

UNIVERSIDAD DE VALLADOLID



E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE
TELECOMUNICACIÓN

MENCIÓN EN TELEMÁTICA

Diseño y Desarrollo de un sistema HMI para una Aplicación de Industria 4.0

Autor:

Manuel Corral González

Tutores:

Javier Manuel Aguiar Pérez

María Ángeles Pérez Juárez

Valladolid, septiembre de 2021

TÍTULO: **Diseño y Desarrollo de un sistema HMI para una Aplicación de Industria 4.0**

AUTOR: **D. Manuel Corral González**

TUTORES: **Dr. Javier Manuel Aguiar Pérez**
Dra. María Ángeles Pérez Juárez

DEPARTAMENTO: **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

TRIBUNAL

PRESIDENTE: **María Ángeles Pérez Juárez**

SECRETARIO: **Javier Manuel Aguiar Pérez**

VOCAL: **David González Ortega**

SUPLENTE 1: **Miriam Antón Rodríguez**

SUPLENTE 2: **Jaime Gómez Gil**

FECHA: **Septiembre 2021**

CALIFICACIÓN:

Resumen

En este Trabajo Fin de Grado se pretende llevar a cabo el diseño y desarrollo de un sistema HMI para una aplicación concreta de Industria 4.0, haciendo uso de tecnologías web, y conforme los requisitos funcionales y no funcionales solicitados por un cliente. Un sistema HMI actúa como interfaz entre un proceso industrial o de fabricación y el operario, en un contexto de Industria 4.0.

Antes de proceder al diseño e implementación de la solución propuesta se realizará un análisis exhaustivo de sistemas HMI para Industria 4.0, tanto de los basados en tecnologías web, como de los basados en otras tecnologías.

Palabras clave:

HMI, SCADA, Web, Angular, Angular Material, IoT, Industria 4.0.

Abstract

The purpose of this Final Degree Project is to carry out the design and development of a HMI system for a specific Industry 4.0 application, using web technologies, and according to the functional and non-functional requirements requested by a client. An HMI system acts as an interface between an industrial or manufacturing process and the user, in an Industry 4.0 context.

Before proceeding to the design and implementation of the proposed solution, an exhaustive analysis of HMI systems for Industry 4.0 will be carried out, both those based on web technologies and those based on other technologies.

Key words:

HMI, SCADA, Web, Angular, Angular Material, IoT, Industria 4.0.

Índice de Contenido

1. Introducción.....	11
1.1. Introducción.....	11
1.2. Motivación.....	12
1.3. Objetivos.....	13
1.4. Fases.....	14
1.5. Estructura de la memoria.....	16
2. Estado del arte.....	18
2.1. Uso de la web.....	18
2.2. Desarrollo de aplicaciones web.....	19
2.3. Opciones para SCADA/HMI.....	20
2.3.1. Open Automation Software.....	20
2.3.2. Ecava IntegraXor.....	21
2.3.3. QuickHMI.....	22
2.3.4. Pvbrowser.....	23
2.3.5. CuteHMI.....	23
2.3.6. WebIQ - Smart HMI.....	23
2.3.7. GenLogic.....	24
2.3.8. mySCADA.....	25
2.3.9. Simatic HMI.....	25
2.3.10. Node – RED.....	25
2.3.11. FUXA.....	27
2.3.12. Otras tecnologías.....	27
3. Tecnologías utilizadas.....	30
3.1. Angular.....	30
3.1.1. Estructura tipo de una aplicación Angular.....	32
3.2. Angular Material.....	35
3.3. Visual Studio Code.....	37
3.4. GIT.....	39
4. Captura de requisitos y análisis.....	40
4.1. Requisitos.....	40
4.1.1. Requisitos funcionales.....	40
4.1.2. Requisitos no funcionales.....	45
4.2. Diagrama de Casos de Uso.....	50
4.3. Descripción de Casos de Uso.....	51
4.3.1. Caso de Uso: GestionarUsuario.....	51
4.3.2. Caso de Uso: SeleccionarImagen.....	56
4.3.3. Caso de Uso: ColocarSensores.....	57
4.3.4. Caso de Uso: VerSensores.....	58
4.3.5. Caso de Uso: ObtenerHistórico.....	59
4.3.6. Caso de Uso: GenerarGráfica.....	60
4.3.7. Caso de Uso: IntroducirValoresVálidos.....	61
4.3.8. Caso de Uso: RecibirAlerta.....	62

5. Manual de usuario.....	63
5.1. Pagina Inicio.....	63
5.2. Página Principal.....	66
5.3. Página Registro de Usuarios.....	68
5.4. Página Contacto.....	71
5.5. Página Usuarios.....	73
5.6. Página Gráficos.....	77
5.7. Página <i>Dashboard</i>	84
5.8. Página Alertas.....	94
6. Conclusiones.....	96
7. Líneas futuras.....	98
8. Referencias.....	101

Índice de Tablas

TABLA 1: RF1.1 CREAR USUARIO.....	41
TABLA 2: RF1.2 VER USUARIO	42
TABLA 3: RF1.3 MODIFICAR USUARIO	42
TABLA 4: RF1.4 ELIMINAR USUARIO	42
TABLA 5: RF2.1 CREAR TRABAJADOR.....	42
TABLA 6: RF2.2 VER TRABAJADOR	43
TABLA 7: RF2.3 MODIFICAR TRABAJADOR.....	43
TABLA 8: RF2.4 ELIMINAR TRABAJADOR	43
TABLA 9: RF3 COLOCAR SENSORES	43
TABLA 10: RF4 OBSERVAR SENSORES	44
TABLA 11: RF5 GENERAR HISTÓRICOS.....	44
TABLA 12: RF6 SELECCIONAR IMAGEN	44
TABLA 13: RF7 INTRODUCIR VALORES VÁLIDOS	45
TABLA 14: RF8 RECIBIR ALERTAS	45
TABLA 15: RFN1 INTERFAZ SENCILLA	47
TABLA 16: RNF2 PROTECCIÓN DE DATOS	47
TABLA 17: RNF3 AUTENTICACIÓN	47
TABLA 18: RNF4 NAVEGADORES COMPATIBLES.....	48
TABLA 19: RNF5 COSTE MÍNIMO.....	48
TABLA 20: RNF6 DISEÑO RESPONSIVO	48
TABLA 21: RNF7 VERSIONES POSTERIORES	49
TABLA 22: RNF8 MANEJO DE ERRORES.....	49
TABLA 23: CASO DE USO: GESTIONARUSUARIO	51
TABLA 24: CASO DE USO: CREARUSUARIO.....	52
TABLA 25: CASO DE USO: VERUSUARIO	53
TABLA 26: CASO DE USO: MODIFICARUSUARIO	54
TABLA 27: CASO DE USO: ELIMINARUSUARIO.....	55
TABLA 28: CASO DE USO: SELECCIONARIMAGEN	56
TABLA 29: CASO DE USO: COLOCARSENSORES.....	57
TABLA 30: CASO DE USO: VERSENSORES	58
TABLA 31: CASO DE USO: OBTENERHISTÓRICO.....	59
TABLA 32: CASO DE USO: GENERARGRÁFICA	60
TABLA 33: CASO DE USO: INTRODUCIRVALORES VÁLIDOS.....	61
TABLA 34: CASO DE USO: RECIBIRALERTA	62

Índice de Figuras

FIGURA 1: EJEMPLO OAS (OPEN AUTOMATION SOFTWARE)	20
FIGURA 2: EJEMPLO GRÁFICA TIEMPO REAL	21
FIGURA 3: EJEMPLO ECAVA IGX.....	22
FIGURA 4: EJEMPLO DE APLICACIÓN GENLOGIC.....	24
FIGURA 5: INTERFAZ NODE-RED.....	26
FIGURA 6: INTERFAZ FUXA.....	27
FIGURA 7: COMPARACIÓN DE LOS DIFERENTES FRAMEWORKS O LIBRERÍAS	28
FIGURA 8: ESTRUCTURA DE UNA APLICACIÓN ANGULAR	33
FIGURA 9: ESTRUCTURA APP.MODULE.TS	35
FIGURA 10: TEMA ELEGIDO EN ANGULAR MATERIAL.....	36
FIGURA 11: COMPARACIÓN IDEs ÚLTIMO AÑO.....	37
FIGURA 12: DIAGRAMA DE CASOS DE USO.....	50
FIGURA 13: PANTALLA DE INICIO.....	63
FIGURA 14: FORMULARIO DE LOGIN	64
FIGURA 15: ERROR DE LOGIN	65
FIGURA 16: PÁGINA PRINCIPAL. ROL SUPER ADMINISTRADOR.....	66
FIGURA 17: PÁGINA PRINCIPAL. ROL TRABAJADOR	66
FIGURA 18: PANEL LATERAL. ROL ADMINISTRADOR	67
FIGURA 19: FORMULARIO DE REGISTRO.....	68
FIGURA 20: EJEMPLO DE FORMULARIO INCOMPLETO.....	69
FIGURA 21: PÁGINA DE CONTACTO.....	71
FIGURA 22: FORMULARIO DE CONTACTO	72
FIGURA 23: TABLA COMPUESTA POR LOS DIFERENTES USUARIOS DEL SISTEMA	73
FIGURA 24: FORMULARIO DE MODIFICACIÓN DE USUARIOS	74
FIGURA 25: CONFIRMACIÓN DE ELIMINACIÓN DE USUARIO	75
FIGURA 26: BORRADO DE USUARIO	75
FIGURA 27: CANCELACIÓN DE BORRADO DE USUARIO	76
FIGURA 28: EJEMPLO DE PETICIÓN DE BORRADO CON ROL DE ADMINISTRADOR	76
FIGURA 29: INICIO DE PÁGINA DE GRÁFICOS.....	77
FIGURA 30: PÁGINA DE GRÁFICOS UNA VEZ SELECCIONADO UN COMPONENTE	78
FIGURA 31: FORMULARIO DE SOLICITUD DE DATOS PERSONALIZADO.....	78
FIGURA 32: INTERFAZ PARA LA SELECCIÓN DE FECHAS PERSONALIZADAS	79
FIGURA 33: TABLA RESPUESTA.....	80
FIGURA 34: EXPORTACIÓN DE DATOS A EXCEL.....	80
FIGURA 35: GRÁFICO DE DATOS OBTENIDOS	81
FIGURA 36: GRÁFICA FILTRADA	82
FIGURA 37: ZOOM EN LA GRÁFICA	83
FIGURA 38: PÁGINA DE DASHBOARD	84
FIGURA 39: GRANJA SELECCIONADA	85
FIGURA 40: SELECCIÓN DE IMAGEN ASOCIADA A UNA GRANJA.....	86
FIGURA 41: TRATAMIENTO IMAGEN SELECCIONADA	86
FIGURA 42: PÁGINA DE DASHBOARD CON LA GRANJA SELECCIONADA.....	87
FIGURA 43: GRANJA A PANTALLA COMPLETA	87
FIGURA 44: MOVIMIENTO DE LOS SENSORES POR LA IMAGEN.....	88
FIGURA 45: MENÚ DE CADA SENSOR.....	89
FIGURA 46: RESULTADO DEL TRATAMIENTO EN LA PÁGINA DASHBOARD.....	90
FIGURA 47: MENÚ DE CAMBIO DE TIPO DE DATO.....	91
FIGURA 48: AJUSTE DE VALORES MÁXIMOS Y MÍNIMOS DE LOS SENSORES.....	92
FIGURA 49: NÚMERO DE ALERTAS DEL SISTEMA	92
FIGURA 50: INFORMACIÓN DE LAS ALARMAS DEL SISTEMA.....	93
FIGURA 51: ALERTAS DEL SISTEMA.....	94
FIGURA 52: EXPORTACIÓN DE ALERTAS	95

1. Introducción.

1.1. Introducción.

Debido a las nuevas y demandantes necesidades de las empresas actuales, se ha acelerado enormemente el desarrollo tecnológico de las mismas, hasta tal punto que se ha comenzado a denominar este proceso como Industria 4.0, surgiendo así una nueva revolución industrial.

Y es que, desde la máquina de vapor, la mecanización, la electricidad y la automatización, no se había vuelto a hablar de una revolución de tales características, hasta ahora.

El surgimiento de tecnologías emergentes tan importantes como son el *Big Data*, el IoT (*Internet of Things*), o la interconexión masiva de dispositivos y sensores digitales ha propiciado un cambio empresarial de grandes magnitudes.

A lo largo de este proyecto nos centraremos en uno de estos aspectos, en concreto, el trabajo que contempla la interconexión de sensores digitales para la observación y explotación de los datos generados por estos por parte de usuarios finales.

Para que esto sea posible, desarrollaremos nuestro proyecto con intención de que sea utilizable globalmente y sin depender de un hardware concreto o un software específico complejo.

Se hará, por tanto, un desarrollo web, centrándonos únicamente en la parte visible al usuario o capa de presentación, es decir, el *front end*.

1.2. Motivación.

La principal motivación de este Trabajo de Fin de Grado surge debido a la necesidad de implementar un sistema independiente de cualquier software, específico y novedoso, además de ser fácilmente ejecutable y sostenible. En concreto, son muchas las ventajas que proporciona un sistema HMI (*Human Machine Interface*) para cualquier sector industrial, siendo este accesible de manera global sin requerimientos específicos de *hardware* o *software*.

En lo que respecta al sector rural, se observa que las TIC (*Tecnologías de la Información y las Comunicaciones*) ayudan en gran medida, mejorando aspectos ya contemplados y añadiendo nuevas características:

- Mejora en el acceso a los datos: Con el objetivo de suplir una carencia habitual en este sector, se busca una recogida de datos automatizada, evitando la recolección manual, o realizada con cierta frecuencia, de manera que esta sea en tiempo real y veraz. Además, gracias a la tecnología actual, estos datos serán almacenados y estarán accesibles en los momentos en que sea necesario y desde cualquier localización.
- Reducción de costes económicos y físicos: El uso de aplicaciones basadas en web produce un ahorro significativo, ayuda a tomar decisiones en momentos claves y evita la realización de operaciones administrativas y físicas, obteniéndose todos los registros desde un mismo lugar de manera unificada.
- Mejora en el tratamiento de los datos: El uso de la tecnología no solo nos permite mantener de manera organizada y homogénea los datos, sino que ahora somos capaces de interactuar con ellos, generando informes, gráficas y porcentajes que serán de gran utilidad en la puesta en marcha de acciones concretas de cara a buscar un beneficio mayor o a evitar contratiempos.

1.3. Objetivos.

Los objetivos de este trabajo son claros. Se va a proporcionar la parte visible y accesible de un software específico, basado en web, a un cliente final.

Ha de ser intuitivo, rápido, fácil de gestionar, seguro, con un diseño acorde a lo solicitado, responsivo, y que cumpla con las exigencias del usuario final. Este constará del mantenimiento de un registro de usuarios y la observación de diferente información procedente de sensores colocados en puntos físicos reales.

Cada uno de los usuarios ha de ser capaz de visualizar su propia nave, taller o edificio y situar de manera manual ciertos componentes en posiciones específicas. A su vez, podrá visualizar en tiempo real los datos recogidos por dichos sensores y tratar con ellos para obtener un beneficio diferencial en su trabajo.

1.4. Fases.

El proyecto se divide en tres fases principales:

- *Desarrollo Front End*: Será la parte en la que se centre este documento. Se trabajará en un sistema uniforme y accesible desde cualquier lugar y dispositivo. Debe cumplir las estrictas necesidades solicitadas por el usuario final, debiendo ser además posible la incorporación de cambios y mejoras venideras.
- *Desarrollo Back End*: Como es inevitable, un desarrollo web ha de estar sustentado en una API (*Application Programming Interface*), la cual nos permitirá mantener conversaciones entre el *front end*, la parte donde se visualizan los datos, y el *back end*, el lugar donde se almacenan estos datos. Este bloque será desarrollado por la Unidad de Ingeniería de Datos de la Universidad de Valladolid, y será de vital importancia para el desarrollo del proyecto.
- *Implementación de la lógica del sistema*: Para obtener los datos de los que se nutrirán ambas partes anteriormente comentadas, es necesario disponer de sistemas e instrumentos que obtengan los diferentes datos y los envíen a un repositorio. Será la parte servidora quien se encargue de recoger estos datos y proporcionarlos a la interfaz de usuario de la aplicación para que sean presentados de forma adecuada al cliente.

Sin duda, la interconexión de estas tres partes tendrá una importancia vital en el desarrollo del proyecto final, por lo que es preciso mantener una comunicación continua entre las partes de cara a optimizar los tiempos de desarrollo y a obtener beneficios mayores en comparación a un trabajo individual.

A partir de estos tres componentes, y bajo su correcta interconexión, dispondremos de una aplicación completamente funcional de la cual se nutrirá el usuario final en su objetivo de digitalizar su sistema de recogida y explotación de datos.

Centrándonos en la parte objeto del presente trabajo y descrita en este documento, en primer lugar, se ha realizado una búsqueda intensiva de cara a elegir la tecnología óptima para realizar el desarrollo de una manera eficiente y controlada. Para ello, se han estudiado más de 10 tecnologías diferentes válidas para un desarrollo de este carácter, realizando un estudio de beneficios y dificultades asociadas a cada una de ellas, explicando de manera más detallada aquellas tecnologías con las cuales vamos a desarrollar nuestro proyecto.

Una vez seleccionadas las tecnologías a utilizar para el desarrollo de nuestra aplicación, se procede a hacer un estudio de los diferentes requisitos (funcionales y no funcionales) y casos de usos que serán requeridos para conseguir nuestro objetivo.

Posteriormente, se comienza con la implementación del código que hará factible desarrollar aquellos fines anteriormente comentados. Para ello, se hará uso de un framework llamado Angular.

Por último, se exponen una serie de conclusiones acerca del desarrollo de esta aplicación web y otras de carácter personal, además de unas posibles líneas de trabajo futuro, en caso de querer ampliar las funcionalidades de este proyecto.

1.5. Estructura de la memoria.

El presente Trabajo de Fin de Grado se divide en siete capítulos diferentes, los cuales permiten presentar todos los aspectos abarcados en el proyecto y que se describen a continuación.

Este primer capítulo introductorio hace énfasis en centrar al lector ante el desarrollo posteriormente realizado de la aplicación de automatización enmarcada en el contexto de la industria 4.0.

El segundo capítulo que sigue se centra en el estudio de las diferentes opciones existentes para crear una aplicación basada en sistemas HMI o SCADA, comentando de cada una de las opciones sus beneficios y desventajas en relación al desarrollo que queremos realizar. Además, sirve finalmente para decantarse por una de esas opciones, la cual nos acompañará en toda nuestra aplicación siendo la herramienta con la que conseguiremos implementar los casos de uso propuestos.

Precisamente, el tercer capítulo de este documento se centra en describir de manera más exhaustiva y detallada, aquellos componentes software que serán de gran utilidad para dar soporte a nuestra aplicación, y sin los cuales no hubiera sido posible llevar a cabo el desarrollo de la aplicación.

Un cuarto capítulo nos permite definir aquellos casos de uso requeridos para que nuestra aplicación sea válida, no sin antes hacer un estudio profundo de los requisitos funcionales y no funcionales esperados. Este apartado tiene gran importancia, ya que será el punto de partida para comenzar con el desarrollo de la aplicación.

En el quinto capítulo se presenta un manual de usuario comentando paso a paso todos aquellos casos de uso que un usuario estándar realizará de manera común dentro de nuestra aplicación. Se tendrá en cuenta las diferentes opciones y casos de uso que podrán realizar los diferentes tipos de usuarios de la aplicación. Será de gran ayuda para aquellos usuarios inexpertos o con carencias en el aprendizaje que desean tener una guía intuitiva con la explicación de las diferentes funcionalidades que provee esta aplicación.

El sexto capítulo nos permite comentar de manera clara y concisa las conclusiones obtenidas del trabajo. Hablaremos de la importancia del desarrollo web, así como de la industria 4.0 y de los beneficios que aportan este tipo de aplicaciones en el día a día de las empresas y los consumidores. Además, se proporcionan unas líneas de opinión acerca de este sector y de la importancia de este proyecto.

Un séptimo y último capítulo servirá para dejar claros los aspectos con los que se deberá continuar la aplicación de cara a proporcionar más funcionalidades en el momento que se decida continuar con el proyecto.

Para finalizar, aparecen recogidas la lista de referencias consultadas en la realización del documento y que servirán de base en caso de querer seguir avanzando con este proyecto.

2. Estado del arte.

Este apartado se centra en el estado actual del tema a tratar. Se realizará un estudio de las posibles soluciones que podrían utilizarse en nuestro desarrollo, profundizando en aquellas que aporten más claridad a nuestro objetivo final.

El estudio realizado servirá, en última instancia, para decantarse por una de las opciones estudiadas como tecnología a utilizar para generar nuestra aplicación.

2.1. Uso de la web.

El concepto de un proyecto así surgió en los años 40, cuando Vannevar Bush propuso un sistema distribuido con una interfaz operativa en el cual la información fuese accesible por múltiples personas en función de claves. No fue hasta 1990 cuando Tim Berners-Lee y Robert Cailliau inventaron la Web, y desde entonces no ha dejado de ser consumida cada día por más usuarios.

El primero de ellos, sentó las bases de la conocida como *World Wide Web*, apareciendo también el primer navegador web, siendo este no propietario, y facilitando el desarrollo de servidores y clientes independientemente y sin restricciones de licencia.

Estas primeras páginas web carecían de contenido dinámico y estilo, ya que únicamente estaban desarrolladas con un lenguaje de marcado, el conocido HTML (*Hypertext Markup Language*) que permitía mostrar contenido estático por pantalla.

Surgieron entonces lenguajes como CSS (*Cascading Style Sheets*), permitiendo aportar estilos al contenido creado mediante HTML, y Java Script, el cual nos otorga la posibilidad de dinamizar este contenido y crear acciones con la finalidad de interactuar con nuestra aplicación web.

Empresas especialistas en el análisis del consumo de Internet a través de la web, como *Statia* o *Hootsuite*, concluyen que más de 4 billones y medio de personas en el mundo consumen este tipo de servicio (Statista, 2001; Kemp, 2021). Además, cabe señalar que el 91% de personas españolas tienen acceso a Internet, lugar de aplicación de nuestro proyecto.

2.2. Desarrollo de aplicaciones web.

Vistas las estadísticas del uso de Internet y de la web, es obligatorio pensar en el gran impacto que pueden llegar a tener las aplicaciones servidas en esta plataforma. Tal es así, que este formato de aplicaciones ha aumentado con el paso del tiempo hasta convertirse en un estándar profesional.

Podría decirse que el desarrollo de aplicaciones web es la disciplina que se encarga de crear el entorno de trabajo, la implementación y el mantenimiento de un sitio web. Se centra entonces en el correcto funcionamiento del sistema, siempre teniendo en cuenta que este será accesible desde cualquier navegador en cualquier dispositivo.

Al contrario que en aplicaciones multiplataforma, donde el software ha de ser desarrollado e instalado para cada uno de los dispositivos finales, en una aplicación web los datos son procesados y almacenados, en la web, por lo que no es necesario instalar ningún software en nuestros equipos, más allá del propio navegador.

De cara a nuestro desarrollo, una parte importante será decidir entre una gran multitud de opciones para elegir bajo qué marco de trabajo queremos continuar.

2.3. Opciones para SCADA/HMI.

Dada la finalidad de nuestro proyecto, donde realizaremos conexiones entre varios sensores y dispositivos instalados en puntos concretos que nos indique el usuario final, será necesario replicar un entorno o sistema de control industrial.

En este punto surgen dos claras definiciones de servicios, interconectadas, pero con claras diferencias. La primera de ellas será un HMI (*Human Machine Interface*) o interfaz hombre-máquina. Hace referencia a la forma en que un humano, el usuario, interactúa con las diferentes máquinas que componen o forman el sistema industrial, donde cada una de ellas muestra, de manera gráfica y en tiempo real, datos o servicios que proporcione, atendiendo siempre a las órdenes del cliente.

Por otra parte, se define un sistema SCADA (*Supervisory Control And Data Acquisition*) como un sistema de automatización. Se podría decir que sube un nivel de abstracción con respecto a un HMI, ya que este se comunica con uno o varios elementos, tales como las redes de automatización, los sistemas de adquisición de datos, la telemetría, los sistemas de seguridad y más, controlando y analizando la comunicación del humano con la máquina. La diferencia radica entonces en que un software de tipo HMI es utilizado para dar soporte y monitorizar uno o varios sistemas SCADA.

En la búsqueda de un entorno de desarrollo válido para nuestro proyecto, han surgido varias opciones.

2.3.1. Open Automation Software.

Se trata de una de las opciones más llamativas para desarrollar un entorno industrial basado en sistemas SCADA. Con más de 28 años de experiencia se sitúan en la cabeza del sector industrial centrado en el desarrollo de sistemas HMI.

Es posible solicitar una versión de prueba que permitirá modificar un pequeño sistema como el que vemos en la Figura 1 e interactuar con componentes de una bomba de agua.



Figura 1: Ejemplo OAS (Open Automation Software)

Además, es posible observar gráficos temporales y en tiempo real, tal y como vemos en la Figura 2.

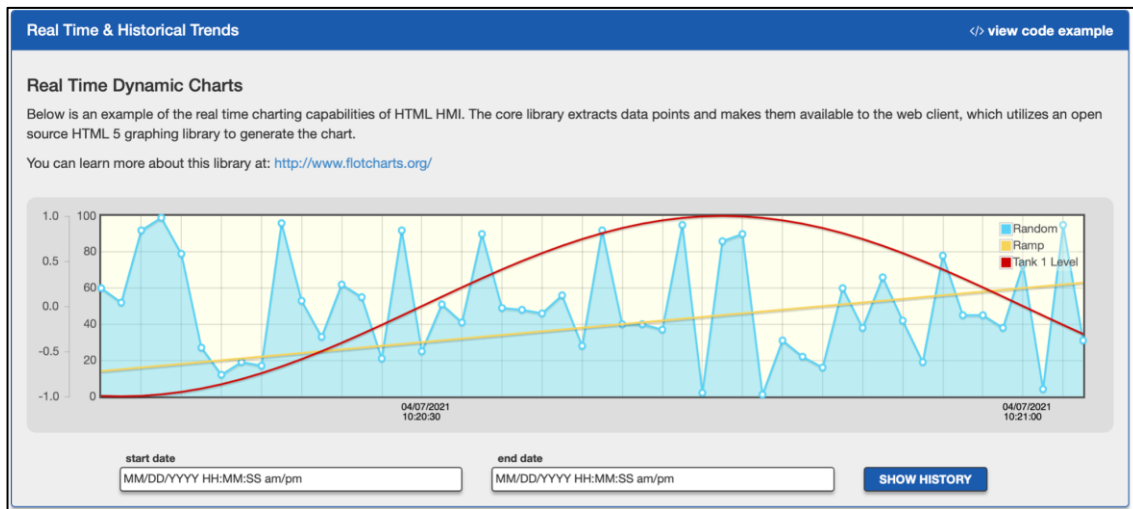


Figura 2: Ejemplo gráfica tiempo real

Desde su página web (Open Automation Software, 2021) aseguran un entorno seguro, ágil, flexible y fiable, pero con un equipo de soporte detrás que hará de ello una solución con un coste asociado, sin comentar que, además, se trata de un software únicamente instalable en Windows, por lo que rápidamente lo descartamos para nuestro proyecto.

2.3.2. Ecava IntegraXor.

Se trata de otro desarrollador de software industrial de renombre. Según comentan en su plataforma online (Ecava IGX Web SCADA, 2020), esta vez sí será un desarrollo totalmente web y responsivo, con funcionalidades interesantes como la creación de gráficos y el manejo de alertas en tiempo real, tal y como se observa en el menú lateral de la Figura 3, pero con el problema de su precio mensual que hace de esta una solución inválida para nuestros requerimientos.

