

# Documentación Aplicación Hospital

OSCAR GARCIA DORADO

I.E.S RIBERA DEL  
TAJO



<b>Introducción:</b>	<b>3</b>
<b>Forma jurídica:</b>	<b>3</b>
<b>Análisis del entorno:</b>	<b>4</b>
<b>Servicios prestados:</b>	<b>4</b>
<b>Inversión:</b>	<b>4</b>
<b>Objetivos:</b>	<b>5</b>
<b>Caso práctico:</b>	<b>5</b>
<b>Base de datos:</b>	<b>6</b>
Modelo entidad-relación:	6
Modelo relacional:	6
<b>Arduino:</b>	<b>7</b>
Circuito eléctrico:	8
Funcionamiento:	9
<b>API-REST:</b>	<b>10</b>
Endpoints:	10
Método GET:	11
NextCita:	11
GetHistory:	12
Método Post:	12
Registro:	13
Login:	13
VerificationCode:	14
SendCode:	15
Método Put:	15
ChangeRol:	15
ChangePassword:	16
Método Delete:	16
Permisos:	16
Admin:	16
Médico:	16
Paciente:	16
Consultas sin rol:	17
<b>Cliente:</b>	<b>17</b>
Index:	17
Registro:	19
Cambiar contraseña:	20
Páginas de los pacientes:	20
Home:	20
Ver cita:	21
Historial:	22
Recetas:	23
Pedir cita:	24

Perfil:.....	24
Páginas de los médicos:.....	25
Home:.....	25
Atender:.....	26
Ver pacientes:.....	27
Ver médicos:.....	28
Perfil:.....	31
<b>Casos de uso:.....</b>	<b>32</b>
Login:.....	32
Atender cita:.....	33
Eliminar cuenta:.....	34
Aplicaciones usadas:.....	35
Visual studio code:.....	35
Git/Github:.....	35
Trello.com:.....	35
Tinkercad:.....	36
Arduino IDE:.....	36
Postman:.....	36
Draw.io:.....	37
TempMail:.....	37
DOI:.....	37

## Introducción:

HealthNet es una aplicación que da soporte principalmente a hospitales e instituciones relacionadas con la salud. Está desarrollada con tecnología web y ofrece una solución sencilla y sobretodo escalable para ofrecer los mejores servicios.

HealthNet es el fruto de mi trabajo individual como freelancer, en el cual he asumido todas las responsabilidades, desde la concepción de la idea hasta la implementación final. He invertido tiempo y esfuerzo en cada etapa del proceso de desarrollo, desde la planificación y el diseño, hasta la codificación y las pruebas.

## Forma jurídica:

La forma jurídica elegida es la de Trabajador por cuenta propia ya que esto me permite ejercer mi actividad de manera independiente.

La forma jurídica escogida goza de las siguientes ventajas y desventajas:

### Ventajas:

- Independencia: al ser el único trabajador soy libre de tomar decisiones y de gestionar mi propio negocio.
- Flexibilidad de horarios.
- Posibilidad de crecimiento personal y profesional.
- Posibilidad de aumento de ingresos.

### Desventajas:

- Carga administrativa.
- Inestabilidad económica.

El ser autónomo también me permite poder solicitar distintas ayudas como:

- Capitalización del paro: esta ayuda permite cobrar el 100% del paro por adelantado.
- Tarifa de autónomo: con este beneficio la cuota de autónomo se reduciría a 60€ al mes durante el primer año.

## Análisis del entorno:

Talavera de la Reina está ganando poder en cuanto a las empresas informáticas se refiere. Este impulso es debido a los ciclos formativos y a la carrera de informática que se imparte en la universidad.

Esto hace que el entorno sea muy competitivo y con competencia. Varias de las empresas que actualmente podrían ser competencia son:

- [GrupoDW](#): es una empresa de desarrollo especializado en tiendas online para farmacias.
- [QuantumStudio](#): es una empresa dedicada al desarrollo y al mantenimiento de software ya sea con tecnología web como otro tipo de tecnologías.

## Servicios prestados:

Los servicios prestados son:

- API-REST: una api-rest que ofrecerá una comunicación sencilla y segura para el manejo de la base de datos.
- Clientes web: también ofrecemos el desarrollo de clientes web con diferentes tecnologías: desde html, css y js puro hasta diferentes frameworks como reactjs o vue para poder satisfacer de la mejor manera las necesidades del cliente.
- Mantenimiento y ampliación de las aplicaciones: también se ofrece el mantenimiento de dichas aplicaciones.

## Inversión:

Para la inversión inicial solo se necesita pagar la cuota de autónomo que con la ayuda se quedarían en 720€, el hosting web elegido es [Webempresa](#) ya que tiene muy buenas opiniones y el plan básico es de 100€ al año.

Debido a que no se necesita pagar salarios la inversión anual es de 820€.

## Objetivos:

Los objetivos no es solo vender los productos y servicios anteriormente mencionados si no ir mejorando poco a poco. Varios objetivos son:

- Mejora del control de roles añadiendo el administrador de departamentos.
- Ofrecer aplicaciones de escritorio y de móvil.
- Mejora de la api.

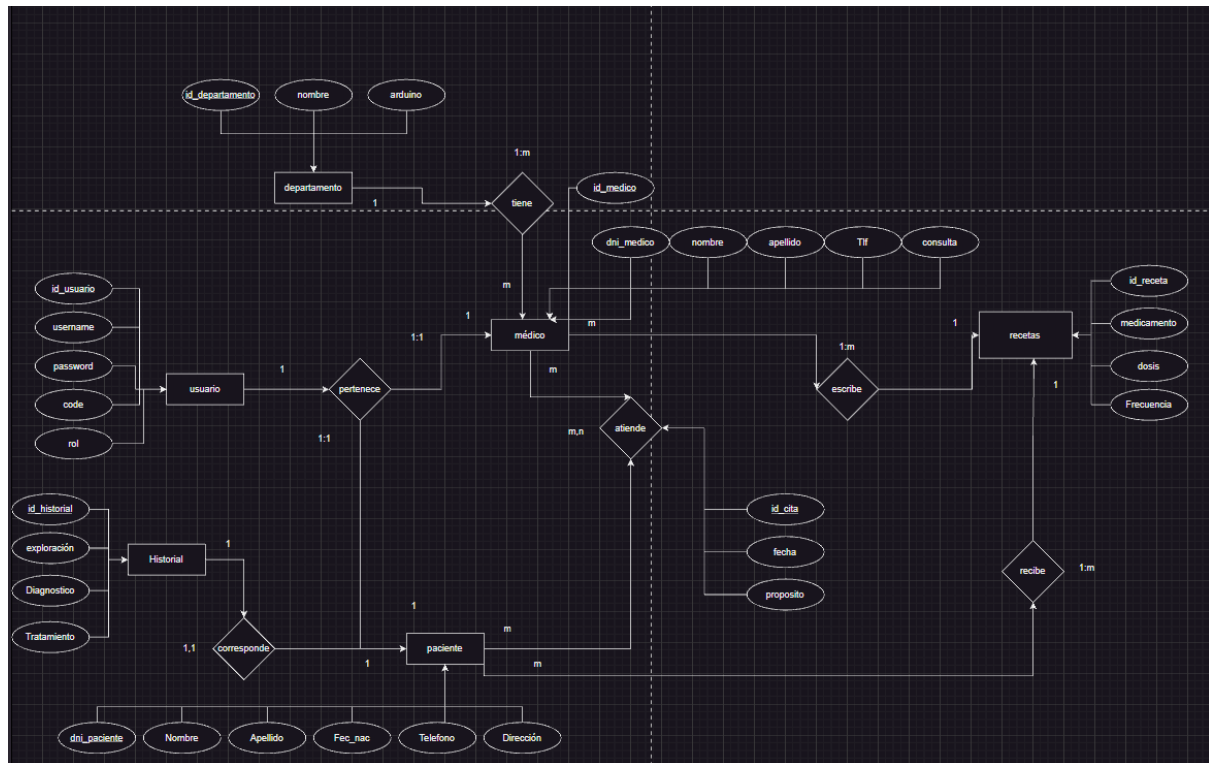
## Caso práctico:

