



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA DEL SOFTWARE**

**HeatMap3D: Mapas de calor 3D para la protección de áreas
de control críticas**

**HeatMap3D: 3D Heat maps for protection of critical control
areas**

Realizado por
Eduardo Guidet Jiménez

Tutorizado por
María Cristina Alcaraz Tello
Francisco Javier López Muñoz

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JUNIO DE 2019

Fecha defensa: Julio de 2019

Fdo. El/la Secretario/a del Tribunal

Resumen

Desde el siglo XVIII, han tenido lugar tres generaciones industriales. En cada una de estas etapas, las técnicas de producción y gestión de la energía han ido progresando para dar servicio y respuesta a las problemáticas de cada época. Expertos de diferentes sectores coinciden en que actualmente estamos entrando en la cuarta revolución industrial. Este fenómeno se caracterizará por transformar el concepto de "producción", digitalizando y automatizando de cada una de las tareas implicadas en los procesos de producción, así como el control de estos procesos [1,2] y la gestión eficiente de la propia energía, propios de los actuales sistemas de "Smart Grids" o redes eléctricas inteligentes. En este ámbito, se hará uso de tecnologías innovadoras como el Internet de las cosas (IoT), inteligencia artificial, Cloud/Fog/Mobile Edge computing, Big Data, nanotecnología, virtualización y sistemas de realidad virtual. El proyecto que se ha desarrollado surge para dar respuesta al último campo, la realidad virtual, modelando un entorno en tres dimensiones en el cual se ha modelado un escenario de Smart Grid en un entorno simulado. La abstracción que se ha elaborado permite la visualización de anomalías y la monitorización en tiempo real de un sistema de control de energía de una forma intuitiva y realista, mejorando, pero no sustituyendo, los tradicionales sistemas de computación y visualización basados en HMI (Human-Machine-Interfaces), utilizados hoy en día para el control local o remoto de subestaciones de energía.

Palabras clave: HeatMap3D, red eléctrica inteligente, Unity, realidad virtual, realidad aumentada, simulación, anomalías, sistema de control, seguridad.

Abstract

Since the eighteenth century, three industrial generations have taken place. In each of these stages, the production techniques and energy management have been progressing in order to provide service and response to the problems of each era. Experts from different sectors agree that we are currently entering on the fourth industrial revolution. This phenomenon will be characterized by transform the concept of "production", digitalizing and automating each of the tasks involved on the production processes, as well as the control of this processes and an efficient energy management, typical of the current systems of "Smart Grids" or intelligent electrical networks. In this area, innovative technologies will be used, such as the Internet of Things (IoT), artificial intelligence, Cloud/Fog/Mobile Edge computing, Big Data, nanotechnology, virtualization and virtual reality systems. The project that has been developed arises to respond to the last field, virtual reality, modeling a three-dimensional environment where a Smart Grid scenario has been modeled in a simulated environment. The abstraction that has been elaborated allows the visualization of anomalies as well as real-time monitoring of a real energy control system in an intuitive and realistic way, improving, but not substituting, the traditional HMI-based (Human-Machine-Interfaces) computing and visualization systems, used today for local or remote control of power substations.

Keywords: HeatMap3D, Smart Grids, Unity, virtual reality, augmented reality, simulation, anomalies, control system, security.

Índice

1. Introducción	1
1.1 Motivación	2
1.2 Contexto	3
1.3 Objetivos principales	4
1.4 Objetivos a medio-largo plazo	5
1.5 Estructura de la memoria	7
2. Avances Tecnológicos de la Industria	11
2.1 Antecedentes	12
2.2 La industria actual	15
2.3 Impacto en la economía	17
3. Smart Grids	19
3.1 Contexto	20
3.2 Tecnologías utilizadas en los Smart Grids	22
4. Análisis y diseño de HeatMap3D	29
4.1 Estimación temporal	30
4.2 Metodología de desarrollo	32
4.3 Análisis del proyecto	34
4.3.1 Requisitos funcionales	34
4.3.2 Requisitos no funcionales	35
4.3.3 Casos de uso	36
4.3.4 Diagrama de clases	55
4.3.5 Diagramas de interacción	57
4.3.6 Diagrama de estados	59

5. Herramientas y técnicas utilizadas	61
5.1 El motor: Unity	62
5.1.1 Contexto	62
5.1.2 La interfaz	63
5.1.3 Assets	66
5.2 El código: Visual Studio Code	67
6. Desarrollo del proyecto	69
6.1 Primeros pasos	70
6.2 Modelado de dispositivos	70
6.3 Modelado del entorno	73
6.4 Modelado del comportamiento	75
6.5 Dificultades durante el desarrollo	77
7. Conclusiones y futuro	81
7.1 Conclusión	82
7.2 Valoración personal y futuro	82
8. Bibliografía y referencias	85
8.1 Bibliografía y referencias	86
8.2 Librerías de código abierto utilizadas	89

1. Introducción

1.1 Motivación

En la industria actual encontramos numerosas formas de presentar información y en ocasiones se requiere un largo periodo de aprendizaje para poder interpretarla. Aunque se han logrado avances significativos en el área de control local y remoto y en la usabilidad de interfaces de usuario y en el campo de la interacción *human-machine*, adquirir los conocimientos necesarios para el manejo de ciertas tecnologías puede resultar monótono y poco intuitivo.

Numerosas empresas han invertido en multitud de recursos con el objeto de “modernizar” este tipo de procesos, desde sencillos cursos sobre políticas de empresa (Game-Learn, 2015) hasta complejas interfaces de sistemas de control (Dede, Salzman & Loftin, 1996). Este tipo de iniciativas permiten a los usuarios una interacción más **intuitiva** con el sistema, alcanzando un nivel más profundo de asimilación del contenido en menor tiempo y con **menor desgaste** mental (Lee, 2012).

Esta necesidad nos ha empujado hoy en día a encontrar multitud de tecnologías de última revolución, como es el caso de la **realidad virtual** o la **realidad aumentada**. Estas tecnologías permiten al usuario experimentar una sensación de inmersión en un entorno controlado, en el cual no nos encontramos limitados por el mundo físico y podemos obtener información de formas creativas e intuitivas.

HeatMap3D aparece como un ejemplo concreto de la aplicación de estas tecnologías novedosas al campo de la gestión de la **industria energética**. A través de la realidad virtual los usuarios podrán situarse dentro de una simulación de un caso de uso concreto, con una representación realista e intuitiva de los diferentes elementos del sistema, permitiendo, además, establecer una conexión directa entre

elementos del mundo virtual y elementos del mundo real. En esta nueva forma de concebir la realidad, la realidad aumentada añade los mecanismos necesarios para interactuar con ciertos elementos del mundo, no solo aquellos relacionados con los sistemas de control sino con aquellos pertenecientes al mundo físico (ej. humedad, nivel de voltaje, temperatura, presión, etc.), ilustrando dicha información de una forma más ilustrativa, atractiva y mucho más clara para el usuario final.

1.2 Contexto

El presente proyecto presenta la caracterización de un modelo virtual en tres dimensiones (comúnmente conocido como 3D), en el cual se pretende simular e ilustrar, de forma clara, intuitiva y atractiva, un caso concreto del mundo real. En el caso particular de este TFG, este modelo está basado en la simulación de una red eléctrica inteligente, conocido hoy en día como: sistemas de **Smart Grid** (Fang, Misra, Xue, & Yang, 2011).

Este sistema se compone de una serie de dispositivos tecnológicos que se definirán posteriormente, entre los que destacan diversos componentes que simulan ciertas condiciones físicas establecidas por la propia infraestructura crítica en cuestión (ej. temperatura, nivel de presión, caudal, etc.), cuyos valores son esenciales para retroalimentar al sistema de control, y de esta forma permitir un mayor control de los estados normales y reaccionar en los casos anómalos (ej. una subida de temperatura, una bajada de presión, etc.).

Cabe destacar en este TFG, que la distribución "geográfica" de los elementos de un Smart Grid se ha establecido teniendo en cuenta el modelo real de generación y distribución de energía; es decir: generación de energía → transformación de energía → distribución de energía. Aparte de esto, y al tratarse de una simulación sobre un entorno ficticio, los datos recibidos de los elementos

de control (ej. sensores) han sido también establecidos de forma virtual proporcionando valores completamente aleatorios. Esto también significa que este proyecto abre la posibilidad de retomar el trabajo en un futuro próximo con el fin de establecer los medios necesarios para conectar el mundo virtual con el mundo real. Por lo tanto, resaltamos una vez más las limitaciones de este trabajo para ajustarnos a los requisitos y magnitud de un TFG, con la posibilidad de extenderlo en un futuro.

1.3 Objetivos principales

Uno de los objetivos principales que se desean alcanzar con este proyecto es dar un primer paso en la **integración** de la realidad virtual en la industria. El sistema pretende demostrar que es posible crear una interfaz de usuario más atractiva e ilustrativa que las existentes (ej. los tradicionales HMI), reduciendo la necesidad de aprender sobre la herramienta o sus opciones de visualización, o establecer pre-requisitos de aprendizaje para su manejo.

Por otro lado, se pretende representar un caso de uso de forma realista, proporcionando al usuario toda la **información** sobre el sistema y su entorno físico en tiempo real. Concretamente, la información se ha representado de diferentes formas: usando mapas de calor, paneles de texto que se actualizan dinámicamente, iconos y otros elementos de mundo físico (como luces, maquinaria o elementos geográficos).

1.4 Objetivos a medio-largo plazo

Desde el principio, se tenía claro que este proyecto sería **el primero de una serie** de trabajos relacionados con este nuevo ámbito de aplicación. De hecho, el trabajo de este TFG es solo la **demostración** de que este tipo de sistemas están al alcance de la industria, y su implementación podría generar enormes beneficios al sector. Este modelo es sólo eso, un **modelo o un entorno de simulación** de lo que podría llegar a representarse a manos de cualquier usuario de la aplicación. Es decir, HeatMap3D es la representación hipotética de un sistema de **control de una red energética inteligente**.

Sin embargo, las extensiones de la propia herramienta HeatMap3D pueden ser muy diversas, como, por ejemplo:

- Acoplar el modelo existente con un otro caso de uso cuyos datos de entrada los proporcione un **sistema real**, permitiendo la monitorización y la gestión de un sistema de control real. En definitiva, establecer conexión del mundo real con el virtual.
- Modelar un nuevo caso de uso o modificar el modelo existente bajo demanda para adaptarlo a las diversas exigencias de la propia industria del sector. Esto abriría numerosas posibilidades en el mercado, y, a largo plazo, podrían acabar sustituyendo a los sistemas HMI tradicionales.
- Conseguir una **interacción plena** por parte de los usuarios, permitiendo modificar parámetros para manejar situaciones críticas. Para ello, será necesario un conocimiento profundo del entorno, sus posibilidades y sus limitaciones. Entendemos por crítico que un pequeño fallo a la hora de modificar los parámetros podría derivar a la inestabilidad de todo el sistema

puesto que el sistema operado no es más que una representación de un sistema real.

- **Auditorías** del sistema real a través de mediciones recogidas por el modelo. Se podrían construir y simular sistemas para monitorizar sistemas de control, algo así como un **meta-sistema de control**, para tener un registro de cómo están funcionando estas simulaciones. Algunos indicadores podrían obtenerse a simple vista, como ver si el motor gráfico está funcionando correctamente, si las interfaces están sobrecargadas o si forzar anomalías para comprobar la respuesta del sistema. Otro tipo de pruebas que se podrían hacer sería realizar **mediciones** simultáneas con instrumental y personal externos a la empresa para así comprobar la precisión y fidelidad de la simulación.

Y pueden aplicarse para diversos ámbitos de aplicación, abriendo nuevas formas de aplicar la simulación y el modelado de entornos 3D, como, por ejemplo:

- **Monitorización** de cada sistema de control desde cualquier parte del mundo y en cualquier momento. Esta flexibilidad permitiría a los operarios trabajar desde casa o desde cualquier otra localización como podrían ser las sedes de una empresa en otros países. La mayor ventaja en este sentido es sin duda que no haría falta conocer el entorno a la perfección para poder comprender sus propiedades, si no que podrías "verlo" mediante la simulación en tres dimensiones.
- Construir sistemas de formación y **entrenamiento** interactivo para nuevos operarios de sistemas de control. En este sentido, ya hay compañías que apuestan por este tipo de "training" puesto que tienen un mayor índice de aceptación y **asimilación de conceptos** por parte de los empleados al

tratarse de tutoriales más dinámicos, no una simple presentación o vídeo que no permite ningún tipo de interacción.

1.5 Estructura de la memoria

Durante la elaboración de esta memoria se ha intentado que siga una curva de aprendizaje lógica e intuitiva. Se ha desglosado en los siguientes apartados:

1. Introducción

Este capítulo introduce el proyecto, estableciendo la motivación, el contexto y los objetivos principales, y resaltando una vez más, algunos objetivos y aplicaciones que se pueden desarrollar a medio-largo plazo con el fin de extender el proyecto en un futuro cercano. Por último, la sección finaliza con la estructura de la memoria será útil para desglosar la organización de este documento.

2. Avances tecnológicos de la Industria

Una introducción sobre la utilización de la energía en la industria es extensamente detallada en este capítulo, donde se repasarán los antecedentes, el estado del arte y algunas iniciativas innovadoras que se están desarrollando en la actualidad y el impacto generacional que tendrá esta nueva ola tecnológica.

3. Smart Grids

Este capítulo describe el concepto de "Smart Grid" o redes eléctricas inteligentes, añadiendo ciertos inputs relevantes al contexto y destaca las tecnologías que se utilizan hoy en día en este campo de aplicación, como puede ser: el Big Data y data analytics, IoT, impresión en 3D, Inteligencia Artificial, etc.

4. Análisis y diseño de HeatMap3D

En este capítulo se expondrá la estimación temporal, así como la metodología de desarrollo utilizada durante la elaboración del modelo y el proceso de ingeniería de requisitos realizado previamente para obtener los requisitos funcionales y no funcionales necesarios para el proyecto.

5. Herramientas y técnicas utilizadas.

Este capítulo recopila y analiza las herramientas utilizadas para llevar a cabo el proyecto. Concretamente, se destacarán las herramientas principales de desarrollo de este proyecto. Una de ellas es **Unity**, la cual se puede considerar como el motor de modelado 3D y **Visual Studio** para el desarrollo de diferentes scripts utilizados en el modelo para alcanzar los requisitos funcionales y estéticos del caso de uso.

6. Desarrollo del proyecto.

En este capítulo se explicarán las diferentes fases a la hora de desarrollar el proyecto y construir el entorno de HeatMap3D. Se explicarán las decisiones de diseño y se desglosarán las dificultades que se han ido encontrando durante el desarrollo del proyecto, así como las soluciones elegidas para enfrentarlas.

7. Conclusiones y futuro.

En este capítulo se expondrán las conclusiones a las que se han llegado tras completar el proyecto. Por último, como se ha mencionado anteriormente, este proyecto está pensado para ser el primero de proyectos futuros, por lo que en este capítulo se estudiarán las posibles ampliaciones posibles.

8. Bibliografía

Finalmente, se agrupan las fuentes consultadas durante el desarrollo del proyecto y la elaboración de la memoria. Además, se recopilará todo el material de código abierto proveniente de páginas externas que se ha utilizado para el modelado del entorno y el desarrollo de los scripts.

2. Avances Tecnológicos de la Industria

2.1 Antecedentes

A comienzos del siglo XVIII la sociedad se dividía en dos estamentos diferenciados: Terratenientes y campesinos. La principal actividad económica era la **agricultura**. Los campesinos labraban la tierra de la nobleza y el clero utilizando técnicas rudimentarias como el barbecho. Era un trabajo duro y muy mal pagado. Las **hambrunas** se sucedían una tras otra en las sociedades preindustriales, no sólo por la falta de alimentos si no por el hecho de que una buena cosecha desencadenaba un aumento de la natalidad, lo que obligaba a los campesinos a sobreexplotar los cultivos, lo que a la larga hacía disminuir la producción y volvía a traer el hambre (Íñigo, 2012).

Aunque no era ni de lejos tan popular como la agricultura, la **artesanía** también era una importante actividad económica. Se desarrollaba principalmente en solitario y se organizaba en gremios. Se fabricaban objetos de forma manual que eran únicos y caros.

Autores como Deane (1979), Gershenfeld (2012) o Cameron (1982) sitúan la **primera revolución industrial** en Inglaterra, en la segunda mitad del siglo XVIII.

Landes (1979) define el término de revolución industrial como el conjunto de innovaciones tecnológicas que sustituyen a la habilidad y la fuerza humana por **maquinaria** y **energía** mecánica. La introducción de sistemas mecánicos como la máquina de vapor (*figura 2.1*) fue sin duda la auténtica revolución, puesto que, a diferencia de los sistemas de producción tradicionales, no dependían de las condiciones climatológicas o de las **limitaciones** físicas de los trabajadores. Este fenómeno aceleró el éxodo de trabajadores procedentes de zonas rurales a los núcleos urbanos donde se situaban las nuevas fábricas.