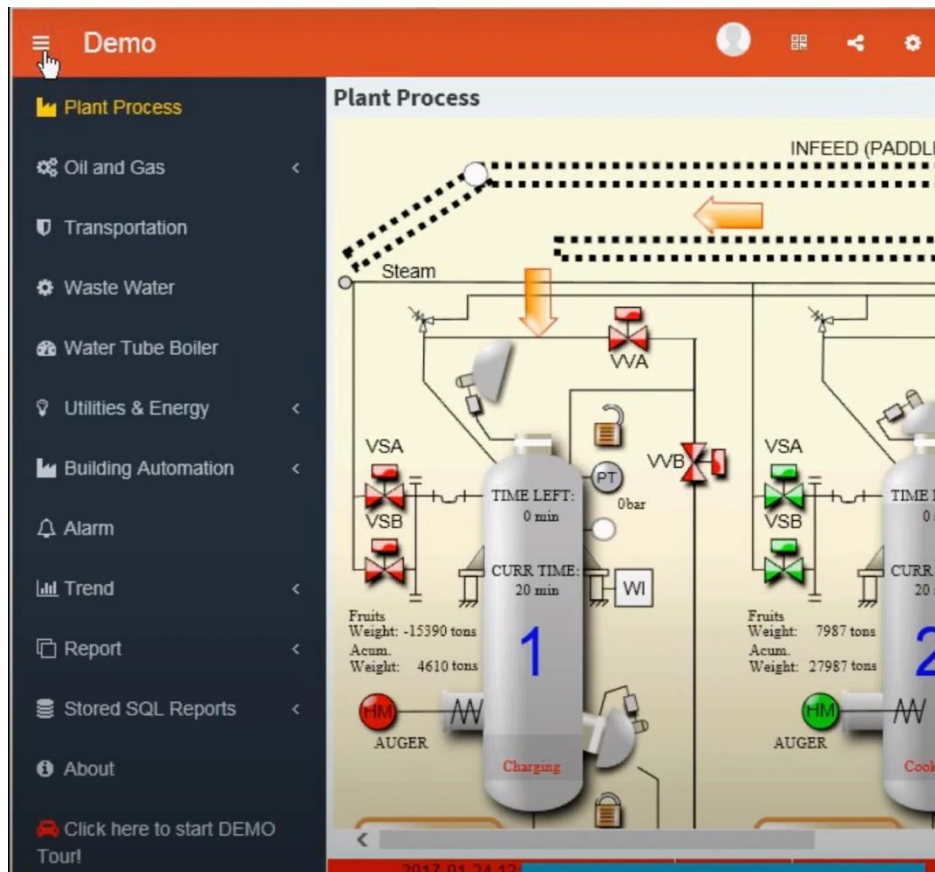


Figura 3: Ejemplo Ecava IGX

Se descarga y evalúa una versión de prueba donde comprobamos que el desarrollo se basa en un entorno propietario en el que, para realizar algún proceso válido, necesitamos una licencia o suscripción.

2.3.3. QuickHMI.

Encontramos otra posible solución basada en web donde podríamos realizar el diseño *front end* de nuestro sistema HMI. Esta solución conlleva un coste asociado de 150 euros durante el primer año, limitando el desarrollo después de los primeros doce meses, pero manteniendo el entorno en tiempo de ejecución.

Se trata de una aplicación de escritorio (QuickHMI, 2019) para sistemas Windows en la cual se puede generar un entorno de desarrollo donde podremos interconectar componentes y monitorizar sus datos. Además, existe la posibilidad de generar alarmas en situaciones excepcionales para avisar al usuario en tiempo real, y es compatible con códigos HTML, Java Script y CSS.

Cuenta también con la posibilidad de generar una aplicación móvil que permite realizar los mismos procedimientos que su versión web, siendo esto un aliciente para su utilización.

Debido al coste asociado se decide dejar esta opción a un lado, aunque servirá como referencia en el desarrollo final.

2.3.4. Pvbrowser.

Durante la búsqueda realizada surge la posibilidad de utilizar el software proporcionado por Pvbrowser al estar bajo licencia libre y accesible para todo el mundo.

Se trata de una aplicación de escritorio basada en el lenguaje de programación C++ disponible para la mayoría de los sistemas operativos (Pvbrowser, 2020). Descargando la aplicación en nuestro dispositivo, y comprobando su funcionamiento, en seguida nos dimos cuenta de sus limitaciones visuales, además de la necesidad de programar todo el proyecto en C++ y con pocas facilidades para su implementación en combinación con códigos HTML o Java Script, decidiendo entonces continuar con la búsqueda activa de una tecnología propia para el desarrollo del proyecto.

2.3.5. CuteHMI.

Esta nueva tecnología de desarrollo recoge un software de código abierto formado por componentes destinado a la creación de aplicaciones HMI y SCADA.

La aplicación de desarrollo está escrita en C++ y QML (*Qt Meta Language*), destacando por su reutilización, modularidad y flexibilidad, aunque no dispone de una implementación web, lo que impide que podamos usar su tecnología en nuestro proyecto (*CuteHMI*, 2020).

2.3.6. WebIQ - Smart HMI.

Esta solución nos ofrece todas las funcionalidades de un sistema SCADA, almacenando y evaluando todo tipo de datos. WebIQ permite una escalabilidad que llega hasta el procesamiento de millones de variables, permitiendo así crear HMI basadas en web para cualquier tipo de máquina y sistema de proceso.

Tal es esto, que es usada tanto para monitoreo y control de procesos de larga duración o continuos, hasta para implementar soluciones destinadas al Internet Industrial de las Cosas (IIoT) o elementos de visualización personalizables para pantallas de vehículos o máquinas móviles. Además, a través de esta aplicación se puede incorporar cualquier tipo de widget personalizado, ajustándose a las necesidades del cliente.

Para establecer cualquier tipo de sistema interconectado mediante esta aplicación, no se requieren grandes conocimientos de programación, ya que la configuración se realiza mediante un simple sistema de *drag and drop* dentro de cualquier navegador web, reduciendo así costes de desarrollo y tiempo. A su vez, es posible crear scripts o clases personalizadas desde un editor integrado en la misma aplicación (*Smart HMI*, 2021).

Sin duda, se trata de una de las opciones más profesionales para desarrollar un HMI, como han podido demostrar gracias a sus trabajos junto a la empresa alemana Bosch, pero que, debido a su coste al tratarse de una solución pensada para grandes empresas, se ha decidido descartar.

2.3.7. GenLogic.

La empresa *Generic Logic* es capaz de proveer gráficos en tiempo real y visualización de datos para interfaces SCADA/HMI, además de soluciones GIS (*Geographical Information System, Sistema de Información Geográfica*) para desarrollo de software en todo el mundo. Sus productos son utilizados actualmente para controlar procesos en tiempo real y con un carácter crítico en empresas industriales, como, por ejemplo, para el control de plantas químicas, sin dejar de lado el carácter geográfico, trabajando incluso para el lanzamiento de satélites y el monitoreo de la seguridad aeroportuaria (GenLogic, 2019).

GenLogic es una herramienta potente y flexible que permite crear interfaces HMI basadas en la recogida de datos, con capacidad para implementarse en cualquier plataforma, incluyendo dispositivos móviles.

Su gran variedad de componentes y su kit de herramientas de visualización, así como widgets, ha despertado gran interés mundial, llegando a mantener acuerdos con el Instituto Técnico de Massachusetts (MIT), el Centro Espacial Kennedy, o la fuerza aérea de los Estados Unidos. Desde su página oficial se pueden realizar demostraciones en vivo de aspectos propios de la aplicación, tal y como vemos en la Figura 4, escritas en HTML5 y Java Script. Este servicio cuenta con una versión gratuita de prueba, la cual incluye un editor gráfico y bibliotecas para el desarrollo en C, C++, C#, Java y Java Script.

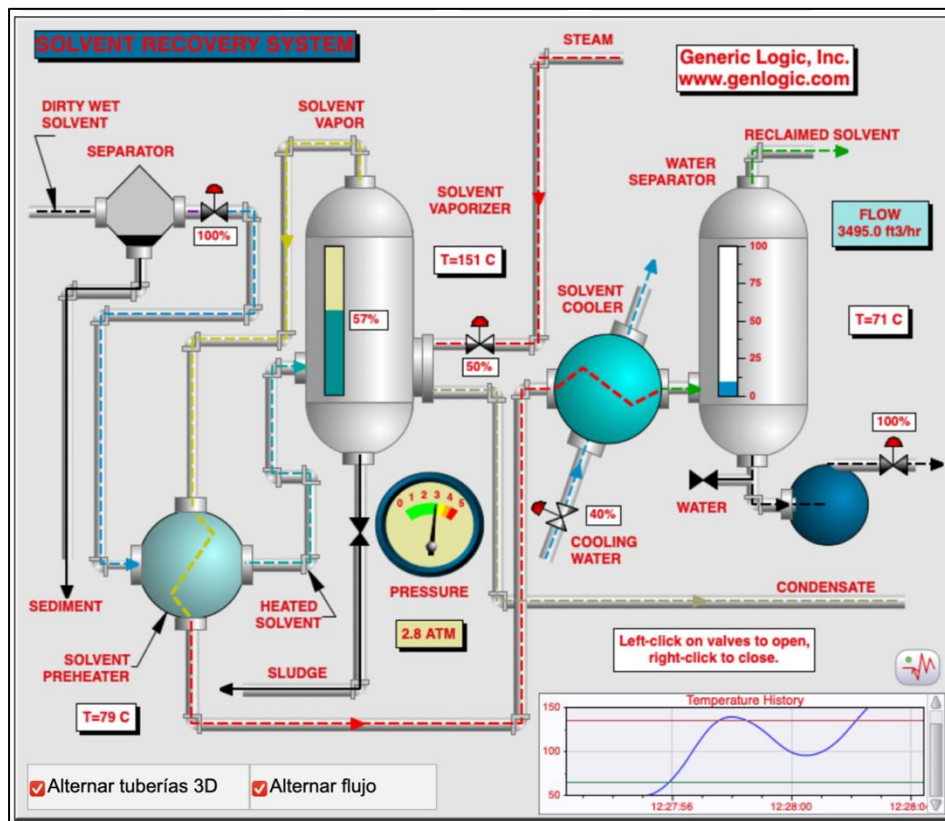


Figura 4: Ejemplo de aplicación GenLogic

Para obtener una versión profesional, estable y duradera, se debe pedir una reunión profesional con carácter de compra del producto, por lo que se decidió continuar con la búsqueda de otras soluciones.

2.3.8. mySCADA.

Se trata de otra opción propietaria para desarrollar y crear visualizaciones ricas en características, datos, notificaciones o alarmas. Es un producto amigable con el usuario y flexible, permitiendo acelerar el proceso de diseño incluyendo bases de datos personalizadas e importación y exportación directa desde Excel. Tiene capacidad de uso en cualquier dispositivo, incluyendo dispositivos móviles.

Actualmente, comercializan dos opciones. La primera de ellas dispone de una versión gratuita pero limitada, y una versión avanzada, siendo ambas una aplicación instalable en la mayoría de sistemas operativos como Windows y Mac. Respecto al segundo producto, se trata de un sistema HMI al uso, con visualización y control de todo tipo de tecnologías, aunque esta vez está basado en la web y por el que tendremos que hacer un desembolso mayor.

A su vez, esta compañía comercializa productos industriales para que estos sistemas SCADA/HMI puedan funcionar de manera idónea. Uno de estos productos, *myBOX*, es un router industrial con el sistema de SCADA integrado, y *myPANEL*, es un panel de operaciones con un sistema de SCADA también integrado (MySCADA Technologies, (2020).

2.3.9. Simatic HMI.

Este producto desarrollado por Siemens cuenta con una biblioteca amplia de plantillas, imágenes y objetos para crear un diseño HMI intuitivo y moderno. Surge así una biblioteca de TIA Portal, un software propio de la compañía, en que apoyarse para seleccionar las pantallas, diálogos y mensajes en la configuración.

Este sistema contiene una estructura modular pudiéndose utilizar con WinCC, un entorno de desarrollo propietario útil en el desarrollo de sistemas HMI y optimizado para su funcionamiento de manera táctil en pantallas de manera intuitiva (Siemens México, 2019).

Se trata de un programa instalable en Windows donde se realizan futuros diseños que después serán instalados en paneles de la propia compañía, siendo, por esta razón, por la que se dejó de pensar en esta opción.

2.3.10. Node – RED.

Esta herramienta desarrollada por IBM permite implementar de manera visual conexiones entre dispositivos hardware. Fue diseñada bajo la finalidad de integrar dispositivos físicos con otros equipos simplificando la programación.

Node-RED se ha consolidado entre los *frameworks* más destacados de carácter *open-source* para el desarrollo, la gestión y la transformación en tiempo real de aplicaciones encuadradas en el ámbito de Industria 4.0, IoT o sistemas de Inteligencia Artificial (Aprendiendo Arduino, 2020), siendo adecuado para conectar dispositivos de *hardware*, APIs, y servicios propios de Internet.

Está basado en NodeJS (un entorno multiplataforma centrado en la capa del servidor), y su sencillez permite no tener que profundizar hasta el mínimo detalle siendo una tecnología óptima para mantener un hogar domotizado, entre otros ejemplos, pero que muestra algunas deficiencias en lo que a un desarrollo web completo se refiere, siendo complejo interconectar esta tecnología con otros aspectos propios de un desarrollo industrial con gran carga de trabajo.

Para probar esta herramienta, se necesita tener instalado NodeJS y el manejador de paquetes NPM (*Node Package Manager*). Una vez instalados, bastará con el comando “*node-red*” y podremos acceder a su interfaz de desarrollo de manera local en el puerto 1880.

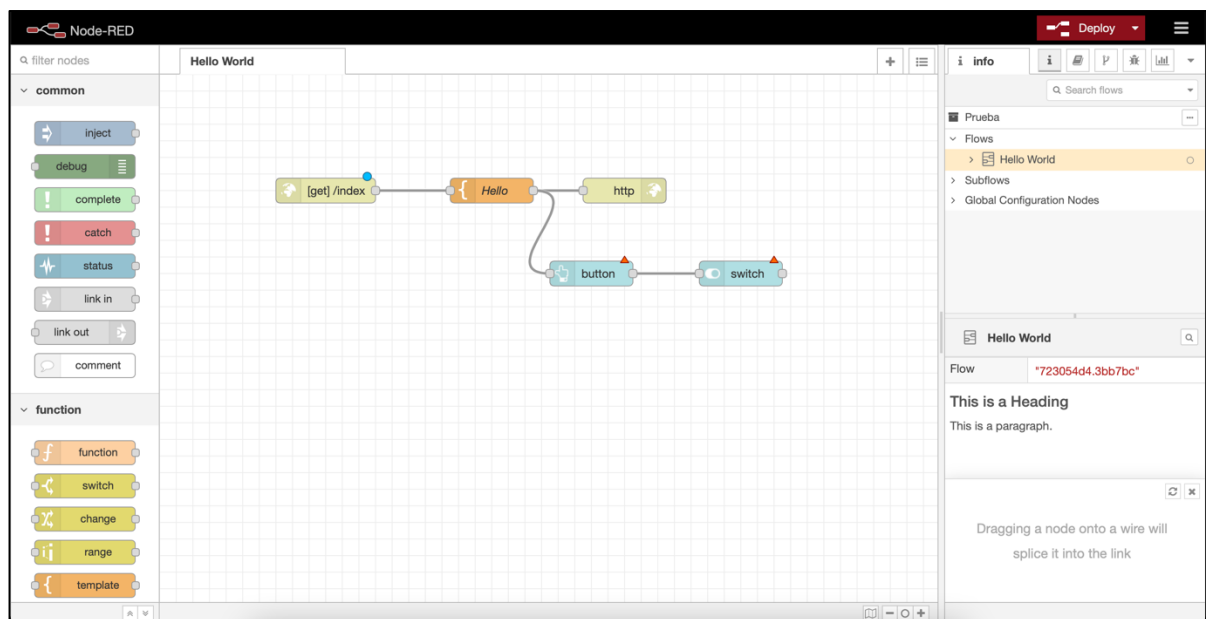


Figura 5: Interfaz Node-RED

Como podemos observar en la Figura 5, se trata de un sistema de *drag-and-drop* a partir de cual iremos colocando los diferentes objetos en orden de apariencia en nuestro sistema, los cuales se ejecutarán inmediatamente después de finalizar el bloque anterior, pudiendo hacer tanto peticiones a servidores externos como manejar opciones al seleccionar los diferentes componentes de nuestro diseño tales como botones, enlaces o switch.

De cara a la industrialización de sensores, resultan de gran interés sus módulos para generar *dashboards* o paneles de control, recibiendo actualizaciones periódicas con

nuevos datos recogidos por dispositivos reales y enviados directamente a la aplicación o recogiendo estos desde una API servidora.

2.3.11. FUXA.

Este sistema alojado en la plataforma de GitHub, (sistema utilizado mundialmente para crear proyectos de código abierto de manera colaborativa entre varios desarrolladores), proporciona una herramienta cuidada para la gestión de dispositivos inteligentes de manera que podamos recoger sus datos y manejarlos de manera interactiva y visual para el usuario (GitHub, 2021).

Surge con la finalidad de gestionar sistemas HMI, proporcionando tanto la parte cliente o *front end*, desarrollada en el *framework* Angular, como la *back end*, desarrollada en NodeJS.

Tal y como podemos observar en la Figura 6, consta de un entorno de desarrollo en el cual podremos seleccionar objetos proporcionados en el menú lateral y pegarlos en el editor para interactuar con ellos.

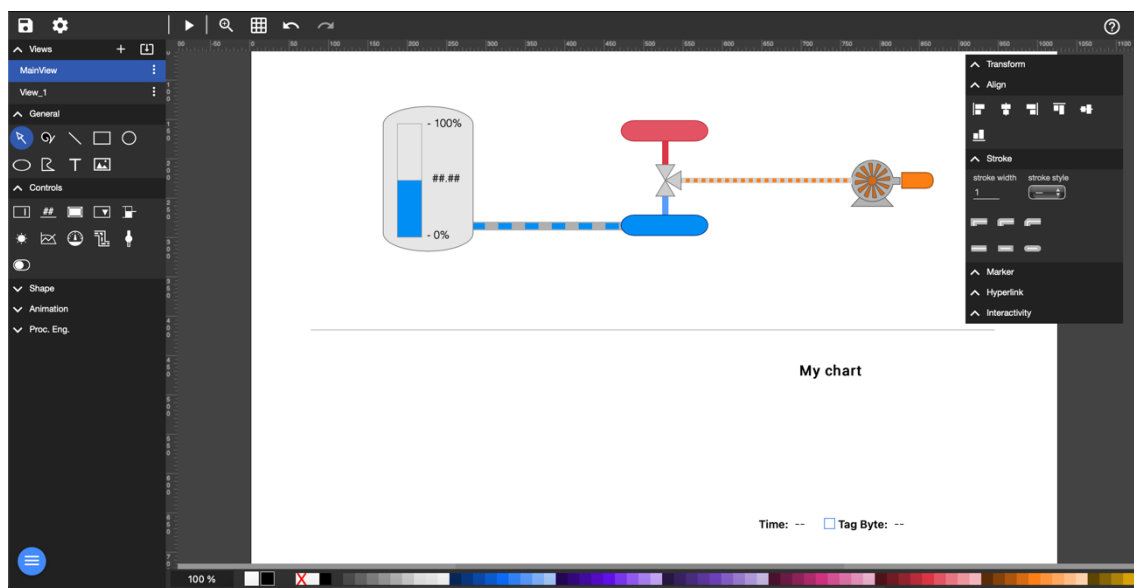


Figura 6: Interfaz FUXA

A pesar de que parece un sistema idóneo para la realización de este proyecto, no dispone de un sistema completo de trabajo, proporcionando escasos elementos que hace que esta opción sea inválida para nuestro desarrollo, aunque será de especial utilidad para poder obtener un esquema del modo de funcionamiento esperado.

2.3.12. Otras tecnologías.

Como consecuencia de la necesidad de un software libre, con capacidad para gestionar tanto componentes como los usuarios del sistema de manera sencilla, decidimos utilizar un *framework* o una librería de Java Script que nos permitiera realizar un desarrollo limpio, modular, usable y con una finalidad clara. En este punto de la labor de

investigación realizada surgen tres librerías en las que sería posible apoyar la implementación de nuestro código *front end*: Angular, React y Vue.

Por orden cronológico, el primero en surgir fue Angular. Google lanzó este *framework* Java Script de código abierto en 2010, con la finalidad de crear aplicaciones de una manera simple en una única página.

Se trata entonces de una *Single Page Application* (SPA), cargando completamente la página al lanzarse la aplicación, aumentando así el tiempo de consulta inicial, pero manejándose con fluidez durante el resto de la navegación.

Angular separa el *back end* del *front end*, evitando escribir código repetitivo y manteniendo, tanto las líneas de código como los ficheros, ordenados gracias a su patrón Modelo-Vista-Controlador, posibilitando modificaciones y actualizaciones en tiempo real sin necesidad de recompilar la aplicación. Además, permite crear nuevas etiquetas HTML personalizables al estar basado en el estándar de componentes web.

React sin embargo fue promovida por Facebook y lanzada en 2013. Se trata de una librería Java Script cuya finalidad es, según ellos mismos afirman, el desarrollo de interfaces de usuario.

Sirve para desarrollar páginas web de manera ordenada y con notablemente menos líneas de código que utilizando Java Script convencional u otras librerías como jQuery. Se centra en la manipulación del DOM (*Document Object Model*) permitiendo que las vistas se asocien con los datos.

Según se puede apreciar en la Figura 7, React se sitúa como el *framework* Java Script con más búsquedas en el último año en Google, seguido de cerca por Angular.

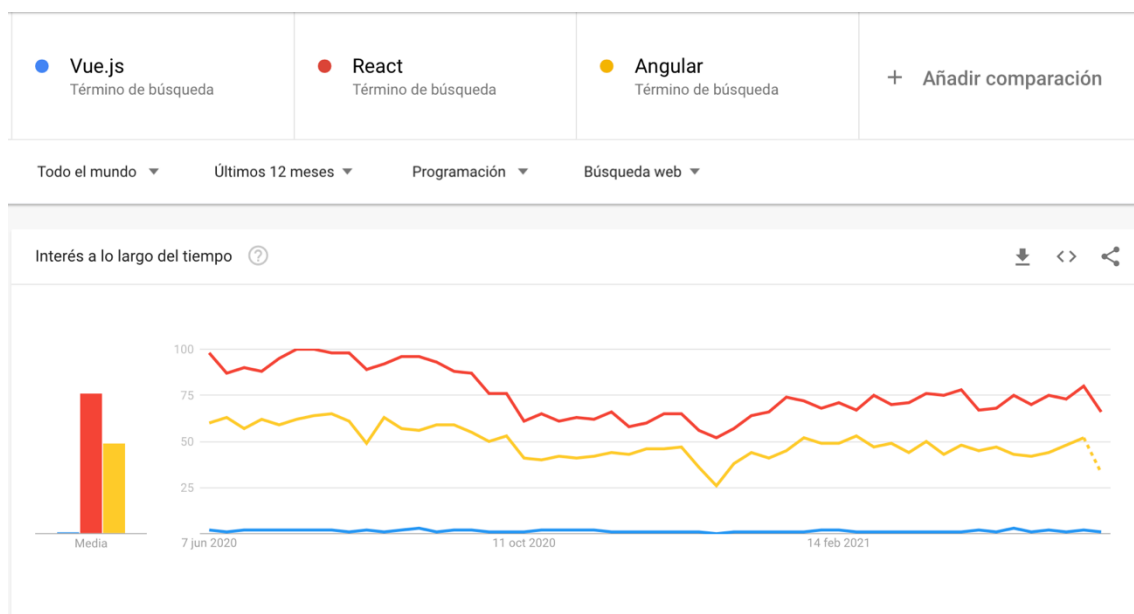


Figura 7: Comparación de los diferentes frameworks o librerías

Por último, Vue fue lanzado en 2014. No está avalada por una empresa de tales dimensiones como en los casos anteriores, sin embargo, va ganando popularidad poco a poco con el paso del tiempo, y reúne características de Angular y de React. Se trata de un *framework open source*, que permite construir interfaces de usuario de manera rápida y sencilla. Al igual que Angular y React, Vue también trabaja con componentes, permitiendo así el desarrollo de proyectos modularizados y fácilmente escalables.

Mientras Angular y React utilizan Type Script para escribir sus componentes, Vue deja libertad al desarrollador para este cometido, permitiéndole trabajar con la herramienta que este prefiera, siendo posible utilizar únicamente HTML, CSS y Java Script, si así se desea.

Debido a su gran aceptación, el respaldo de la empresa estadounidense Google, toda la información que encontramos para el desarrollo e implementación, y las pocas soluciones actualmente disponibles para el desarrollo de sistemas HMI propios, se decidió finalmente utilizar Angular como *framework* de desarrollo para nuestro proyecto.

3. Tecnologías utilizadas.

En este punto realizamos un estudio de las tecnologías que vamos a utilizar para la realización de este proyecto.

3.1. Angular.

Según lo comentado en el apartado 2.3.12, Angular se define como un *framework* para el desarrollo de páginas web centrado exclusivamente en el *front end*. La empresa californiana Google comenzó con el desarrollo de un sistema exclusivo para la puesta en marcha de sitios web que permitieran tanto crear contenido mucho más profesional, como hacerlo con menos esfuerzo y recursos.

En 2010, apareció por primera vez AngularJS o Angular 1, forjado bajo la creencia de que la programación declarativa, que es un paradigma de programación que expresa la lógica de un cálculo sin apenas describir su flujo de control, es la que debe utilizarse para crear interfaces de usuarios y enlazar componentes de software, dejando así la programación imperativa para expresar la lógica del negocio a desarrollar.

Hasta entonces, y para el desarrollo de la parte visible del cliente, se hacía uso de librerías como jQuery, ayudando así a implementar el código Java Script de la aplicación. Estos sistemas ayudan a modificar el DOM (*Document Object Model*) y a añadir efectos a nuestro servicio web, pero no había ningún patrón en el que fijarse y todo nuestro código iba encapsulado en funciones que creaba el usuario de manera manual.

Con esta primera versión de Angular se amplía el HTML tradicional para proporcionar un contenido dinámico, permitiendo la sincronización automática de modelos y vistas. AngularJS no trabaja directamente con la manipulación del DOM y mejora notablemente el rendimiento y la ejecución de pruebas en tiempo real. A continuación, se resumen sus principales objetivos:

- Tratar de separar los intereses de la aplicación. Este principio de diseño se centra en separar las distintas secciones de un programa informático de cara a que cada sección apunte a un interés único y delimitado.
- Considerar las pruebas de la aplicación del mismo grado de importancia que la escritura de la lógica de la aplicación.
- Mantener una separación continua entre el lado cliente y el lado servidor, permitiendo el avance en paralelo del producto final y la reutilización de ambos componentes en un futuro.

Debido al uso de dependencias externas y su fácil inyección en el código, con AngularJS se pueden manejar servicios propios del *back end* desde el *front end*, reduciendo la carga del lado del servidor y consiguiendo así aplicaciones web más ligeras.

En 2016 surgió Angular 2 o simplemente Angular. Se trata de un nuevo *framework*, escrito de nuevo desde sus inicios y con conceptos y formas de trabajar diferentes, por lo que, en vez de tratarse de una nueva versión, se considera una nueva tecnología.

Angular, al contrario que AngularJS, usa un sistema de inyección de dependencias jerárquico, es decir, que permite suministrar objetos a las diferentes clases del proyecto, en lugar de que se tengan que crear dichos objetos en cada clase. Por lo tanto, nuestras clases no crean los elementos que necesitan, si no que estos son obtenidos de una clase contenedora. Además, Angular implementa la detección de cambios incrementando así el rendimiento considerablemente en comparación con AngularJS. A su vez, está más orientado a dispositivos móviles, renderizando el mismo contenido de forma distinta para cada tipo de navegador.

Este nuevo *framework* se caracteriza por estar escrito en Type Script y cumplir con ES6 o ECMAScript 6, una especificación de lenguaje de programación basado en Java Script que soporta algunas características de la programación orientada a objetos y que la mayoría de los navegadores entienden y saben manejar. Desde entonces, y con una frecuencia aproximada de 6 meses, el equipo de Google ha ido lanzando nuevas versiones de Angular, introduciendo aspectos novedosos con cada una de ellas hasta llegar hoy en día a la duodécima versión.

Type Script es el lenguaje utilizado por Angular para el desarrollo de su contenido. Aparece en 2012 respaldado por Microsoft como solución al desarrollo de aplicaciones web con Java Script a gran escala, dando así una posible solución a muchos de los problemas ocasionados con este lenguaje. La intención de Type Script es el desarrollo de aplicaciones robustas mediante la implementación de características en el lenguaje que otorguen la posibilidad de desarrollar herramientas avanzadas para el diseño de la aplicación. Se trata de un *super set* de Java Script, queriendo decir esto que las líneas de código de un programa Java Script, son válidas también en un código Type Script, a pesar de que sean diferentes lenguajes de programación. Otra característica que hace que este sea un *super set* del otro, es que, en el momento de compilar, el código Type Script es convertido a Java Script, que es lo que entienden todas las plataformas.

Actualmente, Type Script es utilizado en numerosos sistemas de programación, tales como los mencionados en el apartado anterior, React y Vue, y por parte de Microsoft en su desarrollo de Bing y Office 365. Type Script aporta como gran novedad el tipado estático y objetos basados en clases. A través del tipado estático podemos detectar errores de asignación de variables en tiempo de desarrollo, ya que será aconsejable definir el tipo de dato que espera cada una de las variables. Por otra parte, una clase resulta una estructura predefinida capaz de servir como molde para crear objetos, que serán los atributos y métodos que participen en nuestra aplicación.

