



FACULTAD DE INFORMÁTICA
UNIVERSIDAD POLITÉCNICA DE MADRID

UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA

TRABAJO FIN DE CARRERA

**PLATAFORMA DE OBJETOS WEB PARA EL CONTROL REMOTO
DEL OBSERVATORIO ASTRONÓMICO FRANCISCO SÁNCHEZ**

AUTOR: JAVIER MARTÍNEZ VILLACAMPA
TUTOR: FRANCISCO JAVIER ROSALES GARCÍA
FECHA DE PRESENTACIÓN: 22 DE SEPTIEMBRE DE 2017

Índice

ÍNDICE DE FIGURAS	III
RESUMEN	IV
1. INTRODUCCIÓN Y OBJETIVOS	1
1.1. Introducción	1
1.2. El observatorio Francisco Sánchez	1
1.3. Objetivos	4
1.4. Estructura del libro	5
2. EL SISTEMA DE CONTROL	6
2.1. Introducción	6
2.2. Plataforma de objetos web	7
2.3. Pasarela de acceso	9
2.3.1. Gestión de roles	9
2.3.2. Gestión de usuarios	11
2.3.3. Gestión de reservas	15
2.3.4. Control de acceso y autorización	18
2.3.5. Funciones de administración	19
2.4. Cúpula	21
2.5. Montura del telescopio	23
2.6. Cámara	26
2.7. Estación Meteorológica	28
3. ESPECIFICACIÓN DE OBJETOS WEB	30
3.1. Introducción	30
3.2. Propuestas de definición de objetos web del W3C	31
3.2.1. Propuesta basada en el Modelo de Objeto Web	32
3.2.2. Propuesta basada en la Descripción de Objeto Web	36
3.3. Definición de objetos web para el sistema de control del observatorio	38
3.3.1. Patrones de interacción	39
3.3.2. Notación de representación del objeto web	41
4. IMPLEMENTACIÓN	43
4.1. Introducción	43
4.2. Arquitectura del sistema	44
4.3. Base tecnológica	47
4.4. Pruebas	51
4.5. Documentación	52
5. CONCLUSIONES	53
5.1. Resultados	53
5.2. Futuras líneas de trabajo	54

BIBLIOGRAFÍA	56
ANEXOS	59
A. Definición de los Objetos Web	59
A.1. Pasarela de acceso	59
A.2. Cúpula	75
A.3. Montura	78
A.4. Cámara	81
A.5. Estación meteorológica	84

Índice de figuras

1.1. Interior del observatorio Francisco Sánchez en julio de 2007	2
1.2. La Antigua y la nueva arquitectura de automatización del observatorio	3
3.1. Patrones de integración de objetos web: directa, a través de una pasarela de acceso, y a través de la nube	32
3.2. Objeto web desplegado en una pasarela de acceso acorde a la nueva propuesta de diseño basada en la Descripción de Objeto Web	36
4.1. Diseño de alto nivel del sistema de control del observatorio	43
4.2. Arquitectura del sistema de control del observatorio	47
4.3. POSTMAN ejecutando interacciones con el sistema del observatorio .	51
4.4. Cobertura de código durante un ciclo de pruebas con POSTMAN . .	52

RESUMEN

El observatorio astronómico Francisco Sánchez está compuesto por diversos elementos controlables a través de varios dispositivos electrónicos. Dichos dispositivos se encuentran conectados en red e implementan cada uno un protocolo adhoc de comunicación sobre HTTP. Cada protocolo permite controlar de forma remota su dispositivo correspondiente, de forma que dichos dispositivos podrían considerarse como objetos del Internet de las cosas.

Este proyecto tiene como objetivo desarrollar una pasarela de acceso a dichos dispositivos que permita el control remoto de los mismos más allá de la propia red local. Integrando además funcionalidades de gestión de usuarios, de autenticación y autorización y de gestión de reservas que aseguren la toma de control del observatorio por parte de un único usuario por franja de tiempo.

La pasarela será implementada como una plataforma de objetos web que, incluyendo la funcionalidad extra descrita, permita desplegar fácilmente sobre la misma la representación virtual de objetos del Internet de las cosas. Y cuyo acceso y control esté supervisado por la funcionalidad extra de la pasarela. De esta forma se añadirán a dicha plataforma los distintos objetos web que representen a cada uno de los controladores electrónicos del observatorio. Y así se permita, remotamente, manejar y conocer el estado de cada uno de ellos.

Se propondrá a su vez un esquema de definición de los objetos web, de forma que sea posible conocer la forma de interactuar con ellos mediante un archivo de definición que podrán aprovechar procesos automatizados de búsqueda para extraer información de la plataforma sin conocer previamente su protocolo de comunicación.

1. INTRODUCCIÓN Y OBJETIVOS

1.1. Introducción

A principios del siglo XXI el observatorio astronómico Francisco Sánchez nació como un laboratorio accesible a través de la web. Concretamente a través de la plataforma Cíclope Astro desarrollada para tal fin por el grupo Cíclope. Dicha plataforma tenía la principal característica de estar desarrollada como software libre y ser desplegada ofreciendo un acceso totalmente gratuito.

Tras más de diez años de existencia y un parón funcional más largo del deseado, el observatorio es objeto de una nueva puesta en marcha y un nuevo diseño e implementación de sus mecanismos de automatización.

El proyecto objeto de este libro se enmarca dentro de la renovación tecnológica del observatorio. En concreto del diseño e implementación de la nueva plataforma de acceso al mismo. La antigua plataforma web, Cíclope Astro, estaba orientada al usuario final, integrando el control directo de los dispositivos junto a las funciones de control de los dispositivos que exponía a través de una capa de interfaz web de usuario. La nueva plataforma, sin embargo, es concebida como una plataforma web orientada a aplicaciones clientes, permitiendo el acceso a la funcionalidad del observatorio a través de una interfaz de programación RESTful basado en el protocolo HTTP. Y deja la implementación de la capa de interfaz de usuario a las aplicaciones cliente.

1.2. El observatorio Francisco Sánchez

El observatorio astronómico Francisco Sánchez está situado en la Escuela Técnica Superior de Ingenieros Informáticos de la Universidad Politécnica de Madrid. Aunque fue inaugurado a principios del año 2008, la idea de su construcción se plantea en el año 2002 por el grupo Cíclope durante su participación en el proyecto Europeo COLDEX (IST-2001-32327)[COL05], con un carácter abierto y de acceso gratuito. Y aunque finalmente su construcción no fue abordada en el marco de dicho proyecto, sí que se realizó la compra de un primer telescopio en torno al cual el grupo Cíclope comenzó a desarrollar diversos proyectos.

El proyecto central, Cíclope Astro, tenía el objetivo de construir un sistema de software libre que permitiera la teleoperación a través de la web del telescopio. Y a su vez de los diferentes dispositivos electrónicos que lo acompañarían para facilitar su teleoperación, inicialmente solo cámaras.

No es hasta el año 2006 cuando por fin se consigue iniciar la construcción del observatorio astronómico. Dentro del campus de Montegancedo de la Universidad Politécnica de Madrid se construye a lo largo de dos años lo que sería la piedra angular para el proyecto de Cíclope Astro: el observatorio de 3,5 metros de diámetro con un pedestal central, instalación eléctrica y una cúpula mecanizada.

El diseño original de la automatización del observatorio comprendía la utilización de varios mecanismos electrónicos que se conectaban todos directamente a un mismo ordenador central localizado en el propio observatorio. Estas conexiones se realizaban a través de los diversos puertos de la placa base del PC: puertos USB, puerto paralelo y puerto serie.

Y en el ordenador central se ejecutaba un sistema operativo GNU/Linux sobre el que se desplegaban los controladores de cada uno de los dispositivos conectados. El ordenador a su vez estaba conectado a la red local del laboratorio del grupo Cíclope, y los controladores exponían diferentes servicios a través de diferentes puertos para controlar remotamente el observatorio. Dichos servicios eran consumidos por la plataforma Cíclope Astro que permitía así la teleoperación vía web del observatorio proporcionando a su vez mecanismos de control de acceso y reservas.

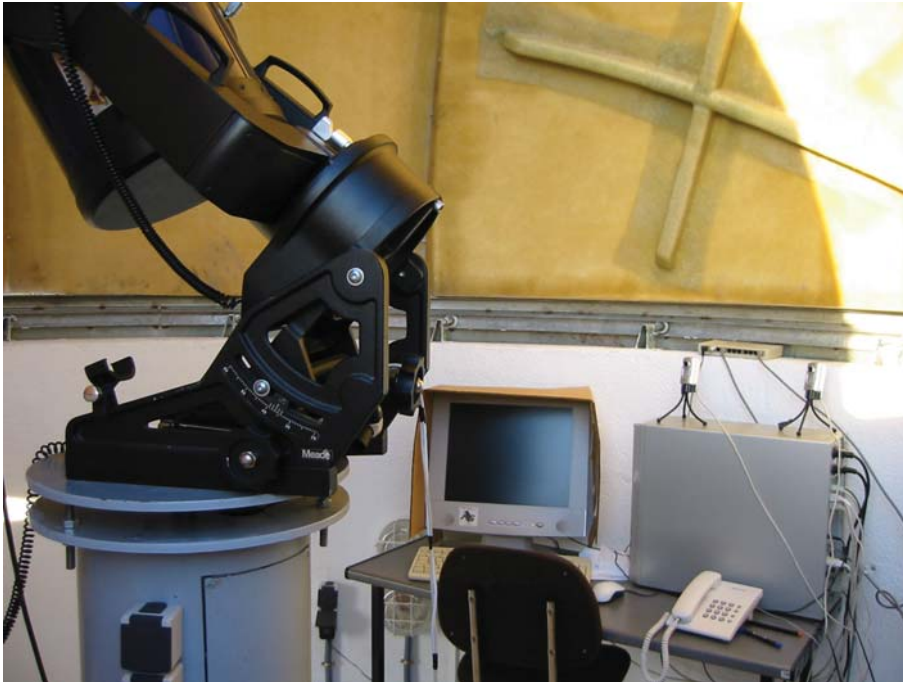


Figura 1.1: Interior del observatorio Francisco Sánchez en julio de 2007

Posteriormente la instalación física del observatorio y la pila de software de teleoperación fueron sufriendo progresivas modificaciones que permitieron la integración del observatorio dentro del proyecto GLORIA[GLO17], un proyecto de la Unión Europea cuyo objetivo principal era crear, con fines científicos, la primera red mundial de telescopios robóticos de libre acceso. Dicho proyecto sirve de actividad principal al grupo Cíclope durante los años 2011 y 2014. Lo que a su vez provoca que el observatorio pierda el foco de atención, ya que los trabajos principales en esos años es el diseño e implementación de la pila de software que permita operar de forma telemática la red de telescopios. Y tras la finalización del proyecto GLORIA se produce el desgraciado fallecimiento del director del grupo, Francisco Sánchez Moreno,

que forzará la postergación sine die de cualquier trabajo en el observatorio durante los meses siguientes.

En el año 2016, y como homenaje a Francisco Sánchez Moreno, se inicia la nueva puesta en marcha del observatorio con una nueva orientación en la arquitectura de automatización. En el desarrollo de los nuevos mecanismos se encuentran colaborando alumnos de varios centros universitarios de la Universidad Politécnica de Madrid. Los principales elementos foco del nuevo diseño son:

- La cúpula con sus motores de giro y apertura
- El controlador de la montura del telescopio
- Las distintas cámaras utilizadas para la captación de imágenes astronómicas y monitorización visual del entorno físico del observatorio, tanto interior como exterior
- La estación meteorológica

Cada uno de estos elementos contará con su propio controlador conectado a la red local que actuará como un nodo independiente, cuya funcionalidad será expuesta a través de un protocolo sobre HTTP. Un nodo podrá emitir eventos relacionados con su actividad y estado del dispositivo que controle. Y el resto de nodos podrán suscribirse a dichos eventos para ser alertados de cuando se producen y actuar en consecuencia. Los controladores mantendrán su implementación lo más sencilla posible y ninguno dispondrá de mecanismos avanzados de control de acceso o de seguridad puesto que estarán orientados a ser utilizados exclusivamente desde la red local en la que se encuentren desplegados.



Fuente [Leó09]

Figura 1.2: La Antigua y la nueva arquitectura de automatización del observatorio

De cara a que los servicios de estos controladores estén disponibles más allá de la red local, se desarrollará una pasarela de acceso a los mismos que exponga de forma indirecta la funcionalidad del observatorio. Esta pasarela implementará una interfaz de programación sobre el protocolo HTTP como capa de abstracción de los controladores desplegados en la red local del observatorio. E integrará también los mecanismos de control de acceso y seguridad necesarios para garantizar la integridad y buen funcionamiento del mismo. La interfaz que exponga la pasarela deberá además estar implementada de forma que motores de búsqueda ajenos al proyecto puedan extraer información pública a través de la misma sin necesidad de conocer previamente cómo funciona.

Además la pasarela de acceso deberá ofrecer la suficiente flexibilidad para ser modificada de cara a ampliar las funciones a exponer a través de la misma en caso de integrar nuevos dispositivos en la red local del observatorio.

1.3. Objetivos

La misión principal de este proyecto es definir e implementar la pasarela de acceso a la funcionalidad del observatorio astronómico Francisco Sánchez. Pero teniendo en cuenta que durante el desarrollo de este proyecto no estarán disponibles gran parte de los controladores del observatorio, las funcionalidades de cada uno de esos dispositivos se ejecutarán de forma simulada. Así se facilitará la integración final en un futuro cercano, y que a su vez se pueda desarrollar y probar aplicaciones cliente de la funcionalidad de la pasarela sin que el trabajo de integración en el observatorio haya finalizado.

Con estas condiciones previas sobre la mesa, los objetivos planteados para el proyecto son:

- Diseñar e implementar una pasarela de acceso de manera modular para poder restringir o ampliar su funcionalidad de manera sencilla.
- Definir la interfaz de programación de aplicaciones para cada una de las áreas funcionales a cubrir:
 - Gestión de usuarios y roles
 - Gestión de reservas de uso
 - Gestión de control y autorización de acceso
 - Control de la cúpula
 - Control de la montura del telescopio
 - Control de las cámaras
 - Control de la estación meteorológica
- Implementar, de manera completa, las funcionalidades de:
 - Gestión de usuarios y roles

- Gestión de reservas de uso
- Gestión de control y autorización de acceso
- Implementar, de manera simulada, las funcionalidades de:
 - Control de la cúpula
 - Control de la montura del telescopio
 - Control de las cámaras
 - Control de la estación meteorológica
- Implementar una interfaz de pruebas que permita verificar de manera sencilla el correcto funcionamiento de la pasarela y sus diferentes funcionalidades.

1.4. Estructura del libro

Este libro está organizado en varios capítulos que desgranar los objetivos principales del proyecto:

- En el primer capítulo se introduce el contexto del proyecto, el problema a resolver y los objetivos perseguidos.
- En el segundo capítulo se detalla el trabajo realizado para la definición de los requisitos que deberá cumplir la implementación del proyecto.
- En el tercer capítulo se expone el proceso seguido para, cumpliendo los requisitos del capítulo 2, llegar a la definición de las interfaces de programación que se expondrán sobre HTTP para permitir la teleoperación del observatorio. En la parte inicial del capítulo se introducirá el trabajo que está realizando sobre objetos web el Consorcio WWW (en inglés World Wide Web Consortium o de manera abreviada W3C). Y en la última parte del capítulo se utilizarán parte de los resultados de ese trabajo para definir los interfaces de programación.
- En el cuarto capítulo el foco es la implementación de la plataforma. Primero se desgrana el diseño de la arquitectura y la tecnología seleccionada para el desarrollo de la misma y de las funcionalidades a cubrir. Y por último se describen las pruebas realizadas al resultado de la implementación.
- En el quinto capítulo se exponen las principales conclusiones extraídas del trabajo en el proyecto y las futuras líneas de trabajo que quedan abiertas.
- Finalmente en los anexos se detallan las definiciones finales de todos los objetos web que representarán las interfaces de programación de la pasarela de acceso del observatorio.

2. EL SISTEMA DE CONTROL

2.1. Introducción

El observatorio astronómico Francisco Sánchez necesitaba de una puesta a punto para volver a ser teleoperado a través de una plataforma abierta y libre. Con vistas a ofrecer un mejor servicio esta puesta a punto comprende una remodelación tecnológica completa que suponga un punto y aparte en la vida operativa del observatorio.

Durante los diez años de existencia del observatorio, el diseño de sus mecanismos de automatización giraban en torno a un único nodo de control central. Un ordenador personal al que se conectaban directamente de manera cableada casi todos los controladores electrónicos existentes en el observatorio: el controlador de los motores de giro y apertura de la cúpula, la montura del telescopio y las diferentes cámaras desplegadas.

En el ordenador personal se ejecutaban los programas de software que permitían comunicarse y controlar los dispositivos conectados. Y a su vez ejecutaba también varios programas informáticos que exponían la funcionalidad de dichos controladores, a través de diversos protocolos de comunicación, en la red local a la que se encontraba conectado el ordenador.

Otro ordenador, conectado en la misma red local y que ejercía de servidor, ejecutaba un sistema informático que implementaba la funcionalidad de teleoperación del observatorio comunicándose como correspondiera con cada controlador software desplegado en el nodo de control central. Este software integraba también la aplicación web de usuario.

Con este diseño, cualquier fallo en el nodo de control central dejaba el observatorio completamente fuera de servicio, y obligaba a un excesivo mantenimiento y vigilancia durante su tiempo de vida funcional. Y la concepción monolítica del sistema informático del nodo servidor dificultaban el desarrollo de nuevas aplicaciones cliente para la realización de experimentos con el observatorio.

La actual remodelación parte de un diseño que pretende ofrecer mayor robustez ante fallos en cualquiera de los componentes electrónicos del observatorio, así como facilitar el mantenimiento del servicio central y el desarrollo de diferentes aplicaciones cliente. Para ello cada uno de los elementos del observatorio será controlado por un dispositivo que actuará como un nodo independiente. Todos los nodos existentes se encontrarán conectados a la red local y permitirán la teleoperación del elemento que controlen, a través de un protocolo sobre HTTP. La intención final es, por tanto, que cada nodo independiente funcione como un dispositivo del Internet de las cosas. Aunque ninguno de ellos estará accesible más allá de la red local.