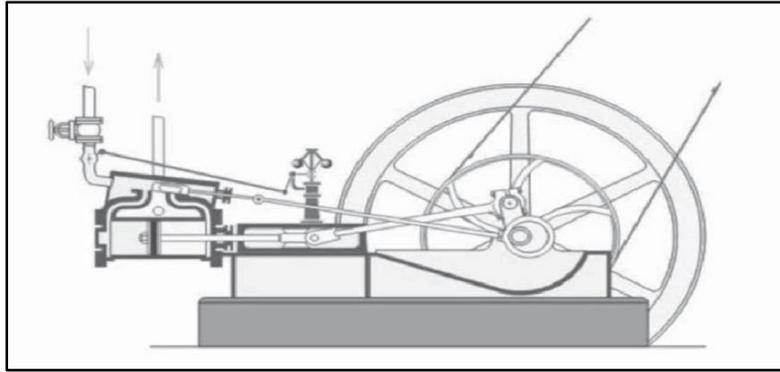


Figura 2.1: Diagrama de una máquina de vapor

Fuente: (Researchgate, 2014).

El periodo localizado entre 1850-1870 se conoce como **segunda revolución industrial** y se caracteriza por una aceleración sustancial en la implementación de mejoras en los sistemas de producción existentes. Esta revolución no fue sólo tecnológica, si no que tuvieron lugar numerosos **avances científicos** que fueron determinantes a la hora de desarrollar la maquinaria de esta nueva oleada tecnológica. Algunas características importantes de este periodo fueron la utilización de combustibles fósiles y el uso de la electricidad en las fábricas. Por otro lado, cabe destacar que este fenómeno también afectó a las propias técnicas de producción (ejemplificadas en la *figura 2.2*) y organización del trabajo, apareciendo el **taylorismo** que inspiró las técnicas de producción en cadena (Littler, 1978).



Figura 2.2: Cadena de montaje

Fuente: (silicon.nyc, 2017)

La **tercera revolución** industrial, también conocida como revolución científico-tecnológica, se vio marcada por el encarecimiento de los **combustibles fósiles**, lo que provocó la búsqueda de fuentes de **energía renovables** (como la producida por aerogeneradores como los de la *figura 2.3*). Además, la introducción de la **informática** y la **robótica** mejoraron sustancialmente el rendimiento en los sistemas de producción al automatizar la maquinaria utilizada en la industria. La ingeniería **genética** irrumpió a mediados del siglo XX, permitiendo avances en el campo de la medicina, la alimentación e incluso en la investigación de nuevos materiales para la industria (Roel, 1998). Otro campo que tuvo notable importancia fue el de las **comunicaciones**, concretamente la aparición de **Internet** permitió el acceso a la información desde cualquier parte del mundo de una forma sencilla y rápida.



Figura 2.3. Aerogeneradores

Fuente: (El País, 2017)

Por último, es necesario introducir el concepto de **cuarta revolución** industrial. Algunos autores hablan de que nos encontramos en plena transición (dramatizada en la *figura 2.4*) a una nueva era tecnológica (Marsh, 2012). Esta oleada estará marcada por la **interconexión** entre los diferentes

dispositivos tecnológicos, que podrán comunicarse entre sí, anticipar posibles fallos, e incluso, recuperarse de estados de error de forma autónoma. Además, el deseo de encontrar energías renovables y **sostenibles** para hacer frente a la creciente **demanda energética** sigue patente hoy en día, tomando el relevo a lo que comenzó con la tercera revolución industrial.



Figura 2.4. Evolución Industrial

Fuente: (bcg.com, 2014)

2.2 La industria actual

Sundblad afirma en (Sundblad, 2018) que cuando las compañías se plantean unirse a esta carrera por el nuevo liderazgo tecnológico y energético de la cuarta revolución, no deberían preguntarse "*cómo adoptar estas técnicas*" si no "*cómo se pueden resolver los problemas de mi compañía utilizando estas técnicas*".

Las grandes compañías están apostando cada vez más por captar a profesionales capaces de desarrollar algoritmos de inteligencia artificial

basados en **machine-learning**. Google, sin ir más lejos, tiene un programa llamado Google AI Residence (ai.google, 2019) en el cual ofrece un programa de tutorización para estudiantes del campo de la inteligencia artificial. Otro gigante del sector, Amazon, ofrece servicios de machine learning completamente funcionales a las empresas que lo deseen (AWS Amazon, 2019). Lo sorprendente es que este tipo de tecnologías no sólo se utilizan en el ámbito del software, en Randstad están desarrollando un algoritmo de selección de personal utilizando Big Data analytics (Randstad.es, 2016)

En España se han llevado a cabo numerosas iniciativas que utilizan la tecnología del Internet de las Cosas (o **Internet of Things**) como, por ejemplo, un despliegue de sensores y otros dispositivos inteligentes en las grandes ciudades para monitorizar el tráfico, la contaminación o la gestión responsable de la luz o el agua. Localizada en nuestro país encontramos ElevenPaths (elevenpaths.com, 2019), la reputada unidad de ciberseguridad de Telefónica que ha creado una división especializada para detectar nuevas amenazas en el campo del IoT.

En el campo de la simulación hay diversos enfoques posibles. En primer lugar, tenemos el ámbito de la simulación estructural, por ejemplo: en la arquitectura, la aeronáutica o la industria automovilística. Ciertos investigadores llevan décadas destacando la importancia de las simulaciones a la hora de estudiar y optimizar la topología, la aerodinámica, e incluso, la resistencia de ciertas estructuras o materiales (Waqas Saleem, Fan Yuqing y Wang Yunqiao, 2008).

Otro enfoque es el que este caso de estudio ocupa, la interacción "**human-machine**". La industria tiene que apuntar en una dirección en la que no sólo se invierta en software intuitivo para los "clientes", sino que

también debe orientarse a los propios trabajadores. Existen incluso técnicas de desarrollo basadas en simulaciones como el desarrollo orientado a simulaciones cuyo flujo está representado en la *figura 2.5*.

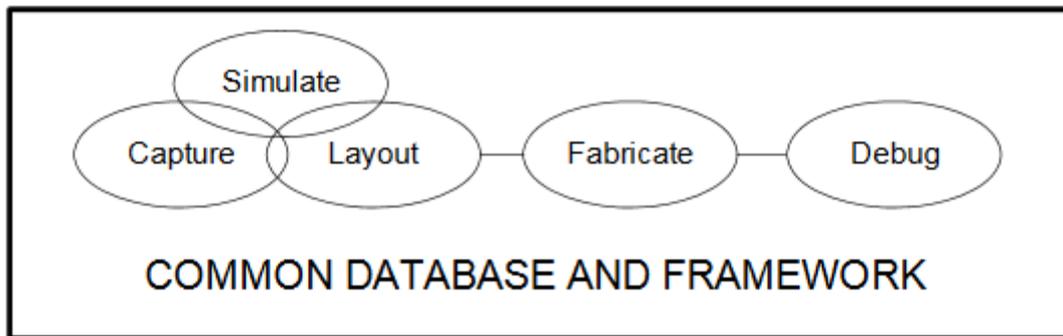


Figura 2.5: Simulation driven development

Fuente: (Glidden, 1993)

2.3 Impacto en la economía

No son pocos los autores que han profetizado un cambio radical en el tipo de perfiles laborales que se buscarán en esta nueva oleada tecnológica. Algunos medios llegan a afirmar que se destruirán 3.5 puestos de trabajo por cada uno que se construya mediante la robótica (El País, 2017). Parece evidente pensar que el uso de robots ahorraría costes a la larga en ciertos casos, como, por ejemplo, el sector agrícola o mecánico. Por el contrario, otros investigadores como Decker, Fischer y Ott afirman en (Decker *et al.*, 2017) que el futuro no reside en sustituir a los humanos por las máquinas, si no la colaboración entre ambos, ejemplificada en la *figura 2.6*, en la que técnicos pilotan drones para inspeccionar infraestructuras.

El presente proyecto pretende ayudar a suavizar el impacto que la llegada de estas tecnologías podría tener, demostrando que es posible crear simulaciones para que los trabajadores “aprendan” a utilizar todo tipo de sistemas, sin importar su complejidad de forma interactiva y eficaz. Es una realidad que existe un cambio

generacional y que la vertiginosidad de los cambios tecnológicos asusta a una gran parte de la población activa (20minutos.es, 2017). Hay muchas personas que no se sienten preparadas para “adaptarse” a cierto tipo de sistemas informáticos o metodologías de trabajo. Las simulaciones pueden reducir esa “pendiente” en la curva de aprendizaje necesaria, acercándose al dominio de usuarios que no sean lo que denominamos “nativos digitales”.



Figura 2.6. Drones para inspeccionar paneles solares.

Fuente: (utilitydive.com, 2018)

Por último, es necesario destacar que cada día que pasa, esta nueva revolución tecnológica va dejando de ser una utopía y se vuelve más real, pero no podemos perder de vista que es algo que todavía se está desarrollando, algo que nunca había pasado y que hay que construir poco a poco. En el Fondo Económico Mundial de 2017 (El País, 2017), se afirmó que el 65% de los jóvenes tendrán puestos de trabajo que no existen a día de hoy. Podemos interpretar que, aunque se reducirá el personal dedicado a ciertas tareas, parcial o completamente, se crearán muchos puestos de trabajo en los ámbitos de la tecnología, la industria, la informática, las telecomunicaciones y, quizás, en otros que aún no somos capaces de imaginar.

3. Smart Grids

3.1 Contexto

Las Smart Grids, también conocidas como redes eléctricas inteligentes, pretenden ser una respuesta más fiable, segura y eficiente que los sistemas eléctricos tradicionales (Boal, 2010). Debido al aumento del precio de los combustibles fósiles y a la concienciación acerca del cambio climático, las principales potencias mundiales están invirtiendo en este tipo de sistemas, con la propuesta de reducir oxido de carbono y mejorar los costes de producción de energía para el año 2020 (Gobierno de España, 2008). Este modelo permitirá la **independencia** (hablando en términos de energía) de aquellos países que utilicen las tecnologías de esta nueva era tecnológica y las implanten en sus plantas energéticas.

En la línea de red "smart" o "inteligente", Massoud y Wollenberg definen el concepto en (Massoud & Wollenberg, 2005), el cual significa añadir inteligencia a la gestión energética, consiguiendo un canal de comunicación bidireccional entre cada componente de las subestaciones y la estación central. En las redes eléctricas inteligentes no se trata solamente de gestionar cada nodo del sistema de forma eficiente: se trata de poder monitorizar y detectar si una pieza de un nodo determinado está fallando o presenta anomalías importantes. En este tipo de sistemas prima la estabilidad y la precisión.

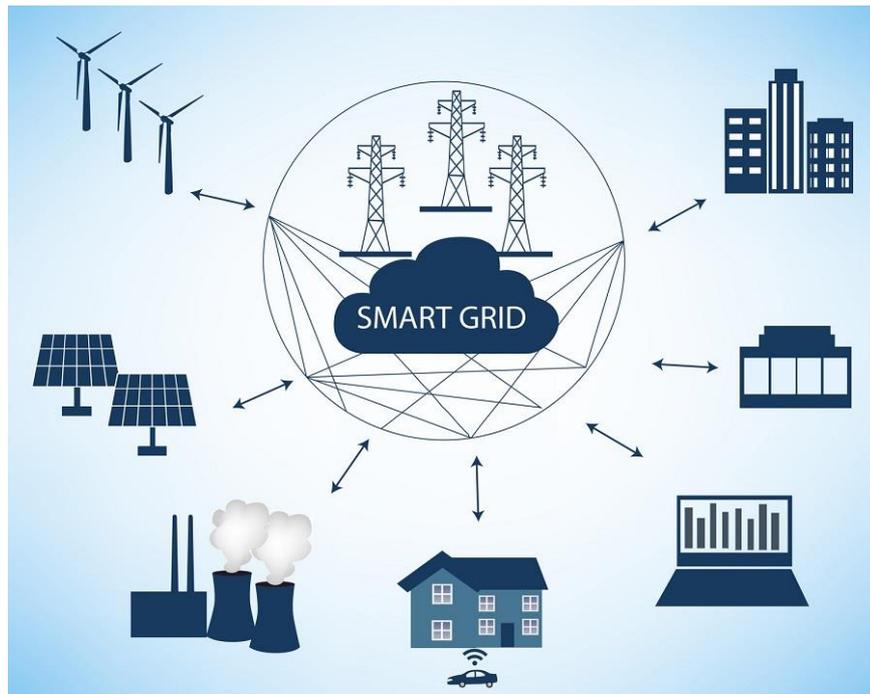


Figura 3.1. Esquema de las Smart Grids

Fuente: (gruponovelec.com, 2017)

Otra característica a destacar es el hecho de que, tradicionalmente, nadie se ha preocupado por el hecho de que, si la energía producida no se consume, se pierde. En una época en la que los recursos están cada vez más limitados es imprescindible cuidar este aspecto. Los Smart Grids aspiran a ser eficientes en este sentido, monitorizar el sistema en tiempo real y evaluar las fluctuaciones en el consumo para producir la energía necesaria en cada momento, ni más, ni menos. Además, estos sistemas rompen con la transmisión unidireccional de energía y permiten que esta fluya en ambos sentidos, permitiendo una gestión más eficiente de la misma (Díaz y Hernández, 2011).

La ciberseguridad adquiere un papel protagonista en esta nueva oleada tecnológica debido a la gran cantidad de información que se almacenará en el lado del cliente final, dispositivos desfasados, transmisión de mensajes por internet, etc.

Algunos autores sentencian que los Smart Grids deben cumplir tres requisitos de seguridad: disponibilidad del servicio en todo momento, asegurar la integridad de las comunicaciones y la confidencialidad de los datos de los clientes de la red energética (Aloul, Al-Ali, Al-Dalky, Al-Mardini & El-Hajj, 2012).

3.2 Tecnologías utilizadas en los Smart Grids

Cuando se habla de este fenómeno que está naciendo, los expertos¹ enumeran una serie de tecnologías que están siendo y serán determinantes para el desarrollo de los nuevos sistemas de producción y gestión energética:

1. Inteligencia artificial:

Esta disciplina es transversal a algunas tecnologías que se mencionarán posteriormente (p.ej. el Internet de las Cosas, la realidad aumentada o el análisis de datos). Para Kolberg y Zühlke (2015), las fábricas inteligentes son aquellas que poseen productos inteligentes que conocen el proceso de producción y lo se comunican con máquinas inteligentes.

El caso de los Smart Grids no es diferente, en este caso el producto inteligente podría ser la energía, que proporciona a los sensores información acerca del consumo, el estado de la red, anomalías, etc. Los sensores transmiten esta información a otros dispositivos localizados en las subestaciones, permitiendo dar una respuesta rápida y eficaz a las necesidades del sistema.

¹ D. Gorecky, M. Schmitt, M. Loskyll and D. Zühlke, "Human-machine-interaction in the industry 4.0 era," *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, Porto Alegre, 2014, pp. 289-294.

2. Robots autónomos:

Como se ha mencionado anteriormente, la utilización de maquinaria está a la orden del día en la industria. Los autómatas cada vez son capaces de desempeñar tareas más complejas. Un robot autónomo es aquel capaz de trabajar por sí mismo, **sin la supervisión de un humano**, siendo capaz de desarrollar nuevas funcionalidades por sí mismo y comunicarse con otros elementos del sistema para optimizar la producción (como es el caso de la "Smartflower" de la *figura 3.2*).



Figura 3.2. "Smartflower", un panel solar inteligente.

Fuente: (gadgetzz.com, 2016)

3. Sistemas integrados vertical y horizontalmente

Se pretende mejorar la experiencia del usuario conectando las diferentes plataformas de las compañías entre sí, permitiendo la

automatización procesos de gestión entre diferentes sistemas, por ejemplo, entre un departamento de servicios y uno de producción.

4. Cloud Computing

Los servicios Cloud o “en la nube” ya son muy populares entre los usuarios. Mell, P. y Grance, T. (2011) definen el Cloud Computing como un modelo para permitir el acceso desde cualquier lugar del planeta y bajo demanda, de una serie de recursos computacionales. Los servicios de monitorización de procesos también se encontrarán en la nube en el futuro.

5. Internet de las Cosas (IoT)

El uso del Internet de las Cosas ya se ha consolidado como una parte esencial de nuestro día a día. En la industria ya se están utilizando sensores y dispositivos inteligentes que permiten la comunicación entre máquinas para recabar datos y tomar decisiones en conjunto. En las redes eléctricas inteligentes se pretende desplegar toda una flota de sensores (*figura 3.3*) capaces de detectar variaciones en el consumo y anomalías en los propios componentes y enviar dicha información a otras subestaciones.



Figura 3.3. Dispositivo Tollgrade para detectar interrupciones del servicio eléctrico.

Fuente: (NY Times, 2015)

6. Impresión 3D

Con estas técnicas se pretende disminuir los costes durante la elaboración de prototipos como el de la *figura 3.4*. Se puede crear piezas de cualquier forma y tamaño a una velocidad asumible, lo que también reduce los gastos en transporte de materiales de una zona a otra. Este tipo de técnicas reducirían el coste material a la hora de cambiar pequeñas piezas defectuosas de ciertos componentes en las redes eléctricas inteligentes.



Figura 3.4. Prototipo impreso en 3D de un componente para electrobombas. Fuente: (3DZ, 2016)

7. Big Data

“Data-driven decisions are better decisions” (Harvard Business Review, 2012). Hoy en día, en plena lucha por la privacidad, los dispositivos electrónicos recopilan datos constantemente. Desde los productos que se han buscado en una web hasta el restaurante que se suele visitar los fines de semana. Las compañías utilizan estos datos para analizar las tendencias de consumo y, por ejemplo, ofrecer productos “personalizados” que puedan interesar al consumidor.

