



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA INFORMÁTICA (COMPUTACIÓN)

**Aplicación para el control y monitorización de un
sistema de reutilización de agua**

**Application for the control and monitoring of a water
reuse system**

Realizado por
Carles Mulugeta Bordas

Tutorizado por
Enrique Soler Castillo

Departamento
Departamento de Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, SEPTIEMBRE DE 2020

Fecha defensa: octubre de 2020

Resumen

Las soluciones industriales para la automatización de procesos requieren una exhaustiva supervisión, de forma que se pueda verificar la correcta actividad de un sistema en funcionamiento y poder modificar parámetros de su comportamiento en tiempo real. Para solucionar esto, es común distribuir sensores por el sistema a fin de conocer el estado de los distintos componentes. El problema de esta solución es que se requiere de la presencia de un operario en la zona de instalación para revisar el estado de estos. Es por esto por lo que a estos sensores se le suman elementos de telemetría, de forma que puedan consultarse las mediciones de los sensores de forma remota.

Este proyecto consiste en el desarrollo de una aplicación accesible desde la web para la monitorización y control de un sistema de reutilización de agua en torres de refrigeración, de forma que se comunique con los elementos telemétricos instalados. La aplicación dispone de un panel que muestra el estado y las mediciones del sistema en tiempo real, además de permitir realizar acciones sobre los elementos de control distribuidos por la instalación. De forma adicional, la aplicación permite consultar estas mediciones a lo largo del tiempo, registrando los valores medidos por los sensores y almacenándolos en caso de tener que revisarlos. Por último, se incorpora una sección que permite visualizar de forma interactiva el sistema a modo de plano interactivo de la instalación, facilitando la interacción con el panel y proporcionando una interfaz amigable para los usuarios.

Palabras clave: Aplicación, Monitorización, Web, Software industrial, Panel interactivo

Abstract

Industrial solutions for process automation require exhaustive supervision, so that the correct activity of a running system can be verified, and its behavior parameters can be modified in real time. To solve this, it is common to distribute sensors throughout the system in order to know the status of the different components. The problem with this solution is that it requires the presence of an operator in the installation area to check their status. Therefore, telemetry elements are added to these sensors, so that the sensor measurements can be consulted remotely.

This project consists of the development of an application accessible from the web for the monitoring and control of a water reuse system in cooling towers, so that it communicates with the telemetric elements installed. The application has a panel that shows the status and measurements of the system in real time, as well as allowing actions to be taken on the control elements distributed by the installation. Additionally, the application allows you to consult these measurements over time, recording the values measured by the sensors and storing them in case they are to be reviewed. Finally, a section is incorporated that allows the system to be viewed interactively as an interactive plan of the installation, facilitating interaction with the panel and providing a friendly interface for users.

Keywords: Application, Monitoring, Web, Industrial software, Interactive panel

Índice

1	Introducción.....	1
1.1.	Motivación.....	1
1.2.	Objetivos.....	4
2	Tecnologías utilizadas.....	5
2.1.	Tecnologías de cliente.....	5
2.2.	Tecnologías de servidor.....	6
2.3.	Microsoft SQL Server.....	8
3	Metodología de trabajo.....	9
3.1.	Desarrollo en cascada.....	9
4	Análisis de requisitos.....	13
4.1.	Requisitos funcionales.....	13
4.2.	Requisitos no funcionales.....	14
5	Diseño de la aplicación.....	15
5.1.	Especificación de alto nivel.....	15
5.2.	API REST.....	16
5.3.	Modelo Entidad-Relación.....	16
5.1.	Diagrama de casos de uso.....	18
5.2.	Diseño de la base de datos.....	19
6	Desarrollo de la aplicación.....	21
6.1.	Desarrollo de la aplicación de servidor.....	21
6.2.	Desarrollo de la aplicación de cliente.....	24
6.3.	Desarrollo del sistema de autenticación.....	35
6.4.	Usuarios en la base de datos.....	35
6.5.	Obtención de autorización.....	36

6.6.	Inicio de sesión en cliente.....	36
7	Pruebas y validación.....	39
7.1.	Prueba de acceso a la aplicación.....	39
7.2.	Prueba de visualización de históricos.....	40
7.3.	Prueba de modificación de los valores de consigna.....	41
7.4.	Pruebas de generación de informes CSV.....	41
8	Conclusiones y líneas futuras.....	43
8.1.	Menús flexibles.....	43
8.2.	WebSocket.....	43
	Referencias.....	45
	Apéndice A Manual de Instalación.....	47
	Requerimientos:.....	47

1

Introducción

1.1. Motivación

La mayoría de los sistemas de climatización en edificios no residenciales en el sur de Europa se basan en sistemas de producción con bombas de calor y enfriadoras que intercambian energía con el aire para asegurar que se cumplan con las condiciones de confort en los espacios a climatizar. Estos sistemas de producción consumen energía eléctrica para trasvasar energía térmica entre espacios a diferentes temperaturas (entre los propios espacios a climatizar y el exterior); para realizar la transferencia de energía al exterior, estos sistemas cuentan con unidades de condensación que realizan la transferencia de calor al medio ambiente u otros sumideros mares, ríos, lagos, etc.



Ilustración 1: Ejemplo de unidad de refrigeración instalada en un edificio

Típicamente, hay dos maneras de hacer la transferencia de calor en las unidades de condensación [1]:

- Sistemas refrigerados por aire [2]. Utilizan el aire ambiente para enfriar el fluido de trabajo y es la solución más extendida en edificios de oficinas. En este caso la temperatura del aire exterior es la que marca el rendimiento esperado de los equipos de producción, en lugares con temperaturas exteriores elevadas, los rendimientos de los equipos son menores.
- Sistemas refrigerados por agua [2]. Estos sistemas se basan en el uso de agua como fluido refrigerante para eliminar el calor de las unidades de condensación. Se usan en lugares donde el agua es un recurso disponible y barato. Las torres de enfriamiento son un ejemplo. En este caso la temperatura del agua es la que marca el rendimiento de los sistemas de producción, generalmente, la temperatura del agua suele ser inferior a la temperatura ambiente por lo que presentan mejor rendimiento.

Los sistemas refrigerados por agua, como es el caso de las torres de refrigeración, se usan de manera generalizada en grandes edificios como hospitales, centros deportivos, centros comerciales, etc., la razón principal radica en la mayor eficiencia energética. Los sistemas enfriados por aire funcionan con el límite de rendimiento marcado por la temperatura de bulbo seco, que siempre está por encima de la temperatura de bulbo húmedo (que es a la temperatura a la que trabajarán las torres de enfriamiento). Esta diferencia de temperatura en el proceso de condensación conduce a un mayor rendimiento según el principio de Carnot [3], esta mejora en el rendimiento se traduce en un menor consumo de energía eléctrica para cubrir la misma demanda de energía térmica. En términos de indicadores de eficiencia energética los equipos de climatización se caracterizan con los parámetros EER y COP (para unas condiciones fijas), y el SEER y SCOP (para la evaluación estacional). Todos estos parámetros muestran mejores resultados para los sistemas refrigerados por agua. Sin embargo, las torres de enfriamiento muestran las siguientes desventajas:

- Se produce un consumo eléctrico por las bombas de distribución de agua que son necesarias.
- Se produce un consumo eléctrico en la bomba de recirculación en torres de refrigeración cerrada o torres de refrigeración híbridas.
- Se producen pérdidas de agua por evaporación, purga y arrastre, que requieren reposición de agua de red en el circuito.
- Costo de productos químicos para el tratamiento del agua.
- Mayor inversión inicial.
- Costo relacionado con análisis de laboratorio para prevenir la aparición de legionelosis.
- Costos de operación y mantenimiento relevantes.

En aquellos emplazamientos en los que la temperatura ambiente es alta (como es el caso de la costa andaluza), este tipo de sistemas muestran un rendimiento mucho más elevado; además cuando los edificios muestran una elevada demanda por su tamaño, este tipo de sistemas consiguen un ahorro de energía significativo en términos absolutos y hace que las inversiones necesarias a realizar sean rentables. Por otro lado, todas las climatizadoras (elementos de los sistemas de climatización que tratan el aire para impulsarlo en las condiciones requeridas a los espacios a climatizar) producen agua de condensación sobre la superficie de los serpentines de las baterías de frío y, por lo general, esta agua se suele verter a la red de saneamiento. Esta agua presenta una serie de ventajas que las hacen idóneas para ser reaprovechadas en las torres de refrigeración, mejorando el rendimiento de los equipos y reduciendo el consumo neto de agua de red.

1.2. Objetivos

Este trabajo se engloba dentro de un proyecto de I+D para el desarrollo de un sistema de recuperación del condensado de las climatizadoras para su aprovechamiento en las torres de refrigeración para edificios de gran consumo, fundamentalmente industriales y sector terciario, este diseño se validará en un edificio real, equipado con climatizadoras y torres de refrigeración, para demostrar las mejoras esperadas en términos de impacto ambiental y eficiencia del sistema (reducción en el consumo de energía eléctrica, reducción en el consumo de agua de red, reducción de las pérdidas por purgas y menor mantenimiento). Esta innovadora solución para investigar se plantea bajo un enfoque de economía circular, en el que un “residuo”, el agua, es utilizado como materia prima para otro proceso con el fin de mejorar el comportamiento global del sistema.

2

Tecnologías utilizadas

2.1. Tecnologías de cliente

En la siguiente sección se exponen las tecnologías empleadas para el desarrollo de la aplicación de cliente. Esta aplicación es la encargada de mostrar la información del sistema e interactuar con el usuario para que éste pueda realizar las distintas acciones sobre el sistema en un entorno gráfico a través de la web.

2.1.1. Angular

Angular [4] comprende un conjunto de herramientas para JavaScript gratuitas y de código abierto creadas por Google para el desarrollo de páginas y aplicaciones web SPA (*Single Page Application*). Este modelo de carga permite la navegación por la página sin necesidad de recargar por completo esta, haciendo que cuando el usuario entra en la web se carga todo el contenido de todas las páginas a la vez. Esto hace que la primera carga nada más entrar es más lenta pero luego los cambios entre páginas son instantáneos. Además, solo se utiliza un fichero con el código de la web por lo que no se requiere de muchas llamadas a servidor.



*Ilustración 2: Logo de Angular
Usado bajo licencia CC BY 4.0 [17]*

Otra de las características de Angular es que utiliza TypeScript [5], una extensión del lenguaje JavaScript la cual se convierte en última instancia a este último. Esta extensión proporciona tipado estricto, interfaces, inyección de dependencias y más cosas.

La metodología de trabajo con Angular se enfoca en el uso en dos conceptos; El patrón modelo-vista-controlador y la distribución en componentes. Esta forma de desarrollo permite organizar proyectos de grandes dimensiones en pequeños componentes y la composición de éstos.

Estas y otras utilidades proporcionadas por las herramientas ayudan al desarrollo de aplicaciones dinámicas donde los elementos mostrados en pantalla se ven modificados en tiempo real, razón por la cual se ha decidido incluir en el desarrollo de esta aplicación donde la visualización de información dinámica es una pieza clave.

2.1.2. Bootstrap

Bootstrap [6] es una librería inicialmente desarrollada por Twitter que incluye estilos predefinidos para elementos visuales web (botones, campos de texto, iconos, ...). Estos elementos facilitan enormemente la maquetación de páginas web, ya que permiten crear una interfaz muy limpia y completamente responsiva, es decir, adaptable a cualquier tamaño de pantalla (uno de los requisitos de la aplicación).

2.2. Tecnologías de servidor

ASP.NET Core [7] es un marco multiplataforma de código abierto y de alto rendimiento que tiene como finalidad compilar aplicaciones modernas conectadas a Internet y habilitadas para la nube.

La utilidad desarrollada por Microsoft facilita el desarrollo de aplicaciones de servidor proporcionando abstracciones sobre los puntos de comunicación con el servidor, dividiendo estos en controladores.

Esta aplicación conforma la pieza central del sistema, dado que será la encargada tanto de proveer a los navegadores web de la aplicación de Angular como de proveer a esta última de la información almacenada en la base de datos para su visualización por el usuario.