A continuación se explicará un ejemplo de un servicio ofrecido a un cliente. Este servicio ha tenido una duración de 3 meses. Para poder realizar este proyecto se ha tenido en cuenta los siguientes puntos:

- Seguridad: la aplicación debe ser segura. Para conseguir este punto se ha realizado un control de roles: paciente, médico y administrador. En la api se usan token cifrados por una clave única para cada usuario la cual se encuentra en un diccionario de claves.
- Escalabilidad: para poder realizar un producto escalable se tomó la decisión de desarrollar una API-REST ya que permitirá tener un cliente web y en un futuro tener aplicaciones de escritorio y/o de móvil.
- Interfaz sencilla: se ha tratado de realizar una interfaz sencilla e intuitiva para el usuario tratando de disminuir los niveles de profundidad de la aplicación.
- Base de datos: se ha desarrollado un modelo de base de datos que satisface todas las necesidades del cliente.
- Arduino: el cliente necesitaba mostrar al paciente a que consulta tiene que acudir. Para poder realizar esta tarea se ha decidido incluir una arduino con un ethernet Shield acoplado el cual recibirá peticiones HTTP de la api para mostrar los datos en una pantalla LCD.

# 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:

- id\_medico
- dni\_medico
- nombre

- apellido
- telefono
- id\_departamento
- id\_user

Tabla Departamento:

- id\_departamento
- nombre
- arduino

Tabla Historial:

- id\_historial
- exploracion
- diagnóstico
- tratamiento
- dni\_paciente

Tabla Cita:

- id\_cita
- fecha
- proposito
- dni\_paciente
- id\_medico

Tabla Receta:

- id\_receta
- medicamento
- dosis
- frecuencia
- dni\_paciente
- id\_medico

Tabla Usuario:

- id\_user
- username
- password
- code
- rol

## Arduino:

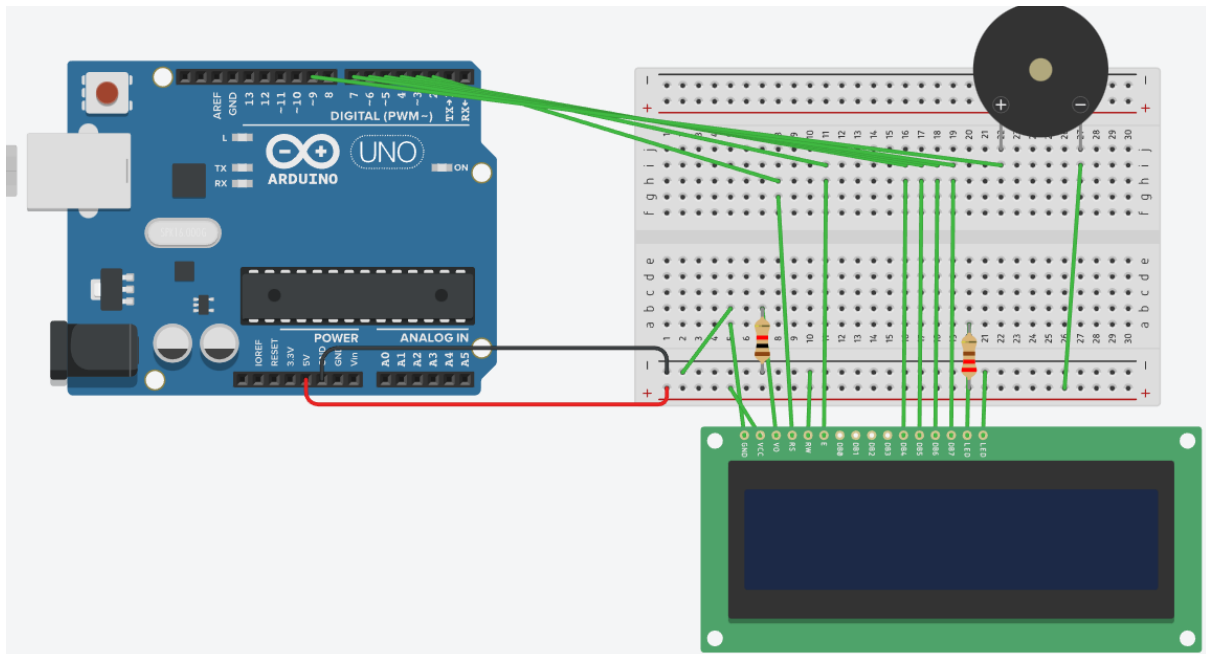
Para poder llamar a un paciente a la consulta de su cita se han usado los siguientes elementos:

- 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

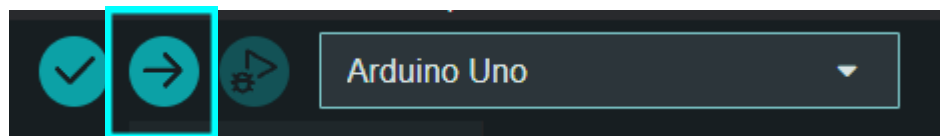
(normalmente se usa un potenciómetro) y el siguiente es el polo positivo, van conectados a 5V y a tierra respectivamente.