Los algoritmos conocidos como de “Big Data” son capaces de procesar una enorme cantidad de información en poco tiempo y tomar decisiones una vez se han analizado los datos. Los Smart Grids pretenden utilizar el análisis de datos para poder dar respuesta a las variaciones de consumo en la red de la forma más eficiente posible.

Además, permitirán al usuario disponer de datos para poder tomar decisiones en cuanto a su propio consumo energético.

8. Simulaciones 3D

El campo de la simulación 3D ya se ha asentado como una opción rentable en la industria de los videojuegos y ahora pretende hacerse un hueco en otro tipo de industrias. Para Uhlemann, Lehmann y Steinhilper (2017), las simulaciones rompen con los estudios estadísticos tradicionales sobre los sistemas de producción, permitiendo un estudio dinámico que tiene en cuenta las variaciones en el tiempo. Cabe destacar que para que las simulaciones sean realistas, se necesita una batería de datos de entrada actualizados.

El presente proyecto va un paso más allá, pretendiendo demostrar que las simulaciones también podrían usarse en la monitorización en tiempo real de una red eléctrica inteligente, donde los datos de entrada no provienen de una base de datos, si no de dispositivos que forman parte del propio sistema representado.

9. Realidad aumentada

Quizás este sea el campo que se encuentra menos desarrollado actualmente si lo comparamos con otras tecnologías coetáneas. La realidad aumentada aspira a superponer elementos virtuales como, por ejemplo: texto, imágenes o colores en el campo de visión del usuario.

Tradicionalmente se utilizaban cámaras para procesar las imágenes y luego introducir los elementos, pero cada vez más se está planteando la viabilidad de dispositivos como gafas, por ejemplo, las Google Glasses de la *figura 3.5*. Azuma (1997) ya profetizaba que en 25 años todas las personas utilizarían este tipo de gafas de realidad aumentada en su día a día. Este tipo de tecnologías podrían ayudar a los operarios de las SG a arreglar averías en diversos componentes de forma rápida sin necesidad de tener un extenso conocimiento sobre cada componente del sistema.



Figura 3.5. Google Glass Enterprise Edition.

Fuente: (Xakata, 2017)

4. Análisis y diseño de HeatMap3D

4.1 Estimación temporal

En este apartado se ha desglosado la estimación temporal de los diferentes apartados del proyecto.

FASE	HORAS
Fase 1	
1.1 Estudio del estado del arte, especialmente en los Smart Grids, la evolución de tecnológica de la industria y las tecnologías emergentes.	12
1.2 Análisis de requisitos del caso de uso propuesto (escenario, objetos, comportamiento y posibles anomalías)	15
1.3 Análisis de las tecnologías disponibles, evaluación y selección.	6
1.4 Aprendizaje básico sobre Unity	6
Fase 2	
2.1 Modelado del escenario en tres dimensiones	20
2.2 Modelado de los objetos necesarios para el caso de uso en tres dimensiones	35
2.3 Búsqueda en fuentes externas de modelos en tres dimensiones de código abierto	20
2.4 Integración del escenario y los modelos obtenidos en 2.2 y 2.3	30
Fase 3	
3.1 Sistema de detección de anomalías, alerta, iconos y mapas de calor.	30
3.2 Integración del modelo con el sistema de avisos de forma que permita interactuar al usuario.	12
3.3 Análisis de los activadores necesarios que desencadenarán los avisos.	4
3.4 Desarrollo de los activadores que permitirán activar las anomalías.	20

3.5 Implementación y depuración de los activadores.	40
Fase 4	
4.1 Validación de los componentes de acuerdo a los requisitos identificados en Fase 1.	10
4.2 Optimización de componentes y servicios de acuerdo a 4.1	6
Fase 5	
5.1 Documentación formal del TFG y preparación de la defensa.	30
TOTAL	296

Tabla 4.1. Estimación temporal.

Fuente: (Planificación temporal de HeatMap3D)

Estas fases junto con la programación real del desarrollo de este TFG también se pueden apreciar en la *figura 4.1*. Esta figura representa el tradicional diagrama de Gantt, caracterizando el desarrollo y ejecución de las actividades dentro del proyecto y a lo largo del tiempo.

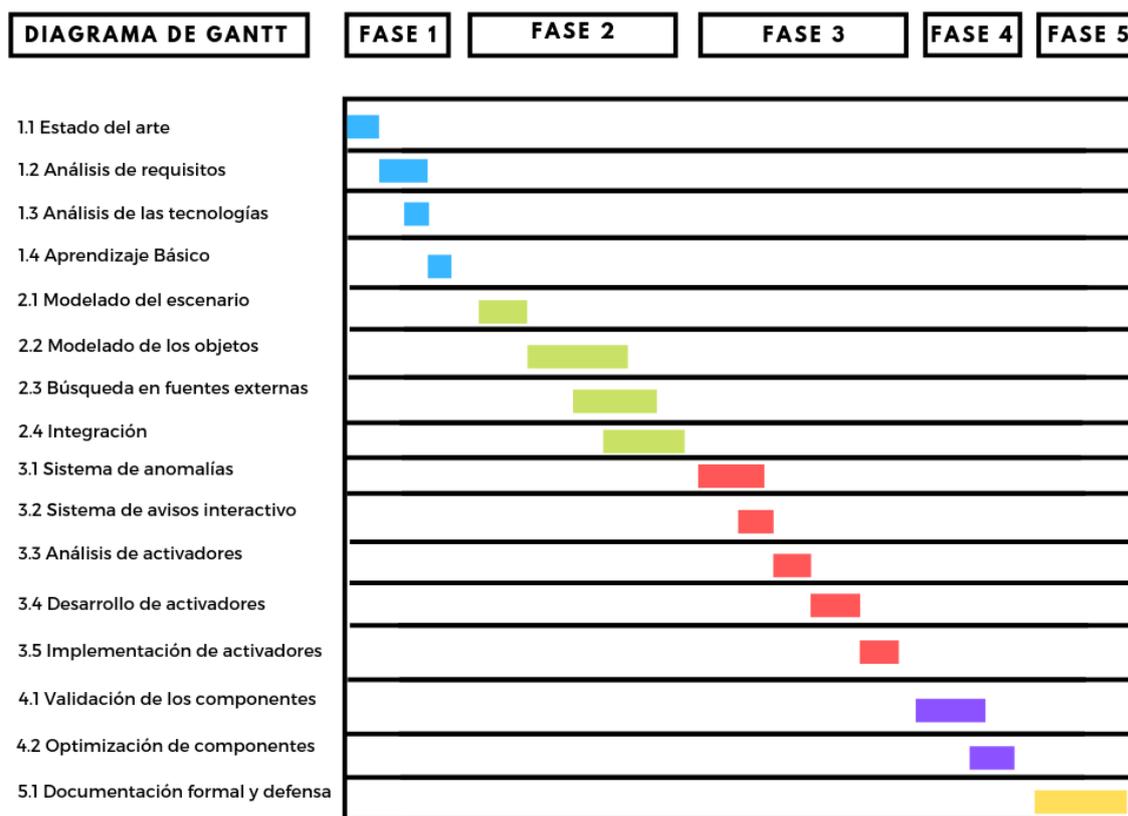


Figura 4.1. Diagrama de Gantt.

Fuente: (Planificación temporal de HeatMap3D)

4.2 Metodología de desarrollo

Para el desarrollo del proyecto se han utilizado metodologías de desarrollo ágil. Concretamente, se ha utilizado **Extreme Programming** (XP o Programación Extrema), representada en la *figura 4.2*. Esta técnica de desarrollo ágil se caracteriza por permitir que, en un escenario favorable, un usuario o cliente pueda ir proponiendo cambios y mejoras en un proyecto sobre la marcha.

Como hemos dicho antes, para cumplir el objetivo de esta metodología (simplicidad y productividad), el escenario debe ser favorable para que esta metodología sea rentable puesto que, en ciertos sistemas, el coste de modificar el

proyecto puede ser demasiado grande en comparación a si se hubiese utilizado otra metodología de desarrollo (como **SCRUM** o Kanban). Otro punto a favor a la hora de elegir fue el hecho de que se pretendía mantener el contacto con los tutores en todo momento. Tras la elaboración del proyecto, se puede afirmar que ellos que han ejecutado el rol de "clientes", proponiendo ciertos cambios en cada iteración para mejorar el proyecto o para evitar desarrollo innecesario, lo que le ha dado realismo a la aplicación de esta metodología.

Extreme Programming (XP)

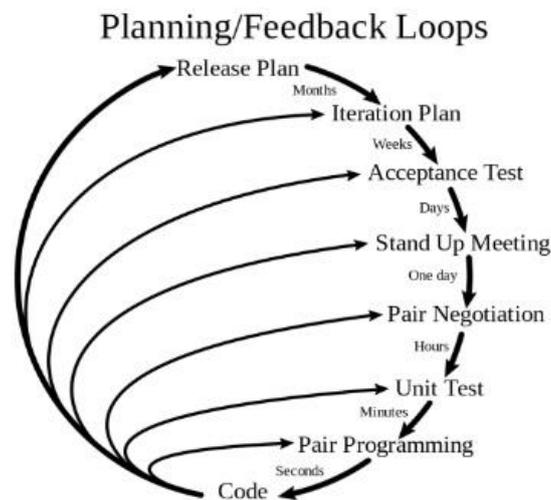


Figura 4.2: Extreme Programming Cycle

Fuente: (Slideshare.net, 2014).

Cabe destacar que no se tenía conocimientos de modelado en tres dimensiones, por lo que desde un principio se tuvo en cuenta que el proyecto podría suponer demasiado esfuerzo para un sólo trabajo, por lo que, desde un principio, se tuvieron en cuenta ciertos márgenes para incorporar los posibles cambios futuros no como un "problema" si no como una parte más del desarrollo del proyecto, tal y como sugiere la filosofía "Extreme Programming".

Utilizar Extreme Programming ha sido beneficioso a la hora de desarrollar el proyecto debido a la cantidad de cambios que se han ido introduciendo durante el ciclo de desarrollo, así como la simplicidad y la independencia entre los elementos del entorno, permitiendo modificaciones que en la mayor parte de las ocasiones no han impactado en el resto del modelo. Debido al tiempo y esfuerzo dedicado a la planificación y la aplicación de la ingeniería de requisitos para aclarar los puntos dudosos en el proyecto, el desarrollo ha sido bastante lineal, sin apenas sorpresas o cambios en la idea inicial.

4.3 Análisis del proyecto

4.3.1 Requisitos funcionales

Los requisitos funcionales son una especie de "contrato" que especifica aquellas funcionalidades que aportará un sistema, especificando el comportamiento final de nuestro proyecto. En la *tabla 4.1* se han descrito brevemente cada uno de los requisitos funcionales identificados.

ID	Nombre	Descripción
RF01	Movimiento	El usuario podrá moverse utilizando las teclas "W" "A" "S" y "D" del teclado.
RF02	Control de la vista	El usuario podrá modificar su campo de visión utilizando el botón derecho del ratón.
RF03	Visualizar texto flotante	El usuario podrá monitorizar en todo momento los datos representados en el texto flotante, los cuales se "girarán" hacia la cámara del usuario.
RF04	Visualizar anomalías	El usuario podrá monitorizar en todo momento la existencia o ausencia de anomalías en el sistema mediante la aparición de iconos que lo indiquen.

RF05	Diferenciar objetos Interactivos	El usuario podrá entrever qué objetos son interactivos.
RF06	Interacción con objetos	El usuario podrá interactuar con ciertos elementos del sistema, ya sea pulsando con el ratón o acercándose con la cámara.
RF07	Gestionar anomalías	El usuario podrá modificar el estado de las anomalías al interactuar con ciertos elementos del sistema.
RF08	Control de scripts	El sistema ejecutará scripts en segundo plano de forma concurrente y producirá cambios en el entorno.
RF09	Iluminación	El sistema gestionará la iluminación y las sombras del entorno.
RF10	Postprocesado	La cámara utilizará técnicas de post-procesado de la imagen final.

Tabla 4.1. Requisitos funcionales de HeatMap3D

4.3.2 Requisitos no funcionales

Los requisitos no funcionales son las especificaciones referidas a las propiedades del sistema tales como el rendimiento o la complejidad del mismo. En la *tabla 4.2* se observan una serie de características sobre cómo ha de comportarse el sistema.

ID	Nombre	Descripción
RNF01	Entorno realista	El entorno del proyecto debe ser lo más realista posible, incorporando representaciones en tres dimensiones de objetos y estructuras reales para facilitar la comprensión del modelo.
RNF02	Conocimientos sobre Smart Grids	Aunque no es obligatorio, es bastante recomendable que el usuario esté familiarizado con el dominio de las

		redes eléctricas inteligentes.
RNF03	Estructuración del modelo	La jerarquía de elementos del modelo seguirá una estructura lógica y estará organizado de forma intuitiva.
RNF04	Limpieza del código	El código seguirá una serie de buenas prácticas para garantizar un código limpio, seguro y bien documentado.
RNF05	Escalabilidad	El proyecto deberá permitir posibles ampliaciones futuras de una forma sencilla gracias a la organización de la jerarquía y a la limpieza del código.
RNF06	Portabilidad	El proyecto permitirá la conexión con un sistema real hipotético sin necesidad de realizar muchos cambios en el mismo.
RNF07	Hardware	El sistema no precisará de componentes de una gama excesivamente alta para poder ejecutar el modelo, pero será necesario utilizar componentes de gama media.

Tabla 4.2 Requisitos no funcionales de HeatMap3D

4.3.3 Casos de uso

4.3.3.1 Actores implicados

Los actores son las entidades que interactúan con el proyecto. En esta ocasión, diferenciamos dos actores:

- **Sistema:** Representa el entorno en el cual se ejecuta el modelo. En este caso Unity.
- **Usuario:** Representa a la persona que va a utilizar HeatMap3D.

4.3.3.2 Diagramas de casos de uso

A continuación, en la *figura 4.3* se observa el diagrama de casos de uso del actor “Sistema”. Se han omitido las tareas que ya realiza el motor Unity de forma intrínseca. Se observan tres casos de uso principales: “Ejecutar Animación”, “Ejecutar Scripts” y “Gestionar Iluminación”. Todos ellos se ramifican para definir un comportamiento más concreto.

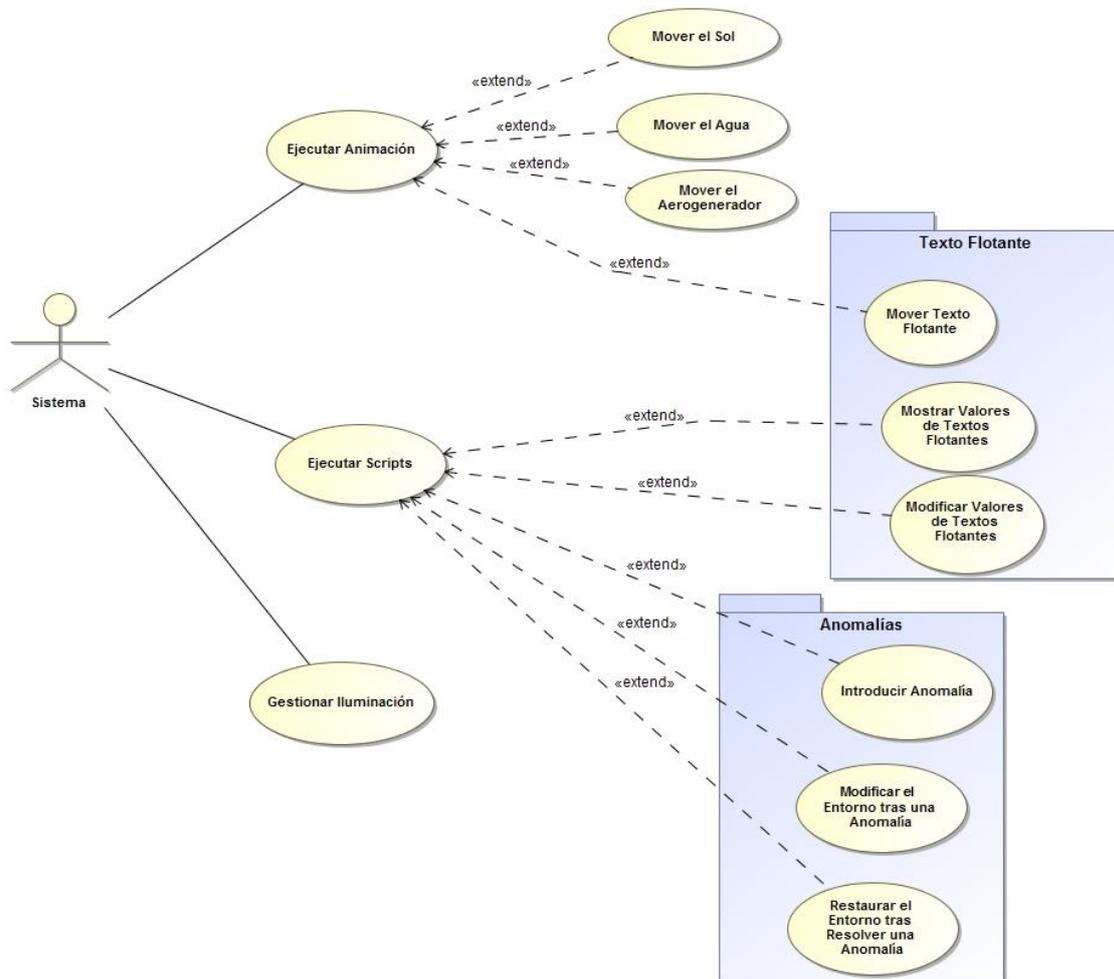


Figura 4.3. Diagrama de casos de uso del sistema.