2.2.1. OpenAPI

La especificación OpenAPI [8] define una interfaz estándar independiente del lenguaje para las interfaces que cumplen con la especificación REST (interfaces *RESTful* [9]), permitiendo que tanto los humanos como las computadoras descubran y comprendan las capacidades del servicio sin acceso al código fuente, documentación o mediante la inspección del tráfico de la red. Cuando se define correctamente, un consumidor puede comprender e interactuar con el servicio remoto con una cantidad mínima de lógica de implementación.

OpenAPI también provee de herramientas de generación de documentación pueden usar una definición de OpenAPI para mostrar la API, herramientas de generación de código para generar servidores y clientes en varios lenguajes de programación, herramientas de prueba y muchos otros casos de uso. La implementación concreta que utilizar será la librería de código abierto NSwag de Rico Suter, la cual proporciona herramientas para generar especificaciones OpenAPI a partir de controladores de ASP.NET Core existentes y código de cliente a partir de estas especificaciones.

2.2.2. Entity Framework Core

Entity Framework Core [10] es una herramienta de código abierto desarrollada por Microsoft para la plataforma .NET. Esta herramienta permite trabajar con clases de C# y que los cambios sobre estas queden persistidos sobre una base de datos relacional. Esta funcionalidad permite al desarrollador abstraerse de las consultas SQL concretas a efectuar y se limita a realizar cambios sobre un lenguaje ya conocido.

Entity Framework Core soporta varios motores de bases de datos, haciendo que ésta pueda ser reemplazada en cualquier momento sin necesidad de reescribir el código de la aplicación.

2.2.3. JWT

JSON Web Token (JWT) [11] es un estándar abierto el cual define una forma compacta y autocontenida de transmitir información entre entidades en formato JSON [12]. Esta tecnología permite la gestión de permisos a los usuarios mediante la otorgación a los usuarios de una cadena secreta que representa de los permisos que disponen sobre la

aplicación. Esta cadena secreta es inmodificable, de forma que los usuarios no pueden manipular los permisos de los que disponen. Estas cadenas secretas únicamente pueden ser emitidas por una entidad autorizada, dado que para su generación es necesario el uso de un valor secreto únicamente conocido por la aplicación.

2.3. Microsoft SQL Server

Microsoft SQL Server [13] es un sistema de gestión de bases de datos relacional desarrollado por Microsoft. Esta herramienta será utilizada para el almacenamiento y la persistencia de los datos de la aplicación, de forma que puedan consultarse los registros de mediciones anteriores.

3

Metodología de trabajo

3.1. Desarrollo en cascada

Se empleará una metodología en cascada [14], ampliamente utilizada en el desarrollo de software en el que se disponga de una base sólida de especificación tanto de requisitos como de diseño. La metodología consiste en realizar un primer análisis de los requisitos que debe cumplir el sistema, seguido de realizar un diseño que cumpla dichos requisitos y por último realizar el desarrollo de la aplicación. En cada fase de esta metodología se busca que cada una se base estrictamente en la anterior, evitando así volver atrás en las fases y permite concretar los tiempos dedicados a cada una de estas.

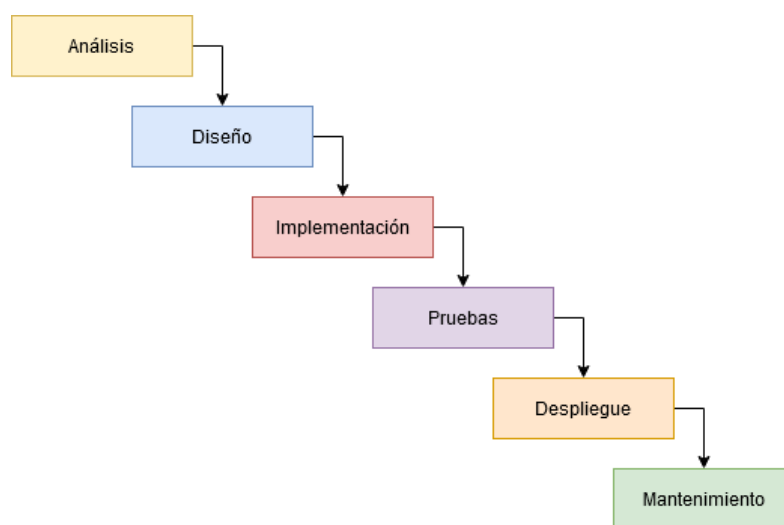


Ilustración 3: Esquema clásico del modelo en cascada

Para este proyecto, se han especificado las siguientes fases:

- Análisis de requisitos y especificación de alto nivel del diseño: Se analizan los requisitos que debe cumplir la aplicación (información a mostrar, control de acceso, ...) y se realizan bocetos de interfaces que la cumplan. Estos servicios, restricciones y metas del sistema se definen a partir de consultas con los usuarios.
- Diseño y especificación de bajo nivel: Se realiza una especificación de los componentes del sistema y su formato de comunicación
- Diseño de la base de datos: Se implementa el diseño realizado de la base de datos de forma que pueda almacenarse toda la información requerida por la aplicación
- Desarrollo de la aplicación de servidor: Se desarrolla la aplicación de ASP.NET Core que se ejecutará en el servidor y que proveerá al cliente de la información del estado del sistema.
- Desarrollo de la aplicación de cliente: Se desarrolla la aplicación de Angular que permitirá a los usuarios acceder a la aplicación y visualizar la información monitorizada por la aplicación, así como tomar acciones sobre el sistema.
- Desarrollo del sistema de autenticación: Se agrega a la aplicación funcionalidad para almacenar y autenticar a los usuarios, de forma que sólo aquellos que dispongan de credenciales válidas puedan acceder a la aplicación.
- Prueba de la aplicación y validación: Se ponen a prueba tanto individualmente como en conjunto todas las partes de la aplicación y se verifica que en su conjunto se cumplen con los requisitos.

La siguiente fase no debe empezar hasta que la fase previa haya finalizado. En la práctica, estas etapas se superponen y proporcionan información a las demás. Durante el diseño se identifican los problemas con los requerimientos; durante el diseño del código se encuentran problemas, y así sucesivamente. El proceso del software no es un modelo lineal simple, sino que implica una serie de iteraciones de las actividades de desarrollo.

Las principales ventajas del modelo en cascada son que la documentación se produce en cada fase y que ésta cuadra con otros modelos del proceso de ingeniería. Su principal problema es su falta de flexibilidad al dividir el proyecto en distintas etapas. Esta división se traduce en que es necesario hacer compromisos en las etapas iniciales, lo que hace

difícil responder a los cambios de requerimientos del cliente. Es por esta razón por lo que modelo en cascada sólo se debe utilizar cuando los requerimientos se comprendan bien y sea improbable que cambien radicalmente durante el desarrollo del sistema.

Es por esta razón que se ha utilizado esta metodología para este proyecto. Por un lado, los requisitos del software se han fijado al inicio, por lo que la falta de flexibilidad del software no es un problema. Por otro lado, la división en fases ha permitido el desarrollo de la presente memoria de forma secuencial tras la finalización de cada fase, haciendo que los capítulos (al igual que las fases) resulten más autocontenidos.

4

Análisis de requisitos

4.1. Requisitos funcionales

A continuación, se describen de forma general los principales requisitos funcionales que debe cumplir la aplicación web que se va a desarrollar en el proyecto.

4.1.1. Control de acceso

La aplicación proporcionará un acceso seguro que permita a los usuarios conectarse al sistema. Este sistema de acceso consistirá en el clásico inicio de sesión basado en credenciales compuestas de un nombre de usuario y una contraseña. El sistema debe asegurar por tanto que únicamente los usuarios que posean dichas credenciales son capaces de acceder a las funcionalidades de monitorización y control del sistema.

4.1.2. Monitorización de datos

La aplicación web deberá proporcionar una interfaz gráfica en la que se pueda visualizar el estado del sistema en base a los valores medidos por los distintos sensores distribuidos por la planta. En concreto, se deben poder visualizar los siguientes elementos:

- Estado de las bombas de impulsión de agua (encendidas o apagadas).
- Mediciones de los caudalímetros y caudal acumulado, pudiendo este último reiniciarse a cero.
- Estado de las válvulas de tres vías, indicando hacia dónde se está divergiendo el caudal.

- Estado de las torres de refrigeración (encendidas o apagadas).
- Medidas de los sensores de temperatura, humedad y conductividad.
- Estado de las válvulas de purga (abiertas o cerradas).
- Nivel de agua de los depósitos.

4.1.3. Configuración de consignas

La aplicación permitirá a los usuarios establecer las consignas de los siguientes parámetros:

- Umbral de nivel de depósito para la activación de las bombas.
- Umbral de conductividad para apertura de válvula de purga.

4.1.4. Histórico de datos

El sistema almacenará datos históricos de al menos 1 mes y medio de antigüedad, y se almacenarán todas las variables monitorizadas descritas en el apartado 4.1.2. Adicionalmente se almacenarán los datos de estados, alarmas, etc. (Estos datos se tendrán que especificar con los miembros del consorcio, a medida que se vaya desarrollando la aplicación).

4.2. Requisitos no funcionales

4.2.1. Interfaz legible

La interfaz gráfica debe permitir identificar con claridad los distintos elementos a monitorizar y sus valores. Además, la navegación por la aplicación ha de ser intuitiva, de forma que se minimice la formación especializada para el manejo de la aplicación.

4.2.2. Aplicación reactiva

La aplicación web debe adaptarse correctamente a todo tipo de tamaños de pantalla, de forma que los elementos se reorganicen dependiendo de las dimensiones disponibles en el dispositivo.

5

Diseño de la aplicación

5.1. Especificación de alto nivel

El sistema se compondrá de tres partes principales:

- Cliente Angular: Será la aplicación web utilizada por los usuarios para la interacción con el sistema. Ofrece una interfaz gráfica con la que los usuarios pueden acceder a la información almacenada en el sistema de forma cómoda e interactiva. Estas aplicaciones no tienen acceso directo a la base de datos, sin que los requieren al servidor mediante el protocolo HTTP.
- Servidor ASP.NET Core: Será el encargado tanto de distribuir a los clientes la aplicación web como de proveer a esta última de la información a mostrar y de los controles sobre el sistema.
- Base de datos SQL Server: Será la encargada del almacenamiento de la información.

En la Ilustración 4 se expone la interacción entre estos componentes. En este caso, los ordenadores personales de los clientes ejecutan la aplicación de Angular, la cual se comunica mediante el protocolo HTTP con el servidor de ASP.NET Core, el cual proporciona tanto el código de la propia aplicación como la información a mostrar. Esta aplicación de servidor a su vez extrae la información de la base de datos SQL Server mediante una conexión TCP/IP y sentencias SQL.

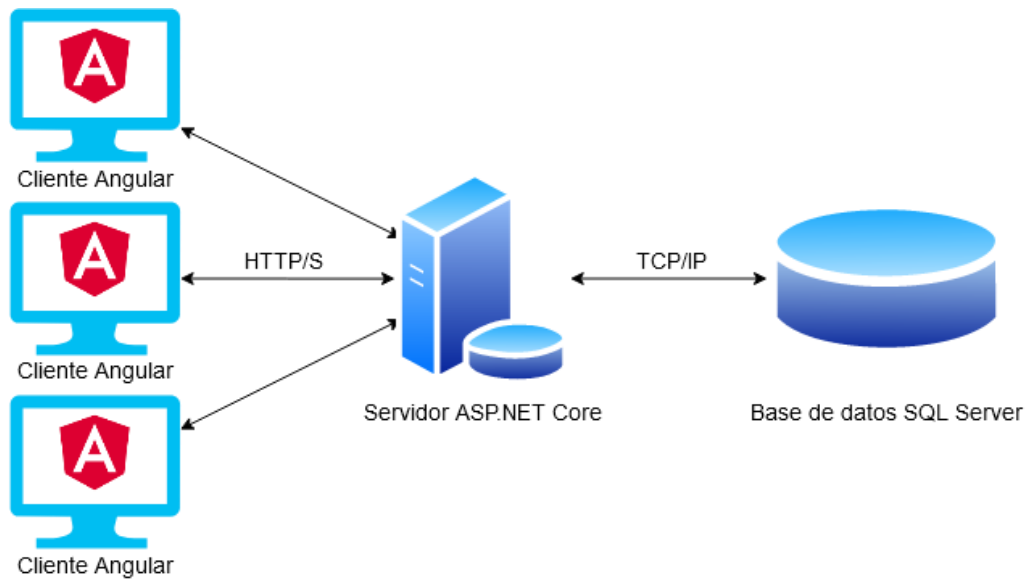


Ilustración 4: Esquema de comunicación / interacción

5.2. API REST

Para la comunicación entre los clientes y la aplicación de servidor es necesario desarrollar una API (*Application Programming Interface*) a la que los clientes puedan realizar peticiones para realizar las distintas acciones sobre el sistema. Para este proyecto, se va a implementar una API siguiendo una arquitectura REST.