A su vez se desplegará un nuevo sistema informático sobre un servidor conectado tanto a la red local como a Internet. Y, con un diseño modular, será capaz de controlar cada uno de los dispositivos del Internet de las cosas del observatorio y ofrecer una interfaz de programación RESTful[Fie00] sobre HTTP en Internet. Interfaz que permitirá la búsqueda teleoperación desde nuevas aplicaciones cliente desarrolladas para tal fin.

Este sistema informático ejercerá por tanto de pasarela de control de acceso y teleoperación del observatorio.

En este proyecto se desarrolla la base del sistema informático con la funcionalidad básica imprescindible de control de acceso y teleoperación. Y en los siguientes apartados de este capítulo se recogen los distintos requisitos que debe satisfacer dicho sistema. Dichos requisitos fueron recogidos a partir de las conversaciones mantenidas entre los diferentes colaboradores en la nueva puesta en marcha del observatorio con la base de partida del conocimiento ya acumulado de las anteriores experiencias de automatización.

Cabe reseñar en este sentido que el propio autor de este proyecto había colaborado en los primeros trabajos de automatización del observatorio, con el desarrollo del software que permitía teleoperar la cúpula mediante un protocolo sobre sockets TCP.

2.2. Plataforma de objetos web

El sistema informático a desarrollar tiene un requisito principal crítico, y es que debe contar con un diseño modular que permita su ampliación o limitación de funcionalidad de una manera sencilla. La funcionalidad básica de la que parte su concepción comprende dos partes bien diferenciadas:

- Funcionalidad de control de acceso
- Funcionalidad de teleoperación

Ambas capas de funcionalidad estarán implementadas por módulos independientes. En el caso concreto de la funcionalidad de teleoperación cada módulo debe corresponder con el control de un dispositivo concreto del observatorio. Dispositivo que será considerado como un objeto del Internet de las cosas. Y el módulo servirá de puente de conexión entre el mundo exterior y dicho dispositivo, ofreciendo una interfaz de programación RESTful sobre HTTP[Fie00] que habilite la teleoperación de dicho dispositivo.

Por otro lado el módulo encargado de la funcionalidad de control de acceso debe incorporar los mecanismos necesarios de gestión de la seguridad y autorización de acceso que garanticen durante una franja de tiempo la teleoperación en exclusiva del observatorio por un único usuario. Aunque existirán dos excepciones concretas para administradores del sistema y usuarios con permisos excepcionales que serán detalladas en el apartado 2.3.1.

La funcionalidad de este módulo se subdividirá, por tanto, en las áreas de gestión de usuarios y roles, gestión de reservas y gestión de las autorizaciones de acceso.

Cada funcionalidad expuesta a través de cada uno de estos módulos podrá limitar su uso a un rol específico (por ejemplo solo usuarios con rol de administrador) y requerir una reserva activa o no. Aunque también podrán exponer funcionalidad no

restringida y que permanezca disponible incluso para usuarios no registrados en el sistema (usuarios anónimos).

Y es que otro requisito que debe cumplir la implementación de la plataforma es facilitar el autodescubrimiento de las interfaces de programación expuestas. El objetivo de este requisito es que agentes externos que desconozcan todo sobre este sistema sean capaces de deducir su interfaz y utilizarlo para extraer información anónima del mismo. Para este requisito se puso sobre la mesa el estudio del trabajo que se encuentra realizando actualmente el consorcio W3C sobre objetos web[W3C17a].

Como fruto del estudio del trabajo del W3C surgió la idea de cumplir con todos los requisitos anteriores implementando el sistema informático como una plataforma de despliegue de objetos web.

Los objetos web son representaciones web de objetos reales, o virtuales, que permiten la interacción con el objeto representado a través de las tecnologías web ya existentes[GT16b]. Usualmente son protocolos REST[Fie00] sobre HTTP para exponer interfaces de programación y JSON-LD[W3C14] como formato de intercambio de datos.

En este caso cada objeto web será la representación de cada área funcional requerida, requiriendo la implementación de los siguientes objetos web por área:

Funcionalidad de control de acceso	Pasarela de acceso
Funcionalidad de teleoperación	Cúpula
	Montura del telescopio
	Cámara
	Estación meteorológica

A partir de la información recopilada sobre los objetos web y la funcionalidad a implementar, los primeros requisitos exigidos al sistema informático a desarrollar se concretaron en los siguientes puntos:

- El sistema funcionará como una plataforma de objetos web que integrará la funcionalidad de servidor web y de encaminamiento hacia los recursos ofrecidos por los objetos web desplegados sobre la plataforma.
- El sistema permitirá configurar el puerto sobre el que habilitar el servidor web, la ruta base del mismo, y las opciones necesarias para desplegar un servidor HTTPS.
- El sistema permitirá, durante su ejecución, el despliegue y repliegue de cualquier objeto web.
- Un objeto web estará compuesto por un controlador (implementación del objeto web), una especificación de su interfaz, y su configuración de despliegue.
- Un objeto web dispondrá al menos de dos parámetros de configuración que serán:

- Nombre con el que se desplegará el objeto web
 - Identificador del controlador a desplegar como objeto web
- Un objeto web podrá ser configurado en el momento de su despliegue.
 - Se podrá desplegar varias instancias de un objeto web siempre que se nombren de forma diferente.
 - Un objeto web tendrá como funcionalidad mínima el acceso a la especificación de su interfaz.
 - La funcionalidad de un objeto web podrá ser de acceso restringido a un rol específico y requerir de una reserva de uso en curso en la plataforma.
 - El sistema expondrá a su vez como un objeto web la funcionalidad de:
 - Recoger la información del estado de la plataforma
 - Obtener el listado de los objetos web desplegados sobre la plataforma y sus rutas de acceso
 - El sistema tratará las fechas y horas siempre con en base a la ISO 8601 en Hora Zulú. Ejemplo de formato a seguir: 2017-04-14T12:30:00.000Z (año-mes-díaT:hora:minutos:segundos:milisegundosZ).

2.3. Pasarela de acceso

La pasarela de acceso es un objeto web que implementa la funcionalidad de control de acceso del sistema de control del observatorio. Sus áreas funcionales comprenden la gestión de roles, gestión de usuarios, gestión de reservas y el control de acceso y autorización. El sistema donde se despliegue este objeto web permitirá conocer las interacciones que sea posible mantener con el objeto a través de la siguiente función:

Obtener la especificación del interfaz del objeto web	
Rol autorizado	Anónimo
Reservable	No
Parámetros de entrada	-
Función	Recuperar la definición de la interfaz del objeto web.
Salida	Definición de la interfaz web

2.3.1. Gestión de roles

Los roles es el recurso básico con el que comenzar el modelado de la pasarela de acceso. Un rol representa un nivel de autorización de acceso a una interacción con un objeto web. Los niveles de permiso se numeran a partir del 0, siendo el 0 el nivel de mayor permiso de acceso. A mayor número, menor será el nivel de permiso.

Un rol se define mediante dos parámetros: un nombre y un nivel de permisos. Solo puede haber un rol con el mismo nombre. Pero puede haber varios roles con el mismo nivel de permiso.

Un rol tiene permiso de acceso a la funcionalidad marcada con ese rol, o a la marcada con roles de nivel inferior.

Sin embargo, la funcionalidad marcada como reservable (concepto que se detallará en la sección 2.3.3) requerirá además una reserva en curso propiedad del usuario que esté intentando acceder a la funcionalidad.

Los roles de nivel 0 tienen la propiedad de poder acceder a cualquier funcionalidad que le permita su permiso de acceso sin disponer de una reserva en curso.

Los roles de nivel 1 tienen la propiedad de poder acceder a la funcionalidad marcada con el nombre del rol sin disponer de una reserva en curso. Pero para el resto de funcionalidades deben de disponer de la reserva si se requiere.

Los roles de nivel inferior siempre deben disponer de una reserva activa si la funcionalidad así lo requiere.

La funcionalidad anónima (no marcada con roles) podrá ser accedida por cualquier rol en cualquier instante.

Un rol no puede acceder a la funcionalidad marcada con otro rol del mismo nivel o nivel superior (menor numeración).

La pasarela de acceso dispondrá de tres roles por defecto:

- Administrador de nivel 0

- Privilegiado de nivel 1

- Autenticado de nivel 2

Las funcionalidades relacionadas con la gestión de roles que integra la pasarela de acceso son:

Obtener listado de todos los roles	
Rol autorizado	Administrador
Reservable	No
Parámetros de entrada	identificador de autenticación
Función	Recuperar todos los roles registrados con los usuarios asignados al rol.
Salida	[{nombre de rol, nivel, [nombre de usuario], fecha de creación}]

Obtener información de rol por nombre	
Rol autorizado	Administrador
Reservable	No
Parámetros de entrada	identificador de autenticación y nombre del rol
Función	Recuperar los datos del rol cuyo nombre coincida con el del parámetro de entrada con los usuarios asignados al rol.
Salida	{nombre de rol, nivel, [nombre de usuario], fecha de creación}

Obtener listado de roles por nivel	
Rol autorizado	Administrador
Reservable	No
Parámetros de entrada	identificador de autenticación y número de nivel
Función	Recuperar todos los roles, cuyo nivel coincida con el del parámetro de entrada, con los usuarios asignados al rol.
Salida	[[{nombre de rol, nivel, [nombre de usuario], fecha de creación}]]

Añadir rol	
Rol autorizado	Administrador
Reservable	No
Parámetros de entrada	identificador de autenticación, nombre del rol y número de nivel
Función	Añadir un nuevo rol a la plataforma con el nombre y nivel indicados.
Salida	-

Borrar rol por nombre	
Rol autorizado	Administrador
Reservable	No
Parámetros de entrada	identificador de autenticación y nombre del rol
Función	Eliminar un rol existente a la plataforma con el nombre indicado.
Salida	-

2.3.2. Gestión de usuarios

El usuario es el recurso principal gestionado por la pasarela de acceso. Se define por un nombre, dirección de correo electrónico y una clave de usuario. Solo puede haber un usuario registrado con un mismo nombre, o con una misma dirección de correo electrónico. Un usuario podrá tener asignado uno o varios roles. Al registrar

un usuario por defecto se le asignará el rol Autenticado. Los permisos de acceso del usuario serán los correspondientes a la suma de todos los permisos que correspondan a cada rol asignado.

Un usuario recién registrado necesitará confirmar su registro como paso previo a poder utilizar la funcionalidad expuesta por la pasarela de acceso.

La pasarela de acceso dispondrá de los siguientes usuarios registrados por defecto:

- administrador, con dirección de correo electrónico administrador@usuario.com, con clave 12345678 y rol Administrador

- privilegiado, con dirección de correo electrónico privilegiado@usuario.com, con clave 12345678 y rol privilegiado

- autenticado, con dirección de correo electrónico autenticado@usuario.com, con clave 12345678 y rol autenticado

Las funcionalidades relacionadas con la gestión de usuario que integra la pasarela de acceso son:

Obtener listado de la información de todos los usuarios	
Rol autorizado	Administrador
Reservable	No
Parámetros de entrada	identificador de autenticación
Función	Recuperar todos los usuarios con todos sus datos y roles asignados.
Salida	[{nombre de usuario, dirección de correo electrónico, estado de conexión, estado de la confirmación de registro, [nombre de rol], fecha de creación, fecha de modificación}]

Obtener información de usuario	
Rol autorizado	Autenticado
Reservable	No
Parámetros de entrada	identificador de autenticación
Función	Recuperar los datos del usuario autenticado con los roles que tenga asignados.
Salida	{nombre de usuario, dirección de correo electrónico, [nombre de rol], fecha de creación, fecha de modificación}

Obtener información de usuario por nombre	
Rol autorizado	Administrador
Reservable	No
Parámetros de entrada	identificador de autenticación y nombre de usuario
Función	Recuperar los datos del usuario cuyo nombre coincida con el del parámetro de entrada, con los roles que tenga asignados.
Salida	{nombre de usuario, dirección de correo electrónico, estado de conexión, estado de la confirmación de registro, [nombre de rol], fecha de creación, fecha de modificación}

Obtener información de usuario por dirección de correo electrónico	
Rol autorizado	Administrador
Reservable	No
Parámetros de entrada	identificador de autenticación
Función	Recuperar los datos del usuario cuya dirección de correo electrónico coincida con el del parámetro de entrada, con los roles que tenga asignados.
Salida	{nombre de usuario, dirección de correo electrónico, estado de conexión, estado de la confirmación de registro, [nombre de rol], fecha de creación, fecha de modificación}

Borrar usuario	
Rol autorizado	Autenticado
Reservable	No
Parámetros de entrada	identificador de autenticación
Función	Borrar el usuario autenticado.
Salida	-

Borrar usuario por nombre	
Rol autorizado	Administrador
Reservable	No
Parámetros de entrada	identificador de autenticación y nombre de usuario
Función	Borrar el usuario cuyo nombre coincida con el del parámetro de entrada.
Salida	-

Cambiar clave de usuario	
Rol autorizado	Autenticado
Reservable	No
Parámetros de entrada	identificador de autenticación, nueva clave de usuario
Función	Cambiar la clave del usuario autenticado
Salida	-

Registrar usuario	
Rol autorizado	Anónimo
Reservable	No
Parámetros de entrada	nombre de usuario, dirección de correo electrónico, clave de usuario
Función	Registrar nuevo usuario si no existe ningún otro registrado con el mismo nombre o dirección de correo electrónico. Se le enviará al usuario un código de confirmación a su dirección de correo electrónico. El nuevo usuario no estará disponible hasta no ser verificado. Si no es verificado en un plazo de una hora, se cancelará el proceso de registro.
Salida	-

Confirmar registro de usuario	
Rol autorizado	Anónimo
Reservable	No
Parámetros de entrada	nombre de usuario, dirección de correo electrónico y código de confirmación
Función	Confirmar el registro del usuario cuyos datos de confirmación coincidan con los parámetros de entrada.
Salida	-

Recuperar clave de usuario	
Rol autorizado	Anónimo
Reservable	No
Parámetros de entrada	nombre de usuario y dirección de correo electrónico
Función	Generar una nueva clave de usuario para el usuario cuyos datos coincidan con los parámetros de entrada. Se enviará la nueva clave a la dirección de correo electrónico del usuario junto a un código de confirmación de cambio de clave. La nueva clave no estará disponible hasta no ser verificada. Si no es verificada en un plazo de una hora, se cancelará el proceso de cambio de clave.
Salida	-

Confirma recuperación de clave	
Rol autorizado	Anónimo
Reservable	No
Parámetros de entrada	nombre de usuario, dirección de correo electrónico y código de confirmación
Función	Confirmar el cambio de clave del usuario cuyos datos de confirmación coincidan con los parámetros de entrada.
Salida	-

Añadir usuario a rol	
Rol autorizado	Administrador
Reservable	No
Parámetros de entrada	identificador de autenticación, nombre de usuario y nombre de rol
Función	Añadir el usuario cuyo nombre coincida con el del parámetro de entrada, al rol cuyo nombre coincida con el del parámetro de entrada.
Salida	-

Borrar usuario de rol	
Rol autorizado	Administrador
Reservable	No
Parámetros de entrada	identificador de autenticación, nombre de usuario y nombre de rol
Función	Borrar el usuario cuyo nombre coincida con el del parámetro de entrada, del rol cuyo nombre coincida con el del parámetro de entrada.
Salida	-

2.3.3. Gestión de reservas

Una reserva permite a un usuario, con el nivel de permiso requerido, utilizar en exclusividad la funcionalidad marcada con un rol y el requerimiento de reserva, de cualquier objeto web desplegado sobre la plataforma. Solo los usuarios con roles de nivel 0 o con nivel 1 pueden romper esa exclusividad de acceso tal y como se explicó en el apartado 2.3.1: los usuarios con roles de nivel 0 no requerirán reserva para ninguna funcionalidad, y los de nivel 1 no requerirán reserva para la funcionalidad marcada específicamente con el nombre del rol de nivel 1 al que pertenezca el usuario.

Una reserva se define con una franja horaria (con una fecha y hora de inicio y una fecha y hora de fin) y un usuario propietario de la reserva.

Una reserva se podrá realizar con quince días máximos de antelación, y será de quince minutos de duración pudiendo empezar solo en los minutos horarios : 0, 15, 30 o 45.

Solo podrá existir una única reserva por franja horaria.

Un usuario podrá realizar máximo 4 reservas para un mismo día, aunque éstas podrán ser en franjas horarias consecutivas. Permitiendo así la reserva máxima de una hora por usuario y día.

