analizan las particularidades de las principales licencias.

## Permisos

Licencia	Uso com.	Distrib.	Modific.	Permiso patentes	Uso privado
AGPLv3	Sí	Sí	Sí	Sí	Sí
GPLv3	Sí	Sí	Sí	Sí	Sí
LGPLv3	Sí	Sí	Sí	Sí	Sí
Mozilla 2.0	Sí	Sí	Sí	Sí	Sí
Apache 2.0	Sí	Sí	Sí	Sí	Sí
MIT	Sí	Sí	Sí	No	Sí

Cuadro 4: Permisos aceptados por las licencias

## Condiciones

Licencia	Cód. fuente	Copia lic.	Uso por red	Misma lic.	Doc. modif.
AGPLv3	Sí	Sí	Sí	Sí	Sí
GPLv3	Sí	Sí	No	Sí	Sí
LGPLv3	Sí	Sí	No	Sí	Sí
Mozilla 2.0	Sí	Sí	No	Sí	No
Apache 2.0	No	Sí	No	No	Sí
MIT	No	Sí	No	No	No

Cuadro 5: Condiciones impuestas por las licencias

## Limitaciones

Licencia	Responsabilidad	Garantía	Derechos de marca
AGPLv3	Sí	Sí	No
GPLv3	Sí	Sí	No
LGPLv3	Sí	Sí	No
Mozilla 2.0	Sí	Sí	Sí
Apache 2.0	Sí	Sí	Sí
MIT	Sí	Sí	No

Cuadro 6: Limitaciones de cada una de las licencias

De todas ellas, la que se ha considerado que ofrece más garantías de que las obras derivadas mantendrán la misma filosofía de desarrollo y de distribución que la de este proyecto sin contener apartados que se consideren innecesarios es la GNU GPLv3. Por tanto, esta ha sido la licencia escogida.

Para información tanto del usuario de la aplicación como del desarrollador que busca recuperar el proyecto y mejorarlo o modificarlo se ha incluido un aviso de licencia tanto en el sitio web como en el repositorio GitLab público en que se encuentra almacenado el código fuente.

## 7.2. Plan de marketing

Un plan de marketing es imprescindible para lograr llegar mejor al público objetivo del proyecto. Su objetivo es definir la misión de la empresa, es decir, la solución que provee, así como la propuesta de valor de la organización. A continuación, es necesario definir con claridad el mercado objetivo, esto es, el cliente ideal. Asimismo, se deben establecer las fortalezas y debilidades de la organización, para enfocar mejor el plan de marketing a realizar. Por último, se realizará una descripción de las principales técnicas de marketing que se van a realizar.

Como ya se ha comentado anteriormente, la principal misión del proyecto Stics es la de proveer a los centros educativos con una herramienta novedosa para el aprendizaje de materias como matemáticas y tecnología industrial de forma dinámica.

### Propuesta de valor

La propuesta de valor del proyecto es una aplicación web para trabajar en el aula y en casa los conocimientos adquiridos por medio de la enseñanza tradicional enfocado a una temática de actualidad: el medioambiente. Se proponen herramientas como la creación de cuestionarios o la visualización de cuadros de mandos con gráficos sobre la evolución de la concentración de contaminantes en las estaciones de calidad ambiental instaladas en la ciudad.

### Mercado objetivo

El cliente ideal del proyecto sería un centro educativo en el que la tecnología esté integrada dentro del proceso de aprendizaje del alumnado. Se trata de un centro de educación primaria y/o secundaria preferiblemente, ya que son dichas enseñanzas a las que más se puede ajustar los contenidos ofrecidos por la plataforma.

### Soluciones existentes

El estudio de soluciones ya existentes en el mercado puede dar una idea de por dónde se debería enfocar el marketing de la aplicación, así como cuáles son los elementos diferenciales de esta frente a las demás. En este caso, existen aplicaciones que permiten a los estudiantes aprender ciertos aspectos de temática medioambiental. No obstante, no hay otra aplicación existente que permita estudiar sensores de contaminación ambiental de la forma en la que esta nueva herramienta lo plantea.

Un ejemplo de estas aplicaciones existentes es **Zooniverse**, una herramienta online que contiene multitud de proyectos a los que se puede contribuir, como, por ejemplo, uno que ayuda a explicar la estructura de una célula y ofrece práctica a los estudiantes en la comparativa de células para encontrar aquellas que son cancerosas, contribuyendo así a investigaciones reales que se están llevando a cabo.

## Análisis DAFO

El análisis DAFO (Debilidades, Amenazas, Fortalezas, Oportunidades) es una herramienta de estudio de la situación de una empresa, institución, proyecto o persona, analizando sus características internas (Debilidades y Fortalezas) y su situación externa (Amenazas y Oportunidades) en una matriz cuadrada. El objetivo del análisis DAFO es determinar las ventajas competitivas de la empresa bajo análisis y la estrategia genérica que más le convenga en función de sus características propias y de las del mercado en que se mueve. A continuación se muestra el análisis DAFO que se ha llevado a cabo para el presente proyecto. Para su realización se ha utilizado la herramienta que ofrece la DGIPYME (Dirección General de Industria y de la Pequeña y Mediana Empresa) [41].

### Matriz de Factores

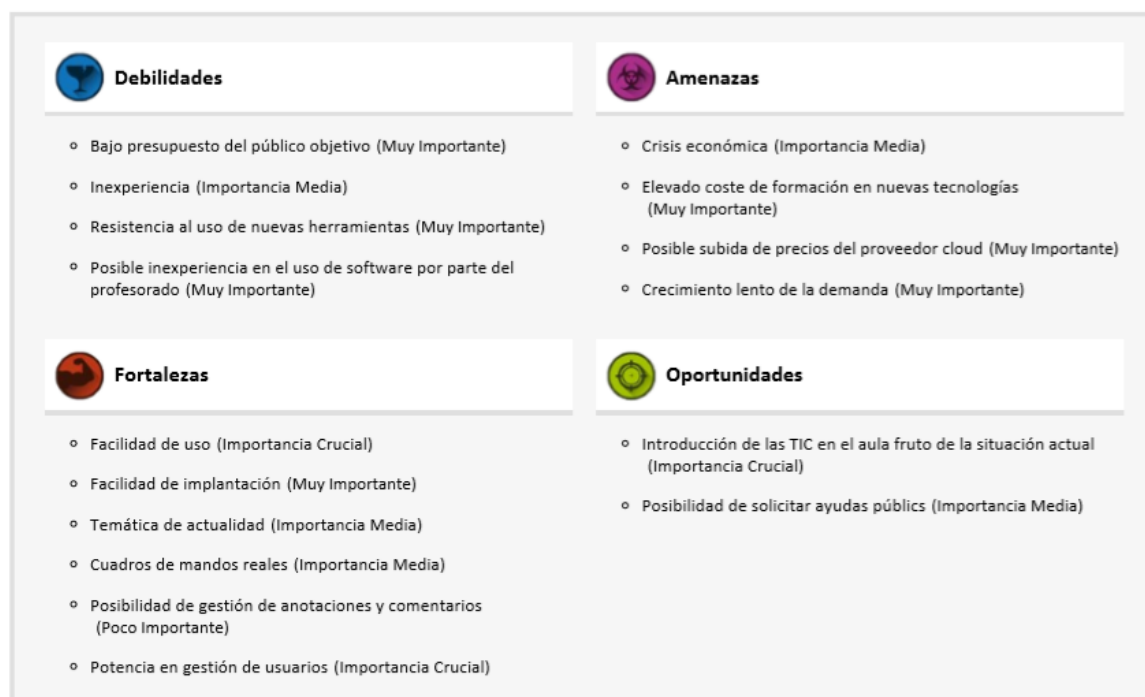


Figura 17: Matriz de factores del análisis DAFO

Tras la realización de la matriz de factores presentada previamente, se han desarrollado una serie de estrategias que confrontan cada una de ellas distintos aspectos de la matriz con el fin de obtener una idea de cuáles deben ser los siguientes pasos a dar por parte de la organización para la consecución de sus objetivos. Los resultados de este estudio se presentan a continuación.

## Resultados

DAFO: Nuevo negocio: Stics. Gestión y visualización de cuadros de mandos de indicadores medioambientales georreferenciados

Estas son las estrategias ordenadas por prioridad y la matriz de factores. El orden está relacionado con los factores DAFO asociados a cada estrategia. Cambiar los factores o el grado de importancia que se conceda a cada uno, puede modificar el orden.

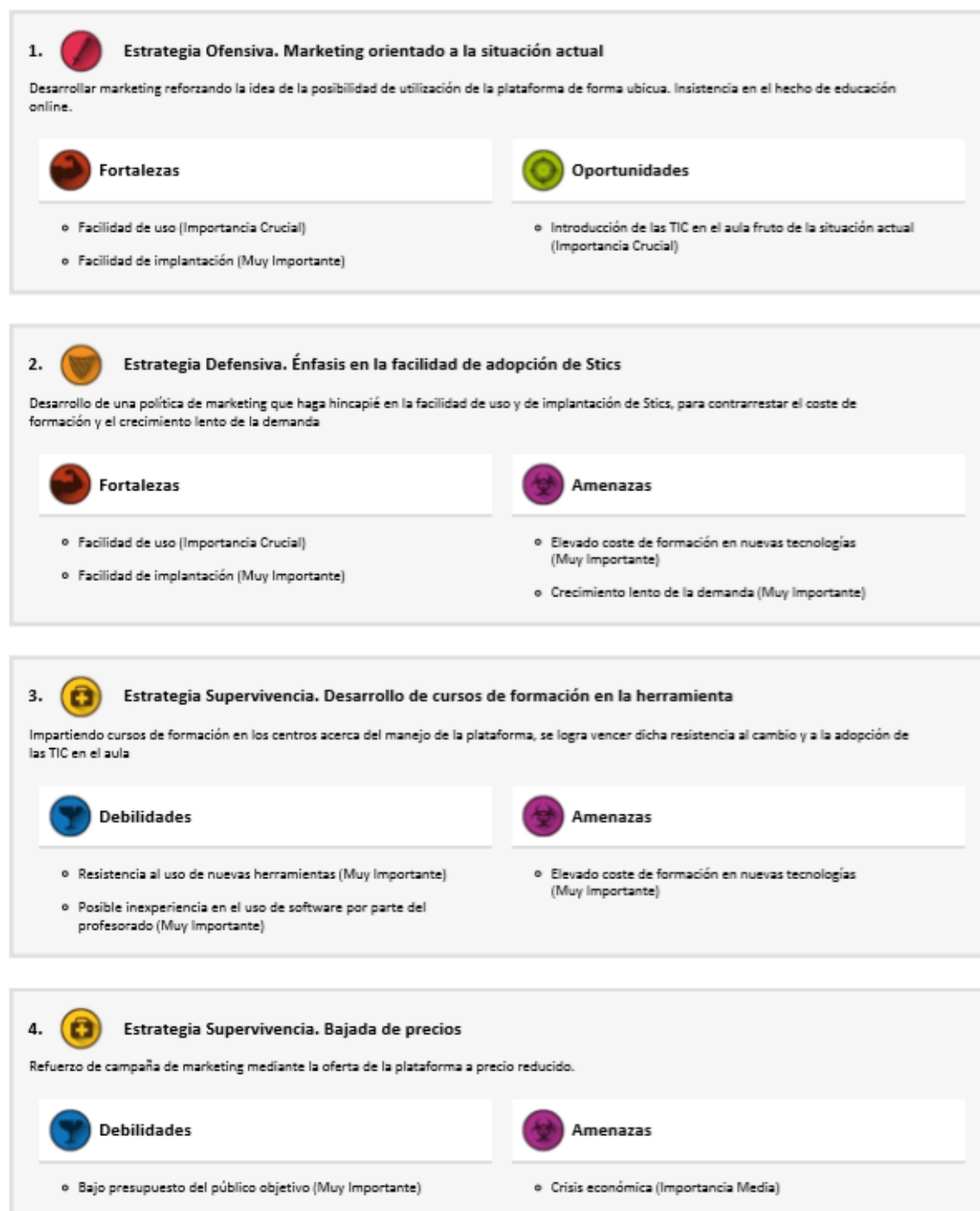


Figura 18: Estrategias a adoptar (1)



Figura 19: Estrategias a adoptar (2)

### Presupuesto para márketing

El presupuesto mensual dedicado a márketing es de 20 €, lo cual hace un total de 240 € anuales, a distribuir en campañas de márketing a lo largo del año.

### Técnicas a aplicar

#### Email marketing

La primera posibilidad es la utilización de técnicas de email marketing para intentar retener a potenciales clientes que se encuentren indecisos y que accedan al sitio web en busca de información en la página de ventas desarrollada a tal efecto. No obstante, hay que tener en cuenta que no se debe enviar correo molesto ya que va a ser considerado como *spam* y va a causar enfado y, por tanto, rechazo hacia la marca.

#### Redes sociales

La publicidad en redes sociales es fundamental en la actualidad. En estos momentos, todo ocurre en redes sociales por lo que es necesario estar presentes en ellas. Estas son un canal más de comunicación con los clientes que ayuda al conocimiento de la marca. Es por esto por lo que para la realización de campañas de márketing del producto debería considerarse la adquisición de publicidad en Facebook. Estas campañas deberían ser monitorizadas para observar el impacto que tienen y así enfocar mejor la estrategia.

#### Otras vías

Otras vías de publicitación del producto son canales más tradicionales pero que pueden resultar igualmente eficaces como el clásico “boca a boca”. El profesorado son personas que hasta la obtención de destino definitivo cuentan con un alto grado de rotación en distintos centros educativos, lo cual los pone en contacto con multitud de compañeros. Esto puede facilitar la difusión del conocimiento sobre la aplicación en cada vez más instituciones.

## 8. Conclusiones

El proyecto desarrollado contribuye al proceso de digitalización de la educación que se viene llevando a cabo desde múltiples ámbitos formativos. La introducción de herramientas informáticas en la educación diaria introduce elementos motivadores al alumnado y transforma el trabajo en el aula, aportando mayor dinamismo al proceso de aprendizaje y logrando una mayor implicación del alumnado en el aprendizaje al resultarle más motivador que una sesión tradicional.

Además, el hecho de abordar una temática relacionada con los Objetivos de Desarrollo Sostenible (ODS) [42], a la orden del día, hace que el alumnado, dentro del proceso de aprendizaje de los objetivos fundamentales de la plataforma en asignaturas como matemáticas o tecnología, pueda tomar también conciencia de la problemática actual entorno a este, ya que se encuentra trabajando en todo caso con datos reales provenientes de su ciudad.

Pero el proyecto no solo resulta formativo y enriquecedor para sus usuarios finales, sino también para el responsable de llevarlo a cabo. Con proyectos como este, en los que la implicación del usuario final está presente desde el comienzo, se aprende sobre la necesidad de centrarse en él durante el desarrollo, corrigiendo aquellos aspectos que, por muy evidentes que puedan parecer, no lo son para el usuario final.

En cuanto a los aspectos técnicos, de nuevo se destaca el aprendizaje que ha supuesto el desarrollo de la herramienta Stics. Se ha logrado ampliar los conocimientos en Sistemas de Información Geográficos que no son abordados en profundidad durante el Grado, pero que, sin embargo, están presentes en multitud de aplicaciones en diversos sitios web. Ocurre lo mismo con el software de visualización de datos métricos, de nuevo a la orden del día en aplicaciones de *Business Intelligence*. Asimismo, al ser un proyecto de cierta entidad, se ha debido gestionar un sistema grande, en el que interaccionan entre sí distinto software, bases de datos etc. con la complejidad que ello conlleva. No obstante, la existencia de conectores y APIs de calidad ya desarrolladas facilitan la tarea en gran medida.

En cuanto a la duración y coste del proyecto, no se ha observado desfase alguno entre el tiempo a dedicar previsto y el realmente invertido, por lo que el trabajo realizado ha abarcado unas 390 horas. De ellas, la mayor parte (un 63%), como se puede observar en el gráfico siguiente, se han dedicado a la fase de “Implementación y desarrollo”. Tras esta, se encuentra la fase de “Documentación”, con un coste global de 60 horas. Las tareas que menos horas de trabajo han requerido han sido las fases de “Estudio del estado del arte” y “Análisis de requisitos”, con un 7% (25h 45’) y un 2% (7h 5’) respectivamente del total. Si se desea obtener un mayor nivel de detalle en los esfuerzos realizados durante el desarrollo del proyecto se puede consultar el anexo E.

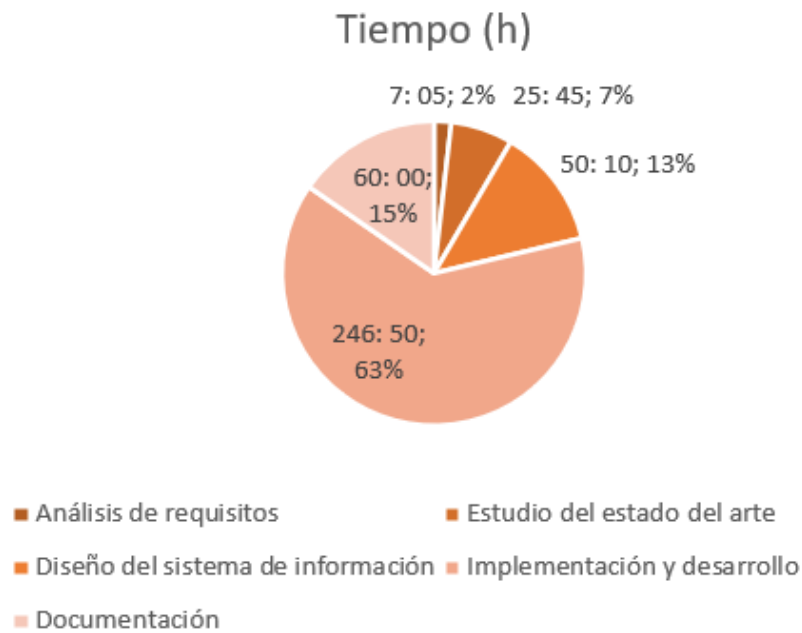


Figura 20: Gráfico de distribución del tiempo en las tareas realizadas

## 8.1. Trabajo futuro

El hecho de que el desarrollo del proyecto en el marco del *TFG* finalice aquí, no quiere decir que el proyecto deba o vaya a abandonarse de la misma manera. El proyecto posee aún muchas líneas de trabajo en las que invertir tiempo para hacer del producto un sistema mejor. Por ejemplo, en línea con la estrategia de publicación de datos abiertos adoptada por entidades públicas como UNIZAR con el “Repositorio Digital de la Universidad de Zaragoza”, el Ayuntamiento de Zaragoza con “Datos Abiertos Ayuntamiento de Zaragoza”, la Comunidad Autónoma de Aragón con “Aragón Open Data” o el Gobierno de España [43], así como entidades privadas como Google con “Google Dataset Search” [44], se podría sacar partido de estos datos desarrollando para ello un *wrapper*, o conector. De este modo, la información ofrecida a los usuarios de la plataforma sería más, y más rica.

