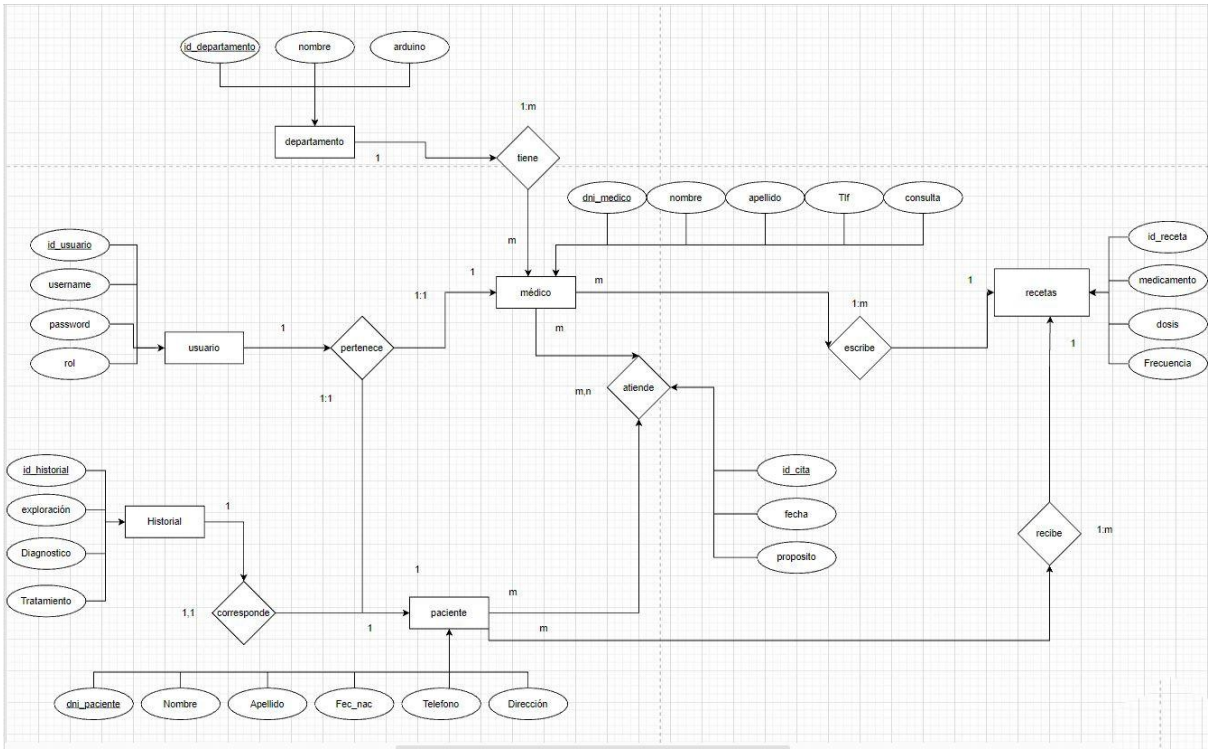
Documentación Aplicación Hospital

OSCAR GARCIA DORADO

I.E.S RIBERA DEL TAJO

Base de datos

Modelo entidad-relación:



Modelo relacional:

Modelo relacional: Las claves primarias están subrayadas y las foráneas están en color azul ya que no puedo poner subrayado doble.

Tabla Paciente:

- dni_paciente
- nombre
- apellido
- fec_nac
- telefono
- direccion
- id_user

Tabla Médico:

- dni_medico
- nombre
- apellido
- telefono
- id_departamento
- id_user

Tabla Departamento:
- id_departamento
- nombre
- arduino

Tabla Historial:
- id_historial
- exploracion
- diagnóstico
- tratamiento
- id_paciente

Tabla Cita:
- id_cita
- fecha
- proposito
- id_paciente
- id_medico

Tabla Receta:
- id_receta
- medicamento
- dosis
- frecuencia
- id_paciente
- id_medico