3.1.1. Estructura tipo de una aplicación Angular.

En este punto vamos a ver cómo comenzar un proyecto en Angular. En primer lugar, debemos aclarar que, para comenzar nuestro desarrollo, será necesario tener instalada una versión de Node.js. Este ambiente de servidor de código abierto nos permitirá ejecutar código Java Script, ayudándonos a manejar librerías que servirán de base para la construcción del proyecto, tal y como es Angular CLI.

Para ello, nos dirigimos a su página oficial (NodeJS, 2017) y seguimos los pasos para su correcta instalación. Node.js nos provee de una herramienta denominada NPM (*Node Package Manager*) que será de gran utilidad para gestionar el uso de paquetes en nuestro entorno. Dado que Node.js no incluye librerías para desarrollar código directamente con Angular, será necesario instalar el paquete Angular CLI, el cual nos ayudará a comenzar con nuestro proyecto.

Para completar este segundo paso, abrimos la consola del sistema y ejecutamos el siguiente comando: `npm install -g @angular/cli`, el cual añadirá el paquete Angular CLI de manera global en nuestro sistema. Una vez instalado, procederemos a crear nuestra aplicación Angular desde la línea de comandos.

En este caso, iremos a la carpeta que contendrá nuestro proyecto y lanzaremos este comando, `ng new ProjectTitle`, donde podremos indicar parámetros de preferencias tales como si añadiremos routing a nuestro trabajo, es decir, si va a ser o no una SPA, y qué formato utilizaremos para dar estilo a nuestros componentes, pudiendo elegir entre CSS, SCSS, Sass, Less y Stylus. Nosotros seleccionamos CSS en una primera instancia, aunque será posible su cambio más adelante.

Es momento de observar detenidamente qué ficheros y estructura de aplicación hemos creado, para ello abrimos nuestra carpeta y vemos la estructura de una aplicación Angular, la cual se muestra en la Figura 8.

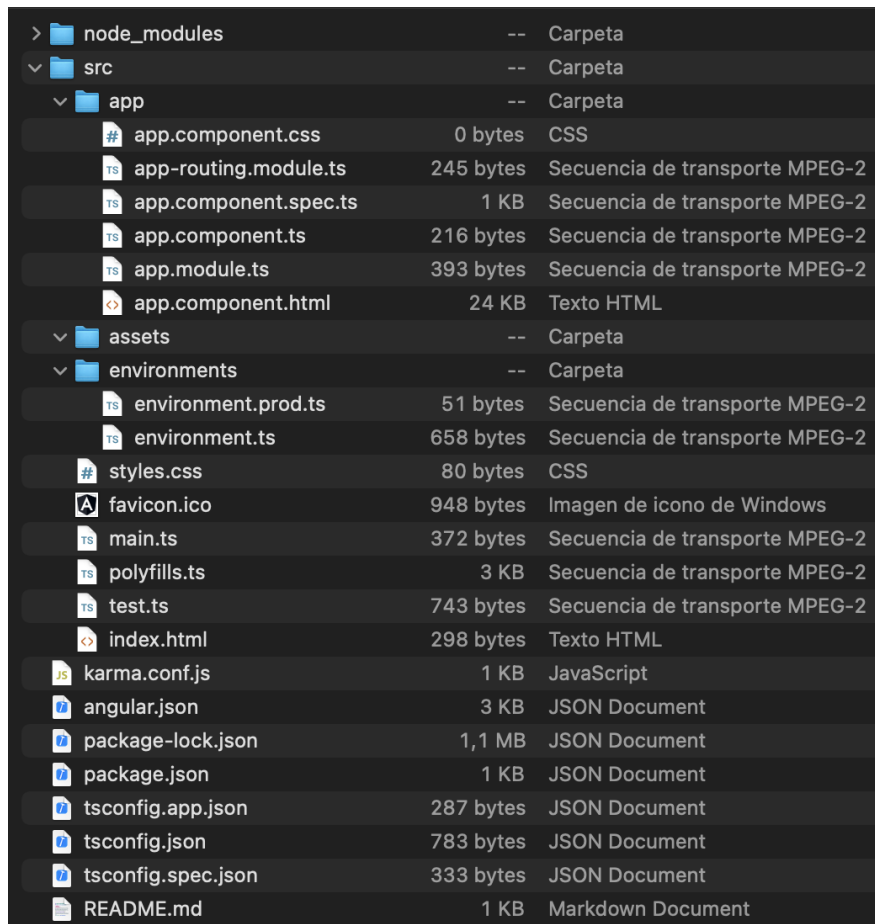


Figura 8: Estructura de una aplicación Angular

Lo primero que nos encontramos en nuestro proyecto es la carpeta de *node_modules*, en la que se incluirán todos los paquetes manejados por NPM en nuestra aplicación. Para poder utilizar dichos paquetes en nuestra aplicación, han de estar definidos dentro de nuestro *package.json*, donde además se define el nombre y versión de nuestro proyecto, así como las dependencias de paquetes de desarrollo sobre los que construiremos nuestra aplicación, almacenadas en la carpeta anteriormente comentada. En base a este último fichero, el servicio NPM descargará, instalará o actualizará las librerías de las que depende nuestro proyecto.

El fichero *angular.json* proporciona los valores predeterminados de configuración de CLI para el espacio de trabajo, incluidas las opciones de configuración para compilar, ejecutar y probar las herramientas que se utilizan. Otro fichero a destacar será *tsconfig.json*, a partir del cual se heredan el resto de archivos de configuración.

Por último, cabe comentar la finalidad de la carpeta denominada *src*. Aquí se encontrarán los ficheros de origen para nuestro proyecto, siendo este el nivel raíz y pudiendo alojar carpetas de componentes dentro de ella, como veremos más adelante.

Dentro de *src*, tenemos ya creadas automáticamente tres carpetas. En *assets* alojaremos principalmente aquellos recursos de componente gráfico de la aplicación, es decir, será el lugar más correcto para almacenar fotos y vídeos que se ofrezcan desde nuestra aplicación Angular. En *environments* vienen predefinidos dos ficheros, *environments.ts*

es utilizado para almacenar variables globales que se requieran en el proyecto, mientras que *environments.prod.ts* se utilizará con la misma finalidad cuando el proyecto se encuentre en fase de producción.

Por otra parte, el fichero *main.ts* será el punto de inicio de nuestra aplicación, lugar donde se carguen las variables globales y el modulo principal del proyecto. El archivo denominado *index.html* contendrá únicamente una etiqueta en el *body*, `<app-root></app-root>`, indicando de esta manera que encerrará todo el contenido *html* dinámico de la aplicación.

La carpeta *app* contendrá el código del proyecto. Está compuesta inicialmente por varios archivos. *app.module.ts* será el módulo de inicio del proyecto, llamado por *main.ts* y lugar de carga de los distintos componentes, módulos y servicios que se encontrarán disponibles para todos los componentes de la aplicación. *app.routing.module.ts* se encarga de gestionar la navegación de nuestra aplicación, ya que indicamos en el proceso de creación que no se trataría de una *Single Page Application*. Por último, los ficheros *app.component.ts*, *app.component.html* y *app.component.css* desarrollan el componente principal de la aplicación. El fichero Type Script define la clase y el comportamiento del componente e indicará la ruta del archivo que contiene su parte visual, el contenido desarrollado en el fichero *html*, y la ruta del archivo que contiene los estilos que diseñan la interfaz visual, el fichero *css*.

Por último, y para poder ejecutar nuestra aplicación escrita en Angular, la publicamos localmente y la ejecutamos desde nuestro navegador. Para ello se hace uso de un comando proporcionado por Angular CLI que levanta un servidor local, también proporcionado por dicha librería, para servir nuestra aplicación.

Navegamos hasta la ruta contenedora de la aplicación y escribimos en una máquina de comandos “*ng serve*”, línea que nos proporcionará la aplicación en el puerto 4200 de nuestro *localhost*, accediendo a la url “*http://localhost:4200*” desde cualquiera de nuestros navegadores, o indicando en el comando la opción “*-open*”, la cual nos abrirá una nueva ventana con la aplicación (Angular, 2017; Enríquez, 2019).

3.2. Angular Material.

Antes de comenzar con esta librería, que será de gran utilidad en nuestro desarrollo, vamos a hablar sobre *Material Design*.

Se trata de una normativa de diseño orientada a la visualización en cualquier plataforma, desde web hasta móvil. Al igual que Angular, fue desarrollada por Google y se lanzó al mercado en 2014 enfocada al diseño en los dispositivos móviles con sistema operativo Android, aplicándose posteriormente a la mayoría de los productos de la compañía (Material, 2020). Su finalidad es crear un diseño limpio donde predominen las transiciones y las animaciones, el relleno y los efectos de profundidad proporcionando luminosidad y sombra de manera interactiva. Intenta aproximarse a la realidad guiándose por las leyes físicas, donde los objetos se superpongan unos a otros y no puedan atravesarse.

Para implementar en nuestro proyecto un *Material Design*, haremos uso de Angular Material, una librería de componentes con un diseño *Material Design* gestionado también por Google que permitirá conseguir de manera sencilla y con pocas líneas de código una aplicación vistosa, limpia y útil.

El primer paso será indicar a nuestro proyecto que haremos uso de esta librería. Para ello primeramente se debe descargar mediante el instalador de paquetes NPM comentado anteriormente y gracias al comando "*npm install angular-material*", o bien gracias a Angular CLI mediante el comando "*ng add @angular/material*".

El siguiente paso consiste en añadir dicha librería al proyecto, importándola al fichero *app.module.ts*, que hará que esté disponible para toda la aplicación, tal y como vemos en la Figura 9.

```
import { MaterialModule } from "./material.module";

@NgModule({
  declarations: [],
  imports: [MaterialModule],
})
export class AppModule {}
```

Figura 9: Estructura *app.module.ts*

Una vez introducida la librería en el código, se deben importar los componentes que se utilizarán en el resto del proyecto. Para ello, y de cara a mantener la aplicación lo más estructurada posible, se puede crear un nuevo módulo, en nuestro caso llamado "*material.module.ts*", alojado dentro de la carpeta *src*, en el cual se definen los

componentes que utilizaremos posteriormente y que iremos incluyendo en este fichero a medida que se vayan necesitando. Así bien, cuando sea necesario incluir algún componente de Angular Material en ciertas partes de la aplicación, se incluirá este módulo en dicho componente.

Otro aspecto importante de Angular Material son sus temas. Estos sirven para asignar un color a los componentes sin necesidad de gestionar esta característica en un fichero de estilo *css* de manera inicial. A la hora de instalar esta librería, el sistema pregunta por la asignación de un tema, dándote a elegir entre varias opciones. Al elegir uno de ellos, se crea de manera automática una dependencia en nuestro *angular.json*, donde se incluirá un apartado de *styles* junto a la ruta del tema elegido, como se observa en la Figura 10, pudiéndose cambiar en el momento deseado.

```
"styles":  
  [  
    "./node_modules/@angular/material/prebuilt-themes/indigo-pink.css",  
    "src/styles.css"  
  ]
```

Figura 10: Tema elegido en Angular Material

A la hora de encontrar los diferentes componentes que se desea incluir en el proyecto, será de gran utilidad la página oficial de Angular Material (Angular Material, 2020), en la cual podremos realizar una búsqueda según las necesidades del creador y una pequeña visualización del componente e instrucciones sobre cómo implementarlo.

3.3. Visual Studio Code.

Una vez realizada la correcta instalación de la aplicación Angular inicial, los siguientes pasos se basan en modificar esta para obtener un resultado esperado. Si bien es factible ir creando ficheros y rellenándolos en cualquier editor de texto básico, un IDE (*Integrated Development Environment*) o entorno de desarrollo integrado, facilitará su manejo y añadirá agilidad y comprensión al código y a la estructura del proyecto. Coloquialmente, un IDE se puede describir como un software que ayuda a desarrollar software, ya que otorga una serie de herramientas que pueden ser de gran ayuda o necesidad para el programador en cuestión.

Existen multitud de opciones de terceros que proporcionan un entorno amigable para el desarrollo de aplicaciones de esta índole, pudiéndose desarrollar en aplicaciones independientes, o incluso desde la nube, como es el caso de AWS Cloud9 (Amazon Web Services, 2020), donde únicamente necesitas un navegador para desarrollar tus códigos con el beneficio de la portabilidad de tu entorno de trabajo.

En nuestro caso, haremos uso del IDE de escritorio desarrollado por Microsoft de código abierto y gratuito lanzado en 2015, Visual Studio Code. Se trata de un editor de código compatible con varios lenguajes de programación y con características propias que lo sitúan globalmente en la cima de los entornos de desarrollo, tal y como podemos observar en la Figura 11 y según Google Trends (Google Trends, 2021).

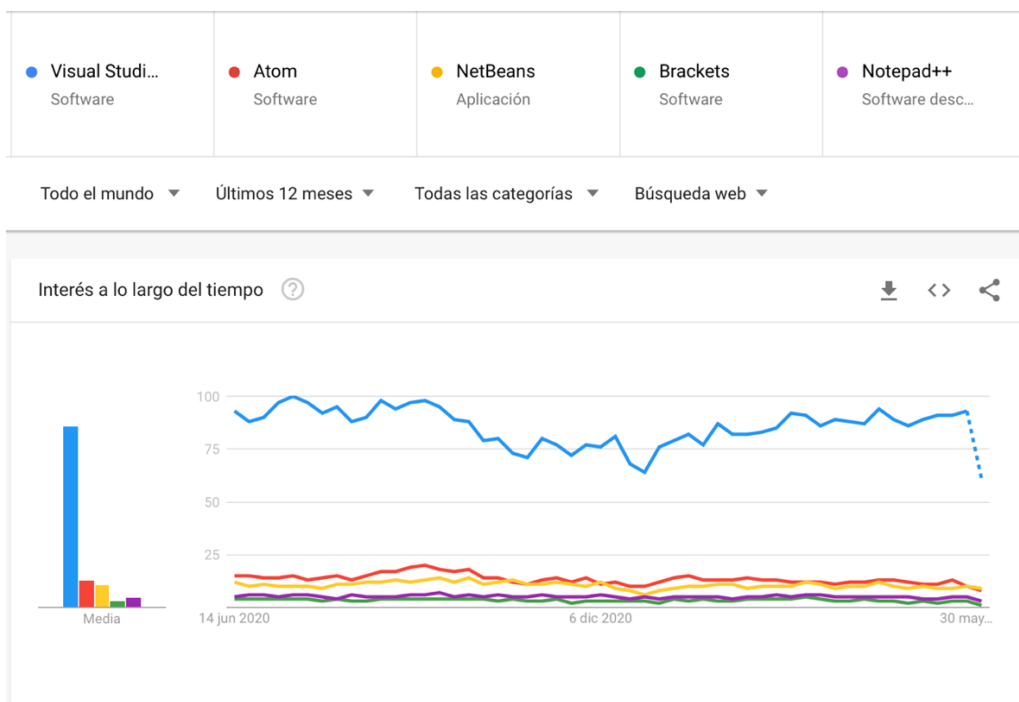


Figura 11: Comparación IDEs último año

Una de sus características más relevantes es *IntelliSense*, la cual ayuda a estructurar el código resaltando la sintaxis escrita y proporcionando al desarrollador funciones de autocompletar el código en base a variables o definiciones entre otros (Microsoft, 2016).

Además, consta de un depurador diseñado para ayudar a encontrar errores en el código y una biblioteca de extensiones que nos pueden ayudar a incrementar la productividad de nuestro trabajo, instalando y desinstalando, aquellas que nos aporten algún beneficio, o que bien, ya no sean necesarias (Bartolomé Sintés. M, 2020).

En comparación con otros entornos de trabajo, Visual Studio Code posee un panel lateral que contiene la estructura del proyecto, organizado en diferentes carpetas y ficheros, permitiendo mantener abiertos cualesquiera de ellos de manera paralela y recibiendo información sobre nuestro código, como qué tipo de variable estamos tratando o cualquier posible error tipográfico o de lógica operacional.

3.4. GIT.

Otra de las opciones que nos aporta Visual Studio Code, es la gestión del código mediante GIT. Este software de control de versiones distribuido de licencia abierta fue diseñado por Linus Torvalds (desarrollador del kernel de Linux) en 2007 con la intención de llevar un registro de los cambios producidos en ficheros de un proyecto donde varias personas pueden realizar cambios en un mismo repositorio sin la necesidad de compartir una misma red.

Proporciona un alto rendimiento debido a la ramificación de versiones, su fusión y comparación con diversas ramas además de la confirmación de nuevos cambios, y es flexible a la hora de obtener varios flujos de trabajo de desarrollo no lineal, como puede ser un gran proyecto con protocolos o sistemas muy diferentes.

Actualmente, al iniciar un proyecto en Angular mediante el comando descrito anteriormente, se iniciará un repositorio GIT en el cual podremos ir guardando nuestros cambios en el código en diferentes versiones. Aunque para usuarios inexpertos GIT puede resultar un poco tedioso, son pocos los comandos que necesitaremos entender para gestionar correctamente las versiones de nuestro código.

Mediante "*git commit*" podemos confirmar los cambios realizados en nuestros ficheros, en el cual podremos indicar una descripción asociada a aquellos cambios que queremos salvar, y que nos permitirá más adelante y en caso de necesitar retroceder, saber en qué punto se encontraba el proyecto.

En la gestión que hace Visual Studio Code de esta herramienta, aparece directamente una entrada de texto donde podremos comentar nuestros cambios y automáticamente realizará esta acción. Acto seguido, nuestra intención será subir esos cambios al repositorio distribuido donde otros compañeros podrán obtener nuestro código para, por ejemplo, seguir editándolo. Para ello, el comando a utilizar sería "*git push*", aunque, al igual que con el comando anterior, la herramienta de Microsoft nos proporciona un botón para realizar dicha acción.

Por otra parte, si somos nosotros quienes deseamos obtener el código producido por otro compañero, realizaremos un "*git pull*", que automáticamente descargaría estos datos del repositorio (Atlassian, 2021). Este sistema nos permitirá mantener un control exhaustivo de nuestro código, pudiendo además compartirlo con los demás integrantes del proyecto final que explicamos en el capítulo introductorio de este documento.

4. Captura de requisitos y análisis.

A lo largo de este cuarto capítulo se verán aspectos clave necesarios para el desarrollo de la aplicación. En concreto, se realiza un estudio de aquellos requisitos indispensables que serán de obligado cumplimiento por parte de la aplicación, definidos estos inicialmente por un usuario final prototipo y continuados por la lógica de usuario que todos desarrollamos al hacer uso de la web.

En particular, se capturarán aquellos requisitos que debe cumplir nuestro sistema para llegar a brindar aquella finalidad para la que fue diseñado, dando lugar, acto seguido, a un análisis del sistema mediante casos de uso.

A través de los requisitos y del modelado mediante los casos de uso, mostrados estos de manera conjunta en un Diagrama de Casos de Uso, definido en el Lenguaje de Modelado Unificado o UML (*Unified Modelling Language*), se consigue recoger las principales características de la aplicación, ayudando a comprender de manera global el sistema y a su posterior desarrollo cumpliendo estos mismos, siempre que hayan sido fijados de manera correcta y sean posibles.

4.1. Requisitos.

En primer lugar, se detallan cada uno de los requisitos de la aplicación software que se desean implementar en este desarrollo, obteniendo una finalidad que satisfaga al usuario final. Los requisitos se dividen, principalmente, en dos tipos, funcionales y no funcionales.

4.1.1. Requisitos funcionales.

Los requisitos funcionales definen el propio comportamiento de la aplicación. Indican cómo tiene que reaccionar, o cómo no, el software ante ciertos eventos. Estos requisitos han de ser lo más precisos posibles, ya que la lógica no se aplica de la misma manera para diferentes personas, pudiendo ocasionar retrasos temporales en la obtención de la aplicación, entre otros problemas. La especificación de este tipo de requerimientos debe estar completa y ser compacta, incluyendo todos los servicios solicitados por el cliente y sin definiciones inexactas o contradictorias.

Después de haber estudiado las necesidades del proyecto, y aclarando la estructura jerarquizada de este mediante tres tipos de usuarios definidos por un rol (Super Administrador, Administrador y Trabajador), se definen los siguientes requisitos funcionales:

RF1: El sistema debe permitir a los usuarios con un rol de Super Administrador una gestión CRUD (*Create, Read, Update, Delete*) de los diferentes usuarios del sistema, siendo estos Super Administradores, Administradores y Trabajadores.

RF2: El sistema debe permitir a los usuarios con un rol de Administrador una gestión CRUD (*Create, Read, Update, Delete*) de los diferentes usuarios de su equipo particular, siendo estos únicamente su mismo perfil de Administrador y su equipo de Trabajadores.

RF3: El sistema debe permitir a los diferentes usuarios (Super Administradores, Administradores y Trabajadores) la gestión de los múltiples sensores de la aplicación, colocándolos en la posición indicada.

RF4: El sistema debe permitir a los diferentes usuarios (Super Administradores, Administradores y Trabajadores) observar la información proporcionada por los sensores previamente colocados.

RF5: El sistema debe permitir a los usuarios (Super Administradores, Administradores y Trabajadores) observar informes históricos que contengan información relacionada con los sistemas de información instalados.

RF6: El sistema debe permitir al usuario (Super Administrador, Administrador y Trabajador) seleccionar una imagen que él desee sobre la cual posicionar los diferentes sensores.

RF7: El sistema debe permitir a los usuarios (Super Administrador, Administrador y Trabajador) elegir los valores máximos y mínimos válidos para cada uno de los sensores.

RF8: El sistema debe avisar a los usuarios (Super Administrador, Administrador y Trabajador) de las alertas que se produzcan en los diferentes sensores.

Identificador	RF1.1
Nombre	Crear usuario
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe permitir a los usuarios con un rol de Super Administrador crear diferentes usuarios del sistema.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 1: RF1.1 Crear Usuario

Identificador	RF1.2
Nombre	Ver usuario
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe permitir a los usuarios con un rol de Super Administrador ver los datos de los diferentes usuarios del sistema.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 2: RF1.2 Ver Usuario

Identificador	RF1.3
Nombre	Modificar usuario
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe permitir a los usuarios con un rol de Super Administrador modificar los datos de los diferentes usuarios del sistema.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 3: RF1.3 Modificar Usuario

Identificador	RF1.4
Nombre	Eliminar usuario
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe permitir a los usuarios con un rol de Super Administrador eliminar a usuarios del sistema.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 4: RF1.4 Eliminar Usuario

Identificador	RF2.1
Nombre	Crear Trabajador
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe permitir a los usuarios con un rol de Administrador crear Trabajadores del sistema.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 5: RF2.1 Crear Trabajador

Identificador	RF2.2
Nombre	Ver trabajador
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe permitir a los usuarios con un rol de Administrador ver Trabajadores del sistema.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 6: RF2.2 Ver Trabajador

Identificador	RF2.3
Nombre	Modificar trabajador
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe permitir a los usuarios con un rol de Administrador modificar Trabajadores del sistema.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 7: RF2.3 Modificar Trabajador

Identificador	RF2.4
Nombre	Eliminar trabajador
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe permitir a los usuarios con un rol de Administrador eliminar Trabajadores del sistema.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 8: RF2.4 Eliminar Trabajador

Identificador	RF3
Nombre	Colocar sensores
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe permitir a los usuarios gestionar los diferentes sensores proporcionados por el sistema colocándolos a su gusto.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 9: RF3 Colocar sensores

Identificador	RF4
Nombre	Observar sensores
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe permitir a los usuarios observar la información proveniente de los diferentes sensores previamente colocados.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 10: RF4 Observar sensores

Identificador	RF5
Nombre	Generar históricos
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe permitir a los usuarios observar la información en un rango de tiempo definido por el usuario proveniente de los diferentes sistemas de información preinstalados.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 11: RF5 Generar históricos

Identificador	RF6
Nombre	Seleccionar imagen
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe permitir a los usuarios seleccionar cualquier imagen que poseen sobre la cual situar los diferentes sensores contratados, de manera que se pueda generar un HMI cercano a la realidad de cada uno de los usuarios.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Media

Tabla 12: RF6 Seleccionar imagen

Identificador	RF7
Nombre	Introducir valores válidos
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe permitir a los usuarios seleccionar los valores máximos y mínimos válidos para cada uno de los sensores.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 13: RF7 Introducir valores válidos

Identificador	RF8
Nombre	Recibir alertas
Tipo	Requisito funcional
Versión	1.0
Descripción	El sistema debe avisar a los usuarios cuando se han sobrepasado los límites introducidos como válidos para cada uno de los sensores.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 14: RF8 Recibir alertas

4.1.2. Requisitos no funcionales.

Los requisitos no funcionales, por el contrario, no definen las funciones específicas que ha de proporcionar el sistema, sino que se encargan de manejar las propiedades del sistema que surgen de la necesidad del usuario y que no están directamente relacionadas con la funcionalidad del sistema. Recogen, por ejemplo, restricciones debidas a políticas propias, de privacidad, restricciones en el presupuesto o necesidades específicas de equipos, espacios temporales o rendimiento.

Se definen para este proyecto los siguientes requisitos no funcionales:

RNF1: La interfaz del sistema ha de ser visible a cierta distancia, teniendo sus elementos, tales como botones, enlaces o paneles, un tamaño adecuado que permita su tratamiento de forma cómoda a usuarios con capacidades visuales medias.

Se trata de lograr una aplicación con una interfaz de usuario visual y ordenada, de manera que tenga una lógica intuitiva y fácil para cualquier tipo de usuario. Para ello, se hace uso de Angular Material, tal y como hemos definido en el apartado 3.2, asignando un tema principal y varios de sus componentes. Para cumplir este cometido, generamos un componente denominado *header*, el cual nos acompaña en toda la navegación como cabecera de la aplicación, indicando opciones y diferentes enlaces y componentes de la web.

RNF2: El sistema debe proteger la privacidad de los usuarios, almacenando adecuadamente sus datos personales. Para ello, se cifrarán las conexiones entre nuestra interfaz y la base de datos en la cual se almacenarán los datos de acceso de cada uno de los usuarios de manera cifrada para que sean ininteligibles por ciberdelincuentes que intenten vulnerar la integridad de los usuarios.

RNF3: El sistema no debe permitir el acceso a personas no autorizadas, garantizando este servicio mediante algún sistema de autenticación previo al uso del software.

Al no tratarse de una web pública y accesible a todo el mundo, lo primero que se debe implementar será el sistema de login de nuestros usuarios. Como comentábamos anteriormente, se definen tres tipos de usuarios caracterizados cada uno de ellos mediante un rol. El rol de Super Administrador está únicamente destinado a los desarrollos de la aplicación, es decir, nosotros, teniendo los privilegios de gestionar todos los usuarios, además de las propias funcionalidades del sistema. El rol de Administrador obtiene menos privilegios, siendo capaz únicamente de gestionar los usuarios con rol de Trabajador asociados a su entorno de trabajo. El rol de Trabajador únicamente podrá gestionar sus datos de acceso, además de disfrutar de las funcionalidades del sistema.

RNF4: El sistema debe poder integrarse en cualquier navegador actual utilizado por usuarios normales, tales como Google Chrome, Mozilla Firefox o Safari. Aunque la mayoría de *frameworks* y librerías ya contemplan esta importante característica, generaremos un estilo capaz de visualizarse de forma adecuada en la mayoría de los navegadores.

RNF5: El sistema debe tener un coste de instalación y mantenimiento mínimo. Se busca una integración sin coste capaz de gestionarse y actualizarse sin necesidad de un nuevo desembolso por parte de los creadores.

RNF6: La interfaz gráfica del sistema ha de ser, en la medida de lo posible, responsiva permitiendo una correcta visualización en dispositivos móviles. Para ello, se hará uso de componentes responsivos adaptando la visualización del contenido a todo tipo de dimensiones de pantallas.

RNF7: El sistema ha de estar preparado, en la medida de lo posible, para posibles actualizaciones en versiones venideras. Es lógico pensar que la aplicación es susceptible a cambios y estos no deben suponer un coste mayor en tiempo y recursos.

RNF8: El sistema ha de avisar al usuario de cualquier fallo relacionado con la utilización errónea de la aplicación. A lo largo de la interacción del usuario con el sistema, es posible que sucedan errores de conexión o de cualquier otro tipo. Ha de estar preparada para avisar al usuario de estos fallos y de cómo solventarlos si es posible.