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$10$qJbDpxb0xxEKrfQDJ87UbuMETceMokvDlZjy/xmonAKNtaGyaCGrC",
      "rol": "admin"
    },
    {
      "id_usuario": "3",
      "username": "prpr3744",
      "password": "$2v$10$GBhJEV9f8HZLxiLT2idd90g3KCsbmkKT93Pv4voLm4NN/fk0Ir0Z2"
    }
  ]
}
```

- 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 es la siguiente:

[http://localhost/api\\_hospital/tabla/](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](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](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](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](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](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](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.

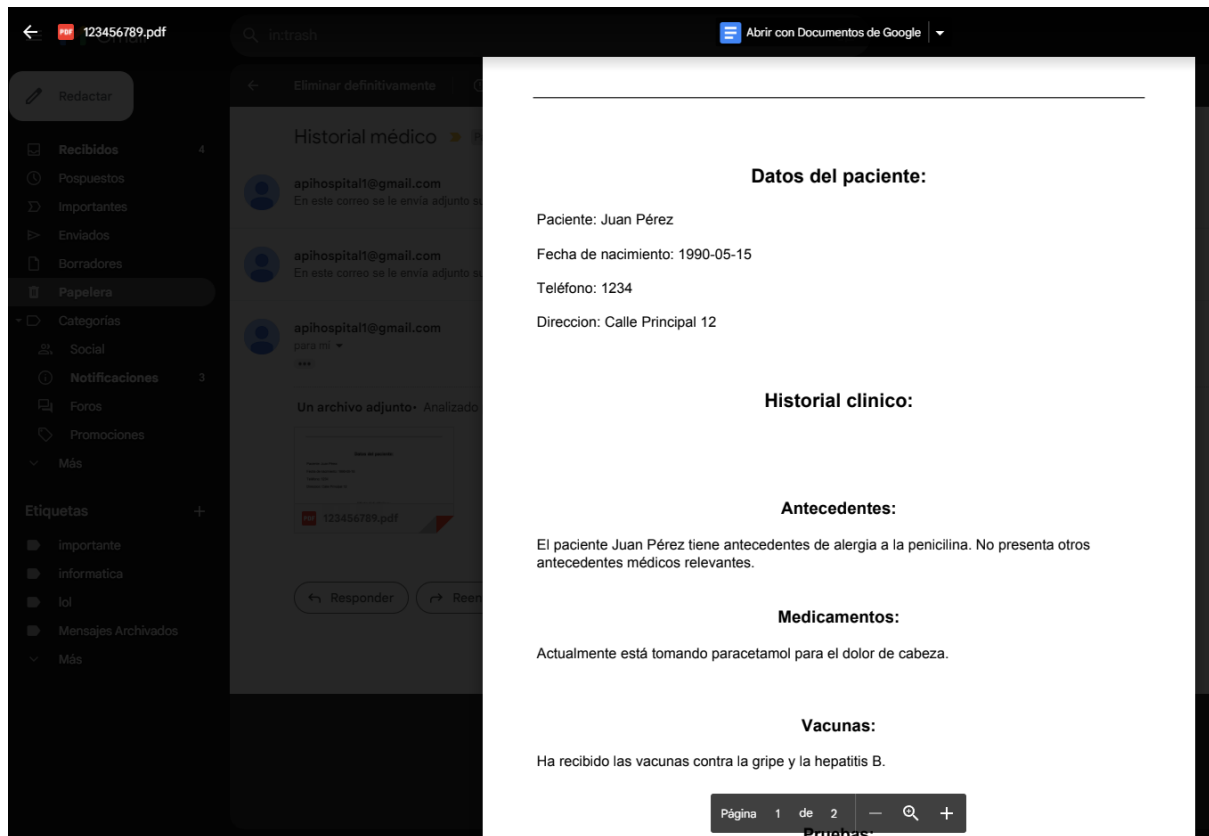
A este endpoint también podemos indicarle que muestre una cita en concreto pasando el id de la cita de la siguiente manera:

[http://localhost/api\\_hospital/nextCita/?id\\_cita=1](http://localhost/api_hospital/nextCita/?id_cita=1)

### GetHistory:

El endpoint getHistory nos permitirá recoger los datos del historial y conformar un pdf con dichos datos para posteriormente ser enviado al correo del paciente.

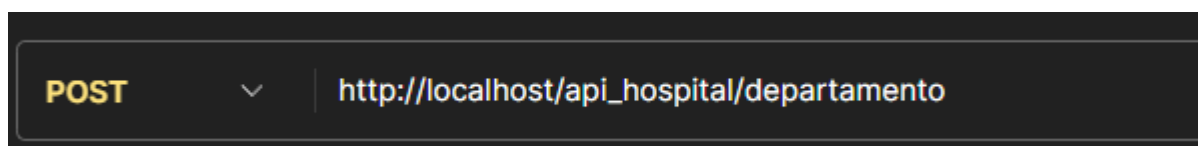
[http://localhost/api\\_hospital/getHistory](http://localhost/api_hospital/getHistory)



### 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 sería el siguiente:



```

1  {
2    "id_departamento": "null",
3    "nombre": "cardiologia",
4    "arduino": "192.168.1.177"
5  }

```

Registro:

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

[http://localhost/api\\_hospital/register/?rol=paciente](http://localhost/api_hospital/register/?rol=paciente)

o

[http://localhost/api\\_hospital/register/?rol=medico](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](http://localhost/api_hospital/login)

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

```

1  {
2    "username": "admin",
3    "password": "admin"
4  }

```

Si tanto el usuario como la contraseña son correctos la API y el usuario es un paciente nos devolverá un token que tendrá una validez de 24 horas el cual tendremos que incluir en el header de las peticiones que vayamos a realizar.

En caso de que el usuario sea un médico o un administrador nos devolverá un estado 200 y enviará al email un código de verificación el cual tendremos que enviar al siguiente endpoint.

```
"status": 200,  
"results": "eyJ0eXN0IjEwMTQ1ICBjbG9ic1JlUzI1NiJ9.  
eyJpYXN0IjEwM0QyNjQxMjgsImV4cCI6MTY4NDI2ZncyOcw1ZGF0YSI6eyJpZF91c2Vhcm1vIjo1MSIsInVzZXJuYmwiIjo1YWRTaw41LCBkYWNzd29yZCI6IiQyeSQQMcRxs  
mJEchhiT3h4RUtyZlFESjg3VWJ1TUUVU2VNb2t2RGx6ankveG1vbKFLTNRhR3lh0dy0yIsInJvbCI6ImFkbWluIn19.  
0V0wxP_ImkSQouSVj3exC9gmcp-GylTv978n_VRC8Y&1"
```

```
1  {
2    "status": 200,
3    "results": "Login done"
4  }
```

apihospital1@gmail.com

para mí ▼


 inglés ▾ > español ▾ Traducir mensaje

Su código de verificación es: ezh2HrRi

VerificationCode:

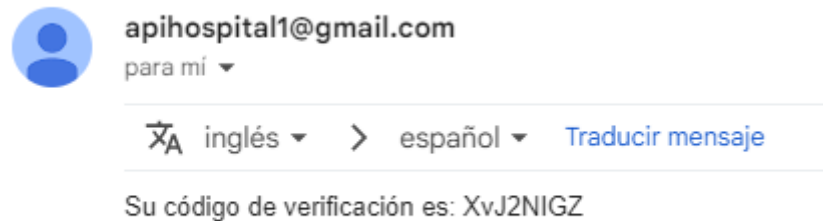
Este endpoint nos permitirá enviar el código de verificación recibido, en caso de ser correcto nos devolverá el token:

[http://localhost/api\\_hospital/verificationCode](http://localhost/api_hospital/verificationCode)

```
"username": "meap5678",  
...  
"code": "XvJ2NiGZ"  
  
[25]  
  
Cookies Headers (10) Test Results  
Status: 200 OK Time: 75 ms Size: 804 B Save as Example  
tty Raw Preview Visualize JSON ↕  
{  
  "status": 200,  
  "results": {  
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.  
            eyJpYXQiOiJlMjEwMDYyOTk0MTY0NDU2NjIzMWwiZGF0YSI6eyJpY3F1c3Vhcm1vIjo1MTYiLCJlc2VybmFtZSI6Im1lYXAiNjc4IiwicGFzc3dvcmQ0IjoiaWkubkkMTAKqZQzL3JBnFB3ZmpETmZqU0J5R3oxT0Qwc3FiV0lCSXRzSEl2SGszRU9uVnVnVllVSGEuQksilCJjb2RIIjo1WHZKMk5JR1oiLCJyb2wiOiJtZWRpY28ifX0.  
            Z_Wn2k4V5B8s44zmEyW0dix5hgghWA-sJ5XEVEH3E74&16",  
    "rol": "medico"  
  }  
}
```

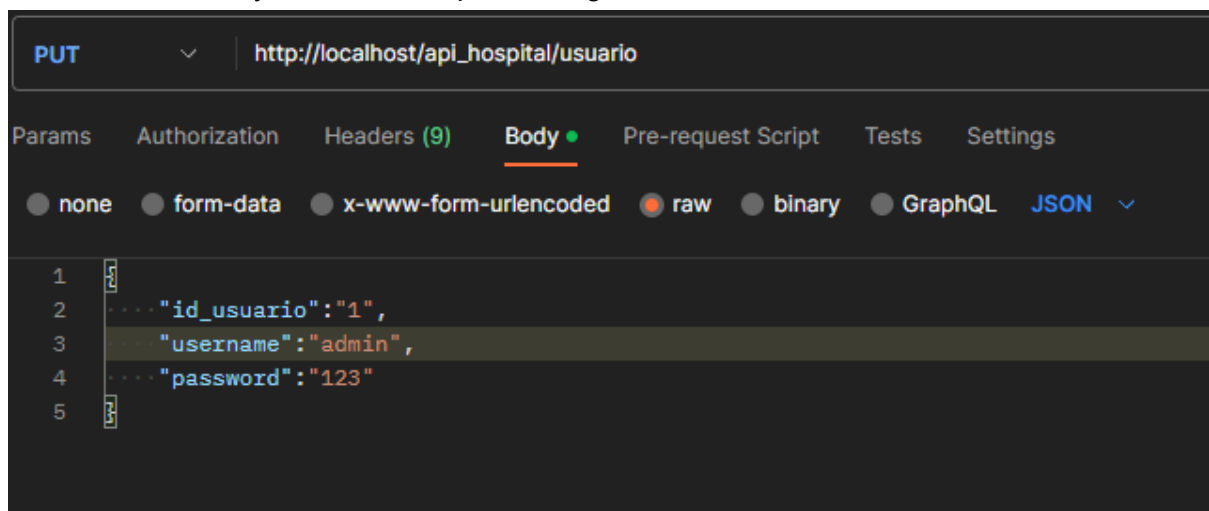
SendCode:

En caso de que no llegue correctamente el código de verificación podremos usar el endpoint sendCode de manera que volverá a enviar el correo con el código:



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](http://localhost/api_hospital/changeRol)

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