La arquitectura REST (*Representational State Transfer*) especifica una serie de normas y restricciones sobre el modo de interacción entre los clientes y el servidor, de forma que se estandarice la comunicación y sea lo más coherente posible. Aunque de base se utiliza el protocolo HTTP para la comunicación, esta arquitectura exige conceptualizar las interacciones entre los clientes y el servidor como cambios en el estado de “recursos”. El servidor ha de definir estos recursos como entidades identificables y referenciales, además de métodos para la modificación de estos recursos en base a sus identificadores.

5.3. Modelo Entidad-Relación

En la Ilustración 5 se muestra un diagrama entidad-relación del sistema, donde se exponen los distintos elementos del sistema (climatizadoras, sensores, torres de refrigeración, ...) además de elementos a almacenar (registros de nivel, de estado).

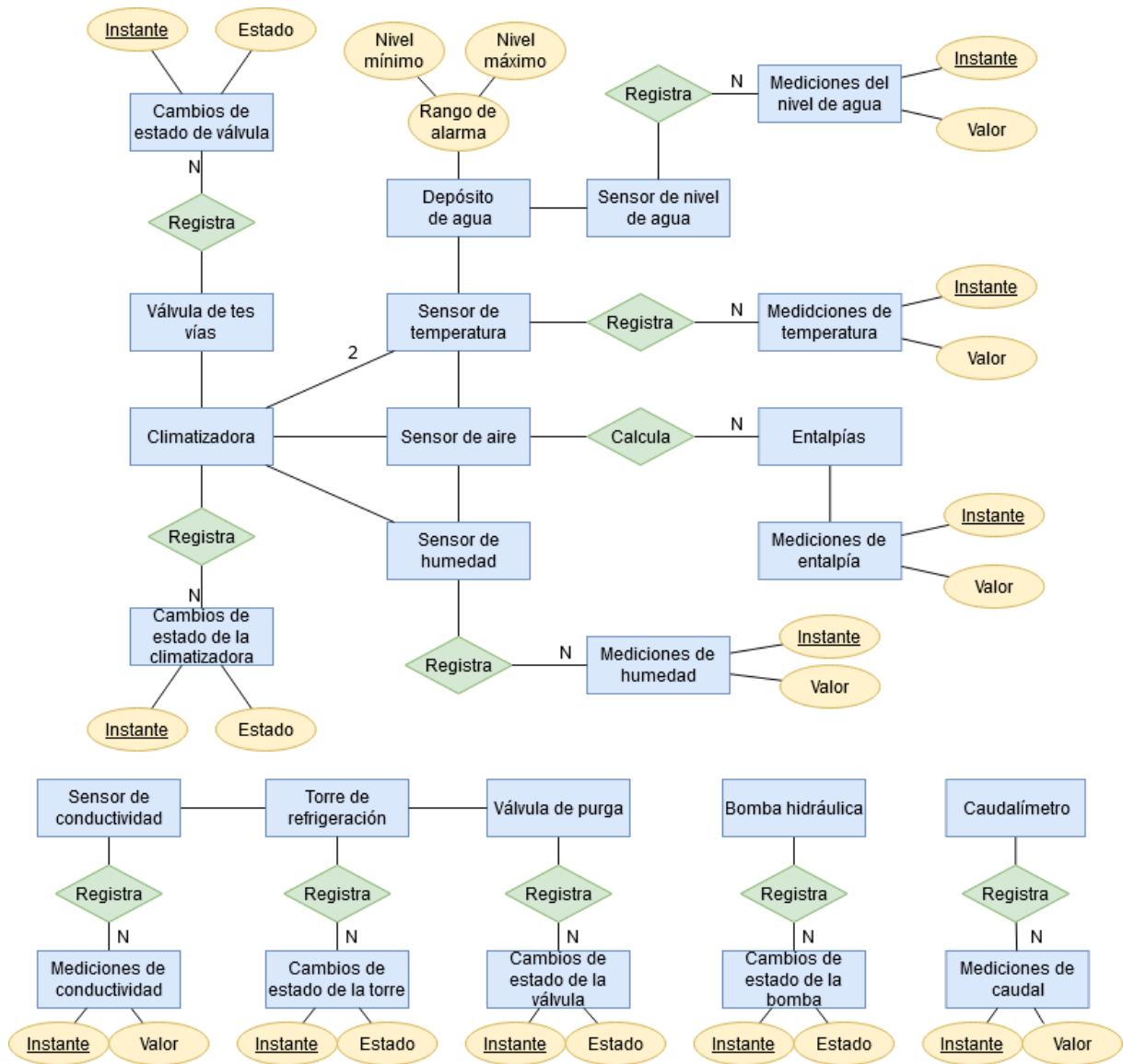


Ilustración 5: Modelo entidad-relación del sistema

En el diagrama se muestra cómo las mediciones de cada sensor se almacenan relacionadas con el sensor asociado, además de almacenar el momento en el que se toma la medición y el valor que alcanza. Este almacenamiento permitirá a la aplicación mostrar un histórico de datos y acceder a los valores medidos a lo largo del tiempo. En el caso de la entalpía, se trata de un valor calculado en base a la temperatura y humedad medidas por los otros sensores.

5.1. Diagrama de casos de uso

En la Ilustración 6 se muestra el diagrama de casos de uso de la aplicación, donde se exponen todas las formas de interactuar con el sistema.

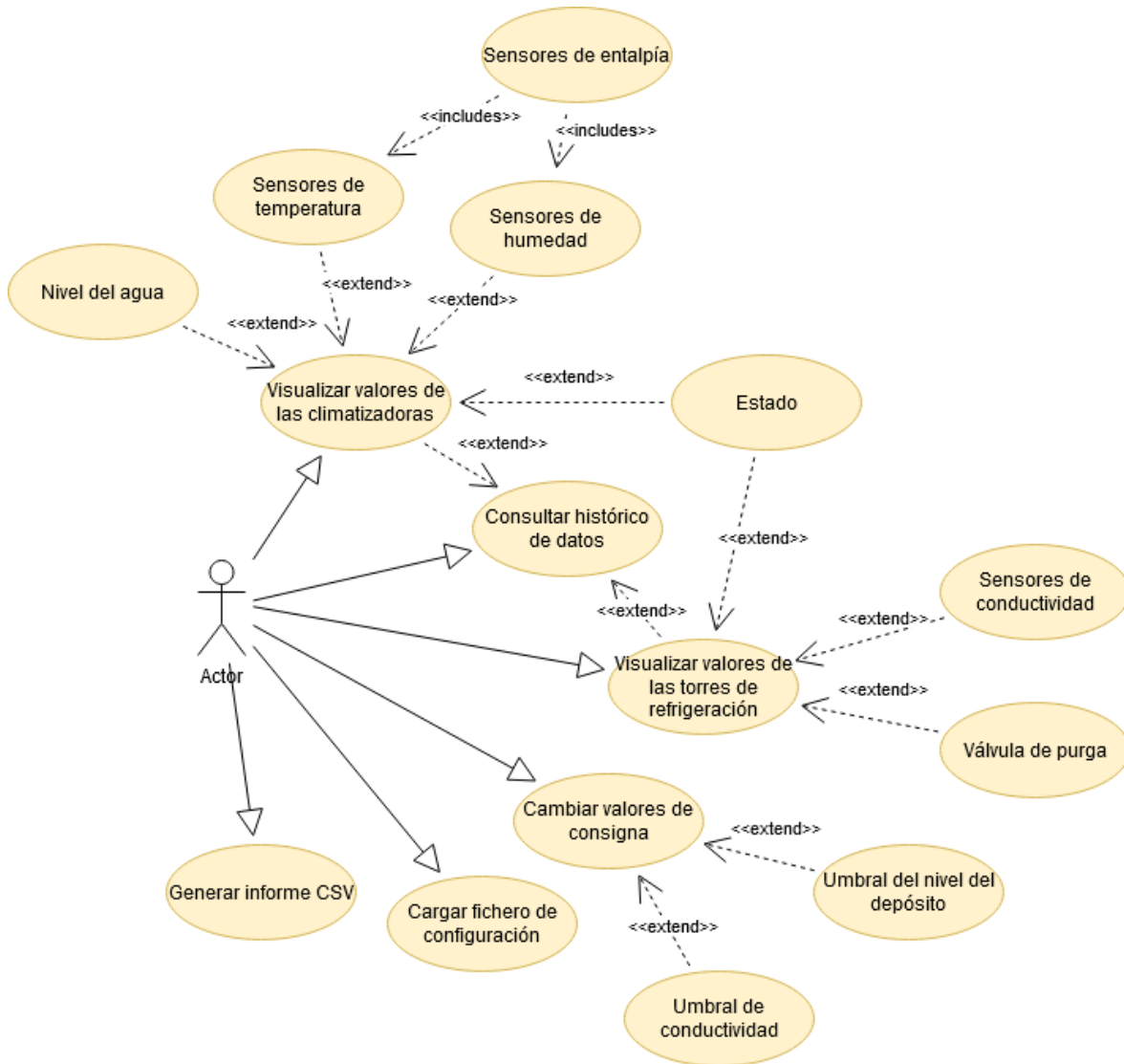


Ilustración 6: Diagrama de casos de uso

Los usuarios autenticados son los únicos con permiso de acceso a la aplicación. Estos pueden visualizar los valores reportados por los elementos de telemetría asociados a las climatizadoras, como son el nivel del agua, los sensores de temperatura y humedad y el estado de las climatizadoras. Los usuarios también pueden consultar los registros previos de los valores reportados, seleccionando el rango de fechas y el tipo de medida a consultar. Todo usuario autenticado también dispone la opción de generar un fichero en formato CSV con la información de registros previos.

5.2. Diseño de la base de datos

La siguiente fase del desarrollo consiste en el diseño del modelo relacional de base de datos. Este modelo debe adecuarse a las necesidades expuestas por el modelo entidad-relación, de forma que pueda almacenarse toda la información requerida para el funcionamiento del sistema. En la Ilustración 7 y la Ilustración 8 se muestra en forma de diagrama el modelo relacional utilizado para la base de datos.