Identificador	RNF1
Nombre	Interfaz sencilla
Tipo	Requisito no funcional
Versión	1.0
Descripción	La interfaz del sistema ha de ser visible a cierta distancia, teniendo sus elementos, tales como botones, enlaces o paneles, un tamaño adecuado que permita su tratamiento de forma cómoda a usuarios con capacidades visuales medias.
Prioridad	Alta
Urgencia	Media
Estabilidad	Media

Tabla 15: RNF1 Interfaz sencilla

Identificador	RNF2
Nombre	Protección de datos
Tipo	Requisito no funcional
Versión	1.0
Descripción	El sistema ha de ser seguro ante posibles fugas de datos personales de usuarios registrados en el sistema, así como frente a su modificación sin permisos válidos.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 16: RNF2 Protección de datos

Identificador	RNF3
Nombre	Autenticación
Tipo	Requisito no funcional
Versión	1.0
Descripción	El sistema no debe permitir el acceso a usuarios no autorizados mediante algún control inicial, como el uso de un identificador y una clave para acceder a la plataforma.
Prioridad	Alta
Urgencia	Alta
Estabilidad	Alta

Tabla 17: RNF3 Autenticación

Identificador	RNF4
Nombre	Navegadores compatibles
Tipo	Requisito no funcional
Versión	1.0
Descripción	El sistema será capaz de funcionar en cualquier navegador actual, como, por ejemplo, Google Chrome, Safari, Mozilla Firefox, Edge u Opera.
Prioridad	Alta
Urgencia	Media
Estabilidad	Alta

Tabla 18: RNF4 Navegadores compatibles

Identificador	RNF5
Nombre	Coste mínimo
Tipo	Requisito no funcional
Versión	1.0
Descripción	El sistema ha de conllevar un coste de mantenimiento e instalación mínimo, facilitando su implementación en cualquier entorno.
Prioridad	Alta
Urgencia	Media
Estabilidad	Baja

Tabla 19: RNF5 Coste mínimo

Identificador	RNF6
Nombre	Diseño responsivo
Tipo	Requisito no funcional
Versión	1.0
Descripción	El sistema debe proporcionar una interfaz responsiva que se adecúe a cualquier tipo de pantalla, incluyendo dispositivos móviles.
Prioridad	Alta
Urgencia	Media
Estabilidad	Alta

Tabla 20: RNF6 Diseño responsivo

Identificador	RNF7
Nombre	Versiones posteriores
Tipo	Requisito no funcional
Versión	1.0
Descripción	El sistema ha de estar preparado para soportar una nueva versión de manera sencilla.
Prioridad	Media
Urgencia	Baja
Estabilidad	Baja

Tabla 21: RNF7 Versiones posteriores

Identificador	RNF8
Nombre	Manejo de errores.
Tipo	Requisito no funcional
Versión	1.0
Descripción	El sistema ha de proporcionar información concisa para el usuario ante problemas causados por el uso inadecuado de la aplicación.
Prioridad	Baja
Urgencia	Baja
Estabilidad	Baja

Tabla 22: RNF8 Manejo de errores

4.2. Diagrama de Casos de Uso.

A través del lenguaje de modelado UML se estandariza el diagrama de casos de uso, en el cual se representan los procesos desarrollados o esperados por los actores del sistema (Universidad de Salamanca, 2018). Este diagrama permite obtener de manera clara y con un simple vistazo, qué objetivos se han de cumplir, marcando los pasos a seguir para desarrollar nuestro software.

En la Figura 12 se puede observar un esquema de las funcionalidades esperadas del sistema, así como de los únicos actores encargados y con la posibilidad de realizar dichos pasos.

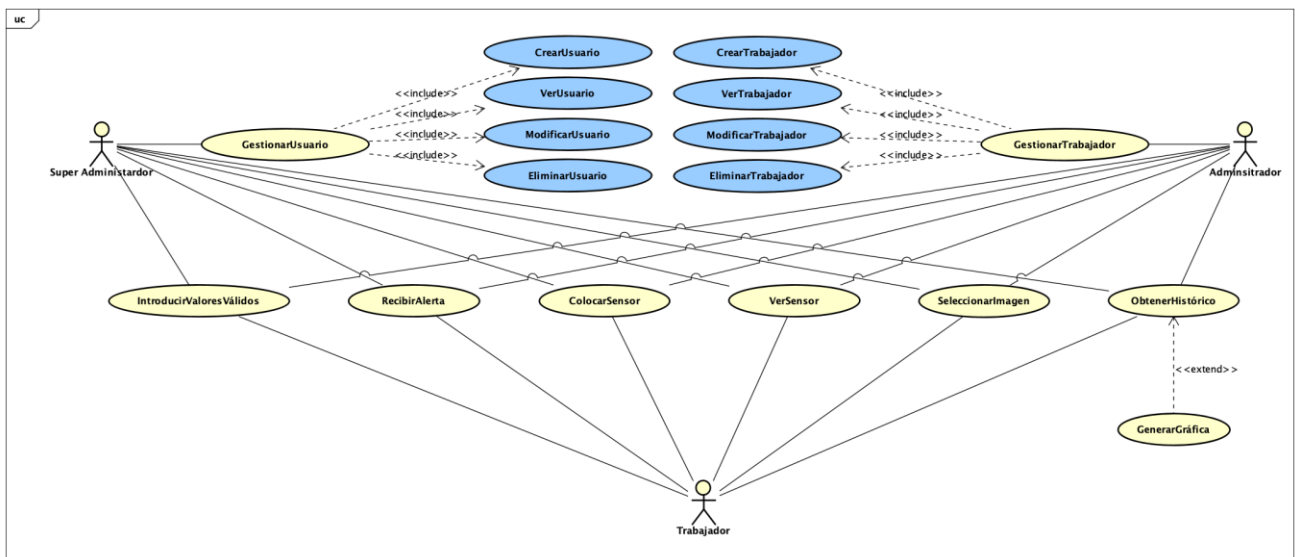


Figura 12: Diagrama de Casos de Uso

4.3. Descripción de Casos de Uso.

Los casos de uso permiten profundizar en los requisitos funcionales del sistema, describiendo la interacción del usuario con el sistema para realizar alguna de las funcionalidades descritas. Como en nuestro sistema se definen tres tipos de actores: Super Administradores, Administradores y Trabajadores, se indica quién de ellos puede realizar cada una de las acciones.

4.3.1. Caso de Uso: GestionarUsuario.

Identificador	GestionarUsuario	
Versión	1.0	
Fecha	16/06/2021	
Descripción	El Super Administrador podrá crear, visualizar, modificar y eliminar tanto Administradores como Trabajadores del sistema.	
Actores	Super Administrador	
Precondición	El sistema ha reconocido al usuario como Super Administrador.	
Secuencia o flujo normal	PASO	Acción
	1	El actor elige la opción "GestionarUsuario".
	2	El sistema presenta al actor las opciones disponibles para gestionar los perfiles de los usuarios.
	3ª	El actor elige la opción "CrearUsuario"
	4ª	<<include>> CrearUsuario
	3b	El actor elige la opción "VerUsuario"
	4b	<<include>> VerUsuario
	3c	El actor elige la opción "ModificarUsuario"
	4c	<<include>> ModificarUsuario
	3d	El actor elige la opción "EliminarUsuario"
4d	<<include>> EliminarUsuario	
Postcondición	Se ha producido el caso de uso seleccionado	
Flujo alternativo	PASO	Acción
	1-4	El actor elige cancelar el proceso y el caso de uso finaliza sin ninguna Postcondición.
Frecuencia esperada	Media	
Importancia	Alta	
Urgencia	Alta	

Tabla 23: Caso de Uso: GestionarUsuario

4.3.1.1. Caso de Uso: CrearUsuario.

Identificador	CrearUsuario	
Versión	1.0	
Fecha	23/06/2021	
Descripción	El Super Administrador podrá crear un nuevo usuario en el sistema, definiendo los datos necesarios de dicho usuario, e indicando también el rol que tendrá el nuevo usuario.	
Actores	Super Administrador	
Precondición	El sistema ha reconocido al usuario como Super Administrador en el proceso de acceso y autenticación al sistema.	
Secuencia o flujo normal	PASO	Acción
	1	El actor elige la opción "CrearUsuario".
	2	El sistema presenta al Super Administrador un formulario a rellenar con los campos necesarios para dar de alta un nuevo usuario.
	3	El Super Administrador rellena todos los campos necesarios para dar de alta un nuevo usuario. Deberá introducir su nombre, apellidos, un nombre de usuario único para el posterior acceso al sistema, una contraseña lo suficientemente segura para que sea indescifrable por cualquier agente exterior y el rol asociado.
	4	El Super Administrador confirma el envío del formulario para la creación del nuevo usuario.
	5	El sistema crea un nuevo usuario de manera satisfactoria.
Postcondición	Se ha registrado un nuevo usuario en el sistema.	
Flujo Alternativo	PASO	Acción
	2-4	El Super administrador decide cancelar el proceso de CrearUsuario seleccionando la opción de cancelar y el Caso de Uso finaliza sin lograr la postcondición.
Frecuencia esperada	Media	
Importancia	Alta	
Urgencia	Alta	

Tabla 24: Caso de Uso: CrearUsuario

4.3.1.2. Caso de Uso: VerUsuario.

Identificador	VerUsuario	
Versión	1.0	
Fecha	23/06/2021	
Descripción	El Super Administrador podrá visualizar todos los usuarios registrados en el sistema, tanto Super Administradores y Administradores como Trabajadores.	
Actores	Super Administrador	
Precondición	El sistema ha reconocido al usuario como Super Administrador en el proceso de acceso y autenticación al sistema.	
Secuencia o flujo normal	PASO	Acción
	1	El actor elige la opción "VerUsuario".
	2	El sistema presenta al Super Administrador una lista completa de los distintos usuarios registrados en el sistema.
Postcondición	Se ha producido el caso de uso seleccionado, obteniendo una lista de usuarios registrados.	
Flujo Alternativo	-	
Frecuencia esperada	Media	
Importancia	Alta	
Urgencia	Alta	

Tabla 25: Caso de Uso: VerUsuario

4.3.1.3. Caso de Uso: ModificarUsuario.

Identificador	ModificarUsuario	
Versión	1.0	
Fecha	23/06/2021	
Descripción	El Super Administrador podrá editar los datos relacionados con cada uno de los usuarios registrados en el sistema, tanto Super Administradores y Administradores como Trabajadores.	
Actores	Super Administrador	
Precondición	El sistema ha reconocido al usuario como Super Administrador en el proceso de acceso y autenticación al sistema.	
Secuencia o flujo normal	PASO	Acción
	1	El actor elige la opción "ModificarUsuario".
	2	El sistema presenta al Super Administrador una lista completa de los distintos usuarios registrados en el sistema.
	3	El Super Administrador elige uno de estos usuarios.
	4	El sistema presenta al Super Administrador un formulario con los datos del usuario como campos editables.
	5	El Super Administrador modifica los campos que desea y pulsa en aceptar. Concretamente, deberá introducir de manera correcta el nombre, apellidos, nombre de usuario, contraseña y rol asociado, al igual que en el proceso de creación previamente indicado.
	6	El sistema formaliza la modificación del usuario, completándose el caso de uso exitosamente.
Postcondición	Se ha modificado la información de un usuario previamente registrado en el sistema.	
Flujo Alternativo	PASO	Acción
	1-5	El Super Administrador presiona en el botón de cancelación y el Caso de Uso finaliza sin lograr la postcondición.
Frecuencia esperada	Media	
Importancia	Alta	
Urgencia	Alta	

Tabla 26: Caso de Uso: ModificarUsuario

4.3.1.4. Caso de Uso: EliminarUsuario.

Identificador	EliminarUsuario	
Versión	1.0	
Fecha	23/06/2021	
Descripción	El Super Administrador podrá eliminar cualquier usuario del sistema, siendo esta acción irrevocable.	
Actores	Super Administrador	
Precondición	El sistema ha reconocido al usuario como Super Administrador en el proceso de acceso y autenticación al sistema.	
Secuencia o flujo normal	PASO	Acción
	1	El actor elige la opción "EliminarUsuario".
	2	El sistema presenta al Super Administrador una lista completa de los distintos usuarios registrados en el sistema.
	3	El Super Administrador elimina un usuario del sistema presionando en el botón colocado a continuación de cada uno de los usuarios.
	4	El sistema presenta al Super Administrador un aviso para cerciorarse de que desea realizar la acción de borrado.
	5	El Super Administrador confirma la acción pulsando en aceptar.
	6	El sistema elimina de manera permanente el usuario seleccionado de la aplicación.
Postcondición	Se ha eliminado un usuario previamente registrado en el sistema.	
Flujo Alternativo	PASO	Acción
	1-5	El Super Administrador presiona en el botón de cancelación y el Caso de Uso finaliza sin lograr la postcondición.
Frecuencia esperada	Media	
Importancia	Alta	
Urgencia	Alta	

Tabla 27: Caso de Uso: EliminarUsuario

Esta gestión CRUD por parte de los Super Administradores es equivalente a la realizada por los Administradores del sistema, escalando los usuarios destino (siempre un escalón por debajo en la escala de privilegios). Es decir, los usuarios que obtengan un rol en el sistema de Administrador únicamente podrán gestionar los usuarios con rol de Trabajador.

Una vez estructurada la gestión de los diferentes usuarios del sistema, comenzamos con la funcionalidad de la aplicación, proporcionando así una interfaz interactiva con el usuario para la gestión de sensores y componentes situados *in-situ* en su lugar de trabajo real.

4.3.2. Caso de Uso: SeleccionarImagen.

Identificador	SeleccionarImagen	
Versión	1.0	
Fecha	25/06/2021	
Descripción	Cualquier Actor del sistema podrá seleccionar una imagen deseada como entorno de trabajo sobre la cual trabajar.	
Actores	Super Administrador, Administrador, Trabajador	
Precondición	El sistema ha reconocido al Actor como usuario válido en el proceso de acceso y autenticación al sistema.	
Secuencia o flujo normal	PASO	Acción
	1	El Actor elige la opción "SeleccionarImagen".
	2	El sistema presenta al usuario un explorador de archivos locales para seleccionar la imagen deseada como entorno de trabajo.
	3	El Actor escoge la imagen concreta y pulsa en aceptar.
	4	El sistema presenta al usuario la imagen seleccionada.
Postcondición	Se ha obtenido una imagen sobre la cual trabajar.	
Flujo Alternativo	PASO	Acción
	1-3	El Actor decide cancelar el proceso de selección y pulsa en cancelar y el Caso de Uso finaliza sin lograr la postcondición.
Frecuencia esperada	Alta	
Importancia	Alta	
Urgencia	Alta	

Tabla 28: Caso de Uso: SeleccionarImagen

4.3.3. Caso de Uso: ColocarSensores.

Identificador	ColocarSensores	
Versión	1.0	
Fecha	25/06/2021	
Descripción	Cualquier Actor del sistema podrá colocar los diferentes sensores que haya contratado en un entorno de diseño.	
Actores	Super Administrador, Administrador, Trabajador	
Precondición	El sistema ha reconocido al Actor como usuario válido en el proceso de acceso y autenticación al sistema.	
Secuencia o flujo normal	PASO	Acción
	1	El Actor elige la opción "ColocarSensores".
	2	El sistema presenta al usuario un entorno de diseño en el cual aparecerán los diferentes sensores.
	3	El Actor coloca cada uno de los diferentes sensores en la posición que estime oportuna.
	4	El sistema muestra los sensores en la situación previamente seleccionada por el usuario.
Postcondición	Se ha gestionado la posición de los diferentes sensores dentro del entorno de trabajo.	
Flujo Alternativo	PASO	Acción
	1-3	El Actor decide eliminar un sensor del sistema y presiona un botón que desactivará de manera temporal ese sensor hasta que se desee volver a colocarlo.
Frecuencia esperada	Alta	
Importancia	Alta	
Urgencia	Alta	

Tabla 29: Caso de Uso: ColocarSensores

4.3.4. Caso de Uso: VerSensores.

Identificador	VerSensores	
Versión	1.0	
Fecha	25/06/2021	
Descripción	Cualquier Actor del sistema podrá observar los detalles proporcionados por los diferentes sensores que haya contratado y colocado en un entorno de diseño.	
Actores	Super Administrador, Administrador, Trabajador	
Precondición	El sistema ha reconocido al Actor como usuario válido en el proceso de acceso y autenticación al sistema.	
Secuencia o flujo normal	PASO	Acción
	1	El Actor elige la opción "VerSensores".
	2	El sistema presenta al usuario un entorno de diseño en el cual aparecerán los diferentes sensores previamente colocados en las posiciones adecuadas.
	3a	El Actor selecciona cada sensor obteniendo una descripción de los valores actuales que proporcionan dichos sensores.
	3b	El Actor desea ver los valores actuales de todos los sensores colocados y selecciona la opción de visualizar todos al mismo tiempo.
	4	El sistema presenta al usuario la información requerida de los sensores indicados.
Postcondición	Se ha observado la información proporcionada por los diferentes sensores dentro del entorno de trabajo.	
Flujo Alternativo	El Actor decide cancelar el proceso de visualización y pulsa en cancelar y el Caso de Uso finaliza sin lograr la postcondición.	
Frecuencia esperada	Alta	
Importancia	Alta	
Urgencia	Alta	

Tabla 30: Caso de Uso: VerSensores

4.3.5. Caso de Uso: ObtenerHistórico.

Identificador	ObtenerHistórico	
Versión	1.0	
Fecha	25/06/2021	
Descripción	Cualquier Actor del sistema podrá generar un informe detallado de la información obtenida por los sensores del sistema entre dos fechas en un formato amigable para el usuario final.	
Actores	Super Administrador, Administrador, Trabajador	
Precondición	El sistema ha reconocido al Actor como usuario válido en el proceso de acceso y autenticación al sistema.	
Secuencia o flujo normal	PASO	Acción
	1	El Actor elige la opción "Obtener Histórico".
	2	El sistema presenta al usuario una interfaz en la cual este podrá indicar los sensores de los cuales obtener sus datos.
	3	El Actor selecciona cada uno de los sensores de los cuales quiere observar sus datos durante un periodo temporal concreto.
	4	El sistema proporciona al Actor un formulario para indicar la fecha de inicio y fin del intervalo temporal.
	5	El Actor introduce la fecha de inicio y fin y selecciona confirmar.
	6	El sistema muestra al usuario la información asociada al espacio temporal indicado.
	7	<<extend>> GenerarGráfica
Postcondición	Se ha presentado la información de un periodo temporal concreto proporcionada por los diferentes sensores seleccionados dentro del entorno de trabajo.	
Flujo Alternativo	PASO	Acción
	1-5	El Actor decide cancelar el proceso y pulsa en cancelar y el Caso de Uso finaliza sin lograr la postcondición.
Frecuencia esperada	Media	
Importancia	Media	
Urgencia	Media	

Tabla 31: Caso de Uso: ObtenerHistórico

4.3.6. Caso de Uso: GenerarGráfica.

Identificador	GenerarGráfica	
Versión	1.0	
Fecha	12/07/2021	
Descripción	Cualquier Actor del sistema podrá generar una gráfica detallada de la información obtenida por los sensores del sistema entre dos fechas en un formato amigable para el usuario final.	
Actores	Super Administrador, Administrador, Trabajador	
Precondición	El sistema ha reconocido al Actor como usuario válido en el proceso de acceso y autenticación al sistema. Además, el usuario ha pedido información histórica acerca de uno de los sensores.	
Secuencia o flujo normal	PASO	Acción
	1	El sistema presenta al usuario la posibilidad de generar una gráfica con los datos obtenidos.
	2	El Actor pulsa sobre el botón destinado a tal efecto.
	3	El sistema proporciona al Actor una gráfica con los datos obtenidos.
Postcondición	Se ha presentado la información de un periodo temporal concreto proporcionada por los diferentes sensores dentro del entorno de trabajo en un formato de gráfica.	
Flujo Alternativo	-	
Frecuencia esperada	Media	
Importancia	Media	
Urgencia	Media	

Tabla 32: Caso de Uso: GenerarGráfica

4.3.7. Caso de Uso: IntroducirValoresVálidos.

Identificador	IntroducirValoresVálidos	
Versión	1.0	
Fecha	12/08/2021	
Descripción	Cualquier Actor del sistema podrá introducir unos valores máximos y mínimos válidos por cada sensor.	
Actores	Super Administrador, Administrador, Trabajador	
Precondición	El sistema ha reconocido al Actor como usuario válido en el proceso de acceso y autenticación al sistema.	
Secuencia o flujo normal	PASO	Acción
	1	El Actor selecciona en uno de los sensores la opción "IntroducirValoresVálidos".
	2	El sistema muestra al Actor un panel en el cual podrá introducir los valores deseados.
	3	El Actor introduce los valores precisos y pulsa aceptar.
	4	El sistema confirma que los valores introducidos se han registrado.
Postcondición	Se han ajustado los valores válidos de cada sensor.	
Flujo Alternativo	-	
Frecuencia esperada	Media	
Importancia	Media	
Urgencia	Media	

Tabla 33: Caso de Uso: IntroducirValoresVálidos

4.3.8. Caso de Uso: RecibirAlerta.

Identificador	RecibirAlerta	
Versión	1.0	
Fecha	12/08/2021	
Descripción	Cualquier Actor del sistema podrá ver las alertas generadas cuando se hayan sobrepasado los límites introducidos como válidos para cada uno de los sensores.	
Actores	Super Administrador, Administrador, Trabajador	
Precondición	El sistema ha reconocido al Actor como usuario válido en el proceso de acceso y autenticación al sistema.	
Secuencia o flujo normal	PASO	Acción
	1	El Actor selecciona la opción "Alertas".
	2	El sistema muestra al Actor un histórico de las alertas generadas en el sistema, relativas a los valores de los sensores, incluyendo la fecha y hora en la que se generó cada alerta.
Postcondición	Se han visualizado las alertas generadas en el sistema relativas a los valores de los sensores.	
Flujo Alternativo	-	
Frecuencia esperada	Media	
Importancia	Media	
Urgencia	Media	

Tabla 34: Caso de Uso: RecibirAlerta

5. Manual de usuario.

Una vez definidos y estudiados los diferentes casos de uso que describen la funcionalidad de la aplicación, se procede en este quinto capítulo a la descripción del funcionamiento de la plataforma web HMI, utilizando para este cometido capturas de pantalla obtenidas de la propia aplicación, sobre las que se apoyará la explicación del uso correcto de la misma.

La finalidad de este capítulo es el de proporcionar un apoyo para aquellos usuarios inexpertos a la hora de familiarizarse con la aplicación, sirviendo, en cualquier caso, como referencia en caso de duda en relación con el correcto funcionamiento de la aplicación.

5.1. Pagina Inicio.

En la Figura 13 podemos observar la pantalla obtenida al acceder a la url (*Uniform Resource Locator*) o localizador de recursos uniforme, es decir, la dirección en la que se servirá nuestra aplicación.

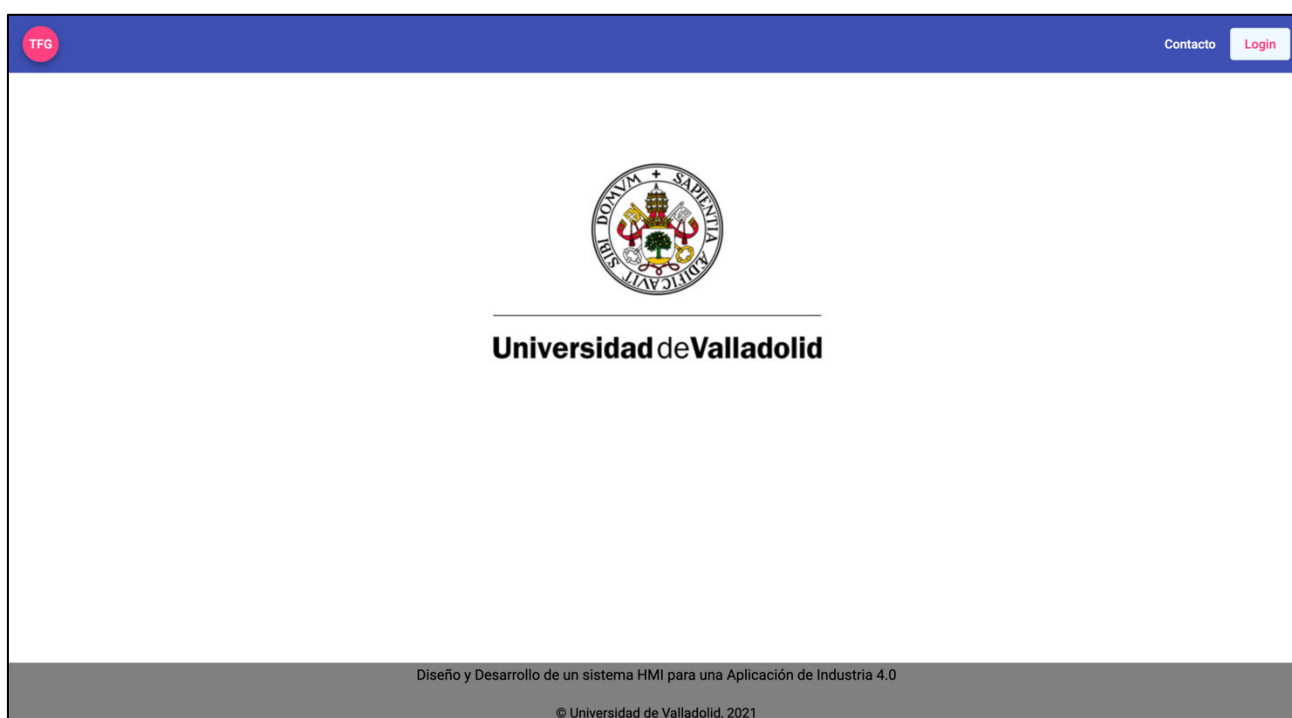
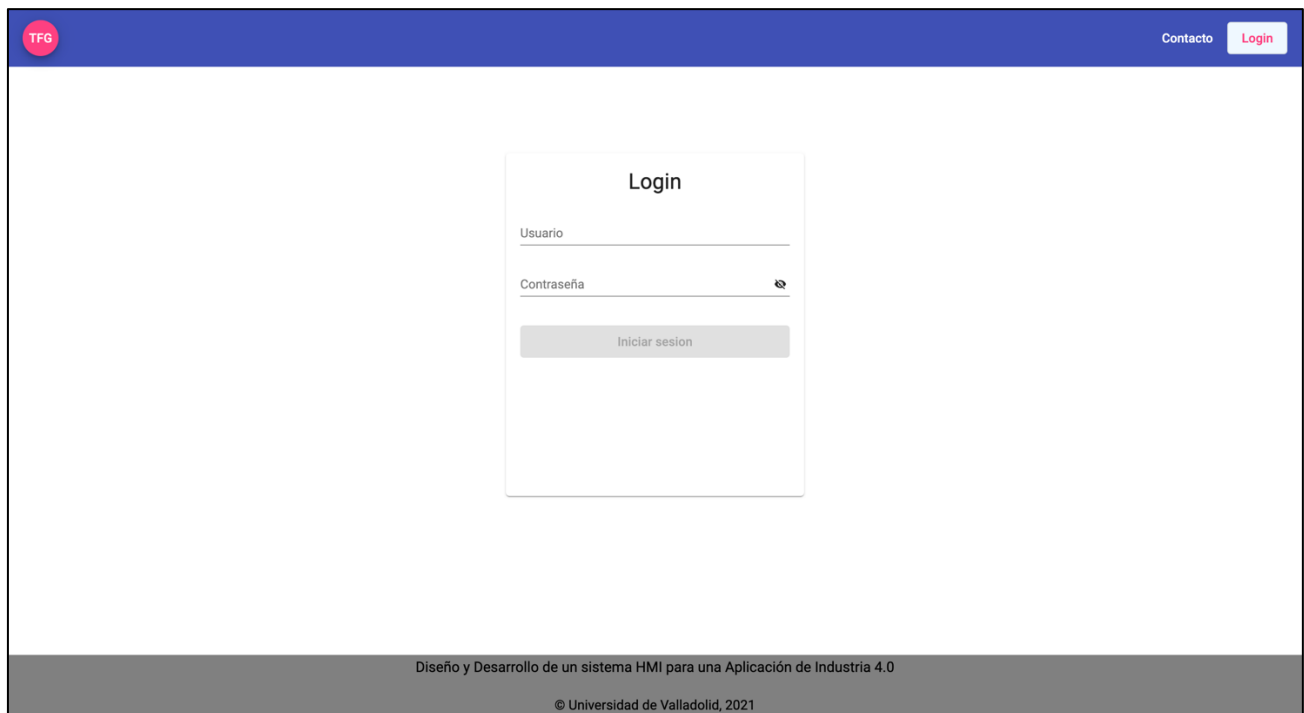


Figura 13: Pantalla de Inicio

Dentro del componente *header*, que nos acompañará en todo nuestro recorrido por los diferentes casos de uso de la aplicación, cabe destacar el botón situado en la esquina superior izquierda, el cual redirige de manera rápida y sencilla a la página principal de la aplicación una vez el usuario haya entrado en la aplicación. Esta acción se completará tras acceder al menú de login situado en la esquina superior derecha y que genera, como resultado, el formulario recogido en la Figura 14.