## 8.2. Colaboraciones

A pesar de la situación derivada de la crisis sanitaria del COVID-19 que ha perturbado la vida de las personas a escala mundial, gracias a las distintas herramientas de comunicación existentes, ha sido posible realizar algunas colaboraciones con grupos de trabajo que han contribuido sin lugar a dudas a hacer del proyecto una aplicación de calidad.

### Proyecto Trafair

El proyecto Trafair reúne a 10 socios de dos países europeos, España e Italia, con



el objetivo de desarrollar servicios innovadores y sostenibles combinando calidad del aire, condiciones meteorológicas y datos de flujo de tráfico para producir nueva información para el beneficio de ciudadanos y cargos gubernamentales implicados en el proceso de toma de determinaciones.

En primer lugar, por tanto, agradecer a los integrantes del proyecto Trafair que me hayan permitido mi integración en cierto modo en él, contribuyendo a este con una nueva rama, el ámbito educativo. En el trabajo, la base de datos medioambientales era la utilizada en dicho proyecto. Esto ha resultado muy útil ya que ha supuesto una descarga de esfuerzo que ha permitido que me haya podido centrar en otros aspectos del sistema.

En especial, agradecer a mi directora del *TFG*, Raquel Trillo Lado, su dedicación y esfuerzo, aún más si cabe en el marco de la crisis sanitaria actual, y su disposición a comentar los aspectos que fuesen necesarios, así como por permitirme plantearle cuantas dudas y preguntas tuviese. Indudablemente, el proyecto no hubiese resultado tan satisfactorio sin su ayuda y colaboración, siempre constructiva, ya que ha permitido crear un ambiente de trabajo enriquecedor que no hacía sino aportar y hacer del trabajo un producto mejor, además de una experiencia sin lugar a dudas positiva.

Por último, agradecer también su colaboración y contribución al trabajo a Ilenia Gaspari, estudiante de Erasmus en la Universidad de Zaragoza, que, de nuevo, se ha mostrado dispuesta a ayudarme con tantos aspectos como fueran necesarios. Su *feedback* ha resultado especialmente relevante al tratarse de un usuario final, que no ha estado implicado en el proceso de desarrollo de la herramienta. Sus planteamientos y dudas han llevado en varias ocasiones a realizar modificaciones de aspectos técnicos del proyecto que, de otro modo, quizá no se habrían detectado.

## A. Glosario

A continuación se incluye un listado de términos utilizados en el presente documento junto con una explicación sobre estos.

**TFG (Trabajo Fin de Grado):** Según la Universidad de Málaga, “El Trabajo Fin de Grado (TFG) es un trabajo autónomo que cada estudiante realizará bajo la orientación de un tutor/a, quien actuará como dinamizador y facilitador del proceso de aprendizaje. Debe ser un trabajo original no presentado con anterioridad por el estudiante para superar otras materias en esta titulación, o en otras titulaciones previamente cursadas en esta u otras universidades. Tanto el contenido como la modalidad la deciden entre el estudiante y su tutor/a.” [45]

**Ciclos PDCA (Plan-Do-Check-Act):** Esta metodología describe los cuatro pasos esenciales que se deben llevar a cabo de forma sistemática para lograr la mejora continua, entendiendo como tal al mejoramiento continuado de la calidad (disminución de fallos, aumento de la eficacia y eficiencia, solución de problemas, previsión y eliminación de riesgos potenciales...). El ciclo de Deming, o ciclo PDCA lo componen 4 etapas cíclicas, de forma que una vez acabada la etapa final se debe volver a la primera y repetir el ciclo de nuevo, de forma que las actividades son reevaluadas periódicamente para incorporar nuevas mejoras. La aplicación de esta metodología está enfocada principalmente para ser usada en empresas y organizaciones.

**API (Application Programming Interface):** Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API le otorgan flexibilidad; simplifican el diseño, la administración y el uso de las aplicaciones, y proporcionan oportunidades de innovación.

**SIGAD:** SIGAD es la plataforma de gestión educativa desarrollada por el Gobierno de Aragón. Permite a padres, madres y tutores legales del alumnado consultar vía web las faltas de asistencia, incidencias, calificaciones y otras comunicaciones relativas a sus hijos e hijas.

**Asociación Española de Control de Calidad:** Entidad privada sin ánimo de lucro cuyo propósito es impulsar la calidad como motor de la competitividad y la sostenibilidad de los profesionales, las empresas y el país. En la actualidad hay más 1.000 empresas y de 3.500 profesionales que colaboran con la misión de la asociación.

**CPD (Centro de procesamiento de datos):** Se denomina centro de procesamiento de datos al espacio donde se concentran los recursos necesarios para el procesamiento de la información de una organización.

**Open Source Initiative:** La Open Source Initiative es una organización dedicada a la promoción del código abierto. Fue fundada en febrero de 1998 por Bruce Perens y Eric S. Raymond.

## B. Estudio del estado del arte

Para tomar decisiones adecuadas en el software a utilizar para el desarrollo de un nuevo sistema, como es el caso, es necesario llevar a cabo una amplia investigación que permita conocer con claridad cuáles son las particularidades de las diferentes soluciones posibles. Solo así se contará con criterio suficiente para elegir aquella que se ajuste mejor a las necesidades planteadas. A continuación, se muestra el desarrollo de dicho trabajo de investigación realizado de forma somera.

### B.1. Frameworks de desarrollo web

A continuación, se encuentra el análisis extendido realizado de los frameworks de desarrollo web que se consideraron para implementar la aplicación.

#### Express.js

Express es un *framework* de aplicaciones web proporcionado gratuitamente y como software libre bajo licencia MIT. Ha sido designado como el estándar de facto para Node.js. Se describe como un framework minimalista y flexible con multitud de funcionalidades disponibles como *plugins*.

Fue lanzado en mayo de 2010 y posteriormente adquirido por, en primer lugar, StrongLoop, siendo esta adquirida finalmente por IBM.

#### Ventajas

- Buena API de enrutado. Permite definir rutas simples a las que añadir posteriormente parámetros.
- Relativamente maduro. Al ser un framework relativamente antiguo para Node.js, es más estable que los competidores y tiene una gran comunidad de soporte.
- *Plugins*. Posibilidad de instalar *plugins* fácilmente mediante NPM.
- Facilidad de comienzo. Al utilizar NPM, la instalación de dependencias es muy simple.
- Ligera.
- Gran comunidad de soporte. Facilidad de localización de guías y tutoriales escritos por desarrolladores con experiencia.
- Alta escalabilidad.

#### Desventajas

- No hay una única forma recomendada de desarrollar algo. Esto conlleva una dificultad añadida para principiantes y resultar en proyectos complicados de mantener.

#### Casos de éxito

Algunas entidades empresariales que han adoptado Express.js para desarrollos web en producción son: Accenture, IBM y FOX Sports.

## Django

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como MVC (Modelo–Vista–Controlador). Fue lanzado en julio de 2005 bajo licencia BSD (*Berkeley Software Distribution*).

Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, del inglés Don't Repeat Yourself). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.

Al igual que Ruby on Rails, cuyo análisis se encuentra a continuación, Django se usó en producción para un propósito específico durante un tiempo antes de que se liberara al público.

### Ventajas

- Soporte a ORM, siglas en inglés de mapeo objeto-relacional. Las entidades de la base de datos se definen como clases de Python. Cada atributo de la clase se convierte en un campo de la base de datos. Django proporciona una API generada automáticamente para acceder a la base de datos.
- Buena documentación y comunidad de desarrolladores. Django es una herramienta popular y con cierto recorrido, lo cual hace que existan desarrolladores con mucha experiencia en su uso que compartan su conocimiento en Internet.
- Personalizable. Django permite la modificación del SGBD de forma sencilla.
- Interfaz de administración de la base de datos. Django proporciona automáticamente un panel de administración para interactuar con la base de datos.
- *Plugins*. Al tratarse de un software lanzado en 2005, cuenta con un gran número de *plugins*.

### Desventajas

- El enrutado requiere de conocimiento de expresiones regulares.
- Los errores en las templates HTML (HyperText Markup Language) no son fácilmente detectables ya que Django no informa de ellos, sino que fallan de forma silenciosa, mostrándose como strings vacíos, por ejemplo.

### Casos de éxito

Algunas entidades empresariales que han adoptado Django para desarrollos web en producción son: Instagram, Spotify, YouTube y The Washington Post.

## Ruby On Rails

Ruby on Rails, también conocido como RoR o Rails, es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma del patrón Modelo Vista Controlador (MVC).

El lenguaje de programación Ruby permite la metaprogramación, de la cual Rails hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible. Rails se distribuye a través de RubyGems, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones Ruby bajo licencia MIT.

Fue lanzado por primera vez en julio de 2004.

### **Ventajas**

- Gran comunidad con tutoriales y guías de desarrollo.
- Amplia disponibilidad de *plugins* bien documentados.
- Capacidad de metaprogramación.
- Soporte en los principales proveedores de infraestructura *cloud*.

### **Desventajas**

- Difícil de detectar errores. La capacidad de metaprogramación puede hacer que en algunos escenarios la detección de errores sea tediosa.
- Curva de aprendizaje. Puede resultar fácil de comenzar a trabajar con él pero la complejidad aumenta conforme lo hace la complejidad del software desarrollado.
- Lenguaje no muy conocido fuera del desarrollo web.
- Uno de los *frameworks* más lentos. Mala escalabilidad.

### **Casos de éxito**

Algunas entidades empresariales que han adoptado Ruby On Rails para desarrollos web en producción son: Airbnb, Bloomberg y GitHub.

## **Spring**

Spring es un *framework* para el desarrollo de aplicaciones de código abierto para la plataforma Java.

El *framework* fue lanzado inicialmente bajo la licencia Apache 2.0 en junio de 2003.

Si bien las características fundamentales de Spring Framework pueden ser usadas en cualquier aplicación desarrollada en Java, existen variadas extensiones para la construcción de aplicaciones web sobre la plataforma Java EE.

### **Ventajas**

- Gran comunidad de desarrolladores. Al ser el más antiguo y más utilizado de los frameworks JVM, cuenta con una gran comunidad de desarrolladores experimentados.
- Gran capacidad de escalar.

- Buena documentación.
- Gran ecosistema de herramientas.

### Desventajas

- Actualización y mantenimiento del código complicada para principiantes.
- Complejo de utilizar.
- Dificultad para producir un prototipo rápido y fácil.

### Casos de éxito

Algunas entidades empresariales que han adoptado Spring framework para desarrollos web en producción son: Accenture (ya vista en Express.js), Zalando y MIT.

### Solución escogida

Producto	Facilidad de uso	Documentación	<i>Plugins</i>	Escalabilidad
Express.js	Media	Sí	Sí	Alta
Django	Sí	Sí	Sí	Alta
Ruby On Rails	Baja	Sí	Sí	Baja
Spring	Baja	Sí	Sí	Alta

Cuadro 7: Comparativa de frameworks de desarrollo web

Tras analizar cada una de las posibilidades de frameworks de desarrollo web, resulta simple extraer algunas conclusiones. Por ejemplo, el patrón de desarrollo MVC está extendido en ellos, lo cual facilita su utilización aunque no se haya trabajado con anterioridad con la herramienta. Todos ellos proporcionan asimismo una gran comunidad de usuarios que ofrecen su conocimiento en Internet, disminuyendo la curva de aprendizaje, así como una buena documentación ofrecida por el propio desarrollador del framework. Esto hace que todos constituyan, a un nivel básico, una buena elección para el desarrollo del proyecto, por lo que serán los detalles los que marquen la diferencia entre uno y otro.

Por un lado, utilizar Express.js como framework para el backend conlleva la utilización de otra herramienta, Node.js para el frontend del sistema. Esto hace que haya que aprender a utilizar dos herramientas desde prácticamente cero, con lo que ello conlleva en términos de tiempo. Además, el hecho de que no haya una única forma recomendada de desarrollar los elementos puede hacer más complejo el proceso de aprendizaje, por lo que el uso de Express.js queda descartado.

En segundo lugar, Django es una herramienta con la que ya se posee experiencia de desarrollo y se conocen sus defectos o inconvenientes. Además, la fácil interacción con la base de datos, tanto para su definición como para su posterior explotación son un gran punto a favor. Asimismo, el SGBD puede ser sustituido en cualquier momento, lo cual es positivo para la escalabilidad caso de ser necesaria.

Ruby On Rails, como el resto, cuenta con una amplia comunidad de desarrolladores, así como con *plugins* variados para cumplir con determinados cometidos en un momento dado. No obstante, otro de sus puntos positivos, la metaprogramación, puede convertirse en una desventaja en la detección de errores. Además, el hecho de ser uno de los *frameworks* más lentos hace que la escalabilidad, necesaria por ejemplo si se desea proporcionar la herramienta en forma de solución SaaS, sea compleja.

Por último, Spring, si bien es cierto que cuenta con, como el resto de *frameworks* analizados, una gran comunidad y ecosistema de herramientas, además de gran capacidad de escalar, el principal defecto de Ruby On Rails, resulta complejo de utilizar. Además, la actualización del código y su mantenimiento pueden ser complejas para alguien que no conoce Spring en profundidad, lo cual es el caso.

En conclusión, tras analizar las ventajas y desventajas de cada una de las herramientas, se ha decidido progresar en el desarrollo del proyecto utilizando Django. Este cuenta con prácticamente todas las ventajas del resto y es un *framework* que ya ha sido utilizado con anterioridad, lo cual permite concentrar los esfuerzos de desarrollo en otros elementos del proyecto, estos sí, desconocidos, como es el caso del software GIS.

## B.2. Sistemas gestores de bases de datos

A continuación, se encuentra el análisis extendido realizado de los Sistemas Gestores de Bases de Datos que se consideraron para implementar la aplicación.

### PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional y de código abierto, publicado bajo la licencia PostgreSQL, similar a la BSD o la MIT. El desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales.

El proyecto PostgreSQL hace lanzamientos principales anualmente y lanzamientos menores de reparación de bugs, todos disponibles bajo la licencia PostgreSQL, y basados en contribuciones de proveedores comerciales, empresas y programadores de código abierto mayormente.

#### Características

- Alta concurrencia.
- Variedad de tipos nativos.
- Herencia de tablas.
- Integridad transaccional.
- Soporte para transacciones distribuidas.

- Gran cantidad de *plugins* disponibles.
- Proporciona herramientas adicionales para escalabilidad.

### **Coste del software**

PostgreSQL no tiene coste asociado al tratarse de un proyecto OpenSource.

### **Casos de éxito usando PostgreSQL**

Algunas de las empresas que utilizan actualmente PostgreSQL en su stack de tecnologías son Uber, Netflix, Instagram o Spotify, entre otras.

## **MariaDB**

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL. Es desarrollado por Michael Widenius, fundador de MySQL, la fundación MariaDB y la comunidad de desarrolladores de software libre.

Este SGBD surge a raíz de la compra de Sun Microsystems por parte de Oracle. MariaDB es una bifurcación directa de MySQL que asegura la existencia de una versión de este producto con licencia GPL. Widenius decidió crear esta variante por la convicción de que el único interés de Oracle en MySQL era reducir la competencia que MySQL suponía para el mayor proveedor de bases de datos relacionales del mundo, que es Oracle. Hay bastantes paquetes privativos y libres de terceros diseñados para MySQL que también están disponibles para integrarse con MariaDB.

### **Características**

- OpenSource.
- MariaDB incluye una amplia selección de motores de almacenamiento.
- Soporte para diversos lenguajes de programación.
- Adición de operaciones y comandos no disponibles en MySQL y supresión de características que afectan negativamente al funcionamiento del sistema.
- Buena escalabilidad.

### **Coste del software**

La versión "Community" del SGBD MariaDB no tiene coste alguno asociado. Si se optase por otro tipo de versiones sí que es necesario el desembolso de dinero.

### **Casos de éxito usando MariaDB**

Algunas de las empresas que utilizan actualmente MariaDB en su stack de tecnologías son 1&1, AWS o BlaBlaCar, entre otras.

## **MySQL**

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation. MySQL



es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de doble licenciamiento anteriormente mencionado. La base de datos se distribuye en varias versiones, una Community, distribuida bajo la Licencia pública general de GNU, versión 2, y varias versiones Enterprise, para aquellas empresas que quieran incorporarlo en productos privativos.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación.

### **Características**

- Fácil de usar.
- Seguro.
- Escalable.
- Gran velocidad.
- Eficiente en el uso de memoria.
- Posibilidad de particionado.

### **Coste del software**

Como el proyecto a desarrollar se pretende distribuir bajo licencia OpenSource para permitir la participación de otros desarrolladores en la ampliación y mejora del software, es posible utilizar la versión gratuita del software MySQL. Si en algún momento se viese viable pasar a un proyecto privativo sería necesario desembolsar el coste asociado a la licencia MySQL para estos proyectos.

### **Casos de éxito usando MySQL**

MySQL es usado por muchos sitios web grandes y populares, como Wikipedia, Google (aunque no para búsquedas), Facebook, Twitter, Flickr y YouTube.

## **Oracle**

Oracle Database es un sistema de gestión de base de datos desarrollado por Oracle Corporation. Su dominio en el mercado de servidores empresariales había sido casi total hasta que recientemente tiene la competencia de Microsoft SQL Server y de la oferta de otros RDBMS (*Relational Database Management System*) con licencia libre como PostgreSQL o MySQL.

Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

### **Características**

- Consistencia en los datos en entornos de ejecución concurrente.
- Manejabilidad.
- Posibilidades de copias de seguridad y procesos de recuperación.