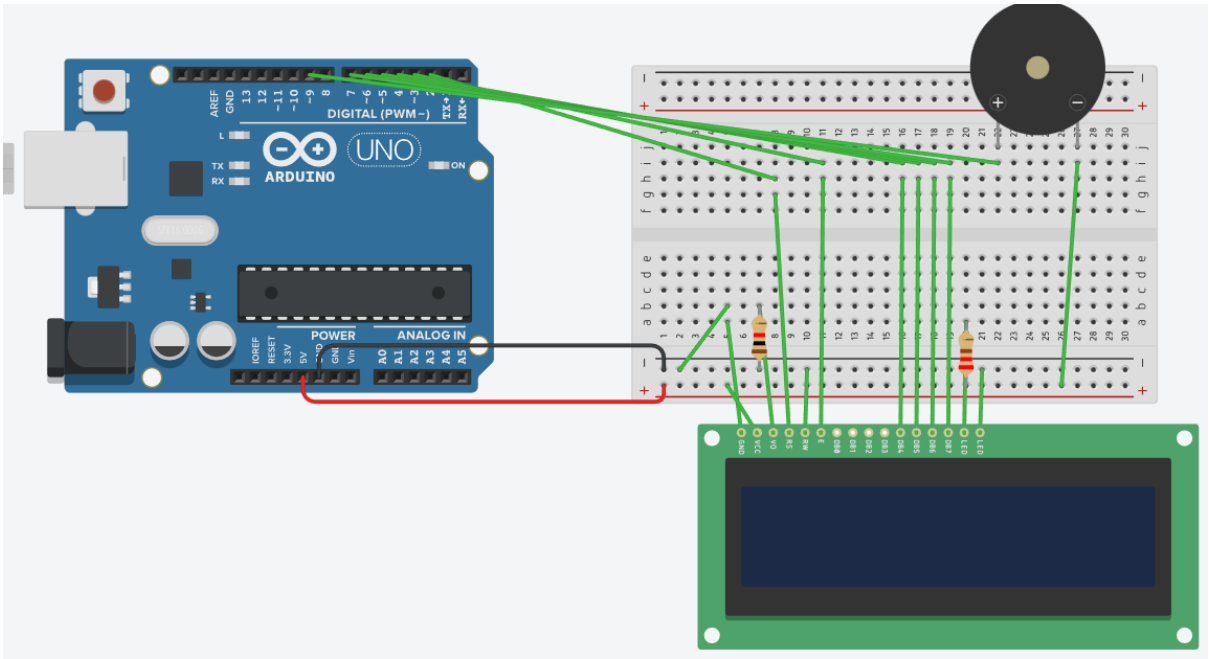
Tabla Usuario:
- id_user
- username
- password
- rol

Arduino:

Para poder llamar a un paciente a la consulta de su cita se han usado los siguientes elementos:

- Arduino uno.
- Ethernet shield.
- Breadboard
- Resistencias: 1k y 220 ohmios.
- Buzzer tmb12a05.
- Módulo LCD.

Circuito eléctrico:



En el circuito eléctrico tendremos el buzzer conectado directamente al puerto 9 ya que no es necesario ninguna resistencia. Luego tendremos la pantalla lcd conectada de la siguiente manera:

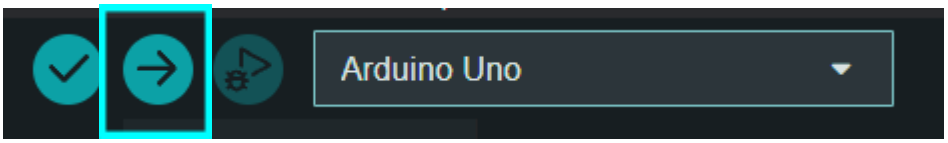
- VSS: conectado a tierra.
- VDD: conectado al puerto de alimentación de 5V.
- V0: este pin se encarga de controlar el contraste. Se conecta a una resistencia y dicha resistencia a tierra de manera que reduciremos el voltaje para controlar el contraste, normalmente se usa un potenciómetro.
- RS: indica si se reciben datos o instrucciones. Este puerto se conectará al puerto nº2 de la arduino para que se controle cuando se envían datos o instrucciones.
- RW: indica si la pantalla está en modo lectura (alto) o escritura (bajo), como a nosotros nos interesa que esté en modo escritura lo conectaremos a tierra.
- E: en este puerto se recibirá cuando termina la operación por lo tanto lo conectaremos al puerto 3 de la arduino.
- Puertos DB4-DB7: estos puertos se usan para recibir los datos, con conectar estos puertos son suficientes en nuestro caso. Estos puertos se conectan directamente a la arduino.
- Puertos LED: estos pines controlan la potencia del led que proporciona la luz de fondo. El primer puerto LED es el polo positivo por lo que pondremos una resistencia de 220 ohmios para controlar que no se vea demasiada luz (normalmente se usa un potenciómetro) y el siguiente es el polo positivo, van conectados a 5V y a tierra respectivamente.