Por otra parte, se observa un botón de Contacto que nos redirige a una página con información acerca del proyecto y un formulario de contacto para cualquier consulta necesaria por parte del usuario.

Otro elemento a destacar es el *footer*, el cual nos permite proporcionar cualquier información que deseemos transmitir.



The image shows a screenshot of a web application's login page. At the top, there is a blue header bar. On the left side of the header, there is a red circular logo with the white text 'TFG'. On the right side of the header, there are two links: 'Contacto' and 'Login', with 'Login' being a red button. The main content area is white and centered, featuring a login form. The form has a title 'Login' at the top. Below the title, there are two input fields: 'Usuario' and 'Contraseña'. The 'Contraseña' field has a small eye icon to its right. Below the input fields is a grey button with the text 'Iniciar sesion'. At the bottom of the page, there is a dark grey footer bar containing the text 'Diseño y Desarrollo de un sistema HMI para una Aplicación de Industria 4.0' and '© Universidad de Valladolid, 2021'.

Figura 14: Formulario de Login

Como hemos comentado anteriormente, para acceder al sistema se debe completar un formulario de login, visible en la Figura 14. En este momento, el usuario ha de rellenar en el formulario dos campos:

- **Usuario:** Esta entrada deberá ser rellenada con el valor único creado para cada usuario al ser registrado en el sistema. Es obligatorio para acceder al sistema.
- **Contraseña:** Se deberá introducir la contraseña asignada al usuario en el momento del registro. Es obligatoria para acceder al sistema.

Este formulario validará los campos introducidos por el usuario y los comparará con los almacenados en la base de datos correspondiente. A su vez, nos deberá informar del estado de la petición de acceso, indicando, en caso de rechazo, el motivo, tal y como vemos en la Figura 15, no dejando además hacer peticiones al servidor si no se ha completado algún campo del formulario, desactivando el botón de acceso.

The image shows a login form titled "Login". It contains two input fields: "Usuario" with the text "pruebaerror" and "Contraseña" with masked characters ".....". A blue button labeled "Iniciar sesion" is positioned below the fields. A red error message, "Usuario o contraseña incorrectos", is displayed below the button. A small eye icon is visible to the right of the password field, indicating a toggle for password visibility.

Figura 15: Error de Login

En caso de que el acceso a la aplicación haya sido resuelto de manera positiva, el usuario accederá a la pantalla principal de la aplicación.

5.2. Página Principal.

Esta dispondrá de diferentes opciones en función del tipo de usuario que haya accedido al sistema. Podemos ver un ejemplo de lo que será la página principal para un usuario con rol de Super Administrador en la Figura 16, y otro de la página principal de que dispondrá un usuario logado en el sistema con rol de Trabajador en la Figura 17.

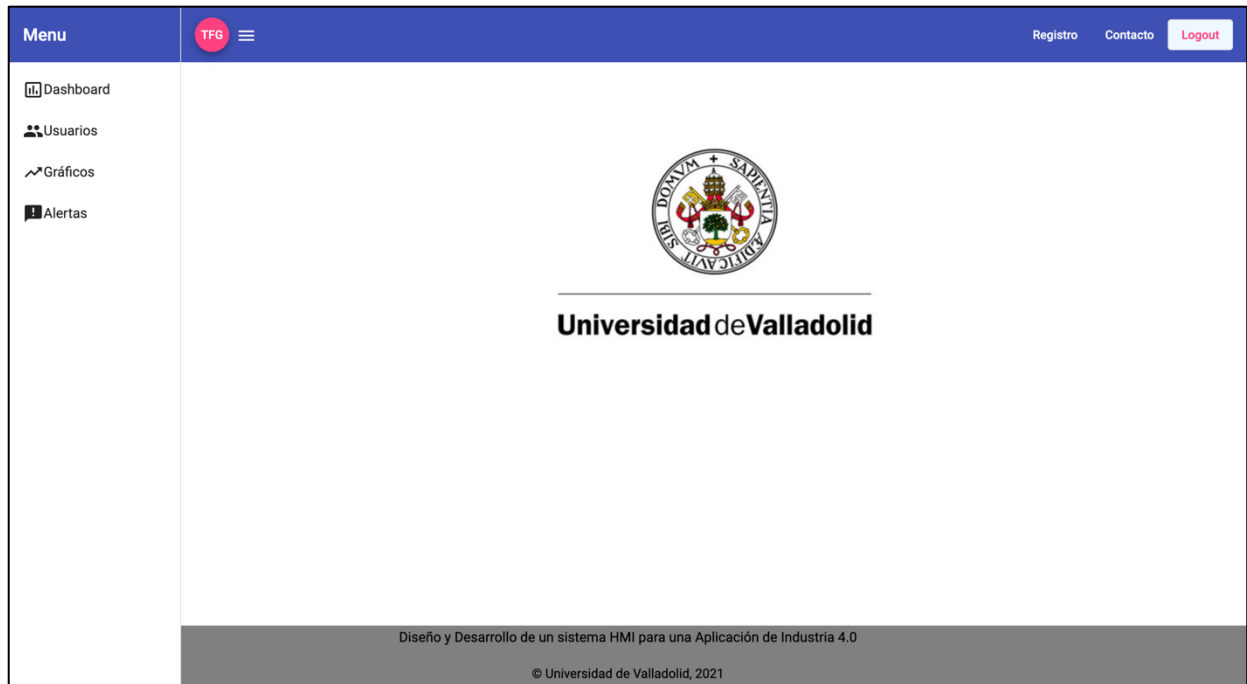


Figura 16: Página principal. Rol Super Administrador



Figura 17: Página principal. Rol Trabajador

En ambos escenarios, se observa un botón al lado derecho del logo de la aplicación, el cual maneja la presencia o ausencia del panel lateral bajo el título de “Menu”. Este menú dispondrá de diferentes opciones para los diferentes tipos de usuarios del sistema.

En primer lugar, y tras observar la Figura 16 y la Figura 17, ambos usuarios disponen de las opciones de “Dashboard”, “Gráficos” y “Alertas”, pero tan solo los usuarios con rol de Super Administrador o Administrador disponen de los privilegios para situarse en el menú de “Usuarios”, visible en la Figura 16 y la Figura 18.



Figura 18: Panel lateral. Rol Administrador

En la parte superior derecha de la Figura 16 y la Figura 17, se puede observar como ambos tipos de usuarios disponen de la opción de cerrar sesión, o “logout” (situándose en lugar del botón de acceso al sistema denominado como login visto, por ejemplo, en la Figura 13), pero tan solo el usuario con rol de Super Administrador podrá registrar nuevos usuarios en el sistema.

Al igual que los Trabajadores, aquellos usuarios con rol de Administrador tampoco están autorizados a registrar nuevos usuarios en el sistema por lo que también tendrán esta opción deshabilitada.

5.3. Página Registro de Usuarios.

Tal y como comentábamos, esta opción está únicamente disponible para los Super Administradores del sistema. Pulsando el botón superior derecho destinado a tal fin, accedemos a una página que presenta un formulario a rellenar por parte del Super Administrador, el cual podemos ver en la Figura 19, para dar de alta a nuevos usuarios, los cuales podrán acceder posteriormente mediante el Usuario y la Contraseña introducidos en este formulario.

TFG

Registro Logout

Registro

Nombre *

Apellidos *

Usuario *

Contraseña *

Rol

Registrarse

Diseño y Desarrollo de un sistema HMI para una Aplicación de Industria 4.0

© Universidad de Valladolid, 2021

Figura 19: Formulario de Registro

Al igual que en el formulario de login mostrado en la Figura 14, el sistema asegura que todos los campos se hayan completado antes de solicitar la petición oportuna, deshabilitando el botón para tal fin e indicando los campos que faltan aún por rellenar, como se puede observar en la Figura 20.

The image shows a registration form titled "Registro". It contains five input fields: "Nombre *" (empty), "Apellidos *" (empty), "Usuario *" (containing "ejemploError"), "Contraseña *" (containing "*****" and a toggle icon), and "Rol" (a dropdown menu with "Administrador" selected). A "Registrarse" button is at the bottom.

Figura 20: Ejemplo de formulario incompleto

Este formulario de registro de usuarios está compuesto por cinco campos descritos a continuación:

- **Nombre:** Se rellenará con el nombre del usuario a registrar. Es obligatorio para registrar un usuario satisfactoriamente.
- **Apellidos:** Se rellenará con los apellidos del usuario a registrar. Es obligatorio para registrar un usuario satisfactoriamente.
- **Usuario:** Se rellenará con el nombre de usuario que adquirirá el usuario a registrar. Ha de ser único ya que será con el cual acceda posteriormente a la aplicación. Es obligatorio para registrar un usuario satisfactoriamente.
- **Contraseña:** Se rellenará con la contraseña que proporcionará acceso único a la aplicación junto con el nombre de usuario. Es obligatorio para registrar un usuario satisfactoriamente.
- **Rol:** Este campo únicamente es posible rellenarlo con uno de los 3 tipos de usuarios del sistema, impidiendo cualquier otro tipo de valor. Es obligatorio para registrar un usuario satisfactoriamente.

De cara a preservar la integridad del usuario, las contraseñas introducidas serán tratadas mediante un mecanismo de *hash* o función resumen, la cual transforma el texto plano introducido en el campo de Contraseña a un bloque de caracteres arbitrarios de longitud fija (Donohue, 2021). Esta función resumen será almacenada en la base de datos, impidiendo que un acceso no autorizado en esta obtenga dichas contraseñas y vulnere la privacidad del usuario.

5.4. Página Contacto.

Esta parte de la aplicación es visible tanto para los usuarios registrados en el sistema, como para aquellos que han llegado a la página principal de la aplicación y desean obtener más información de los servicios que se ofrecen en este desarrollo.

En la Figura 21, se observa esta nueva página, en la cual se encuentran tres campos diferenciados en los cuales se presenta información sobre los desarrolladores, sobre el sistema o aplicación, y, por último, un formulario de contacto en el cual se pueden realizar peticiones o comentarios.

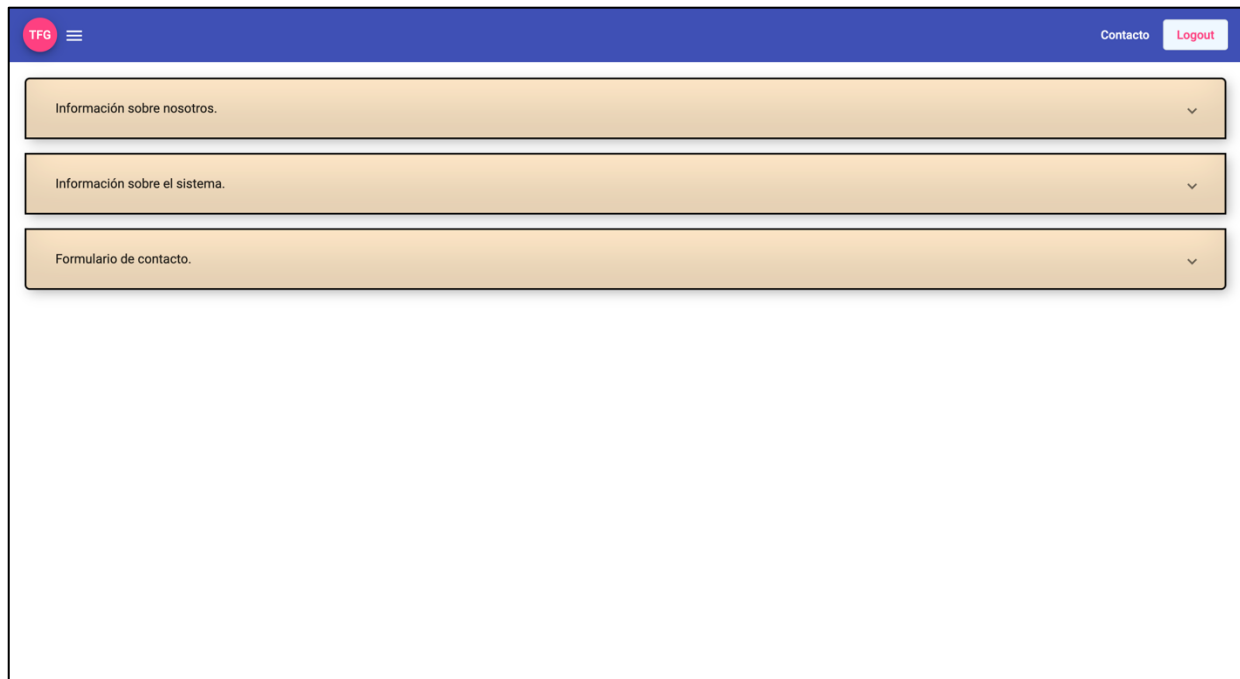


Figura 21: Página de Contacto

Cada una de las tres divisiones da acceso a un panel de expansión, el cual, y tras pulsar sobre él, despliega aquella información que se desee transmitir. Por ejemplo, en el formulario de contacto, se presentan tres campos a rellenar por parte del usuario interesado en contactar con nosotros, visible en la Figura 22.

Información sobre nosotros. ▾

Información sobre el sistema. ▾

Formulario de contacto. ▲

Nombre

Correo

Comentario, queja o sugerencia

✎

Enviar

Figura 22: Formulario de contacto

- **Nombre:** Se pide el nombre de contacto del usuario interesado, de manera que, en una posible respuesta, se pueda tratar al individuo por su nombre.
- **Correo:** Se solicita el correo de contacto para poder establecer una conversación con el usuario en cuestión.
- **Comentario, queja o sugerencia:** En este tercer campo del formulario de contacto se espera que los usuarios lo rellenen de una forma cordial y sincera con aquella información que nos quieran hacer llegar, introduciendo cualquier dato de ayuda que nos posibilite identificar el problema para resolverlo a la mayor brevedad posible. A su vez, es el espacio correcto para notificar cualquier incidencia que se haya podido experimentar en el uso de la aplicación.

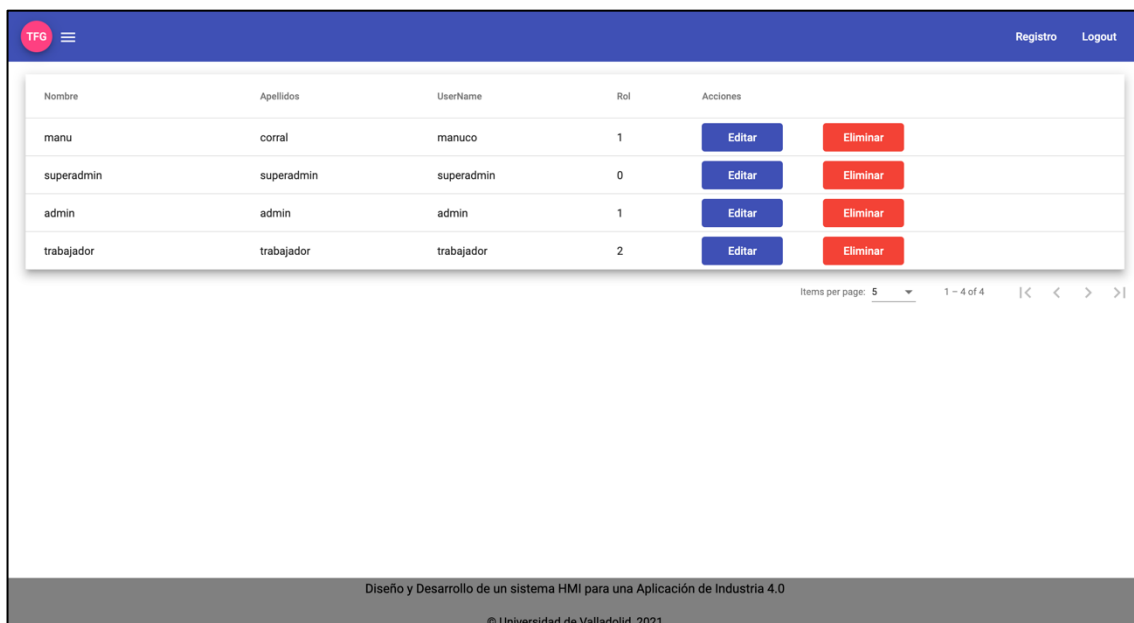
Estos tres campos son obligatorios para poder contactar con los desarrolladores de la aplicación, por lo que no se podrá realizar el envío del formulario hasta que todos hayan sido correctamente cumplimentados.

5.5. Página Usuarios.

Adentrándonos en el panel lateral o *sidebar*, una de las opciones que nos encontramos es el panel de usuario, visible únicamente a Administradores y Super Administradores del sistema.

Este apartado permitirá mantener un control de los usuarios registrados en el sistema, es decir, será desde este lugar desde donde podremos modificar los datos de cualquier usuario registrado, pudiendo incluso llegar a eliminarlos.

Una vez el usuario pulse en este apartado, se muestra una tabla compuesta por los diferentes usuarios registrados, visible en la Figura 23.



Nombre	Apellidos	UserName	Rol	Acciones	
manu	corral	manuco	1	Editar	Eliminar
superadmin	superadmin	superadmin	0	Editar	Eliminar
admin	admin	admin	1	Editar	Eliminar
trabajador	trabajador	trabajador	2	Editar	Eliminar

Items per page: 5 | 1 - 4 of 4 | < > >|

Diseño y Desarrollo de un sistema HMI para una Aplicación de Industria 4.0
© Universidad de Valladolid, 2021

Figura 23: Tabla compuesta por los diferentes usuarios del sistema

Esta tabla muestra cada usuario en una nueva fila, mostrando sus datos personales y el rol asignado en la aplicación. Por motivos de rendimiento y seguridad, se simplifica el envío y la recogida del rol de usuario, asignado el valor 0 a usuarios con rol de Super Administrador, el 1 a Administradores y el 2 a Trabajadores.

La última columna de la tabla muestra dos botones que definen las posibles acciones a realizar con cada uno de los usuarios por parte del Super Administrador, rol con el que debemos haber iniciado sesión para poder realizar estas acciones.

El primero de ellos, “Editar”, ofrece la posibilidad de cambiar alguno de los valores del usuario, pudiendo manejar desde el nombre usuario hasta el rol asignado. Este nuevo formulario de modificación será proporcionado en forma de “pop-up” sin necesidad de pasar a una nueva pantalla, mejorando la interactividad de la aplicación con el usuario, como se ve en la Figura 24.

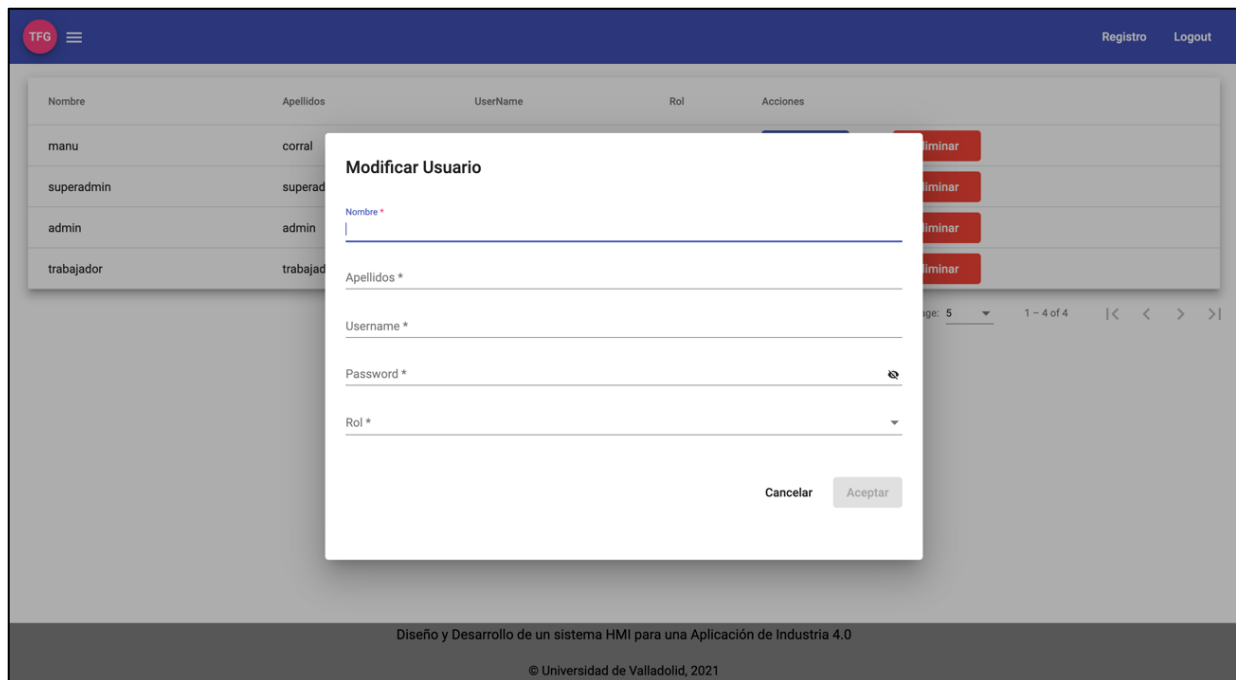


Figura 24: Formulario de modificación de Usuarios

Al igual que en los formularios de inicio de sesión y de registro de usuarios, aquí tampoco es posible realizar una petición de modificación de usuario sin completar todos los campos adecuadamente.

Los campos de este formulario de modificación son idénticos a los comentados en el apartado 5.3 al describir el registro de nuevos usuarios en la aplicación.

A la hora de borrar usuarios se deberá pulsar sobre el botón situado a la derecha de cada fila de usuario con el nombre “Eliminar” y en un característico color rojo. Una vez se haya pulsado sobre este botón, el sistema mostrará por pantalla un aviso para solicitar la confirmación por parte del Super Administrador para finalizar la acción de borrado, tal y como se muestra en la Figura 25.

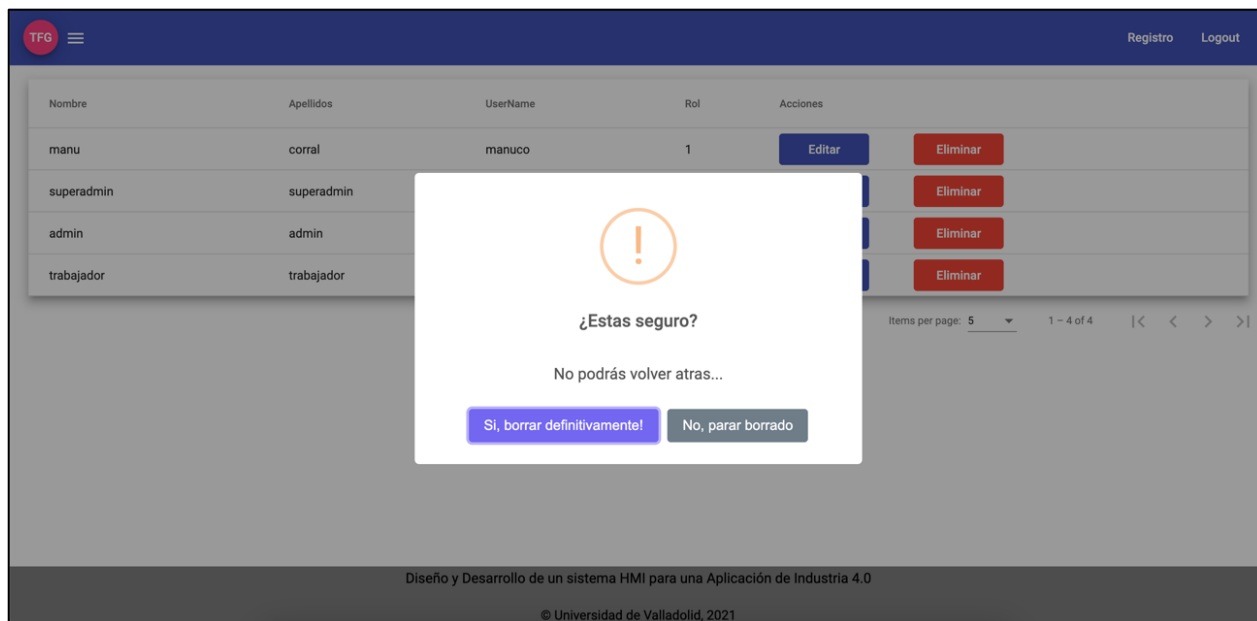


Figura 25: Confirmación de eliminación de Usuario

Mediante el botón situado a la izquierda, se eliminará definitivamente el usuario seleccionado, devolviendo al usuario a la tabla contenedora anterior observándose un registro menos, mientras que seleccionando el botón derecho volveremos a la página de usuarios, manteniendo el registro de usuario sin realizar ningún cambio. Observamos ambas acciones en la Figura 26 y la Figura 27 respectivamente.

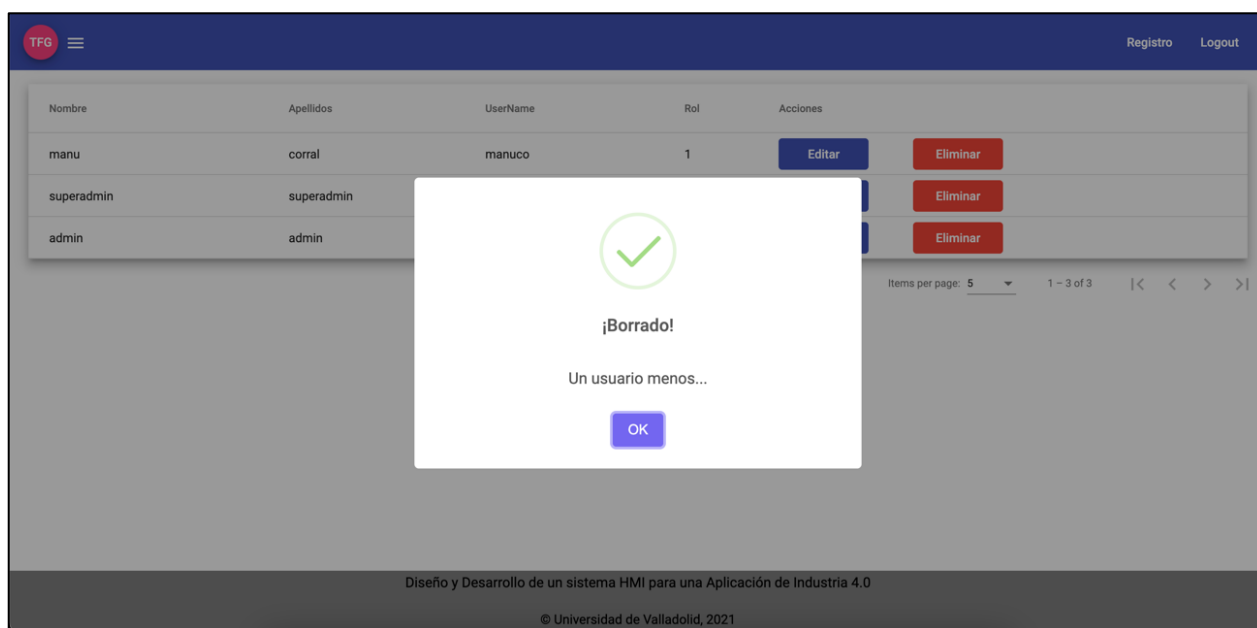


Figura 26: Borrado de Usuario

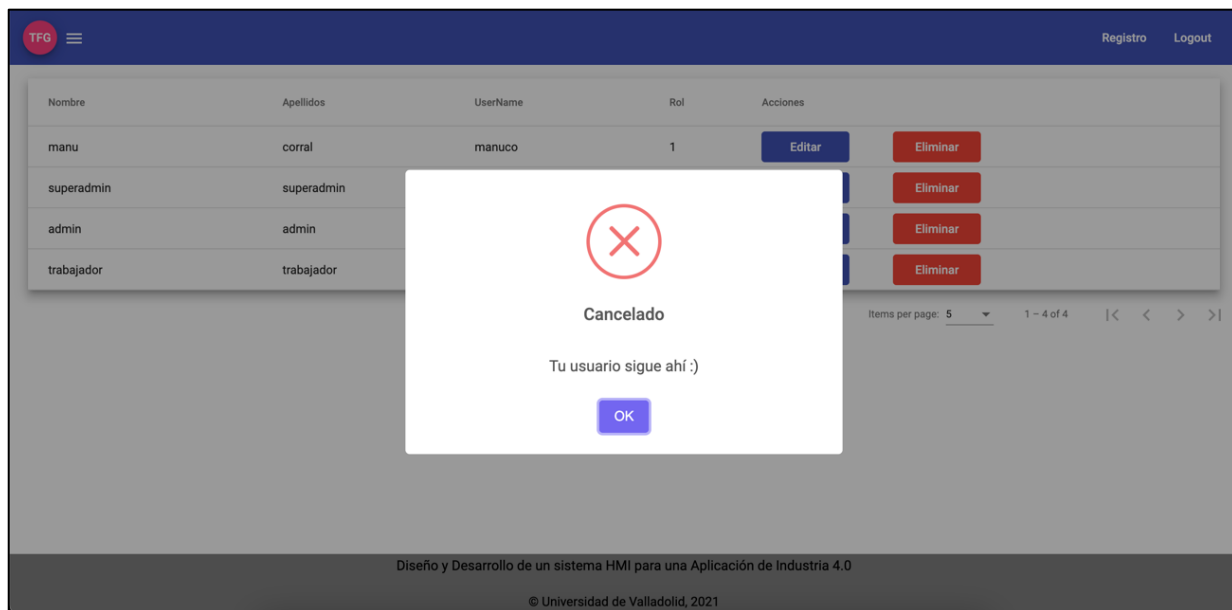


Figura 27: Cancelación de borrado de Usuario

Por otra parte, y tal como comentábamos anteriormente, las acciones de modificación y borrado de usuarios están únicamente destinadas a usuarios con rol 0 o de Super Administrador, advirtiendo de este hecho a posibles usuarios con rol 1 o de Administrador que intenten gestionar los usuarios a su cargo.