- Alta disponibilidad.
- Soporte a BD muy grandes.
- Seguridad.
- Líder en el Cuadrante Mágico de Gartner para la gestión de bases de datos.

### Coste del software

El coste del software por usuario en la base de datos es, según las tablas de precios proporcionadas por Oracle, de 350 USD para la edición estándar. Si se escogiese la edición empresarial el coste aumentaría hasta los 950 USD. En el caso de optar por licencias tipo *Processor License*, el coste asciende a 17500 USD o 47500 USD según el tipo de edición escogida.

### Casos de éxito usando Oracle

Oracle es usado por muchos sitios web grandes y populares, como Epsilon, Henry Ford Health, Wargaming o OUTFRONT Media, entre otros.

### Solución escogida

Producto	Coste	Escalabilidad	Backup	<i>Plugins</i>	Trabajo previo
PostgreSQL	Gratuito	Sí	Sí	Sí	Sí
MariaDB	Gratuito	Sí	Sí	Sí	No
MySQL	Gratuito	Sí <sup>2</sup>	Sí	Sí	No
Oracle	Alto	Sí	Sí	Sí	Sí

Cuadro 8: Comparativa de SGBD

El análisis anterior permite la extracción de conclusiones que llevan a descartar algunos SGBD directamente. Por ejemplo, el hecho de que el equipo detrás del proyecto Stics carezca de otras fuentes de ingresos distintas de las posibles derivadas de la venta y licencia del software desarrollado hace inviable la adquisición y puesta en marcha de un software como el de Oracle Corporation, con precios de varios cientos de euros por cada licencia para cada usuario. Inclusive en el escenario que se presenta teniendo en cuenta las mejores proyecciones, la adquisición de Oracle es inviable en el corto-medio plazo ya que se trata de una inversión muy importante que no aporta elementos diferenciales definitivos con el resto de software analizado que hagan decantarse por él.

Con MySQL, Oracle ofrece un probado sistema de gestión de bases de datos que, desde hace más de 20 años, goza de una gran popularidad en todo el mundo. Bajo el patrocinio de Oracle, MySQL puede considerarse como proyecto de software independiente que cuenta con mantenimiento, desarrollo y mejoras constantes. No obstante, si se debe hacer uso de funcionalidades adicionales, es posible encontrarse con las limitaciones de la versión Community gratuita.

<sup>2</sup>Su uso puede provocar problemas de integridad en entornos de alta concurrencia en la modificación.

Detrás de MariaDB se encuentra el equipo de desarrolladores de MySQL, que cuenta con el respaldo de la comunidad de código abierto. Esto hace que entre ambos SGBD se decida descartar MySQL. Además, una posible evolución de MySQL hacia el software privativo dejaría al proyecto en la encrucijada de tener que modificar el SGBD utilizado y migrar los datos a otro o de adquirir una licencia, lo cual puede ser un gasto difícilmente asumible en función de la evolución de las ventas del producto. Por otro lado, el hecho de pertenecer MySQL a Oracle Corporation, propietario de Oracle Database, resulta en la inquietud de que, en un momento dado, la estrategia empresarial de la compañía pase por abandonar el desarrollo de MySQL y no ofrecer más soporte al producto.

Por último, entre MariaDB y PostgreSQL, la extensa historia de este último que hace que haya sido testado en profundidad por la comunidad, permaneciendo gratuito y OpenSource siempre, además de la fiabilidad de este, reconocida por la industria, la integridad en los datos, su corrección, la estabilidad y su escalabilidad hace que se haya decidido proseguir con el desarrollo del proyecto utilizando PostgreSQL. Además, la gran cantidad de *plugins* fácilmente instalables disponibles que pueden ser interesantes para desarrollar funcionalidad adicional hace que la elección de PostgreSQL resulte lo más interesante. A ello se suma el mejor rendimiento de PostgreSQL frente a MariaDB. No obstante, si posteriormente, la progresión del proyecto, así como de los SGBD hiciese interesante la utilización de MariaDB, se podría adoptar rápidamente gracias a la facilidad que ofrece Django en este sentido.

### **B.3. Sistemas de visualización y formato de datos métricos**

A continuación, se encuentra el análisis pormenorizado realizado de los sistemas de visualización y formato de datos métricos que se consideraron para implementar la aplicación.

#### **Grafana**

Grafana es un software libre basado en licencia de Apache 2.0, que permite la visualización y el formato de datos métricos. Permite crear cuadros de mando y gráficos a partir de múltiples fuentes, incluidas bases de datos de series de tiempo como Graphite, InfluxDB [46] y OpenTSDB [47]. Originalmente comenzó como un componente de Kibana [48], software que también será objeto de análisis a continuación y que luego le fue realizada una bifurcación.

Grafana es multiplataforma sin ninguna dependencia y también se puede implementar con Docker [49]. Está escrito en lenguaje Go [50] y tiene un HTTP API completo.

Además de administrar cuadros de mando clásicos (adiciones, eliminaciones, favoritos), Grafana ofrece compartir un cuadro de mando actual mediante la creación de un enlace o una instantánea estática del mismo. La herramienta ofrece asimismo la posibilidad de configurar alertas.

## Características

- Variedad de tipos de gráficos para distintos propósitos.
- Posibilidad de definición de alertas.
- Posibilidad de mezclar diferentes fuentes de datos en un mismo dashboard.
- OpenSource.
- Multitud de *plugins*.
- Posibilidad de embeber paneles, no dashboards completos.

## Coste del software

Al ser software libre distribuido bajo licencia Apache, Grafana no cuenta con costes derivados de su descarga e implantación, ni tampoco de su utilización.

## Casos de éxito usando Grafana

Grafana es utilizado por muchas organizaciones populares para llevar a cabo labores de monitorización, como PayPal, ebay, Intel o TED, entre otros.

## Graphite

Graphite es un software libre gratuito que monitoriza y muestra en gráficos datos numéricos de series temporales. Este producto fue desarrollado por Orbitz Worldwide Inc. y lanzado en 2008.

Graphite recoge, almacena y muestra datos de series temporales en tiempo real. Cuenta con tres componentes, Carbon, un demonio para escuchar datos, Whisper, una librería de base de datos para su almacenamiento y la webapp, desarrollada en Django, que muestra los gráficos bajo demanda.

## Características

- Multitud de gráficos.
- Almacenamiento de datos en su base de datos.
- API Sockets.
- Imposibilidad de definir alarmas.

## Coste del software

Nuevamente, al igual que Grafana, Graphite es licenciado bajo Apache 2.0 license, lo cual hace que la elección de este software para su integración en el proyecto a desarrollar no haga incurrir en costes adicionales.

## Casos de éxito usando Graphite

La popularidad de Graphite en el tejido empresarial es notable. Muchas organizaciones lo han escogido para desarrollar sus soluciones de monitorización, como, por ejemplo, Etsy, Booking.com, GitHub o Salesforce.

## Chart.js

Chart.js es una biblioteca JavaScript gratuita de código abierto para la visualización de datos, que admite diversos tipos de gráficos. Creado en Londres por Nick Downie en 2013, ahora es mantenido por la comunidad y es la segunda biblioteca de gráficos JS más popular en GitHub por el número de estrellas después de D3.js. Chart.js se representa en el <canvas>HTML5 y es reconocido como una de las mejores bibliotecas de visualización de datos. Está disponible bajo la licencia MIT.

### Características

- Gran ligereza y rapidez.
- Gráficos responsive.
- Documentación clara.
- Soporte a *plugins*.
- Características limitadas.

### Coste del software

La adopción y el uso de Chart.js en el proyecto no lleva asociado ningún coste adicional al ser OpenSource.

### Casos de éxito usando Chart.js

La popularidad de Chart.js en su segmento de mercado hace que esta herramienta esté siendo utilizada por múltiples organizaciones reconocidas mundialmente como GoDaddy, Deloitte o Edmodo.

## Netdata

Netdata es una herramienta para visualizar y monitorizar métricas en tiempo real, optimizada para acumular todo tipo de datos, como uso de CPU, actividad de disco, consultas SQL, visitas a un sitio web, etc. Consiste en un demonio que, al ejecutarse, se encarga de obtener información en tiempo real, por segundo, y presentarla en un sitio web para su visualización y análisis. La presentación es interactiva y en tiempo real.

Puede ejecutarse en cualquier kernel GNU/Linux para monitorizar cualquier sistema o aplicación, siendo posible correrla en PCs, servidores o dispositivos embebidos Linux.

### Características

- Ligera en ejecución.
- *Plugins* instalados por defecto, además de posibilidad de instalar otros.
- Posibilidad de embeberse en páginas web.
- Personalizable mediante lenguaje HTML.
- Simplicidad de uso.

- Posibilidad de definición de alarmas.

### Coste del software

El software Netdata es gratuito para su utilización en cualquier entorno, por lo que, nuevamente, no se volvería a incurrir en gastos derivados de la implementación de esta nueva funcionalidad.

### Solución escogida

Producto	Tipos de gráficos	Coste	<i>Plugins</i>	Alarmas	Casos de éxito
Grafana	Muchos	Gratuito	Sí	Sí	Sí
Graphite	Muchos	Gratuito	Sí	No	Sí
Chart.js	Limitados	Gratuito	Sí	No	Sí
Netdata	Muchos	Gratuito	Sí	Sí	No

Cuadro 9: Comparativa de sistemas de visualización y formato de datos métricos

El análisis anterior ha permitido conocer en mayor profundidad las ventajas e inconvenientes de la adopción de cada una de las soluciones analizadas. Por ejemplo, variables como los casos de éxito de empresas reputadas utilizando la herramienta analizada permite verificar si el sistema es realmente un producto solvente con capacidad de implantación en un entorno real para organizaciones con requisitos concretos o no. Por otro lado, el coste del software, como se ha comentado con anterioridad, es otro factor determinante en la toma de la decisión de elección de la herramienta a utilizar debido a las restricciones en materia de financiación que presenta un producto nuevo como el que se está desarrollando.

Así pues, el primero de los productos que se descarta en base a estos criterios es Netdata ya que los resultados obtenidos a partir de una búsqueda exhaustiva de empresas que hayan decidido con anterioridad elegir ese software para su organización en la propia página web de la empresa, así como en Internet en general no ha ofrecido resultados convincentes.

Por otro lado, se ha descartado la utilización de Chart.js por varios motivos. El primero de ellos es por contar con características limitadas al ser una herramienta simple en JavaScript. Esto hace que aunque pueda resultar suficientemente solvente al comienzo, si en el futuro se buscara ampliar su uso o se detectaran nuevas necesidades, la herramienta pudiese resultar limitante para ello y se tuviese que mover todo el sistema de visualización y mostrado de datos temporales a otra herramienta. En segundo lugar, Chart.js es simplemente una herramienta a la que se le pasan los datos desde el backend manualmente y genera el gráfico correspondiente con alguna funcionalidad adicional como mostrar información más detallada al pasar el ratón por encima del gráfico concreto. Esto hace el proceso notablemente más laborioso y lioso que en el caso del uso de una herramienta que recolecte los datos de forma autónoma y la cual sea llamada mediante una HTTP API para embeber los gráficos en la página web. Asimismo, la adopción de una herramienta tipo Grafana o Graphite, basados en dashboards,

en lugar de Chart.js permite ir adquiriendo soltura en el manejo de la herramienta y, en el futuro, si se creyese conveniente, se podría definir fácilmente un dashboard con indicadores del uso del software Stics desarrollado en el proyecto para obtener un panel de administración de la herramienta que permita conocer su evolución.

Por último, la elección entre Grafana y Graphite ha sido realizada en base a características más concretas ya que son productos bastante similares. No obstante, la imposibilidad de definir alarmas y el uso de una base de datos propia para almacenar los datos que posteriormente se mostrarán en los gráficos, con el consecuente uso de almacenamiento extra hace que el software escogido finalmente sea Grafana.

## **B.4. Servidores de Sistemas de Información Geográficos**

A continuación se encuentra el análisis pormenorizado realizado de los servidores de Sistemas de Información Geográficos que se consideraron para implementar la aplicación.

### **ArcGIS Server**

Según figura en el sitio web de la herramienta, “ArcGIS Server es un componente de software de servidor back-end de ArcGIS Enterprise que pone su información geográfica a disposición de otras personas de su organización y, de manera opcional, a disposición de cualquiera con una conexión a Internet”.

Se puede implementar ArcGIS Server como sistema independiente simplemente para proporcionar a los usuarios servicios SIG, o bien integrarlo con la plataforma ArcGIS Enterprise como una implementación completa de SIG web. La integración se realiza federando uno o varios sitios de ArcGIS Server con un portal de ArcGIS Enterprise.

### **Características**

- ArcGIS Server controla los modelos de seguridad y uso compartido. Implementación de control de acceso, privilegios de publicación, etc.
- Posibilidad de distribuir los componentes en varias máquinas.
- Balanceo de carga por parte del servidor web. Compatible con los servidores web más populares.

### **Coste del software**

Al ser software privativo, existe un coste asociado a la licencia del producto. En el sitio web de la herramienta no se ofrece información detallada sobre el precio pero una licencia única de ArcGIS for Desktop y un usuario de ArcGIS Online cuesta \$1500

### **Casos de éxito**

Múltiples ciudades estadounidenses han implementado el uso de software ArcGIS para

diversos proyectos. Incluso, la Universidad de Zaragoza cuenta con licencias de la herramienta utilizadas, por ejemplo, en el grado de Geología para llevar a cabo diversos proyectos.

## GeoServer

GeoServer es un servidor de código abierto escrito en Java permite a los usuarios compartir y editar datos geospaciales. Diseñado para la interoperabilidad, publica datos de las principales fuentes de datos espaciales usando estándares abiertos.

GeoServer sirve de implementación de referencia del estándar Open Geospatial Consortium Web Feature Service, y también implementa las especificaciones de Web Map Service y Web Coverage Service. Fue diseñado para que la ciudadanía pudiese implicarse en el gobierno y el planeamiento urbano en el año 2001 por medio del Proyecto de Planificación Abierta.

### Características

- Implementación de los principales protocolos definidos por el OGC.
- Integración en GeoServer de OpenLayers para facilitar la generación de mapas.
- Posibilidad de adición de funcionalidad mediante *plugins*.
- Integración con APIs de mapping como OpenLayers y Leaflet pero también con Google Maps, Google Earth, Bing Maps y MapBox.
- Posibilidad de conexión con arquitecturas GIS tradicionales como ESRI ArcGIS.

### Coste del software

Al ser software OpenSource, no existe coste asociado al uso del software

### Casos de éxito

Ordnance Survey, la agencia cartográfica nacional del Reino Unido, utiliza GeoServer.

## GvSIG

GvSIG es un proyecto de desarrollo de software para Sistemas de Información Geográfica basado en software libre. gvSIG Desktop fue el primer software que se desarrolló dentro del proyecto gvSIG. Además de gvSIG Desktop se han unido al catálogo de productos gvSIG aplicaciones como gvSIG Mobile o gvSIG Online. Este proyecto fue inicialmente impulsado por el gobierno regional de la Comunidad Valenciana, dentro de un proceso de migración a software libre de todos los sistemas informáticos de la organización.

### Características

- Herramientas SIG de carga de datos, navegación, información, etc.
- Interoperabilidad. Soporte a los principales formatos.



- Herramientas de diseño asistido por ordenador avanzadas integradas.
- Implementación de los principales protocolos del OGC.

### Coste del software

Al ser software OpenSource, no existe coste asociado al uso del software

### Casos de éxito

gvSIG es, lógicamente utilizado por el gobierno regional de la Comunidad Valenciana.

### Solución escogida

El estudio realizado ha permitido conocer en mayor profundidad las ventajas y desventajas del uso de cada uno de los productos analizados. Por ejemplo, se ha podido analizar si la adopción de un software privativo como es el caso de ArcGIS Server podría llegar a resultar rentable en base a las funcionalidades ofrecidas o no. Además, se ha intentado, en la medida de las posibilidad, analizar herramientas de disinta naturaleza. Como ya se ha comentado, al analizar ArcGIS se estudia software propietario, que junto a GeoServer, software libre, y gvGIS, también software libre pero fruto de un proyecto de menor entidad y de desarrollo regional, se logra obtener una fotografía clara del estado del arte en este sentido, permitiendo tomar la decisión más ajustada en base a las necesidades detectadas. A continuación, se incluye una tabla en la que se puede observar los protocolos implementados por las herramientas ArcGIS y GeoServer [51].

Name	Language	WMS	WFS	WFS-T	WCS	WMC	SLD	FES	Other
ArcGIS Server	.NET/Java	Yes	Yes	Yes	Yes	No	Yes	No	SOAP, REST, KML
GeoServer	Java	Yes	Yes	Yes	Yes	Yes	Yes	Yes	

Figura 21: Tabla comparativa de software GIS

De la tabla anterior se extrae que las dos herramientas comparadas implementan los principales protocolos definidos por el OGC pero que, sin embargo, GeoServer implementa dos más que no implementa ArcGIS, lo cual da indicios de poder tratarse de un producto más completo. Aunque no aparezca en la tabla, gvSIG también implementa, como se ha comentado, los principales protocolos.

No obstante, el uso de gvGIS se ha descartado por tratarse de un producto que, si bien podría cumplir con las necesidades, al ser de desarrollo regional, puede existir menor documentación y soporte de la comunidad.

En cuanto a ArcGIS Server y GeoServer, un estudio realizado por Jennifer R. Bauer para un máster de ciencias de la geografía de la Universidad de Oregón comparó ArcGIS Server y GeoServer en diferentes escenarios, de entre los cuales, los tiempos de respuesta y tamaños resultantes utilizando WFS y *getFeature*. A continuación, se presentan los resultados obtenidos [52].