Las funcionalidades relacionadas con la gestión de reservar que integra la pasarela de acceso son:

Obtener todas las reservas en una fecha concreta	
Rol autorizado	Anónimo
Reservable	No
Parámetros de entrada	identificador de autenticación y fecha
Función	Recuperar todas las reservas realizadas en una fecha concreta, indicada como parámetro de entrada.
Salida	Si el usuario no tiene el rol Administrador: [fecha y hora de inicio, fecha y hora de fin, fecha de creación] Si el usuario tiene el rol Administrador: [fecha y hora de inicio, fecha y hora de fin, nombre de usuario, fecha de creación]

Obtener listado de todas las reservas	
Rol autorizado	Anónimo
Reservable	No
Parámetros de entrada	identificador de autenticación
Función	Recuperar todas las reservas existentes en el sistema para los próximos 15 días.
Salida	Si el usuario no tiene el rol Administrador: [fecha y hora de inicio, fecha y hora de fin, fecha de creación] Si el usuario tiene el rol Administrador: [fecha y hora de inicio, fecha y hora de fin, nombre de usuario, fecha de creación]

Obtener las reservas en una fecha concreta de un usuario autenticado	
Rol autorizado	Autenticado
Reservable	No
Parámetros de entrada	identificador de autenticación y fecha
Función	Recuperar todas las reservas existentes para el usuario autenticado y la fecha indicada como parámetro de entrada.
Salida	[fecha y hora de inicio, fecha y hora de fin, fecha de creación]

Obtener las reservas en una fecha concreta por nombre de usuario	
Rol autorizado	Administrador
Reservable	No
Parámetros de entrada	identificador de autenticación, nombre de usuario y fecha
Función	Recuperar todas las reservas existentes para un usuario y fecha, indicados como parámetros de entrada.
Salida	{{fecha y hora de inicio, fecha y hora de fin, fecha de creación}}

Obtener listado de todas las reservas de un usuario autenticado	
Rol autorizado	Autenticado
Reservable	No
Parámetros de entrada	identificador de autenticación
Función	Recuperar todas las reservas existentes para el usuario autenticado.
Salida	{{fecha y hora de inicio, fecha y hora de fin, fecha de creación}}

Obtener listado de todas las reservas por nombre de usuario	
Rol autorizado	Administrador
Reservable	No
Parámetros de entrada	identificador de autenticación y nombre de usuario
Función	Recuperar todas las reservas existentes para un usuario, indicado como parámetro de entrada.
Salida	{{fecha y hora de inicio, fecha y hora de fin, fecha de creación}}

Obtener disponibilidad en una fecha concreta	
Rol autorizado	Anónimo
Reservable	No
Parámetros de entrada	identificador de autenticación
Función	Recuperar las franjas horarias disponibles para reservar en la fecha indicada como parámetro, junto al número total de franjas que puede reservar el usuario en esa fecha.
Salida	{número de reservas disponibles para el usuario, {{fecha y hora de inicio, fecha y hora de fin}}}

Realizar una reserva	
Rol autorizado	Autenticado
Reservable	No
Parámetros de entrada	identificador de autenticación y fecha y hora de inicio
Función	Reservar para el usuario autenticado la franja horaria que inicia en la fecha indicada como parámetro si no supera el número máximo de reservas por día.
Salida	-

Eliminar una reserva	
Rol autorizado	Autenticado
Reservable	No
Parámetros de entrada	identificador de autenticación y fecha y hora de inicio
Función	Eliminar la reserva en la franja horaria que inicia en la fecha indicada como parámetro siempre que el usuario autenticado sea el dueño de la reserva o el rol del usuario sea Administrador.
Salida	-

Obtener reserva en curso	
Rol autorizado	Autenticado
Reservable	No
Parámetros de entrada	identificador de autenticación
Función	Recuperar los datos de la reserva en curso, si existe, y el usuario propietario de la reserva si la petición la realiza un rol Administrador o si el usuario autenticado es el propietario de la reserva.
Salida	{fecha y hora de inicio, fecha y hora de fin, nombre de usuario, fecha de creación}

2.3.4. Control de acceso y autorización

Una vez que un usuario registrado haya sido verificado podrá requerir un identificador de autenticación que le permitirá acceder a las funcionalidades de los objetos web desplegados sobre la plataforma. Tras obtener el identificador de autenticación lo incluirá en cada interacción que realice con las funciones expuestas por los objetos web. Y cada objeto web comprobará con la pasarela de acceso quién es el dueño del identificador, los niveles de permiso y roles a los que tiene autorización de acceso y si tiene una reserva en curso. Con esa información decidirá si deniega o no la interacción requerida por el usuario.

Los identificadores de autenticación tendrán un tiempo de vida de una hora. Pasado ese tiempo el usuario deberá solicitar un nuevo identificador de autenticación.

Las funcionalidades relacionadas con el control de acceso y autorización que integra la pasarela de acceso son:

Obtener identificador de autenticación de acceso	
Rol autorizado	Anónimo
Reservable	No
Parámetros	nombre y clave de usuario
Función	Generar un identificador de autenticación.
Salida	Identificador de autenticación, , fecha y hora de expiración

Revocar identificador de autenticación	
Rol autorizado	Autenticado
Reservable	No
Parámetros	identificador de autenticación
Función	Cancelar el identificador de autenticación indicado como parámetro.
Salida	-

Obtener permisos de acceso del usuario propietario de un identificador de autenticación	
Rol autorizado	Autenticado
Reservable	No
Parámetros	identificador de autenticación
Función	Recuperar todos los datos de acceso del usuario con un identificador de autenticación vigente igual al pasado como parámetro.
Salida	{nombre de usuario, indicador de si existe una reserva en curso, [nombre de rol]}

2.3.5. Funciones de administración

La pasarela de acceso expondrá varios datos de su estado actual de funcionamiento. Entre ellos se encuentra el modo de funcionamiento, que determina el comportamiento de la pasarela en el proceso de autenticación. Existen 4 posibles modos:

- Modo no restringido: cualquier usuario puede autenticarse y obtener todos los permisos que le correspondan.
- Modo restringido: Solo los usuarios con roles de nivel 0 y/o 1 pueden autenticarse y obtener sus permisos de acceso. La funcionalidad anónima de cualquier objeto web está disponible para todos los usuarios.

- Modo de mantenimiento: Solo los usuarios con rol de nivel 0 pueden autenticarse y obtener sus permisos de acceso. La única funcionalidad anónima disponible será conocer el estado de los objetos web desplegados en la plataforma.
- Parada de emergencia: Ningún usuario puede autenticarse y la plataforma no puede ser teleoperada. La única funcionalidad anónima disponible será conocer el estado de los objetos web desplegados en la plataforma. Este es un modo de funcionamiento que solo debe invocarse en el caso de producirse un problema técnico en el observatorio.

Las funcionalidades relacionadas con la administración de la pasarela de acceso son:

Obtener estado de funcionamiento de la pasarela	
Rol autorizado	Anónimo
Reservable	No
Parámetros	-
Función	Recuperar el estado de funcionamiento del objeto web.
Salida	{nombre del objeto web, modo de funcionamiento, número de usuarios registrados, número de roles registrados, número de usuarios registrados activos, indicador de si existe una reserva en curso}

Fijar modo de funcionamiento no restringido	
Rol autorizado	Administrador
Reservable	No
Parámetros	identificador de autenticación
Función	Activar el modo no restringido de funcionamiento.
Salida	-

Fijar modo de funcionamiento restringido	
Rol autorizado	Administrador
Reservable	No
Parámetros	identificador de autenticación
Función	Activar el modo restringido de funcionamiento.
Salida	-

Fijar modo de funcionamiento de mantenimiento	
Rol autorizado	Administrador
Reservable	No
Parámetros	identificador de autenticación
Función	Activar el modo de mantenimiento de funcionamiento.
Salida	-

Parada de emergencia	
Rol autorizado	Administrador
Reservable	No
Parámetros	identificador de autenticación
Función	Activar el modo de parada de emergencia de funcionamiento.
Salida	-

2.4. Cúpula

Este objeto web permite teleoperar el controlador físico de la cúpula del observatorio. La mayoría de las funciones que expone este objeto web son asíncronas y no devuelven su salida directamente. Devuelven la ruta a un nuevo recurso que reflejará el estado de ejecución de la función:

- Pendiente de ejecución: el comando no ha sido transmitido aún al controlador de la cúpula.
- En ejecución: el comando ha sido transmitido al controlador de la cúpula.
- Fin con éxito: el comando ha sido ejecutado con éxito por el controlador de la cúpula.
- Fin con error: el controlador de la cúpula no ha sido capaz de ejecutar con éxito el comando.

Estas funciones producirán un cambio de estado en la cúpula que se reflejará en los valores devueltos en el estado de funcionamiento del objeto web.

Entre los tipos de datos que manejan las funciones de este objeto web se encuentran:

- latitud, longitud y azimut: vendrán definidos con el formato: XX.YYYY.
- estado de la ventana: la ventana del observatorio podrá estar abierta o cerrada.
- estado del controlador físico: es el último estado de operación conocido del controlador físico, que podrá ser ok si funciona correctamente, error si no funciona correctamente o desconocido si no es posible tener comunicación con el controlador.

Las funcionalidades de este objeto web son:

Obtener la especificación del interfaz del objeto web	
Rol autorizado	Anónimo
Reservable	No
Parámetros	-
Función	Recuperar la definición de la interfaz del objeto web.
Salida	Definición de la interfaz web

Obtener estado de funcionamiento del objeto web	
Rol autorizado	Anónimo
Reservable	No
Parámetros	-
Función	Recuperar el estado de funcionamiento del objeto web, y del controlador físico de la cúpula.
Salida	{nombre del objeto web, nombre del observatorio, localización del observatorio {latitud, longitud}, estado de la ventana, posición de la cúpula {azimut, marca de la posición}, estado del seguimiento, estado del controlador físico, posición HOME {azimut}, posición PARKING {azimut}, fecha y hora de la última actualización de estado}

Abrir ventana	
Rol autorizado	Administrador
Reservable	Sí
Parámetros	identificador de autenticación
Función	Solicitar al controlador de la cúpula que abra la ventana.
Salida	Éxito o no de la ejecución de la función.

Cerrar ventana	
Rol autorizado	Administrador
Reservable	Sí
Parámetros	identificador de autenticación
Función	Solicitar al controlador de la cúpula que cierre la ventana.
Salida	Éxito o no de la ejecución de la función.

Activar seguimiento	
Rol autorizado	Autenticado
Reservable	Sí
Parámetros	identificador de autenticación
Función	Solicitar al controlador de la cúpula que active el modo de seguimiento.
Salida	Éxito o no de la ejecución de la función.

Desactivar seguimiento	
Rol autorizado	Autenticado
Reservable	Sí
Parámetros	identificador de autenticación
Función	Solicitar al controlador de la cúpula que desactive el modo de seguimiento.
Salida	Éxito o no de la ejecución de la función.

Ir a posición Home	
Rol autorizado	Administrador
Reservable	Sí
Parámetros	identificador de autenticación
Función	Solicitar al controlador de la cúpula que sitúe la ventana en la posición de Home.
Salida	Éxito o no de la ejecución de la función.

Aparcar	
Rol autorizado	Administrador
Reservable	Sí
Parámetros	identificador de autenticación
Función	Solicitar al controlador de la cúpula que sitúe la ventana en la posición de Parking.
Salida	Éxito o no de la ejecución de la función.

Definir posición de Parking	
Rol autorizado	Administrador
Reservable	Sí
Parámetros	identificador de autenticación, azimut
Función	Solicitar al controlador de la cúpula cambiar las coordenadas de la posición de Parking a las azimutales indicadas como parámetro.
Salida	Éxito o no de la ejecución de la función.

Ir a azimut	
Rol autorizado	Autenticado
Reservable	Sí
Parámetros	identificador de autenticación, azimut
Función	Solicitar al controlador de la cúpula posicionar la ventana en las coordenadas azimutales indicadas como parámetro.
Salida	Éxito o no de la ejecución de la función.

2.5. Montura del telescopio

Este objeto web permite teleoperar el controlador físico de la montura del telescopio. La mayoría de las funciones que expone este objeto web son asíncronas y no devuelven su salida directamente. Devuelven la ruta a un nuevo recurso que reflejará el estado de ejecución de la función:

- Pendiente de ejecución: el comando no ha sido transmitido aún al controlador de la montura del telescopio.
- En ejecución: el comando ha sido transmitido al controlador de la montura del telescopio.

- Fin con éxito: el comando ha sido ejecutado con éxito por el controlador de la montura del telescopio.
- Fin con error: el controlador de la montura del telescopio no ha sido capaz de ejecutar con éxito el comando.

Estas funciones producirán un cambio de estado en la montura del telescopio que se reflejará en los valores devueltos en el estado de funcionamiento del objeto web.

Entre los tipos de datos que manejan las funciones de este objeto web se encuentran:

- latitud, longitud, azimut, ra y dec: vendrán definidos con el formato: XX.YYYY.
- estado de la función de seguimiento: el seguimiento podrá estar activado o desactivado.
- estado del controlador físico: es el último estado de operación conocido del controlador físico, que podrá ser ok si funciona correctamente, error si no funciona correctamente o desconocido si no es posible tener comunicación con el controlador.
- dirección: irá definido por uno de los puntos cardinales norte, sur, este u oeste.
- intervalo de tiempo: irá definido en milisegundos.

Las funcionalidades de este objeto web son:

Obtener la especificación del interfaz del objeto web	
Rol autorizado	Anónimo
Reservable	No
Parámetros	-
Función	Recuperar la definición de la interfaz del objeto web.
Salida	Definición de la interfaz web

Obtener estado de funcionamiento del objeto web	
Rol autorizado	Anónimo
Reservable	No
Parámetros	-
Función	Recuperar el estado de funcionamiento del objeto web, y del controlador físico de la montura del telescopio.
Salida	{nombre del objeto web, estado de la función de seguimiento, intervalo de tiempo de seguimiento, posición de la montura (altitud, azimut, ra, dec), estado del controlador físico, fecha y hora de la última actualización de estado}

Activar seguimiento	
Rol autorizado	Autenticado
Reservable	Sí
Parámetros	identificador de autenticación, tiempo de actualización
Función	Solicitar al controlador de la montura que active la función de seguimiento que irá corrigiendo la posición de la montura cada intervalo de tiempo indicado como parámetro.
Salida	Éxito o no de la ejecución de la función.

Desactivar seguimiento	
Rol autorizado	Autenticado
Reservable	Sí
Parámetros	identificador de autenticación
Función	Solicitar al controlador de la montura que desactive la función de seguimiento.
Salida	Éxito o no de la ejecución de la función.

Mover por tiempo	
Rol autorizado	Autenticado
Reservable	Sí
Parámetros	identificador de autenticación, dirección , intervalo de tiempo
Función	Solicitar al controlador de la montura que active la función de mover la montura en la dirección indicada durante el intervalo de tiempo indicado.
Salida	Éxito o no de la ejecución de la función.

Mover por coordenadas altazimutales	
Rol autorizado	Autenticado
Reservable	Sí
Parámetros	identificador de autenticación, altitud, azimut
Función	Solicitar al controlador de la montura que desplace la montura hasta las coordenadas altazimutales indicadas como parámetro.
Salida	Éxito o no de la ejecución de la función.

Mover por coordenadas ecuatoriales	
Rol autorizado	Autenticado
Reservable	Sí
Parámetros	identificador de autenticación, ra, dec
Función	Solicitar al controlador de la montura que desplace la montura hasta las coordenadas ecuatoriales indicadas como parámetro.
Salida	Éxito o no de la ejecución de la función.

2.6. Cámara

Este objeto web permite teleoperar el controlador físico de una cámara del observatorio. En el observatorio pueden existir diversas cámaras por lo que este objeto web podría encontrarse desplegado varias veces con nombres diferentes, representando cada instancia a un controlador físico de cámara diferente. Algunas de las funciones que expone este objeto web son asíncronas y no devuelven su salida directamente. Devuelven la ruta a un nuevo recurso que reflejará el estado de ejecución de la función:

- Pendiente de ejecución: el comando no ha sido transmitido aún al controlador de la cámara.
- En ejecución: el comando ha sido transmitido al controlador de la cámara.
- Fin con éxito: el comando ha sido ejecutado con éxito por el controlador de la cámara.
- Fin con error: el controlador de la cámara no ha sido capaz de ejecutar con éxito el comando.

Algunas funciones producirán un cambio de estado en la cámara que se reflejará en los valores devueltos en el estado de funcionamiento del objeto web.

Entre los tipos de datos que manejan las funciones de este objeto web se encuentran:

- tiempo de exposición: irá definido en milisegundos.
- brillo, ganancia y calidad: vendrán definidos por un número entero de 0 a 100.

Las funcionalidades de este objeto web son:

Obtener la especificación del interfaz del objeto web	
Rol autorizado	Anónimo
Reservable	No
Parámetros	-
Función	Recuperar la definición de la interfaz del objeto web.
Salida	Definición de la interfaz web

Obtener estado de funcionamiento del objeto web	
Rol autorizado	Anónimo
Reservable	No
Parámetros	-
Función	Recuperar el estado de funcionamiento del objeto web, y del controlador físico de la cámara.
Salida	{nombre del objeto web, estado del controlador físico, número de capturas realizadas, fecha de la última actualización de estado}

Capturar imagen	
Rol autorizado	Autenticado
Reservable	Sí
Parámetros	identificador de autenticación, tiempo de exposición, brillo, ganancia y calidad
Función	Solicitar al controlador de la cámara que capture una imagen con los parámetros de configuración indicados. Una vez capturada la imagen se registra su información.
Salida	{Éxito o no de la ejecución de la función, identificador de la captura, ruta de la foto capturada}

Cancelar captura de imagen en curso	
Rol autorizado	Autenticado
Reservable	Sí
Parámetros	identificador de autenticación
Función	Solicitar al controlador de la cámara que cancele la captura de imagen en curso.
Salida	Éxito o no de la ejecución de la función.

Obtener listado de todas las capturas con su información	
Rol autorizado	Anónimo
Reservable	No
Parámetros	-
Función	Recuperar la información de todas las capturas realizadas con éxito por la cámara.
Salida	{[identificador de la captura, nombre de imagen, descripción, ruta de la imagen, tiempo de exposición, brillo, ganancia, calidad, fecha de captura, nombre de usuario que solicitó la captura, fecha de modificación]}

Obtener información de una captura concreta	
Rol autorizado	Anónimo
Reservable	No
Parámetros	identificador de la captura
Función	Recuperar la información de la captura cuyo identificador se pasa por parámetro
Salida	{identificador de la captura, nombre de imagen, descripción, ruta de la imagen, tiempo de exposición, brillo, ganancia, calidad, fecha de captura, nombre de usuario que solicitó la captura, fecha de modificación}

Modificar información de una captura concreta	
Rol autorizado	Dueño del recurso con rol Autenticado
Reservable	No
Parámetros	identificador de autenticación, identificador de la captura, nombre de imagen y descripción
Función	Recuperar la información de la captura cuya identificador se pasa por parámetro
Salida	Éxito o no de la ejecución de la función.

2.7. Estación Meteorológica

Este objeto web permite conocer el estado de operación del controlador físico de la estación meteorológica. Y a su vez mantiene un histórico de las lecturas realizadas por los sensores de la estación.

Entre los tipos de datos que manejan las funciones de este objeto web se encuentran:

- latitud, longitud: vendrán definidos con el formato: XX.YYYY.
- estado del controlador físico: es el último estado de operación conocido del controlador físico, que podrá ser ok si funciona correctamente, error si no funciona correctamente o desconocido si no es posible tener comunicación con el controlador.
- temperatura: medida en grados centígrados con el formato XX.Y
- presión: medida en hectopascales con el formato XXXX.Y.
- humedad: medida en % de 0 a 100.
- velocidad del viento: medido en kilómetros por hora con el formato XX.
- dirección del viento: definida por los puntos cardinales norte, noreste, este, sureste, sur, suroeste, oeste, noroeste.