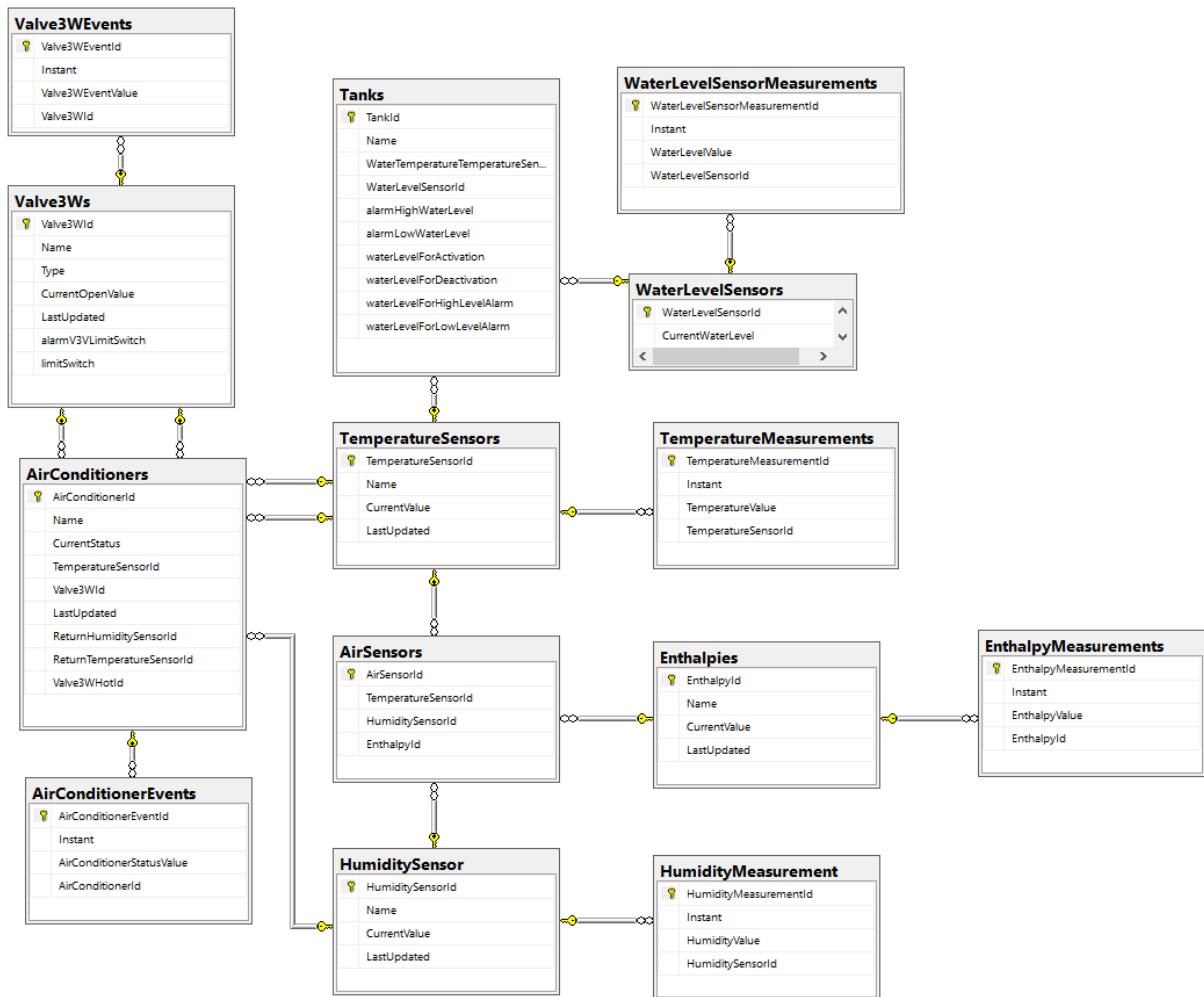


Ilustración 7: Primera parte del modelo relacional del sistema

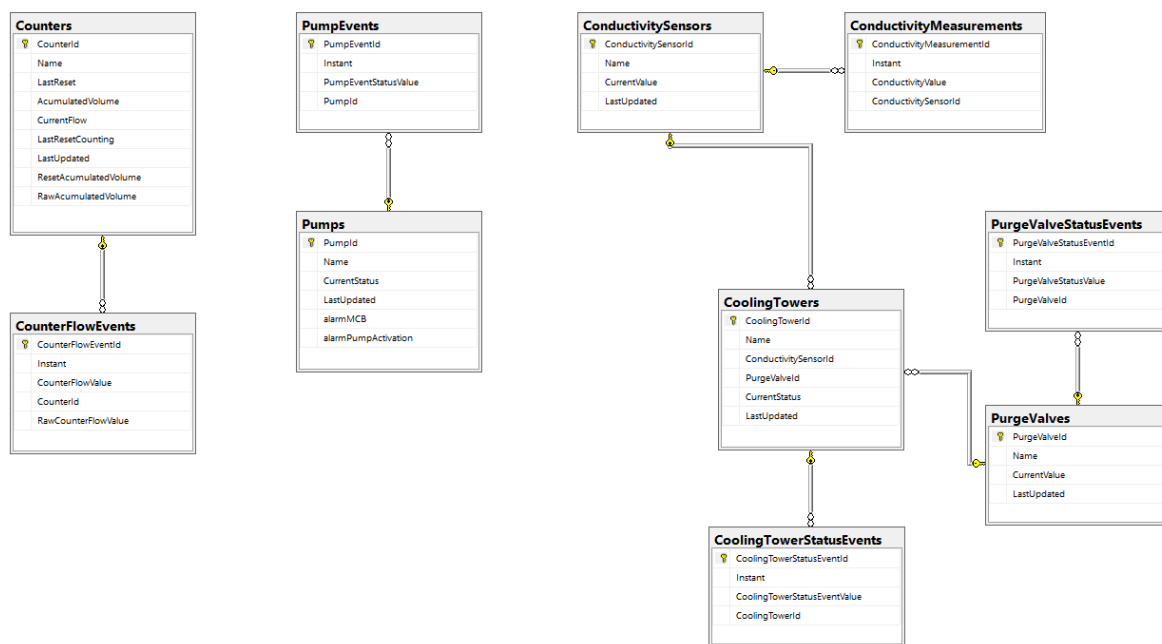


Ilustración 8: Segunda parte del modelo relacional del sistema

Este diagrama adapta los requisitos plasmados en el modelo entidad-relación expuesto en la sección anterior, de forma que se pueda almacenar toda la información requerida.

A todas las entidades se les ha agregado un campo “id” que identifica de forma numérica a cada una de las entidades de las tablas. Las tablas para el almacenamiento de registros de mediciones de telemetría (temperatura, humedad, conductividad, ...) se acompañan de una clave foránea que referencia al sensor concreto que emite las mediciones.

Para el cálculo de la entalpía, se considera un sensor de entalpía virtual, de forma que se combina la información de un sensor de temperatura y un sensor de humedad para generar en base a estos los valores de entalpía en cada momento.

6

Desarrollo de la aplicación

6.1. Desarrollo de la aplicación de servidor

Para el desarrollo de la aplicación de servidor se ha utilizado el lenguaje de programación C# utilizando el marco de trabajo ASP.NET Core. Este entorno de trabajo divide las entradas de la API en clases de C# denominadas controladores. Los distintos métodos de estos controladores son los encargados de procesar las peticiones HTTP que se reciben, mientras que ASP.NET Core se encarga de redirigir las peticiones al controlador correspondiente y de realizar las conversiones necesarias entre los controladores y el protocolo.

Para este proyecto se han desarrollado un total de 16 controladores, de forma que los métodos de éstos permiten obtener información sobre el sistema monitorizado y realizar acciones sobre éste. En la Ilustración 10 se muestra una interfaz generada por la librería NSwag donde se exponen todos los puntos de entrada accesibles por las aplicaciones, así como las direcciones URL y los verbos HTTP a utilizar para conectarse a cada uno de ellos.

AirConditioners ▼	
GET	/api/AirConditioners
GET	/api/AirConditioners/{id}
GET	/api/AirConditioners/{id}/events
AirSensors ▼	
GET	/api/AirSensors
GET	/api/AirSensors/{id}
ConductivitySensors ▼	
GET	/api/ConductivitySensors
GET	/api/ConductivitySensors/{id}
GET	/api/ConductivitySensors/{id}/measurements
CoolingTowers ▼	
GET	/api/CoolingTowers
GET	/api/CoolingTowers/{id}
GET	/api/CoolingTowers/{id}/events
Counters ▼	
GET	/api/Counters
GET	/api/Counters/{id}
PUT	/api/Counters/{id}/reset
GET	/api/Counters/{id}/flowEvents
Enthalpies ▼	
GET	/api/Enthalpies
GET	/api/Enthalpies/{id}
GET	/api/Enthalpies/{id}/measurements
HumiditySensors ▼	
GET	/api/HumiditySensors
GET	/api/HumiditySensors/{id}
GET	/api/HumiditySensors/{id}/measurements
ManagerMeasurementsFile ▼	
POST	/api/ManagerMeasurementsFile/postFile

Ilustración 9: Puntos de entrada de la interfaz web. Interfaz generada por la librería NSwag

Pumps ▼	
GET	/api/Pumps
GET	/api/Pumps/{id}
GET	/api/Pumps/{id}/events
PurgeValves ▼	
GET	/api/PurgeValves
GET	/api/PurgeValves/{id}
GET	/api/PurgeValves/{id}/statusEvents
ReportsCSV ▼	
GET	/api/ReportsCSV
Tanks ▼	
GET	/api/Tanks
GET	/api/Tanks/{id}
PUT	/api/Tanks/{id}/consigna
TemperatureSensors ▼	
GET	/api/TemperatureSensors
GET	/api/TemperatureSensors/{id}
GET	/api/TemperatureSensors/{id}/measurements
Valve3W ▼	
GET	/api/Valve3W
GET	/api/Valve3W/{id}
GET	/api/Valve3W/{id}/events
WaterLevelSensors ▼	
GET	/api/WaterLevelSensors
GET	/api/WaterLevelSensors/{id}
GET	/api/WaterLevelSensors/{id}/measurements

Ilustración 10: Puntos de entrada de la interfaz web. Interfaz generada por la librería NSwag

6.2. Desarrollo de la aplicación de cliente

Tras haber finalizado todo el desarrollo de la aplicación de servidor, el siguiente paso es crear una interfaz gráfica para la visualización de los datos. Esta aplicación realiza peticiones HTTP al servidor de forma asíncrona, con las cuales obtiene la información almacenada en la base de datos y la muestra al usuario de forma organizada.

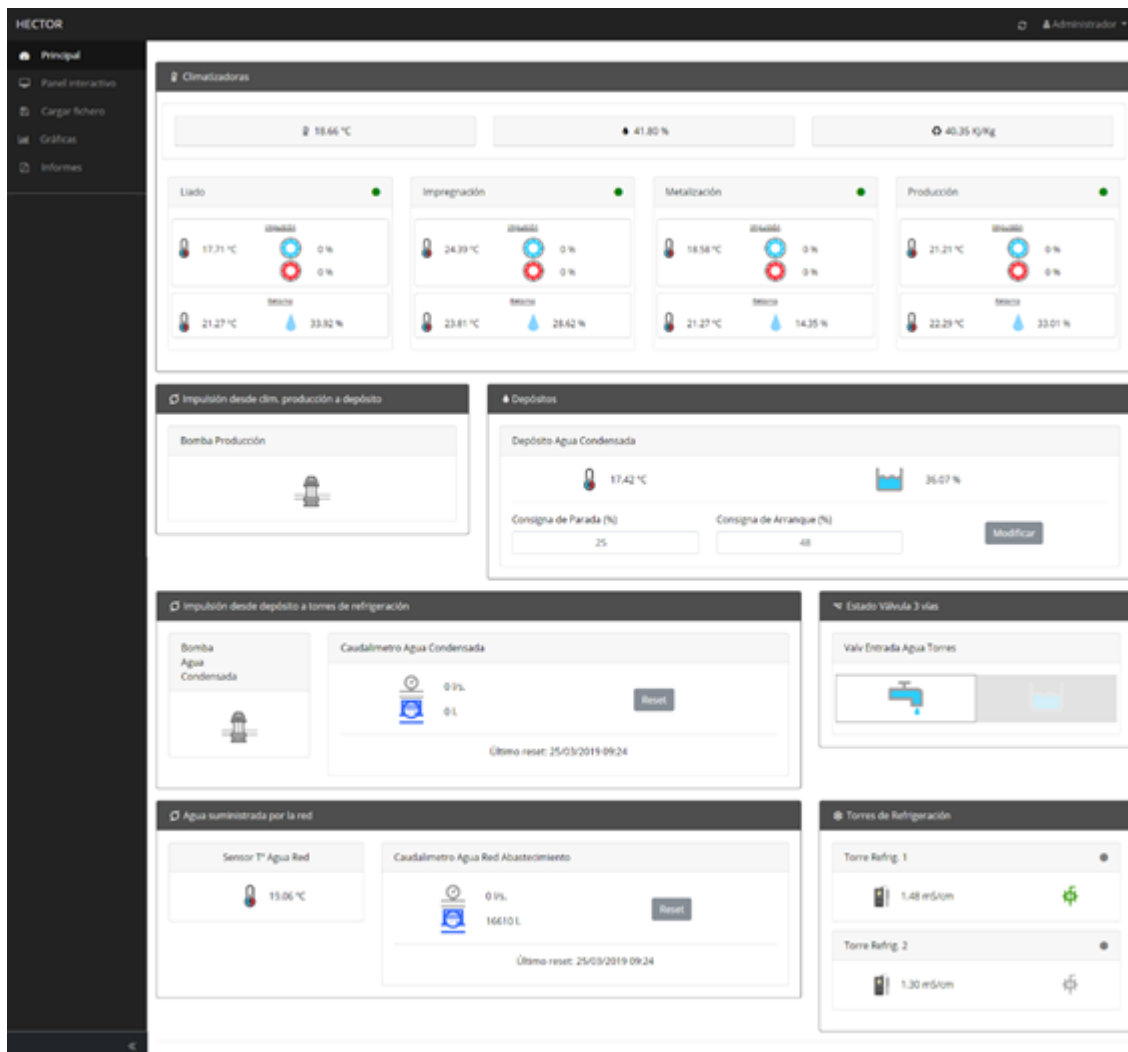


Ilustración 11: Panel principal de la aplicación

En esta pantalla se pueden diferenciar principalmente 3 áreas: el menú lateral izquierdo, el menú situado en la barra superior y la zona central (que es la parte que cambia al seleccionar los diferentes enlaces del menú lateral izquierdo).