Por otro lado, se ha diseñado un diagrama de casos de uso para el usuario representado en la *figura 4.4*, con todas aquellas operaciones que puede realizar en el sistema. No se ha contemplado el caso de uso de visualizar los elementos del entorno puesto que se considera transversal al resto de casos de uso y, además, bastante trivial como para ser definido.

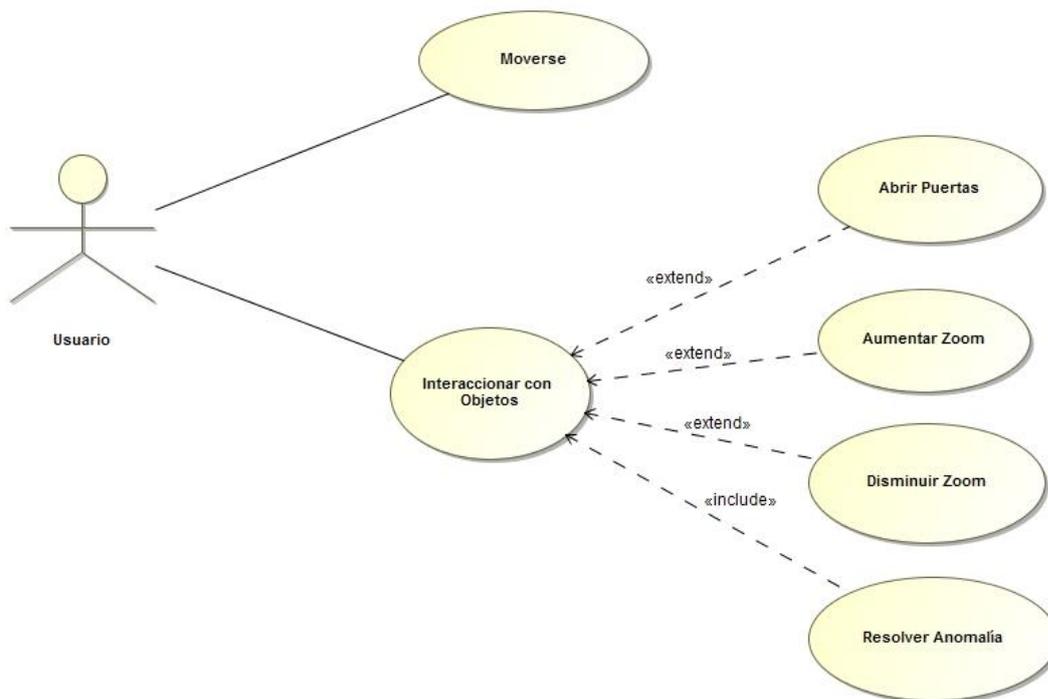


Figura 4.4 Diagrama de casos de uso del usuario.

4.3.3.3 Definición de los casos de uso

Para definir cada uno de los casos de uso de la *figura 4.3: Diagrama de casos de uso del sistema*, se han elaborado las *tablas 4.3, 4.5, 4.6, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13 y 4.14*. En ellas se han definido sus características junto a una breve descripción y una secuencia en la cual se ejemplifica una ejecución del caso de uso.

Caso de Uso 1	
ID	CU-1
Nombre	Ejecutar Animación
Actor	Sistema

Descripción	El sistema es capaz de modificar la posición de una serie de elementos del entorno, simulando "movimiento" de algunos objetos.
Precondiciones	Existencia de un objeto con apariencia física que tiene el componente "Animator" y una animación configurada previamente.
Secuencia	1. El sistema reproducirá la animación siguiendo un patrón temporal. 2. El sistema detendrá la animación o volverá a reproducirla tras completar el ciclo temporal o recibir un evento externo.
Postcondición	La animación continúa ejecutándose en bucle o bien termina su animación y vuelve a su estado inicial.
Excepciones	En la mayoría de los objetos, las animaciones se encuentran "activas" en todo momento, pero en otros casos (como en los modelos de las puertas), las animaciones se reproducirán tras producirse un evento (por ejemplo, un <i>click</i> o una aproximación de la cámara).
Requisitos Asociados	RF01, RF02, RF03, RF06, RF07, RF08

Tabla 4.3 Tabla del caso de uso "CU-1"

Caso de Uso 2	
ID	CU-2
Nombre	Ejecutar scripts
Actor	Sistema
Descripción	El sistema es capaz de modificar el estado de una serie de elementos del entorno, simulando "movimiento" de algunos objetos.
Precondiciones	Existencia de un objeto que tiene el componente "Script" y una script previamente creada y sin errores de compilación vinculado al mismo.

Secuencia	<ol style="list-style-type: none"> 1. El sistema ejecutará el script en una serie de intervalos previamente definidos en el propio script. 2. El sistema modificará el estado de un objeto. 3. El sistema terminará el ciclo y volverá a ejecutarse en caso de ser un script ejecutado continuamente.
Postcondición	El script continúa ejecutándose en bucle o bien termina su ejecución y vuelve a su estado inicial.
Excepciones	En la mayoría de los objetos, los scripts se encuentran ejecutándose en todo momento, realizando cambios en modelos del entorno. Sin embargo, en otros casos (por ejemplo, en los modelos de las puertas o en los aerogeneradores), los scripts se encontrarán "en espera" a que un evento los despierte (por ejemplo un click o una aproximación de la cámara).
Requisitos Asociados	RF01, RF03, RF05, RF06, RF07, RF08, RF09

Tabla 4.4 Tabla del caso de uso "CU-2"

Caso de Uso 3	
ID	CU-3
Nombre	Gestionar Iluminación
Actor	Sistema
Descripción	El sistema es capaz de modificar la iluminación y generar sombras.
Precondiciones	Existencia de un objeto físico que pueda producir sombra y de un objeto que emita luz.
Secuencia	<ol style="list-style-type: none"> 1. Dependiendo del ángulo de la luz, un objeto proyectará una sombra.
Postcondición	El objeto en cuestión proyecta una sombra, y su apariencia física se ve alterada por la iluminación.
Excepciones	Para facilitar el renderizado y aliviar la carga procesado del modelo, se ha desactivado la generación de sombras

	para algunos modelos del entorno.
Requisitos Asociados	RF02, RF08, RF09, RF10

Tabla 4.5 Tabla del caso de uso "CU-3"

Caso de Uso 4	
ID	CU-4
Nombre	Mover el Sol
Actor	Sistema
Descripción	El sistema es capaz de modificar posición, el ángulo y la intensidad de la iluminación generada por el modelo llamado "Sol", que representa al Sol.
Precondiciones	Existencia de un objeto físico que emita luz que tenga un componente de tipo "Animator" y una animación que modifique sus parámetros de posición, intensidad y ángulo a lo largo del tiempo.
Secuencia	<ol style="list-style-type: none"> 1. Cuando el proyecto se ejecuta, se puede observar como el Sol va apareciendo poco a poco, aumentando la iluminación del entorno de forma progresiva. 2. El Sol se va moviendo, simulando el transcurso de un día completo. 3. El Sol se esconde y la iluminación se reduce drásticamente, simulando la noche. 4. El Sol vuelve a su posición inicial para comenzar un nuevo ciclo.
Postcondición	El Sol ha vuelto a su posición inicial y la iluminación del entorno es la misma que antes de comenzar el ciclo.
Excepciones	
Requisitos Asociados	RF02, RF09, RF10

Tabla 4.6 Tabla del caso de uso "CU-4"

Caso de Uso 5	
ID	CU-5
Nombre	Mover el Agua
Actor	Sistema
Descripción	El sistema es capaz de modificar posición, el ángulo de flujo y los reflejos de los modelos que simulan ser "agua".
Precondiciones	Existencia de un modelo que utilice las librerías predefinidas de Unity para generar un elemento que simule ser "agua".
Secuencia	1. De forma continuada, se puede observar como las corrientes del agua va cambiando su dirección (no su sentido), simulando cambios en el flujo y olas.
Postcondición	
Excepciones	
Requisitos Asociados	RF02, RF09, RF10

Tabla 4.7 Tabla del caso de uso "CU-5"

Caso de Uso 6	
ID	CU-6
Nombre	Mover el Aerogenerador
Actor	Sistema
Descripción	El sistema es capaz de modificar posición y el ángulo de las aspas del modelo que simula ser un aerogenerador.
Precondiciones	Existencia del modelo que simula ser un aerogenerador, que tenga un componente de tipo "Animator" y una animación que modifique sus parámetros de posición y ángulo en sus aspas.
Secuencia	1. De forma continuada, se puede observar como las aspas del aerogenerador van girando, simulando el

	comportamiento real de un aerogenerador.
Postcondición	
Excepciones	
Requisitos Asociados	RF02, RF05, RF06

Tabla 4.8 Tabla del caso de uso "CU-6"

Caso de Uso 7	
ID	CU-7
Nombre	Mover Texto Flotante
Actor	Sistema
Descripción	El sistema es capaz de modificar posición y el ángulo los elementos de texto del modelo, simulando que el texto está "flotando" en el aire.
Precondiciones	Existencia de un objeto de texto con un contenido, que tenga un componente de tipo "Animator" y una animación que modifique sus parámetros de posición y rotación.
Secuencia	1. De forma continuada, se puede observar como el texto flotante se mueve y se va girando para "mirar" al usuario.
Postcondición	El texto flotante reproduce una animación continuada. El texto flotante apunta siempre a la cámara del usuario.
Excepciones	Cuando además se produce una anomalía, el sistema modificará el elemento de texto para indicarlo.
Requisitos Asociados	RF02, RF03, RF04, RF07, RF08

Tabla 4.9 Tabla del caso de uso "CU-7"

Caso de Uso 8	
ID	CU-8

Nombre	Mostrar Valores de Textos Flotantes
Actor	Sistema
Descripción	El sistema es capaz de modificar mostrar el contenido de los textos flotantes de una forma sencilla e intuitiva.
Precondiciones	Existencia de un objeto de texto con un contenido previamente introducido. Además, este objeto dependerá de otros parámetros como el color o la fuente.
Secuencia	<ol style="list-style-type: none"> 1. De forma continuada, se puede observar el contenido del texto flotante, que nos muestra información sobre diferentes unidades de medida como temperatura, humedad, velocidad del viento, etc. 2. El usuario mira a ese texto flotante para obtener información.
Postcondición	
Excepciones	
Requisitos Asociados	RF02, RF03, RF04, RF07, RF08

Tabla 4.10 Tabla del caso de uso "CU-8"

Caso de Uso 9	
ID	CU-9
Nombre	Modificar Valores de Textos Flotantes
Actor	Sistema
Descripción	El sistema es capaz de modificar el contenido de los textos flotantes obteniendo la entrada de unos scripts adjuntos al componente en cuestión.
Precondiciones	Existencia de un objeto de texto con un contenido previamente introducido. Además, este objeto dependerá de otros parámetros como el color o la fuente.

Secuencia	<p>1. De forma continuada, el script va modificando el contenido de los textos flotantes mediante valores generados aleatoriamente dentro de un rango.</p> <p>2. Cuando un usuario mira a un texto flotante, si espera unos segundos verá que el contenido cambia con el tiempo.</p>
Postcondición	El contenido del texto flotante ha cambiado.
Excepciones	Cuando se alcanza un estado anómalo, el color del texto flotante cambia y se activa una sección de los scripts que proporciona un comportamiento especial a dicho texto flotante.
Requisitos Asociados	RF02, RF03, RF04, RF07, RF08

Tabla 4.11 Tabla del caso de uso "CU-9"

Caso de Uso 10	
ID	CU-10
Nombre	Introducir Anomalía
Actor	Sistema
Descripción	Cada cierto tiempo, o bien cuando se alcanzan unos valores determinados de humedad, temperatura o velocidad del viento, se activa una variable que desbloquea una sección especial de algunos scripts. Esta sección modifica el estado de los objetos para mostrar que hay una anomalía en el sistema.
Precondiciones	Existencia de un objeto con un script adjunto que admita anomalías (no todos los elementos del modelo admiten estados anómalos).
Secuencia	<p>1. De forma continuada, el script va leyendo el contenido de los textos flotantes.</p> <p>2. Cuando el valor del texto supera un rango concreto, se activa una variable.</p> <p>3. Esta variable desbloquea una sección en el código de</p>

	ciertos scripts que provoca cambios en el estado de ciertos objetos (señales, colores, posición, etc.).
Postcondición	La variable de modificación del estado de los elementos ha sido activada.
Excepciones	
Requisitos Asociados	RF03, RF04, RF08

Tabla 4.12 Tabla del caso de uso "CU-10"

Caso de Uso 11	
ID	CU-11
Nombre	Modificar el Entorno tras una Anomalía
Actor	Sistema
Descripción	Cuando se ha introducido una anomalía en el sistema, se desbloquea una sección de código que produce modificaciones en ciertos elementos del entorno.
Precondiciones	Existencia de un objeto con un script adjunto que admita anomalías (no todos los elementos del modelo admiten estados anómalos).
Secuencia	<ol style="list-style-type: none"> 1. Los scripts de los objetos se van ejecutando de forma normal. 2. Cuando se detecta que una variable de anomalía ha sido activada, el código entra por una sección que produce modificaciones en el objeto. 3. Se activa una variable para indicar que la anomalía debe ser resuelta. 4. Se muestra en la vista una señal para indicar que la anomalía debe ser resuelta.

Postcondición	La variable de modificación del estado de los elementos sigue activada. Los objetos han sido modificados y se activa otra variable para indicar que la anomalía debe ser resuelta.
Excepciones	
Requisitos Asociados	RF03, RF04, RF08

Tabla 4.13 Tabla del caso de uso "CU-11"

Caso de Uso 12	
ID	CU-12
Nombre	Restaurar el Entorno tras Resolver una Anomalía
Actor	Sistema
Descripción	Una vez se ha modificado el entorno y se ha activado la variable que indica que la anomalía debe ser resuelta, aparece un icono con el que se podrá interactuar para resolver la anomalía.
Precondiciones	Existencia de un objeto con un script adjunto que admita anomalías (no todos los elementos del modelo admiten estados anómalos). Existencia de un icono para que el usuario interactúe con él y resuelva la anomalía.
Secuencia	<ol style="list-style-type: none"> 1. Los scripts se encuentran ejecutando la sección del código que corresponde al comportamiento anómalo. 2. Aparece un icono indicando el comportamiento anómalo. 3. El usuario puede interactuar con el icono para resolver la anomalía. 4. Al hacerlo, se activará una sección del código que hará que el sistema vaya recuperando la normalidad poco a poco. 5. Una vez se alcancen valores aceptables en los textos flotantes, se desactivarán las variables relacionadas con el comportamiento anómalo.

	6. Tras desactivar las variables, los textos flotantes y los iconos relacionados con anomalías desaparecerán.
Postcondición	La variable de modificación del estado de los elementos sigue activada. Los objetos han sido modificados y se activa otra variable para indicar que la anomalía debe ser resuelta.
Excepciones	
Requisitos Asociados	RF03, RF04, RF08

Tabla 4.14 Tabla del caso de uso "CU-12"

A continuación, tal y como se ha hecho para los casos de uso referentes al sistema, se procede a definir los casos de uso para el usuario recopilados en la figura 4.4: Diagrama de casos de uso del usuario en las tablas 4.15, 4.16, 4.17, 4.18, 4.19 y 4.20.

Caso de Uso 13	
ID	CU-13
Nombre	Moverse
Actor	Usuario
Descripción	El usuario podrá moverse por el sistema utilizando las teclas "W", "A", "S" y "D" de su teclado. Además, podrá cambiar la orientación de la vista (permitiendo así subir y bajar de altura) manteniendo pulsado el botón derecho del ratón y moviéndolo en la dirección deseada.
Precondiciones	Existencia de un objeto de tipo "cámara" que tenga un script que pueda leer entradas de teclado y de ratón. El script estará previamente configurado para aceptar las entradas enumeradas anteriormente.

Secuencia	<ol style="list-style-type: none"> 1. La cámara se encuentra en una posición inicial. 2. Cuando se pulsa "W", "A", "S" o "D", la cámara se mueve adelante, a la izquierda, atrás y a la derecha respectivamente. 3. Cuando se pulsa el botón derecho del ratón y se mueve, el campo de visión del usuario se moverá también en esa dirección. 4. El usuario puede utilizar estos controles para navegar por el entorno, acercarse a elementos e interactuar con ellos.
Postcondición	El usuario ha movido la cámara y/o ha modificado su campo de visión.
Excepciones	
Requisitos Asociados	RF01, RF02, RF03, RF04, RF05, RF09, RF10

Tabla 4.15 Tabla del caso de uso "CU-13"

Caso de Uso 14	
ID	CU-14
Nombre	Interaccionar con Objetos
Actor	Usuario
Descripción	El usuario será capaz de detectar con qué objetos puede interaccionar. El usuario podrá pulsar o acercarse a ciertos objetos. Estas acciones provocarán cambios en esos objetos o en otros que estén relacionados (por ejemplo, al resolver una anomalía al pulsar en un icono, se acabará modificando el estado del objeto que está sufriendo la anomalía).
Precondiciones	Existencia de un objeto de tipo "cámara" que tenga un script que pueda leer entradas de teclado y de ratón. El script estará previamente configurado para aceptar las entradas enumeradas anteriormente. Existencia de objetos que permitan interacción.