Las funcionalidades de este objeto web son:

Obtener la especificación del interfaz del objeto web	
Rol autorizado	Anónimo
Reservable	No
Parámetros	-
Función	Recuperar la definición de la interfaz del objeto web.
Salida	Definición de la interfaz web

Obtener estado de funcionamiento del objeto web	
Rol autorizado	Anónimo
Reservable	No
Parámetros	-
Función	Recuperar el estado de funcionamiento del objeto web, la última lectura de los sensores de la estación meteorológica y el estado de funcionamiento de su controlador físico.
Salida	{nombre del objeto web, localización (latitud, longitud), temperatura, presión, humedad, velocidad del viento, dirección del viento, estado del controlador físico, fecha de la última actualización de estado}

Obtener histórico de registros de lectura	
Rol autorizado	Anónimo
Reservable	No
Parámetros	-
Función	Recuperar todos los registros existentes de las lecturas realizadas de los sensores de la estación meteorológica durante las últimas 24 horas.
Salida	[[{temperatura, presión, humedad, velocidad del viento, dirección del viento, estado del controlador físico, fecha de lectura}]]

3. ESPECIFICACIÓN DE OBJETOS WEB

3.1. Introducción

Al comenzar el trabajo en este proyecto se propuso el estudio del concepto de objeto web basado en el trabajo que se encuentra realizando el consorcio WWW (W3C a partir de ahora) para desarrollar los estándares iniciales de la web de las cosas (Web of Things en inglés, y también referido por las siglas WoT).

El primer paso en este estudio es aclarar el concepto de la web de las cosas. Este concepto nace a partir de otro muy en voga actualmente: el Internet de las cosas (Internet of Things en inglés, referido habitualmente por las siglas IoT). El Internet de las cosas no es más que la interconexión de cualquier tipo de objeto del mundo físico con redes digitales y eventualmente a Internet. A través de esta interconexión el mundo físico se integra con las redes digitales y se hace posible el acceso a la información generada en el mundo real, y la monitorización y el control total o parcial de los objetos físicos.[GT16b]

Claros ejemplos en el mercado son los productos denominados termostatos inteligentes para el entorno doméstico. Estos dispositivos permiten conocer y ajustar el estado de confortabilidad de un hogar en base a la temperatura interior y exterior. El ajuste de la confortabilidad lo realiza el termostato encendiendo o apagando la calefacción cuando lo estime necesario siguiendo las indicaciones dadas por el usuario y el conocimiento que va adquiriendo del hogar y las costumbres de sus ocupantes con el paso del tiempo. Por ejemplo van aprendiendo la inercia térmica del hogar o los horarios de ocupación y desocupación del mismo.

Pero para conocer algunos de esos datos se vale de sensores que no forman parte del propio termostato. Sensores de temperatura repartidos por la casa, por el exterior, de sistemas de predicción meteorológica o de sistemas de posicionamiento que indiquen la localización de cada ocupante del hogar. Todos estos dispositivos se interconectan digitalmente a través de diferentes redes y las interacciones entre ellos complementan el funcionamiento del termostato inteligente. Sobre el que además los ocupantes del hogar pueden interactuar física o remotamente gracias la conexión a Internet del termostato inteligente.

Actualmente el desarrollo de la denominada Internet de las cosas tiene un problema fundamental. Las tecnologías utilizadas para la interconexión digital son muy diversas, muchas de ellas no estandarizadas, y en muchos casos sin capacidad de operar unas con otras. Esto se traduce en la generación de redes digitales de objetos conectados pero aisladas unas de otras. Para resolver este problema es necesario definir y expandir el uso de un estándar que permita al menos la conexión entre redes a nivel de aplicación. Y el trabajo que desarrolla el W3C con la web de las cosas está orientado a proponer un estándar basado en el uso de la web. La web de las cosas nace como representación web del Internet de las cosas. Cada objeto del mundo real interconectado digitalmente adquiere su representación web que permite obtener los datos generados por el objeto, monitorizar su estado y/o controlar

el propio objeto. Y estas representaciones web permiten romper el aislamiento y habilitar la interconexión completa entre dispositivos, es decir, ayudan a reducir la fragmentación en el Internet de las cosas.

La representación web de un objeto del Internet de las cosas corresponde con el concepto de objeto web (Web Thing en inglés). Pero el concepto de objeto web es más amplio aún. Un objeto web no tiene por qué ser solo una representación web de un objeto real del Internet de las cosas. También puede ser la representación web de un objeto virtual. De un objeto no existente en el mundo físico pero sí en el mundo digital.

Por ejemplo un programador de riego de jardín que únicamente existe de forma digital. Pero permite controlar el arranque y la parada del sistema físico de riegos en base al programa de inicio y finalización que un usuario pueda introducir en dicho objeto web. Este poder de abstracción y virtualización de objetos que se integren en el Internet de las cosas hace más interesante aún el concepto de la web de las cosas.

A continuación se estudiarán las propuestas sobre las que está trabajando el W3C para estandarizar la definición de los objetos web y las formas de interactuar con ellos.

3.2. Propuestas de definición de objetos web del W3C

Parte del esfuerzo del grupo de trabajo de la web de las cosas (Web of Things Working Group) del W3C está el generar un estándar para definir un objeto web y sus interacciones. Las diferentes propuestas que han ido surgiendo tienen una característica en común, y es utilizar los estándares web reconocidos por el W3C para modelar el objeto y sus interacciones.

El objeto de estudio para este proyecto son las dos propuestas existentes hasta el momento. La primera en surgir fue el Modelo de Objeto Web (Web Thing Model en inglés), que derivó en el 2015 en una propuesta completa de especificación [W3C15] y que fue igualmente la primera en ser estudiada. Esta definición es usada incluso para generar algunos productos comerciales por parte de algunos de los contribuidores en su definición. Posteriormente el W3C reinicia el trabajo de definición de un objeto web con las lecciones aprendidas y genera otra propuesta, la Descripción de Objeto Web (Web Thing Description en inglés) [W3C17c]. Esta propuesta aún está incompleta en la definición de algunas de sus partes aunque la base de la misma tiene ya suficiente cuerpo como para ser considerada en este estudio.

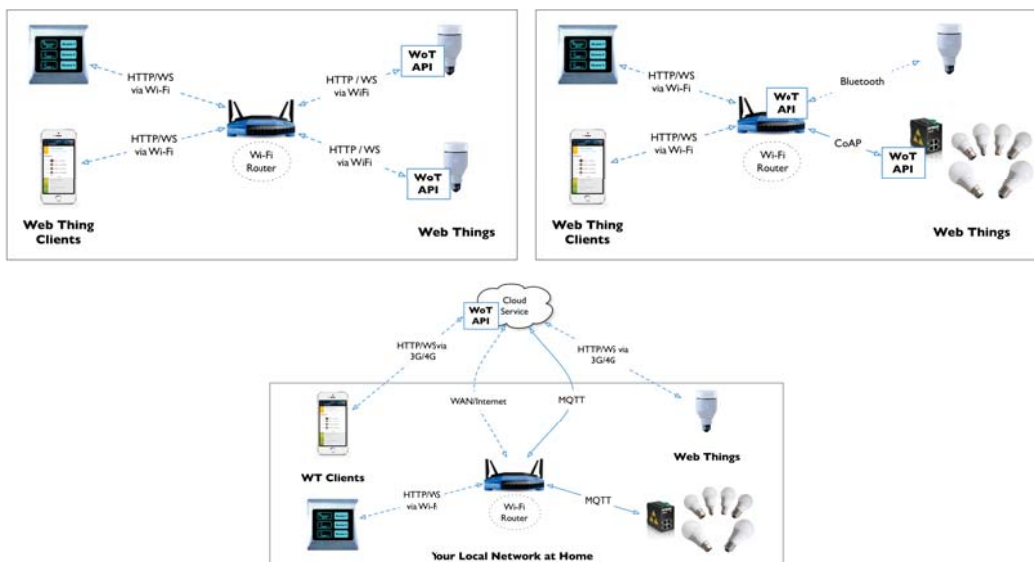
La diferencia más importante entre ambas propuestas es que el Modelo de Objeto Web por defecto define las interacciones con el objeto web en torno al uso del protocolo HTTP. Mientras que la Descripción de Objeto Web permanece agnóstica en ese sentido y permite definir para cada interacción uno o varios protocolos de acceso (HTTP, MQTT, CoAP, etc).

3.2.1. Propuesta basada en el Modelo de Objeto Web

En esta propuesta la definición de un objeto web parte de un conjunto de requisitos mínimos que permitan al objeto intercambiar información con cualquier otro objeto de la web de las cosas. Cualquier servidor web que los respete puede ser considerado un objeto web. Pero para que dicho objeto web entienda realmente con qué tipo de entidades está comunicándose y el tipo de servicios que ofrecen, el Modelo de Objeto Web propone el uso de un protocolo Web RESTful[Thi15] con un conjunto de recursos, modelos de datos y sintaxis de datos determinados. Al objeto web que cumple con dichos requisitos se le denomina objeto web extendido. Por último a este objeto web extendido puede añadirse información semántica utilizando extensiones como JSON-LD [W3C14], convirtiéndolo en el denominado objeto web semántico.

La base de la propuesta es la consideración de tres posibles formas de integrar un objeto físico en la web de las cosas:

- El objeto físico implementa y expone él directamente un interfaz web de programación y es en sí mismo un objeto web.
- El objeto físico no expone ningún interfaz web, pero se conecta a un dispositivo que actúa como pasarela de acceso y a su vez expone un interfaz web de programación representando al objeto físico.
- El objeto físico no expone ningún interfaz web, pero se conecta a un servicio en la nube que actúa como pasarela de acceso y a su vez expone un interfaz web de programación representando al objeto físico.



Fuente: W3C[W3C15]

Figura 3.1: Patrones de integración de objetos web: directa, a través de una pasarela de acceso, y a través de la nube

La interfaz web expuesta en cada uno de esos casos, para ser considerada como un objeto web tiene obligación de cumplir con una serie de requisitos:

- Ser un servidor HTTP/1.1 [FR14e] (y cuando sea posible soportar también HTTP/2).
- Tener un recurso raíz accesible mediante una URL HTTP (dirección web).
- Soportar los métodos GET, POST, PUT y DELETE de HTTP.
- Implementar los códigos de estado HTTP 200 (éxito), 400 (error en el lado del cliente), 500 (error en el lado del servidor).
- Soportar JSON como la representación por defecto de la salida y entrada de datos en las peticiones HTTP.
- Soportar el método GET de HTTP en su URL principal.

Otros requisitos no son obligatorios pero sí deberían cumplirse salvo limitaciones técnicas:

- Soportar el uso de conexiones HTTP seguras (HTTPS).
- Implementar el protocolo de websockets para ofrecer un interfaz de suscripción/publicación de notificaciones.
- Soportar el modelo de objetos web (descrito posteriormente).
- Devolver 204 para cualquier operación de escritura (métodos PUT, POST, DELETE de HTTP).
- Proveer de una documentación por defecto legible por humanos.

Y por último se definen los requisitos recomendados y que no se esperan por defecto del objeto web:

- Soportar el método OPTIONS de HTTP para cada recurso del que disponga.
- Proveer de representaciones adicionales de los datos (RDF, JSON-LD...).
- Proveer de un interfaz de usuario basado en HTML.
- Proveer de información precisa sobre el significado de cada parte individual del modelo de objeto web representado.

Una vez definido qué es un objeto web y qué debe cumplir para ser un objeto web, un objeto web extendido, o un objeto web semántico, a continuación toca definir el Modelo de Objeto Web. Éste contiene la especificación del modelo, la sintaxis de los JSON y del interfaz web de programación REST que deberá implementar un objeto web extendido para permitir a los clientes web descubrir y utilizar automáticamente sus propiedades.

Lo fundamental de esta especificación es el uso de cinco recursos principales:

- El modelo: la descripción estandarizada del objeto web extendido.
- Las propiedades: contienen valores de estado del objeto web, y cada una posee un identificador único.
- Las acciones: las funciones que pueden invocarse sobre el objeto web, y cada una posee un identificador único. Además la invocación de una acción puede generar un recurso nuevo denominado tarea con un identificador nuevo (posteriormente se hará referencia a éste como `actionId`).
- Los objetos: otros objetos web que son accesibles desde éste objeto web, y que también poseen un identificador único.
- Las suscripciones: recursos que permiten monitorizar otros recursos.

Sobre cada uno de esos recursos podrán utilizarse uno o varios de los métodos HTTP, cuya intención queda resumida en la siguiente tabla:

	GET	POST	PUT	DELETE	OPTIONS
/	Obtener la información básica del objeto web	Añadir un nuevo objeto web	Actualizar la información del objeto web	Eliminar el objeto web	Obtener los métodos HTTP disponibles
/model	Obtener el modelo	-	Actualizar el modelo	-	Obtener los métodos HTTP disponibles
/properties	Listar todas las propiedades existentes.	Añadir una nueva propiedad	-	-	Obtener los métodos HTTP disponibles
/properties/{id}	Obtener los valores de una propiedad	-	Actualizar el valor de una propiedad	-	Obtener los métodos HTTP disponibles
/actions	Listar todas las acciones existentes	Añadir una nueva acción	-	-	Obtener los métodos HTTP disponibles
/actions/{id}	Obtener la definición de la acción	Ejecutar la acción	-	-	Obtener los métodos HTTP disponibles
/actions/{id}/{actionId}	Obtener el valor de la tarea de una acción	-	Modificar los parámetros de ejecución de la tarea (solo permitida si la tarea aún está pendiente)	Cancelar la tarea (solo permitida si la tarea aún está pendiente)	Obtener los métodos HTTP disponibles
/subscriptions	Listar las suscripciones existentes	-	-	Eliminar una suscripción	Obtener los métodos HTTP disponibles
/things	Listar los objetos web disponibles	Añadir un nuevo objeto web	-	-	Obtener los métodos HTTP disponibles
/things/{id}	Obtener la información básica de un objeto web	-	Actualizar la información del objeto web	Eliminar el objeto web	Obtener los métodos HTTP disponibles

El único caso no detallado en la tabla superior es la creación de suscripciones a un recurso. Para crear una suscripción a un recurso, y desde ese momento poder monitorizar de forma asíncrona por websockets los cambios producidos en dicho recurso,

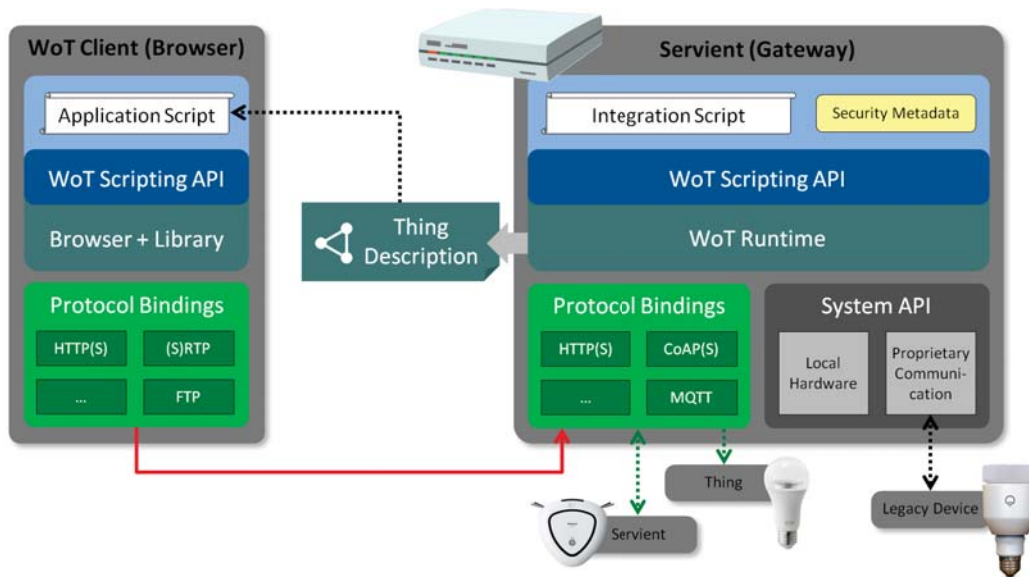
es necesario invocar el método GET de HTTP incluyendo la opción UPGRADE de HTTP con el valor «websocket».

3.2.2. Propuesta basada en la Descripción de Objeto Web

La Descripción de Objeto Web forma parte en realidad de una especificación mucho más amplia. Dentro de esta nueva propuesta la web de objetos se construye en torno a objetos compuestos por varios bloques [W3C17a].

Debido a que esta propuesta se encuentra aún en plena definición, los bloques no están aún definidos en su totalidad. Pero las partes más importantes de cara al proyecto como son el modelado de la interacción con el objeto web y su representación mediante la Descripción de Objeto Web están lo suficientemente detallados como para ser aprovechados.

De la arquitectura que se propone lo importante es conocer que está orientada a facilitar la interoperabilidad del objeto web a través de diferentes redes digitales y diferentes interfaces de programación. Para ello se estandarizan todos los tipos de interacciones que son posibles con un objeto web. Y la Descripción de Objeto Web detalla todas y cada una de las interacciones posibles con el objeto web de acuerdo a los estándares definidos anteriormente.



Fuente: W3C[W3C17a]

Figura 3.2: Objeto web desplegado en una pasarela de acceso acorde a la nueva propuesta de diseño basada en la Descripción de Objeto Web

Tal y como puede observarse en la figura 3.2, a través de la Descripción de Objeto Web cualquier objeto sabe interactuar con otro y comprender su propio funcionamiento. Y el resto de bloques que componen un objeto web se encargan de conectar cada red digital e interfaz de programación soportados por el objeto con las interacciones detalladas en la Descripción de Objeto Web.

En la capa de aplicación se implementan las interacciones del objeto web y se conectan con el resto de bloques del objeto, habilitando la interoperación del objeto en varias redes con cualquier objeto de las mismas.

Por tanto la parte más interesante de esta propuesta para el desarrollo del proyecto es la definición de las posibles interacciones con un objeto web. Se basan en el uso de tres patrones de interacción:

- Propiedades: interacciones que proveen datos que pueden ser leídos y/o escritos, y ser estáticos o dinámicos.
- Acciones: interacciones que disparan cambios o procesos en un objeto que pueden llevar cierto tiempo para ser completados. La ejecución pendiente de una acción suele modelarse como un recurso denominado Tarea. Este recurso se crea al invocarse una acción.
- Eventos: interacciones que habilitan un mecanismo de notificación por parte de un objeto en base a una condición de disparo.