6.2.1. Menú superior

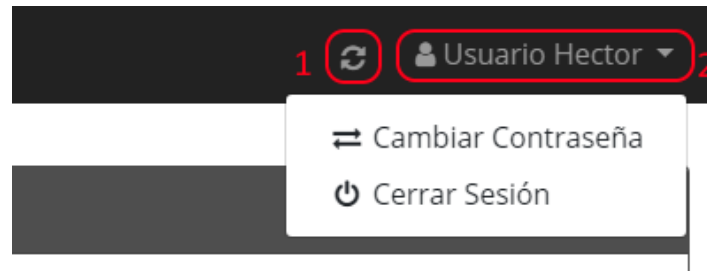


Ilustración 12: Menú superior

En el menú superior permite recargar la información mostrada por la aplicación y opciones sobre el usuario conectado. Estas últimas incluyen cambiar la contraseña del usuario o cerrar la sesión. Para acceder a este menú se debe pulsar en el nombre del usuario y se mostrarán estas dos opciones.

6.2.2. Menú lateral

En el menú lateral situado a la izquierda de la aplicación (Ilustración 13) se muestran un listado de las secciones disponibles en la aplicación. Al pulsar sobre cualquiera de estas secciones el usuario es redirigido a la página correspondiente.

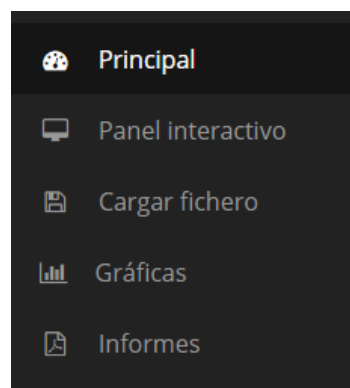


Ilustración 13: Menú lateral

- Acceder a la pantalla principal, monitorización de variables y estado general.
- Acceder al Panel Interactivo, esquema de ingeniería de la planta que permite interactuar con los diferentes elementos, ver su estado y gráficas con los últimos valores obtenidos.

- Acceder al menú para la Carga del fichero Excel. Esta opción permite al usuario cargar los datos correspondientes a las torres de refrigeración desde el fichero de datos (conductividad y purga).
- Acceder a las Gráficas, con los datos históricos relativos a cada una de las secciones de la pantalla principal (climatizadoras, depósito, agua suministrada por la red...).
- Acceder al menú de Informes, para solicitar un CSV con los datos referentes a una o varias secciones monitorizadas.

6.2.3. Pantalla principal de la aplicación

El objetivo de esta sección es ofrecer al usuario una visualización de las variables del demostrador de modo que de un vistazo se puede obtener toda la información del estado de las variables que se están monitorizando.

En esta pantalla podemos ver que aparecen diferentes *widgets* (recuadros) con el contenido relativo a cada una de las zonas o partes que vamos a monitorizar.

6.2.4. Climatizadoras

Este widget contiene toda la información relativa a las climatizadoras. En particular incluye la información de:

- Sensor de temperatura y humedad situado en la fachada exterior de la sala, y de este sensor se obtiene la temperatura, humedad relativa y entalpía exterior. En la Ilustración 14 se muestra ese parte etiquetado por '1'.
- Información relativa a las climatizadoras propiamente dicho se muestra en la zona marcada '2' de la Ilustración 14. A su vez, dentro de esta sección se muestran las siguientes:
 - Nombre de la climatizadora
 - Estado de funcionamiento de la climatizadora (led de estado, verde/ON o gris/OFF)
 - Información de impulsión (etiquetado por '3' en la Ilustración 14):
 - Temperatura de impulsión de aire frío (° C)
 - Porcentaje de apertura de válvula de aire frío (%)
 - Porcentaje de apertura de válvula de calor (%)

- Información de retorno (etiquetado por '4' en la Ilustración 14):
 - Temperatura de retorno de aire (° C)
 - Humedad relativa de retorno de aire (%)

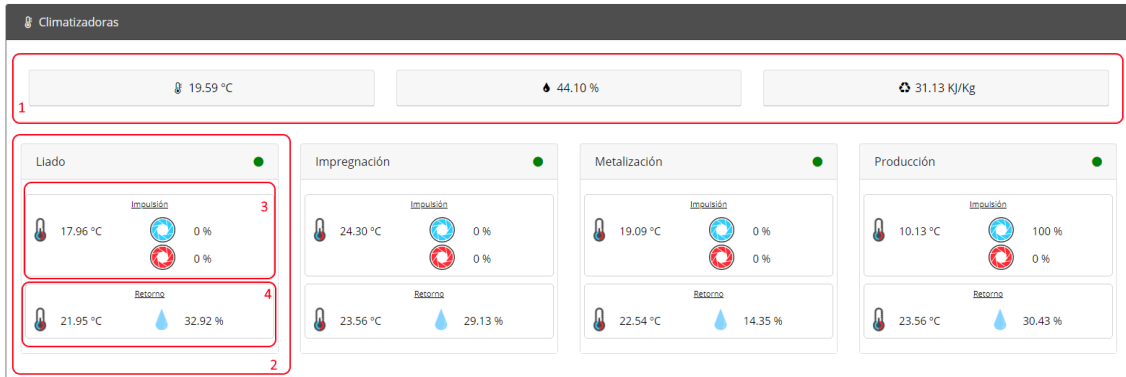


Ilustración 14: Panel de climatizadoras

6.2.5. Impulsión desde climatizadora de producción a depósito

En esta sección se muestra el estado de una de las bombas de impulsión, denominada en este caso “Bomba Producción”, que bombea agua desde la climatizadora de producción hasta el depósito de agua condensada.



Ilustración 15: Panel que muestra el estado de la bomba de impulsión apagada

6.2.6. Depósitos

En este apartado (ver Ilustración 16) aparece toda la información referente al depósito de agua condensada. En este depósito se almacena toda el agua que se genera por condensación en las cuatro climatizadoras de la instalación. Se puede visualizar la temperatura del agua y el porcentaje de llenado del tanque.



Ilustración 16: Panel de información sobre el depósito de agua

También se pueden modificar los valores de consigna de Arranque y Parada. El valor de consigna de Arranque indica que una vez superado el valor se suministrará agua desde el depósito y no desde el agua de red, y al bajar el nivel del depósito por debajo del valor de consigna de Parada, dejará de suministrar agua a las torres de refrigeración y por lo tanto este suministro de agua será mediante el agua de red de nuevo.

6.2.7. Impulsión desde depósito a torres de refrigeración

En este caso se monitorizan dos variables principales (ver Ilustración 17):

- El estado del motor de impulsión de agua desde el depósito de agua condensada hacia las Torres de Refrigeración (incluyendo las averías asociadas a este componente, descritas anteriormente).
- El contador que contabiliza los litros de agua que se han suministrado a las torres desde el depósito de acumulación. La aplicación permite al usuario reiniciar el contador a nivel lógico, para que el valor del contador se reinicie a 0.

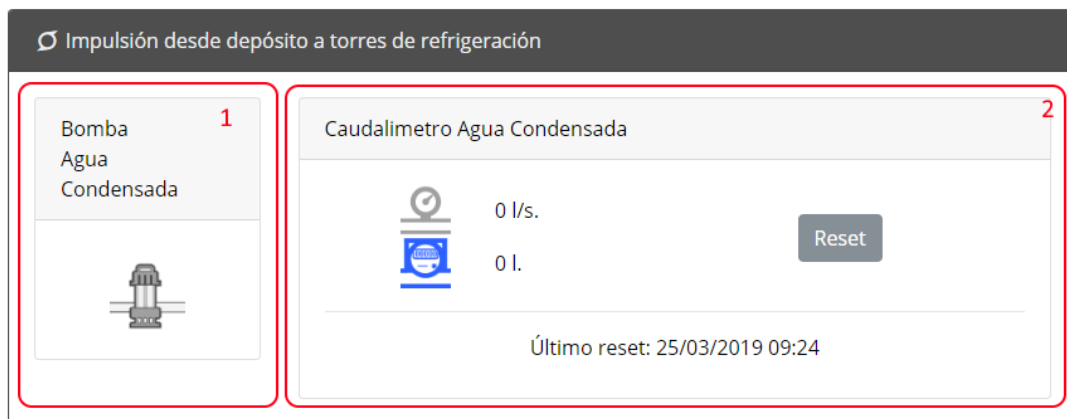


Ilustración 17: Panel de información de agua de condensados, mostrando el estado de la bomba y el caudalímetro

6.2.8. Estado Válvula 3 vías a torres de refrigeración

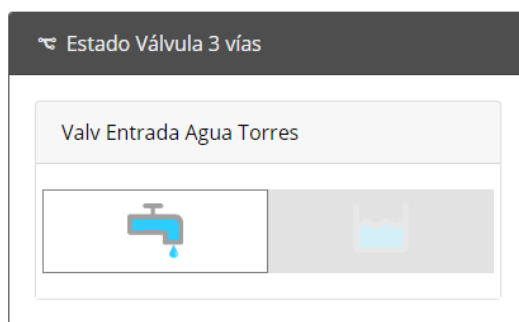


Ilustración 18: Panel de estado de válvula de 3 vías

Esta sección (ver Ilustración 18) muestra el estado de la válvula de 3 vías que proporciona el agua a las torres de refrigeración. La válvula permitirá dar paso de agua a las torres desde agua de red de abastecimiento o bien desde agua del depósito de condensación.

6.2.9. Agua suministrada por la red

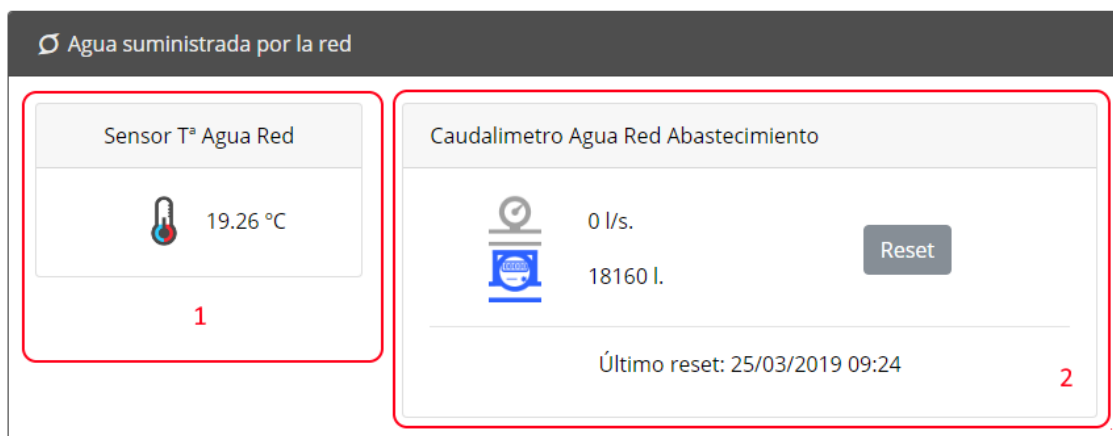


Ilustración 19: Panel de información sobre el agua de la red de abastecimiento

El panel de agua de red (ver Ilustración 19) muestra por un lado la temperatura actual del agua suministrada por la red de abastecimiento

- La temperatura actual del agua suministrada por la red de abastecimiento.
- La cantidad de litros que han demandado las torres de refrigeración desde el último reinicio del contador.