Secuencia	<ol style="list-style-type: none"> 1. El usuario intuye con qué objetos puede interaccionar. 2. El usuario elige uno y se acerca a él. 3. Dependiendo del objeto, el usuario interacciona con él o bien utilizando el ratón o bien acercándose a él. 4. El objeto modificará su comportamiento o el de otros objetos tras producirse la interacción.
Postcondición	Ciertos elementos del entorno han modificado su comportamiento.
Excepciones	
Requisitos Asociados	RF01, RF02, RF04, RF05, RF06, RF07

Tabla 4.16 Tabla del caso de uso "CU-14"

Caso de Uso 15	
ID	CU-15
Nombre	Abrir Puertas
Actor	Usuario
Descripción	El usuario localizará una puerta en un edificio. El usuario se acercará a la puerta. Si la puerta se abre, es una puerta con la que se puede interactuar y, si no, es solo parte del decorado. Si es una puerta con la que se puede interactuar, esta se abre por proximidad con la cámara del usuario. Cuando el usuario se aleje se cerrará.
Precondiciones	Existencia de un objeto de tipo "cámara" que tenga un script que pueda leer entradas de teclado y de ratón. El script estará previamente configurado para aceptar las entradas enumeradas anteriormente. Existencia de objetos de tipo "puerta" que permitan interacción. Existencia de scripts asociados a esos objetos de tipo puerta que detectan la colisión con la cámara.

Secuencia	<ol style="list-style-type: none"> 1. El usuario intuye con qué objetos de tipo "puerta" puede interactuar. 2. El usuario elige uno que permita interacción y se acerca a él. 3. Cuando el usuario está muy cerca, la puerta se abrirá. 4. Cuando el usuario se aleje, la puerta se cerrará.
Postcondición	Si el usuario no ha pasado el umbral de la puerta esta permanecerá abierta. Si el usuario ha pasado el umbral de la puerta, esta se cerrará.
Excepciones	Si el usuario realiza movimientos anómalos, como entrar y salir repetidas veces en un corto espacio de tiempo, la puerta podría quedarse abierta (aunque el usuario ya haya pasado) o cerrada (aunque el usuario se quede dentro del umbral).
Requisitos Asociados	RF01, RF05, RF08

Tabla 4.17 Tabla del caso de uso "CU-15"

Caso de Uso 16	
ID	CU-16
Nombre	Aumentar Zoom
Actor	Usuario
Descripción	El usuario debe intuir qué elementos son interactivos. Cuando el usuario pulsa sobre algún elemento interactivo que permita zoom, se cambiará a una cámara fija, simulando un efecto de "zoom" sobre ese objeto interactivo.
Precondiciones	Existencia de un objeto interactivo que tenga adjunto un script que permita la interacción de tipo "zoom". El script estará previamente configurado para aceptar las entradas enumeradas anteriormente. Existencia de cámaras que enfoquen estos objetos interactivos.

Secuencia	<ol style="list-style-type: none"> 1. El usuario intuye qué objetos permiten una interacción de tipo "zoom". 2. El usuario pulsa sobre dichos objetos. 3. Se produce un cambio de cámara a una cámara fija que enfoca el objeto.
Postcondición	La cámara principal ha cambiado a una cámara fija que enfoca el objeto o conjunto de objetos en cuestión.
Excepciones	
Requisitos Asociados	RF05, RF06, RF08, RF10

Tabla 4.18 Tabla del caso de uso "CU-16"

Caso de Uso 17	
ID	CU-17
Nombre	Disminuir Zoom
Actor	Usuario
Descripción	<p>Cuando el usuario pulsa sobre algún elemento interactivo que permita zoom, se cambiará a una cámara fija, simulando un efecto de "zoom" sobre ese objeto interactivo. El usuario podrá volver a la vista anterior pulsando el botón derecho del ratón.</p>
Precondiciones	<p>Existencia de un objeto interactivo que tenga adjunto un script que permita la interacción de tipo "zoom". El script estará previamente configurado para aceptar las entradas enumeradas anteriormente. Existencia de cámaras que enfoquen estos objetos interactivos.</p>
Secuencia	<ol style="list-style-type: none"> 1. El usuario ha cambiado a una cámara fija que enfoca un objeto interactivo. 2. El usuario pulsa el botón derecho del ratón. 3. El usuario vuelve a la cámara móvil principal.
Postcondición	La cámara principal ha vuelto a cambiar de la cámara fija a una cámara móvil.

Excepciones	
Requisitos Asociados	RF05, RF06, RF08, RF10

Tabla 4.19 Tabla del caso de uso "CU-17"

Caso de Uso 18	
ID	CU-18
Nombre	Resolver Anomalía
Actor	Usuario
Descripción	Cuando un usuario detecta una anomalía, podrá acercarse al icono que informa de la anomalía y tras pulsarlo, podrá iniciar la recuperación del sistema.
Precondiciones	Existencia de una anomalía en el sistema. Existencia de un objeto que permita ser modificado por el sistema tras introducir una anomalía. Existencia de un objeto que indique la existencia de una anomalía en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El usuario se da cuenta de que hay una anomalía. 2. El usuario pulsa en el icono que indica la existencia de una anomalía en el sistema. 3. El usuario ha iniciado la recuperación del sistema.
Postcondición	El sistema comienza a recuperarse y los valores vuelven a un rango aceptable de forma progresiva.
Excepciones	Durante la recuperación de una anomalía, puede aparecer otra anomalía en otro objeto diferente del sistema.
Requisitos Asociados	RF03, RF04, RF08

Tabla 4.20 Tabla del caso de uso "CU-18"

4.3.3.3 Matriz de Trazabilidad

CU/RE Q	RF01	RF02	RF03	RF04	RF05	RF06	RF07	RF08	RF09	RF10
CU01	x	x	x			x	x	x		
CU02	x		x		x		x	x	x	
CU03		x						x	x	x
CU04		x							x	x
CU05		x							x	x
CU06		x			x	x				
CU07		x	x	x			x	x		
CU08		x	x	x			x	x		
CU09		x	x	x			x	x		
CU10			x	x				x		
CU11			x	x				x		
CU12			x	x				x		
CU13	x	x	x	x	x				x	x
CU14	x	x		x	x	x	x			
CU15	x				x			x		
CU16					x	x		x		x
CU17					x	x		x		x
CU18			x	x				x		

Tabla 4.21 Matriz de trazabilidad de los casos de uso

4.3.4 Diagrama de clases

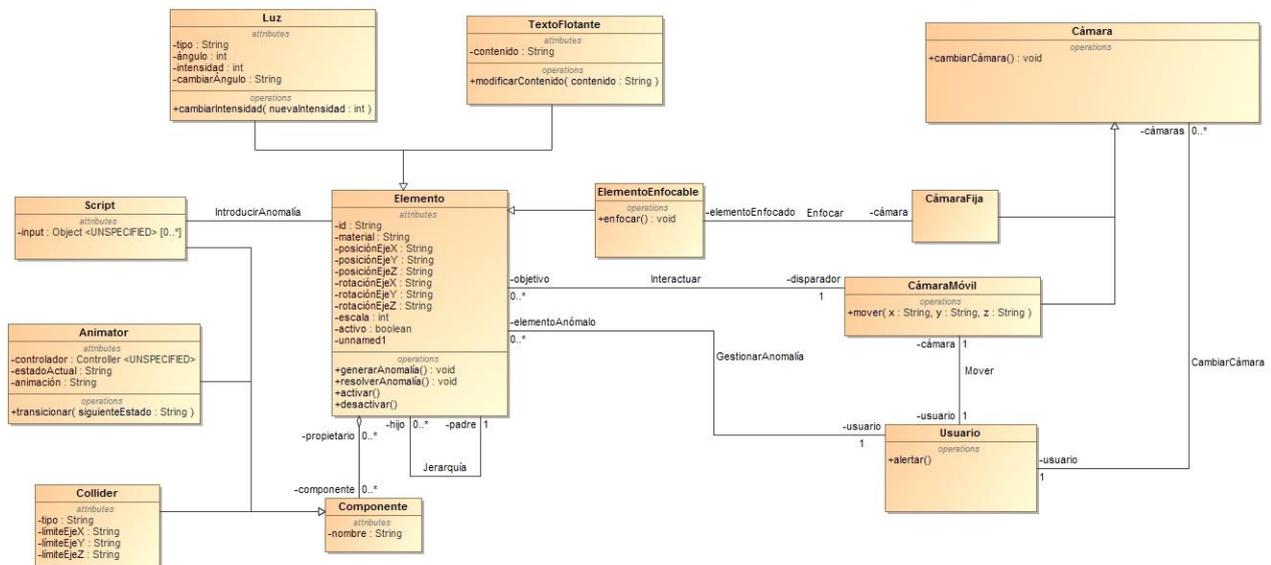


Figura 4.5 Diagrama de clases de HeatMap3D

Fuente: (HeatMap3D)

A continuación, describiremos brevemente las entidades del diagrama de clases. No se explicarán sus atributos ni sus operaciones por considerarlas triviales.

- **Usuario:** representa al usuario que va a navegar por nuestra simulación. Este usuario puede manejar una cámara utilizando el teclado y el ratón. Además, puede interactuar con elementos interactivos del entorno.
- **Elemento:** se trata de la unidad más simple del modelo, un objeto que se encuentra en el entorno y puede ser de diversos tipos:
 - **ElementoEnfocable:** este tipo de elementos son interactivos. El usuario puede pulsar sobre ellos para cambiar la cámara y conseguir un efecto “zoom”.

- **Luz:** representa a un objeto que emite luz.
Unity tiene algunos predefinidos, pero en esta ocasión hemos utilizado algunas librerías externas.
- **Texto Flotante:** es un objeto en el cual se puede mostrar una secuencia de caracteres para mostrar información relevante.
Este texto se puede modificar mediante scripts.
- **Cámara:** una cámara es un objeto predefinido de Unity que permite “ver” una vista u otra.
 - **Cámara Fija:** esta cámara no tiene ningún script asociado que permita movimiento. Se usan en los objetos interactivos que permiten “zoom”.
 - **Cámara Móvil:** esta cámara posee un script asociado que permite al usuario navegar por el entorno. Además, esta cámara puede activar una serie de animaciones en el modelo, como por ejemplo la apertura de puertas.
- **Componente:** los objetos poseen una serie de componentes que permiten añadir comportamiento adicional a los objetos tales como gravedad, detección de colisiones, animaciones, irradiar luz, etc. Se han definido algunos de los más utilizados en el proyecto.
 - **Script:** un script permite modificaciones prácticamente ilimitadas en un objeto. Permite literalmente reescribir sus propiedades, modificar su posición, sus características e incluso invocar a otros scripts de otros objetos, pudiendo modificar comportamientos de otros objetos.
 - **Animator:** este componente permite que un objeto transiciones de un estado a otro, modificando sus propiedades por el camino. El cambio de estado se conoce como “animación”.

- **Collider:** el componente "Collider" es de los más importantes puesto que permite dotar al objeto de consciencia física. Concretamente, permite definir el espacio en el cual el objeto colisionará con otros elementos del modelo.

4.3.5 Diagramas de interacción

En esta sección se muestran tres diagramas de interacción que ejemplifican tres flujos diferentes: generación de una anomalía, recuperación del sistema tras la aparición de una anomalía e interacción con una puerta.

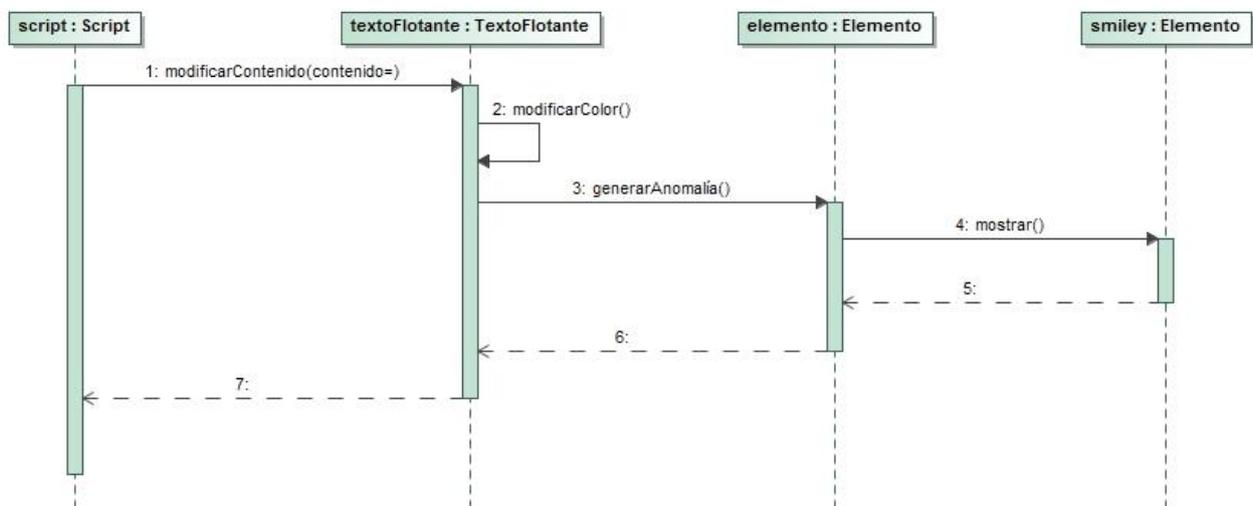


Figura 4.6. Diagrama de interacción para generar anomalías

Fuente: (HeatMap3D)

En primer lugar, en la *Figura 4.6*, se puede apreciar la secuencia que sigue el sistema para generar y notificar una anomalía. Primero, un script relacionado con un texto flotante, genera números aleatorios y modifica su contenido. Cuando este nuevo valor es anómalo, el texto pasa a ser de color rojo y se activa una variable

para indicar el estado anómalo. Es entonces cuando se activa un icono "smiley" o "cara sonriente" de color rojo que indica este nuevo estado del sistema.

En segundo lugar, en la *figura 4.7* se encuentra la interacción entre un usuario y un icono de color rojo. Cuando el usuario pulsa sobre el icono, este cambia y se vuelve de color naranja. Es entonces cuando el sistema comienza a recuperarse hasta alcanzar unos valores aceptables, dentro de los límites. Una vez alcanza estos valores, el icono cambia una vez más y se vuelve verde. El icono desaparecerá una vez el valor anómalo se recupere por completo adquiriendo un valor cercano a la media aritmética entre los límites superior e inferior.

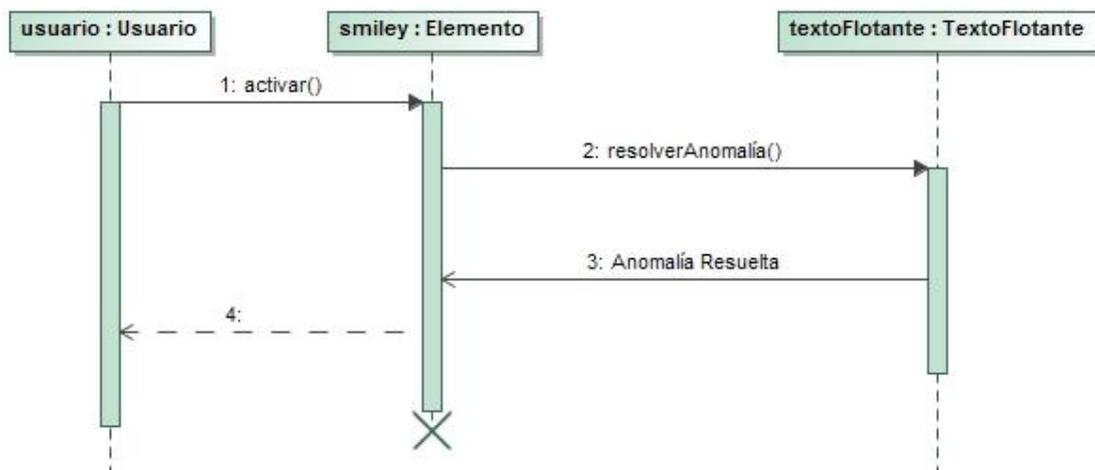


Figura 4.7. Diagrama de interacción para resolver anomalías

Fuente: (HeatMap3D)

Por último, en la *figura 4.8* se puede apreciar la interacción del usuario con una puerta interactiva. El usuario mueve la cámara y, cuando atraviesa el umbral de la puerta, la puerta detecta la colisión y cambia de estado (se

abre). Una vez el usuario abandona el umbral de la puerta, detecta la ausencia de colisión con la cámara y se cierra.