Usando el protocolo HTTP y partiendo de estos patrones de interacción, el interfaz que detalle una Descripción Web de Objeto corresponderá con:

	GET	POST	PUT	DELETE	OPTIONS
URI de una propiedad	Obtener los datos provistos por una propiedad	-	Actualizar los datos de la propiedad.	-	Obtener los métodos HTTP disponibles
URI de una acción	-	Invoca la ejecución de la acción, y podrá crear un recurso Tarea que represente el estado de ejecución de la acción	-	-	Obtener los métodos HTTP disponibles
URI de la tarea de una acción	Obtener el estado de la ejecución de la Tarea	-	Actualizar los parámetros de la Tarea si aún está pendiente de ejecución	Cancelar la Tarea si aún está pendiente de ejecución	Obtener los métodos HTTP disponibles
URI de un evento	Obtener la información del evento	Crear un nuevo evento basado en éste con nuevas condiciones de disparo de notificación	Modificar las condiciones de disparo del evento	Eliminar el evento	Obtener los métodos HTTP disponibles

En este aspecto es una especificación relativamente parecida al Modelo de Objeto Web. Pero la gracia de éste reside en que, aunque en la tabla superior se hayan detallado las operaciones en base al protocolo HTTP, en realidad pueden realizarse a través de otros protocolos. Sin embargo no se entrará en ese detalle al no ser de interés de estudio para los objetivos de este proyecto.

3.3. Definición de objetos web para el sistema de control del observatorio

Una vez estudiadas las dos propuestas anteriores nacidas en el seno del W3C, es necesario decidir qué especificación formará parte de la plataforma de objetos web a desarrollar.

Por un lado la propuesta del Modelo de Objeto Web está completamente definida y no deja lugar a duda en ninguno de sus aspectos. No sucede lo mismo con la Descripción de Objeto Web donde diversos elementos, como por ejemplo los Eventos, no se encuentran completamente definidos e incluso se expone desde el W3C la

necesidad de ahondar en la definición del concepto en sus próximas reuniones de trabajo.

Siendo las dos parecidas en algunas concepciones, la notación utilizada en la segunda propuesta resulta más clara y potente en vistas de los tipos de interfaces que se pretenden modelar. La Descripción de Objeto Web permite la definición tanto de parámetros de entrada como los datos de salida de las acciones.

Además las estructuras de dichos datos pueden ser más complejas y sin embargo se obtienen definiciones menos farragosas a nivel de estructura. A pesar de ello ambas propuestas carecen de elementos que resultan en principio imprescindibles para este proyecto como el marcado con roles de acceso de las distintas interacciones o la indicación de si una interacción requiere o no reserva.

Es por ello que se esquematiza aquí la propuesta del modelado de las interacciones con un objeto web en la plataforma. Y posteriormente se detalla la notación de su descriptor.

3.3.1. Patrones de interacción

A nivel de arquitectura el objetivo es implementar una pasarela de acceso que contenga los objetos web que representan a los distintos dispositivos del observatorio. Expondrá sus interfaces con la Descripción de Objeto Web y servirá de puerta de acceso al exterior para conectar por HTTP los objetos a los clientes de fuera de la red (acorde a lo representado en la figura 3.2).

Tomando como base la Descripción de Objeto Web, se utilizan los mismos patrones de interacción a excepción de los Eventos. Estos carecen actualmente de suficiente nivel de definición como para ser usados. Y la interacción que habilitan, la notificación de eventos desde el objeto web, puede ser suplida temporalmente mediante el uso de técnicas cíclicas de muestreo de los datos.

Así, si el cliente de un objeto web requiere conocer el cambio de estado de una propiedad del objeto, deberá leer el valor de la propiedad cada cierto tiempo para detectar dicho cambio. Podría plantearse el uso del sistema de suscripciones del Modelo de Objeto Web con websockets. El problema es que en la práctica el uso de websockets no genera resultados consistentes, especialmente si se utilizan desde dispositivos móviles donde los websockets tienden a perder la conexión y obliga a realizar un Ping de forma periódica para evitarlo. Y desgraciadamente tampoco esta técnica consigue mantener siempre activo el websocket.

Teniendo todo lo anterior en cuenta, las posibles interacciones con un objeto web en la plataforma en desarrollo serán:

	GET	POST	PUT	DELETE	OPTIONS
URI raíz del objeto	Obtener la Descripción de Objeto Web	-	Actualizar la Descripción de Objeto Web	-	Obtener los métodos HTTP disponibles
URI de una propiedad	Obtener los datos provistos por una propiedad	-	Actualizar los datos de la propiedad.	-	Obtener los métodos HTTP disponibles
URI de una acción	-	Invoca la ejecución de la acción, y si es una tarea asíncrona podrá crear un recurso Tarea que represente el estado de ejecución de la acción. Sino devolverá directamente el resultado de la ejecución.	-	-	Obtener los métodos HTTP disponibles
URI de la tarea de una acción	Obtener el estado de la ejecución de la Tarea	-	Actualizar los parámetros de la Tarea si aún está pendiente de ejecución	Cancelar la Tarea si aún está pendiente de ejecución	Obtener los métodos HTTP disponibles

La diferencia fundamental con la Descripción de Objeto Web, además de la carencia de las interacciones de Eventos, es el añadido de la posibilidad de que una acción pueda invocarse de manera síncrona. Esto implica que la ejecución de la acción no crea un recurso nuevo Tarea, sino que devuelve directamente el resultado de la ejecución de la acción.

Además tanto las interacciones con las propiedades como con las acciones podrán ser de acceso restringido para un rol específico. Rol que podrá ser diferente para las interacciones de lectura frente a las de escritura.

Y también podrá requerirse el disponer de una reserva activa para poder invocar cualquier interacción de tipo acción.

Los permisos de acceso a las interacciones de un objeto web podrán modificarse actualizando su Descripción de Objeto Web con los cambios pertinentes sobre dicha notación.

La Descripción de Objeto Web también podrá especificar la pasarela de objetos web usada por el objeto para comprobar la autenticación de usuarios, sus niveles de permisos y la existencia de reserva activa.

3.3.2. Notación de representación del objeto web

La notación escogida como base para representar un objeto web es la Descripción de Objeto Web [W3C17c]. Pero con algunas diferencias en cuanto a su definición:

- La Descripción de Objeto Web utiliza JSON-LD como formato de datos. En este proyecto se utilizará JSON como formato base de datos, pero utilizando la cabecera «Link Header» de HTTP para referenciar un documento de contexto JSON-LD. Así el documento de base JSON podrá ser interpretado igualmente como JSON-LD, y cualquier cliente de este objeto web podrá reconocer las URIs como tal sobre dicho documento (requisito imprescindible para que un interfaz de programación sea RESTful).
- La Descripción de Objeto Web podrá incluir la definición de un recurso de seguridad especificando qué pasarela de acceso utilizar para la autenticación de usuarios con la etiqueta «gatekeeper» y como valor la URI del objeto web que actúe como pasarela de acceso.
- Una propiedad y una acción podrán restringir el nivel de permiso de acceso utilizando la etiqueta «roleBasedAccessControl» que define el rol mínimo necesario para invocar estas interacciones.
- Puede definirse un nivel de permiso más granular restringiendo con un rol diferente las interacciones que involucren la escritura y/o modificación de una propiedad, o de la Tarea creada como invocación de una acción. La etiqueta a usar en este caso será «roleBasedWritingAccessControl». Si esta etiqueta no existiese pero sí la de «roleBasedAccessControl», éste segunda será la que defina el rol mínimo de permiso de acceso para las interacciones de escritura.
- Por defecto se asume que ninguna interacción de un objeto web requiere de reserva para ser accedida. En caso de que se requiera reserva se debe indicar con el parámetro «reservable» con valor a «true» (verdadero) en la definición de la interacción.
- Por defecto se asume que ni la Descripción de Objeto Web ni una propiedad son modificables, salvo que se indique con el parámetro «writable» con valor a «true» (verdadero) en la definición de cada uno de ellos.
- Para que un cliente del objeto web pueda diferenciar previamente qué acciones son asíncronas y cuales no (lo que involucra un tipo diferente de respuesta en cada caso) se utilizará la etiqueta «asynchronous» con valor verdadero o falso («true» o «false» en inglés). Una acción es asíncrona por defecto.

- La representación de un recurso de tipo Tarea creada por la invocación de una acción asíncrona, y la respuesta de la ejecución de una acción de tipo síncrono, serán iguales. Un archivo JSON con tres propiedades, una representando los parámetros de entrada de la acción («inputData»), otra para el resultado de la acción («outputData») y un tercero que indique el estado de ejecución de la acción («status», con los posibles valores de «OK», «PENDING», «ERROR», «RUNNING»).

En el Anexo A pueden consultarse las definiciones finales de cada uno de los objetos web implementados para el proyecto.

4. IMPLEMENTACIÓN

4.1. Introducción

Tras el análisis de los requisitos a cumplir por el sistema de control del observatorio se decidió concebir al mismo como una plataforma de objetos web. La misión principal de esta plataforma es soportar el despliegue de objetos web. Para ello debe ofrecer las funcionalidades de gestión de objetos web (que incluya las operaciones de despliegue, repliegue y configuración), así como la de un servidor web y un encaminador que permitan recibir y distribuir las interacciones con los objetos web. A su vez la funcionalidad requerida para teleoperar el observatorio y gestionar el control de acceso será ofrecida por los objetos web a desplegar en dicha plataforma.

De aquí surge ya la primera concepción del diseño de alto nivel a implementar, tal y como puede observarse en la siguiente figura.

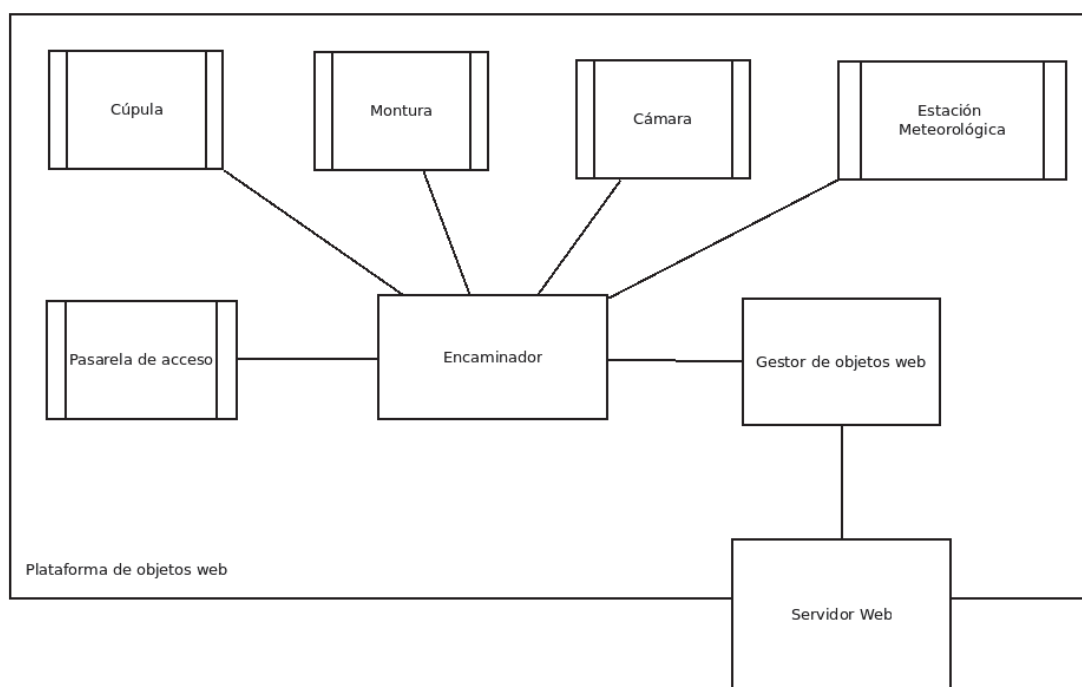


Figura 4.1: Diseño de alto nivel del sistema de control del observatorio

Como se detalla en el capítulo 2, cada objeto web es responsable de un área funcional concreta. Los objetos web Cúpula, Montura, Estación Meteorológica y Cámara controlan cada uno un dispositivo físico concreto. Y el objeto Pasarela de acceso gestiona toda la funcionalidad de gestión de usuarios, roles, reservas y autorizaciones.

La pasarela de acceso será utilizada tanto por los usuarios del observatorio para obtener las autorizaciones de acceso, como por los objetos web para verificar el

identificador de autenticación que utilicen los usuarios en las interacciones con los objetos y comprobar el nivel de permisos y la existencia de una reserva activa del usuario autenticado.

La verificación de permisos está dividida por tanto en dos partes cuya responsabilidad recae en entidades diferentes. La primera parte, implementada por la Pasarela de acceso, corresponde con la gestión de los permisos de acceso de los usuarios que permite conocer para un usuario qué permisos posee (que identificamos como roles), su nombre de usuario y si tiene reserva activa.

Pero la segunda parte de verificación de permisos recae en el objeto web destino de la interacción. Es este objeto el que debe comprobar en base a su Descripción de Objeto Web (que detalla los roles mínimos necesarios para cada interacción y si se requiere de reserva activa), si los permisos que posee el usuario le habilitan para invocar la interacción requerida.

De cara a facilitar el desarrollo de objetos web para la plataforma, ésta y otras funcionalidades serán embebidas dentro de un objeto web genérico. Y el desarrollador de objetos web deberá extender dicho objeto para implementar el suyo heredando toda la funcionalidad del objeto genérico. Así el desarrollo de nuevos objetos web quedará simplificado y exigirá menos conocimientos sobre la plataforma, agilizando también el proceso de aprendizaje de implementación de los objetos.

Respecto a la implementación de los objetos web de la Cúpula, Montura, Estación Meteorológica y Cámara es necesario destacar que no son implementaciones de objetos virtuales y que no mantendrán interacción alguna con dispositivos físicos reales desplegados en el observatorio. Esto es así porque en el momento de encarar la recta final de este proyecto la mayoría de dichos dispositivos no se encuentran finalizados ni mucho menos desplegados. Y el servidor que debía alojar el sistema de control del observatorio dejó de estar operativo justo antes de la recta final del proyecto. Luego no iba a ser posible ejecutar ninguna prueba con dispositivos reales y no podría garantizarse la integración funcional de cada parte del sistema.

Como remedio los objetos virtuales implementados simulan (aunque de forma muy sencilla) la operativa de cada interacción definida en el capítulo 2, permitiendo que las aplicaciones cliente sí pueden desarrollar y probar su integración con las funciones de teleoperación del observatorio.

A su vez los objetos web virtuales están desarrollados de tal forma que sirven de base para desarrollar los objetos web que sí controlen cada dispositivo físico correspondiente.

Se detallará a continuación con mayor nivel la arquitectura diseñada y la base tecnológica empleada para construir el sistema.

4.2. Arquitectura del sistema

Una vez están claros todos los requisitos principales del sistema a desarrollar y las áreas funcionales donde recae cada requisito, es posible diseñar la arquitectura del sistema. Para ello se definen en primer lugar los componentes básicos de la

plataforma de objetos web, y de los objetos web en sí mismos. Dichos componentes y sus definiciones son los siguientes:

- Plataforma de objetos web
 - Servidor HTTP: habilita las interacciones web con los objetos web desplegados sobre la plataforma. Cada interacción recibida la redirige al «Gestor de objetos web».
 - Gestor de objetos web: se encarga del despliegue y repliegue de un objeto web, del registro de su información en el despliegue, y de redirigir las peticiones HTTP a cada objeto web correspondiente. Para el registro de la información de los objetos web utiliza el «Gestor de base de datos». También gestiona las respuestas HTTP de los objetos web y las redirige al Servidor «HTTP». Las redirecciones HTTP a los objetos web las resuelve utilizando el «Gestor de resolución de nombres» y posteriormente reenviando la petición por el «Bus de comunicación».
 - Gestor de base de datos: gestiona una base de datos para la plataforma de objetos web, donde el «Gestor de objetos web» registra la información de cada objeto web desplegado.
 - Gestor de resolución de nombres: permite transformar una dirección URL HTTP a una dirección única interna para usar con el «Bus de comunicación».
 - Bus de comunicación: conecta todos los objetos web entre sí y con el «Gestor de objetos web». Cada objeto web y sus interacciones son asignados direcciones internas únicas que será posible resolver desde URLs HTTP por el «Gestor de resolución de nombres».
 - Generador de objetos web: crea una instancia de un objeto web a partir de una configuración, el nombre de la implementación del objeto web y la Descripción de Objeto Web que le corresponda. Lo invoca el «Gestor de objetos web» cuando recibe el comando de despliegue de un objeto.
- Objeto web
 - Objeto web genérico: implementa la gestión inicial de las peticiones recibidas a través del «Bus de comunicación». Comprueba en base a su Descripción de Objeto Web que exista la interacción demandada, y si necesita de permisos de acceso. En ese caso solicita al «Controlador de acceso» que compruebe si la interacción está permitida. En caso afirmativo invoca la función registrada para dicha interacción que deberá estar implementada por la «Aplicación de objeto web». Cuando finaliza la ejecución de la función, el «Objeto web genérico» se encarga a su vez de responder a la petición con el resultado de la función. Responde a través del «Bus de comunicación» al «Gestor de objetos web» que a su vez redirigirá la petición al «Servidor HTTP».

- Controlador de acceso: verifica, usando el objeto web «Pasarela de acceso», que la petición de interacción se realiza con un usuario autenticado que disponga del rol necesario para invocar la interacción en curso, y si requiere o no de reserva en curso.
- Gestor de base de datos: gestiona una base de datos para la «Aplicación de objeto web». Esta base de datos es de libre disposición de la «Aplicación de objeto web» y puede incluso no usarla si no lo requiere.
- Aplicación de objeto web: implementa las funciones que procesan cada posible interacción con el objeto web. Estas funciones son invocadas por el «Objeto web genérico» cuando recibe la petición de una interacción permitida, utilizando la función que corresponda a esa interacción. Son el núcleo de desarrollo de los objetos web, y la única parte de la que deberán preocuparse los futuros desarrolladores de objetos web para la plataforma. Podrá usar el «Gestor de base de datos» para registrar de forma no volátil cualquier tipo de información.

La figura 4.2 contiene el esquema arquitectónico del sistema con todos estos componentes.