Para hacer funcionar a la arduino solamente hay que conectar la placa al pc mediante USB para recibir alimentación y un cable RJ45 al shield, ya sea directo al PC o al router. Es importante tener en cuenta que tanto la arduino como el servidor en el que esté alojado la API deben estar en el mismo segmento de red. Por defecto la arduino tendrá la siguiente IP: 192.168.1.177 pero esto se puede cambiar desde el código de la arduino para ello seguiremos los siguientes pasos:

1. Conectaremos la Arduino al pc mediante el USB (no es necesario conectarlo de momento a la red).
2. Abrimos el IDE de arduino.
3. Modificamos la IP en la siguiente parte del código.

```
IPAddress ip(192,168,1,177); //Asignamos  la IP al Arduino
```

4. Inyectamos el código en la arduino haciendo click en el siguiente botón:



Una vez hayamos seguido estos pasos ya tendremos configurada nuestra arduino, es importante tener en cuenta que la primera conexión que se haga después de inyectar el código tardará un poco más de lo normal.

Funcionamiento:

El arduino nos permite programar circuitos eléctricos. El lenguaje usado es una versión de C reducida. Gracias al Ethernet Shield y sus librerías podemos crear un servidor web que recibirá peticiones HTTP.
Dicho servidor estará constantemente escuchando, en el momento que reciba una petición guardaremos la petición para poder leer los datos que necesitamos enviados a través de la URL. Los datos recibidos son enviados a la pantalla LCD para que sean mostrados. Por último se comprobará si hay un salto de línea lo que significa que la petición ha terminado.

API-REST:

Para realizar peticiones a la base de datos se usará una API-REST que nos permitirá realizar consultas de una manera sencilla añadiendo una capa de seguridad. El funcionamiento de dicha API se basa en el uso de endpoints y del acceso a estos endpoints con los distintos métodos (GET, POST, PUT y DELETE). Para poder realizar ciertas peticiones se necesitarán ciertos permisos los cuales se detallarán más adelante.

El resultado que nos devolverá la API es un JSON con el siguiente formato:


```
{
    "status": 200,
    "results": [
        {
            "id_usuario": "1",
            "username": "admin",
            "password": "$2y$10$qJbDpxbOxxEKrfQDJ87UbuMETceMokvDlzjy/xmonAKNtaGyaCGrC",
            "rol": "admin"
        },
        {
            "id_usuario": "3",
            "username": "prpr3744",
            "password": "$2y$10$GBhJEV9f8HZLxiLT2idd9Og3KCsbmkKT93Pv4vqLm4NN/fkOIr0Z2".
```


- Status: en este campo recibiremos el código del resultado de la petición. Los diferentes códigos que podremos recibir serán detallados más adelante.
- Results: aquí recibiremos el resultado de la petición, dicho resultado. Dependiendo de la petición realizada nos devolverá un mensaje con el resultado o uno o varios registros.

Endpoints:

Para realizar las operaciones con la base de datos se usan los endpoint para indicar la tabla sobre la que se va a operar y el método para indicar el tipo de operación que se va a realizar.
La forma de un endpoint en la siguiente:

http://localhost/api_hospital/tabla/

En el endpoint sustituiremos “tabla” por el nombre de la tabla a la que queramos acceder.

Método GET:

Para poder recoger datos usaremos el método GET. En este método tenemos la posibilidad de personalizar la consulta que se necesite hacer, para ello podemos añadir los siguientes elementos al endpoint:

- startAt: con startAt podremos indicar que nos devuelva registros a partir de x registro: http://localhost/api_hospital/usuario/?startAt=2
- limit: limit nos permite limitar el nº de registros que queremos recibir: http://localhost/api_hospital/usuario/?limit=5
- orderBy: si queremos ordenar los datos por algún campo tendremos que usar este parámetro seguido del campo por el que queremos ordenar. Junto a este parámetro podremos usar orderMode para indicar si queremos que se ordene de manera ascendente (asc) o descendente (desc): http://localhost/api_hospital/usuario/?orderBy=id_usuario&orderMode=desc
- linkTo y equalTo: si queremos filtrar tendremos que indicar el/los campos por los que queremos filtrar con linkTo seguido de equalTo con las condiciones que tienen que cumplir. Si queremos poner varias condiciones tendremos que ponerlas separadas por comas: http://localhost/api_hospital/usuario/?linkTo=id_usuario&equalTo=1