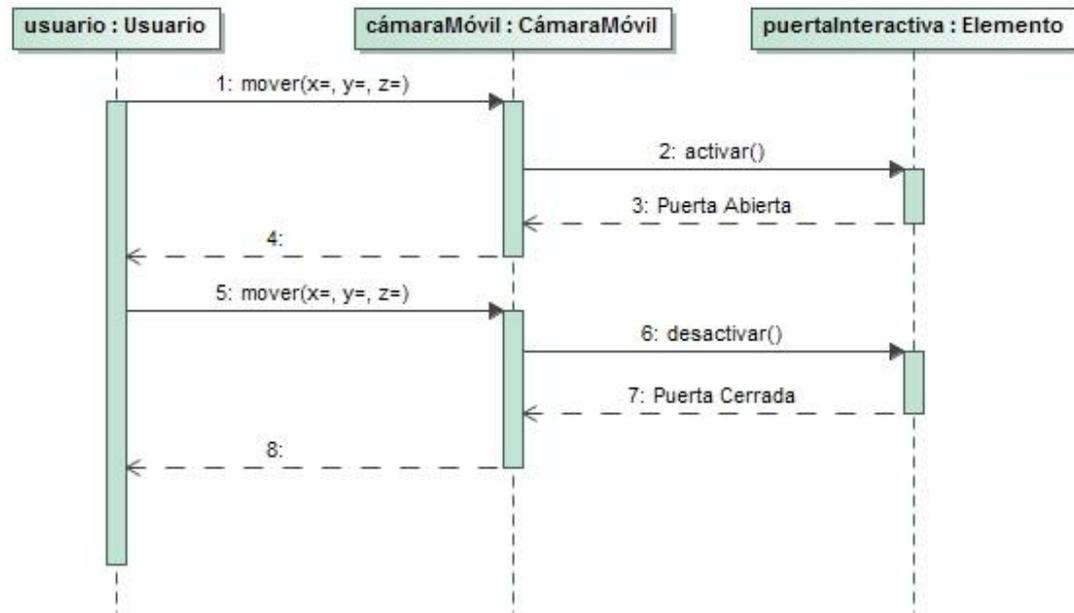


Figura 4.8. Diagrama de interacción entre un usuario y una puerta interactiva.

Fuente: (HeatMap3D)

4.3.6 Diagrama de estados

En este apartado se adjunta la *figura 4.9*, un diagrama de estados que refleja las transiciones entre iconos. Cabe destacar que en el siguiente diagrama se ha modelado una máquina de estados para una anomalía en una sola magnitud. El diagrama en el que interaccionan dos magnitudes se considera trivial puesto que es igual que el de la Figura 4.8 pero con la excepción de que un icono naranja (Smiley Naranja en la máquina de estados) puede volver de nuevo a convertirse en un icono rojo (Smiley Rojo en la máquina de estados) en caso de que la segunda anomalía se produzca después de que el usuario haya interactuado con el icono rojo para transformarlo en uno naranja e iniciar la recuperación del sistema. Una vez el sistema alcance un valor estable, fuera de rangos anómalos, este transicionará al icono verde (Smiley Verde en la máquina de estados) indicando que, aunque el sistema sigue recuperándose, se encuentra fuera de peligro.

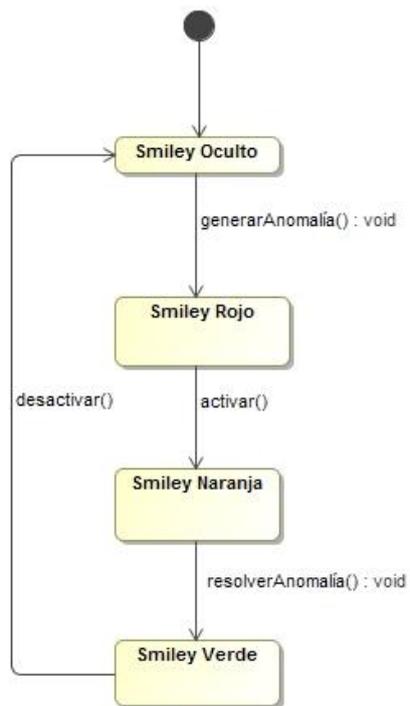


Figura 4.9. Máquina de estados de iconos

Fuente: (HeatMap3D)

5. Herramientas y técnicas utilizadas

5.1 El motor: Unity

5.1.1 Contexto

Unity es un motor de desarrollo 2D y 3D en tiempo real, en la *figura 5.1* podemos ver la cantidad de plataformas que soporta. Según sus creadores, se utiliza para "crear la mitad de los juegos del mundo" (Unity3D.com, 2019), algunos de los que han tenido más éxito en el mercado han sido Hearthstone de Blizzard, Super Mario Run de Nintendo o Pokémon GO de Niantic y Nintendo. Cuando se habla de tiempo real, no se exagera: el motor tiene la opción de "iniciar" el entorno modelado, como si de un vídeo se tratase, por tanto, no es necesario guardar y exportar el proyecto para luego abrirlo con un visualizador.

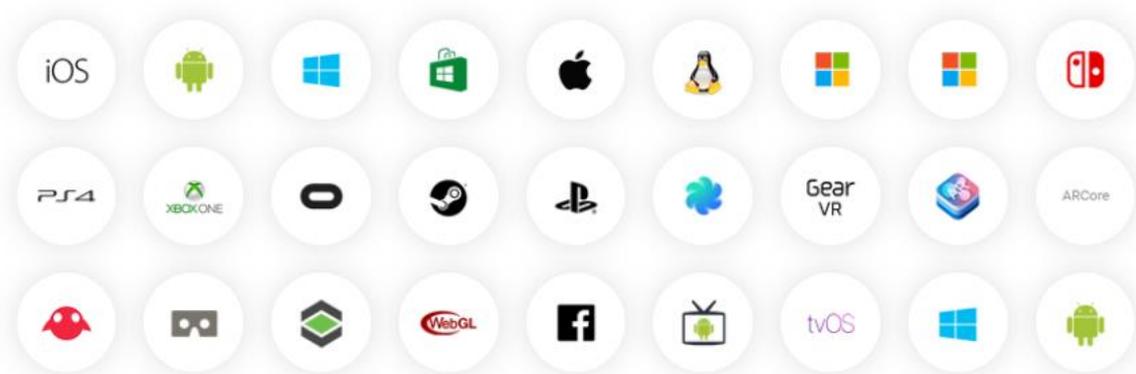


Figura 5.1. Plataformas soportadas

Fuente: (Unity3D.com, 2019)

Una de sus mayores fortalezas no es la plataforma de desarrollo en sí, sino toda la comunidad que la rodea. Son los propios usuarios los que se encargan por "*amor al arte*" de crear contenido, guías, tutoriales y también de responder las dudas de otros usuarios de la plataforma.

En cuanto a las posibilidades de uso, el programa cuenta con tres versiones. La primera se llama *Personal* y está pensada para uso "doméstico", usuarios que

no tengan grandes conocimientos y quieran aprender a utilizar el programa como hobby o pequeñas compañías (cuyos ingresos no superen los 100.000 euros al año). Por otro lado, tenemos la versión *Plus* que requiere el pago de una pequeña cuota mensual y está enfocada a desarrolladores con un conocimiento más amplio de la herramienta, ofreciéndole acceso a tutoriales y charlas "en vivo" con el equipo de Unity Engine. Las empresas no podrán superar los 200.000 euros en ingresos anuales.

Por último, encontramos la versión *Pro* que requiere también de un desembolso mensual, pero en esta ocasión la cantidad es bastante mayor. Está enfocada a grandes estudios y se les ofrece un servicio de asesoría, acceso exclusivo a diseños, chats "en vivo" con ingenieros de Unity Engine en los cuales podrán preguntarles para solucionar problemas.

5.1.2 La interfaz

Aunque la interfaz de Unity tiene numerosas posibilidades y configuraciones, vamos a nombrar las más utilizadas a la hora de desarrollar un proyecto, ayudándonos de la *figura 5.2*, una captura de pantalla de la interfaz.

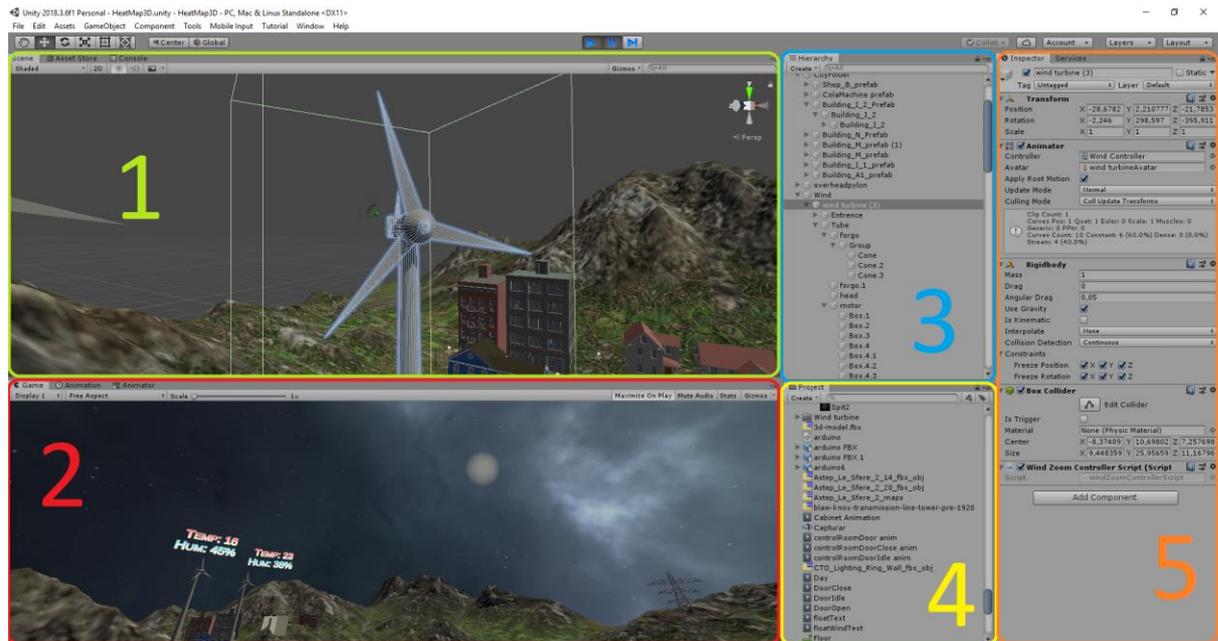


Figura 5.2. Interfaz del programa

Fuente: (HeatMap3D)

1. Escenario (Scene)

Se trata de la ventana “principal” por así decirlo. Al comenzar un proyecto no habrá nada, será una especie de “lienzo en blanco” en la que puedes ir colocando tus modelos de objetos en 2D o 3D, moverlos, rotarlos, cambiarlos de tamaño, etc.

Unity cuenta con un complejo sistema de cámaras (o vistas) que permiten libertad de movimiento por el entorno a la hora de desarrollar el proyecto.

2. Juego (Game)

Se podría decir que, si desarrollar un proyecto en Unity fuese como ver una escena de una película, podríamos colocar y mover a los actores en la pestaña “Scene”, así como darles “vida” mediante

configuraciones, animaciones en el tiempo, disparadores (triggers), etc. Siguiendo con el símil, la pestaña "Game" sería como "darle al play" en esa película, los actores empezarían a moverse por el entorno y sólo podríamos verlos desde las cámaras que hayamos designado previamente en la pestaña "Scene". Es en la pestaña "Game" cuando realmente podemos comprobar el comportamiento que tienen nuestros objetos en el entorno que hemos modelado.

3. Jerarquía (Hierarchy)

En esta ventana podemos ver los elementos que tenemos en nuestro modelo, aquellos que hemos situado en la pestaña "Scene" aunque pueden no ser visibles en ella (bien porque no tengan forma física o bien porque la hayamos ocultado). Podemos organizar los elementos de diversas formas, aunque siguen una jerarquía similar a la estructura normal de carpetas de un sistema operativo corriente. La jerarquía es de tipo "padre-hijo" y los efectos o modificaciones que se apliquen a un padre también afectarán a los nodos hijo que tenga, a los hijos de estos y así sucesivamente.

4. Proyecto (Project)

Aquí podemos encontrar todos los elementos que tenemos importados en nuestro proyecto. Es, por así decirlo, como un catálogo del que luego escogemos una serie de objetos, cero, una o muchas veces para colocar "copias" en la escena "Scene" y a su vez en la jerarquía "Hierarchy". Además, en el proyecto podemos encontrar otro tipo de elementos como animaciones, texturas o materiales que aplicaremos a nuestros objetos pero que no se verán reflejados en

"Hierarchy" aunque quizás sí en la escena "Scene" si se trata de modificaciones estéticas o en "Game" si se trata de animaciones. Este tipo de elementos que no son modelos 3D en sí, se pueden gestionar en la ventana "Inspector".

5. Inspector (Inspector)

Cada modelo colocado en el entorno tiene sus propias características, que al principio vienen definidas por el elemento del proyecto del que procedan. Una vez se arrastra un elemento del proyecto a la escena, queda reflejada en la jerarquía una nueva instancia de ese elemento, que puede ser configurada o modificada a placer. Algunas de estas modificaciones son complejas y requieren de componentes. Los componentes son una especie de paquetes de configuraciones que se aplican a los objetos del modelo. Por ejemplo, hay componentes relacionados con la gravedad, con las colisiones entre objetos, con las animaciones, etc.

5.1.3 Assets

Los assets de Unity son una especie de paquetes que contienen modelos "prefabricados" (o prefabs) listos para colocarlos en la escena. Para este proyecto hemos utilizado assets y modelos de tipo "**open source**" o de "código abierto". En futuras ampliaciones podría considerarse la viabilidad de invertir en ciertos modelos de estructuras para mejorar el rendimiento o hacer la simulación más realista. Los modelos utilizados en el proyecto provienen de cuatro fuentes principales,

ordenados de menor a mayor cantidad e impacto a la hora de desarrollar este proyecto:

- **Default assets:** Son los modelos que vienen preinstalados con Unity. Se trata de modelos en 2D y 3D muy sencillos por lo general, pero también hay paquetes que nos permiten generar **luces**, **terrenos** e incluso **agua**.
- **Asset store:** Plataforma oficial de Unity, accesible desde la propia aplicación, que nos permite buscar e importar contenido a nuestro proyecto, desde un simple fondo hasta un personaje animado totalmente funcional. El contenido de la tienda puede ser gratuito o de pago.
- **Fuentes externas:** Se han utilizado gran cantidad de **modelos de código abierto** de diversas webs externas a Unity, principalmente de *Turbosquid*, *CGTrader* y *GrabCad*.
- **Otros:** Ciertos objetos muy concretos se han **modelado para la ocasión** en el propio Unity, aunque también se ha utilizado Paint3D para modelar y pintar a mano algunos dispositivos electrónicos.

5.2 El código: Visual Studio Code

Aunque Unity ofrece posibilidades prácticamente ilimitadas a la hora de modelar un entorno en tres dimensiones, no cuenta con un compilador de código. En este proyecto se han desarrollado numerosos scripts, desde simples bucles

temporales para ir actualizando los mensajes mostrados en los letreros hasta complejos algoritmos para detectar colisiones entre objetos, modificar el comportamiento de un objeto en función de la posición de la cámara e incluso detectar esas "anomalías" que se introducen en nuestro sistema para intentar desestabilizarlo como es el caso del algoritmo de la *figura 5.3*.

```
119 |
120 |     if (issue)
121 |     {
122 |         issue = false;
123 |         shutdownScript.decrementIssueNumber();
124 |         Debug.Log("Humedad normal recuperada.");
125 |
126 |         if(tempScript.acceptableValue()) {
127 |             sadSmile.SetActive(false);
128 |             normalSmile.SetActive(false);
129 |             happySmile.SetActive(true);
130 |         }
131 |
132 |         txt.outlineColor = Color.white;
133 |         txt.color = savedColor;
134 |     }
135 |     else
136 |     {
137 |         if (rightValue())
138 |         { //Sistema recuperado
139 |             recover = false;
140 |             if (tempScript.rightValue()) {
141 |                 sadSmile.SetActive(false);
142 |                 normalSmile.SetActive(false);
143 |                 happySmile.SetActive(false);
144 |             }
145 |         }
146 |     }
```

Figura 5.3. Algoritmo de control de anomalías

Fuente: (HeatMap3D)

Entre los lenguajes propuestos por Unity Engine, se encuentran **C#**, JavaScript y Boo. Para este proyecto se ha decidido utilizar C# porque ya se tenía una base del mismo y por su gran parecido Java, lenguaje que se conoce bastante bien. Además, al plantear este proyecto, se presentó una oportunidad única para que se profundizase en este lenguaje, permitiendo ampliar el conocimiento que se tenía sobre el mismo y así poder elaborar un código más eficiente con cada script desarrollado. Por otro lado, la balanza se decantó por esta opción debido a la gran cantidad de documentación oficial existente y el soporte de la comunidad de desarrolladores, cosa que no se puede decir de los otros lenguajes. Tras este análisis, C# parecía la mejor opción para usuarios que no tenían conocimiento alguno de Unity.

6. Desarrollo del proyecto

6.1 Primeros pasos

Aunque el proyecto ha sufrido modificaciones durante su ciclo de vida, el núcleo del mismo ha permanecido intacto: desarrollar una simulación en tres dimensiones de una Smart Grid permitiendo monitorizar el sistema en tiempo real.

En un principio, se planteó realizar un modelo simple, poco realista pero centrado en cómo podría ser una implementación real. En las primeras conversaciones se acordó dar mucha importancia a los dispositivos que podríamos observar en un sistema de control hipotético, e introducir las anomalías en los componentes electrónicos.

Por otro lado, se imaginaba un proyecto que fuera verdaderamente vulnerable a los ataques informáticos, es decir, las anomalías no se generarían de forma "arbitraria" mediante contadores si no con simulaciones de "ataques" más elaborados. Este último punto acabó siendo descartado debido al alto coste de implementación que tendría y a la poca "utilidad" que proporcionaría teniendo en cuenta de que se trataba de una simulación (no tenía por qué ser real, sólo parecerlo para entender el modelo).