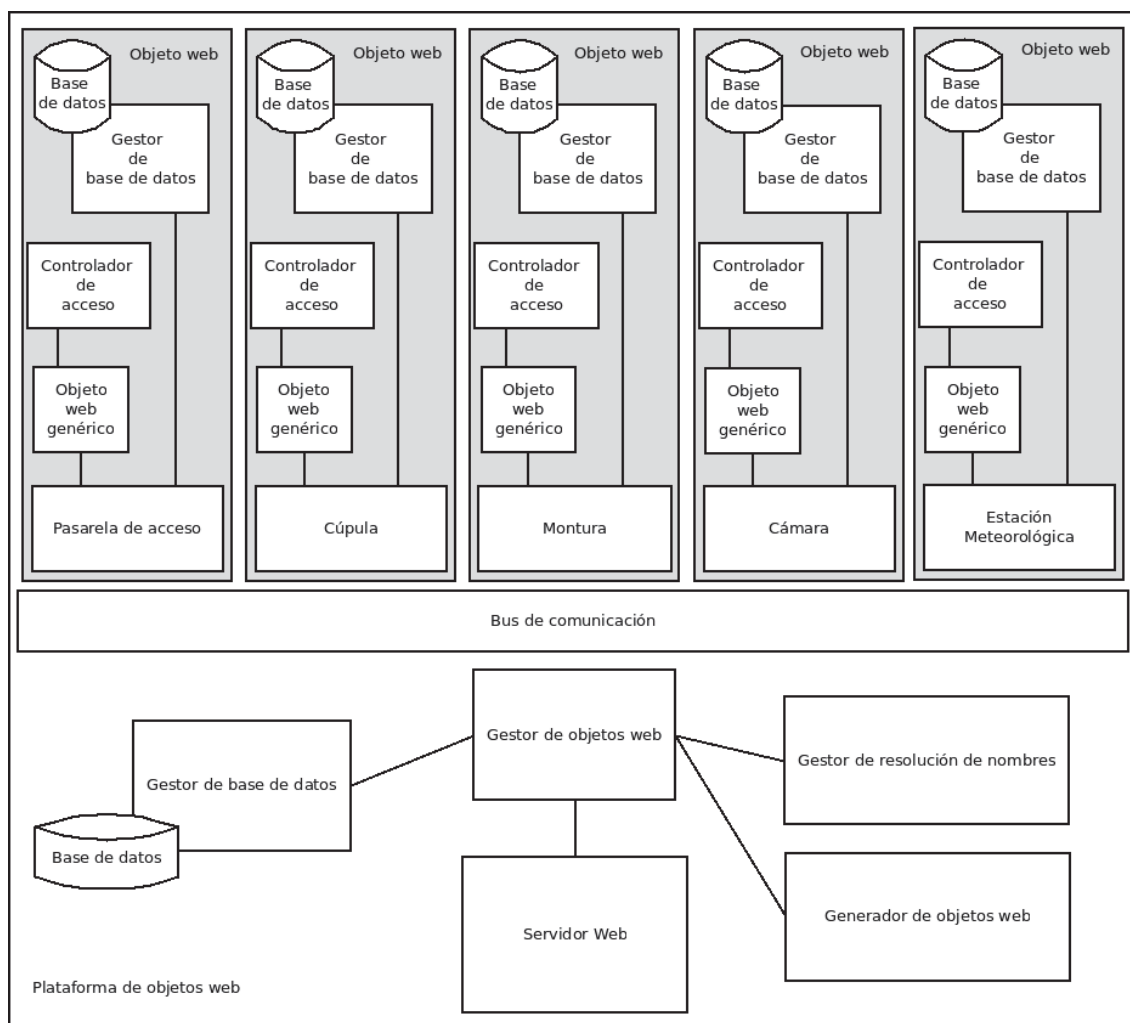


Figura 4.2: Arquitectura del sistema de control del observatorio

4.3. Base tecnológica

A la hora de seleccionar qué tecnologías utilizar para el desarrollo del sistema se proponen una serie de requisitos como punto de partida:

- Utilizar un entorno de programación multiplataforma para no condicionar a los futuros alumnos que continúen el desarrollo del mismo.
- Todas las herramientas y el código que se genere debe ser de código abierto. Forma parte del ADN del Grupo Cíclope desde sus inicios.
- El entorno de programación debe facilitar el uso de herramientas de automatización de pruebas y el uso de técnicas y procesos de integración continua.
- Por la naturaleza de los objetos web, y puesto que el sistema a desarrollar tiene como base una plataforma de objetos web, se requiere el uso de un entorno de programación dirigido por eventos.

Uno de los entornos considerados en primera instancia fue Node.js. Node.js es, tal y como se requiere, un entorno multiplataforma, de software libre y basado en eventos. Usa Javascript como lenguaje de programación y es muy utilizado actualmente para desarrollar microservicios.

Una arquitectura de software basada en microservicios se caracteriza por descomponer las funciones de una aplicación de servidor en diferentes servicios independientes de pequeño tamaño que se comuniquen unos con otros utilizando protocolos de comunicación ligeros. Esta arquitectura facilita el desarrollo en paralelo de las diferentes partes de la aplicación así como el despliegue de todo el servicio pues puede ser desplegado o actualizado sólo por partes sin afectar al resto de los componentes de la aplicación.

Sin embargo el uso de Javascript posee varios inconvenientes en opinión del autor de este proyecto. Por ejemplo que los entornos de desarrollo no ofrecen las mismas facilidades para refactorizar, probar y asegurar la calidad de código en Javascript que sí se ofrecen en otros lenguajes.

La experiencia del autor en el mundo Java motivó la búsqueda de una solución similar en concepción a Node.js pero basada en la tecnología de Java. Y tras un estudio pormenorizado de diversas opciones, incluso en otros lenguajes que no eran Java, se escogió Vert.x como la principal base tecnológica del proyecto.

Vert.x no solo soporta los requisitos descritos, sino que además ofrece otras características de mucho valor de cara a facilitar una más rápida implementación de la plataforma de objetos web diseñada.

Vert.x se define como un conjunto de herramientas para construir aplicaciones reactivas sobre la máquina virtual de Java. Se trata de un entorno de programación orientado a eventos, pero no sólo usando un bucle de eventos como Node.js, sino que tiene la capacidad de desplegar varios bucles de eventos y usar varios de los núcleos de los procesadores actuales para ejecutar código de forma concurrente.[PSV17]

Pero a su vez Vert.x garantiza que cada trozo de código que se ejecuta lo hace desde un solo hilo (salvo que el desarrollador de la aplicación lo evite). Favorece el uso de la programación asíncrona y dirigida a eventos. Y toda su funcionalidad se divide en varios módulos de forma que en un proyecto Java basta con añadir los que se utilicen y el resto dejarlos fuera.

La base de una aplicación construida con Vert.x es el uso de los denominados «Verticles». Un «Verticle» es en Vert.x la unidad mínima de despliegue de servicio. Procesa los eventos que van llegando al sistema a través del uso de un bucle de eventos. Cada bucle de eventos está asignado a un hilo de ejecución concreto. Por defecto Vert.x provee de dos bucles de eventos por hilo de ejecución correspondiente a un core de la CPU. Así cada «Verticle» siempre procesa los eventos en el mismo hilo, y se evita tener que usar mecanismos de sincronización de hilos para modificar el estado de un «Verticle».

Otras tres características importantes son que un «Verticle» puede ser desplegado dinámicamente, puede ser configurado en el momento del despliegue, y puede ser desplegado varias veces de forma simultánea.

Un ejemplo de esto es el uso de un «Verticle» como servidor HTTP. Se puede desplegar varias veces configurado de la misma manera, con el mismo puerto de entrada. En este caso cada vez que llegue una petición HTTP a dicho puerto, Vert.x pasará la petición a un único «Verticle» en base a un algoritmo basado en el algoritmo round-robin.

Y es que otra característica importante de Vert.x es el bus de eventos que despliega con el que conecta todos los «Verticles». Se trata de un bus de paso de mensajes asíncrono que soporta varios patrones de comunicación:

- Paso de mensajes punto a punto
- Paso de mensajes basado en el mecanismo de petición-respuesta
- Uso de funciones de publicación y suscripción para difundir mensajes a todos los objetos que estén escuchando

Aunque son muchísimas más las características interesantes de Vert.x, son éstas las que permiten rápidamente entender cómo puede ayudar en la implementación de la arquitectura de la plataforma de objetos web planteada.

La clave son las siguientes consideraciones:

- Un «Verticle» desplegado como objeto web ejercerá de Gestor de objetos web.
- Otro «Verticle» ejercerá de servidor HTTP.
- El bus de eventos de Vert.x será el Bus de comunicación de la arquitectura diseñada.
- El objeto web genérico será una extensión de un «Verticle» y las herramientas integradas en Vert.x facilitarán que pueda ser desplegado o replegado de forma dinámica pasándole una configuración determinada.
- Un objeto web implementado como «Verticle» podrá ser desplegado tantas veces como se desee con configuraciones diferentes.

Para la base de datos la selección resulta más sencilla. Se busca la facilidad de despliegue del sistema en su totalidad. Y que cada objeto web pueda disponer de su propia base de datos embebida. El candidato que resulta más apropiado es SQLite, que no funciona como aplicación servidor sino se basa en el uso de una librería de acceso a un archivo en el disco. Está muy optimizada para ser ligera y facilitar especialmente los procesos de lectura. Además el autor del proyecto cuenta ya con experiencia previa usando SQLite con grandes bases de datos (más de un millón de registros) y puede certificar su buen rendimiento incluso en esos casos.

Como entorno de desarrollo se escoge IntelliJ también por la buena experiencia previa del autor. Facilita todas las tareas de programación, e integra herramientas muy útiles para verificar continuamente la calidad del código desarrollado y la integración de la automatización de pruebas con el entorno.

Otro elemento tecnológico a escoger es el uso de una herramienta de construcción automática de software. Siendo un proyecto basado en Java, actualmente la opción más habitual es el uso de Gradle, posiblemente por influencia del mundo Android donde actualmente es la herramienta de construcción por defecto (frente a la segunda más habitual Maven).

Finalmente la selección más importante que resta es el gestor de proyectos a utilizar. Aquí hay dos buenos candidatos para proyectos de software libre: Github y Gitlab. Ambos son soluciones en la nube que facilitan el uso de sus servicios de forma gratuita para cualquier proyecto de software libre o código abierto. Github es más conocido y utilizado. Pero Gitlab tiene a su favor varios puntos:

- Gitlab distribuye una versión de software libre de su plataforma y puede desplegarse en un servidor propio.
- Contiene un servicio de integración continua que permite, por cada actualización hecha al repositorio de código, compilar todo el código en una máquina Docker creada adhoc para dicha compilación. Si finaliza con éxito la compilación se pueden configurar también varias fases de pruebas automáticas. Si todos los procesos terminan con éxito el software fruto de esta compilación se puede empaquetar y dejar disponible para descargar.
- Permite realizar despliegues automáticos del software generado, facilitando no sólo procesos de integración continua, sino de despliegue continuo.
- Se integra perfectamente con Gradle.

Se decide usar Gitlab y el proyecto queda disponible de forma pública en:

<https://gitlab.com/observatoriofranciscosanchez/wotgate>

En esa dirección se puede encontrar todo el material relacionado con el proyecto: código, documentación, pruebas, software generado, etc.

4.4. Pruebas

Para realizar las pruebas del sistema se ha utilizado principalmente la herramienta POSTMAN.

Es un complemento del navegador web Chrome que permite definir las interacciones con un interfaz web de programación y guardarlas en formato JSON como una colección de definiciones.

Con estas definiciones se pueden realizar las peticiones correspondientes con los parámetros de entrada que hayan sido configurados y obtener de forma gráfica la respuesta del servidor web.

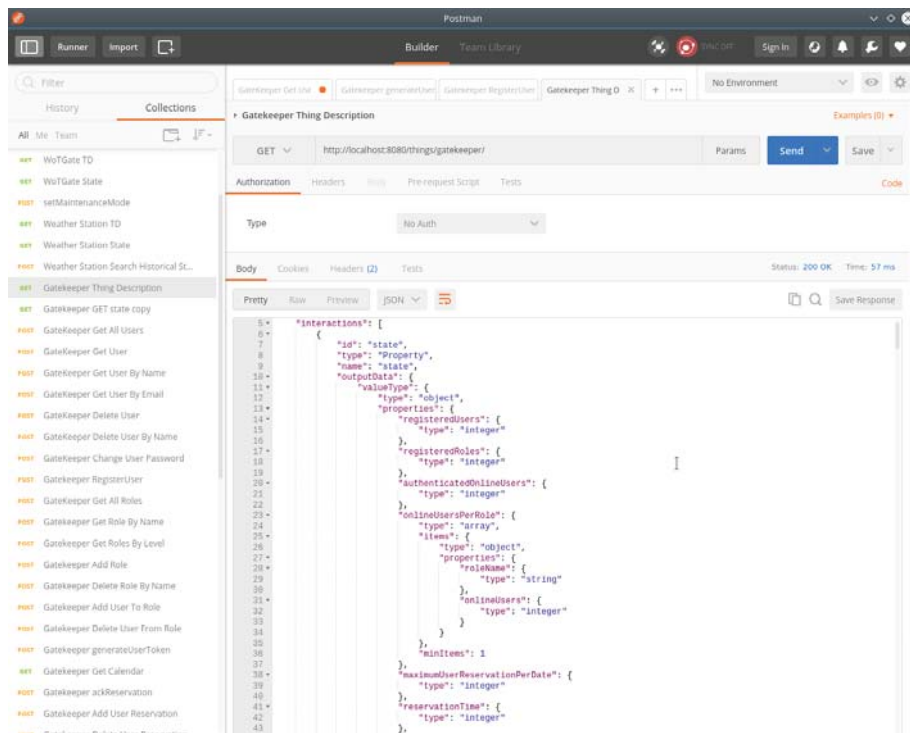


Figura 4.3: POSTMAN ejecutando interacciones con el sistema del observatorio

Usando esta herramienta se han modelado en ella todas las posibles interacciones HTTP con los objetos web a desplegar sobre la plataforma, y se han ejecutado de múltiples formas realizando pruebas unitarias y exploratorias manuales.

Estas pruebas pueden a su vez ejecutarse con el software en modo depuración con el entorno de desarrollo IntelliJ, y tras la ejecución de las pruebas se obtienen cuales han sido los niveles de cobertura de código obtenidos.

Coverage WoTGate			
49% classes, 24% lines covered in package 'wotgate'			
Element	Class, %	Method, %	Line, %
http	66% (2/3)	77% (7/9)	83% (49/59)
injector	100% (1/1)	100% (6/6)	100% (11/11)
storage	50% (2/4)	42% (30/70)	38% (97/255)
thing	42% (17/40)	27% (121/441)	18% (500/2642)
thingmanager	100% (4/4)	48% (28/58)	37% (118/317)
ErrorCode	0% (0/1)	0% (0/1)	0% (0/2)
MainVerticle	100% (1/1)	100% (10/10)	87% (35/40)
WoTGateStates	0% (0/1)	0% (0/1)	0% (0/2)

Figura 4.4: Cobertura de código durante un ciclo de pruebas con POSTMAN

De cara al futuro se ha configurado todo el proceso de integración continua en Gitlab para que cualquier envío de código al repositorio en la rama de master inicia un ciclo de compilación y pruebas que en caso de éxito genere un producto empaquetado y entregable. Tan solo se han integrado un par de pruebas que sirvan de verificación de que el proceso de integración continua esté activo.

4.5. Documentación

Durante la ejecución del proyecto se utiliza Google Docs para mantener y compartir diversos documentos relacionados con la especificación de la funcionalidad que debe cubrir el sistema y su arquitectura.

El resto de la documentación del proyecto se mantiene en la página del proyecto en Gitlab. Tanto en el propio repositorio de Git como en la wiki de Gitlab. Esta wiki es a su vez un repositorio diferente de Git donde se versiona cada página y se almacena en formato «Gitlab Flavored MARKDOWN» (GFM).

Parte de los documentos generados son:

- Memoria del proyecto
- Documento de introducción a la plataforma con las instrucciones básicas de compilación, ejecución y operación. Se mantiene en formato GFM dentro de la raíz del repositorio del proyecto, de forma que sirve como página principal del proyecto en Gitlab.
- Hoja de cálculo con una relación de toda la funcionalidad del observatorio a implementar.
- Documento de configuración de despliegue de la plataforma de objetos web usando una máquina Docker.

5. CONCLUSIONES

5.1. Resultados

Una vez finalizados todos los trabajos en el proyecto llega el momento de recapitular qué objetivos se perseguían y los resultados obtenidos.

El proyecto, enmarcado dentro de la reforma tecnológica del observatorio astronómico Francisco Sánchez, tenía como principal objetivo desarrollar una pasarela de acceso a la funcionalidad del observatorio.

Con algo más de detalle, la pasarela debía cumplir los siguientes requisitos:

- Ofrecer mecanismos de control de acceso.
- Ofrecer un sistema de reservas que habilite el uso exclusivo del observatorio por parte de un usuario durante una franja de tiempo.
- Definir e implementar el interfaz de programación web del controlador de la cúpula.
- Definir e implementar el interfaz de programación web del controlador de la montura del telescopio.
- Definir e implementar el interfaz de programación web del controlador de una cámara.
- Definir e implementar el interfaz de programación web del controlador de una estación meteorológica.
- Ser fácilmente modificable y ampliable.

El trabajo final desarrollado cubre dichos objetivos aunque no de la forma en que se pensaba al iniciar el proyecto.

Por un lado la propuesta de partida para el proyecto era construir simplemente un interfaz web RESTful que facilitara el consumo de sus servicios por parte de buscadores semánticos.

Pero rápidamente el trabajo derivó a diseñar y desarrollar toda una plataforma de despliegue de objetos web. Plataforma sobre la que se desplegarían objetos web con los que, a través del uso de sus interfaces web, sería posible teleoperar el observatorio.

De esta forma el objetivo queda cumplido. Además de ofrecer una plataforma con mucho potencial para ser reutilizada en contextos similares. Y con facilidad de ser ampliada y mejorada a través de la implementación de nuevos objetos web cuyo desarrollo es rápido y sencillo.

Por otro lado se partía del supuesto de que los distintos controladores físicos del observatorio serían desplegados a lo largo de la vida del proyecto. Pero eso nunca ocurrió y no pudo realizarse la integración real con los mismos.

Sin embargo, cada uno de los objetos web que representa un controlador físico del observatorio simula un funcionamiento simplificado del controlador. Y los interfaces

web que exponen pueden ser utilizados de igual forma que si el observatorio estuviera operativo. Por tanto se cumplen los objetivos de definición e implementación de los interfaces web de las funcionalidades del telescopio. Pero además se facilita que se puedan realizar pruebas de integración con las aplicaciones cliente en desarrollo. Y hacerlo de forma deslocalizada y sin poner en riesgo la integridad física del observatorio.