Si queremos realizar joins tendremos que usar el siguiente endpoint:
http://localhost/api_hospital/relations/?rel=cita,paciente,medico&key=dni_paciente,dni_medico

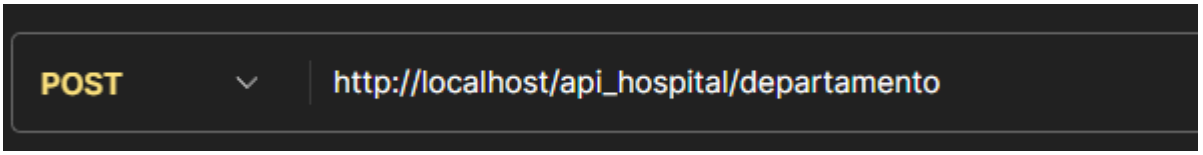
Con relations indicaremos que vamos a hacer join y luego con rel indicaremos primero la tabla y luego, separados por comas, las tablas que vamos a unir. Para indicar el campo por el que se va a unir usaremos key y el campo con el que se relaciona la tabla que se va a unir con la primera tabla.

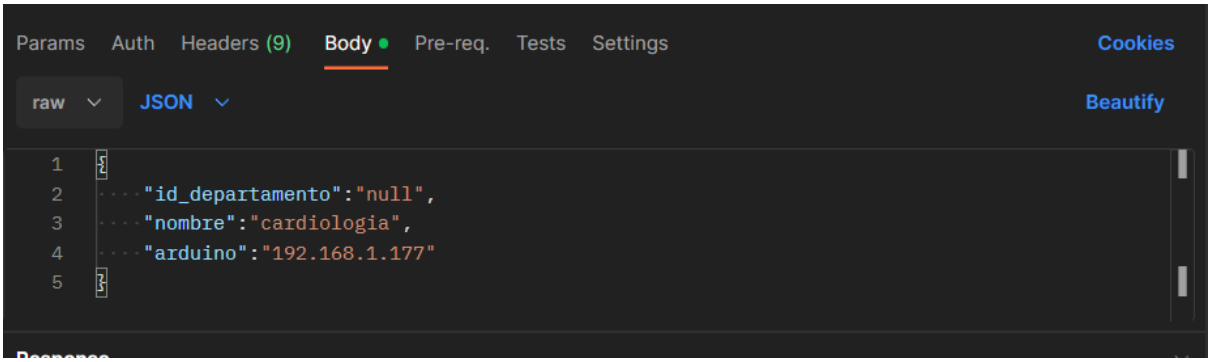
NextCita:
Para mostrar la siguiente cita en la arduino usaremos el siguiente endpoint:
http://localhost/api_hospital/nextCita/?fecha=2023/03/08
La fecha debe ser la fecha de la cita y automáticamente la api realizará la petición a la arduino con los datos necesarios.

Método Post:

El método Post nos permite insertar datos en la bbdd. Para ello tendremos que pasar todos los datos en el body de la petición. Es importante que pongamos todas las columnas en el body, en caso de que la clave primaria sea autoincremental el valor será null.

Un ejemplo de petición Post seria el siguiente:





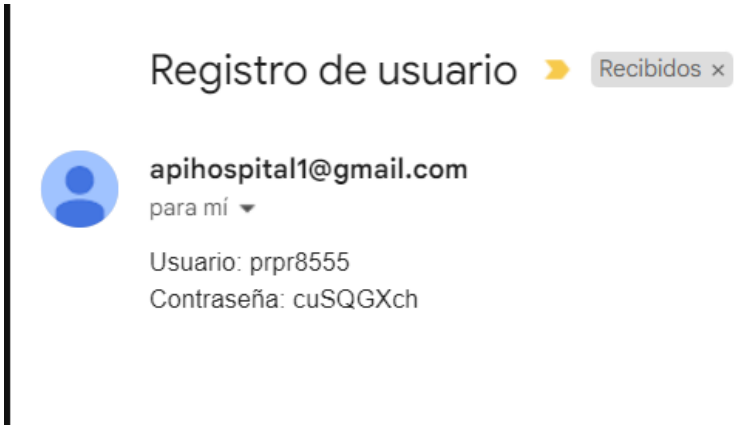
Registro:
Para registrar a una persona, ya sea médico o paciente, tendremos que usar el siguiente endpoint:

http://localhost/api_hospital/register/?rol=paciente

o

http://localhost/api_hospital/register/?rol=medico

El body de la request es igual que una petición post pero al registrar no solo devolverá una respuesta si no que enviará un correo con el nombre de usuario y la contraseña generada.