6.2.10. Torres de refrigeración

La información relativa a las Torres de Refrigeración actualmente se carga a través de un fichero de Excel.



Ilustración 20: Panel de torres de refrigeración

En este panel podemos ver (Ilustración 20) los últimos valores de conductividad medidos en cada una de las torres, además de su estado de funcionamiento y el estado de las válvulas de purga asociadas a cada una.

6.2.11. Panel Interactivo

La aplicación ofrece una visualización tipo SCADA (*Supervisory Control And Data Acquisition*), en el que se muestra el diseño industrial de la planta que ha sido dotado de funciones adicionales. De este modo, al pulsar en el “Panel Interactivo” (dentro del panel situado en la parte izquierda) nos aparecerá un plano de la situación de las climatizadoras, depósito, torres de refrigeración, bombas, contadores y sensores de temperatura, es decir, toda la información que se muestra en el panel principal.

En este caso es posible pulsar cada uno de los elementos que componen la parte de refrigeración y aparecerá su estado actual además de una o varias gráficas de las últimas horas, tal y como se muestra en la Ilustración 21.

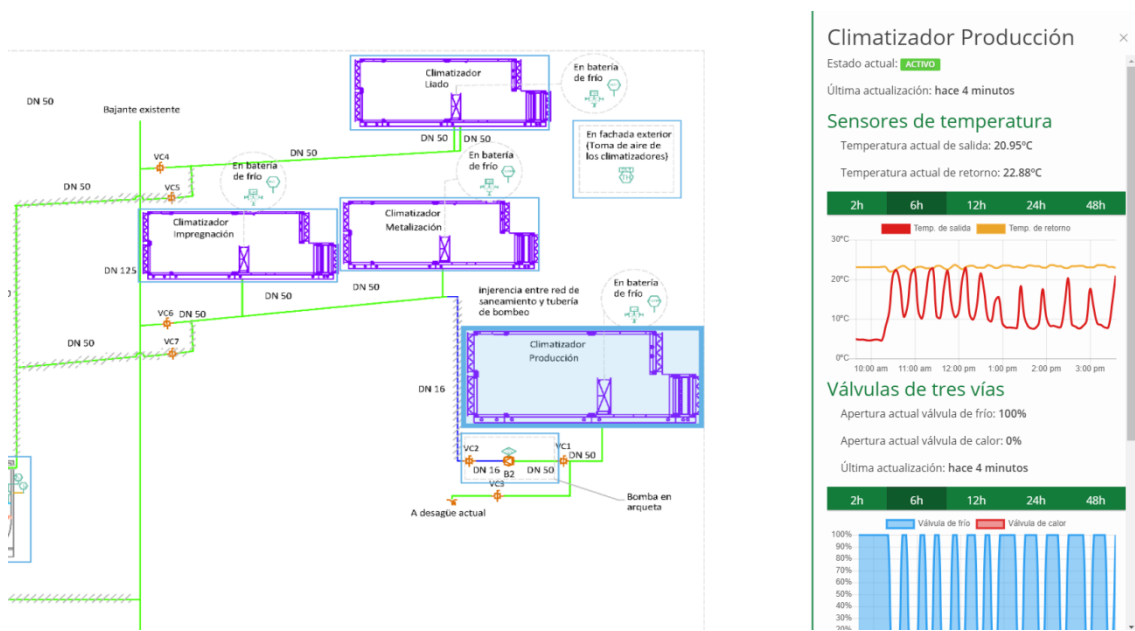


Ilustración 21: Panel interactivo donde se puede ver la información de todo el sistema

6.2.12. Cargar fichero CSV de datos de torres de refrigeración

Este apartado permite realizar la carga del fichero Excel con la información de las torres de refrigeración. En la Ilustración 22 se muestra el control que permite al usuario realizar la carga del fichero.



Ilustración 22: Carga de fichero CSV de torres de refrigeración

Una vez pulsado el botón de ‘Seleccionar archivo’ aparecerá una ventana emergente en la que se debe seleccionar el archivo con los datos a subir, como se muestra en la Ilustración 23.

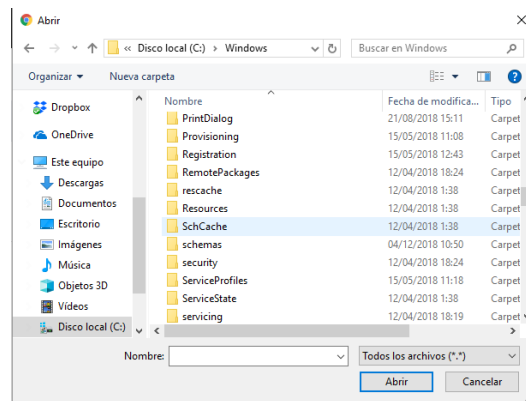


Ilustración 23: Dialogo de selección de fichero CSV

Una vez seleccionado el archivo, el usuario deberá pulsar el botón de ‘Enviar a Base de datos’ para guardar todos los datos en base de datos. Una vez hecho esto y si el fichero tiene el formato correcto, la información de las torres de refrigeración se verá actualizada con aquella recogida en el fichero CSV seleccionado.

6.2.13. Gráficas de datos históricos

Este menú permite realizar diferentes gráficas sobre todos los apartados que se han descrito anteriormente.

Debido a la gran cantidad de datos (de media se obtienen datos cada 5 minutos) el rango de fechas está limitado a un mes y medio.

Para obtener las gráficas se selecciona el rango de fechas y se selecciona alguna de las opciones del menú desplegable, que se muestra en la Ilustración 24.

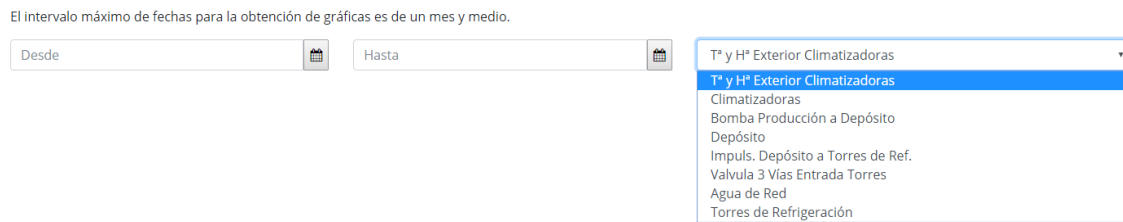





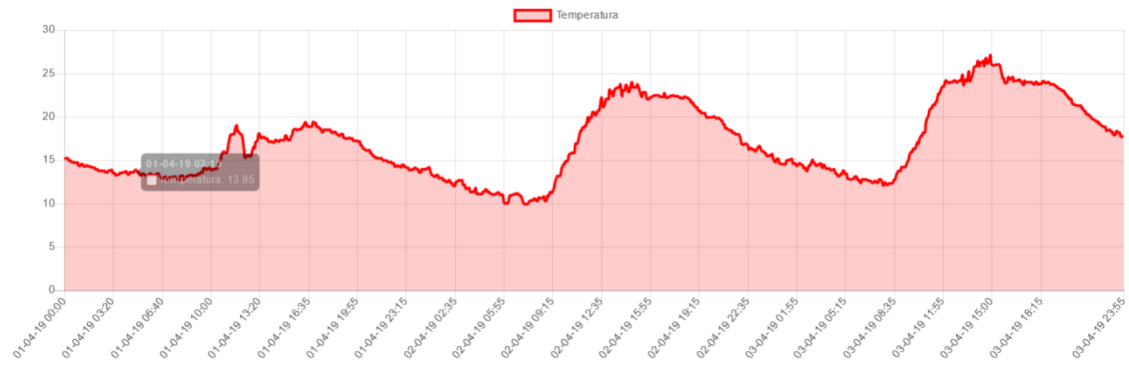
Ilustración 24 Pantalla de configuración de gráficas de datos históricos

Una vez hecho esto se pulsará el botón **Mostrar** y aparecerán las gráficas seleccionadas. En la Ilustración 25 se muestra un ejemplo de las gráficas que muestra la aplicación.

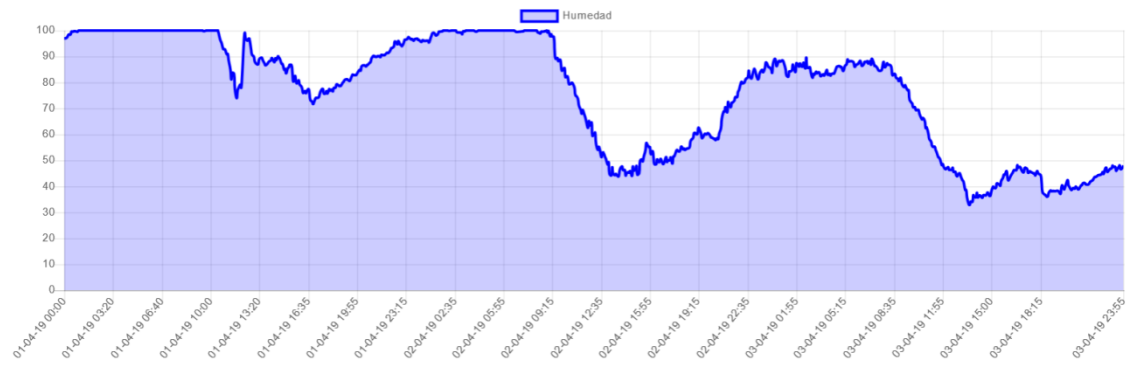
El intervalo máximo de fechas para la obtención de gráficas es de un mes y medio.

2019-04-01  2019-04-03  Tª y Hª Exterior Climatizadoras  Mostrar

Temperatura Exterior Climatizadoras (°C)



Humedad Exterior Climatizadoras (%)



Entalpia (KJ/Kg)



Ilustración 25: Ejemplo de gráficas de datos históricos (temperatura, humedad y entalpía exterior) del 1 al 3 de abril

6.2.14. Generación de informes en formato CSV

El menú de Informes es similar al anterior, con la salvedad que se pueden seleccionar más de una opción en el menú desplegable, por ejemplo, se podrán obtener los datos en un fichero CSV de las climatizadoras, temperatura y humedad exterior y la impulsión del Depósito a las Torres de refrigeración, como se muestra en la Ilustración 26.



Ilustración 26: Interfaz para la generación de informes

Una vez seleccionadas las opciones y el rango de fechas, se pulsa el botón “Obtener Informe” y una vez generado el archivo se descargará automáticamente en nuestro equipo.

6.3. Desarrollo del sistema de autenticación

A fin de cumplir el requisito de seguridad de la aplicación, se requiere implementar un sistema de autenticación que permita restringir el acceso a la aplicación únicamente a usuarios que dispongan de credenciales de acceso. Para este fin, se realizan modificaciones en todos los niveles de la aplicación

6.4. Usuarios en la base de datos

En primer lugar, es necesario tener una forma de almacenar los usuarios que deben tener acceso a la aplicación. Para esto se agrega a la base de datos una nueva tabla donde almacenar la información de estos usuarios. En la ilustración Ilustración 27 se muestra esta tabla, la cual contiene toda la información necesaria a almacenar.