		WFS Requests average file size (Mb) and download time (s)					
Web Mapping Server	County	Get Capabilities		Describe Feature Type		Get Feature	
		file size (Mb)	time (s)	file size (Mb)	time (s)	file size (Mb)	time (s)
<b>MapServer</b>	Ashland	0.03	0.117	0.01	0.096	89.26	1.405
	Bayfield	0.03	0.097	0.01	0.095	293.61	8.813
	Douglas	0.03	0.099	0.01	0.095	456.72	5.472
	Iron	0.03	0.098	0.01	0.096	3.93	0.204
<b>GeoServer</b>	Ashland	0.10	0.011	0.01	0.007	78.97	0.536
	Bayfield	0.10	0.012	0.01	0.010	263.02	1.676
	Douglas	0.10	0.012	0.01	0.009	427.96	3.041
	Iron	0.10	0.010	0.01	0.008	6.77	0.048
<b>ArcServer</b>	Ashland	0.02	0.403	0.01	0.397	58.52	4.734
	Bayfield	0.02	0.402	0.01	0.400	342.04	24.931
	Douglas	0.02	0.396	0.01	0.417	538.88	38.738
	Iron	0.02	0.423	0.01	0.411	8.26	1.096

Figura 22: Tabla comparativa de resultados WFS

De la tabla anterior se extrae el sensible aumento de tamaño en los ficheros resultantes cuando se utiliza ArcGIS Server en lugar de GeoServer. Además, los tiempos son notablemente superiores en la primera (38.738s frente a 3.041s en el caso del condado de *Douglas*).

## B.5. Integración de soluciones existentes

A continuación se encuentra la investigación realizada de las soluciones existentes que se pudiesen integrar en el proyecto a desarrollar.

### EINAmbiente

EINAmbiente consiste en un proyecto realizado en las prácticas de la asignatura de tercer curso **Sistemas de Información I**.

El propósito principal, coincidente con el del presente trabajo, era el de ofrecer una plataforma en la que interaccionan alumnado y profesorado, en este caso, de ámbito universitario, entorno a un tema común, el medio ambiente. En ella, el alumnado publicaba los carteles desarrollados en la asignatura **Ingeniería del medio ambiente** y reaccionaba a los de sus compañeros, bien publicando comentarios, bien respondiendo a los retos planteados.

Esta idea de responder a cuestiones planteadas, que, en este caso, serán formuladas por el profesorado responsable, ha sido integrada en el presente trabajo, tal y como se comentará con posterioridad.

## Building blocks

CEF, o “Connecting Europe Facility”, es un instrumento financiero clave de la Unión Europea para promover el crecimiento y competitividad. Dispone también de una parte digital que invierte grandes sumas de dinero en el desarrollo de “Building Blocks” [53] digitales, infraestructuras y redes de banda ancha.

En el marco de estos “Building Blocks”, se encuentran diversas soluciones, como, Big Data, firma digital, traducción automática e identificación digital, entre otros.

Para este proyecto, se detectó un interés en la integración, en primer lugar, del sistema eID, ya que, al tratarse de un trabajo con vocación de implantación en centros públicos, se creyó interesante integrar esta solución, que ya es utilizada por el funcionario público en muchas de sus gestiones diarias.

No obstante el Building Block eID se trata de una solución a nivel europeo que, después, cada Estado Miembro debe trasladar a su territorio. En España, esto se tradujo en la creación del sistema Cl@ve, que permite la identificación de usuarios mediante una clave ocasional, o PIN, o mediante una clave permanente.

La gran ventaja de que sea una solución internacional es que cualquier usuario de un Estado Miembro que haya adoptado este sistema puede acceder a servicios de otro país con su clave personal.

El otro Building Block que resultó de interés fue el servicio eTranslation. Este sistema abre la posibilidad a que los usuarios suban documentos a la plataforma, que, por su parte, lleva a cabo una traducción automática de estos, manteniendo el formato de origen en la mayor parte de los casos.

Esto hace posible el intercambio de información entre instituciones públicas de la UE sin que las barreras idiomáticas sean un impedimento para ello.

### Problemática asociada y decisión adoptada

Indudablemente, el servicio de Building Blocks supone una gran herramienta y un paso adelante en la tarea de facilitar la intercomunicación entre las diferentes Administraciones Públicas. No obstante, este último punto supone el ineludible impedimento para su integración en el presente proyecto.

Para hacer uso de estos servicios, como se hace saber en el sitio web oficial, hay que, en primer lugar crear una cuenta “EU Login” para posteriormente enviar una petición con los credenciales creados, certificando que el usuario es un funcionario trabajando en una Administración Pública de la Unión Europea, Islandia o Noruega, tal y como se observa a continuación.

Send a request to [help@cefat-tools-services.eu](mailto:help@cefat-tools-services.eu) with your “EU login” credentials, certifying that you are a civil servant working in a public administration from EU, Iceland or Norway

Es por esto que la utilización de los Building Blocks en el proyecto quedó descartada.

## **Plataformas de gestión educativa**

Uno de los casos de uso de la aplicación a desarrollar en el caso de la administración del centro educativo es el de añadir alumnado a uno o varios grupos en función de las asignaturas en que se decida implantar la utilización del software. No obstante, la labor de añadir el alumnado uno a uno a cada uno de los grupos puede resultar tediosa y redundar en una insatisfacción del cliente con el sistema, con lo que se lograría una menor implantación de la aplicación en los centros educativos.

Es por esto por lo que es necesaria la investigación de soluciones de integración entre las plataformas de gestión docente implantadas en los centros y el software planteado. Esto permitiría facilitar la labor planteada ya que dicha asignación de alumnado a grupos se podría llevar a cabo de forma masiva.

### **Solución adoptada**

La investigación de alternativas se ha llevado a cabo teniendo en cuenta la herramienta de gestión educativa utilizada en la Comunidad Autónoma de Aragón: SIGAD. Se han analizado las distintas posibilidades de integración, desde la comprobación de la existencia de una API pública con la que interactuar directamente para obtener el funcionamiento deseado hasta la posibilidad de exportar datos en un fichero con la información necesaria que después se pueda cargar en el sistema. Las conclusiones obtenidas son las siguientes:

- No existe una API pública con la que interactuar directamente sin intermediación de personal administrativo.
- Existe la posibilidad de descargar un fichero en formato .csv con la información necesaria para llevar a cabo la carga de estos datos en el sistema mediante la carga del fichero utilizando para ello un formulario. disponible a la administración de cada centro.

Por todo ello se ha decidido adoptar la opción de carga masiva mediante fichero. No obstante, para evitar problemas en la carga derivados de errores en la descarga de datos de la aplicación de gestión se realizará un manual de usuario disponible para su descarga desde el propio formulario de carga de datos.

Debido a la voluntad de expansión de la plataforma a otras CCAA o incluso en otros países en las que las plataformas de gestión educativa utilizadas sean diferentes, se indicará específicamente el formato de fichero que se acepta en la aplicación para evitar problemas que puedan inducir a errores.

## C. Casos de uso del sistema

En el presente anexo se incluye, en primer lugar, el diagrama de casos de uso del sistema, en el que se aprecian los diferentes tipos de usuarios identificados con anterioridad. Además, se adjunta la descripción de escenarios principal y alternativo(s) de cada uno de los casos de uso presentes.

### C.1. Diagrama de casos de uso

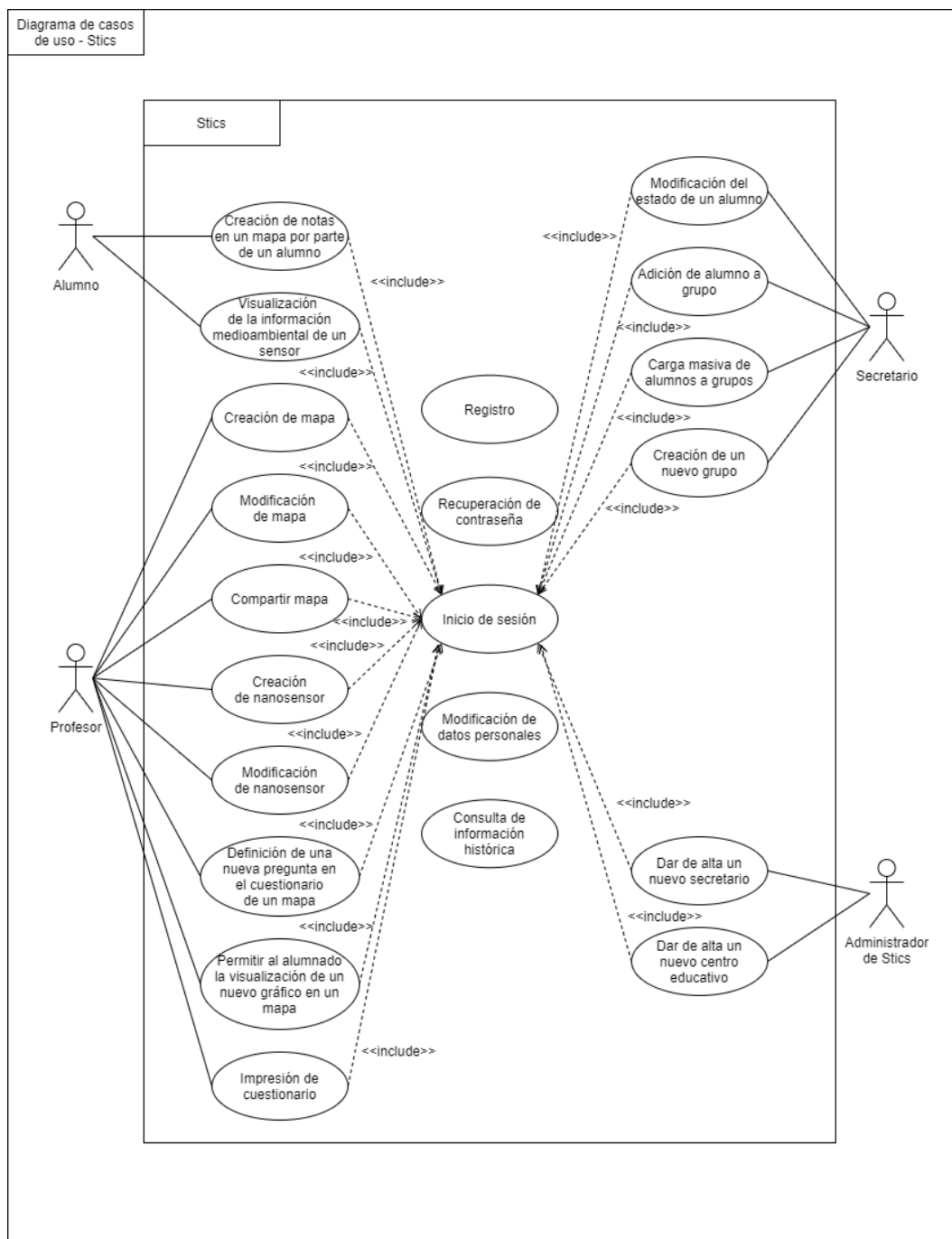


Figura 23: Diagrama de casos de uso

Los casos de uso “Registro”, “Recuperación de contraseña”, “Modificación de datos personales”, “Consulta de información histórica” pueden ser realizados por el alumnado, profesorado y administración del centro. No se incluyen las flechas correspondientes por legibilidad del diagrama.

## **C.2. Descripción de escenarios**

Como se ha comentado previamente, a continuación se encuentra la descripción de escenarios correspondiente a los casos de uso anteriores, incluyendo los usuarios afectados, así como las precondiciones y escenario principal y alternativo(s), si ha lugar.

### **Creación de notas en un mapa por parte del alumnado**

Usuarios afectados: alumnado

Precondiciones: el usuario ha iniciado sesión y ha pulsado en un mapa

Escenario principal:

1. El usuario introduce el texto que desea en el cuadro de texto dispuesto a tal efecto en la sección “Notas y comentarios”.
2. El usuario pulsa en “Guardar”, finalizando el caso de uso.

### **Visualización de la información medioambiental de un sensor**

Usuarios afectados: profesorado y alumnado

Precondiciones: el usuario ha iniciado sesión y ha pulsado en un mapa

1. El usuario activa una o varias capas de sensores medioambientales.
2. El usuario pulsa en el sensor que desee.
3. Al pulsar, el cuadro de mandos carga automáticamente, mostrándose los gráficos asociados a los datos recogidos por esa estación, finalizando el caso de uso.

### **Creación de mapa**

Usuarios afectados: profesorado

Precondiciones: El usuario ha iniciado sesión

Escenario principal:

1. El usuario elige la localización de entre las disponibles en el sistema.
2. El usuario introduce el nombre del nuevo mapa.
3. El usuario selecciona las capas de datos de sensores que desea activar en el nuevo mapa y pulsa en “Guardar”, finalizando el caso de uso.

### **Modificación de mapa**

Usuarios afectados: profesorado

Precondiciones: El usuario ha iniciado sesión

Escenario principal:

1. El usuario pulsa en el botón “editar” del mapa que desee.
2. Al abrirse el popup de edición, el usuario introduce los nuevos datos del mapa.
3. El usuario pulsa en “Guardar cambios”, finalizando el caso de uso.

Escenario alternativo: Cancelación de edición

1. El usuario pulsa en “Descartar”, finalizando el caso de uso.

### **Compartir mapa**

Usuarios afectados: profesorado

Precondiciones: El usuario ha iniciado sesión

Escenario principal:

1. El usuario pulsa en el cuadro de texto al lado de cada mapa.
2. Se abre un desplegable con los usuarios y grupos en los que el docente participa.
3. El usuario selecciona los grupos o estudiantes con los que quiera compartir el mapa.
4. El usuario pulsa en el botón de compartir, finalizando el caso de uso.

### **Creación de nanosensor**

Usuarios afectados: profesorado

Precondiciones: El usuario ha iniciado sesión

Escenario principal:

1. El usuario selecciona en el mapa el punto en el que se localiza el nuevo nanosensor. Si no, escribe sus coordenadas manualmente en los cuadros de texto dispuestos a tal efecto.
2. El usuario rellena los datos de modelo y estado del sensor.
3. Opcional: el usuario introduce una breve descripción del nuevo nanosensor.
4. El usuario pulsa en “Guardar”, finalizando el caso de uso.

### **Modificación de nanosensor**

Usuarios afectados: profesorado

Precondiciones: el usuario ha iniciado sesión

Escenario principal:

1. El usuario pulsa en el botón de editar nanosensor situado al lado de los datos del nanosensor que desee.
2. Al abrirse el popup, el usuario introduce la nueva información referente al sensor.
3. El usuario pulsa en “Guardar cambios”, finalizando el caso de uso.

Escenario alternativo: Cancelación de edición

1. El usuario desiste de realizar las modificaciones deseadas, pulsando en “Descartar” y finalizando el caso de uso.

### **Definición de una nueva pregunta en el cuestionario de un mapa**

Usuarios afectados: profesorado

Precondiciones: el usuario ha iniciado sesión y pulsado en un mapa

1. El usuario pulsa en el tipo de pregunta que desea de las que se muestran como opciones.
2. El usuario introduce el texto de la pregunta.
3. Si es de tipo textual, introduce el texto de la respuesta. Si no, introduce tantas opciones como desee
4. Si el usuario desea mostrar más o menos opciones, pulsa en los botones de “Añadir respuesta” y “Eliminar respuesta” y selecciona cuál es la respuesta correcta.
5. El usuario pulsa en “Guardar”, finalizando el caso de uso.

Escenario alternativo: Cancelación de creación de la pregunta

1. El usuario pulsa en “Cerrar”, finalizando el caso de uso.

### **Permitir al alumnado la visualización de un nuevo gráfico en un mapa**

Usuarios afectados: profesorado

Precondiciones: el usuario ha iniciado sesión y ha pulsado en un mapa

Escenario principal:

1. El usuario desciende hasta el final de la página.
2. El usuario selecciona el tipo de gráfico que desea añadir al cuadro de mandos.
3. El usuario pulsa en “Añadir”, finalizando el caso de uso.

### **Impresión de cuestionario**

Usuarios afectados: profesorado

Precondiciones: el usuario ha iniciado sesión y ha pulsado en un mapa

Escenario principal:

1. El usuario pulsa en el botón marrón con el icono de la impresora que aparece debajo del título *Quiz* o en el botón dorado si desea la corrección.
2. Comienza la descarga del PDF del cuestionario, finalizando el caso de uso.

### **Modificación del estado de un estudiante**

Usuarios afectados: administración del centro

Precondiciones: el usuario ha iniciado sesión

Escenario principal:



1. El usuario selecciona un estudiante del desplegable en la sección “Modificar el estado de un estudiante”.
2. El usuario selecciona un estado de la lista o introduce uno a mano en el cuadro de texto dispuesto a tal efecto.
3. El usuario pulsa en “Guardar cambios”, finalizando el caso de uso.

### **Adición de alumnado a un grupo**

Usuarios afectados: administración del centro

Precondiciones: el usuario ha iniciado sesión

Escenario principal:

1. El usuario selecciona un estudiante del desplegable de la sección “Alumno por alumno”.
2. El usuario selecciona un grupo del desplegable.
3. El usuario selecciona el botón “Añadir a grupo”, finalizando el caso de uso.

### **Carga masiva de alumnado a grupos**

Usuarios afectados: administración del centro

Precondiciones: el usuario ha iniciado sesión

Escenario principal:

1. El usuario selecciona el grupo al que añadir a los usuarios del desplegable de la sección “Carga múltiple”.
2. El usuario pulsa en el botón “Explorar” para escoger el fichero a cargar.
3. El usuario pulsa en el botón “Cargar fichero”, finalizando el caso de uso.

### **Creación de un nuevo grupo**

Usuarios afectados: administración del centro

Precondiciones: el usuario ha iniciado sesión

Escenario principal:

1. El usuario introduce el nombre del nuevo grupo en el cuadro de texto de la pestaña “Grupos”.
2. El usuario pulsa en “Añadir grupo”, finalizando el caso de uso.

### **Registro en el sistema**

Usuarios afectados: alumnado y profesorado

Precondiciones: —

Escenario principal:

1. Encontrándose el usuario en la página principal de la aplicación, este pulsa en el enlace “darse de alta”.
2. El usuario selecciona el centro educativo al que pertenece.

3. El alumnado selecciona el grupo al que pertenece.
4. El usuario completa el resto de campos de información personal requerida.
5. El usuario selecciona su rol: estudiante o docente.
6. Opcional: el usuario elige la foto de perfil que desea.
7. El usuario pulsa en “Registrarme”, finalizando el caso de uso.