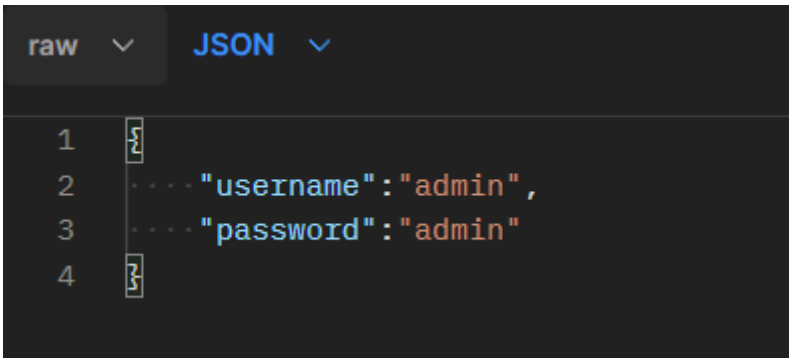


Login:

Para acceder tendremos que usar el siguiente endpoint:

http://localhost/api_hospital/login

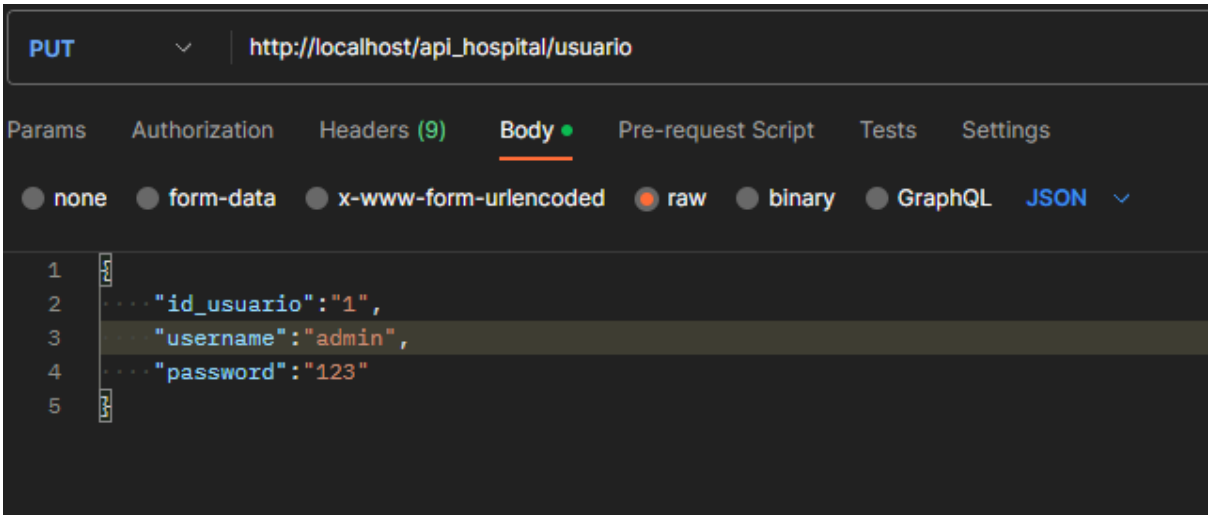
En el body tendremos que poner tanto el usuario como la contraseña:



Si tanto el usuario como la contraseña son correctos la API nos devolverá un token que tendrá una validez de 1 hora el cual tendremos que incluir en el header de las peticiones que vayamos a realizar.

Método Put:

El método Put nos permite modificar la información de un registro, para ello tendremos que introducir en el body todos los campos del registro.



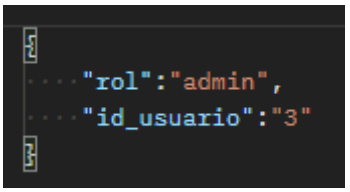
En el caso de querer modificar un usuario, un paciente o un médico no hace falta incluir el rol y es importante tener en cuenta que solo se podrá modificar el usuario con el que se ha accedido.