User	
	UserId
	FirstName
	LastName
	Username
	Password
	Token
	PasswordHash
	PasswordSalt
	Email

Ilustración 27: Tabla para el almacenamiento de usuarios de la aplicación

En esta tabla se almacena la información de la persona con acceso (nombre, apellidos, correo electrónico) además de sus credenciales de acceso. Para acceder a la aplicación, el usuario deberá introducir un nombre de usuario además de una contraseña.

A fin de dotar de mayor seguridad a la aplicación, en lugar de almacenar las contraseñas tal cual las introduce el usuario, estas pasan por una función *hash* [15]. Estas funciones permiten convertir una cadena de texto en una serie de bytes de longitud fija. Al aplicar esta función sobre las contraseñas, se consigue un valor distinto al que el usuario debe conocer para acceder a la aplicación, por lo que si este valor se filtrase no se podría utilizar directamente para acceder a la cuenta del usuario. La característica principal de las funciones *hash* es que son, a efectos prácticos, funciones unidireccionales, resultando de extrema dificultad y coste computacional obtener el valor original de una cadena.

A fin de incrementar la seguridad de las contraseñas se implementa el uso de valores SALT, los cuales se añaden a las contraseñas previo a aplicar la función *hash* sobre estas. Esta adición permite evitar ataques de tipo *Rainbow table* [15].

6.5. Obtención de autorización

Para que un usuario tenga acceso a requerir información del servidor, este debe adjuntar en cada petición una clave JWT. Esta clave identifica y autentica al usuario que realiza la petición, de forma que sin este el servidor se deniega la petición.

Para que un usuario obtenga una clave JWT, se ha añadido una nueva entrada en la API del servidor, la cual requiere que se introduzcan el nombre y la contraseña de un usuario. Si estas credenciales son correctas, el servidor genera una clave JWT para el usuario.

6.6. Inicio de sesión en cliente

El último paso para la implementación del sistema de autenticación es la inclusión de un formulario de inicio de sesión. En la Ilustración 28 se muestra el formulario de inicio de sesión, en el cual se requieren las credenciales para el acceso a la aplicación.

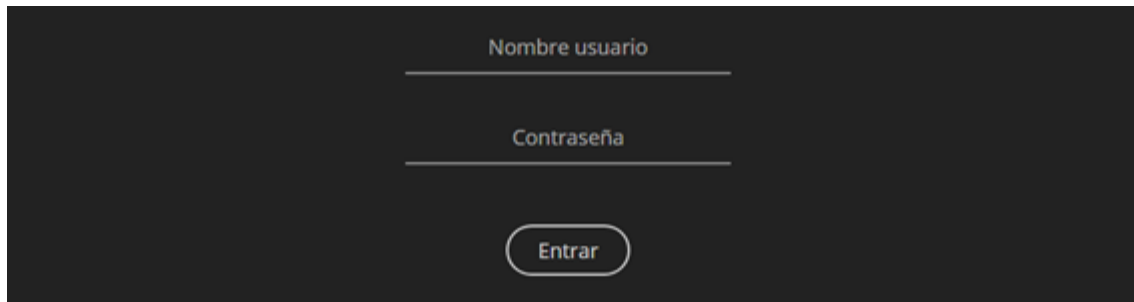


Ilustración 28: Pantalla de acceso a la aplicación web

En este punto, el usuario deberá introducir su nombre de usuario y la contraseña proporcionados para acceder a la plataforma, y hacer clic en el botón “Entrar”. En caso de introducir los datos de forma incorrecta se informará al usuario que no ha sido posible acceder a la aplicación.

Si se han introducido unas credenciales válidas, el navegador será redirigido a la pantalla principal de la aplicación.

7

Pruebas y validación

En este capítulo, como fase final del método de desarrollo en cascada, se exponen una serie de pruebas de integración realizadas sobre la aplicación para comprobar que las soluciones implementadas cumplen con los requisitos especificados.

7.1. Prueba de acceso a la aplicación

En la siguiente sección se exponen una serie de pruebas realizadas sobre la pantalla de inicio de sesión, en el que se pone a prueba la seguridad de la aplicación y del sistema de control de acceso.

Se ha comprobado que, utilizando credenciales correctas (almacenadas en la base de datos), la aplicación permite al usuario acceder al resto de funcionalidades. También se ha comprobado que, utilizando credenciales incorrectas, la aplicación deniega el acceso al usuario, además de aparecer un mensaje de error indicando que el usuario y/o la contraseña son incorrectos.

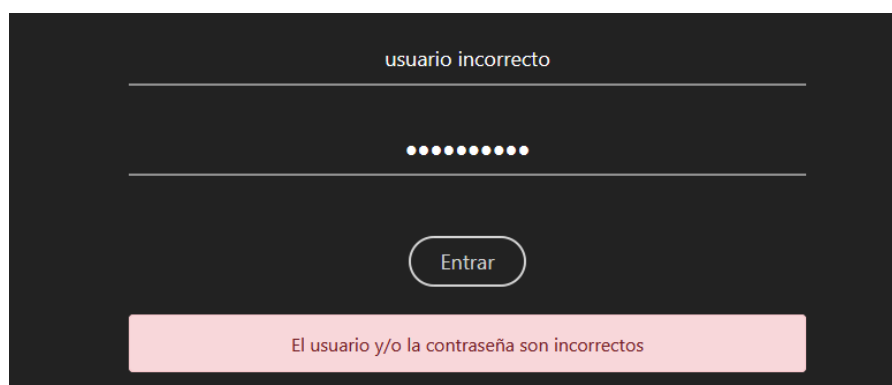


Ilustración 29: Intento de acceso a la aplicación con credenciales incorrectas

7.2. Prueba de visualización de históricos

En la siguiente sección se exponen una serie de pruebas realizadas sobre la pantalla de visualización de históricos, en el que se pone a prueba revisión de valores medidos por los sensores a lo largo del tiempo.

En la Ilustración 30 se muestra un histórico de la humedad registrada por los sensores en las cuatro climatizadoras utilizadas.



Ilustración 30: Historial de humedad relativa de retorno de las climatizadoras

A su vez, en la Ilustración 31 se muestra el histórico del nivel del agua del depósito a lo largo del tiempo. En el gráfico se puede apreciar como los valores de consigna están especificados al 20% y el 70%, e forma que cuando el nivel del agua llega al 70% se utiliza el agua del depósito hasta que éste llega al 20%.



Ilustración 31: Historial del nivel del agua en el depósito

7.3. Prueba de modificación de los valores de consigna

Se ha probado a cambiar los valores de consigna de las torres de refrigeración, comprobándose que al introducir un nuevo valor y pulsar sobre el botón ‘Aplicar’ estos persisten en la base de datos en la aplicación.

7.4. Pruebas de generación de informes CSV

En la siguiente sección se exponen una serie de pruebas realizadas sobre la pantalla de generación de informes CSV, en el que se pone a prueba el fichero resultante de la generación.

Para la ejecución de la prueba, se ha accedido a la sección “Informes” desde el panel lateral. Una vez seleccionadas las opciones y el rango de fechas, se pulsa el botón “Obtener Informe”, momento en el que se iniciará la descarga del archivo automáticamente en nuestro equipo. En la Ilustración 32 se muestra el contenido de este fichero, donde se muestran los registros de las distintas mediciones en el rango de fechas seleccionado.

```
Fecha;Temperatura Exterior Clim.;Humedad Exterior Clim.;Entalpia Clim.;Humedad Abs.;Temperatura Impulsion (Liado) oC;Porc. Apertura Valvula Frio(Liado) %;Porc.
14/09/2020 - 0:00;23,22;71,77;56,20385;0,01290963;17,03;100;83;20,34;54,285;12,02;100;0;21,35;50,97847;19,93;0;53;20,85;59,66;23,22;0;0;22,88;60,02
14/09/2020 - 0:05;23,22;71,185;55,93613;0,0128044;17,03;100;83;20,34;54,285;12,02;100;0;21,35;50,97847;19,93;0;53;20,85;59,66;23,22;0;0;22,88;58,985
14/09/2020 - 0:10;23,39;71,085;56,41098;0,01292222;17,28;100;83;20,34;54,58;12,38;100;0;21,27;51,36987;20,85;100;62;20,85;61,82;18;0;0;22,88;60,615
14/09/2020 - 0:15;23,39;71,085;56,41098;0,01292222;17,28;100;83;20,34;54,58;12,38;100;0;21,27;51,36987;22,46;100;57;20,85;62,43;22,30;0;0;22,88;61,655
14/09/2020 - 0:20;23,3,70,77;55,99006;0,01279321;17,28;100;83;20,34;54,48;12,02;100;0;21,18;51,22309;19,26;0;57;21,01;61,095;22,88;0;0;22,46;61,14
14/09/2020 - 0:25;23,3,70,77;55,99006;0,01279321;17,28;100;83;20,34;54,48;12,02;100;0;21,18;51,22309;19,59;0;62;21,01;60,48;22,88;0;0;22,46;60,63
14/09/2020 - 0:30;23,22;71,87;56,24962;0,01292762;16,94;100;86;20,27;54,385;11,76;100;0;21,1;51,07632;19,17;0;0;20,85;59,27;23,22;0;0;22,88;60,115
14/09/2020 - 0:35;23,22;71,87;56,24962;0,01292762;17,45;100;86;20,27;54,385;11,76;100;0;21,1;51,07632;20,27;0;62;20,85;60,79;16,05;52,49996;0;22,88;58,575
14/09/2020 - 0:40;23,3,72,57;56,82026;0,01311951;17,62;100;86;20,27;54,675;12,29;100;0;21,1;51,41879;21,1;0;53;20,76;60,28;18,6;0;0;22,88;60,21
14/09/2020 - 0:45;23,3,72,57;56,82026;0,01311951;16,94;100;83;20,27;54,675;12,29;100;0;21,1;51,41879;18,83;100;0;20,76;60,275;22,31;0;0;22,37;61,76
14/09/2020 - 0:50;23,22;71,87;56,24962;0,01292762;17,11;100;86;20,27;54,48;12,02;100;0;21,01;51,56556;20,76;100;62;20,85;61,4;22,88;0;0;22,46;61,76
14/09/2020 - 0:55;23,22;72,38;56,48301;0,01301935;16,61;100;86;20,27;54,48;12,02;100;0;21,01;51,56556;20,93;100;53;20,85;61,205;22,88;0;0;22,46;60,73
14/09/2020 - 1:00;23,39;72,965;57,28059;0,01326397;16,86;100;86;20,27;54,385;12,02;100;0;21,01;51,56556;20,59;100;53;20,93;60,685;23,22;0;0;22,88;60,22
14/09/2020 - 1:05;23,39;72,965;57,28059;0,01326397;16,86;100;90;20,27;54,385;12,02;100;0;21,01;51,56556;19,26;100;0;20,93;60,79;16,99;52,49996;0;22,88;59,19
14/09/2020 - 1:10;23,73;73,575;58,63907;0,01366004;17,62;100;86;20,27;54,58;12,29;100;0;21,01;51,90802;21,18;100;74;20,93;63,355;18,34;0;0;22,97;60,215
14/09/2020 - 1:15;23,73;74,165;58,91787;0,01376958;17,62;100;86;20,27;54,58;12,29;100;0;21,01;51,90802;20,51;0;66;20,93;61,3;21,54;0;0;22,46;62,785
14/09/2020 - 1:20;23,3,74,97;57,92183;0,01355246;17,03;100;86;20,27;54,675;11,94;100;0;21,01;52,05479;19,42;0;53;20,85;59,765;22,63;0;0;22,37;62,785
14/09/2020 - 1:25;23,3,74,26;57,59527;0,01342411;17,03;100;86;20,27;54,675;11,94;100;0;21,01;52,05479;19,26;0;53;20,85;59,765;22,63;0;0;22,37;62,28
14/09/2020 - 1:30;23,22;74,455;57,43261;0,01339259;17,11;100;86;20,27;54,58;11,85;100;0;20,93;51,78572;19,51;0;53;20,93;59,665;23,05;0;0;22,8;61,245
14/09/2020 - 1:35;22,71;74,97;56,07036;0,01306369;17,11;100;86;20,27;54,58;11,85;100;0;20,93;51,78572;19,59;0;53;20,93;60,175;23,05;0;0;22,8;60,73
14/09/2020 - 1:40;22,63;75,46;56,03861;0,01308361;17,45;100;90;20,27;54,48;11,85;100;0;20,85;51,83464;20,34;0;53;20,93;60,18;17,16;52,49996;0;23,05;59,715
14/09/2020 - 1:45;22,63;75,46;56,03861;0,01308361;17,45;100;86;20,27;55,065;12,38;100;0;20,85;52,34834;20,34;0;53;20,93;60,69;18,34;0;0;23,05;61,235
14/09/2020 - 1:50;22,63;75,36;55,99452;0,01306628;17,11;100;86;20,27;54,87;12,02;100;0;20,85;52,34834;18,07;0;62;21,01;60,585;21,97;0;0;22,54;62,275
14/09/2020 - 1:55;22,63;76,46;56,47955;0,013257;17,11;100;90;20,27;54,87;12,02;100;0;20,85;52,34834;19,76;0;66;21,01;61,22;8;0;0;22,54;62,275
14/09/2020 - 2:00;22,37;76,46;55,66874;0,0130435;17,37;100;90;20,19;54,775;11,94;100;0;20,85;52,32388;20,1;0;53;20,76;59,87;22,88;0;0;22,63;61,76
14/09/2020 - 2:05;22,88;75,85;56,99368;0,01335783;16,77;100;90;20,19;54,775;11,94;100;0;20,85;52,32388;21,18;100;53;20,76;60,59;23,39;0;0;22,63;61,255
14/09/2020 - 2:10;23,05;75,555;57,397;0,01344749;17,28;100;90;20,19;54,48;11,94;100;0;20,76;52,12818;21,01;100;53;20,93;60,69;23,47;0;0;22,97;61,255
14/09/2020 - 2:15;23,05;76,46;;17,28;100;90;20,19;54,48;11,94;100;0;20,76;52,12818;21,01;100;53;20,93;60,82;17,24;52,49996;0;22,97;59,705
14/09/2020 - 2:20;22,63;75,555;56,0805;0,01310009;17,71;100;90;20,19;54,87;12,29;100;0;20,76;52,56849;21,01;100;53;20,93;60,795;18,43;0;0;22,88;61,45
14/09/2020 - 2:25;22,63;74,97;55,82256;0,01299866;17,2,100;90;20,19;54,87;12,29;100;0;20,76;52,56849;20,93;100;53;20,93;60,795;22,39;0;0;22,88;62,48
14/09/2020 - 2:30;21,95;78,95;55,42778;0,01311894;16,94;100;90;20,19;54,87;12,02;100;0;20,76;52,66634;20,51;100;53;20,93;60,69;22,88;0;0;22,46;62,48
14/09/2020 - 2:35;21,95;78,95;55,42778;0,01311894;16,94;100;90;20,19;54,87;12,02;100;0;20,76;52,66634;20,42;100;53;20,93;60,69;22,88;0;0;22,46;61,975
14/09/2020 - 2:40;21,44;79,73;54,17139;0,01283129;16,94;100;95;20,1;54,58;11,85;100;0;20,68;52,91096;20,1;0;57;20,85;60,285;18,26;0;0;22,97;60,94
14/09/2020 - 3:00;21,44;79,73;54,17139;0,01283129;16,94;100;95;20,1;54,58;11,85;100;0;20,68;52,91096;20,1;0;57;20,85;60,285;18,26;0;0;22,97;60,94
14/09/2020 - 3:05;21,44;79,73;54,17139;0,01283129;17,45;100;95;20,1;54,58;11,85;100;0;20,68;52,91096;20,1;0;57;20,85;60,285;18,26;0;0;22,97;60,94
14/09/2020 - 3:10;21,35;79,73;53,89478;0,0127589;17,11;100;95;20,1;55,065;12,02;100;0;20,59;52,98434;19,26;0;57;20,85;59,77;22,14;0;0;22,54;61,455
14/09/2020 - 3:15;21,86;78,95;55,14751;0,01304516;17,11;100;95;20,1;55,065;12,02;100;0;20,59;52,98434;19,26;0;57;20,85;60,28;22,71;0;0;22,54;61,455
14/09/2020 - 3:20;21,52;79,925;54,49822;0,01292749;17,37;100;95;20,1;54,58;11,67;100;0;20,59;52,54403;20,68;0;57;20,85;60,295;22,63;0;0;22,63;61,455
14/09/2020 - 3:25;21,52;80,535;54,74893;0,01302615;17,37;100;95;20,1;54,58;11,67;100;0;20,59;52,54403;20,68;100;57;20,85;60,79;22,63;0;0;22,63;60,945
14/09/2020 - 3:30;21,35;80,145;54,06351;0,01282531;17,37;100;95;20,1;54,48;11,58;100;0;20,51;52,39726;21,69;100;57;20,85;61,31;22,97;0;0;22,8;60,43
14/09/2020 - 3:35;21,35;80,83;54,34202;0,01293493;17,37;100;99;20,1;54,48;11,58;100;0;20,51;52,39726;22,29;100;57;20,85;61,31;22,97;0;0;22,8;60,43
14/09/2020 - 3:40;21,95;80,145;55,93251;0,01331751;17,45;100;95;20,02;54,48;11,76;100;0;20,51;52,59296;21,27;100;57;20,85;61,82;23,39;0;0;22,97;60,43
```