Escenario alternativo: El centro educativo no existe

1. El usuario pulsa en contáctanos y accede al formulario de contacto para resolver el problema.

Escenario alternativo: La dirección de email ya está siendo utilizada

1. La aplicación vuelve a la misma página informando del error.
2. El usuario vuelve a introducir sus datos.

### **Inicio de sesión**

Usuarios afectados: todos

Precondiciones: —

Escenario principal:

1. El usuario introduce su dirección de email y su contraseña.
2. El usuario pulsa en “Entrar”, finalizando el caso de uso.

Escenario alternativo: Los datos introducidos son incorrectos

1. El usuario es informado del error.
2. El usuario vuelve a introducir sus datos y pulsa en “Entrar”.

### **Recuperación de contraseña**

Usuarios afectados: todos

Precondiciones: —

Escenario principal:

1. El usuario pulsa en el enlace destinado a la recuperación de contraseña.
2. El usuario introduce la dirección de email utilizada para registrarse.
3. El usuario pulsa en “Enviar”, finalizando el caso de uso.

Escenario alternativo: No se encuentra la dirección de email introducida

1. El usuario es informado del error por el sistema.
2. El usuario introduce la dirección de email correcta y pulsa de nuevo en “Enviar”.

### **Modificación de datos personales**

Usuarios afectados: todos

Precondiciones: el usuario ha iniciado sesión

Escenario principal:

1. El usuario desciende hasta la parte de abajo de la página principal e introduce los datos personales que desea modificar.
2. El usuario pulsa en “Guardar”, finalizando el caso de uso.

### **Consulta de información histórica**

Usuarios afectados: todos

Precondiciones: el usuario ha iniciado sesión y se encuentra en la página principal

Escenario principal:

1. El usuario pulsa en el desplegable de cursos y pulsa en el que desea consultar.
2. El usuario pulsa en “Ir a curso”, finalizando el caso de uso.

## D. Prototipado wireframe

A continuación se muestran los prototipos *wireframe* de cada una de las pantallas de la aplicación. Como ya se ha comentado, fueron realizados con la herramienta de prototipado rápido Balsamiq Cloud.

