

Trabajo Práctico Final

Este trabajo ES de entrega obligatoria.

Formato de entrega:

Deberán subir el proyecto completo (aplicación Ionic, API backend y base de datos) a un repositorio Git. Puede ser GitHub, GitLab, etc...

El proyecto debe contar con un archivo README en el que se explique con claridad de que se trata y los pasos que se deben seguir para ejecutarlo.

Descripción del problema:

Se debe crear una aplicación en Ionic para controlar sensores IoT y válvulas de riego en cultivos. Cada sensor puede medir la humedad del ambiente. Mediante la aplicación se debe poder visualizar un listado de sensores y se debe poder acceder a cada uno. Al ingresar a un sensor debe mostrar la última medición de humedad y debe permitir accionar una válvula de riego. A su vez, desde la vista del sensor se debe poder acceder a todas las mediciones asociadas a este en formato de lista.

Enunciado:

La aplicación Ionic debe contar con las siguientes características:

- Al ingresar en la ruta /home debe mostrar un listado de dispositivos traídos desde la base de datos SQL. En esta lista se debe especificar el nombre de cada dispositivo y la ubicación en la que se encuentra.
- Al hacer click sobre un dispositivo, la aplicación debe redireccionar a una página de dispositivos en la que se muestre el detalle de este. En este se debe mostrar la última medición de humedad asociada al dispositivo. A su vez, deben existir dos botones/accionadores. Uno de los botones debe permitir abrir y cerrar la válvula asociada (si está abierta debe cerrarla, si está cerrada debe abrirla). El otro debe dirigir a una página en la que se muestren todas las mediciones asociadas al dispositivo en formato de lista.
- Al hacer click en el botón para accionar la válvula (tanto para abrirla como para cerrarla), se debe guardar en la base de datos el registro del riego y el registro de la última medición de humedad.
- Debe comunicarse con la API backend para obtener y guardar datos.
- Debe tener al menos una directiva de atributo custom y un pipe custom.
- Debe utilizar al menos dos @for y un @if.

Aclaraciones:

- Dado a que no se contará con sensores reales generando las mediciones. Se propone generar datos utilizando la función `Math.random()` (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random).
- Solamente se guardarán datos en la base de datos, al clicar sobre el botón de abrir/cerrar válvula. En ese punto se guardarán los datos de humedad y del riego.
- Solamente se obtendrán datos de la base de datos al entrar en la página /home y al acceder al historial de mediciones asociadas al dispositivo. De manera opcional, se puede realizar una consulta get para ver el estado de la electroválvula y reflejarlo en el botón de abrir/cerrar electroválvula.

Estructura de la base de datos:



- Tabla Mediciones
 - Tabla en donde se registra fecha y hora de la medición junto con el id perteneciente al dispositivo que registró esa medición.
- Tabla Dispositivos
 - Representa a un sensor de humedad . Esta tabla va a tener el identificador único junto con un nombre y un lugar en donde esté ubicado.
- Tabla Electroválvula
 - Representa a la válvula que permite el paso o no del flujo de agua. Esta tabla contiene el identificador propio junto con el identificador del dispositivo al cual está conectado.
- Tabla Log_Riegos
 - Representa al histórico de aperturas/cierres de electroválvula, por cada vez que se abra o se cierre una electroválvula, se insertará un registro en esta tabla.

Material de interés:

<https://ionicframework.com/docs/api/list>

<https://ionicframework.com/docs/api/card>

<https://ionicframework.com/docs/api/buttons>