6.2 Modelado de dispositivos

Durante estos primeros compases se apostó por localizar y fabricar modelos fieles en tres dimensiones de dispositivos (figura 6.1) que se utilizarían en sistemas reales, por ejemplo: relés, sensores de corriente eléctrica y de temperatura, pasarelas o plataformas de desarrollo (Galileo, Raspberry Pi y Arduino). La búsqueda de estos primeros modelos por la web ayudó a localizar diferentes portales web dedicados a ofrecer modelos de código abierto, así como a aprender formas eficaces para encontrar los modelos deseados. Además, se constató un

problema que se tuvo en cuenta durante la planificación: *no había modelos de código abierto para algunas piezas que se necesitaban en un primer momento.*

Ante esta problemática, se tomaron tres decisiones:

1. Intentar sustituir los componentes por otros similares que pudieran utilizarse en un entorno real.
2. Intentar reproducir estos componentes de forma precisa, modelándolos "a mano".
3. Reducir el nivel de exigencia en cuanto a la fidelidad que debía tener el sistema.

Es necesario destacar que en la mayor parte de los casos se ha conseguido sustituir o reproducir a mano los modelos necesarios, sin embargo, ciertos componentes (como pasarelas, sensores inalámbricos como el TelosB o controladores modulares como el SIMANTIC S7-1200) tuvieron que ser simplificados mediante abstracciones más sencillas. Esta última decisión se tomó tras reflexionar mucho acerca de si realmente merecía la pena invertir más tiempo en desarrollar ese tipo de componentes y "descuidar" otras partes importantes del proyecto. Como el objetivo principal era elaborar una simulación, se llegó a la conclusión de que ciertas partes del sistema se podían simplificar mediante abstracciones.



Figura 6.1. Nombres de los dispositivos del banco de pruebas de la estación principal

Fuente: (HeatMap3D)

Por otra parte, la disposición de los elementos en el modelo también sufrió cambios con el tiempo. En un principio se pensó que la disposición física de los elementos no era tan relevante. Se decidió que los dispositivos electrónicos podrían estar colocados “dentro” de un contenedor, concretamente un armario industrial (figura 6.2) al cual un usuario podría acercarse e interactuar con los mismos. Finalmente se optó por disponer los elementos en la pared, en una especie de sistema de tuberías o “pipeline” en el cual los dispositivos utilizados para monitorizar un sistema real hipotético estuviesen colocados en un orden lógico.



Figura 6.2 Armario de la subestación principal cerrado y abierto. Fuente: (HeatMap3D, versión deprecada)

6.3 Modelado del entorno

Una vez se tenían localizados la mayor parte de dispositivos que se querían utilizar en el proyecto, se comenzó a buscar elementos para modelar el pequeño “mundo” en el que iban a situarse los elementos. En una primera aproximación se pensó en realizar un modelo muy simple: situar los componentes sobre un plano para poder visualizar los elementos interactuando a vista de pájaro, como si se mirara un diagrama. Esta idea se descartó rápidamente puesto que se empezó a priorizar por el usuario final, el que tendría que utilizar el modelo. Por ello, se tomó la determinación hacer el entorno lo más realista e intuitivo posible sin llegar a agotar los recursos computacionales disponibles.

Tras esta decisión, se emprendió la búsqueda de un terreno (o “terrain” en términos de Unity) que permitiera nuestro modelo sobre él. Se encontraron numerosos modelos posibles de terreno, con requerimientos diferentes y diversos

accidentes geográficos. Al final se optó por utilizar un diseño "realista" pero a su vez preparado para no consumir demasiados recursos. En este modelo de terreno se pueden encontrar diversas montañas, una cordillera que rodea un extremo del terreno y un valle. Estos accidentes geográficos se aprovecharon para estructurar las diferentes formas de generación de energía. Al poco tiempo de elegir el diseño se cayó en la cuenta de que este no tenía agua, lo que sin duda dificultaría el modelado de la central hidroeléctrica. En lugar de verlo como un problema, se vio como una oportunidad para aprender a modelar agua en Unity. En la *figura 6.3* puede apreciarse el resultado final.

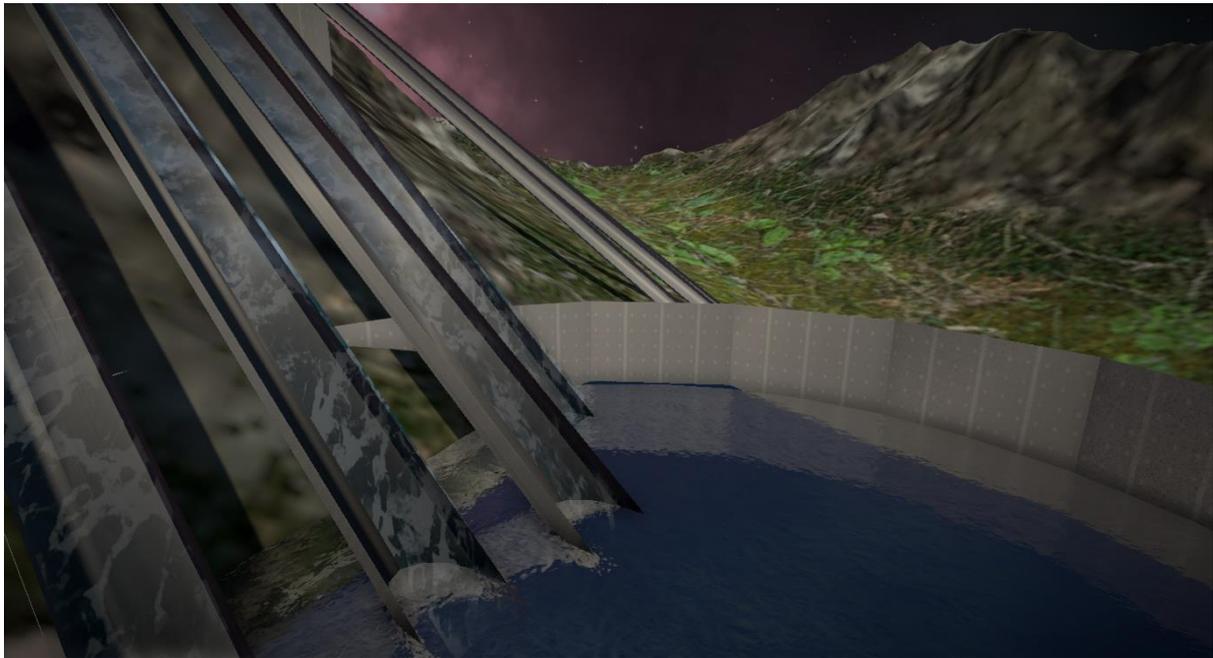


Figura 6.3. Tuberías de la estación hidroeléctrica.

Fuente: (HeatMap3D)

Por último, cabe destacar que dentro de esta agrupación que se ha denominado "entorno" también se encuentran ubicadas todas las estructuras necesarias en el proceso de generación de energía como los aerogeneradores (figura 6.4), los paneles solares y la central hidroeléctrica. También se consideraron

elementos del entorno todas las estaciones, subestaciones y torres eléctricas (o "pylons") modeladas para dar coherencia al sistema.



Figura 6.4. Aerogeneradores y textos flotantes.

Fuente: (HeatMap3D)

6.4 Modelado del comportamiento

Paralelamente, mientras se modelaban los dispositivos y el entorno, se comenzó a experimentar creando programas sencillos o "*scripts*" que implementaran algunas funcionalidades extra para satisfacer el caso de uso.

El primer script que se creó fue para poder manejar la cámara con el teclado, requisito fundamental para nuestro proyecto (sin esto el usuario no podría navegar por el entorno). Más tarde, se crearon un par de programas sencillos que permitían hacer un falso "zoom" a los objetos, aunque realmente era un cambio de cámara a una que estaba más cerca del objeto.

Uno de los hitos más importantes a la hora de mejorar la interactividad fue la implementación de un script que permitía a un objeto percibir la proximidad de la cámara del usuario y modificar su estado tras un "contacto". Esta última funcionalidad posibilitó el modelado de puertas que se abrían al paso del usuario que controlaba la cámara (*figuras 6.5 y 6.6*).



Figura 6.5. Puerta interactiva abierta y cerrada.
Fuente: (HeatMap3D)

```
public void TrigerInteraction()
{
    if (!GetComponent<Animation>().isPlaying)
    {
        Debug.Log("Interactive object");
        switch (m_State)
        {
            case eInteractiveState.Active:
                GetComponent<Animation>().Play("Close");
                m_State = eInteractiveState.Inactive;
                break;
            case eInteractiveState.Inactive:
                GetComponent<Animation>().Play("Open");
                m_State = eInteractiveState.Active;
                break;
            default:
                break;
        }
    }
}
```

Figura 6.6. Código para percibir colisiones.
Fuente: (HeatMap3D)

En último lugar, se desarrollaron algunos scripts más para la monitorización y la gestión de anomalías. Por un lado, se crearon algunos que modificaban los valores de los textos flotantes como por ejemplo temperatura, humedad o velocidad del viento. Además, estas funciones comprobaban que dichos valores no rebasaran un límite y si lo hacían, activaban un parámetro que avisaba a otros scripts para iniciar la simulación de anomalías. Por otro lado, quedaba definir el propio comportamiento anómalo, es por eso que se desarrollaron secciones de código que comprobaban cada cierto tiempo si había una anomalía en el sistema y, si la encontraban, realizaban modificaciones en otros objetos del entorno como por ejemplo apagar algunas luces, parar la animación del aerogenerador o activar

los iconos que avisaban de la anomalía. Finalmente es necesario mencionar el papel de los scripts que “resolvían” las anomalías tras interactuar con los iconos, restaurando progresivamente los valores de los textos flotantes y devolviendo a los objetos modificados a su estado normal.

6.5 Dificultades durante el desarrollo

Como ya se ha mencionado en los puntos anteriores, la dificultad más evidente nada más comenzar fue el total desconocimiento sobre el entorno de modelado Unity. Este primer obstáculo, se solventó con muchas horas de lectura y aprendizaje sobre el entorno.

Una vez se consiguió un dominio aceptable del entorno, se descubrió una segunda dificultad: el proyecto podría ser bastante más complejo de desarrollar de lo que se había planificado. Esto es porque en un principio se pensó que la mayoría de dispositivos electrónicos necesarios para el modelo se podrían encontrar en páginas web de código abierto, pero no fue así. Se planteó la posibilidad de simplificar la funcionalidad del proyecto a costa de representar esos dispositivos de forma más realista (modelando piezas muy complejas a mano, como la mostrada en la *figura 6.7*). Al final se optó por simplificar las pocas piezas, y crear abstracciones de aquellas cuyo coste de diseño fuera demasiado grande e invertir ese tiempo en mejorar la interacción con el usuario, el realismo y las funcionalidades del entorno.

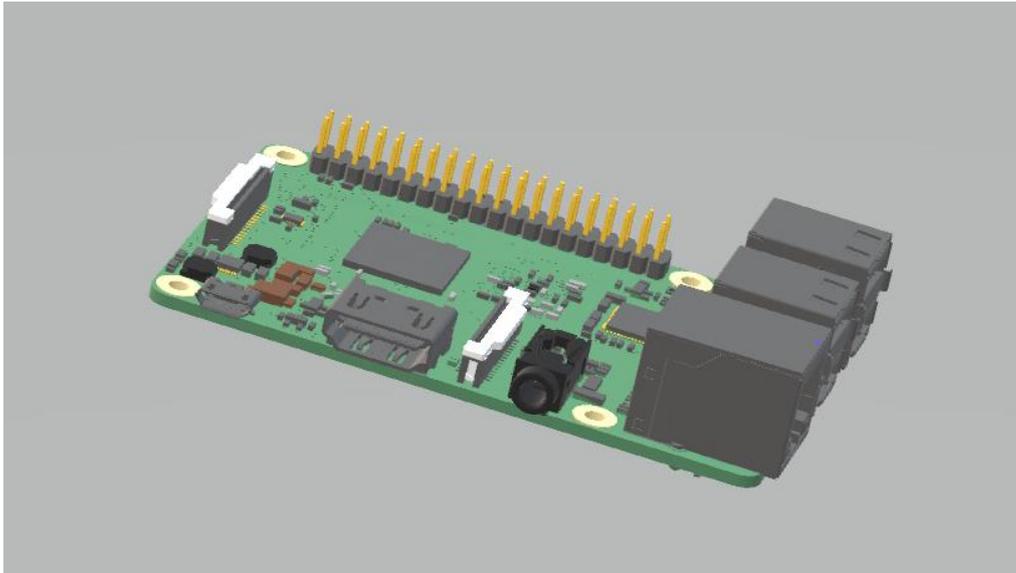


Figura 6.7. Raspberry Pi 3 pintada a mano.

Fuente: (Original sin pintar encontrado en CGTrader, Autor: Alexandre-Willame)

Tras una época de aparente calma, en la cual se incluyeron una cantidad respetable de modelos y se mejoraron los aspectos estéticos del entorno, se detectó el último problema: los requisitos computacionales del proyecto comenzaban a sobrepasar la capacidad de la máquina en la que se estaba desarrollando. Esto se debió a que lo que en un principio iba a ser una especie de diagrama a vista de pájaro de una red eléctrica inteligente, acabó convirtiéndose en una simulación en primera persona de un entorno muy realista (ver *figura 6.8*), con numerosos scripts que se ejecutaban en segundo plano y que además permitía al usuario interactuar con modelos. Tras numerosas reuniones, al final se optó por reducir y simplificar el número de elementos restantes por incluir en el proyecto, lo que reduciría el riesgo de volver a esta situación.

Cabe destacar que se trabajó en reducir el consumo computacional del entorno, eliminando la iluminación en ciertos casos y reduciendo la calidad de ciertos modelos y texturas. En cuanto a las estructuras que quedaban por modelarse, como, por ejemplo: la central hidroeléctrica y las subestaciones. En este

caso, se modelaron de la forma más simple posible, abstrayendo el contenido por considerarlo poco relevante para nuestra simulación. Otra opción para reducir el consumo que se barajó inicialmente fue sustituir los modelos por otros denominados "*lowpoly*", es decir, modelos muy simples y toscos que alivian la carga computacional. Esta idea se descartó porque requería modificar la mayoría de objetos del entorno debido al avanzado estado en el que se encontraba, y se concluyó que no sería necesario puesto que se había acordado reducir el número de elementos que quedaban por incorporar.

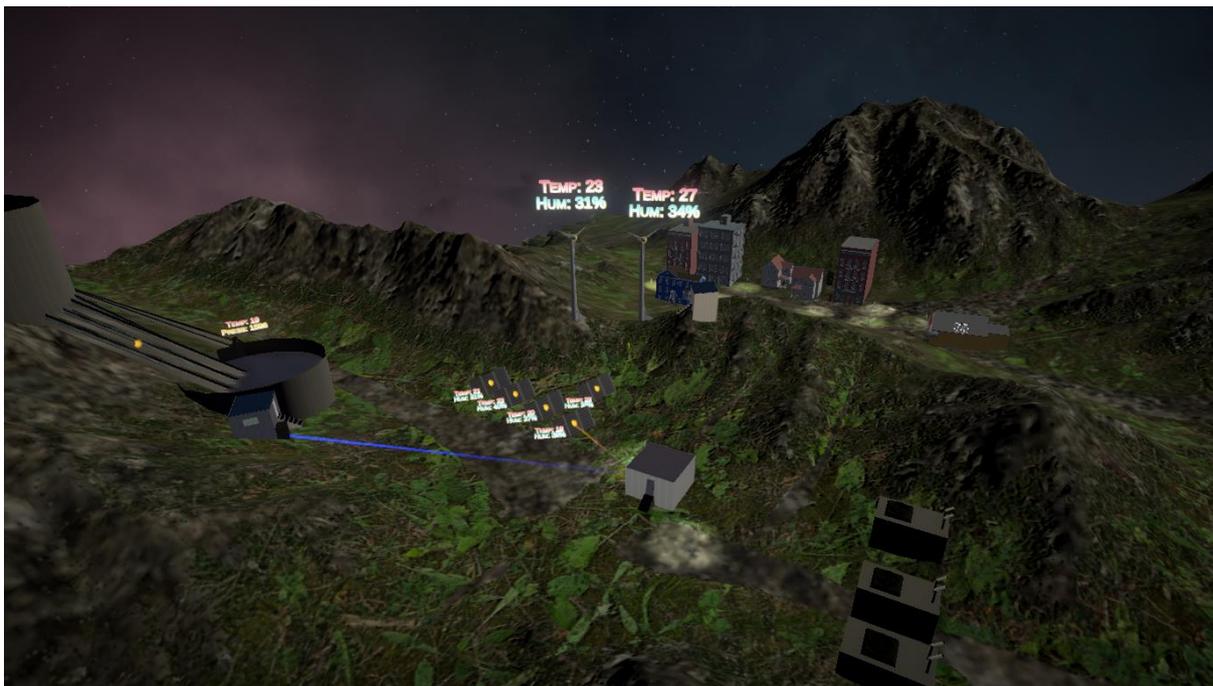


Figura 6.8. Vista del entorno del proyecto.