```

{
  "rol": "admin",
  "id_usuario": "3"
}

```

### ChangePassword:

Si el usuario no recuerda la contraseña podremos usar este endpoint para que se genere una nueva contraseña y sea enviada al correo:

[http://localhost/api\\_hospital/password](http://localhost/api_hospital/password)

### 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](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":

<input checked="" type="checkbox"/> token	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiJlZ2...
---	--

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 varias 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.

Otra consulta que se puede realizar sin rol es verificationCode ya que es el endpoint que usaremos para recibir el token.

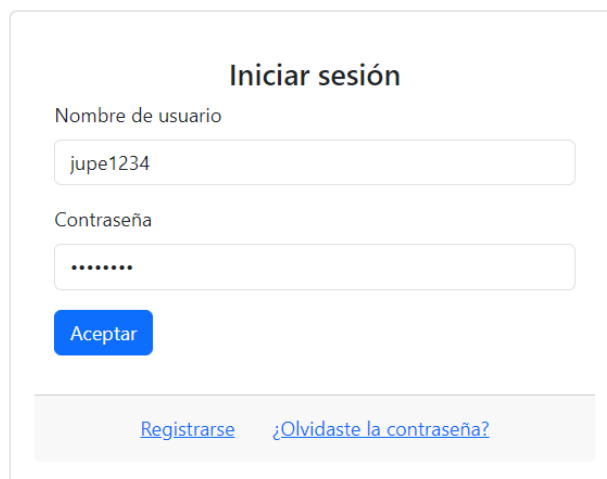
## Cliente:

Para realizar las peticiones a la API se ha creado un cliente que nos permitirá manejar la información de una manera sencilla. Este cliente está formado por varias páginas comunes: login, registro y recuperación de contraseña y luego dependiendo del rol se podrá acceder a unas u otras.

A continuación se detallarán todas las páginas:

## Index:

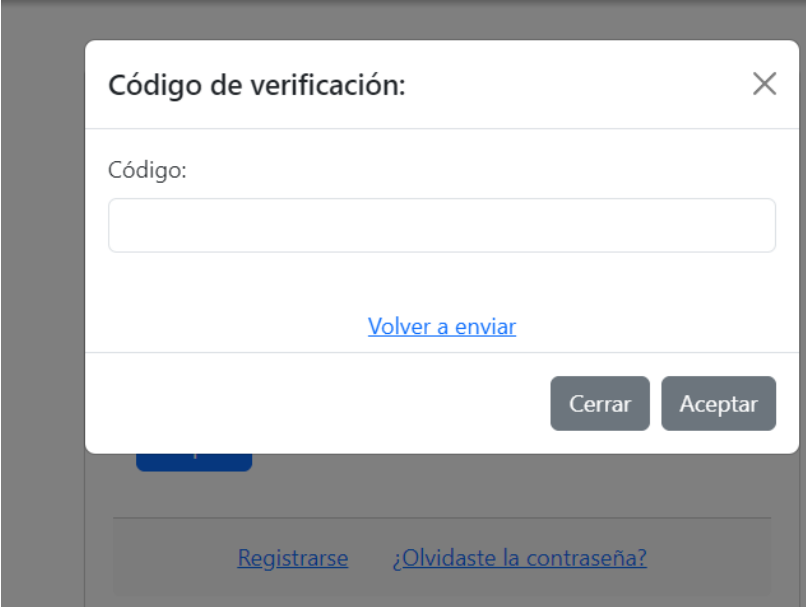
En el index encontraremos el formulario de login con dos enlaces que nos llevarán a las páginas de registro y de recuperación de contraseña. Para poder realizar el inicio de sesión es obligatorio introducir tanto el nombre de usuario como la contraseña.



The image shows a login form titled "Iniciar sesión". It contains two input fields: "Nombre de usuario" with the text "jupe1234" and "Contraseña" with masked characters ".....". Below the fields is a blue "Aceptar" button. At the bottom, there are two links: "Registrarse" and "¿Olvidaste la contraseña?".

Al realizar el inicio de sesión se realizará una comprobación del rol del usuario y dependiendo del rol tendremos dos casos:

- El rol del usuario es paciente: si el usuario es paciente se guardará su información en la sessionstorage y será enviado a su página de inicio.
- El rol del usuario es médico/admin: si el rol es médico o administrador entonces de mostrará un modal donde se tendrá que introducir el código recibido. En dicho modal nos encontraremos con un botón que volverá a enviar el código en caso de no recibirlo. Una vez introducido el código y comprobado que es correcto se guardará la información en la sessionstorage y será redirigido a su página de inicio.

Un modal de verificación de código con un título "Código de verificación:" y un botón de cerrar (X). El modal contiene un campo de entrada etiquetado "Código:" y un botón "Volver a enviar" en azul. En la parte inferior del modal hay dos botones: "Cerrar" y "Aceptar".

Código de verificación: X

Código:

[Volver a enviar](#)

Cerrar Aceptar

[Registrarse](#) [¿Olvidaste la contraseña?](#)

## Registro:

En la página de registro los pacientes podrán introducir todos sus datos para posteriormente recibir en su correo sus datos de inicio de sesión. Para poder registrarse es necesario ingresar todos los datos con un formato correcto además es obligatorio ser mayor de edad.

### Registrarse

DNI: \*

Nombre: \*

Apellidos: \*

Fecha de nacimiento: \*

Teléfono: \*

Dirección: \*

Correo electrónico: \*

[Aceptar](#) [Volver](#)

Al realizar el registro se mostrará un mensaje indicando que recibirá la información en su email.

Registro de usuario ➤ [Recibidos x](#)

**apihospital1@gmail.com**

para mí ▼

Usuario: osdf9396

Contraseña: QUVNndds

## Cambiar contraseña:

Si el usuario se ha olvidado de su contraseña en esta página podrá solicitar una nueva. Para ello lo único que tendrá que hacer es ingresar su dirección de correo y la nueva contraseña será enviada al email indicado.

### Recuperar contraseña

Dirección de correo:

### Recuperación de contraseña Recibidos x

apihospital1@gmail.com  
para mí ▼

Usuario: sa6468  
Contraseña: p1xV2JJL

## Páginas de los pacientes:

Los pacientes solo podrán acceder a determinadas páginas las cuales mostraremos a continuación:

### Home:

La página home es la página principal del paciente donde podrá ver las citas que tiene pendiente. En cada cita tendrá un enlace que llevará a una página donde se mostrarán los detalles de la cita.

Inicio	Historial	Recetas	Pedir cita	Perfil ▼
Fecha	Hora	Médico	Propósito	Propósito
2023-06-11	08:00:00	Médico1 Apellido1	Consulta de rutina	<a href="#">Ver cita</a>