A su vez los objetos web desarrollados son la base a utilizar para los futuros objetos web que realmente se comuniquen con los controladores físicos una vez desplegados.

Como valor extra del resultado del trabajo del proyecto se puede destacar que la plataforma de objetos web desarrollada abre nuevas vías de exploración en el control y gestión de la automatización de laboratorios (como el propio observatorio), así como su uso en entornos más grandes como ciudades inteligentes o en ámbito doméstico como el hogar inteligente.

El concepto de objeto web como representación de un objeto del Internet de las cosas, o de un objeto virtual, está aún en la fase de exploración y descubrimiento, y este proyecto espera servir de ayuda a avanzar dentro de esta fase.

5.2. Futuras líneas de trabajo

Existen multitud de áreas donde podría trabajarse para mejorar e impulsar los resultados obtenidos en este proyecto.

Las mejoras más interesantes se pueden concentrar en torno a dos bloques de trabajo:

1. Mejoras como pasarela de acceso al observatorio

- Implementar los objetos web que se integren con los controladores físicos de los elementos del observatorio.
- Desarrollar objetos web que representen experimentos astronómicos concretos y abstraigan al usuario del manejo de bajo nivel del observatorio.
- Evolucionar el sistema de reservas para que una reserva sea sobre un objeto concreto (por ejemplo un objeto web representando un experimento) y no sobre todos los objetos web desplegados.
- Integrar el mecanismo de alarmas utilizado en el diseño del observatorio para que todos los dispositivos conozcan la situación del mismo y actúen en caso de situación anómala.

2. Mejoras como plataforma de objetos web:

- Evolucionar el mecanismo de autenticación a Json Web Token para fortalecer la seguridad de la plataforma.

- Habilitar que los propios objetos web puedan ser usuarios registrados de la pasarela de acceso y puedan invocar las interacciones de otros objetos web con un acceso autenticado.
- Implementar el concepto de concentrador de objetos (objeto web que genera objetos virtuales y controla todas las interacciones con cualquiera de ellos). Así, para controlar mil sensores iguales no haría falta desplegar mil objetos web (uno por cada sensor). Sino solo un concentrador que controlaría los mil sensores y de cara al exterior de la plataforma simularía la existencia de los mil objetos web.
- Evolucionar la plataforma para que, aprovechando las capacidades de Vert.x, pueda funcionar como una plataforma distribuida de objetos web.
- Utilizar la capacidad de Vert.x de integrarse con múltiples lenguajes de programación para poder programar los objetos web en otros lenguajes aparte de Java. Así podrá facilitarse la programación más ligera y sencilla de los objetos web utilizando lenguajes como Kotlin.

BIBLIOGRAFÍA

Referencias

- [Blo08] Joshua Bloch. *Effective Java (Second Edition)*. Addison-Wesley, 2008.
- [COL05] COLDEX. Coldex - collaborative learning and distributed experimentation, Octubre 2005. URL: <http://www.coldex.info/>.
- [Com17] SQLite Community. Sqlite documentation, Agosto 2017. URL: <https://sqlite.org/docs.html>.
- [Dor16] Sergio Rodríguez Dorado. Desarrollo de una aplicación móvil para la teleoperación de los telescopios robóticos de la red gloria. Master's thesis, Universidad Politécnica de Madrid, 2016.
- [Esc17] Clement Escoffier. *Building Reactive Microservices in Java*. O'Reilly Media, 2017.
- [Fie00] Roy Thomas Fielding. Architectural styles and the design of network-based software architectures, 2000. URL: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.
- [Fie08] Roy T. Fielding. Rest apis must be hypertext-driven, Octubre 2008. URL: <http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>.
- [Fow10] Martin Fowler. Richardson maturity model steps toward the glory of rest, Marzo 2010. URL: <https://martinfowler.com/articles/richardsonMaturityModel.html>.
- [FR14a] R. Fielding and J. Reschke. Hypertext transfer protocol (http/1.1): Authentication, Junio 2014. URL: <https://tools.ietf.org/html/rfc7235>.
- [FR14b] R. Fielding and J. Reschke. Hypertext transfer protocol (http/1.1): Conditional requests, Junio 2014. URL: <https://tools.ietf.org/html/rfc7232>.
- [FR14c] R. Fielding and J. Reschke. Hypertext transfer protocol (http/1.1): Message syntax and routing, Junio 2014. URL: <https://tools.ietf.org/html/rfc7230>.
- [FR14d] R. Fielding and J. Reschke. Hypertext transfer protocol (http/1.1): Range requests, Junio 2014. URL: <https://tools.ietf.org/html/rfc7233>.

- [FR14e] R. Fielding and J. Reschke. Hypertext transfer protocol (http/1.1): Semantics and content, Junio 2014. URL: <https://tools.ietf.org/html/rfc7231>.
- [Gar07] Roberto García. Ciclope astro [astrofotografía], Septiembre 2007. URL: http://laurel.datsi.fi.upm.es/~ssoo/DAW/Trabajos/2006-2007/Septiembre/005/func_es.
- [GLO17] GLORIA. Gloria project, 2017. URL: <http://gloria-project.eu/es/>.
- [GT16a] Dominique D. Guinard and Vlad M. Trifa. *Building the Web of Things*. Manning Publications, 2016.
- [GT16b] Dominique D. Guinard and Vlad M. Trifa. *Using the Web to Build the IoT*. Manning Publications, 2016.
- [Leó09] Raquel Cedazo León. *Metodología de aprendizaje y generación de conocimiento mediante colaboración vía Internet: Aplicación al Observatorio Astronómico Montegancedo*. PhD thesis, Universidad Politécnica de Madrid, 2009.
- [PSV17] Julien Ponge, Thomas Segismont, and Jullie Viet. *A gentle guide to asynchronous programming with Eclipse Vert.x for Java developers*. Vert.x, 2017.
- [San16] Pablo Lozano Santiuste. Diseño de un controlador para la cúpula del observatorio astronómico montegancedo. Master's thesis, Universidad Politécnica de Madrid, 2016.
- [Thi15] Joshua Thijssen. Rest cookbook how to do stuff restul, Agosto 2015. URL: <http://restcookbook.com/>.
- [W3C14] W3C. Json-ld 1.0 a json-based serialization for linked data, Enero 2014. URL: <https://www.w3.org/TR/json-ld/#interpreting-json-as-json-ld>.
- [W3C15] W3C. Web thing model, Agosto 2015. URL: <https://www.w3.org/Submission/2015/SUBM-wot-model-20150824/>.
- [W3C17a] W3C. Web of things (wot) architecture, Agosto 2017. URL: <https://w3c.github.io/wot-architecture/>.
- [W3C17b] W3C. Web of things (wot) scripting api, Agosto 2017. URL: <https://w3c.github.io/wot-scripting-api/>.
- [W3C17c] W3C. Web of things (wot) thing description, Agosto 2017. URL: <https://w3c.github.io/wot-thing-description/>.

- [WA17] A. Wright and H. Andrews. Json schema: A media type for describing json documents, Abril 2017. URL: <http://json-schema.org/latest/json-schema-core.html>.
- [WAL17] A. Wright, H. Andrews, and G. Luff. Json hyper-schema: A vocabulary for hypermedia annotation of json, Abril 2017. URL: <http://json-schema.org/latest/json-schema-hypermedia.html>.
- [WLA17] A. Wright, G. Luff, and H. Andrews. Json schema validation: A vocabulary for structural validation of json, Abril 2017. URL: <http://json-schema.org/latest/json-schema-validation.html>.

ANEXOS

A. Definición de los Objetos Web

A.1. Pasarela de acceso

```
{
  "type": "Thing",
  "name": "gatekeeper",
  "base": "http://wotgate.om.fi.upm.es/things/gatekeeper/",
  "gatekeeper": "http://wotgate.om.fi.upm.es/things/gatekeeper/",
  "roleBasedWritingAccessControl": "Administrator",
  "interactions": [
    {
      "type": "Property",
      "name": "state",
      "outputData": {
        "type": "object",
        "fields": {
          "name": {
            "type": "string"
          },
          "registeredUsers": {
            "type": "integer"
          },
          "registeredRoles": {
            "type": "integer"
          },
          "registeredUsersOnline": {
            "type": "integer"
          },
          "reservationOngoing": {
            "type": "boolean"
          },
          "status": {
            "enum": [
              "UNRESTRICTED",
              "RESTRICTED",
              "MAINTENANCE",
              "MALFUNCTION"
            ]
          },
          "timestamp": {
            "type": "string"
          }
        }
      },
      "writable": false,
      "link": [
        {
          "href": "state",
          "mediaType": "application/json"
        }
      ]
    },
    {
      "type": "Action",
      "name": "getAllRoles",
      "asynchronous": "false",
      "roleBasedAccessControl": "Administrator",
      "outputData": {
        "type": "array",
        "items": {

```

```

    "type": "object",
    "fields": {
      "name": {
        "type": "string"
      },
      "level": {
        "type": "integer"
      },
      "userNames": {
        "type": "array",
        "items": {
          "type": "string"
        }
      },
      "dateCreated": {
        "type": "string"
      }
    }
  },
  "minItems": 0
},
"link": [
  {
    "href": "getAllRoles",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",
  "name": "getRoleByName",
  "asynchronous": "false",
  "roleBasedAccessControl": "Administrator",
  "inputData": {
    "type": "object",
    "fields": {
      "name": {
        "type": "string"
      }
    }
  },
  "required": [
    "name"
  ]
},
"outputData": {
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "level": {
      "type": "integer"
    },
    "userNames": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "dateCreated": {
      "type": "string"
    }
  }
}
},
"link": [
  {
    "href": "getRoleByName",

```

```

        "mediaType": "application/json"
    }
}
},
{
    "type": "Action",
    "name": "getRolesByLevel",
    "asynchronous": "false",
    "roleBasedAccessControl": "Administrator",
    "inputData": {
        "type": "object",
        "properties": {
            "level": {
                "type": "integer"
            }
        }
    },
    "required": [
        "level"
    ]
},
"outputData": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "name": {
                "type": "string"
            },
            "level": {
                "type": "integer"
            },
            "userNames": {
                "type": "array",
                "items": {
                    "type": "string"
                }
            }
        }
    },
    "dateCreated": {
        "type": "string"
    }
},
    "minItems": 0
}
},
"link": [
    {
        "href": "getRolesByLevel",
        "mediaType": "application/json"
    }
]
},
{
    "type": "Action",
    "name": "addRole",
    "asynchronous": "false",
    "roleBasedAccessControl": "Administrator",
    "inputData": {
        "type": "object",
        "properties": {
            "name": {
                "type": "string"
            },
            "level": {
                "type": "integer"
            }
        }
    },
    "required": [
        "name",

```

```

        "level"
      ]
    }
  },
  "link": [
    {
      "href": "addRole",
      "mediaType": "application/json"
    }
  ]
},
{
  "type": "Action",
  "name": "addUserToRole",
  "asynchronous": "false",
  "roleBasedAccessControl": "Administrator",
  "inputData": {
    "type": "object",
    "properties": {
      "userName": {
        "type": "string"
      },
      "roleName": {
        "type": "string"
      }
    }
  },
  "required": [
    "userName",
    "roleName"
  ]
},
"link": [
  {
    "href": "addUserToRole",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",
  "name": "deleteUserFromRole",
  "asynchronous": "false",
  "roleBasedAccessControl": "Administrator",
  "inputData": {
    "type": "object",
    "properties": {
      "userName": {
        "type": "string"
      },
      "roleName": {
        "type": "string"
      }
    }
  },
  "required": [
    "userName",
    "roleName"
  ]
},
"link": [
  {
    "href": "deleteUserFromRole",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",

```

```

    "name": "deleteRoleByName",
    "asynchronous": "false",
    "roleBasedAccessControl": "Administrator",
    "inputData": {
      "type": "object",
      "properties": {
        "name": {
          "type": "string"
        }
      },
      "required": [
        "name"
      ]
    },
    "link": [
      {
        "href": "deleteRoleByName",
        "mediaType": "application/json"
      }
    ]
  },
  {
    "type": "Action",
    "name": "getAllUsers",
    "asynchronous": "false",
    "roleBasedAccessControl": "Administrator",
    "outputData": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "email": {
            "type": "string"
          },
          "roleNames": {
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "validated": {
            "type": "boolean"
          },
          "online": {
            "type": "boolean"
          },
          "dateCreated": {
            "type": "string"
          },
          "dateModified": {
            "type": "string"
          }
        }
      },
      "minItems": 0
    }
  },
  {
    "link": [
      {
        "href": "getAllUsers",
        "mediaType": "application/json"
      }
    ]
  },
  {

```

```

"type": "Action",
"name": "getUser",
"asynchronous": "false",
"roleBasedAccessControl": "Authenticated",
"outputData": {
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "email": {
      "type": "string"
    },
    "roleNames": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "dateCreated": {
      "type": "string"
    },
    "dateModified": {
      "type": "string"
    }
  }
},
"link": [
  {
    "href": "getUser",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",
  "name": "getUserByName",
  "asynchronous": "false",
  "roleBasedAccessControl": "Administrator",
  "inputData": {
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      }
    }
  },
  "required": [
    "name"
  ]
},
"outputData": {
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "email": {
      "type": "string"
    },
    "roleNames": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  },
  "validated": {
    "type": "boolean"
  }
}

```

```

    },
    "online": {
      "type": "boolean"
    },
    "dateCreated": {
      "type": "string"
    },
    "dateModified": {
      "type": "string"
    }
  }
},
"link": [
  {
    "href": "getUserByName",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",
  "name": "getUserByEmail",
  "asynchronous": "false",
  "roleBasedAccessControl": "Administrator",
  "inputData": {
    "type": "object",
    "properties": {
      "email": {
        "type": "string"
      }
    }
  },
  "required": [
    "email"
  ]
},
"outputData": {
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "email": {
      "type": "string"
    },
    "roleNames": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  },
  "validated": {
    "type": "boolean"
  },
  "online": {
    "type": "boolean"
  },
  "dateCreated": {
    "type": "string"
  },
  "dateModified": {
    "type": "string"
  }
}
},
"link": [
  {
    "href": "getUserByEmail",

```

```

        "mediaType": "application/json"
    }
  ],
},
{
  "type": "Action",
  "name": "deleteUser",
  "asynchronous": "false",
  "roleBasedAccessControl": "Authenticated",
  "link": [
    {
      "href": "deleteUser",
      "mediaType": "application/json"
    }
  ]
},
{
  "type": "Action",
  "name": "deleteUserByName",
  "asynchronous": "false",
  "roleBasedAccessControl": "Administrator",
  "inputData": {
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      }
    }
  },
  "required": [
    "name"
  ]
},
{
  "link": [
    {
      "href": "deleteUserByName",
      "mediaType": "application/json"
    }
  ]
},
{
  "type": "Action",
  "name": "changeUserPassword",
  "asynchronous": "false",
  "roleBasedAccessControl": "Authenticated",
  "inputData": {
    "type": "object",
    "properties": {
      "password": {
        "type": "string"
      }
    }
  },
  "required": [
    "password"
  ]
},
{
  "link": [
    {
      "href": "changeUserPassword",
      "mediaType": "application/json"
    }
  ]
},
{
  "type": "Action",
  "name": "registerUser",
  "asynchronous": "false",
  "inputData": {

```



```

    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      },
      "password": {
        "type": "string"
      },
      "email": {
        "type": "string"
      }
    },
    "required": [
      "name",
      "password",
      "email"
    ]
  },
  "link": [
    {
      "href": "registerUser",
      "mediaType": "application/json"
    }
  ]
},
{
  "type": "Action",
  "name": "confirmUserRegistration",
  "asynchronous": "false",
  "inputData": {
    "type": "object",
    "properties": {
      "token": {
        "type": "string"
      }
    }
  },
  "required": [
    "token"
  ]
},
"link": [
  {
    "href": "confirmUserRegistration",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",
  "name": "recoverUserPassword",
  "asynchronous": "false",
  "inputData": {
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      },
      "email": {
        "type": "string"
      }
    }
  },
  "required": [
    "name",
    "email"
  ]
},
"link": [

```

```

    {
      "href": "recoverUserPassword",
      "mediaType": "application/json"
    }
  ]
},
{
  "type": "Action",
  "name": "confirmPasswordRecovery",
  "asynchronous": "false",
  "inputData": {
    "type": "object",
    "properties": {
      "token": {
        "type": "string"
      }
    }
  },
  "required": [
    "token"
  ]
},
"link": [
  {
    "href": "confirmPasswordRecovery",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",
  "name": "generateUserToken",
  "asynchronous": "false",
  "inputData": {
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      },
      "password": {
        "type": "string"
      }
    }
  },
  "required": [
    "name",
    "password"
  ]
},
"outputData": {
  "type": "object",
  "properties": {
    "token": {
      "type": "string"
    },
    "expirationTime": {
      "type": "string"
    }
  }
}
},
"link": [
  {
    "href": "generateUserToken",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",

```

```

"name": "revokeUserToken",
"asynchronous": "false",
"roleBasedAccessControl": "Authenticated",
"link": [
  {
    "href": "revokeUserToken",
    "mediaType": "application/json"
  }
]
},
{
"type": "Action",
"name": "getUserPermissions",
"asynchronous": "false",
"outputData": {
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "roleNames": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "reservationOngoing": {
      "type": "boolean"
    }
  }
},
"link": [
  {
    "href": "generateUserToken",
    "mediaType": "application/json"
  }
]
},
{
"type": "Action",
"name": "getAllReservationsByDate",
"asynchronous": "false",
"inputData": {
  "type": "object",
  "properties": {
    "startDate": {
      "type": "string"
    }
  }
},
"required": [
  "startDate"
]
},
"outputData": {
  "type": "object",
  "properties": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "startDate": {
          "type": "string"
        },
        "endDate": {
          "type": "string"
        },
        "userName": {

```

```

        "type": "string"
      },
      "dateCreated": {
        "type": "string"
      }
    },
    "minItems": 0
  }
}
},
"link": [
  {
    "href": "getAllReservationsByDate",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",
  "name": "getAllReservations",
  "asynchronous": "false",
  "outputData": {
    "type": "object",
    "properties": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "startDate": {
            "type": "string"
          },
          "endDate": {
            "type": "string"
          },
          "userName": {
            "type": "string"
          },
          "dateCreated": {
            "type": "string"
          }
        }
      }
    },
    "minItems": 0
  }
}
},
"link": [
  {
    "href": "getAllReservations",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",
  "name": "getUserReservationsByDate",
  "asynchronous": "false",
  "inputData": {
    "type": "object",
    "properties": {
      "startDate": {
        "type": "string"
      }
    }
  },
  "required": [
    "startDate"
  ]
}
},