Para obtener dicho tratamiento, deberán contactar con los encargados del sistema para gestionar la operación, pudiendo realizar este trámite a través del aviso que aparece por pantalla al pinchar sobre el botón de modificar o eliminar, tal y como se muestra en la Figura 28.

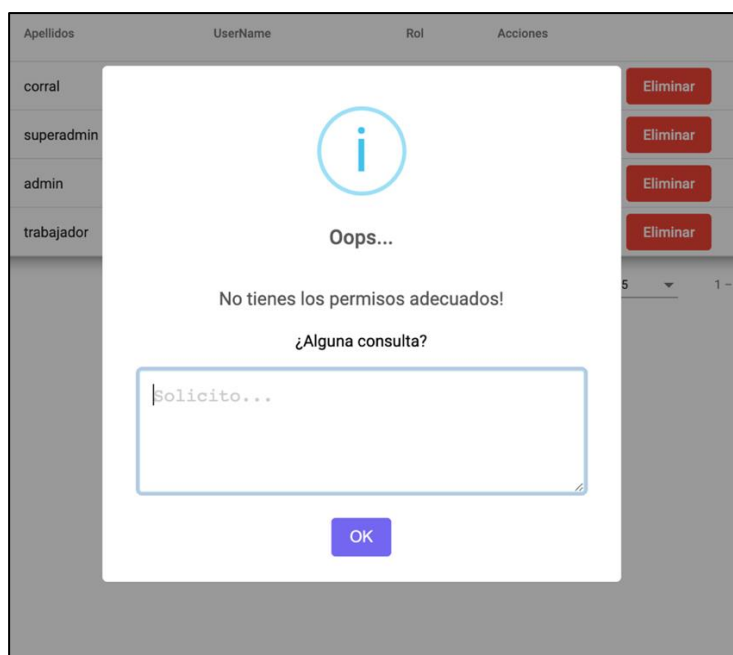


Figura 28: Ejemplo de petición de borrado con rol de Administrador

5.6. Página Gráficos.

Otro de los apartados que observamos en el menú lateral permite generar gráficos a partir de los datos recogidos por nuestros sensores. Este submenú será visible y accesible a los diferentes tipos de usuarios disponibles en nuestra aplicación, y al acceder al mismo nos conducirá a una nueva página simple a la vista y prácticamente vacía, como se observa en la Figura 29, en la cual el usuario deberá seleccionar el componente sobre el cual quiere obtener datos históricos y generar una gráfica con la evolución temporal.

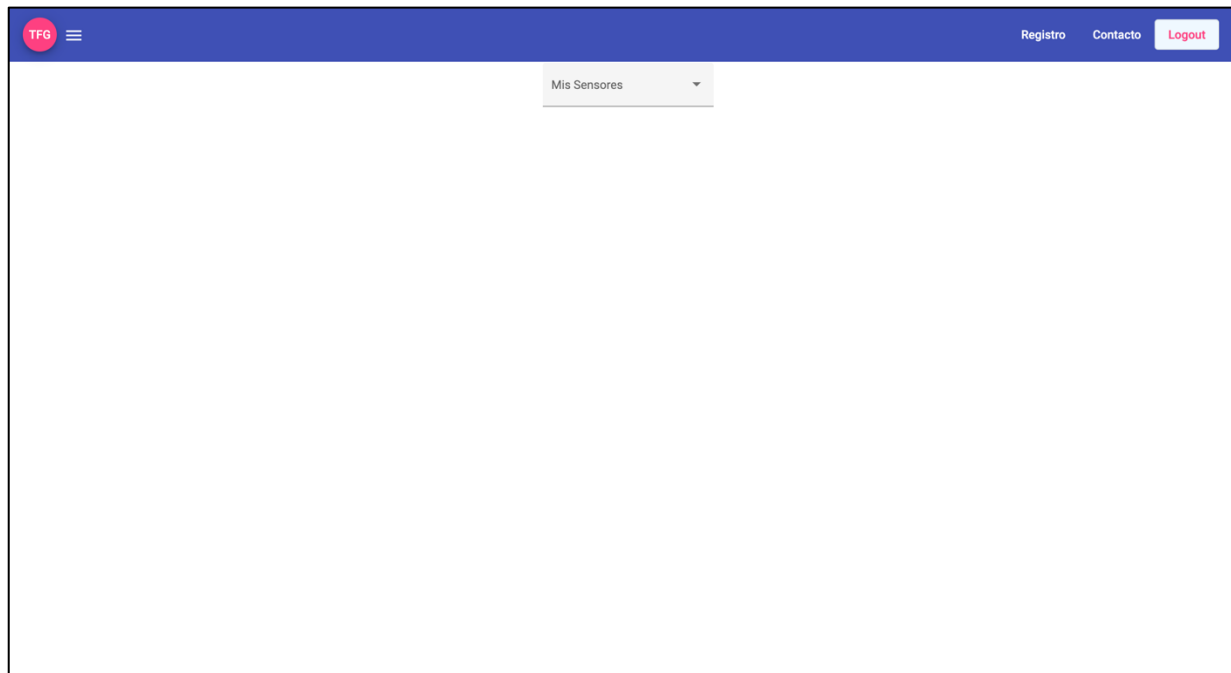


Figura 29: Inicio de página de Gráficos

Una vez seleccionado uno de los sensores disponibles para el usuario, se presentarán cuatro opciones para elegir el espacio temporal deseado, tal como se visualiza en la Figura 30.

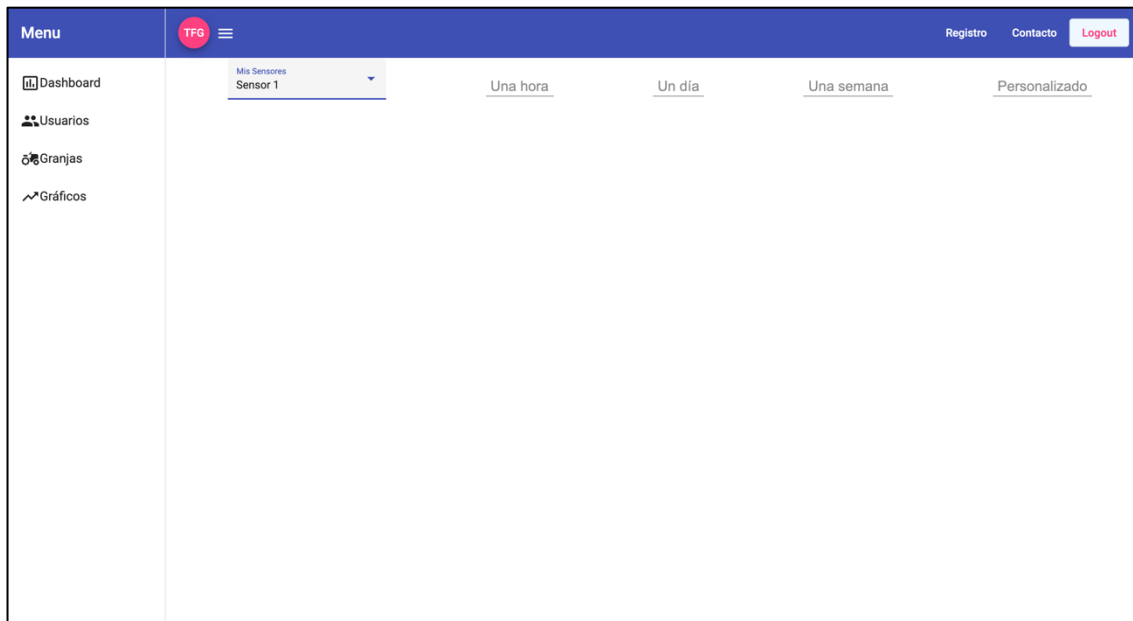


Figura 30: Página de Gráficos una vez seleccionado un componente

La primera de estas opciones permite al usuario a través de un simple *click* la visualización de una tabla rellena con los datos generados por los sensores en la última hora, al igual que la segunda y tercera opción, las cuales nos mostrarían los datos recogidos en el último día, o en la última semana, respectivamente.

La última opción, visible en la Figura 30, nos permite seleccionar un rango de fechas concretas mediante una interfaz externa. Al seleccionar esta opción, aparece en la pantalla un formulario de entrada a través del cual el usuario podrá seleccionar una fecha de inicio y otra de fin, como se observa en la Figura 31.

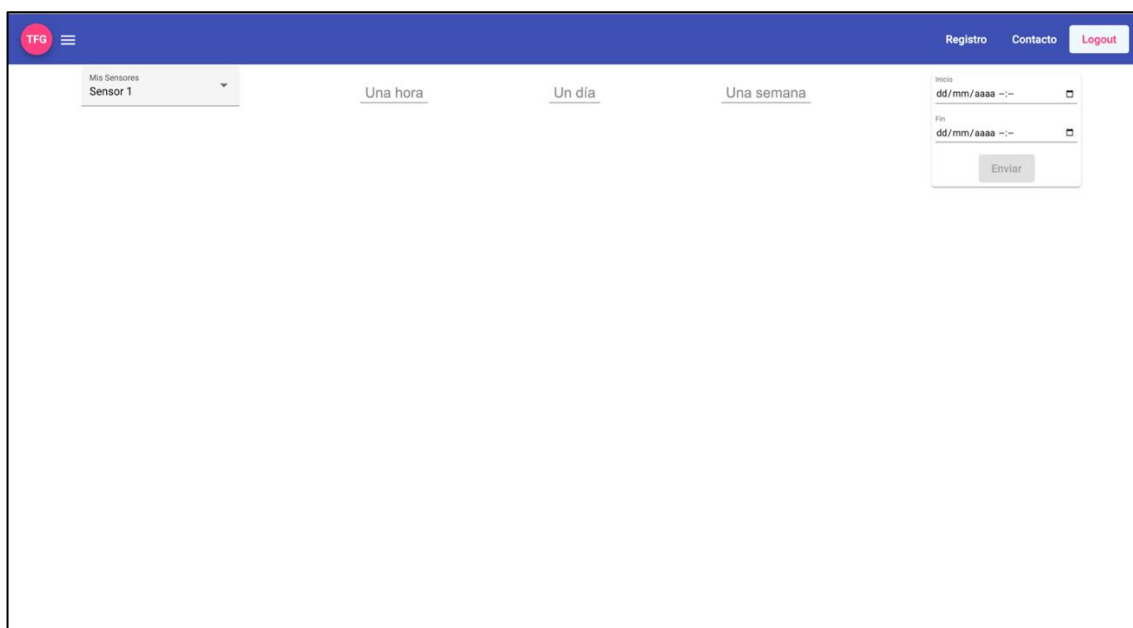


Figura 31: Formulario de solicitud de datos personalizado

Al seleccionar uno de los campos del formulario visible en la esquina superior derecha de la Figura 31, se muestra por pantalla la interfaz interactiva visible en la Figura 32 para que el usuario introduzca la fecha deseada de manera sencilla, sin la necesidad de escribir los campos de manera manual y con un orden concreto.

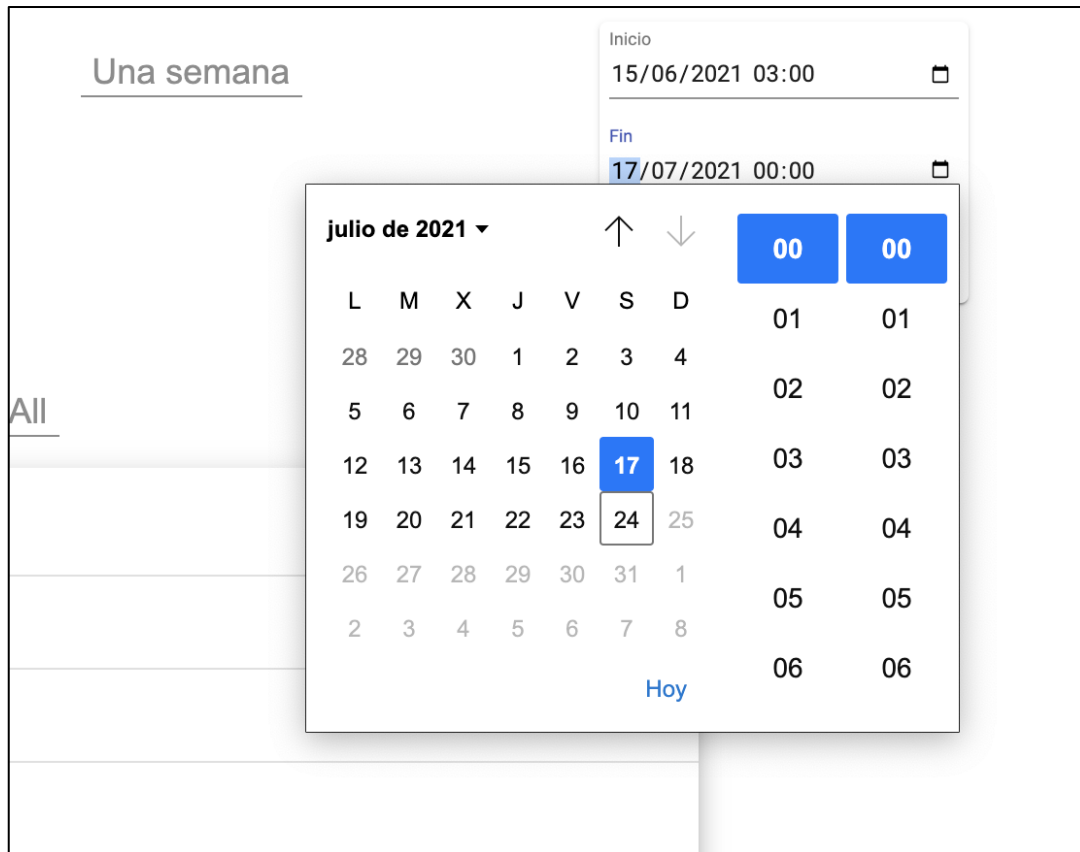


Figura 32: Interfaz para la selección de fechas personalizadas

Después de ejecutar una de las cuatro opciones anteriores, el sistema genera y muestra una tabla con los datos obtenidos en el periodo de tiempo seleccionado. La tabla, para proporcionar una mejor experiencia de usuario, dispone de un filtro rápido para seleccionar únicamente los registros correspondientes a alguno de los tipos de sensor existentes, que serán, normalmente, de temperatura, de humedad y de luminosidad.

Además, esta tabla se nos presenta paginada, con 5 registros por página, aunque con la capacidad de incluir hasta 20 filas por página, y con la posibilidad de ser ordenada por cualquiera de las columnas, tal y como se observa en la Figura 33, después de haber realizado la petición anterior.

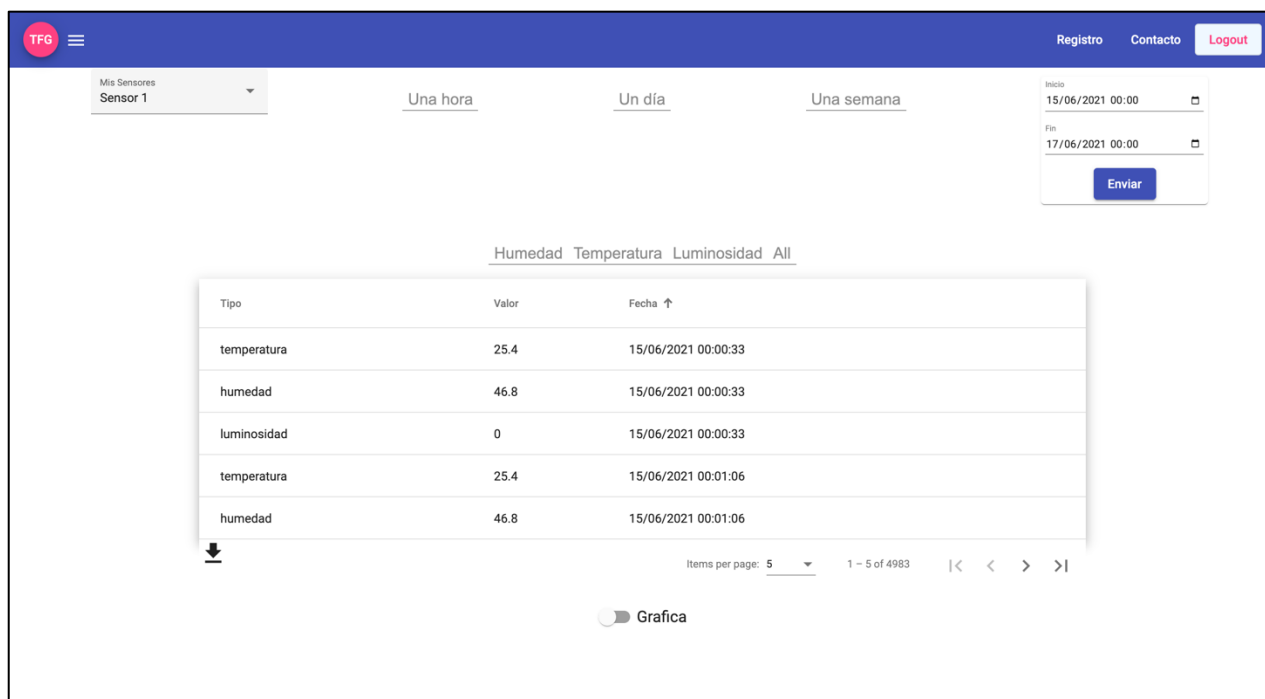


Figura 33: Tabla respuesta

Dentro de la Figura 33, se observan dos componentes de gran interés para el usuario. El primero de ellos, la flecha de descarga situada debajo de la tabla en la esquina inferior izquierda, permite la exportación de los datos seleccionados por el usuario a formato *xlsx* o *csv*, permitiendo su tratamiento en la herramienta *Excel* de *Microsoft*. Observamos en la Figura 34 un ejemplo de esta exportación.

	A	B	C	D
1	type	id_datahunter	valor	time_stamp
2	temperatura	1	25,4	15/06/2021 00:00:33
3	temperatura	1	25,4	15/06/2021 00:01:06
4	temperatura	1	25,3	15/06/2021 00:01:39
5	temperatura	1	25,4	15/06/2021 00:02:12
6	temperatura	1	25,4	15/06/2021 00:02:44
7	temperatura	1	25,4	15/06/2021 00:03:17
8	temperatura	1	25,4	15/06/2021 00:03:49
9	temperatura	1	25,4	15/06/2021 00:04:22
10	temperatura	1	25,4	15/06/2021 00:04:54
11	temperatura	1	25,4	15/06/2021 00:05:46
12	temperatura	1	25,4	15/06/2021 00:06:18
13	temperatura	1	25,4	15/06/2021 00:06:50
14	temperatura	1	25,4	15/06/2021 00:07:24
15	temperatura	1	25,4	15/06/2021 00:07:56
16	temperatura	1	25,4	15/06/2021 00:08:28
17	temperatura	1	25,4	15/06/2021 00:09:00
18	temperatura	1	25,4	15/06/2021 00:09:32
19	temperatura	1	25,4	15/06/2021 00:10:05
20	temperatura	1	25,4	15/06/2021 00:10:56
21	temperatura	1	25,4	15/06/2021 00:11:29
22	temperatura	1	25,4	15/06/2021 00:12:01

Figura 34: Exportación de datos a Excel

El segundo componente que destacar, el cual se observa en la Figura 33, es un botón o switch situado al final de la tabla bajo el título de “Gráfica”. Este nos permitirá generar un gráfico en el que se podrán estudiar los datos de una manera más visual y ver el comportamiento a lo largo del intervalo temporal seleccionado, tal y como vemos en la Figura 35.

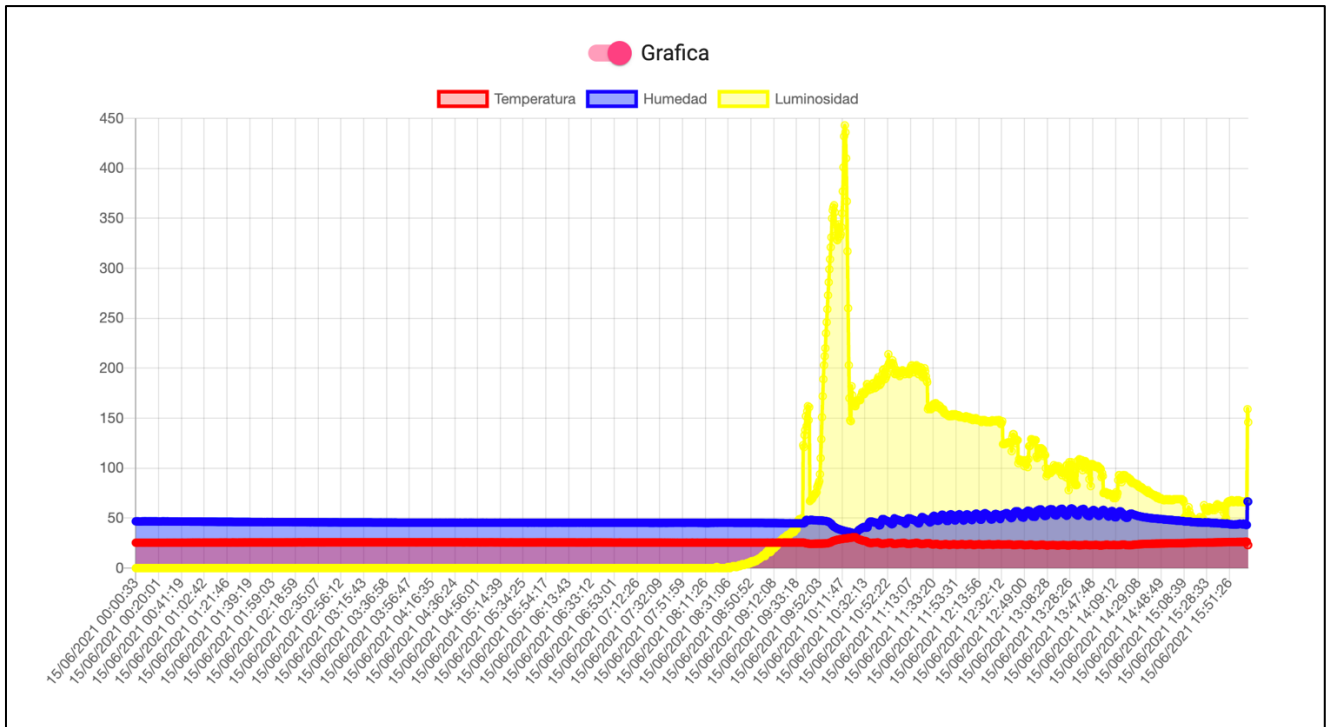


Figura 35: Gráfico de datos obtenidos

Si se quisiera modificar este gráfico y ver únicamente alguno de los 3 tipos de datos, es suficiente con pulsar sobre cada uno de los títulos que se sitúan encima de la gráfica para sacarlo de la visualización, apareciendo, por ejemplo, así en la Figura 36 una imagen filtrada de los datos de temperatura y humedad únicamente.

Además, si se posiciona el ratón o cursor encima de alguno de los puntos de esta gráfica, se puede observar la información específica del dato correspondiente a esa hora, como se observa igualmente en la Figura 36.

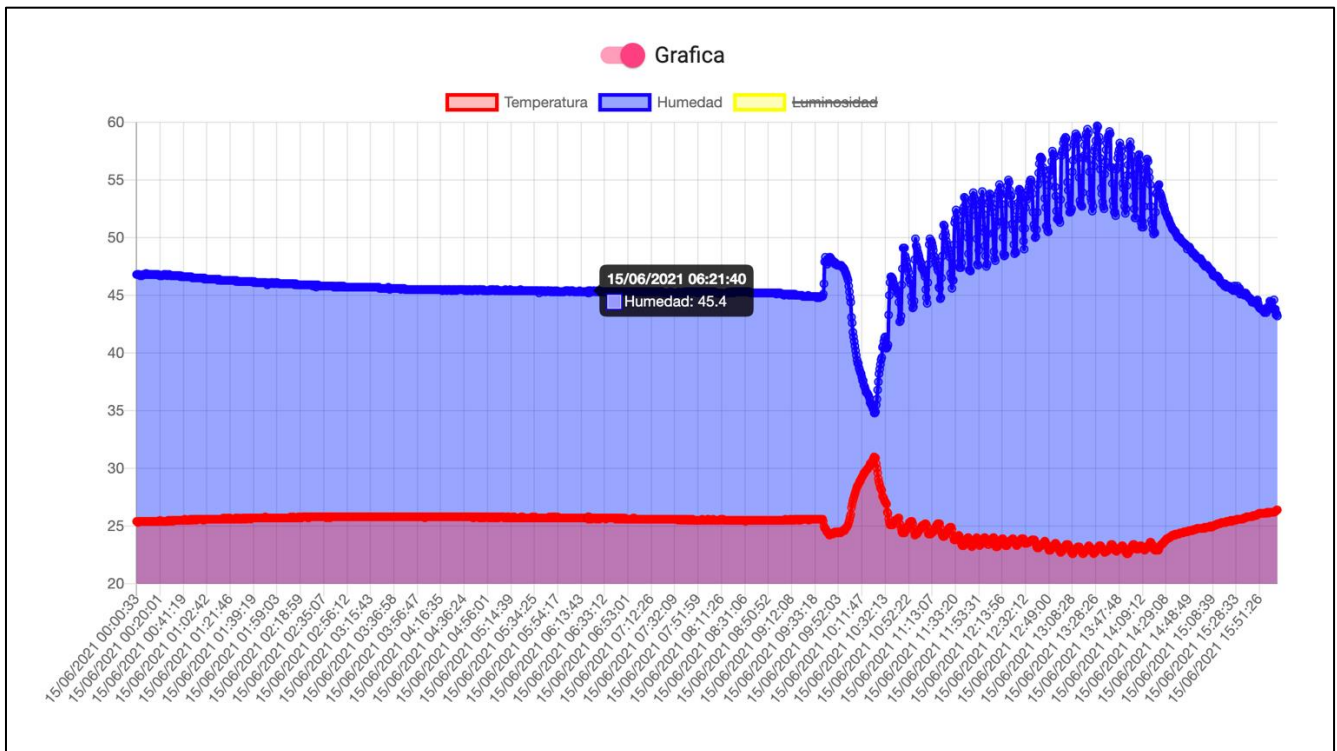


Figura 36: Gráfica filtrada

Otro aspecto importante a la hora de la visualización de las gráficas y el estudio de los datos es la focalización en aquellas zonas que nos avisan de valores distintos de los esperados, para ello, se ha dotado a la aplicación con la capacidad de hacer zoom en cualquier punto de la gráfica.

Mediante una simple selección en rectángulo, se puede ir ampliando la gráfica en la parte especificada, tal como se ve en la Figura 37. Esta funcionalidad nos permite observar, de un simple vistazo, aquellos valores más extremos, sin necesidad de tener que utilizar la tabla anterior para tal fin.