2023-06-11	08:10:00	Médico2 Apellido2	Control de presión arterial	<a href="#">Ver cita</a>
2023-06-11	08:20:00	Médico3 Apellido3	Seguimiento de tratamiento	<a href="#">Ver cita</a>
2023-06-11	08:30:00	Médico4 Apellido4	Revisiones postoperatorias	<a href="#">Ver cita</a>
2023-06-11	08:40:00	Médico5 Apellido5	Consulta por síntomas respiratorios	<a href="#">Ver cita</a>
2023-06-11	08:50:00	Médico6 Apellido6	Control de diabetes	<a href="#">Ver cita</a>
2023-06-11	09:00:00	Médico7 Apellido7	Consulta por dolor abdominal	<a href="#">Ver cita</a>
2023-06-11	09:10:00	Médico8 Apellido8	Control de colesterol	<a href="#">Ver cita</a>
2023-06-11	09:20:00	Médico9 Apellido9	Seguimiento de enfermedad crónica	<a href="#">Ver cita</a>
2023-06-11	09:30:00	Médico10 Apellido10	Consulta de bienestar general	<a href="#">Ver cita</a>

## Ver cita:

Cuando el paciente accede a esta página se le mostrará los datos de la cita seleccionada.

**Datos de la cita:**

Fecha: 2023-06-11

Hora: 08:00:00

Proposito : Consulta de rutina

**Datos del médico:**

Nombre: Médico1 Apellido1

Hora: Cardiología

Consulta: Cons1

[Modificar cita](#) [Cancelar cita](#)

Debajo de los datos tendremos dos botones:

- **Modificar cita:** este botón mostrará un modal con un formulario para editar la información de la cita:

**Modificar cita** ×

Fecha:

11/06/2023 📅

Hora:

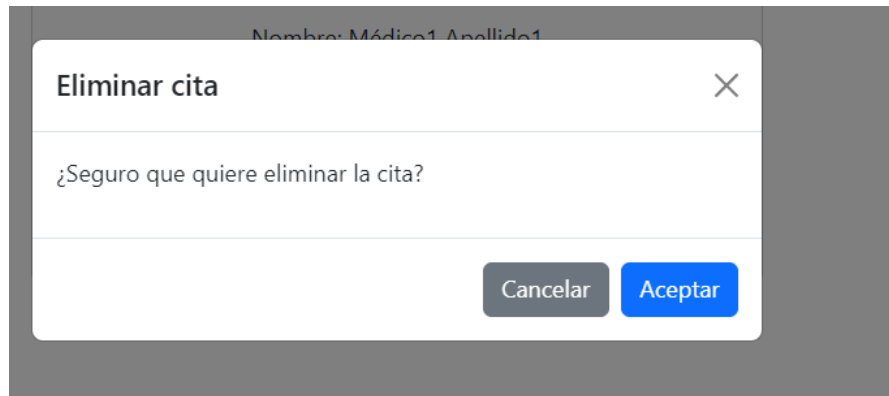
08:00 🕒

Hora:

Consulta de rutina

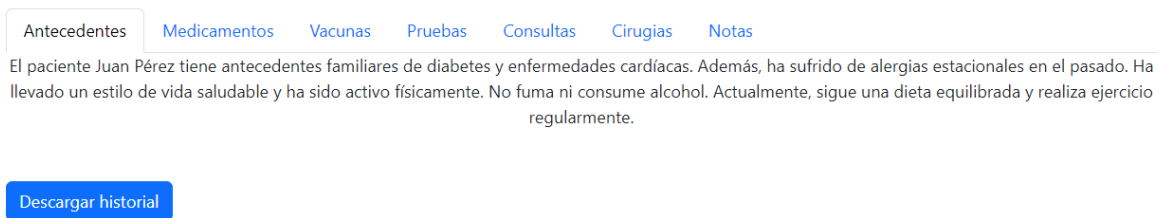
[Cancelar](#) [Aceptar](#)

- Cancelar cita: este botón nos permitirá cancelar la cita.

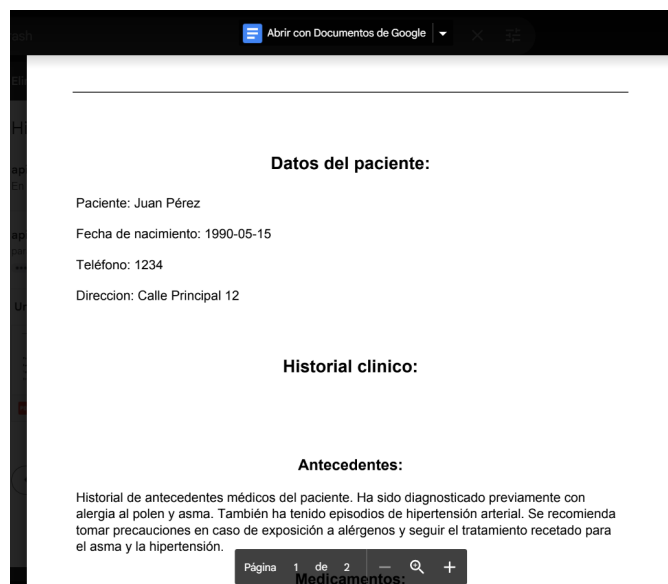


## Historial:

En la página del historial se mostrará los datos del historial del paciente.



Debajo de los datos tendremos un botón que permitirá recibir en el correo una copia en formato pdf del historial.

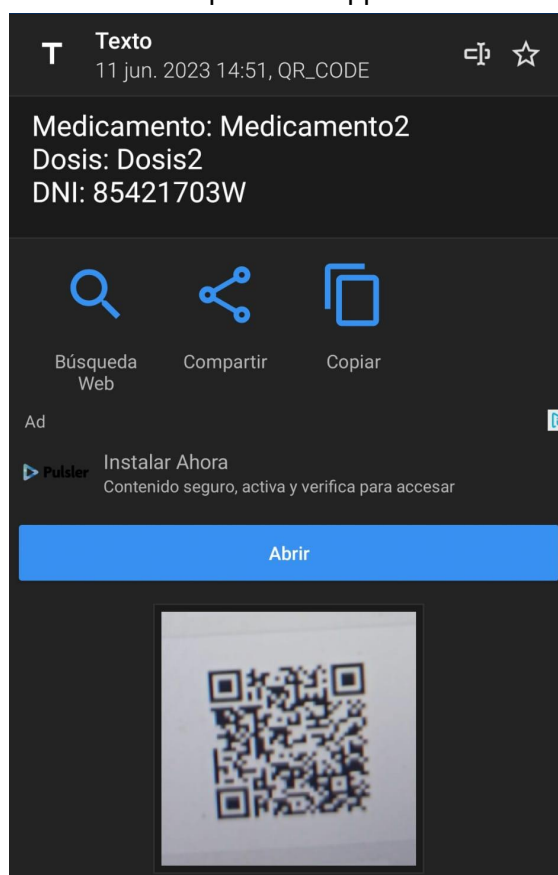


## Recetas:

En la página de recetas el paciente podrá ver las recetas que tiene.

Medicamento	Dosis	Frecuencia	Médico	Qr
Medicamento1	Dosis1	Frecuencia1	Médico1 Apellido1	
Medicamento2	Dosis2	Frecuencia2	Médico2 Apellido2	
Medicamento3	Dosis3	Frecuencia3	Médico3 Apellido3	
Medicamento4	Dosis4	Frecuencia4	Médico4 Apellido4	

Junto a cada receta se tendrá un código qr que permitirá ver la información de la receta. Es posible que el código no funcione en dispositivos apple.





## Pedir cita:

En la página pedir cita el paciente podrá pedir una cita introduciendo los datos en el formulario. Es importante tener en cuenta que la fecha no puede ser anterior a la actual y que el horario de citas es de 8 am a 8 pm.

Fecha:	<input type="text" value="dd/mm/aaaa"/>	
Hora:	<input type="text" value="--:--"/>	
Propósito:	<input type="text"/>	
Departamento:	<input type="text" value="Departamentos"/>	
<input type="button" value="Aceptar"/>		

Para poder seleccionar al médico previamente tendremos que seleccionar el departamento de la cita. Una vez seleccionado se mostrará un select con los médicos del departamento seleccionado.

## Perfil:

En el perfil el paciente podrá modificar sus datos o eliminar su cuenta.

Detalles de la cuenta:	Detalles del paciente:
Nombre de usuario: <input type="text" value="jupe1234"/>	DNI: <input type="text" value="85421703W"/>
Contraseña: <input type="password" value="*****"/>	Nombre: <input type="text" value="Juan"/>
	Apellidos: <input type="text" value="Pérez"/>
	Fecha de nacimiento: <input type="text" value="15/05/1990"/>
	Teléfono: <input type="text" value="1234"/>
	Dirección: <input type="text" value="Calle Principal 12"/>
	Email: <input type="text" value="juan@gmail.com"/>
<div><input type="button" value="Aceptar"/><input type="button" value="Eliminar cuenta"/></div>	

Para modificar un dato tendremos que hacer click en el botón de editar lo que habilitará el campo para poder introducir la nueva información.

Nombre:

Juan

✕

Apellidos:

Pérez

✎

Fecha de nacimiento:

## Páginas de los médicos:

Ahora se explicarán las páginas a las que solo pueden acceder los médicos, estas páginas comprobarán el rol del usuario, en caso de que no sea ni médico ni administrador se destruirá la sesión y será redirigido al login.

Home:

Es la página principal de los médicos, en esta página el médico podrá ver las citas que tiene que atender ese día.

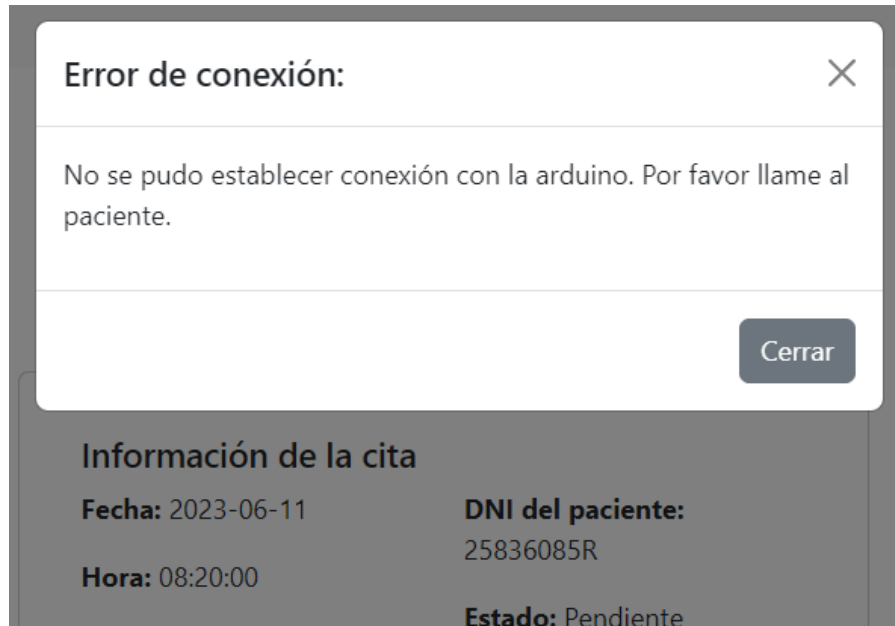
Atender Ver pacientes Ver médicos					
Fecha	Hora	Paciente	Proposito	Estado	