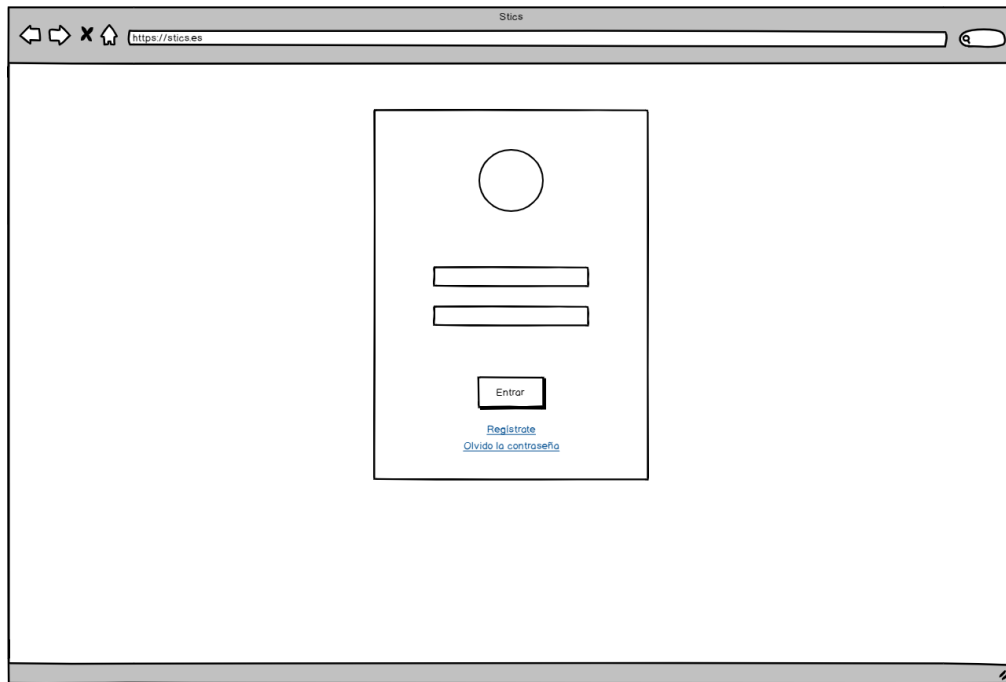


Figura 24: Login de usuarios

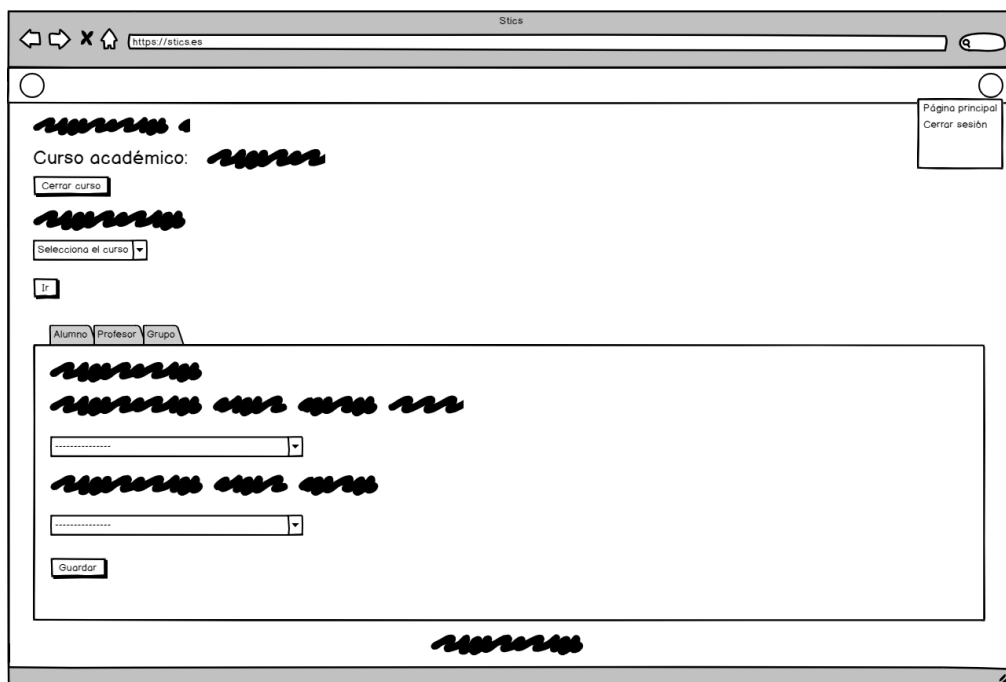


Figura 25: Vista de administración de centro - Gestión de usuarios

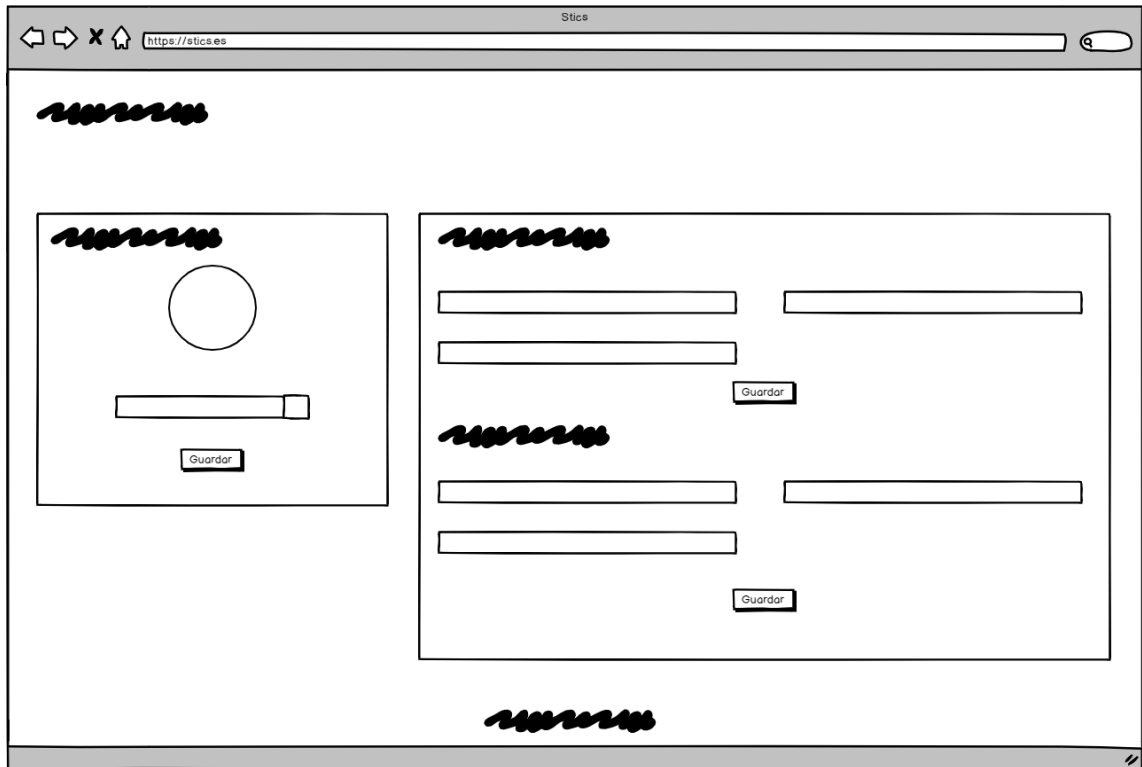


Figura 26: Vista de administración de centro - Modificación de datos personales

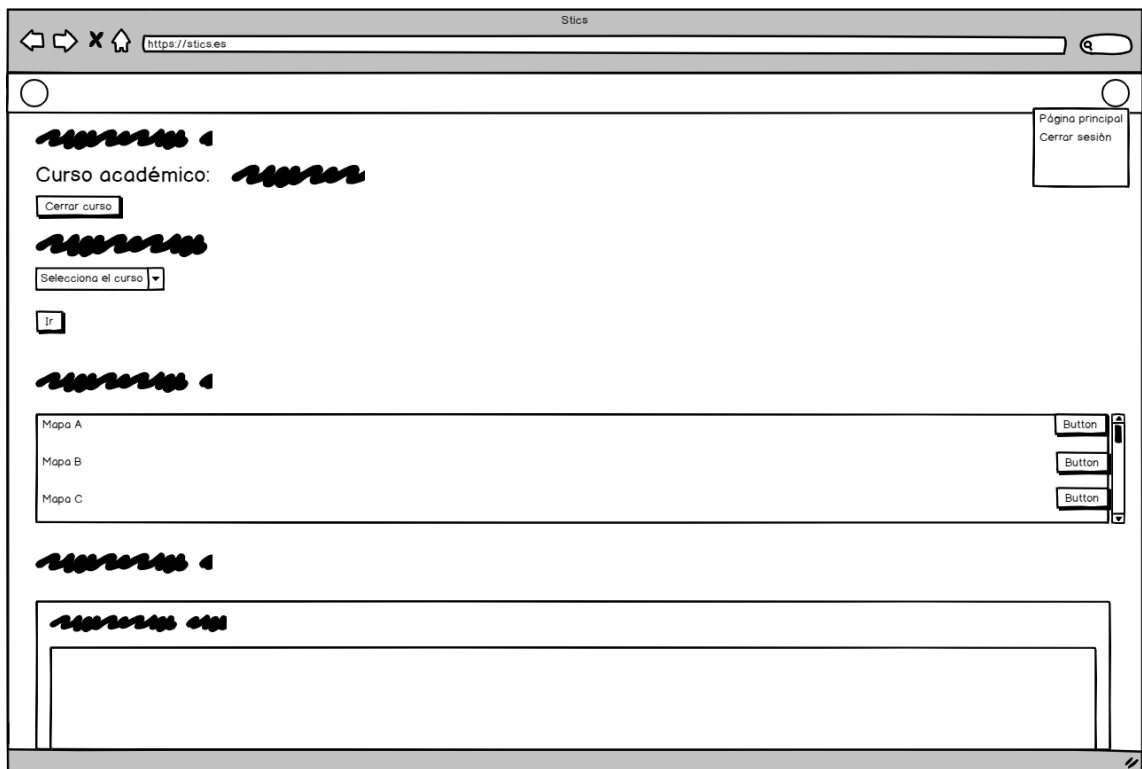


Figura 27: Vista de docente - Gestión de histórico y mapas

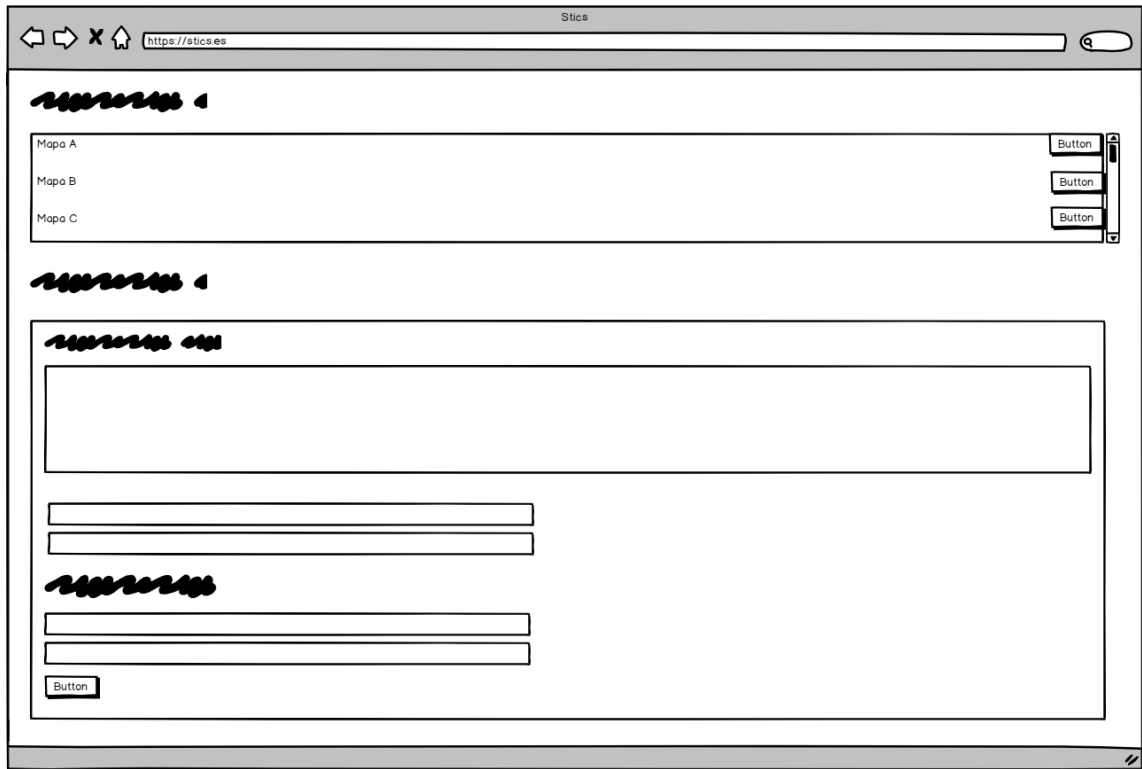


Figura 28: Vista de docente - Gestión de nanosensores propios

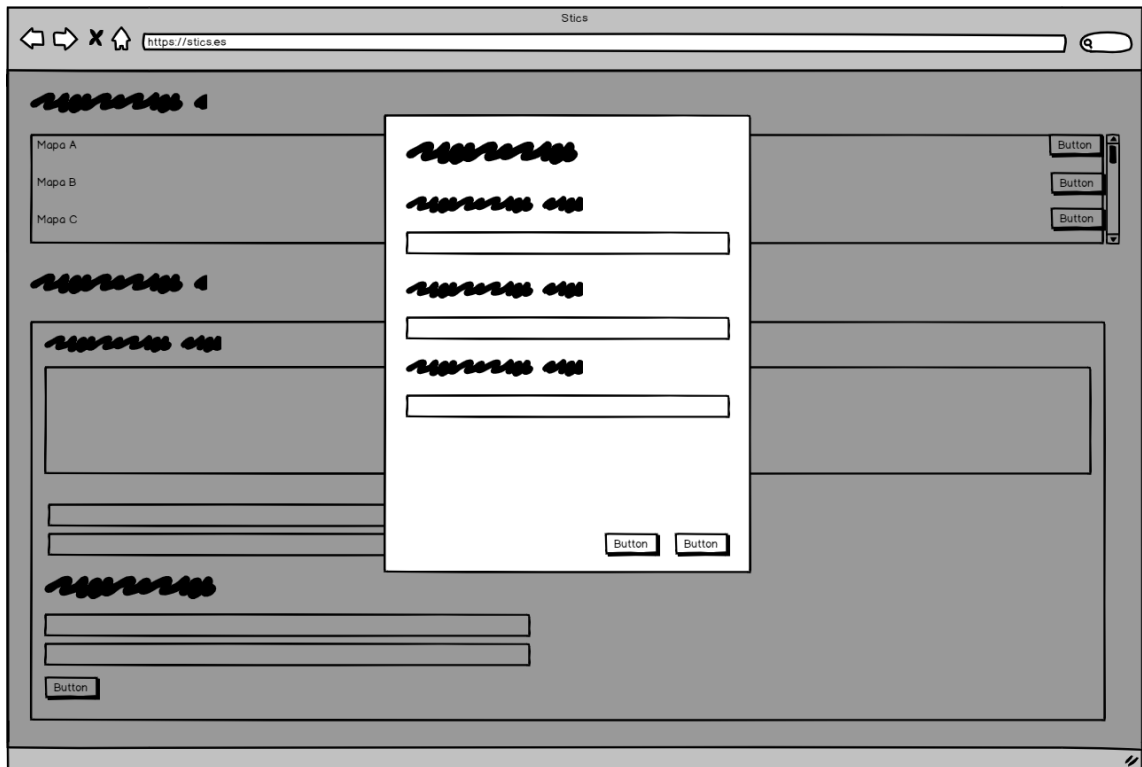


Figura 29: Popup de modificación de información

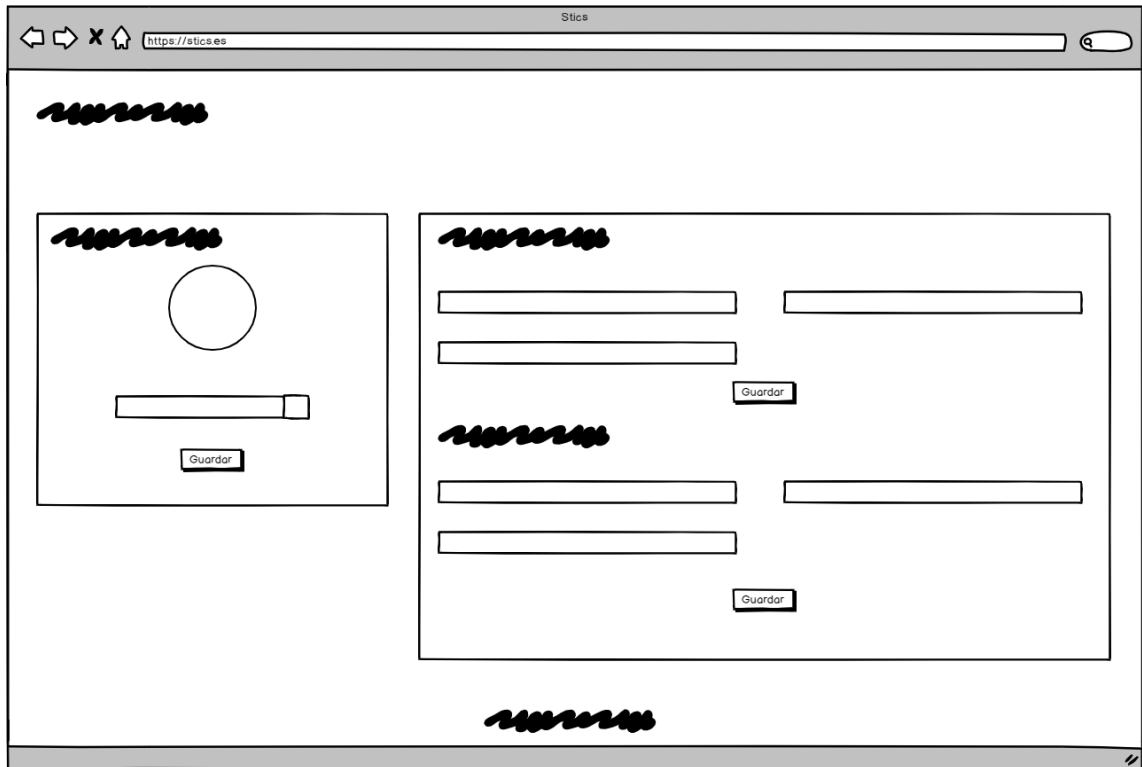


Figura 30: Vista de docente - Modificación de datos personales



Figura 31: Vista de docente - Cuadro de mandos

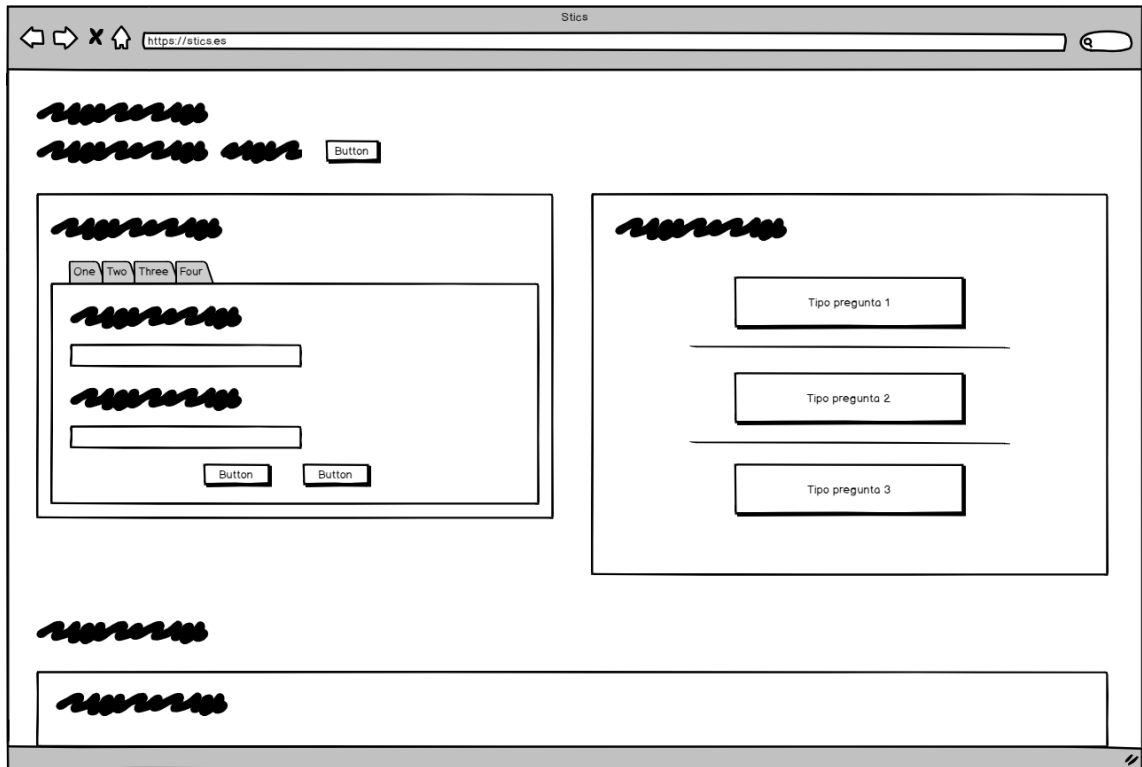


Figura 32: Vista de docente - Gestión de quiz en el mapa

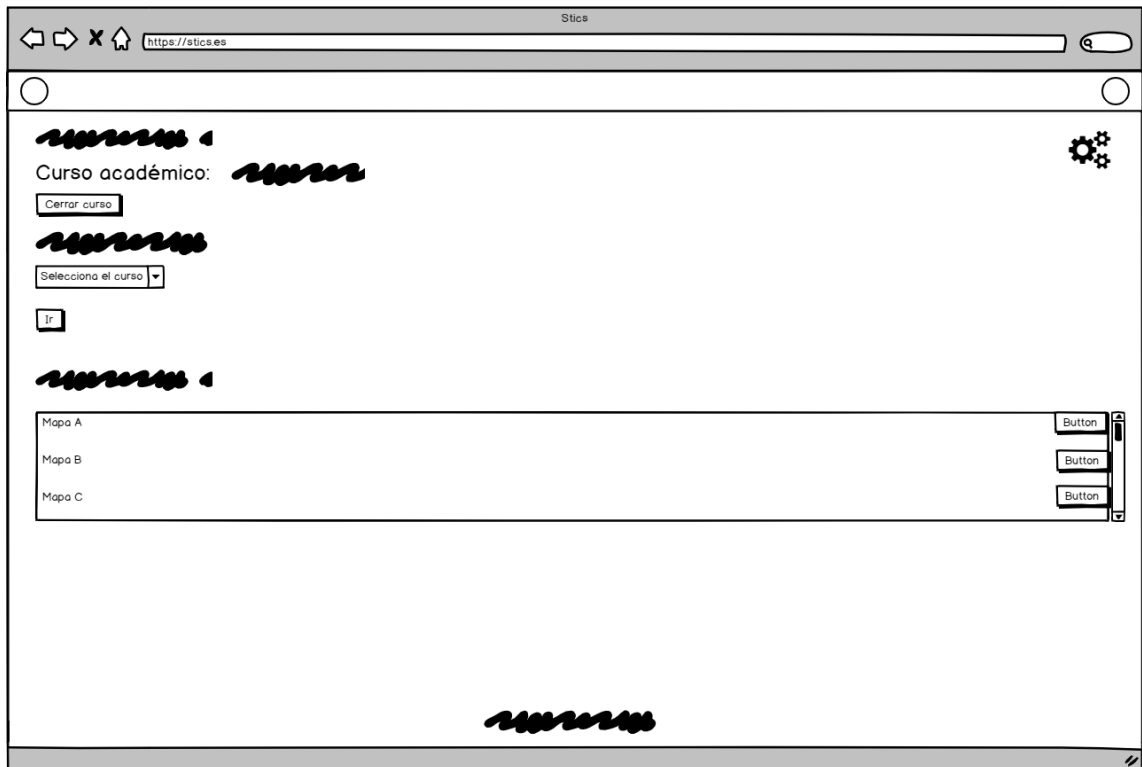


Figura 33: Vista de estudiante - Consulta de histórico y de mapas



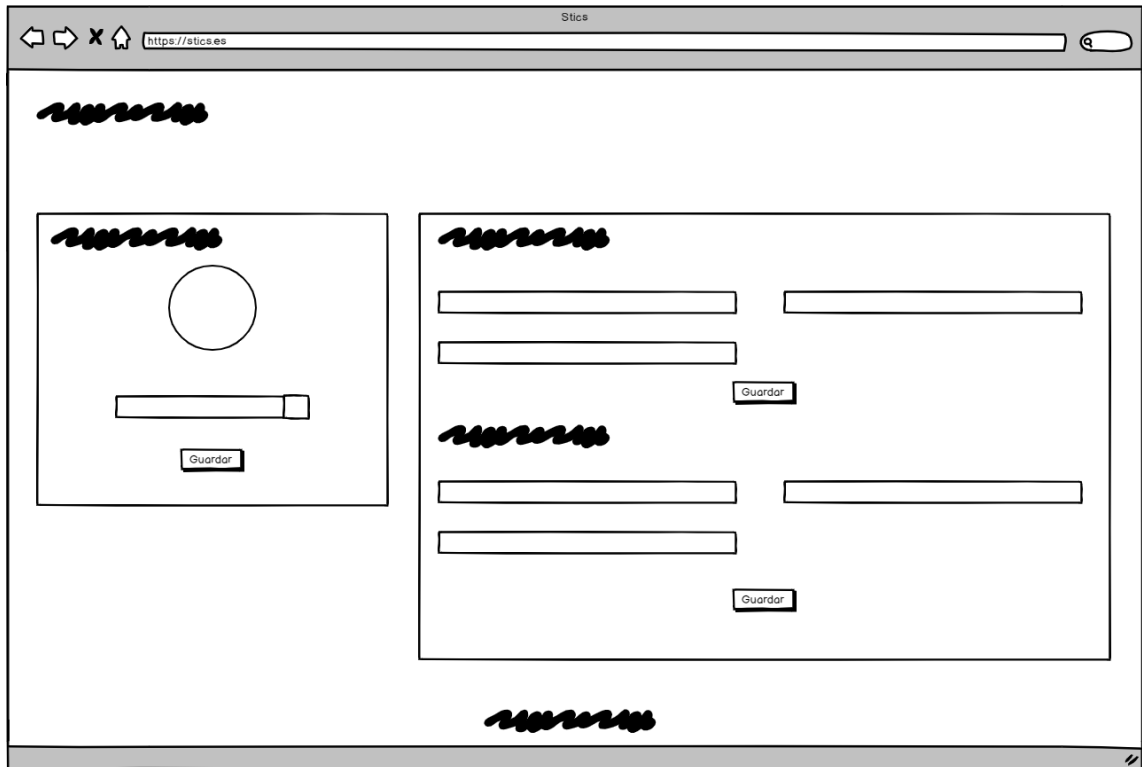


Figura 34: Vista de estudiante - Modificación de datos personales

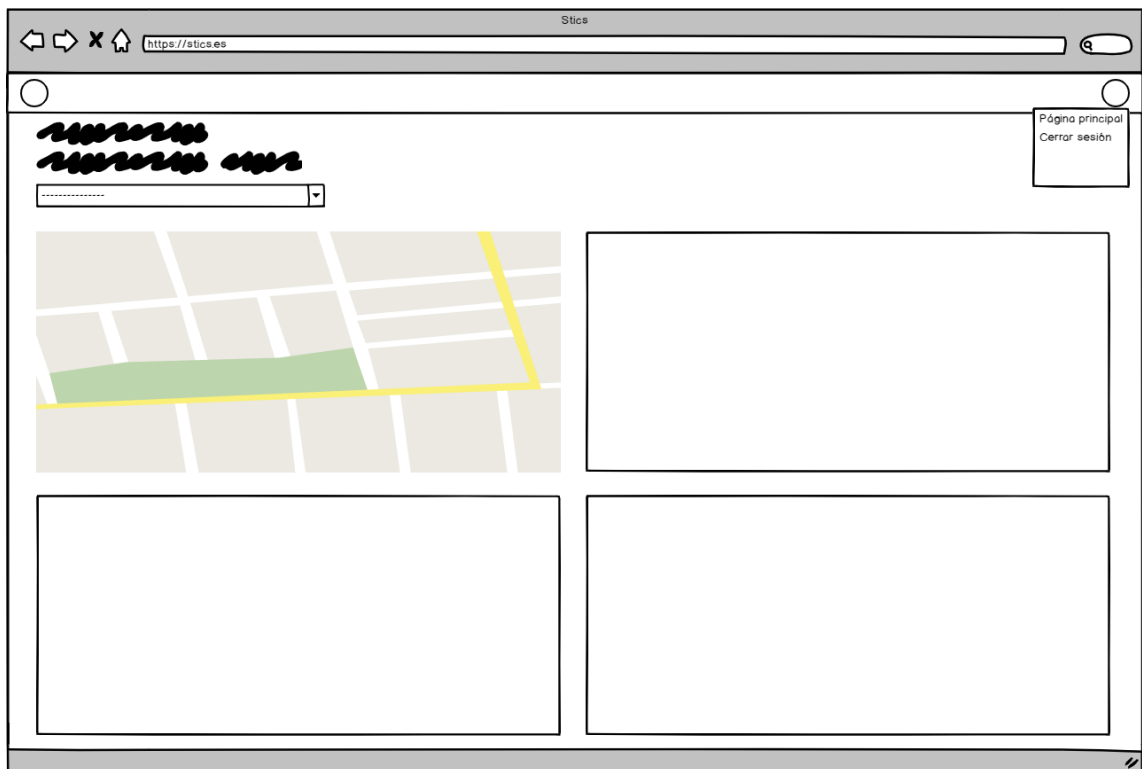


Figura 35: Vista de estudiante - Cuadro de mandos

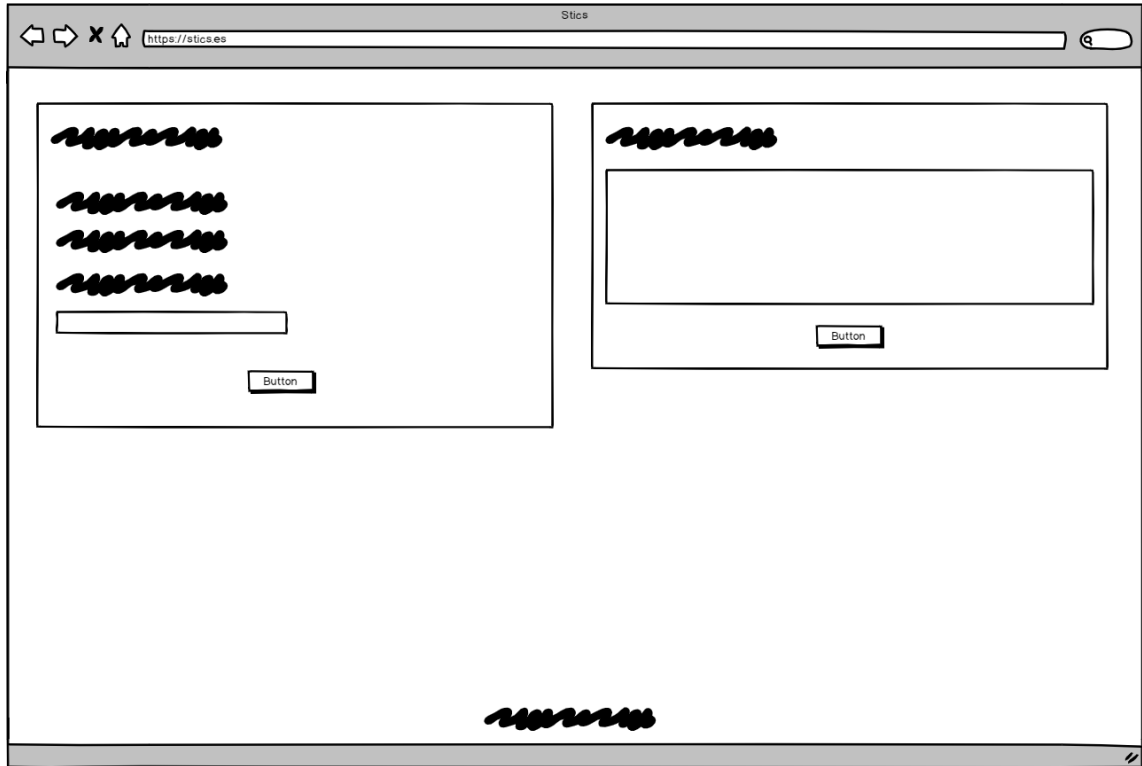


Figura 36: Vista de estudiante - Quiz y notas en el mapa

## E. Gestión del proyecto

La gestión de proyectos es la disciplina que estudia el planeamiento, la organización, la motivación y el control de los recursos con el propósito de alcanzar uno o varios objetivos. Este es el objeto de la presente sección. A continuación, en primer lugar, se muestra una tabla con los esfuerzos realizados en el presente proyecto. En segundo lugar, se muestran dos modelos de presupuesto, en base a cada una de las soluciones ofrecidas al cliente como producto.

### E.1. Esfuerzos realizados

A continuación se incluye la tabla de distribución de esfuerzos realizados para el desarrollo del presente proyecto.