ChangeRol:

En caso de necesitar cambiar el rol de un usuario, por ejemplo cambiar de médico a administrador se puede usar el siguiente endpoint:

http://localhost/api_hospital/changeRol

Para poder realizar la petición tendremos que incluir en el body el id del usuario y el rol:



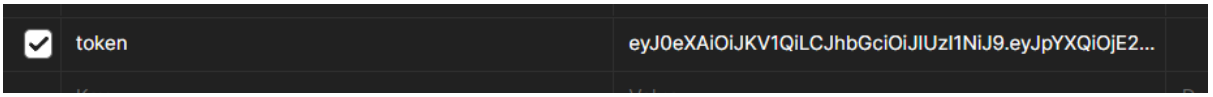
Método Delete:

En caso de querer borrar un registro podemos usar el método delete. Para poder realizar la petición tendremos que incluir en el endpoint el campo por el que vamos a filtrar.

http://localhost/api_hospital/paciente/?dni_paciente=1

Permisos:

Para poder realizar el control de acceso se ha creado tres roles: admin, médico y paciente. Para la mayoría de accesos (todos los accesos excepto el login o el registro de pacientes) se necesita incluir un token en el header de la petición, dicho token se obtiene realizando la petición al login. El token debe llamarse “token”:



Cada rol podrá realizar ciertas acciones:

Admin:
El administrador de la aplicación podrá realizar cualquier acción, hay ciertas acciones que solo se podrán realizar con este rol:

- changeRol
- Operar con los departamentos
- Borrado de usuarios
- Registro de médicos

La aplicación cuenta con un usuario administrador el cual usa las siguientes credenciales:

- Nombre de usuario: admin
- Contraseña: admin

Médico:
Los usuarios con rol médico pueden realizar operaciones de cualquier tipo sobre los historiales o las recetas además de consultar datos de otras tablas. Otra operación que solo podrán realizar médicos o administradores es el uso del endpoint nextCita para realizar peticiones a la arduino.

Paciente:
Los pacientes mayormente podrán consultar datos de las tablas aparte de modificar los datos del propio paciente.

Consultas sin rol:
Hay dos consultas que no necesitan rol que son el login y el registro.
En el registro solo se podrá registrar paciente ya que si el rol es médico necesitará haber accedido con un usuario administrador.

Aplicaciones usadas:

Para el desarrollo de la aplicación se han usado distintas herramientas las cuales van a ser explicadas a continuación:

Visual studio code:

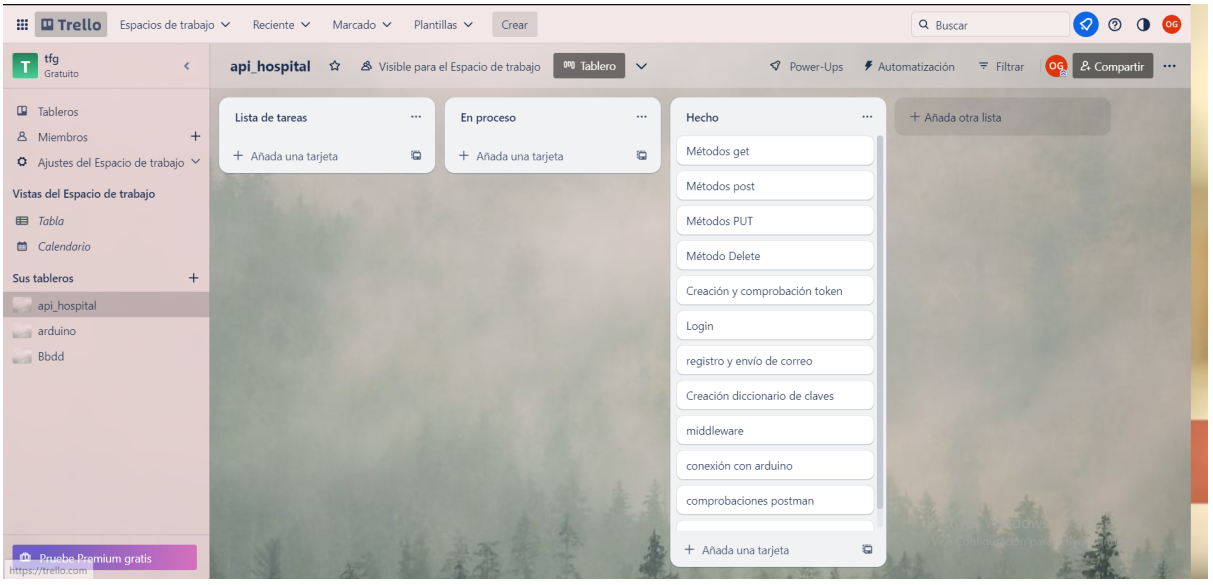
Visual studio code es un ide el cual permite desarrollar aplicaciones de una manera cómoda debido a la gran cantidad de extensiones que se pueden instalar. Algunas extensiones instaladas son: debug para php y para js.