2023-06-11	08:00:00	Juan Pérez	Consulta de rutina	Pendiente	<a href="#">Atender</a>
2023-06-11	08:10:00	María López	Consulta de rutina	Pendiente	<a href="#">Atender</a>
2023-06-11	08:20:00	María García Pérez	Consulta de rutina	Pendiente	<a href="#">Atender</a>
2023-06-11	08:30:00	Juan López Rodríguez	Consulta de rutina	Pendiente	<a href="#">Atender</a>

El médico podrá atender de dos maneras:

- Seleccionando la cita que quiere atender.
- Accediendo a la página de atender desde la barra de navegación lo que le mostrará la primera cita pendiente de ese día.

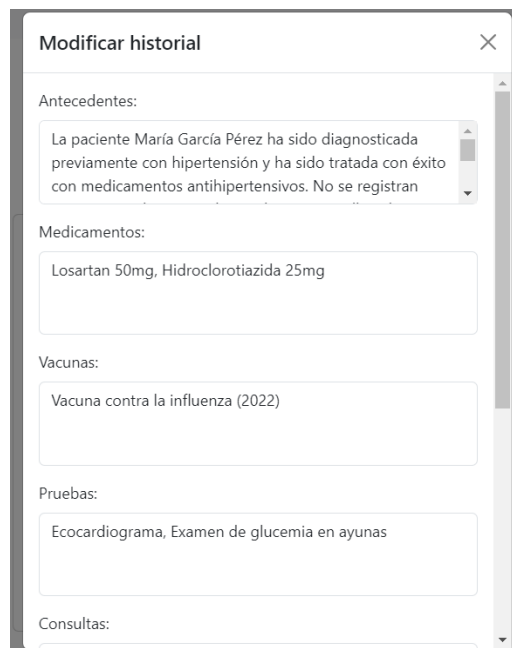
## Atender:

En esta página se mostrará la información de la cita. Al cargar la página se intentará conectar a la arduino, en caso de no poder realizar la petición se mostrará un modal mostrando el error.



Debajo de los datos tendremos varios botones que nos permitirán realizar las siguientes acciones:

- Cambiar el estado de la cita: con los botones de atendido o no presentado podremos cambiar el estado de la cita y pasar a la siguiente cita.
- Ver historial: abrirá un modal con los datos del historial del paciente lo que nos permitirá modificar dichos datos.



- Ver recetas: con este botón se abrirá un modal con un listado de las recetas del paciente:

Medicamento	Dosis	Frecuencia	
Medicamento1	Dosis1	Frecuencia1	Eliminar
Medicamento2	Dosis2	Frecuencia2	Eliminar
Medicamento3	Dosis3	Frecuencia3	Eliminar
Medicamento4	Dosis4	Frecuencia4	Eliminar

Cerrar Crear Receta

Dicho modal nos permitirá eliminar una receta o crear una:

Medicamento

Dosis

Frecuencia

Cerrar Aceptar

## Ver pacientes:

En esta página el médico podrá consultar los datos de los pacientes a los que ha atendido:

Paciente	Fecha de nacimiento	Teléfono	Dirección	Email	
Juan Pérez	1990-05-15	1234	Calle Principal 12	juan@gmail.com	Ver detalles
María López	1985-12-02	1234	Avenida Central 6	maria@gmail.com	Ver detalles
María García Pérez	1980-05-12	612345678	Calle Mayor, 123, Madrid	4abcgarciaoscar@gmail.com	Ver detalles
Juan López Rodríguez	1975-09-18	623456789	Avenida del Parque, 45, Barcelona	4abcgarciaoscar@gmail.com	Ver detalles

Al lado de cada paciente hay un botón que permite abrir un modal donde se podrá consultar los detalles del usuario junto a su historial.

Detalles del Registro

**Nombre:**  
Juan

**Apellido:**  
Pérez

**Fecha de nacimiento:**  
1990-05-15

**Teléfono:**  
1234

**Dirección:**  
Calle Principal 12

**Email:**  
juan@gmail.com

**Antecedentes:**  
El paciente Juan Pérez tiene antecedentes familiares de diabetes y enfermedades cardíacas. Además, ha sufrido de

Close

## Ver médicos:

Esta página solo está disponible para administradores. En dicha página se puede ver los distintos departamentos junto a los médicos que pertenecen a ese departamento.

[Insertar médico](#)
[Insertar departamento](#)

Departamento		Dni	Nombre	Teléfono	Dirección de correo	Consulta	
Cardiología	<a href="#">Detalles</a>	11373919M	Médico1 Apellido1	123456789	4abcgarciaoscar@gmail.com	Cons1	<a href="#">Modificar</a>
		75228739V	Médico2 Apellido2	234567890	4abcgarciaoscar@gmail.com	Cons2	<a href="#">Modificar</a>
Neurología	<a href="#">Detalles</a>	64610917F	Médico3 Apellido3	345678901	4abcgarciaoscar@gmail.com	Cons3	<a href="#">Modificar</a>
		83904441G	Médico4 Apellido4	456789012	4abcgarciaoscar@gmail.com	Cons4	<a href="#">Modificar</a>
Pediatria	<a href="#">Detalles</a>	50221124A	Médico5 Apellido5	567890123	4abcgarciaoscar@gmail.com	Cons5	<a href="#">Modificar</a>
		43289746C	Médico6 Apellido6	678901234	4abcgarciaoscar@gmail.com	Cons6	<a href="#">Modificar</a>
Oncología	<a href="#">Detalles</a>	42402437M	Médico7 Apellido7	789012345	4abcgarciaoscar@gmail.com	Cons7	<a href="#">Modificar</a>
		98097850K	Médico8 Apellido8	890123456	4abcgarciaoscar@gmail.com	Cons8	<a href="#">Modificar</a>

En esta página hay varios botones que nos permitirán realizar las siguientes acciones:

- Insertar departamento: este botón abrirá un modal con un formulario que permitirá introducir la información del nuevo departamento. Es importante tener en cuenta que la ip de la arduino debe ser una ip válida.

ID	Nombre	IP	Email	Departamento
11373				
75228				
64610917F	Médico3 Apellido3	345678901	4abcgarciaoscar@gmail.com	Cor
83904441G	Médico4 Apellido4	456789012	4abcgarciaoscar@gmail.com	Cor