Fecha	Hora comienzo	Hora final	Duración	Descripción
14/1/2020	9:25	11:00	1:35	Análisis de requisitos
18/1/2020	9:15	12:45	3:30	Análisis de requisitos y diseño web
19/1/2020	9:30	12:15	2:45	Estudio del estado del arte
20/1/2020	9:00	12:30	3:30	Estudio del estado del arte
21/1/2020	16:30	19:00	2:30	Estudio del estado del arte
22/1/2020	9:00	13:30	4:30	Estudio del estado del arte
25/1/2020	10:00	12:00	2:00	Análisis de requisitos y estado del arte
4/2/2020	9:00	11:00	2:00	Análisis de requisitos y estado del arte
5/2/2020	9:00	12:00	3:00	Diseño web
6/2/2020	14:30	18:30	4:00	Diseño web
8/2/2020	9:00	13:30	4:30	Diseño web
10/2/2020	9:00	12:10	3:10	Diseño web
12/2/2020	9:30	12:00	2:30	Diseño web
18/2/2020	10:00	12:00	2:00	Diseño web
18/2/2020	17:00	20:30	3:30	Diseño web
19/2/2020	9:00	12:00	3:00	Diseño web
22/2/2020	9:00	12:00	3:00	Diseño web
22/2/2020	16:00	19:30	3:30	Diseño web
23/2/2020	10:30	13:00	2:30	Diseño web
6/3/2020	9:00	11:50	2:50	Diseño web
9/3/2020	18:00	19:30	1:30	Diseño web
11/3/2020	10:30	11:30	1:00	Estudio del estado del arte
15/3/2020	9:00	12:00	3:00	Diseño web
16/3/2020	9:00	10:00	1:00	Esquema E/R de base de datos
17/3/2020	15:00	19:00	4:00	Tutorización y diseño web
18/3/2020	9:00	12:00	3:00	Creación de servidor y configuración inicial
18/3/2020	18:00	19:00	1:00	Configuración inicial
21/3/2020	16:30	17:30	1:00	Plan de gestión de riesgos
22/3/2020	16:00	19:00	3:00	Creación de base de datos
23/3/2020	9:00	12:30	3:30	Registro y conexión de usuarios
23/3/2020	18:00	19:30	1:30	Generación dinámica de datasets de usuarios
24/3/2020	9:00	11:30	2:30	Generación dinámica de datasets de usuarios
24/03/2020	16:30	18:15	1:45	Modificación de datos personales dinámica
25/03/2020	10:00	12:30	2:30	Creación y eliminación de mapas
25/03/2020	16:30	21:00	4:30	GeoServer, publicación de shapefiles y muestr
26/03/2020	9:00	11:30	2:30	Estudio del estado del arte OpenLayers / Leafli
27/03/2020	10:00	13:00	3:00	GeoServer, publicación de shapefiles y muestr
27/03	16:30	20:30	4:00	Configuración de Leaflet y uso de plugin
28/03/2020	9:00	13:00	4:00	Búsqueda, descarga y configuración de shapef
29/03/2020	9:30	13:00	3:30	Modificación de datos personales dinámica er
29/03/2020	16:30	19:30	3:00	Compartición de mapas con alumnos
02/04/2020	9:00	12:30	3:30	Modificación de la estética de la tabla de map
04/04/2020	10:00	11:00	1:00	Quiz - Definición de la persistencia de las preg
04/04/2020	17:00	19:30	2:30	Quiz - Definición de la persistencia de las preg
06/04/2020	10:00	12:30	2:30	Quiz - Mostrado dinámico de preguntas
06/04/2020	15:30	19:00	3:30	Quiz - Mostrado dinámico de preguntas y elirr
09/04/2020	9:00	13:00	4:00	Instalación de base de datos Trafair
09/04/2020	16:00	19:00	3:00	Instalación de base de datos Trafair
10/04/2020	9:30	13:00	3:30	Conexión de base de datos Trafair con Grafan
11/04/2020	15:30	19:00	3:30	Generación de cuestionario en PDF

Figura 37: Distribución de esfuerzos (1)

12/04/2020	16:00	19:45	3:45 Peticiones a Grafana desde JavaScript
13/04/2020	10:00	13:30	3:30 Peticiones a Grafana desde JavaScript
13/04/2020	17:00	20:30	3:30 Instalación de nginx y configuración como pro
14/04/2020	9:00	13:00	4:00 Configuración de nginx como proxy inverso pa
14/04/2020	16:30	18:00	1:30 Resolución de errores de configuración de ngi
16/04/2020	17:00	19:30	2:30 Finalización de conexión js - nginx - grafana
17/04/2020	9:00	13:00	4:00 Definición de tipos de gráficos
17/04/2020	16:30	19:00	2:30 Definición de tipos de gráficos
18/04/2020	9:00	13:00	4:00 Persistencia de gráficos y conexión frontend b
18/04/2020	17:00	19:30	2:30 Estudio de la API de Grafana
19/04/2020	10:00	13:30	3:30 Estudio de la API de Grafana
19/04/2020	17:00	20:00	3:00 Estudio de la API de Grafana
20/04/2020	9:00	13:00	4:00 Creación de dashboard de ejemplo
21/04/2020	10:00	13:30	3:30 Creación de dashboard de ejemplo
23/04/2020	9:00	13:30	4:30 Mostrado de gráficos tras pulsación en estació
25/04/2020	10:30	13:00	2:30 Mostrado de gráficos tras pulsación en estació
25/04/2020	16:00	20:00	4:00 Mostrado de gráficos tras pulsación en estació
26/04/2020	9:00	13:30	4:30 Mostrado de gráficos tras pulsación en estació
27/04/2020	16:30	20:00	3:30 Gestión de nanosensores registrados por el us
28/04	10:00	13:30	3:30 Gestión de nanosensores registrados por el us
29/04/2020	10:00	11:30	1:30 Generación de PDF corrección y bug quiz
30/04/2020	10:30	13:00	2:30 Generación de PDF corrección
30/04/2020	17:00	18:30	1:30 Generación de PDF corrección
02/05/2020	10:00	13:00	3:00 Estudio integración servidores de correo con I
02/05/2020	16:30	19:30	3:00 Configuración de mail en Django
03/05/2020	10:00	11:30	1:30 Configuración de mail en Django
03/05/2020	11:30	13:00	1:30 Diseño web de gestión de olvido de contraseñ
03/05/2020	17:00	20:00	3:00 Diseño web de gestión de olvido de contraseñ
04/05/2020	10:00	13:30	3:30 Implementación de la recuperación de contra
05/05/2020	16:30	20:30	4:00 Implementación de la recuperación de contra
08/05/2020	10:00	13:00	3:00 Correcciones estéticas a la página de cuadros
09/05/2020	16:30	17:30	1:00 Correcciones estéticas a la página de cuadros
09/05/2020	17:30	19:30	2:00 Corrección de bugs
10/05/2020	9:30	13:30	4:00 Gestión de cursos académicos
10/05/2020	17:00	19:30	2:30 Estudio de la implementación de multilinguaj
11/05/2020	10:00	13:30	3:30 Gestión de cursos académicos
11/05/2020	17:00	20:00	3:00 Gestión de cursos académicos
12/05/2020	10:00	11:30	1:30 Gestión de cursos académicos
12/05/2020	12:00	13:30	1:30 Configuración de Django para soporte a multil
12/05/2020	16:30	19:30	3:00 Etiquetado de cadenas de texto para multilen
13/05/2020	10:00	11:00	1:00 Etiquetado de cadenas de texto para multilen
13/05/2020	11:00	13:30	2:30 Generación de ficheros de traducción a inglés
13/05/2020	16:30	19:30	3:00 Traducción a inglés
14/05/2020	10:00	13:30	3:30 Generación de ficheros de traducción a francé
19/05/2020	10:00	13:00	3:00 Estudio de posibilidades de ficheros de grupos
19/05/2020	16:30	19:30	3:00 Carga masiva de grupos desde fichero
20/05/2020	10:00	13:30	3:30 Carga masiva de grupos desde fichero
20/05/2020	16:30	18:00	1:30 Carga masiva de grupos desde fichero
29/05/2020	11:00	12:30	1:30 Estados de alumnos
29/05/2020	16:30	20:00	3:30 Modelo y estado de nanosensores
30/05/2020	10:00	11:30	1:30 Modelo y estado de nanosensores
30/05/2020	16:30	18:00	1:30 Detección de bug carga masiva (Caracteres es

Figura 38: Distribución de esfuerzos (2)

31/05/2020	10:00	13:00	3:00 Migración de base de datos a PostgreSQL
31/05/2020	16:30	20:30	4:00 Dockerización de Stics
01/06/2020	10:00	11:30	1:30 Dockerización de Stics
01/06/2020	11:30	13:00	1:30 Despliegue en Google Cloud
01/06/2020	16:30	20:00	3:30 Despliegue en Google Cloud
02/06/2020	10:00	12:00	2:00 Despliegue en Google Cloud
02/06/2020	16:00	17:00	1:00 Estudio de licencias de software
02/06/2020	17:00	20:30	3:30 Página de menciones legales - Licencia y políti
03/06/2020	16:30	18:30	2:00 Página de ventas de Stics
04/06/2020	10:00	13:30	3:30 Mejoras estéticas
04/06/2020	16:00	18:00	2:00 Videotutoriales - Aprendizaje de Stics
05/06/2020	10:00	12:00	2:00 Videotutoriales - Aprendizaje de Stics
06/06/2020	10:00	13:30	3:30 Bugs y adición de posibilidad de modificación
07/06/2020	10:00	13:00	3:00 Redacción de documentación
08/06/2020	16:30	20:00	3:30 Redacción de documentación
09/06/2020	9:30	12:30	3:00 Redacción de documentación
10/06/2020	10:30	13:30	3:00 Creación masiva de cuentas por la administrac
10/06/2020	16:30	19:00	2:30 Modificación del quiz por el profesorado
11/06/2020	10:00	12:00	2:00 Modificación del quiz por el profesorado
11/06/2020	16:30	20:00	3:30 Modificaciones estéticas de disposición de ele
12/06/2020	10:00	12:00	2:00 Realización y carga de nuevas traducciones inj
12/06/2020	12:00	13:30	1:30 Redacción de documentación
13/06/2020	10:00	13:30	3:30 Redacción de documentación
14/06/2020	10:30	13:00	2:30 Redacción de documentación
14/06/2020	17:00	20:00	3:00 Redacción de documentación
15/06/2020	10:00	13:30	3:30 Redacción de documentación
15/06/2020	16:30	19:30	3:00 Redacción de documentación
16/06/2020	9:30	13:30	4:00 Redacción de documentación
16/06/2020	17:00	19:30	2:30 Redacción de documentación
17/06/2020	9:00	13:30	4:30 Redacción de documentación
17/06/2020	17:00	19:30	2:30 Redacción de documentación
18/06/2020	9:15	14:00	4:45 Redacción de documentación
18/06/2020	16:30	19:30	3:00 Redacción de documentación
19/06/2020	9:15	13:30	4:15 Redacción de documentación
19/06/2020	16:30	19:30	3:00 Redacción de documentación

Figura 39: Distribución de esfuerzos (3)

## E.2. Presupuesto

Una parte clave si se desea que un proyecto tenga éxito y pueda perdurar en el tiempo es el establecimiento de un plan adecuado de viabilidad, márketing, etc. No obstante, no hay que descuidar otro aspecto que tiene un gran impacto en el cliente a la hora de escoger o no el producto. Este es, el presupuesto. Si este es demasiado elevado, el cliente no escogerá el producto y buscará otra solución en la competencia. Por el contrario, si es demasiado bajo, el cliente lo adquirirá pero su valor no logrará satisfacer los gastos de explotación del producto y no generará beneficio.

En el caso de este proyecto, se ha comentado con anterioridad que el objetivo es ofrecerlo en dos variantes:

- Despliegue en el centro educativo utilizando la infraestructura preexistente.
- Utilizar la solución SaaS desplegada en cloud.

En primer lugar, el coste de mano de obra por hora se ha establecido en 20€. En el apartado anterior, se ha estimado el número de horas totales en 385h. Por tanto,  $20\text{€} \times 385\text{h} = 7700\text{€}$  es el coste asociado al desarrollo de la plataforma. A esta cifra, habrá que sumar los costes derivados de la puesta en marcha de la instalación y de la explotación del software en aquellos casos en los que aplique. Asimismo, como se trata de dos modelos de negocio diferentes con características particulares, a continuación se pasa a detallar cada uno de ellos:

## Despliegue en el centro

Para el cálculo del coste imputable al cliente del desarrollo de la plataforma se ha previsto el despliegue en 15 instituciones educativas. Asimismo, para el coste derivado de la instalación en el centro se han previsto 15h de mano de obra a 20€ la hora.

Desglose de costes:

- Desarrollo de la plataforma: 500€.
- Instalación en el centro: 300€.
- **Total: 800€.**
- Margen de beneficio (20%): 160€.
- **Total antes de impuestos: 960€.**
- Impuestos (21%): 201.60€.
- **Total después de impuestos: 1161,60€.**

## Despliegue en cloud

En el caso del despliegue en cloud, el coste imputable al desarrollo de la plataforma se ha calculado tomando un tiempo de uso del sistema estimado de 3 años. Para el coste de la infraestructura en cloud se ha tomado como referencia el coste de una máquina en Google Cloud con 3.75Gb de memoria con lo que el coste se establecería en 25 dólares mensuales, a dividir entre 5 centros educativos.

Desglose de coste mensual:

- Desarrollo de la plataforma: 15.67€.
- Infraestructura cloud: 5€.
- **Total: 20,67€.**
- Margen de beneficio (20%): 4,13€.

- **Total antes de impuestos: 24,80 €.**
- Impuestos (21 %): 5,21 €.
- **Total después de impuestos: 30 €.**

## F. Guía de instalación y despliegue

El desarrollo de una buena guía de instalación y despliegue es básico para facilitar el trabajo de implantación de la herramienta en un nuevo entorno, por ejemplo, en un nuevo centro educativo. Definir unos pasos claros y concisos ahorra tiempo y dinero en el proceso. A continuación se muestran, en primer lugar, los requisitos del sistema, además de una explicación de los pasos a seguir en caso de que únicamente se deseen realizar algunas pruebas del software o su concepto. Posteriormente, se presentan los pasos a seguir en caso de que se desee desplegar el software en un servidor real.

### F.1. Requisitos de instalación

Para el despliegue en cloud se tomará como referencia un sistema Debian. Dicha máquina deberá disponer del siguiente software previamente instalado y actualizado. De no ser así, existen guías que describen de forma somera la instalación de dichas herramientas.

- **Python3.** Python es el lenguaje de programación en el que está basado el *framework* Django. Se recomienda el uso de la versión 3, o superior, de este software.
- **Pip3.** Pip es un sistema de gestión de paquetes utilizado para instalar y administrar paquetes de software escritos en Python. Se utilizará la versión 3, o superior, para su compatibilidad con Python3.
- **El SGBD PostgreSQL.** Se deberá contar con al menos un usuario con permisos totales sobre una base de datos.
- **Nginx.** Nginx es un servidor web ligero de alto rendimiento. Es software libre y de código abierto, licenciado bajo la Licencia BSD simplificada.
- **GeoServer.** Su instalación es necesaria ya que es el software escogido como servidor de Sistemas de Información Geográficos.
- **Grafana.** De nuevo, es necesario instalarlo ya que es el software utilizado para la creación de los *dashboards*.

#### Realización de pruebas sobre el software

La realización de pruebas del software, su concepto, etc. es posible accediendo a la dirección siguiente: <http://34.90.15.171:8000/>. Del mismo modo, el código fuente es accesible en el siguiente repositorio de GitLab: <https://gitlab.com/Cildo98/stics>.

### F.2. Pasos a seguir

1. Descarga de los ficheros del proyecto del repositorio de GitLab.
2. A continuación, acceder al directorio y ejecutar el comando `sudo pip3 install -r requirements.txt`.



3. Se deberá instalar también el software uwsgi, que actuará como mediador entre Django y el servidor web nginx. Para ello, se ejecutará el comando `sudo pip3 install uwsgi`.
4. Asimismo, se deberá añadir un fichero de configuración de nginx. Se tomará como base el fichero que se muestra en el anexo F.
5. Con el fin de soportar conexiones mediante el protocolo HTTPS será necesaria la obtención de un certificado SSL. Se recomienda Let's Encrypt como CA (*Certificate Authority*) debido a la ausencia de costes. Para automatizar el proceso, se usará Certbot, un paquete que simplifica el proceso. Para su instalación, se ejecutará `sudo apt-get install sudo apt-get install certbot python-certbot-nginx`. Tras esto, se procederá a lanzar el asistente de configuración con el comando `sudo certbot --nginx`, tras lo cual se deberán seguir las instrucciones mostradas por pantalla.
6. A continuación, se deberá modificar la configuración del fichero `settings.py` para adaptarlo al nuevo despliegue. En concreto, se recomienda la comprobación de los siguientes parámetros.
  - `ALLOWED_HOSTS` deberá contener el dominio al que apunte el servidor, o, en su defecto, la dirección IP del servidor.
  - `EMAIL_HOST`, `EMAIL_HOST_USER`, `EMAIL_HOST_PASSWORD`, `EMAIL_PORT` y `DEFAULT_FROM_EMAIL` deberán contener los parámetros adecuados del proveedor de servicio de correo electrónico.
  - Los parámetros `NAME`, `USER` y `PASSWORD` de la sección `DATABASES` deberán corresponderse con los datos de la base de datos PostgreSQL disponible.
7. Asimismo, se deberán modificar las URLs de GeoServer y Grafana en el código HTML para que se correspondan con la IP del servidor en el que se está desplegando el sistema.
8. Una vez Django esté configurado correctamente, se deberán ejecutar los comandos que se indican a continuación para crear las entidades de la base de datos.
  - `python3 manage.py makemigrations`
  - `python3 manage.py migrate`
9. Para iniciar el sistema se deberá utilizar el comando `uwsgi --ini stics.ini` desde el directorio de la aplicación. Esto hará el sistema disponible desde Internet. Se recomienda el uso de la herramienta `screen` para mantener el sistema en ejecución.

### F.3. Fichero de configuración para Nginx

```
# stics_nginx.conf

upstream django {
    server unix:///ruta/stics/stics.sock;
}

server {
    listen 4000;
    server_name 10.164.0.4;

    location / {

        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Forwarded-Server $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Authorization 'Bearer eyJrIjoiTTB3RVpMNUNzWW9sdjIqNnZlQzV4aVFLVDVycjI1TVYiLCJuIjoiY3R3Y3MiLCJpZCI6MX0=';

        add_header 'Access-Control-Allow-Origin' * always;
        add_header 'Access-Control-Allow-Credentials' 'true' always;
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS' always;
        add_header 'Access-Control-Expose-Headers' 'Content-Type,Content-Length,Content-Range';
        add_header 'Access-Control-Allow-Headers' 'Accept, Authorization, Cache-Control, Content-Type, DNT, If-Modified-Since, Keep-Alive, Origin, User-Agent, X-Requested-With' always;

        if ($request_method = 'OPTIONS') {
            return 204;
        }
    }
}
```

```
# stics_nginx.conf

    proxy_pass http://10.164.0.4:3000/;
}
}

server {
    server_name dominio.es;
    charset utf-8;

    location /media {
        alias /ruta/sticsapp/media;
    }

    location /static {
        alias /ruta/sticsapp/static;
    }

    location / {
        uwsgi_pass django;
        include /ruta/uwsgi_params;
    }
}
```

En el fichero anterior será necesaria la modificación de `server_name` y `proxy_pass` para que contengan la dirección IP del servidor en el que se está desplegando el sistema. Asimismo, se deberá modificar `Authorization` con la API *key* obtenida con la interfaz gráfica de Grafana.