Git/Github:

Git es un software de control de versiones, el cual permite controlar y moverse entre las versiones del programa. Junto a git se ha usado github que es una plataforma que permite alojar nuestras aplicaciones y controlar sus versiones.
La url del repositorio es: https://github.com/varialheel/tfc

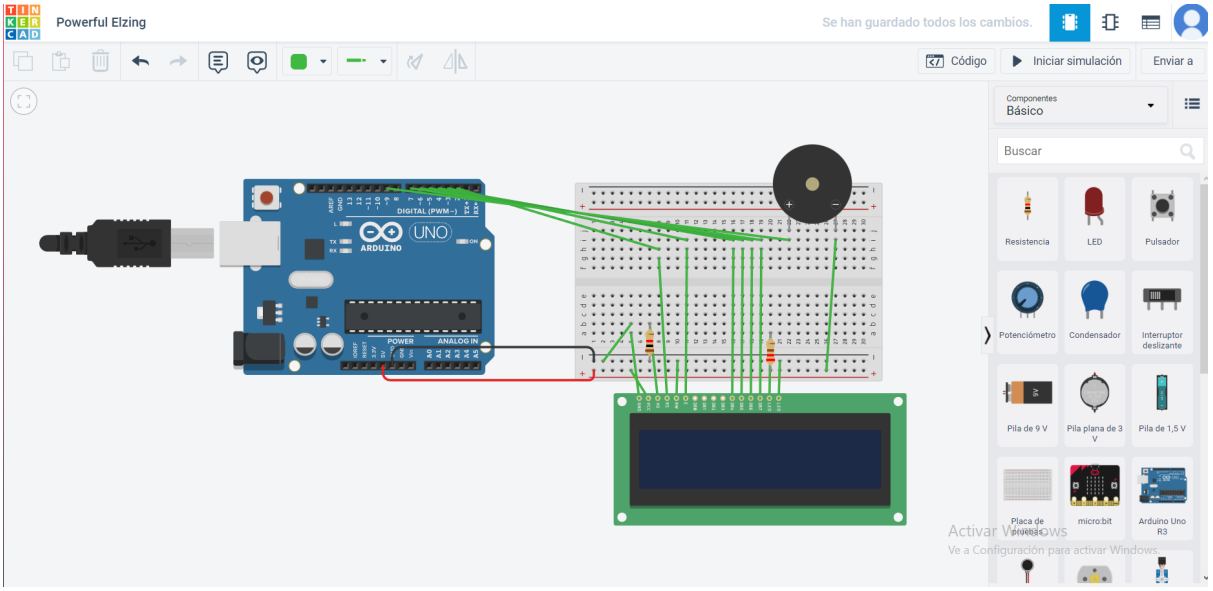
Trello.com:

Trello es una aplicación web que permite gestionar las tareas, esto permite crear tareas y clasificarla según el estado de la tarea.



Tinkercad:

Tinkercad es una aplicación que permite realizar cursos, uso de simuladores… Pero en nuestro caso se ha usado para realizar el esquema del circuito eléctrico de la arduino, esto ha permitido realizar la instalación de una manera más clara.



Arduino IDE:

El ide de arduino es un software que nos permite desarrollar aplicaciones para nuestra arduino. Con este ide hemos podido desarrollar el código, compilarlo y subirlo a la memoria de nuestro arduino e instalación de librerías.
En la raíz del proyecto tenemos una pequeña guía para instalar tanto el ide como la librería utilizada.

Postman:

Postman es un software usado para realizar peticiones a API, en este caso se ha usado para realizar las pruebas a nuestra API. En la raíz del proyecto se encuentra un archivo json con algunas pruebas realizadas. Este archivo tendremos que importarlo y ya podremos realizar las pruebas.

Draw.io:

Draw.io es un software que nos permite realizar distintos tipos de diagramas. Tiene dos versiones: una web y otra de escritorio. Con esta aplicación se ha realizado el modelo entidad-relación de la base de datos.