```

```

"outputData": {
  "type": "object",
  "properties": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "startDate": {
          "type": "string"
        },
        "endDate": {
          "type": "string"
        },
        "userName": {
          "type": "string"
        },
        "dateCreated": {
          "type": "string"
        }
      }
    },
    "minItems": 0
  }
},
"link": [
  {
    "href": "getUserReservationsByDate",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",
  "name": "getUserReservationsByNameAndDate",
  "asynchronous": "false",
  "inputData": {
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      },
      "startDate": {
        "type": "string"
      }
    }
  },
  "required": [
    "name",
    "startDate"
  ]
},
"outputData": {
  "type": "object",
  "properties": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "startDate": {
          "type": "string"
        },
        "endDate": {
          "type": "string"
        },
        "userName": {
          "type": "string"
        },
        "dateCreated": {

```

```

        "type": "string"
      }
    },
    "minItems": 0
  }
}
},
"link": [
  {
    "href": "getUserReservationsByNameAndDate",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",
  "name": "getAllUserReservations",
  "asynchronous": "false",
  "outputData": {
    "type": "object",
    "properties": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "startDate": {
            "type": "string"
          },
          "endDate": {
            "type": "string"
          },
          "userName": {
            "type": "string"
          },
          "dateCreated": {
            "type": "string"
          }
        }
      },
      "minItems": 0
    }
  },
  "link": [
    {
      "href": "getAllUserReservations",
      "mediaType": "application/json"
    }
  ]
},
{
  "type": "Action",
  "name": "getAllUserReservationsByName",
  "asynchronous": "false",
  "inputData": {
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      }
    }
  },
  "required": [
    "name"
  ]
},
"outputData": {
  "type": "object",
  "properties": {

```

```

        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "startDate": {
              "type": "string"
            },
            "endDate": {
              "type": "string"
            },
            "dateCreated": {
              "type": "string"
            }
          }
        },
        "minItems": 0
      }
    },
    "link": [
      {
        "href": "getAllUserReservationsByName",
        "mediaType": "application/json"
      }
    ]
  },
  {
    "type": "Action",
    "name": "addUserReservation",
    "asynchronous": "false",
    "roleBasedAccessControl": "Authenticated",
    "inputData": {
      "type": "object",
      "properties": {
        "startDate": {
          "type": "string"
        }
      }
    },
    "required": [
      "startDate"
    ]
  },
  "link": [
    {
      "href": "addUserReservation",
      "mediaType": "application/json"
    }
  ]
},
{
  "type": "Action",
  "name": "deleteUserReservation",
  "asynchronous": "false",
  "roleBasedAccessControl": "Authenticated",
  "inputData": {
    "type": "object",
    "properties": {
      "startDate": {
        "type": "string"
      }
    }
  },
  "required": [
    "startDate"
  ]
},
"link": [
  {
    "href": "deleteUserReservation",

```

```

        "mediaType": "application/json"
    }
  ],
  {
    "type": "Action",
    "name": "getOngoingReservation",
    "asynchronous": "false",
    "outputData": {
      "type": "object",
      "properties": {
        "startDate": {
          "type": "string"
        },
        "endDate": {
          "type": "string"
        },
        "userName": {
          "type": "string"
        },
        "dateCreated": {
          "type": "string"
        }
      }
    }
  },
  "link": [
    {
      "href": "getOngoingReservation",
      "mediaType": "application/json"
    }
  ]
},
{
  "type": "Action",
  "name": "setUnrestrictedMode",
  "asynchronous": "false",
  "roleBasedAccessControl": "Administrator",
  "link": [
    {
      "href": "setUnrestrictedMode",
      "mediaType": "application/json"
    }
  ]
},
{
  "type": "Action",
  "name": "setRestrictedMode",
  "asynchronous": "false",
  "roleBasedAccessControl": "Administrator",
  "link": [
    {
      "href": "setRestrictedMode",
      "mediaType": "application/json"
    }
  ]
},
{
  "type": "Action",
  "name": "setMaintenanceMode",
  "asynchronous": "false",
  "roleBasedAccessControl": "Administrator",
  "link": [
    {
      "href": "setMaintenanceMode",
      "mediaType": "application/json"
    }
  ]
}
]

```



```

    },
    {
      "type": "Action",
      "name": "doEmergencyStop",
      "asynchronous": "false",
      "roleBasedAccessControl": "Administrator",
      "link": [
        {
          "href": "doEmergencyStop",
          "mediaType": "application/json"
        }
      ]
    }
  ]
}

```

A.2. Cúpula

```

{
  "type": "Thing",
  "name": "dome",
  "base": "http://wotgate.om.fi.upm.es/things/dome/",
  "gatekeeper": "http://wotgate.om.fi.upm.es/things/gatekeeper/",
  "roleBasedWritingAccessControl": "Administrator",
  "interactions": [
    {
      "type": "Property",
      "name": "state",
      "outputData": {
        "type": "object",
        "fields": {
          "name": {
            "type": "string"
          }
        },
        "observatoryName": {
          "type": "string"
        },
        "observatoryLocation": {
          "type": "object",
          "fields": {
            "altitude": {
              "type": "number"
            },
            "longitude": {
              "type": "number"
            }
          }
        }
      }
    },
    {
      "windowStatus": {
        "status": {
          "enum": [
            "OPEN",
            "CLOSED"
          ]
        }
      }
    },
    {
      "position": {
        "type": "object",
        "fields": {
          "altitude": {
            "type": "number"
          },
          "azimuth": {
            "type": "number"
          },
          "label": {
            "enum": [

```

```

        "HOME",
        "PARKING"
    ]
}
},
"onTracking": {
    "type": "boolean"
},
"operatingStatus": {
    "enum": [
        "OK",
        "ERROR",
        "UNKNOWN"
    ]
},
"homePosition": {
    "type": "object",
    "fields": {
        "azimuth": {
            "type": "number"
        }
    }
},
"parkingPosition": {
    "type": "object",
    "fields": {
        "azimuth": {
            "type": "number"
        }
    }
},
"timestamp": {
    "type": "string"
}
}
},
"writable": false,
"link": [
    {
        "href": "state",
        "mediaType": "application/json"
    }
]
},
{
    "type": "Action",
    "name": "setParkingPosition",
    "roleBasedAccessControl": "Administrator",
    "reservable": "true",
    "inputData": {
        "type": "object",
        "fields": {
            "azimuth": {
                "type": "number"
            }
        }
    },
    "required": [
        "azimuth"
    ]
},
"link": [
    {
        "href": "setParkingPosition",
        "mediaType": "application/json"
    }
]
}
]

```

```

    },
    {
      "type": "Action",
      "name": "openWindow",
      "roleBasedAccessControl": "Administrator",
      "reservable": "true",
      "link": [
        {
          "href": "openWindow",
          "mediaType": "application/json"
        }
      ]
    }
  ],
  {
    "type": "Action",
    "name": "closeWindow",
    "roleBasedAccessControl": "Administrator",
    "reservable": "true",
    "link": [
      {
        "href": "closeWindow",
        "mediaType": "application/json"
      }
    ]
  }
],
  {
    "type": "Action",
    "name": "goHome",
    "roleBasedAccessControl": "Administrator",
    "reservable": "true",
    "link": [
      {
        "href": "goHome",
        "mediaType": "application/json"
      }
    ]
  }
],
  {
    "type": "Action",
    "name": "park",
    "roleBasedAccessControl": "Administrator",
    "reservable": "true",
    "link": [
      {
        "href": "park",
        "mediaType": "application/json"
      }
    ]
  }
],
  {
    "type": "Action",
    "name": "goAzimuth",
    "roleBasedAccessControl": "Authenticated",
    "reservable": "true",
    "inputData": {
      "type": "object",
      "fields": {
        "azimuth": {
          "type": "number"
        }
      }
    },
    "required": [
      "azimuth"
    ]
  }
],
  "link": [
    {

```

```

        "href": "goAzimuth",
        "mediaType": "application/json"
    }
]
},
{
    "type": "Action",
    "name": "activateTracking",
    "roleBasedAccessControl": "Authenticated",
    "reservable": "true",
    "link": [
        {
            "href": "activateTracking",
            "mediaType": "application/json"
        }
    ]
},
{
    "type": "Action",
    "name": "deactivateTracking",
    "roleBasedAccessControl": "Authenticated",
    "reservable": "true",
    "link": [
        {
            "href": "deactivateTracking",
            "mediaType": "application/json"
        }
    ]
}
]
}
}

```

A.3. Montura

```

{
    "type": "Thing",
    "name": "mount",
    "base": "http://wotgate.om.fi.upm.es/things/mount/",
    "gatekeeper": "http://wotgate.om.fi.upm.es/things/gatekeeper/",
    "roleBasedWritingAccessControl": "Administrator",
    "interactions": [
        {
            "type": "Property",
            "name": "state",
            "outputData": {
                "type": "object",
                "fields": {
                    "name": {
                        "type": "string"
                    },
                    "onMonitoring": {
                        "type": "boolean"
                    },
                    "monitoringInterval": {
                        "type": "integer"
                    },
                    "position": {
                        "ra": {
                            "type": "number"
                        },
                        "dec": {
                            "type": "number"
                        },
                        "altitude": {
                            "type": "number"
                        },
                        "azimuth": {

```

```

        "type": "number"
      }
    },
    "operatingStatus": {
      "enum": [
        "OK",
        "ERROR",
        "UNKNOWN"
      ]
    },
    "timestamp": {
      "type": "string"
    }
  }
},
"writable": false,
"link": [
  {
    "href": "state",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",
  "name": "enableMonitoring",
  "roleBasedAccessControl": "Authenticated",
  "reservable": "true",
  "inputData": {
    "type": "object",
    "fields": {
      "monitoringInterval": {
        "type": "integer"
      }
    }
  },
  "required": [
    "monitoringInterval"
  ]
},
"link": [
  {
    "href": "enableMonitoring",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",
  "name": "disableMonitoring",
  "roleBasedAccessControl": "Authenticated",
  "reservable": "true",
  "link": [
    {
      "href": "disableMonitoring",
      "mediaType": "application/json"
    }
  ]
},
{
  "type": "Action",
  "name": "moveByTime",
  "roleBasedAccessControl": "Authenticated",
  "reservable": "true",
  "inputData": {
    "type": "object",
    "fields": {
      "direction": {

```

```

        "enum": [
            "NORTH",
            "SOUTH",
            "EAST",
            "WEST"
        ]
    },
    "timer": {
        "type": "integer"
    }
},
"required": [
    "direction", "timer"
]
},
"link": [
    {
        "href": "moveByTime",
        "mediaType": "application/json"
    }
]
},
{
    "type": "Action",
    "name": "moveByAltazimuthCoordinates",
    "roleBasedAccessControl": "Authenticated",
    "reservable": "true",
    "inputData": {
        "type": "object",
        "fields": {
            "altitude": {
                "type": "number"
            },
            "azimuth": {
                "type": "number"
            }
        }
    },
    "required": [
        "altitude", "azimuth"
    ]
},
"link": [
    {
        "href": "moveByAltazimuthCoordinates",
        "mediaType": "application/json"
    }
]
},
{
    "type": "Action",
    "name": "moveByEquatorialCoordinates",
    "roleBasedAccessControl": "Authenticated",
    "reservable": "true",
    "inputData": {
        "type": "object",
        "fields": {
            "ra": {
                "type": "number"
            },
            "dec": {
                "type": "number"
            }
        }
    },
    "required": [
        "ra", "dec"
    ]
}
},

```

```

    "link": [
      {
        "href": "moveByEquatorialCoordinates",
        "mediaType": "application/json"
      }
    ]
  }
]
}

```

A.4. Cámara

```

{
  "type": "Thing",
  "name": "camera",
  "base": "http://wotgate.om.fi.upm.es/things/camera/",
  "gatekeeper": "http://wotgate.om.fi.upm.es/things/gatekeeper/",
  "roleBasedWritingAccessControl": "Administrator",
  "interactions": [
    {
      "type": "Property",
      "name": "state",
      "outputData": {
        "type": "object",
        "fields": {
          "name": {
            "type": "string"
          },
          "operatingStatus": {
            "enum": [
              "OK",
              "WORKING",
              "ERROR",
              "UNKNOWN"
            ]
          },
          "numberOfStoredCaptures": {
            "type": "integer"
          },
          "timestamp": {
            "type": "string"
          }
        }
      },
      "writable": false,
      "link": [
        {
          "href": "state",
          "mediaType": "application/json"
        }
      ]
    },
    {
      "type": "Action",
      "name": "captureImage",
      "roleBasedAccessControl": "Authenticated",
      "reservable": "true",
      "inputData": {
        "type": "object",
        "fields": {
          "exposureTime": {
            "type": "integer"
          },
          "brightness": {
            "type": "integer"
          },
          "gain": {

```

```

        "type": "integer"
    },
    "quality": {
        "type": "integer"
    }
},
"required": [
    "exposureTime",
    "brightness",
    "gain",
    "quality"
]
},
"outputData": {
    "type": "object",
    "fields": {
        "id": {
            "type": "string"
        },
        "href": {
            "type": "string"
        }
    }
},
"link": [
    {
        "href": "captureImage",
        "mediaType": "application/json"
    }
]
},
{
    "type": "Action",
    "name": "cancelCaptureImage",
    "roleBasedAccessControl": "Authenticated",
    "reservable": "true",
    "link": [
        {
            "href": "cancelCaptureImage",
            "mediaType": "application/json"
        }
    ]
},
{
    "type": "Action",
    "name": "getAllStoredCaptures",
    "asynchronous": "false",
    "outputData": {
        "type": "array",
        "items": {
            "type": "object",
            "fields": {
                "id": {
                    "type": "string"
                },
                "name": {
                    "type": "string"
                },
                "description": {
                    "type": "string"
                },
                "href": {
                    "type": "string"
                },
                "exposureTime": {
                    "type": "integer"
                }
            }
        }
    }
}

```



```

        "brightness": {
            "type": "integer"
        },
        "gain": {
            "type": "integer"
        },
        "quality": {
            "type": "integer"
        },
        "userName": {
            "type": "string"
        },
        "dateCreated": {
            "type": "string"
        },
        "dateModified": {
            "type": "string"
        }
    }
},
"minItems": 0
},
"link": [
    {
        "href": "getAllStoredCaptures",
        "mediaType": "application/json"
    }
]
},
{
    "type": "Action",
    "name": "getStoredCapture",
    "asynchronous": "false",
    "inputData": {
        "type": "object",
        "fields": {
            "id": {
                "type": "string"
            }
        }
    },
    "required": [
        "id"
    ]
},
"outputData": {
    "type": "object",
    "fields": {
        "id": {
            "type": "string"
        },
        "name": {
            "type": "string"
        },
        "description": {
            "type": "string"
        },
        "href": {
            "type": "string"
        },
        "exposureTime": {
            "type": "integer"
        },
        "brightness": {
            "type": "integer"
        },
        "gain": {
            "type": "integer"
        }
    }
}

```

```

    },
    "quality": {
      "type": "integer"
    },
    "userName": {
      "type": "string"
    },
    "dateCreated": {
      "type": "string"
    },
    "dateModified": {
      "type": "string"
    }
  }
},
"link": [
  {
    "href": "getStoredCapture",
    "mediaType": "application/json"
  }
]
},
{
  "type": "Action",
  "name": "updateStoredCaptureInformation",
  "roleBasedAccessControl": "Authenticated",
  "asynchronous": "false",
  "inputData": {
    "type": "object",
    "fields": {
      "id": {
        "type": "string"
      },
      "name": {
        "type": "string"
      },
      "description": {
        "type": "string"
      }
    }
  },
  "required": [
    "id",
    "name",
    "description"
  ]
},
"link": [
  {
    "href": "updateStoredCaptureInformation",
    "mediaType": "application/json"
  }
]
}
]
}
}

```

A.5. Estación meteorológica

```

{
  "type": "Thing",
  "name": "weatherstation",
  "base": "http://wotgate.om.fi.upm.es/things/weatherstation/",
  "gatekeeper": "http://wotgate.om.fi.upm.es/things/gatekeeper/",
  "roleBasedWritingAccessControl": "Administrator",
  "interactions": [
    {
      "type": "Property",

```

```

"name": "state",
"outputData": {
  "type": "object",
  "fields": {
    "name": {
      "type": "string"
    },
    "location": {
      "type": "object",
      "fields": {
        "altitude": {
          "type": "number"
        },
        "longitude": {
          "type": "number"
        }
      }
    },
    "temperature": {
      "type": "number"
    },
    "pressure": {
      "type": "number"
    },
    "humidity": {
      "type": "number"
    },
    "rainfall": {
      "type": "number"
    },
    "windSpeed": {
      "type": "number"
    },
    "windDirection": {
      "enum": [
        "N",
        "NE",
        "E",
        "SE",
        "S",
        "SW",
        "W",
        "NW"
      ]
    },
    "operatingStatus": {
      "enum": [
        "OK",
        "ERROR",
        "UNKNOWN"
      ]
    },
    "timestamp": {
      "type": "string"
    }
  }
},
"writable": false,
"stability": 300000,
"link": [
  {
    "href": "state",
    "mediaType": "application/json"
  }
]
},
{

```

```

"type": "Action",
"name": "getHistoricalState",
"asynchronous": "false",
"outputData": {
  "type": "array",
  "items": {
    "type": "object",
    "fields": {
      "temperature": {
        "type": "number"
      },
      "pressure": {
        "type": "number"
      },
      "humidity": {
        "type": "number"
      },
      "rainfall": {
        "type": "number"
      },
      "windSpeed": {
        "type": "number"
      },
      "windDirection": {
        "enum": [
          "N",
          "NE",
          "E",
          "SE",
          "S",
          "SW",
          "W",
          "NW"
        ]
      },
      "operatingStatus": {
        "enum": [
          "OK",
          "ERROR",
          "UNKNOWN"
        ]
      },
      "timestamp": {
        "type": "string"
      }
    }
  },
  "minItems": 0
},
"link": [
  {
    "href": "searchHistoricalState",
    "mediaType": "application/json"
  }
]
}

```