## G. Amenazas, vulnerabilidades y salvaguardas identificadas en el plan de análisis y gestión de riesgos

Esta sección forma parte del plan de análisis y gestión de riesgos que se ha presentado en secciones anteriores. Aquí se recogen las principales amenazas detectadas aplicadas a la aplicación web que se ha desarrollado, además de las vulnerabilidades a las que está expuesta y las principales salvaguardas que se han planteado.

### Amenazas

#### N.1/I.1 - Fuego en la sala de servidores

- **Activos afectados:** Sv1 (servidores), Rd1 (infraestructura de red).
- **Descripción:** Posibilidad de que el fuego destruya los recursos hardware necesarios para proporcionar el servicio de aprendizaje.
- **Posible origen:** Accidental/humano.

#### N.2 - Inundación en la sala de servidores

- **Activos afectados:** Sv1 (servidores), Rd1 (infraestructura de red).
- **Descripción:** Posibilidad de que el agua destruya los recursos hardware necesarios para proporcionar el servicio de aprendizaje.
- **Posible origen:** Accidental/humano.

#### I.5 - Avería de origen físico/lógico en el servidor donde se aloja el sistema

- **Activos afectados:** Sv1 (servidores), Rd1 (infraestructura de red).
- **Descripción:** Fallos en equipos y/o programas debidos a un defecto de origen, causas sobrevenidas durante el funcionamiento y/o fallos de configuración del software instalado en dicho servidor.
- **Posible origen:** Accidental/humano.

#### E.1 - Errores de los usuarios en el uso de software

- **Activos afectados:** Dt1.
- **Descripción:** Posibilidad de que el usuario cometa un error durante el uso del software que produzca un error grave en el servidor que afecte a la disponibilidad del mismo.
- **Posible origen:** Humano.

#### E.21 - Errores de la administración en el mantenimiento / actualización de software

- **Activos afectados:** Dt1 y Sv1.
- **Descripción:** Problemas de interoperabilidad entre distinto software que haga que el sistema no esté disponible para su acceso desde el exterior.
- **Posible origen:** Humano.

## Vulnerabilidades y salvaguardas

En el desarrollo de la operativa normal de la organización, existe un gran número de vulnerabilidades a las que esta está expuesta. Por ejemplo, un servidor en un centro educativo cuyo software básico de mantenimiento no está actualizado (antivirus, firewall, versión de SO, etc), servidor sin mantenimiento por parte del fabricante o un CPD sin protección de control de acceso a las instalaciones que puede conducir a intrusiones en el sistema o fugas de información.

Asimismo, frente a dichas vulnerabilidades los centros educativos deberían haber implementado algunas medidas como, por ejemplo, impedimento de acceso a la sala de servidores por estar cerrada con llave, aislamiento reforzado con el exterior para evitar problemas derivados de fenómenos como el fuego o el agua planteados en la sección anterior, etc.

## H. Tecnología utilizada

Además de la elección de los componentes principales del proyecto, es decir, *framework* de desarrollo web, SGBD, etc., resulta a menudo interesante tomar en consideración algunos de los *plugins* que estas herramientas ofrecen ya que pueden proporcionar valor añadido que mejore el resultado final. A continuación se describen las principales elecciones realizadas en el marco del proyecto.

### PostgreSQL - Unaccent

Según el sitio web de PostgreSQL, *unaccent* [54] es un diccionario de búsqueda de texto que elimina signos diacríticos de los lexemas. Es un diccionario filtrante, lo cual quiere decir que su salida se pasa siempre al siguiente diccionario, si lo hay. Este ha resultado muy útil para la implementación de la gestión masiva de usuarios utilizando para ello ficheros de datos, tanto para crear nuevas cuentas como para la asignación de alumnado a grupos. Si no, se habría tenido que implementar esta funcionalidad de algún modo ya que es requisito indispensable en un idioma que cuenta con tantos signos diacríticos.

### Leaflet - MarkerCluster

Una vez se decidió el uso de Leaflet como librería JavaScript, se realizó una pequeña tarea de búsqueda para conocer qué *plugins* existían y si alguno de ellos podía ser interesante para su integración al proyecto. Esto ha sido el caso de MarkerCluster, que permite la agrupación de datos para que la presentación de estos sea más cómoda, mejorando asimismo el rendimiento y los tiempos de carga.

### WFS

Según el Open Geospatial Consortium, WFS [55], o *Web Feature Service*, representa un cambio en el modo en que la información geográfica es creada, modificada y compartida. Este protocolo ofrece acceso directo con gran granularidad a la información geográfica a nivel de elemento y de propiedad de dicho elemento. En el proyecto, WFS es utilizado mediante Ajax para realizar una petición a GeoServer con el fin de recuperar en formato GeoJSON los datos de sensores ambientales seleccionados.

### Multilenguaje en Django

Django ofrece soporte a multiidioma sin necesidad de instalar *plugins* adicionales. Se permite la traducción de todos los elementos de una aplicación web, ya sea en el código HTML, en JavaScript, o de mensajes de respuesta a peticiones en el propio código Python del backend.

Cada vez que se desea añadir un nuevo idioma, con un comando se generan los ficheros necesarios en el directorio concreto de cada idioma. Este fichero es el que se transfiere al servicio de traducción. Asimismo, permite almacenar información sobre el último traductor, etc. para poder realizar el seguimiento.

## PostgreSQL - TimescaleDB

Por último, para el almacenamiento y gestión de los datos medioambientales se ha utilizado TimescaleDB [56], una extensión con versión OpenSource de PostgreSQL para el análisis de datos de series temporales con la potencia de SQL. Posee una gran capacidad de escalar y buen rendimiento.

## Git - Software de control de versiones

Git es un sistema de control de versiones distribuido gratuito y *OpenSource* diseñado para gestionar proyectos, ya sean pequeños o grandes con velocidad y eficiencia. Cuenta con características de apoyo al desarrollo de software en equipos como es la posibilidad de gestión de ramas, permitiendo que cada programador se centre en un aspecto concreto, contribuyendo luego al proyecto global.

Construídas sobre él, existen multitud de soluciones de gestión de repositorios de Git, aunque los más conocidos son GitHub y GitLab. En este proyecto se ha decidido el uso de GitLab ya que oferta alguna característica más que GitHub en su versión gratuita, como, por ejemplo, la posibilidad de gestionar repositorios privados.

No obstante, fruto de la naturaleza OpenSource del proyecto, el repositorio de este es público y accesible en la siguiente URL: <https://gitlab.com/Cildo98/stics>.

# Referencias

- [1] Fridays For Future. *Fridays For Future - Movement*. URL: <https://fridaysforfuture.org/>. (accedido el: 18.06.2020).
- [2] Wikipedia the free encyclopedia. *2019 United Nations Climate Change Conference*. URL: [https://en.wikipedia.org/wiki/2019\\_United\\_Nations\\_Climate\\_Change\\_Conference#:~:text=Website,presidency%5C%20of%5C%20the%5C%20Chilean%5C%20government..](https://en.wikipedia.org/wiki/2019_United_Nations_Climate_Change_Conference#:~:text=Website,presidency%5C%20of%5C%20the%5C%20Chilean%5C%20government..) (accedido el: 18.06.2020).
- [3] Wikipedia the free encyclopedia. *Lluvia de ideas*. URL: [https://es.wikipedia.org/wiki/Lluvia\\_de\\_ideas](https://es.wikipedia.org/wiki/Lluvia_de_ideas). (accedido el: 18.06.2020).
- [4] Zooniverse. *People-powered research*. URL: <https://www.zooniverse.org/>. (accedido el: 19.06.2020).
- [5] Project Copernicus. *Europe's eyes on Earth*. URL: <https://www.copernicus.eu/es>. (accedido el: 19.06.2020).
- [6] W3C. *Architecture of the World Wide Web, Volume One*. URL: <https://www.w3.org/TR/webarch/>. (accedido el: 18.06.2020).
- [7] Express. *Express. Infraestructura web rápida, minimalista y flexible para Node.js*. URL: <https://expressjs.com/es/>. (accedido el: 16.06.2020).
- [8] Django Software Foundation. *Django. The web framework for perfectionists with deadlines*. URL: <https://www.djangoproject.com/>. (accedido el: 16.06.2020).
- [9] Ruby on Rails. *Imagine what you could build if you learned Ruby on Rails...* URL: <https://rubyonrails.org/>. (accedido el: 16.06.2020).
- [10] Spring. *Spring makes Java simple, productive, modern, reactive*. URL: <https://spring.io/>. (accedido el: 16.06.2020).
- [11] PostgreSQL. *PostgreSQL: The World's Most Advanced Open Source Relational Database*. URL: <https://www.postgresql.org/>. (accedido el: 16.06.2020).
- [12] MySQL. *MySQL. The world's most popular open source database*. URL: <https://www.mysql.com/>. (accedido el: 16.06.2020).
- [13] MariaDB Foundation. *MariaDB Server: The open source relational database*. URL: <https://mariadb.org/>. (accedido el: 16.06.2020).
- [14] Oracle. *Oracle Database*. URL: <https://www.oracle.com/es/index.html>. (accedido el: 16.06.2020).
- [15] SQLite. *SQLite. Small. Fast. Reliable. Choose any three*. URL: <https://www.sqlite.org/index.html>. (accedido el: 18.06.2020).
- [16] Grafana Labs. *Grafana: The open observability platform*. URL: <https://grafana.com/>. (accedido el: 16.06.2020).
- [17] Graphite. *Graphite. Make it easy to store and graph metrics*. URL: <https://graphiteapp.org/>. (accedido el: 16.06.2020).
- [18] Chart.js. *Simple yet flexible JavaScript charting for designers & developers*. URL: <https://www.chartjs.org/>. (accedido el: 16.06.2020).
- [19] Netdata. *Monitor everything in real time – for free*. URL: <https://www.netdata.cloud/>. (accedido el: 16.06.2020).



- [20] Alan L. Melnick. *Introduction to Geographic Information Systems in Public Health*. Gaithersburg, Maryland: Aspen Publishers, Inc., 2002.
- [21] gvSIG asociación. *gvSIG. La mayor red de profesionales en geomática libre*. URL: <http://www.gvsig.com/>. (accedido el: 16.06.2020).
- [22] ArcGIS. *ArcGIS: Incorpore la representación cartográfica y análisis líderes del sector a su infraestructura y en la nube*. URL: <https://enterprise.arcgis.com/es/>. (accedido el: 16.06.2020).
- [23] GeoServer. *An open source server for sharing geospatial data*. URL: <http://geoserver.org/>. (accedido el: 16.06.2020).
- [24] OGC. *The Home of Location Technology Innovation and Collaboration*. URL: <https://www.ogc.org/>. (accedido el: 18.06.2020).
- [25] OpenLayers. *A high-performance, feature-packed library for all your mapping needs*. URL: <https://openlayers.org/>. (accedido el: 16.06.2020).
- [26] Leaflet. *An open-source JavaScript library for mobile-friendly interactive maps*. URL: <https://leafletjs.com/>. (accedido el: 16.06.2020).
- [27] Comisión Europea. *Protección de datos*. URL: [https://ec.europa.eu/info/law/law-topic/data-protection\\_es](https://ec.europa.eu/info/law/law-topic/data-protection_es). (accedido el: 18.06.2020).
- [28] Google. *Proteger sitios web con el protocolo HTTPS*. URL: <https://support.google.com/webmasters/answer/6073543?hl=es>. (accedido el: 18.06.2020).
- [29] Wikipedia the free encyclopedia. *Software como servicio*. URL: [https://es.wikipedia.org/wiki/Software\\_como\\_servicio](https://es.wikipedia.org/wiki/Software_como_servicio). (accedido el: 18.06.2020).
- [30] AECC. *Glosario de Términos de Calidad e Ingeniería de Software*. AECC, 1986.
- [31] Erich Gamma y otros. *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.
- [32] PostGIS. *Spatial and Geographic Objects for PostgreSQL*. URL: <https://postgis.net/>. (accedido el: 16.06.2020).
- [33] Balsamiq. *Balsamiq Cloud. Fast, Approachable, Collaborative Wireframing*. URL: <https://balsamiq.cloud/>. (accedido el: 18.06.2020).
- [34] Wikipedia the free encyclopedia. *Centro de procesamiento de datos*. URL: [https://es.wikipedia.org/wiki/Centro\\_de\\_procesamiento\\_de\\_datos#:~:text=Se%5C%20denomina%5C%20centro%5C%20de%5C%20procesamiento,la%5C%20informaci%5C%C3%5C%B3n%5C%20de%5C%20una%5C%20organizaci%5C%C3%5C%B3n..](https://es.wikipedia.org/wiki/Centro_de_procesamiento_de_datos#:~:text=Se%5C%20denomina%5C%20centro%5C%20de%5C%20procesamiento,la%5C%20informaci%5C%C3%5C%B3n%5C%20de%5C%20una%5C%20organizaci%5C%C3%5C%B3n..) (accedido el: 18.06.2020).
- [35] INCIBE. *Instituto Nacional de Ciberseguridad*. URL: <https://www.incibe.es/>. (accedido el: 18.06.2020).
- [36] Incibe. *Plan Director de Seguridad*. URL: <https://www.incibe.es/protege-tu-empresa/que-te-interesa/plan-director-seguridad>. (accedido el: 16.06.2020).
- [37] Incibe. *Análisis y gestión de riesgos*. URL: <https://www.incibe.es/protege-tu-empresa/blog/analisis-riesgos-pasos-sencillo>. (accedido el: 16.06.2020).
- [38] Jefatura del Estado. *Ley 31/1995, de 8 de noviembre, de prevención de Riesgos Laborales*. URL: <https://www.boe.es/eli/es/l/1995/11/08/31>. (accedido el: 18.06.2020).

- [39] Ministerio de la Presidencia. *Real Decreto 1215/1997, de 18 de julio, por el que se establecen las disposiciones mínimas de seguridad y salud para la utilización por los trabajadores de los equipos de trabajo*. URL: <https://www.boe.es/buscar/doc.php?id=BOE-A-1997-17824>. (accedido el: 18.06.2020).
- [40] Ministerio de Trabajo y Asuntos Sociales. *Real Decreto 486/1997, de 14 de abril, por el que se establecen las disposiciones mínimas de seguridad y salud en los lugares de trabajo*. URL: <https://www.boe.es/buscar/act.php?id=BOE-A-1997-8669>. (accedido el: 18.06.2020).
- [41] Dirección General de Industria y de la Pequeña y Mediana Empresa. *Análisis DAFO*. URL: <https://dafo.ipyme.org/Home>. (accedido el: 18.06.2020).
- [42] Naciones Unidas. *Objetivos y metas de desarrollo sostenible*. URL: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>. (accedido el: 23.06.2020).
- [43] Gobierno de España. *Reutiliza la información pública*. URL: [https://datos.gob.es/es/iniciativas?f%5C%5B0%5C%5D=field\\_initiative\\_admin\\_level%5C%3A1962](https://datos.gob.es/es/iniciativas?f%5C%5B0%5C%5D=field_initiative_admin_level%5C%3A1962). (accedido el: 16.06.2020).
- [44] Google. *Google Dataset Search*. URL: <https://datasetsearch.research.google.com/>. (accedido el: 16.06.2020).
- [45] Facultad de Ciencias de la Educación. Universidad de Málaga. *Preguntas frecuentes sobre el Trabajo de Fin de Grado (TFG)*. URL: [https://www.uma.es/media/files/Preguntas\\_frecuentes\\_TFG.pdf](https://www.uma.es/media/files/Preguntas_frecuentes_TFG.pdf). (accedido el: 18.06.2020).
- [46] InfluxDB. *Real-time visibility into stacks, sensors and systems*. URL: <https://www.influxdata.com/>. (accedido el: 23.06.2020).
- [47] OpenTSDB. *The Scalable Time Series Database*. URL: <http://opentsdb.net/>. (accedido el: 23.06.2020).
- [48] Elastic. *Kibana. Tu ventana al Elastic Stack*. URL: <https://www.elastic.co/es/kibana>. (accedido el: 23.06.2020).
- [49] Docker. *Empowering App Development for Developers*. URL: <https://www.docker.com/>. (accedido el: 23.06.2020).
- [50] Go. *The Go Programming Language*. URL: <https://golang.org/>. (accedido el: 23.06.2020).
- [51] Wikipedia the free encyclopedia. *Comparison of geographic information systems software*. URL: [https://en.wikipedia.org/wiki/Comparison\\_of\\_geographic\\_information\\_systems\\_software](https://en.wikipedia.org/wiki/Comparison_of_geographic_information_systems_software). (accedido el: 16.06.2020).
- [52] Jennifer R. Bauer. *Assessing the Robustness of Web Feature Services Necessary to Satisfy the Requirements of Coastal Management Applications*. Oregon State University, Oregon: The College of Earth, Ocean, y Atmospheric Sciences, Oregon State University, 2012.
- [53] CEF Digital. Connecting Europe. *CEF. Give your digital project a boost*. URL: <https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/CEF+Digital+Home>. (accedido el: 16.06.2020).
- [54] PostgreSQL. *Unaccent*. URL: <https://www.postgresql.org/docs/9.1/unaccent.html>. (accedido el: 16.06.2020).

- [55] Wikipedia the free encyclopedia. *Web Feature Service*. URL: [https://es.wikipedia.org/wiki/Web\\_Feature\\_Service](https://es.wikipedia.org/wiki/Web_Feature_Service). (accedido el: 16.06.2020).
- [56] Timescale. *All of your time-series data, instantly accessible*. URL: <https://www.timescale.com/>. (accedido el: 16.06.2020).