Ilustración 32: Informe exportado en formato CSV

8

Conclusiones y líneas futuras

8.1. Menús flexibles

Actualmente, la distribución de los distintos elementos del sistema a monitorizar está embebida en el código del cliente. En otros términos, la modificación de la distribución de los elementos del sistema es parte del código y por tanto para modificarla hay que realizar cambios en éste. Esto resulta un inconveniente si se quiere poder distribuir la aplicación a varios clientes, ya que para cada uno debería reprogramarse para adaptarse al esquema del sistema monitorizado.

La aplicación por tanto podría expandirse añadiendo la posibilidad de crear, modificar y eliminar elementos de planta (depósitos de agua, caudalímetros, etc.) desde una interfaz gráfica, siendo esta accesible a los usuarios de la aplicación sin tener que modificar el código. Para esto también deberían añadirse utilidades para distribuir y reorganizar los elementos del panel principal, dado que estos dependen de los elementos disponibles.

8.2. WebSocket

Actualmente, una vez el usuario accede a la aplicación se realizan las consultas a la base de datos sobre el estado actual del sistema, siendo este el que se muestra al usuario. Una vez realizadas estas consultas, la aplicación muestra esta información hasta que se produce una recarga de la aplicación. Dado que se está trabajando con información cambiante a tiempo real, sería conveniente que la aplicación actualizase los valores de acuerdo con estos cambios. Una forma de conseguir este resultado sería actualizar la

página de forma programada pasado determinado tiempo, de forma que se obtenga la información más reciente. El problema de esta aproximación es la posibilidad de sobrecarga del servidor ya que, dada la naturaleza del protocolo HTTP, se realiza una conexión TCP por cada petición que realiza el navegador al servidor. Esto produce una sobrecarga que puede saturar al servidor de peticiones.

WebSocket [16] es un protocolo de comunicaciones bidireccional que, al igual que HTTP, se basa en el protocolo TCP. A diferencia de HTTP, WebSocket utiliza una sola conexión TCP y la reutiliza hasta que una de las dos partes la termina. Esto permite aprovechar la conexión ya existente, eliminando la sobrecarga de generar varias de estas conexiones. Otra de las ventajas de WebSocket es que es un protocolo de comunicaciones bidireccional, haciendo que ambas partes de la comunicación puedan enviar información de forma independiente.

Esta tecnología permitiría al servidor enviar la información más reciente sobre el estado del sistema a todos los clientes conectados, sin necesidad de que estos requieran esta información en intervalos de tiempo que pueden no corresponderse con la frecuencia de actualización de los sensores del sistema.

Referencias

- [1] E. Hernández Goríbar, Fundamentos de aire acondicionado y refrigeración, Limusa, 1997.
- [2] S. M. B. Novillo, J. C. C. Valdivieso, L. S. O. Cantos y E. Á. J. Domínguez, «Análisis del funcionamiento de sistemas de refrigeración por compresión y absorción,» Marzo 2018. [En línea]. Available: <https://www.eumed.net/rev/oel/2018/03/sistemas-refrigeracion.html>.
- [3] M. J. Moran, Fundamentos de la termodinámica técnica, Reverté, 2018.
- [4] Google, «Angular,» [En línea]. Available: <https://angular.io/>.
- [5] Microsoft, [En línea]. Available: <https://www.typescriptlang.org/>.
- [6] Equipo de Bootstrap, «La librería Bootstrap,» [En línea]. Available: <https://getbootstrap.com/>.
- [7] D. Roth, R. Anderson y S. Luttin, «Introduction to ASP.NET Core,» [En línea]. Available: <https://docs.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-2.2>.
- [8] Linux Foundation, «OpenAPI Initiative,» [En línea]. Available: <https://www.openapis.org/>.
- [9] L. Richardson y S. Ruby, RESTful Web Services, O'Reilly Media, Inc, 2008.

- [10] Microsoft, «Entity Framework Core,» [En línea]. Available: <https://docs.microsoft.com/en-us/ef/core/>.
- [11] M.Jones, Microsoft, J.Bradley, P. Identity y N.Sakimura, «JSON Web Token (JWT),» Mayo 2015. [En línea]. Available: <https://tools.ietf.org/html/rfc7519>.
- [12] ECMA International, «The JSON Data Interchange Syntax,» Diciembre 2017. [En línea]. Available: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>.
- [13] Microsoft, «Introducing SQL Server 2019,» [En línea]. Available: <https://www.microsoft.com/en-us/sql-server/sql-server-2019>.
- [14] I. Sommerville, Ingeniería del Software, Pearson Education, 2005.
- [15] S. Boonkrong y C. Somboonpattanakit, «Dynamic Salt Generation and Placement for Secure Password Storing,» *IAENG International Journal of Computer Science*, 2016.
- [16] I. Fette, A.Melnikov, G. I. y I. L. , «The WebSocket Protocol,» Diciembre 2011. [En línea]. Available: <https://tools.ietf.org/html/rfc6455>.
- [17] Google, «Angular Press Kit,» [En línea]. Available: <https://angular.io/presskit>.

Apéndice A

Manual de Instalación

Requerimientos:

La instalación de la aplicación se compone de dos fases, siendo estas compilación del proyecto y despliegue del mismo.

Requerimientos de compilación

Para realizar la compilación de los proyectos es necesario tener instaladas en el equipo las siguientes herramientas:

- SDK de .NET Core 2.2
- NPM (Node Package Manager)
- Node.js

Compilación del cliente

Para compilar el cliente de Angular, es necesario dirigirse a la carpeta “src/Frontend” y ejecutar el comando “`npm run build`” desde un terminal. Tras ejecutar este comando se habrá creado una nueva carpeta en el directorio con el nombre “dist”, donde se encontrará el código de cliente listo para su despliegue.

Compilación del servidor

Para compilar la aplicación de servidor de ASP.NET Core, es necesario dirigirse a la carpeta “src/Backend” y ejecutar el comando “`dotnet publish`” desde un terminal. Tras ejecutar este comando se habrá creado una carpeta en el directorio con el nombre “build” donde se encontrará el código de servidor listo para su despliegue.

Requerimientos de despliegue

Para desplegar la aplicación y que quede disponible en internet es necesario disponer de un equipo con una dirección IP estática accesible desde internet.

También es necesario disponer de un servidor web compatible con aplicaciones de ASP.NET Core, como pueden ser *Internet Information Services (IIS)*, Nginx o Apache.

Independientemente del servidor web que se utilice, es necesario configurarlo para que despliegue dos nuevas aplicaciones, haciendo que los valores a servir apunten a los compilados generados en las secciones anteriores.

En el caso de la aplicación de cliente, Angular requiere que se modifique el sistema de enrutamiento del servidor, de forma que todas las peticiones que se realicen a una sección de la aplicación sean redireccionadas al fichero “`index.html`”, dado que Angular incluye su propio sistema para el manejo de rutas.