**Memoria:**

Realizada por: ***Roberto Arasti Blanco, Guillermo Alcuaz Temiño, Diego Garrido Calvo y Elsa Tolín Carrasco***

*Entrega a fecha:* ***20/4/2020***



índice

[Introducción 3](#_Toc59356179)

[Manuales: 4](#_Toc59356180)

[Instalación 4](#_Toc59356181)

[Uso de la aplicación 4](#_Toc59356182)

[Documentación del protocolo de comunicaciones cliente-servidor 5](#_Toc59356183)

[Arquitectura y diseño del servicio sensor: 6](#_Toc59356184)

[Arquitectura y diseño del cliente: 7](#_Toc59356185)

## Introducción

En esta primera parte de la entrega, debido a una gestión del tiempo poco optima, únicamente hemos desarrollado y añadido las funciones de creación de usuario y gestión de privilegio de usuario.

Nos hemos organizado mal el tiempo a la hora de realizar la parte de programación en python, asi que hemos decidido explicar en la memoria el diseño de la parte no implementada esperando recibir feedback para la realización de la segunda practica de la mejor manera posible.

Manuales:

### Instalación

Una vez descargada la aplicación, a través de la consola se va a la carpeta que contiene la aplicación. Una vez allí se introducen los siguientes comandos:

* **docker-compose-f Docker/config/dev.yml up -d**: Este comando ejecuta la aplicación con Docker.
* **docker cointainer attach dms2021client**: Este segundo comando monta el cliente en el terminal que estamos usando, tras esto pedirá al usuario que se introduzca usuario y contraseña, de esta forma inicia sesión.

### Uso de la aplicación

Ya con la sesión iniciada, pide al usuario que seleccione una opción del menú de las cuatro opciones que ofrece. Las opciones que ofrece son las siguientes:

* “create user”: Si el usuario que está logeado en el sistema tiene permisos para hacerlo, tiene la opción de crear un nuevo usuario y guardar a este en la base de datos. Al seleccionar esta opción se pedirá introducir un nuevo usuario y una contraseña.
* “manage user rights”: Esta opción permite conceder y retirar permisos a un usuario. Si se selecciona esta opción, esto llevará a un menú en el que se podrán modificar los permisos del usuario.
* “change sensor rules”: Si el usuario elige esta opción le llevará a una lista de opciones/ajustes del sistema que se podrán modificar.
* “show latest monitored values”: En esta opción se puede ver el último registro de la base de datos de los valores detectados por el sensor.

De todas estas opciones, hemos implementado las dos primeras.

Documentación del protocolo de comunicaciones cliente-servidor

El API REST es la interfaz encargada de permitir interactuar con los servicios mediante HTTP. En esta comunicación los *endpoints* toman los roles de Cliente y Servidor.

En el Servidor definimos las operaciones que va a poder pedir el Cliente, en el cual definimos la llamada a la función que realizamos en el Servidor a través de HTTP. En la comunicación entre el Cliente y el Servidor las funciones son:

* Crear usuario: si el usuario tiene permisos para hacerlo, pedimos al mismo que introduzca un nuevo nombre de usuario junto a una contraseña que no existan ya en la base de datos (con esta función no es posible crear superusuarios\*).
* Log in: pedimos al usuario que introduzca una combinación usuario-contraseña ya existente.
* Log out: cerramos la sesión del usuario logueado actualmente.
* Conceder derechos: si el usuario tiene permisos para hacerlo, es capaz de asignar derechos a otros usuarios ya existentes.
* Revocar derechos: si el usuario tiene permisos para hacerlo, es capaz de quitar permisos a otros usuarios.

\*Los usuarios con el flag de *superuser* en estado de *true*, no necesitan la comprobación de permisos.

Cada una de estas funciones se asocia a una operación de CRUD de HTTP y a una URL. Cuando el servicio recibe la llamada a esa operación CRUD con esa URL, ejecuta la función correspondiente.

Arquitectura y diseño del servicio sensor: Texto.

Arquitectura y diseño del cliente: Texto.