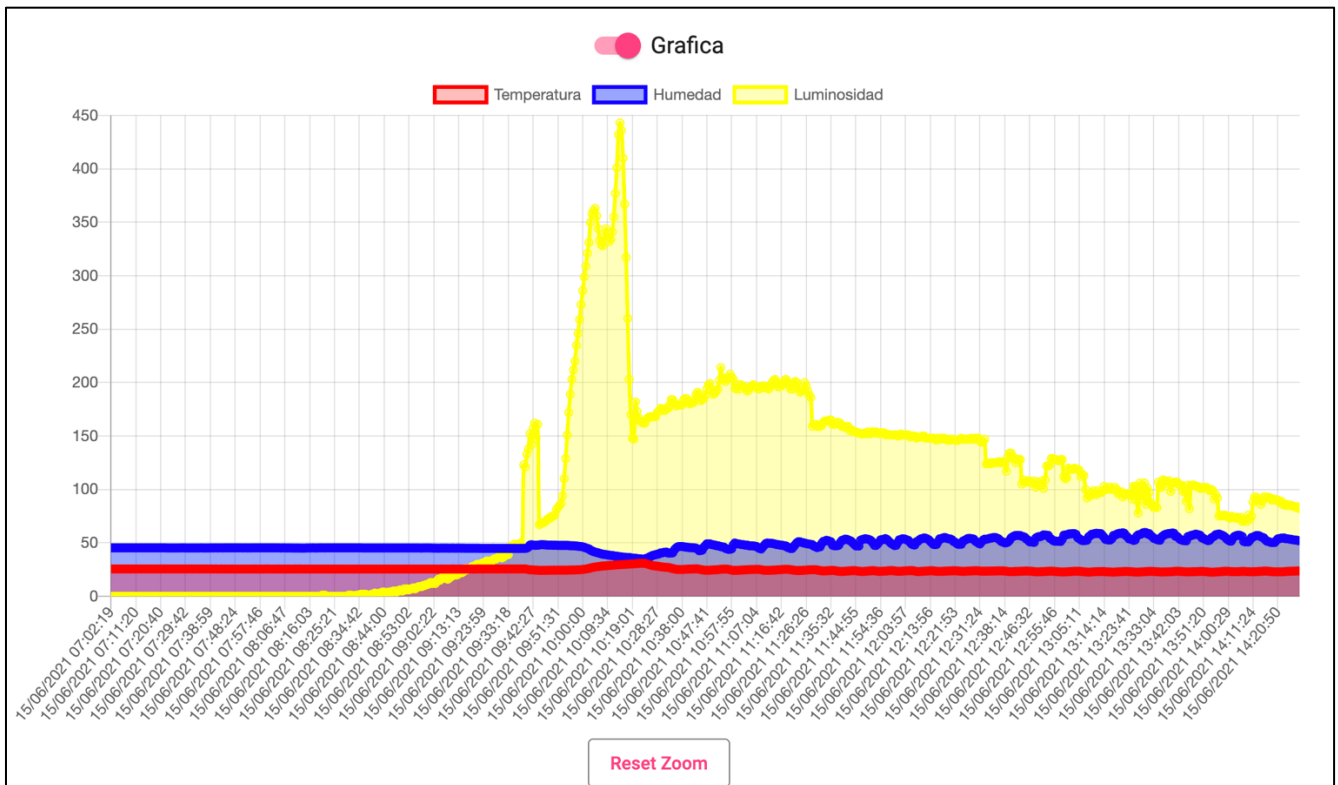


Figura 37: Zoom en la gráfica

Tal y como se ve en la Figura 37, aparece un botón en la parte inferior de la gráfica a través del cual podemos volver a la imagen original, eliminando el zoom aplicado.

5.7. **Página *Dashboard*.**

Sin duda alguna, esta página permite al usuario realizar muchos de las funcionalidades necesarias para cubrir los requisitos propuestos en esta aplicación.

Según accedemos a la pantalla, observamos un único campo para rellenar, como se observa en la Figura 38.



Figura 38: Página de dashboard

Aquí se podrá indicar la granja sobre la cual se quiere trabajar. Este primer paso será necesario debido a que un mismo usuario puede tener asignadas varias granjas, y únicamente podrá trabajar con una de ellas cada vez.

Una vez seleccionada una de las granjas disponibles para el usuario a través del botón de selección visible en la parte lateral izquierda de la Figura 38, aparecen todas las opciones disponibles para la interacción por parte del usuario con la aplicación, las cuales se observan en la Figura 39 y que pasaremos a comentar a lo largo de este apartado.

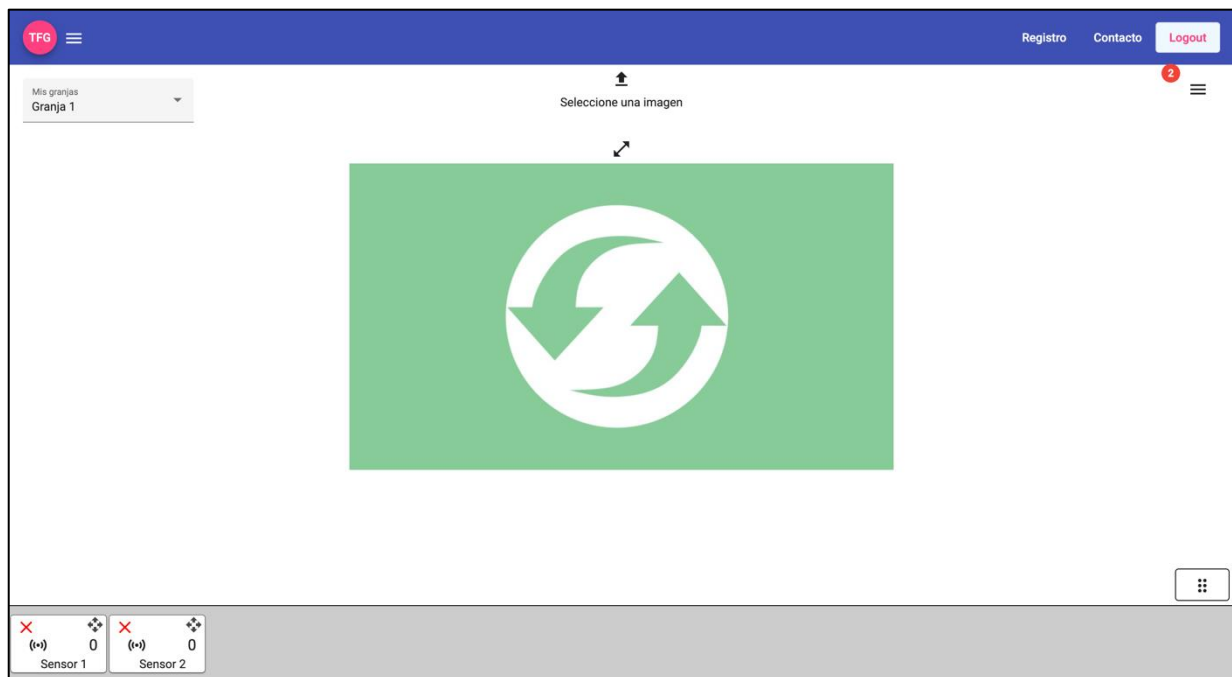


Figura 39: Granja seleccionada

En la Figura 39 podemos observar la pantalla principal que se vería la primera vez que un usuario acceda a una de sus granjas.

En el centro de la pantalla, se puede visualizar una imagen de color verde con un símbolo de reciclaje. Esto quiere indicar que es el espacio dedicado para que un usuario cargue una imagen, preferiblemente una foto aérea de una de sus granjas. Para cumplir tal objetivo, aparece sobre la imagen y debajo del *header*, un botón el cual permite seleccionar una imagen de manera local que será almacenada en la aplicación y estará asociada a dicha granja, siendo esta modificable a través de este procedimiento siempre que el usuario lo desee. En la Figura 40 se visualiza un ejemplo de selección de imagen a través del botón ofrecido para tal fin.

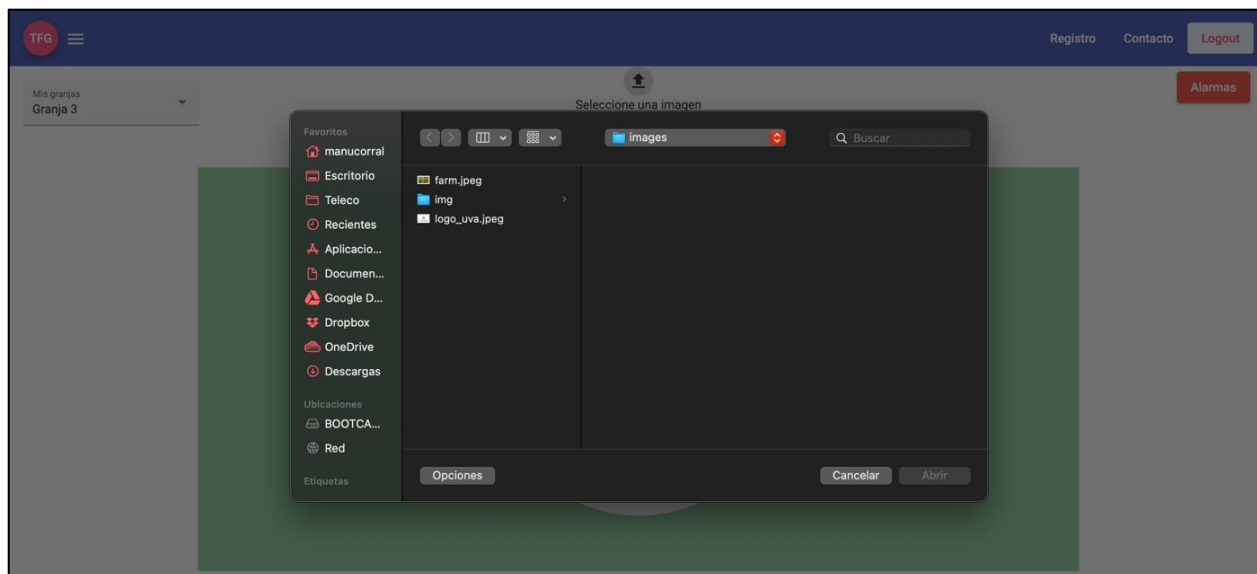


Figura 40: Selección de imagen asociada a una granja

Una vez seleccionada la imagen que el usuario quiere añadir a la aplicación, se presenta un campo de edición el cual está destinado al tratamiento de la imagen, como se observa en la Figura 41.

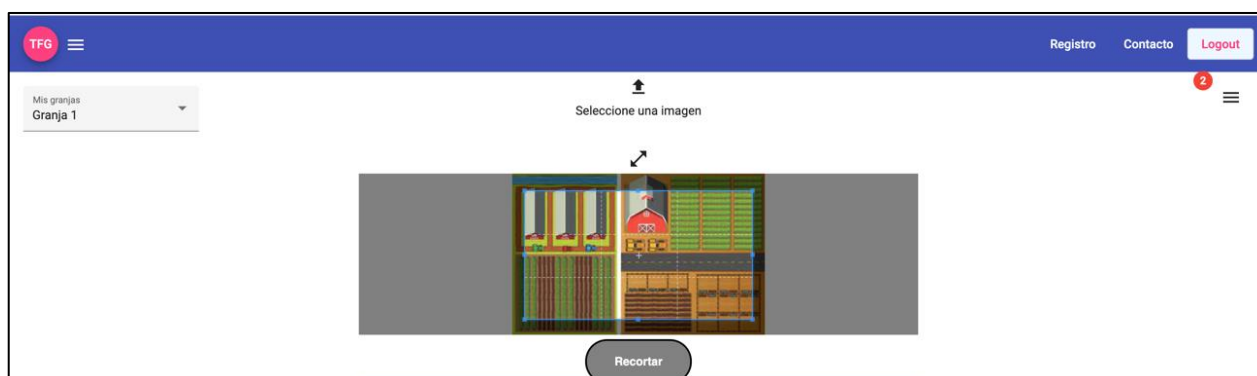


Figura 41: Tratamiento imagen seleccionada

A continuación, la figura resultante aparece en lugar de la situada por defecto. En nuestro caso, y a modo de demostración, hemos añadido una imagen de una granja digital de prueba, visible en la Figura 42 y con la que seguiremos la explicación.

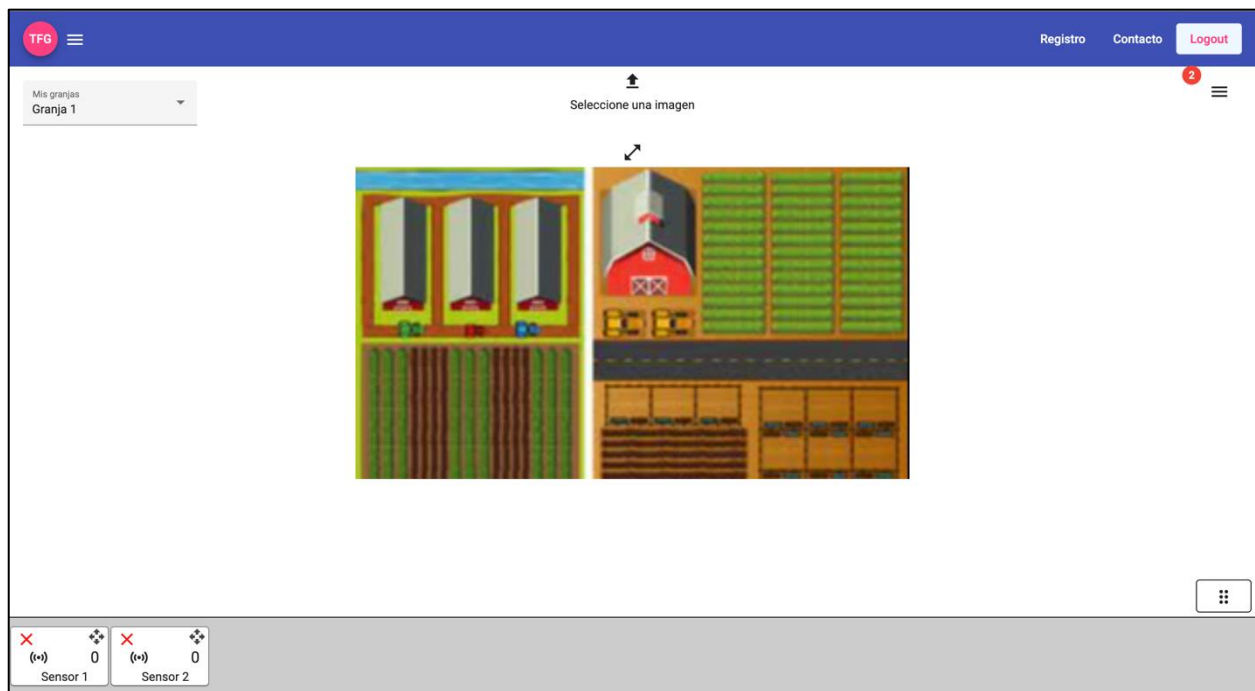


Figura 42: Página de dashboard con la granja seleccionada

Dentro de esta pantalla se observan varios componentes. En primer lugar, justo encima de la figura de la granja, se observa un botón con forma de cuadrado. Este servirá para visualizar la imagen seleccionada y los diferentes sensores a pantalla completa, de cara a que esta sea estable y quede fija en pantallas propias en el lugar de trabajo. Podemos ver el resultado de pulsar dicho botón en la Figura 43.

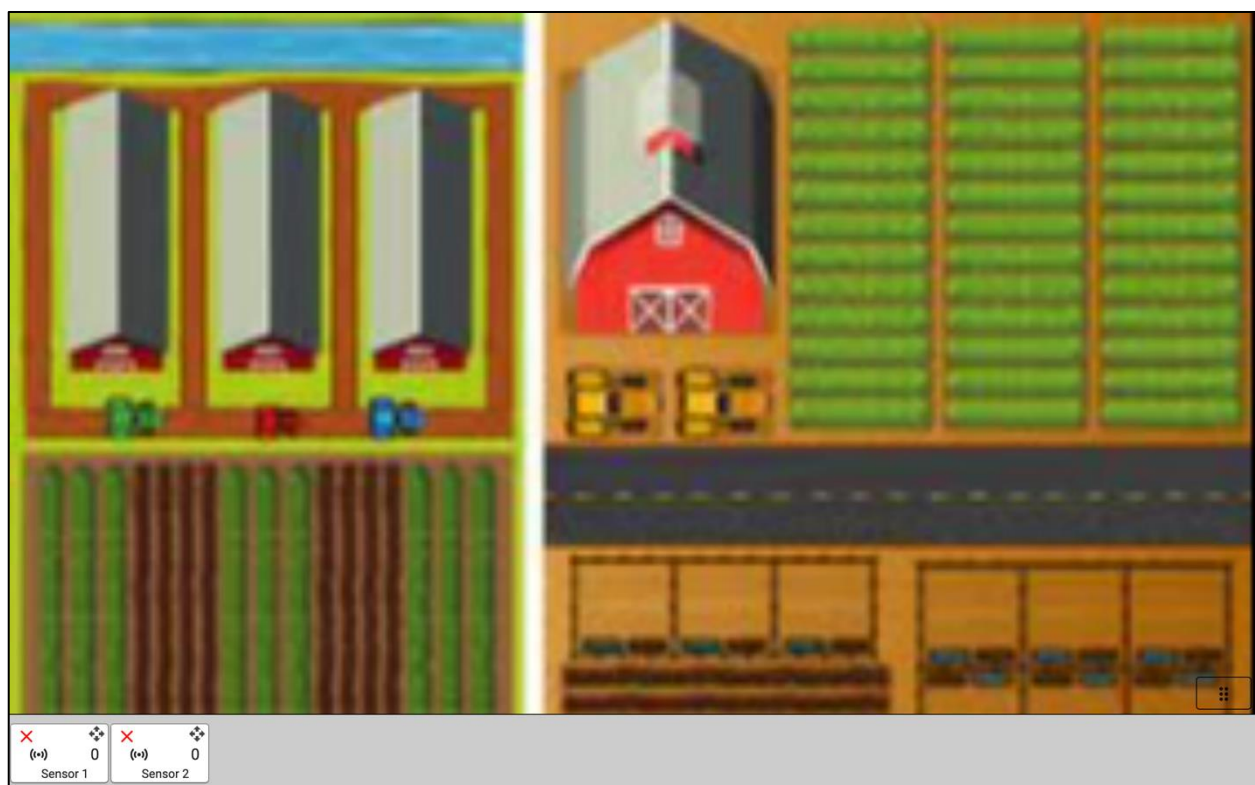


Figura 43: Granja a pantalla completa

Uno de los aspectos más importantes de la aplicación se ofrece a través de la barra inferior de color gris en la cual se sitúan todos aquellos sensores que haya contratado el usuario para la granja seleccionada.

Estos sensores tienen la particularidad de poder moverse por la totalidad de la imagen, colocándolos en las posiciones más precisas posibles asemejándose a su ubicación real en la actualidad, para ello, será necesario pulsar en la cruz situada en la esquina superior derecha de cada sensor para su posterior desplazamiento. Si quisiéramos desactivar estos sensores, y llevarlos a su lugar original en la barra de herramientas inferior, será suficiente con pulsar en la cruz de color rojo en la parte izquierda superior. Podemos ver un ejemplo del resultado del movimiento de estos sensores en la Figura 44.

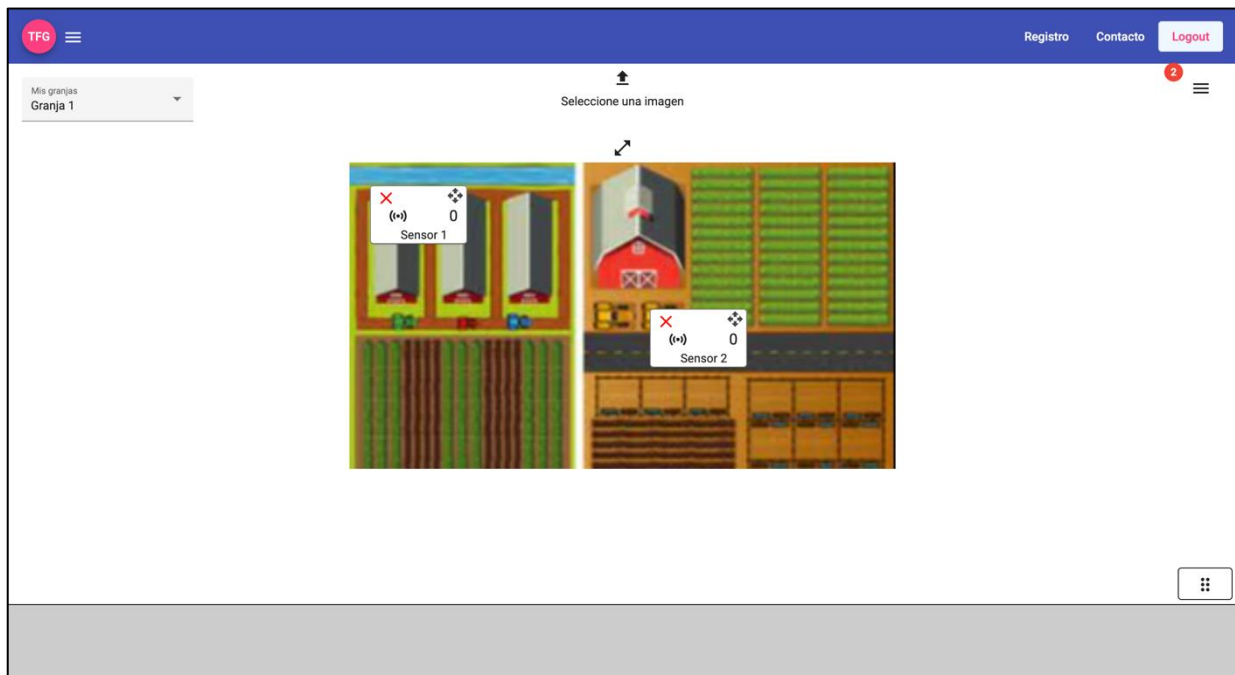


Figura 44: Movimiento de los sensores por la imagen

Tal y como veíamos en el apartado 5.6, cada sensor recoge tres tipos de datos: temperatura, humedad y luminosidad. Mediante la pulsación en el centro de cada sensor, aparece un menú a través del cual podemos seleccionar qué tipo de dato queremos visualizar en dicho sensor. A su vez, y para ayudar al usuario a distinguir entre los componentes, se permite cambiar el nombre del sensor seleccionado, como se observa en la Figura 45.

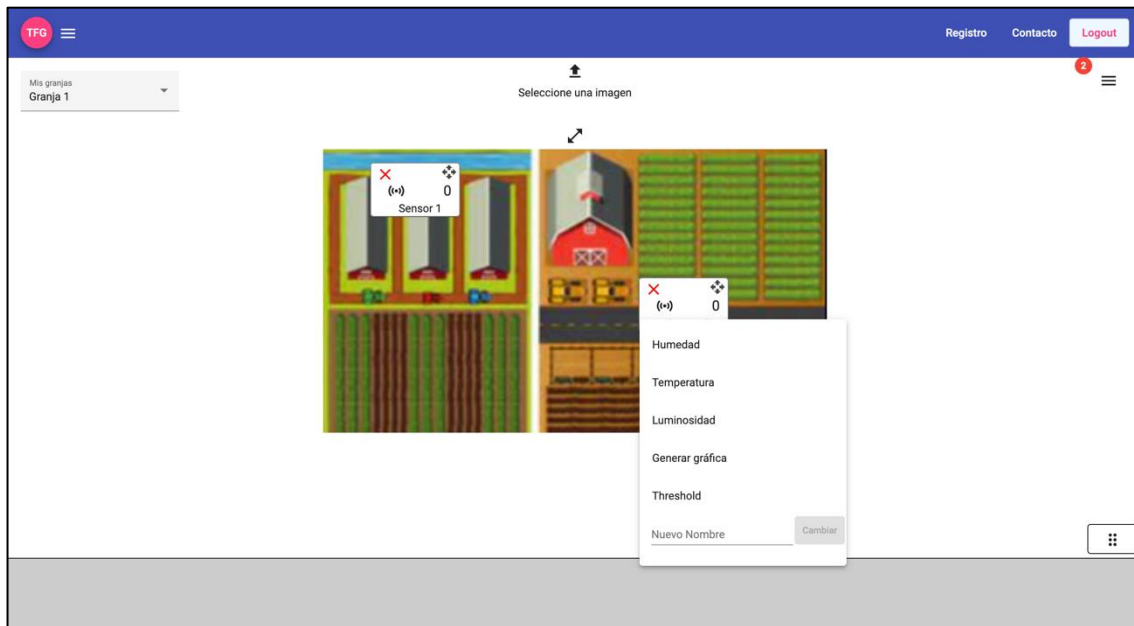


Figura 45: Menú de cada sensor

Por otra parte, en la Figura 45 se observa una opción dentro del menú de cada sensor para pasar directamente al menú de graficar, sin necesidad de indicar previamente un sensor como veíamos en la Figura 29.

Al primer componente, le vamos a asignar el nombre de Río, y procederemos a observar en tiempo real sus valores de humedad, mientras que, para el segundo sensor, probamos a cambiar el nombre del Sensor 2 por Jardín, e indicamos que queremos observar la evolución de la temperatura en tiempo real.

Podemos observar el resultado final en la Figura 46 de nuestra página de *dashboard*.

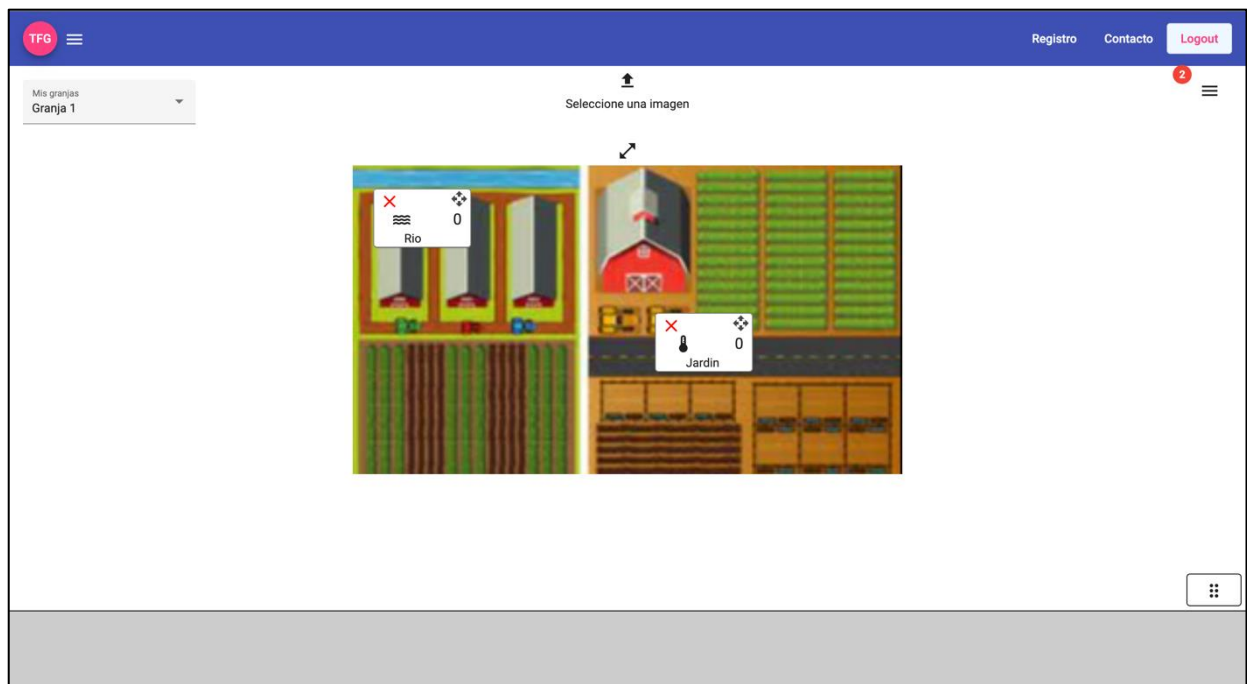


Figura 46: Resultado del tratamiento en la página Dashboard

A lo largo del uso de la aplicación, es lógico pensar que un usuario puede tener que interactuar con multitud de sensores, y es posible que desee ir cambiando asiduamente entre alguno de los datos que estos sensores proporcionan para todos los dispositivos. Para ello, se ha dotado a la aplicación del botón situado en la parte inferior derecha visible en la Figura 44, la Figura 45 o la Figura 46, por ejemplo.

Con un simple *click*, aparecerá un menú de selección para elegir qué tipo de dato queremos visualizar en todos los sensores de nuestra granja, tal y como se observa en la Figura 47.