Fuente: (HeatMap3D, versión final)

7. Conclusiones y futuro

7.1 Conclusión

En este capítulo se van a enunciar las conclusiones a las que se ha llegado tras desarrollar el presente proyecto, cuyo objetivo principal ha sido la de desarrollar una herramienta para monitorizar una red eléctrica inteligente o Smart Grid mediante el uso de realidad virtual. Desde un principio se buscó la necesidad de modelar un entorno intuitivo y con una curva de aprendizaje suave. Además, se deseaba mostrar información relevante de forma sencilla y simulando que se obtenía en tiempo real.

En primer lugar, este trabajo me ha resultado todo un desafío puesto que desconocía completamente el entorno de modelado Unity y tampoco tenía conocimientos acerca de las redes eléctricas inteligentes ni de los dispositivos electrónicos utilizados para monitorizar este tipo de sistemas. Adquirir los conocimientos necesarios sobre el dominio del proyecto ha sido sin duda un gran reto que he afrontado con ilusión y con la motivación de que este tipo de aprendizaje autónomo me serviría de experiencia para futuros proyectos en los que tampoco controle las tecnologías necesarias y necesite adaptarme rápidamente. Además, he podido participar en el desarrollo de un proyecto viéndolo desde un punto de vista de alto nivel, utilizando dispositivos electrónicos cuyo funcionamiento comprendía de una forma básica, pero sin necesidad de tener conocimientos muy extensos sobre la materia, lo que me ha permitido centrarme en el ámbito de la simulación, que era el objetivo principal de este proyecto.

En segundo lugar, el ámbito del modelado en tres dimensiones siempre me ha llamado mucho la atención. He descubierto que hay todo un mundo de posibilidades más allá del tradicional diseño de videojuegos y que el campo de la simulación se encuentra cada vez más ligado a los procesos industriales.

Por último, creo que algo que me llevo de este proyecto es el hecho de haber sabido superar las dificultades que he encontrado por el camino de forma autónoma. En mi opinión he mejorado en mi forma de analizar y tomar las decisiones, entendiendo que tenía libertad de elección, pero también la responsabilidad de tomar la mejor opción para mi proyecto, sin tener miedo a descartar ideas anteriores.

7.2 Valoración personal y futuro

En cuanto a la herramienta, he de decir que ha superado mis expectativas. Desde mi punto de vista HeatMap3D es la demostración de que es posible implementar simulaciones en la industria energética para monitorizar sus redes de una forma intuitiva y realista.

Se trata de un proyecto pionero en el ámbito de la realidad virtual aplicada a la industria. Futuras ampliaciones permitirían la validación del estado de un sistema desde cualquier parte del mundo, así como la revisión de sus componentes y procedimientos. Además, la herramienta facilitaría la formación del personal necesario para monitorizar los sistemas. Por otro lado, este trabajo también podría servir en el ámbito educativo, permitiendo

a los estudiantes comprender y asimilar más fácilmente los conceptos relativos a la transmisión de energía mediante la inmersión en la simulación.

También destacaría el impacto económico que podría tener la utilización de este tipo de tecnologías en la industria energética española, permitiendo optimizar la producción de energías renovables aprovechando las condiciones climáticas y geográficas de nuestro país. Finalmente, destacaría el impacto medioambiental que podría tener la utilización de esta herramienta, permitiendo monitorizar de forma más precisa el entorno y, quizás en futuras ampliaciones, permitiría simular las condiciones geográficas idóneas para cada una de las formas de generación de energía.

8. Bibliografía y referencias

8.1 Bibliografía y referencias

- Simulators in Corporate Training. (n.d.). Retrieved June 9, 2019, from <https://www.game-learn.com/simulators-in-corporate-training/>
- El Big Data en los RRHH | Randstad. (n.d.). Retrieved June 9, 2019, from <https://www.randstad.es/tendencias360/el-big-data-en-los-recursos-humanos/>
- La tecnología va a cambiar los entornos de trabajo: ¿estás dispuesto a adaptarte? (n.d.). Retrieved June 9, 2019, from <https://www.20minutos.es/noticia/2981349/0/tecnologia-cambia-entorno-trabajo-empleados-dispuestos-adaptarse/>
- Objetivos de reducción de emisiones de gases de efecto invernadero. (n.d.). Retrieved June 9, 2019, from <https://www.miteco.gob.es/es/cambio-climatico/temas/mitigacion-politicas-y-medidas/objetivos.aspx>
- Smart Grid - Jaime Boal. (n.d.). Retrieved June 9, 2019, from <https://es.scribd.com/document/232217312/Smart-Grid-Jaime-Boal>
- Telefónica velará por la seguridad de Internet de las Cosas con una nueva unidad de ciberseguridad IoT | Detalle | Noticias | Sala de Prensa | Telefónica. (n.d.). Retrieved June 9, 2019, from <https://www.telefonica.com/es/web/sala-de-prensa/-/telefonica-velara-por-la-seguridad-de-internet-de-las-cosas-con-una-nueva-unidad-de-ciberseguridad-iot>
- AI Residency – Google AI. (n.d.). Retrieved June 9, 2019, from <https://ai.google/research/join-us/ai-residency/>
- Inteligencia artificial en AWS Marketplace. (n.d.). Retrieved June 9, 2019, from <https://aws.amazon.com/es/mp/ai/>
- What's At Stake In The Race To Industry 4.0? (n.d.). Retrieved June 9, 2019, from <https://www.forbes.com/sites/willemsundbladeurope/2018/07/30/whats-at-stake-in-the-race-to-industry-4-0/#70be05be7d11>
- Aloul, F., Al-Ali, A. R., Al-Dalky, R., Al-Mardini, M., & El-Hajj, W. (2012). Smart grid security: Threats, vulnerabilities and solutions. *International Journal of Smart Grid and Clean Energy*, 1(1), 1–6.
- Amin, S. M., & Wollenberg, B. F. (2005). Toward a smart grid: power delivery for the 21st century. *IEEE Power and Energy Magazine*, 3(5), 34–41.
- Amin, S. M., & Wollenberg, B. F. (2005). Toward a smart grid: power delivery for the 21st century. *IEEE Power and Energy Magazine*, 3(5), 34–41.
- Andrade, C. A. D., & Hernández, J. C. D. (2011). *Smart Grid: Las TICs y la modernización de las redes de energía eléctrica–Estado del Arte*.

- Aurich, J. C., Biermann, D., Blum, H., Brecher, C., Carstensen, C., Denkena, B., ... Weinert, K. (2009). Modelling and simulation of process: machine interaction in grinding. *Production Engineering*, 3(1), 111–120.
- Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4), 355–385.
- Beck, K., & Andres, C. (2005). *Extreme programming explained: embrace change*. Retrieved from <https://www.slideshare.net/DamianGordon1/the-extreme-programming-xp-model>
- Boy, G. (2014, July 22). *Introduction Handbook HMI*.
- Cameron, R. (1982). The industrial revolution: A misnomer. *The History Teacher*, 15(3), 377–384.
- Cascajo Sastre, M. (n.d.). Grandes avances españoles en IoT - Think Big. Retrieved June 9, 2019, from <https://empresas.blogthinkbig.com/grandes-avances-espanoles-en-iot/>
- Chaves Palacios, J. (2004). Desarrollo tecnológico en la Primera Revolución Industrial. *Norba. Revista de Historia, ISSN 0213-375X, N° 17, 2004, Págs. 93-109, 17(17), 93–109*.
- Chen, B., Wan, J., Shu, L., Li, P., Mukherjee, M., & Yin, B. (2017). Smart factory of industry 4.0: Key technologies, application case, and challenges. *IEEE Access*, 6, 6505–6519.
- Deane, P. M. (1979). *The first industrial revolution*. Cambridge University Press.
- Decker, M., Fischer, M., & Ott, I. (2017). Service Robotics and Human Labor: A first technology assessment of substitution and cooperation. *Robotics and Autonomous Systems*, 87, 348–354. <https://doi.org/10.1016/J.ROBOT.2016.09.017>
- Dede, C., Salzman, M. C., & Loftin, R. B. (1996). ScienceSpace: Virtual realities for learning complex and abstract scientific concepts. *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*, 246–252. IEEE.
- Fang, X., Misra, S., Xue, G., & Yang, D. (2011). Smart grid—The new and improved power grid: A survey. *IEEE Communications Surveys & Tutorials*, 14(4), 944–980.
- Fernández, L. E. Í. (2012). *Breve historia de la Revolución industrial*. Retrieved from <https://books.google.es/books?id=aiQfAgAAQBAJ>
- Fernández, M. (2017). El futuro de la educación: el 65% de no-sé-quién va a hacer no-sé-qué. Retrieved June 9, 2019, from El país website: https://elpais.com/elpais/2017/03/10/hechos/1489146364_790212.html
- Galindo, J. (n.d.). Máquinas y robots nos quitan el empleo, pero mejoran nuestras vidas | Tecnología | EL PAÍS. Retrieved June 9, 2019, from https://elpais.com/tecnologia/2017/06/21/actualidad/1498036655_438636.html

-
- Gershenfeld, N. (2012). How to make almost anything: The digital fabrication revolution. *Foreign Aff.*, 91, 43.
- Karlberg, M., Löfstrand, M., Sandberg, S., & Lundin, M. (2013). State of the art in simulation-driven design. *Article in International Journal of Product Development*, 18(1), 68–87. <https://doi.org/10.1504/IJPD.2013.052166>
- Kolberg, D., & Zühlke, D. (2015). Lean automation enabled by industry 4.0 technologies. *IFAC-PapersOnLine*, 48(3), 1870–1875.
- Landes, D. S. (1979). *Progreso tecnológico y revolución industrial*. Retrieved from <https://books.google.es/books?id=ZRAhOgAACAAJ>
- Lee, J., Bagheri, B., & Kao, H.-A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3, 18–23.
- Lee, K. (2012). Augmented reality in education and training. *TechTrends*, 56(2), 13–21.
- Littler, C. R. (1978). Understanding taylorism. *British Journal of Sociology*, 185–202.
- Marsh, P. (2012). *The new industrial revolution: consumers, globalization and the end of mass production*. Yale University Press.
- McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D. J., & Barton, D. (2012). Big data: the management revolution. *Harvard Business Review*, 90(10), 60–68.
- Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing*.
- Orborne, C. (n.d.). Most companies take over six months to detect data breaches | ZDNet. Retrieved June 9, 2019, from <https://www.zdnet.com/article/businesses-take-over-six-months-to-detect-data-breaches/>
- Rodella, F. (n.d.). Ya hay algoritmos ayudando a seleccionar personal: la búsqueda de trabajo en la era del algoritmo. Retrieved June 9, 2019, from <https://www.xataka.com/empresas-y-economia/hay-algoritmos-ayudando-a-seleccionar-personal-busqueda-trabajo-era-algoritmo>
- Roel, V. (1998). *La tercera revolución industrial y la era del conocimiento*. Retrieved from <https://books.google.es/books?id=o9j37L4-IEsC>
- Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., & Harnisch, M. (2015). Industry 4.0: The future of productivity and growth in manufacturing industries. *Boston Consulting Group*, 9(1), 54–89.
- Saleem, W., Yuqing, F., & Yunqiao, W. (2008). Application of Topology Optimization and Manufacturing Simulations-A new trend in design of Aircraft components. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2. Citeseer.

- Torrent i Sellens, J. (2002). *De la Nueva economía a la economía del conocimiento : hacia la tercera revolución industrial*. Retrieved from <http://hdl.handle.net/10272/418>
- Uhlemann, T. H.-J., Lehmann, C., & Steinhilper, R. (2017). The digital twin: Realizing the cyber-physical production system for industry 4.0. *Procedia Cirp*, 61, 335–340.
- Walton, R. (n.d.). Elevation: Drones rising as tools to improve power plant performance | Utility Dive. Retrieved June 9, 2019, from <https://www.utilitydive.com/news/elevation-drones-rising-as-tools-to-improve-power-plant-performance/530725/>

8.2 Librerías de código abierto utilizadas

- 3DHaupt. (n.d.). 3D model Animated Industrial Fan Low Poly | CGTrader. Retrieved June 11, 2019, from <https://www.cgtrader.com/free-3d-models/exterior/industrial/animated-industrial-fan-low-poly>
- 3d-rina. (n.d.). Wall TVset 3D model | CGTrader. Retrieved June 11, 2019, from <https://www.cgtrader.com/free-3d-models/electronics/video/wall-tvset>
- 400M Creations. (n.d.). Realistic Terrain Collection LITE. Retrieved June 11, 2019, from <https://assetstore.unity.com/packages/3d/environments/landscapes/realistic-terrain-collection-lite-47726>
- ADVENTURE FORGE. (n.d.). Industrial Cabinets and Shelves. Retrieved June 11, 2019, from <https://assetstore.unity.com/packages/3d/props/industrial/industrial-cabinets-and-shelves-55411>
- alexbes. (n.d.). 3D asset Game Ready Japanese Signs Set Sign Neon 24H. Retrieved June 11, 2019, from <https://www.cgtrader.com/free-3d-models/exterior/street/game-ready-japanese-signs-set-sign-neon-24h>
- benzin. (n.d.). Monitor Modelo 3D gratis - .3ds .obj .blend .fbx .stl - Free3D. Retrieved June 11, 2019, from <https://free3d.com/3d-model/monitor-79124.html>
- c-brenner. (n.d.). Realistic Doors glass 3D | CGTrader. Retrieved June 11, 2019, from <https://www.cgtrader.com/free-3d-models/interior/house/realistic-doors-glass>
- Chandrasekaran Ellappan. (n.d.). Roof AC | 3D CAD Model Library | GrabCAD. Retrieved June 11, 2019, from <https://grabcad.com/library/roof-ac-1>
- CLOD. (n.d.). Skybox. Retrieved June 11, 2019, from <https://assetstore.unity.com/packages/2d/textures-materials/sky/skybox-4183>
- designconnected. (n.d.). 3D XZ3 200 | CGTrader. Retrieved June 11, 2019, from <https://www.cgtrader.com/free-3d-models/furniture/table/xz3-200>

-
- Forester. (n.d.). Medium Arc Water Sheet Flow - OBJ Version 3D model. Retrieved June 11, 2019, from <https://www.cgtrader.com/free-3d-models/architectural/decoration/medium-arc-water-sheetflow>
- kabeltelevizio. (n.d.). free wind turbine 3d model. Retrieved June 11, 2019, from <https://www.turbosquid.com/3d-models/free-wind-turbine-3d-model/642388>
- luis88luis. (n.d.). Wireless mouse 3D model | CGTrader. Retrieved June 11, 2019, from <https://www.cgtrader.com/free-3d-models/electronics/computer/wireless-mouse-21c6a742-9028-4324-b52e-4a517fe75224>
- michael-mihalyfi. (n.d.). Solar Photovoltaic Panel 3D model | CGTrader. Retrieved June 11, 2019, from <https://www.cgtrader.com/free-3d-models/industrial/other/solar-photovoltaic-panel-354567da9c35a23e001dc6e7635fad68>
- Monterde, D. (n.d.). Arduino 3D model | CGTrader. Retrieved June 11, 2019, from <https://www.cgtrader.com/free-3d-models/industrial/other/arduino>
- NIGHTSOUNDGAMES. (n.d.). Starfield Skybox. Retrieved June 11, 2019, from <https://assetstore.unity.com/packages/2d/textures-materials/sky/starfield-skybox-92717>
- nikiyani. (n.d.). Neon sign PBR 3D model | CGTrader. Retrieved June 11, 2019, from <https://www.cgtrader.com/free-3d-models/exterior/street/neon-sign-pbr>
- POLYGON LAND. (n.d.). Simple City pack plain. Retrieved June 11, 2019, from <https://assetstore.unity.com/packages/3d/environments/urban/simple-city-pack-plain-100348>
- Shevkar, O. (n.d.). Keyboard with TrackPad 3D model | CGTrader. Retrieved June 11, 2019, from <https://www.cgtrader.com/free-3d-models/electronics/computer/keyboard-with-trackpad>
- silverpak. (n.d.). 3D Dining tables | CGTrader. Retrieved June 11, 2019, from <https://www.cgtrader.com/3d-models/furniture/cabinet/dining-tables>
- speirso. (n.d.). 3d model pylon. Retrieved June 11, 2019, from <https://www.turbosquid.com/3d-models/3d-model-ylon/519112>
- tridem. (n.d.). 3D smiling faces | CGTrader. Retrieved June 11, 2019, from <https://www.cgtrader.com/free-3d-models/character/other/smiling-faces>
- UNITY TECHNOLOGIES. (n.d.). Post Processing Stack. Retrieved June 11, 2019, from <https://assetstore.unity.com/packages/essentials/post-processing-stack-83912>
- UNITY TECHNOLOGIES. (n.d.). Probuilder 2.x. Retrieved June 11, 2019, from <https://assetstore.unity.com/packages/tools/modeling/probuilder-2-x-111418>
- UNITY TECHNOLOGIES. (n.d.). TextMesh Pro. Retrieved June 11, 2019, from <https://assetstore.unity.com/packages/essentials/beta-projects/textmesh-pro-84126>

Willame, A. (n.d.). Raspberry Pi 3 Reference Design Model B Rpi Raspberrypi free 3D model STL SLDPRT SLDASM SLDDRW IGE IGS IGES. Retrieved June 11, 2019, from <https://www.cgtrader.com/free-3d-models/electronics/computer/raspberry-pi-3-reference-design-model-b-rpi-raspberrypi>