- Insertar médico: los médicos no pueden registrarse por su cuenta, para poder registrar un médico se debe ser administrador. Para poder realizar esta acción se abrirá un modal con el formulario necesario. Al terminar el registro el médico recibirá por correo los datos de inicio de sesión.

- Ver departamento: este botón mostrará la información del departamento seleccionado. En dicho modal encontraremos un botón de modificar lo que habilitará los inputs para poder introducir la información y otro de eliminar lo que permitirá eliminar el departamento siempre y cuando no tenga ningún médico.



The modal titled "Detalles del departamento" displays the following information:

- Departamento:** Cardiológia
- Ip de la arduino:** 192.168.1.177
- Buttons:** "Borrar" (red) and "Modificar" (blue).

- Modificar médico: para poder cambiar la consulta, el departamento o el rol de un médico tendremos que usar este botón lo que abrirá un modal con la información a cambiar. Es importante saber que no se podrá cambiar la información del usuario que ha accedido.



The modal titled "Modificar médico" displays the following information:

- Consulta:** Cons2
- Departamento:** Cardiológia (dropdown menu)
- Rol:** medico (dropdown menu with a refresh icon)
- Buttons:** "Cerrar" (grey) and "Enviar" (blue).

## Perfil:

En el perfil el médico podrá modificar sus datos o eliminar su cuenta.

### Detalles de la cuenta:

Nombre de usuario:

meap1234

Contraseña:

\*\*\*\*\*

### Detalles del médico:

DNI:

11373919M

Nombre:

Médico1

Apellidos:

Apellido1

Teléfono:

123456789

Email:

4abcgarciaoscar@gmail.com

Aceptar

Eliminar cuenta

Para modificar un dato tendremos que hacer click en el botón de editar lo que habilitará el campo para poder introducir la nueva información.

Nombre:

Médico1

Apellidos:

Apellido1

Teléfono:



## Casos de uso:

### Login:

Caso de uso: Inicio de sesión.
Descripción: Este caso de uso permite al usuario iniciar sesión y acceder a la aplicación.
Precondición: Estar registrado en la aplicación.
Actores principales: Usuario. Actores secundarios: aplicación y API.
Curso normal: <ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación.</li> <li>2. Introduce sus datos.</li> <li>3. La aplicación valida los datos.</li> <li>4. La aplicación envía los datos.</li> <li>5. La api verifica los datos en la base de datos.</li> <li>6. La api comprueba el rol del usuario</li> </ol>
Alternativas: <ol style="list-style-type: none"> <li>6.1 El rol del usuario es paciente.             <ol style="list-style-type: none"> <li>1. La api envía una respuesta.</li> <li>2. La aplicación recibe la respuesta.</li> <li>3. La aplicación comprueba el rol del usuario.</li> <li>4. La aplicación redirige al usuario a su página Home.</li> <li>5. Termina el caso de uso.</li> </ol> </li> <li>6.2 El rol del usuario es médico o admin.             <ol style="list-style-type: none"> <li>6. La api genera un código de verificación.</li> <li>7. La api guarda el código en la base de datos</li> <li>8. La api envía un email a la dirección del usuario.</li> <li>9. La api envía una respuesta a la aplicación.</li> <li>10. La aplicación recibe la respuesta.</li> <li>11. La aplicación muestra una ventana para introducir el código.</li> <li>12. El usuario introduce el código.</li> <li>13. La aplicación envía el código a la api.</li> <li>14. La api verifica que el código coincide con el generado.</li> <li>15. La api envía una respuesta a la aplicación.</li> <li>16. La aplicación recibe la respuesta.</li> <li>17. La aplicación comprueba la respuesta.</li> <li>18. La aplicación redirige al usuario a su Home.</li> <li>19. Termina el caso de uso.</li> </ol> </li> <li>2.1 El usuario escribe mal los datos.             <ol style="list-style-type: none"> <li>1. La aplicación valida los datos.</li> <li>2. La aplicación muestra un mensaje de error.</li> <li>3. Termina el caso de uso.</li> </ol> </li> </ol>

<p>5.1 El usuario y la contraseña no coinciden.</p> <ol style="list-style-type: none"> <li>1. La api envía una respuesta negativa.</li> <li>2. La aplicación recibe la respuesta.</li> <li>3. La aplicación comprueba la respuesta.</li> <li>4. La aplicación muestra un mensaje de error.</li> <li>5. Termina el caso de uso.</li> </ol>
<p>Postcondiciones:</p> <p>En la sesión quedarán guardados los datos del usuario</p>

## Atender cita:

Caso de uso: Atender cita.
<p>Descripción:</p> <p>Este caso de uso permite al médico atender una cita.</p>