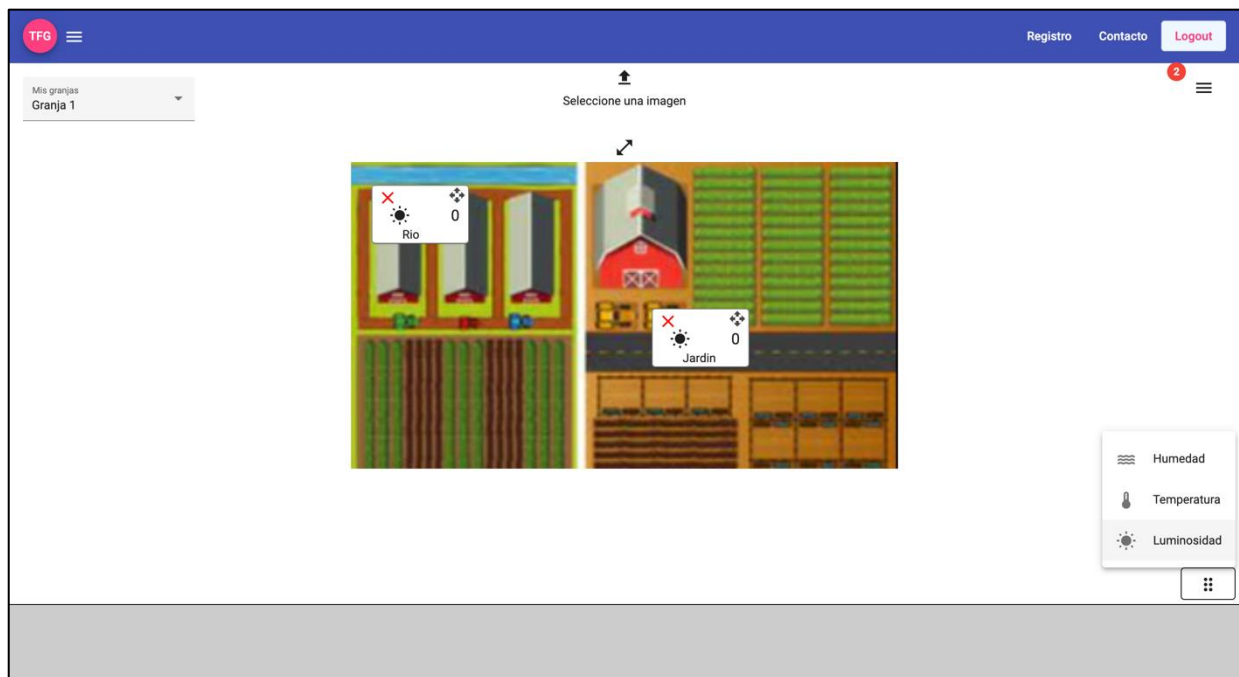


Figura 47: Menú de cambio de tipo de dato

Se observa también un número situado en la parte derecha de los sensores. Este proporcionará al usuario, mediante un simple vistazo, el último dato obtenido del tipo de dato que se esté mostrando en cada sensor, actualizándose en tiempo real según se recojan los últimos valores.

A su vez, también existe un botón por cada sensor denominado “*threshold*” a partir del cual se pueden ajustar los valores entre los que se consideraría válidos los datos obtenidos por cada uno de los sensores. Si pulsamos sobre el mismo, se muestra por pantalla un “*pop-up*” a través del cual podemos ajustar dichos valores, tal y como se ve en la Figura 48.

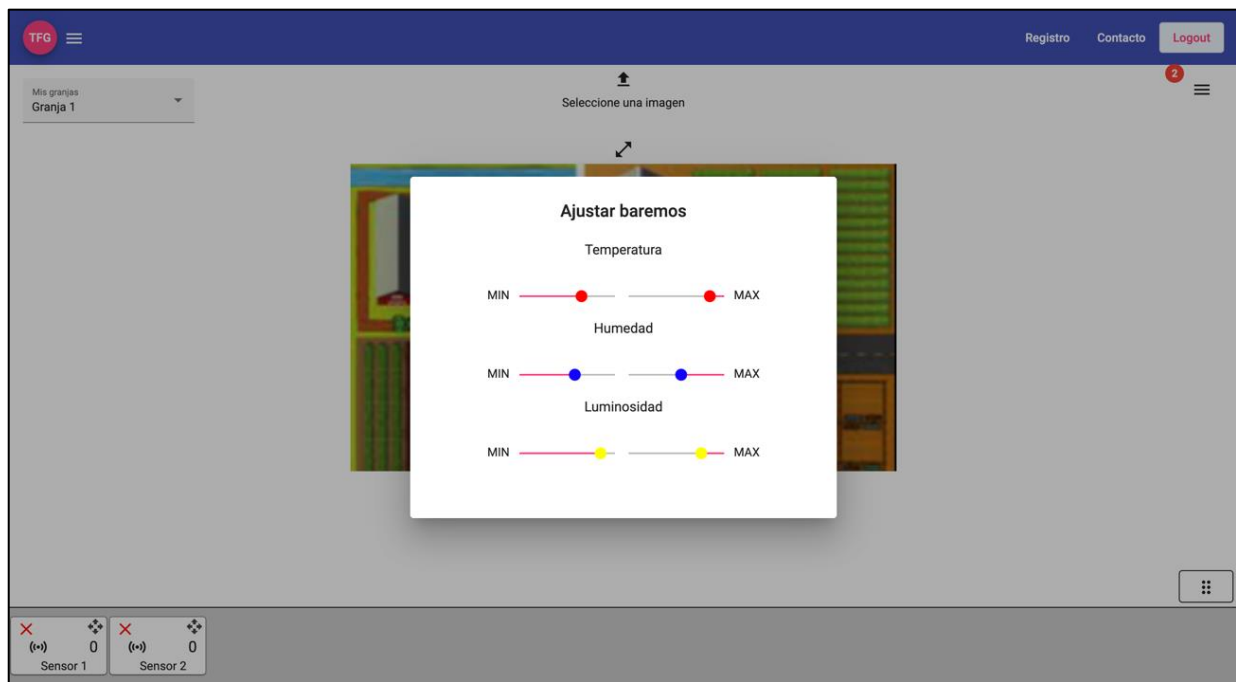


Figura 48: Ajuste de valores máximos y mínimos de los sensores

Por último, cabe destacar el botón rojo de Alertas situado en la parte superior derecha de la pantalla de *dashboard*.

Es posible que el usuario defina una serie de puntos críticos en sus sensores, y que, mientras está mostrando otro tipo de información, alguno de estos puntos críticos se haya visto superado. Es entonces en ese momento cuando se crean alertas, visualizándose mediante un círculo en el cual se incluye el número de alertas obtenidas, como se observa en la esquina superior derecha de la Figura 49.

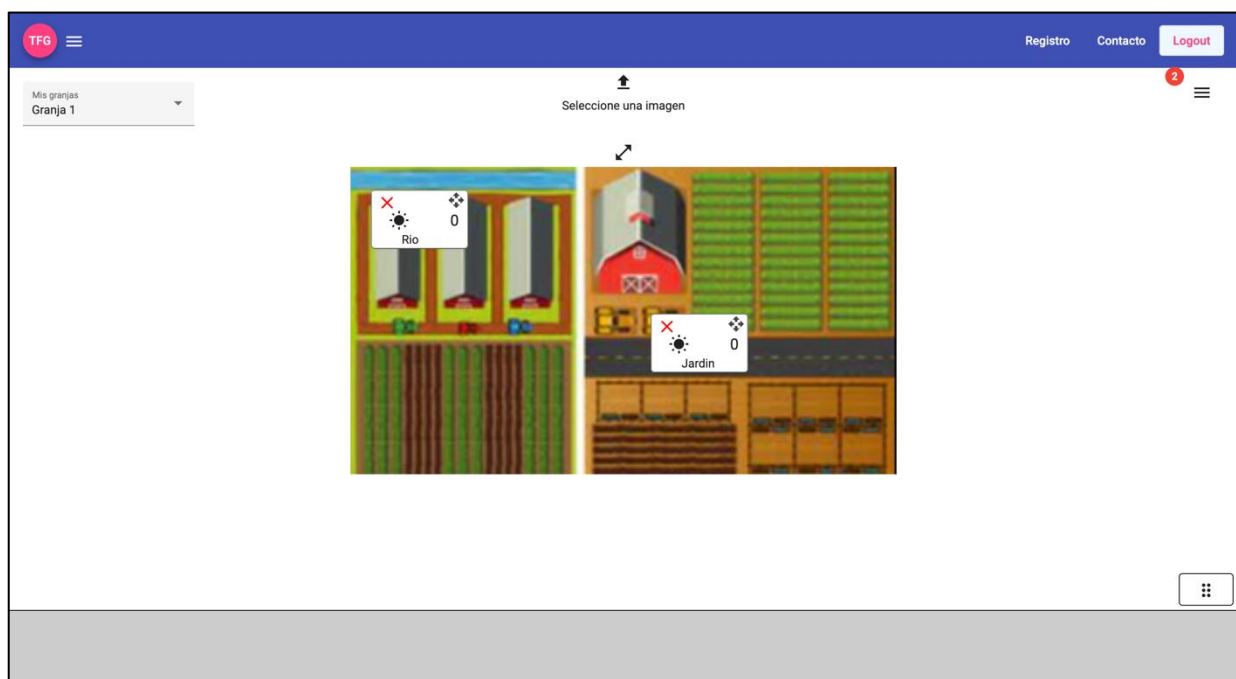


Figura 49: Número de alertas del sistema

Si pulsamos sobre el botón de alarmas, ahora con un aviso de que hemos obtenido 2 puntos críticos, aparecerá un menú lateral con la información relacionada de los avisos, como se ve en la Figura 50.

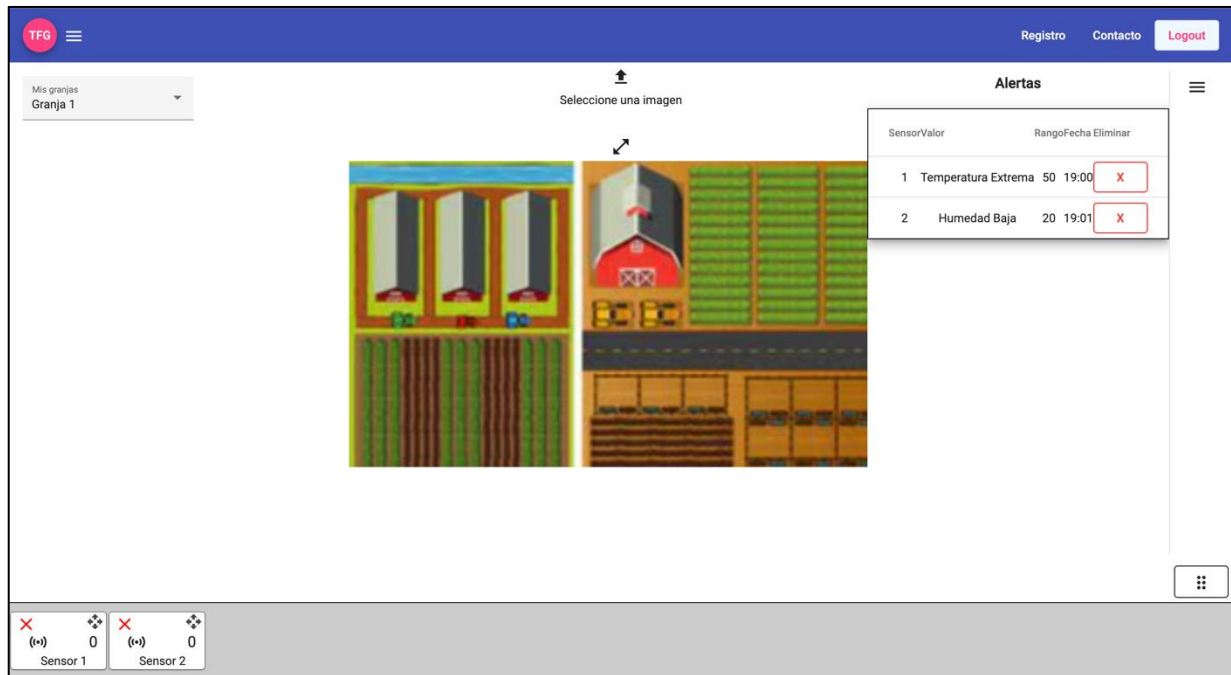


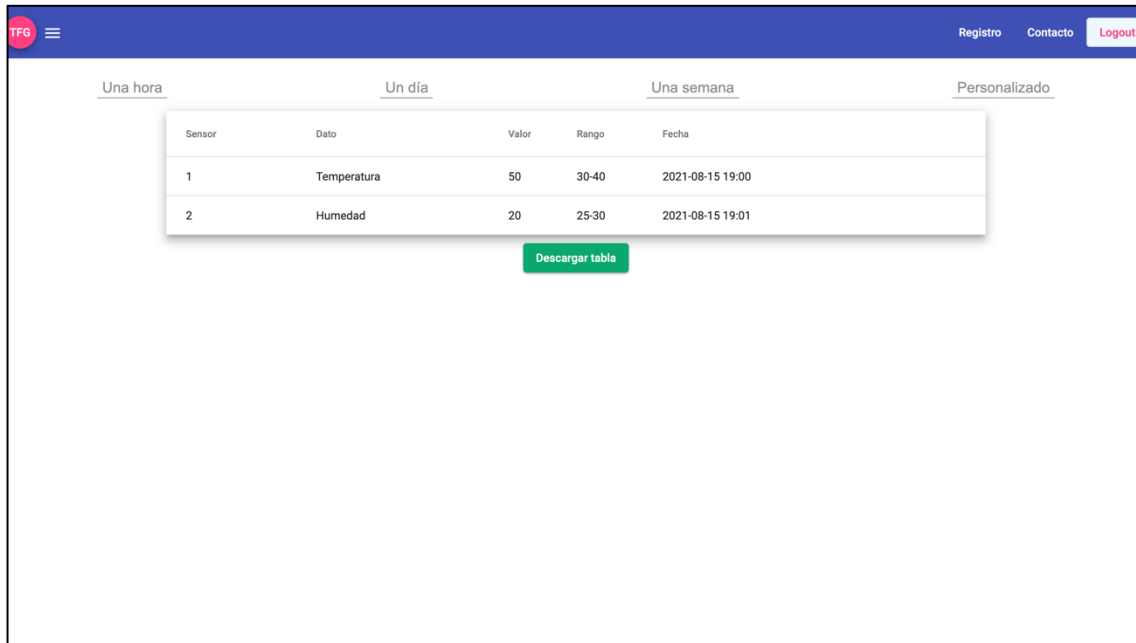
Figura 50: Información de las alarmas del sistema

Una vez visualizadas dichas alarmas, el contador vuelve a situarse a 0, a la espera de que se produzcan nuevos avisos en algún sensor contratado.

5.8. Página Alertas.

Al igual que en el apartado 5.6 en esta nueva página accesible desde el menú lateral de nuestra aplicación, tal y como vemos en la Figura 18, se puede acceder a un registro histórico de las alertas generadas en el sistema.

En la Figura 51, se puede observar el registro de todas las alertas que han sucedido en los últimos días.



The screenshot shows a web interface for 'Alertas del sistema'. At the top, there is a navigation bar with 'Registro', 'Contacto', and 'Logout' links. Below the navigation bar, there are four tabs: 'Una hora', 'Un día', 'Una semana', and 'Personalizado'. The 'Un día' tab is selected. The main content area displays a table with the following data:

Sensor	Dato	Valor	Rango	Fecha
1	Temperatura	50	30-40	2021-08-15 19:00
2	Humedad	20	25-30	2021-08-15 19:01

Below the table, there is a green button labeled 'Descargar tabla'.

Figura 51: Alertas del sistema

Se observan 4 campos en la tabla de alertas correspondiente a la Figura 51:

- Sensor: Se indica el sensor el cual ha generado la alerta.
- Dato: Se indica cuál de los tres tipos de datos ha sido el causante de la alerta.
- Valor: Se indica el valor que proporcionaba el sensor en el momento de la alerta.
- Rango: Se indica el rango válido que estaba configurado en el sensor en el momento de la alerta.
- Fecha: Se indica la fecha en la que se originó la alerta.

Por último, al igual que se observó en el apartado 5.6, se ha dispuesto un botón que permite descargar la tabla en formato xlsx o csv, tal como se visualiza en la Figura 52.

The screenshot shows a web application interface with a dark blue header. On the left, there is a red circular logo with 'TFG' and a menu icon. On the right, there are links for 'Registro', 'Contacto', and a 'Logout' button. Below the header, there are four tabs: 'Una hora', 'Un día', 'Una semana', and 'Personalizado'. A table is displayed with the following data:

Sensor	Dato	Valor	Rango	Fecha
1	Temperatura	50	30-40	2021-08-15 19:00
2	Humedad	20	25-30	2021-08-15 19:01

Below the table is a green button labeled 'Descargar tabla'. At the bottom of the interface, there are two green buttons: 'Descargar CSV' and 'Descargar XLSX'.

Figura 52: Exportación de Alertas

6. Conclusiones.

La finalidad de este Trabajo de Fin de Grado se centra en la puesta en marcha de una aplicación web para su uso y explotación en industrias del sector agrícola, tratando de mejorar la rentabilidad de las mismas, creando una herramienta de análisis que ayude a los usuarios en su trabajo.

La puesta en marcha de este desarrollo web surge debido al proceso de digitalización en el que se encuentra inmerso todo el sector industrial en los últimos años. En concreto, se busca la creación de un sistema propio con finalidades similares a los llamados HMI o SCADA, en los cuales, y mediante un simple vistazo, se puede obtener información en tiempo real de aquellas máquinas o elementos que se encuentran físicamente en las instalaciones de nuestro negocio.

Para ello, se han estudiado las diferentes tecnologías actuales que nos brindan las mejores opciones para desarrollar este tipo de plataformas online. Después de analizar las ventajas e inconvenientes que tenían cada una de las posibilidades identificadas, se decidió finalmente el uso del *framework* desarrollado por Google, llamado Angular, para la realización de una de las partes del proyecto, el *front end* de nuestra aplicación web.

Angular, previamente comentado en el apartado 3.1 del presente documento, permite realizar un desarrollo web basado en componentes mediante el lenguaje de programación Type Script, a la vez que podemos modificar el DOM mediante ficheros HTML y el estilo de nuestros elementos con códigos basados en CSS. Además, resulta sencillo utilizar librerías externas que sirven de gran ayuda como *ng2-chart* para la creación y gestión de gráficos, el módulo de descargas *xlsx* para el tratamiento externo de los datos, o la librería de *drag and drop* propia de Angular Material indispensable para este proyecto.

Puede afirmarse que se han utilizado tecnologías modernas y actualizadas, proporcionando seguridad y robustez al código y a los usuarios, además de obtener un rendimiento óptimo ajustándose a un espacio ocupado razonable en el servidor en el que se instale.

Antes de comenzar con el desarrollo del código que nos posibilitará la creación y el uso de la aplicación, se ha realizado un estudio de requisitos, tanto funcionales como no funcionales, en el que se recogen aspectos claves necesarios de la aplicación. Acto seguido, se han visualizado los casos de uso en el diagrama de Casos de Uso, que es uno de los artefactos que proporciona el Lenguaje Unificado de Modelado. Esta tarea de análisis será de gran utilidad en el posterior desarrollo del código.

En último lugar, y como capítulo anterior a estas conclusiones, se ha desarrollado un manual de usuario en el que se presentan los diferentes casos de uso desarrollados, además de aquellos posibles errores cometidos por un usuario básico en el uso normal de la aplicación, mediante las capturas de pantalla oportunas. Su finalidad es ayudar a aquellos usuarios destinatarios de la aplicación a comprender el funcionamiento de la plataforma, sirviendo de ayuda adicional en caso de duda.

Esta aplicación resulta de gran ayuda a los destinatarios, ya que serán capaces de observar la información proporcionada por sus sistemas en tiempo real de una manera sencilla, precisa y cómoda. Tendrán disponible también la opción de generar gráficas correspondientes a espacios temporales concretos, permitiéndoles realizar un seguimiento de sus dispositivos. Además, será sencillo instalar o eliminar nuevos componentes, al igual que manejar los diferentes usuarios que tienen acceso a la aplicación.

En el plano personal, la realización de este Trabajo de Fin de Grado ha supuesto un reto en la validación del título proporcionado por la Universidad de Valladolid, debido a las condiciones actuales en las que estamos inmersos. Aun así, y tras haber descubierto el mundo del desarrollo de aplicaciones web con gran entusiasmo a lo largo del grado, la creación de un proyecto amplio desde el inicio, con la necesidad de interactuar entre varios equipos, asemejándose más al mundo laboral real y obteniendo requisitos de usuarios reales, ha supuesto un salto importante en el desarrollo personal.

Por otra parte, trabajar con un *framework* tan demandado hoy en día como el utilizado en este proyecto, posibilita mi lanzamiento en este sector, tras haber adquirido cierto grado de conocimiento en el tratamiento de tecnologías como Type Script, HTML, o librerías externas como Angular Material para el diseño web.

La Industria 4.0 conlleva el uso de multitud de nuevas tecnologías, siendo el desarrollo de aplicaciones web una parte de estas. Supone un reto dominar estos lenguajes de programación y crear sistemas cada vez más sencillos e intuitivos de cara a facilitar el trabajo a los usuarios de los mismos.

7. Líneas futuras.

En este último apartado del documento se explican posibles líneas sobre las que continuar con el desarrollo del proyecto. En un primer momento, se presentan las tendencias que probablemente sigan los desarrollos web como el realizado, para después concretar algunos aspectos importantes con los que se puede dotar a la aplicación, si su desarrollo se continúa, para ofrecer una mejor experiencia de usuario.

El mundo del desarrollo web cada vez se vuelve más necesario, pero también más complejo. Multitud de pequeñas empresas y particulares desean obtener una aplicación web en la que apoyarse para obtener beneficios en sus proyectos personales, bien de cara a promocionar su propia marca personal o bien para mejorar su situación laboral.

Serán entonces los ingenieros de diseño y desarrollo web los encargados de crear estos servicios con un estilo diferenciado aportando todas las funcionalidades requeridas, siendo estas cada vez más complejas y precisas. Si bien es cierto que cada vez se obtienen funcionalidades más interesantes, estas han de poder desarrollarse de una manera intuitiva para el usuario, no siendo necesarias muchas horas de entrenamiento por parte del mismo para sacar el máximo rendimiento posible a la herramienta creada.

En cuanto al desarrollo de esta aplicación, el planteamiento inicial se presentó en el apartado 1.3, junto con la captura de requisitos explicada detalladamente en el apartado 4.1. Aun así, como cualquier proyecto, está sujeto a cambios sustanciales y a posibles mejoras.

En primer lugar, esta aplicación está destinada a un usuario final concreto, orientándose al mundo industrial, en especial al sector de la agricultura y la ganadería. En un futuro se podría estudiar la extrapolación de este sistema a cualquier empresa o particular que desee instalar algún tipo de sensor en sus instalaciones, independientemente del sector al que pertenezca. Para ello, habría que dotar a la aplicación de una capa de abstracción en la cual poder manejar cualquier tipo de situación en la que el usuario final obtenga un beneficio.

Otro aspecto clave que dotaría a nuestro desarrollo de un nivel mayor de funcionalidad sería la posibilidad de crear alertas si se llegasen a cumplir ciertas condiciones, avisando en tiempo real al usuario mediante herramientas como email o SMS con información de lo que está sucediendo en ese preciso momento en alguna de sus instalaciones.

Si en un momento dado se permitiera acceder a usuarios al sistema desde cualquier ubicación, sería recomendable dotar al sistema de una capa de seguridad extra en las comunicaciones que realiza el sistema de *front end* desarrollado y descrito en este documento, con el sistema de *back end* mediante el protocolo HTTPS (*HyperText Transfer Protocol Secure*), impidiendo la interceptación de los mensajes entre ambas partes por personas ajenas no autorizadas.

Por otra parte, si se deseara promocionar la aplicación desarrollada y abrir su uso a cualquier usuario que deseara utilizar la misma, sería interesante realizar tanto un

estudio de accesibilidad para el código creado como un análisis de posicionamiento en buscadores o SEO (*Search Engine Optimization*).

Mediante una mejora en la accesibilidad proporcionaríamos un código capaz de cubrir las necesidades de aquellos usuarios con algún tipo de carencia en aspectos vitales para el correcto uso de la aplicación, como, por ejemplo, personas con deficiencia visual.

Por otra parte, las técnicas SEO nos permiten alcanzar una posición ventajosa a la hora de promocionarse en las búsquedas de usuarios online a través de las palabras claves más importantes de nuestro ámbito. Según estudios realizados por empresas del sector del marketing digital como *Rock Content* (Mousinho, A. ,2021), o la Cámara de Comercio de Valencia (Cámara de Comercio de Valencia, 2021), más del 90% de personas que realizan algún tipo de búsqueda en Google, no pasan de la primera página de resultados para acceder a aquello que estaban buscando, convirtiéndose en un punto diferencial nuestra posición en dichas búsquedas para atraer más público a utilizar nuestra aplicación.

Como una última opción de desarrollo viable, y debido a que los *smartphones* o dispositivos inteligentes son cada vez más utilizados para acceder a cualquier aplicación, es lógico pensar en la creación de una aplicación para dispositivos móviles que cumpla con las especificaciones de este proyecto. El acceso a través de una aplicación instalada en su dispositivo puede resultar más útil y cómodo a algunos usuarios que el acceso a través de una web. Por ello, la creación de una aplicación para dispositivos móviles que brinde las mismas funcionalidades que este desarrollo será una opción plausible en el futuro desarrollo del proyecto.

8. Referencias.

Statista. (2021). *Worldwide digital population as of January 2021*. <https://www.statista.com/statistics/617136/digital-population-worldwide/>

Kemp, S. (2021). *Digital in Spain: All the Statistics You Need in 2021*. DataReportal – Global Digital Insights. <https://datareportal.com/reports/digital-2021-spain>

Open Automation Software. (2021). *IIOT Software Platform, Industrial Internet of Things*. <https://openautomationsoftware.com>

Ecava IGX Web SCADA. (2020). *Pure Web SCADA with HTML5, CSS3 & SVG*. <https://www.integraxor.com>

QuickHMI. (2019). *QuickHMI professional SCADA- / HMI- software S7, Modbus, Twincat, OPC, KNX/EIB. Allen Bradley, Rockwell, MQTT – QuickHMI*. <https://www.quickhmi.com/start.html>

Pvbrowser. (2021). *Pvbrowser – The Process Visualization Browser. HMI and Scada for every platform*. <https://pvbrowser.de/pvbrowser/index.php?lang=en&menu=1>

CuteHMI. (2020) *CuteHMI*. <https://cutehmi.kde.org/docs/master/>

Smart HMI. (2021). *WebIQ – The Web HMI / SCADA Toolbox For Professional HMIs*. <https://www.smart-hmi.com>

GenLogic. (2019). *Real-Time HMI and SCADA for C/C++/C#.NET, Java, HTML5 & Java Script, Linux, Windows, Web, Emdeded and Mobile*. <http://www.genlogic.com>

MySCADA Technologies. (2020). *mySCADA Technologies – Active Control of the technology*. <https://www.myscada.org/solutions-active-control/>

Siemens México. (2019). *Software Siemens México*. <https://new.siemens.com/mx/es/productos/automatizacion/industry-software/automation-software/tia-portal/software.html>

NodeJS. (2017). *NodeJS*. <https://nodejs.org/es/>

Angular. (2020). *Angular*. <https://angular.io/guide/file-structure>

Enríquez, F. (2019). *Angular 8 Paso a paso. Preparación del ambiente e Introducción a Angular*. Medium. <https://medium.com/@fernando.enriquez.silerio/angular-8-paso-a-paso-preparaci%C3%B3n-del-ambiente-e-introducci%C3%B3n-a-angular-dbc9f7c08c50>

Material. (2020). *Material*. <https://material.io>

Angular Material. (2020). *Angular Material*. <https://material.angular.io/components/categories>

Amazon Web Services, Inc. (2020). *Using AWS Cloud9*. <https://aws.amazon.com/es/cloud9/>

Google Trends. (2021). *Google Trends*. https://trends.google.com.br/trends/explore?q=%2Fm%2F0134xwrk,%2Fm%2F0_x5x3g,%2Fm%2F01fchg,%2Fm%2F0k2ki45

Microsoft. (2016). *Documentation for Visual Studio Code*. <https://code.visualstudio.com/docs>

Bartolomé Sintés, M. (2020). *Visual Studio Code. Informática. Bartolomé Sintés Marco*. <https://www.mclibre.org/consultar/informatica/lecciones/vsc.html>

Atlassian. (2021). *Qué es Git*. <https://www.atlassian.com/es/git/tutorials/what-is-git>

Universidad de Salamanca. (2018). *Fundamentos de la vista de casos de uso*. <https://repositorio.grial.eu/bitstream/grial/1155/1/UML%20-%20Casos%20de%20uso.pdf>

Aprendiendo Arduino. (2020). *Qué es Node-RED*. <https://aprendiendoarduino.wordpress.com/2020/03/05/que-es-node-red/>

GitHub. (2021) *frangoteam/FUXA*. <https://github.com/frangoteam/FUXA>

Donohue, B. (2021). *¿Qué Es Un Hash Y Cómo Funciona?* Blog oficial de Kaspersky. <https://latam.kaspersky.com/blog/que-es-un-hash-y-como-funciona/2806/>

Cámara de Comercio Valencia. (2021) *Qué es el SEO y por qué es importante para las empresas*. <https://www.mastermarketing-valencia.com/marketing-digital/blog/que-es-el-seo/>

Mousinho, A. (2021). *SEO: la guía completa para que conquistes la cima de Google en el 2021*. Rock Content. <https://rockcontent.com/es/blog/que-es-seo/>