<p>Precondición:</p> <p>Haber iniciado sesión con un usuario cuyo rol sea médico o administrador.</p>
<p>Actores principales: Usuario.</p> <p>Actores secundarios: aplicación, API y arduino.</p>
<p>Curso normal:</p> <ol style="list-style-type: none"> <li>1. El usuario accede a la página.</li> <li>2. La aplicación realiza una petición a la ap con la fecha o un idi.</li> <li>3. La api comprueba el token.</li> <li>4. La api consulta los datos en la base de datos.</li> <li>5. La api realiza una petición a la arduino.</li> <li>6. La arduino recibe los datos.</li> <li>7. La arduino maneja los datos.</li> <li>8. La arduino muestra los datos.</li> <li>9. La arduino manda una respuesta.</li> <li>10. La api recibe la respuesta.</li> <li>11. La api envía una respuesta a la aplicación.</li> <li>12. La aplicación muestra los datos</li> <li>13. Termina el caso de uso.</li> </ol>
<p>Alternativas:</p> <p>3.1 El rol del usuario es paciente.</p> <ol style="list-style-type: none"> <li>1. La api envía una respuesta negativa.</li> <li>2. La aplicación recibe la respuesta.</li> <li>3. La aplicación comprueba la respuesta.</li> <li>4. La aplicación cierra la sesión del usuario.</li> <li>5. Termina el caso de uso.</li> </ol> <p>4.1 No se encuentran datos de la cita.</p> <ol style="list-style-type: none"> <li>1. La api envía una respuesta negativa.</li> <li>2. La aplicación recibe la respuesta.</li> <li>3. La aplicación comprueba la respuesta.</li> <li>4. La aplicación muestra el error.</li> <li>5. Termina el caso de uso.</li> </ol>

Caso de uso: Atender cita.
<p>5.1 La api no recibe respuesta.</p> <ol style="list-style-type: none"> <li>1. La api indica en la respuesta el estado de la arduino.</li> <li>2. La api envía la respuesta.</li> <li>3. La aplicación recibe la respuesta.</li> <li>4. La aplicación comprueba la respuesta.</li> <li>5. La aplicación muestra un mensaje de error de conexión.</li> <li>6. La aplicación muestra los datos de la cita.</li> <li>7. Termina el caso de uso.</li> </ol>
<p>Postcondiciones:</p> <p>En la página se mostrarán los datos de la cita.</p>

## Eliminar cuenta:

Caso de uso: Eliminar cuenta.
<p>Descripción:</p> <p>Este caso de uso permite al usuario eliminar los datos de su cuenta.</p>
<p>Precondición:</p> <p>Haber iniciado sesión con un usuario.</p>
<p>Actores principales: Usuario.</p> <p>Actores secundarios: aplicación y API.</p>
<p>Curso normal:</p> <ol style="list-style-type: none"> <li>1. El usuario accede a la página de su perfil.</li> <li>2. El usuario hace click en el botón de eliminar.</li> <li>3. La aplicación muestra una ventana de confirmación.</li> <li>4. El usuario hace click en aceptar.</li> <li>5. La aplicación realiza una petición a la api.</li> <li>6. La api realiza una consulta a la base de datos.</li> <li>7. La api envía una respuesta a la aplicación.</li> <li>8. La aplicación recibe la respuesta.</li> <li>9. La aplicación comprueba la respuesta.</li> <li>10. La aplicación abre una ventana indicando que los datos fueron eliminados.</li> <li>11. El usuario cierra la ventana.</li> <li>12. La aplicación destruye la sesión.</li> <li>13. La aplicación redirige al usuario al login.</li> <li>14. Termina el caso de uso.</li> </ol>
<p>Alternativas:</p> <p>6.1 La consulta no se realiza correctamente.</p> <ol style="list-style-type: none"> <li>6. La api envía una respuesta negativa.</li> <li>7. La aplicación recibe la respuesta.</li> <li>8. La aplicación comprueba la respuesta.</li> <li>9. La aplicación muestra el mensaje de error.</li> <li>10. Termina el caso de uso.</li> </ol>
<p>Postcondiciones:</p>

Los datos del usuario se habrán eliminado de la base de datos.

## 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.

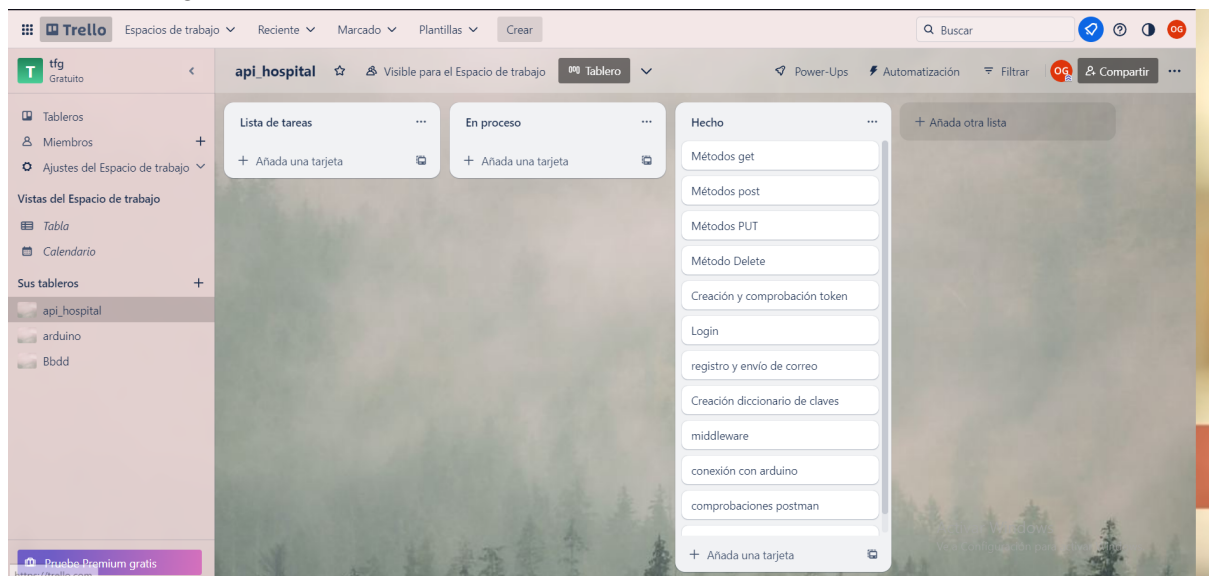
### Git/Github:

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>

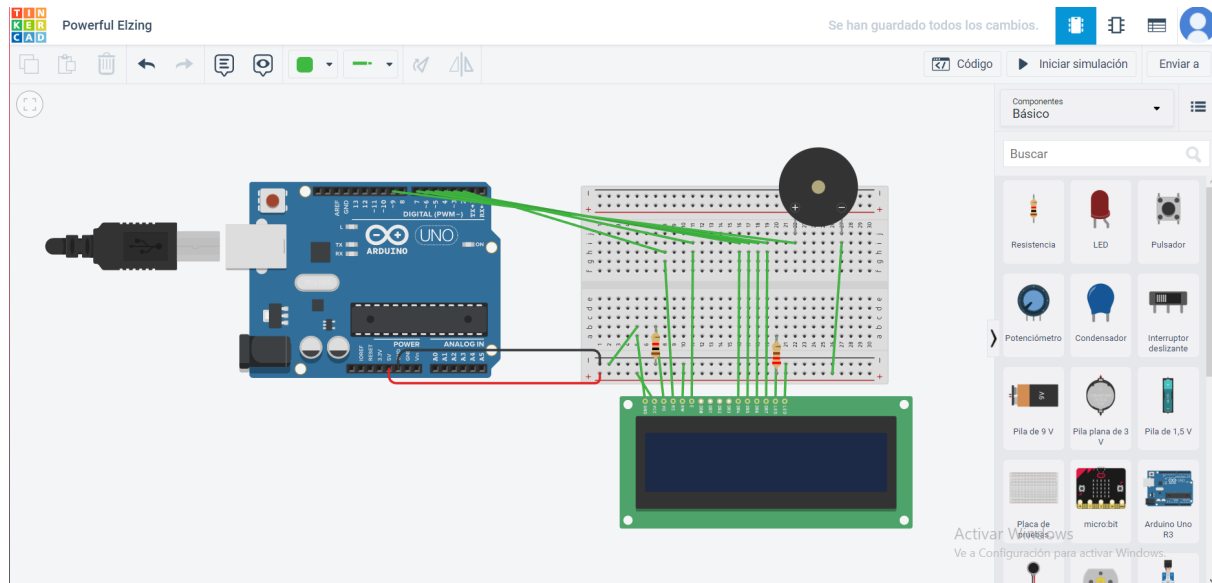
### Trello.com:

[Trello](https://trello.com) 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.

### TempMail:

[TempMail](#) es una aplicación web que permite generar direcciones de correo temporales lo cual ha permitido probar todas las funcionalidades que requieren un email de la aplicación.

### DOI:

[Doi](#) es una aplicación web que permite generar datos de ejemplo, en nuestro caso ha permitido generar diferentes DNI para el registro tanto de médicos como de pacientes.