# Viernes Care Covid

## descripción

Viernes Care Covid es una aplicación web destinada a la gestión de ciudadanos infectados de Coronavirus. Su objetivo es llevar un registro de las personas infectadas así como un seguimiento de su estado durante el desarrollo de la enfermedad hasta su curación o deceso.

Para ello, la aplicación pone en manos del personal sanitario una serie de herramientas que van a facilitar su labor reduciendo los tiempos de gestión administrativa y aumentando la eficacia de sus funciones en el seguimiento de los infectados.

La herramienta está destinada a los profesionales sanitarios encargados del rastreo de contagiados y a los médicos responsables del tratamiento y seguimiento de la enfermedad en los pacientes. Para ello, pone a su disposición una interfaz sencilla que les permite añadir notas firmadas, acompañadas de fecha y estado del paciente.

La aplicación también permite a los pacientes obtener un informe de su estado sin tener que contactar con los servicios sanitarios, lo que redunda en una reducción de efectivos y tiempo en consultas de pacientes.

## características

La herramienta presenta una interfaz de validación común para usuarios (personal sanitario y administradores de sistema) y para pacientes.

La interfaz de paciente permite las consultas directas de los pacientes mediante una clave proporcionada al mismo y es totalmente independiente de la interfaz de usuarios.

La interfaz de usuario permite tres roles:

* Administrador: destinado a la gestión del usuarios y mantenimiento de las bases de datos.
* Rastreador: para los profesionales encargados de buscar infectados, y dispone de la capacidad de crear nuevos pacientes en el sistema, indicando su estado.
* Médico: para los especialistas de atención primaria, permite llevar un seguimiento de los pacientes del sistema, pudiendo actualizar su estado.

La herramienta hace uso de dos bases de datos independientes:

* Base de datos de pacientes: Se encuentra en un servidor externo al del alojamiento de la herramienta.
* Base de datos de usuarios: en el servidor de la herramienta, se ocupa de los datos del personal sanitario y administradores.

### Tecnologías

La herramienta hace uso de las siguientes tecnologías:

* HTML
* CSS, BOOTSTRAP
* Javascript
* PHP
* MYSQL

## instalación de la herramienta

La aplicación se proporciona en un archivo comprimido. Una vez descomprimido, podrá observar una serie archivos y carpetas. De estas últimas, en “*SERVIDOR EXTERNO”* encontrará los archivos que deben ser enviados al servidor y en *“documenacion”* están tanto este documento así como el resto de información que le pueden ser útil, tanto para la compresión de la herramienta como para su mantenimiento, mejora o reparación.

### Servidor externo

Se recomienda un equipo con sistema operativo Linux, donde deberemos Debemos tener instalado Apache como servidor web, PHP, y MySQL o MariaDB como gestor de bases de datos.

#### Instalación de Apache y PHP

Esto lo podemos hacer con la siguiente serie de comandos:

sudo apt-get update

sudo apt-get upgrade

sudo apt install apache2

Los dos primeros actualizan el sistema y el tercero se ocupa de instalar el Apache 2. Para verificar que se ha instalado correctamente, podemos ejecutar sudo systemctl status apache2.

Para la instalación de PHP:

sudo apt install php libapache2-mod-php php-mysql

A continuación necesitamos dar prioridad en Apache a la apertura de archivos PHP. Para ello editamos el siguiente archivo dir.conf con:

sudo nano /etc/apache2/mods-enabled/dir.conf

Y sustituimos la línea

DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm

por

DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm

Guardamos y cerramos el archivo y reiniciamos el Apache con:

sudo systemctl restart apache2.

Una vez hecho esto procedemos a cargar los archivos.

Al instalar el Apache debe crearse una carpeta donde podemos descargar nuestro proyecto. Algo similar a /var/www/html/. Dentro de aquí creamos nuestro directorio, que debe llamarse “**covid**”. Dentro de este directorio copiaremos los archivos:

* Config.php
* Serv\_pac.php
* Serv\_usu.php
* Utils.php

#### Instalación de MariaDB

El gestor de bases de datos que usamos es MariaDB:

sudo apt install mariadb-server

Una vez instalada, procedemos a cargar nuestras bases de datos:

Vamos al directorio donde está nuestra base de datos:

Cd /var/www/html/covid/

Y accedemos al servidor mediante:

sudo mysql -uroot -p

no requiriéndose contraseña. A continuación ejecutamos los siguientes comandos para instalar la base de datos de prueba y los privilegios de usuario.

Source covid\_paciente.sql;

Source usuarios.sql;

Para eliminar los datos de prueba de la base de datos de pacientes, ejecutamos los siguientes comandos.

Truncate table nota;

Truncate table paciente;

Salimos del gestor ejecutando exit.

Esto debe dejar listo nuestro servidor para trabajar.

Antes de salir debemos anotar la ip de nuestro servidor, que podemos recuperar con el comando:

Ifconfig

Dependiendo si la conexión es inalámbrica o por cable anotamos la ip que acompaña a la línea “*inet”*.

### Servidor local

En nuestro servidor local, el procedimiento es similar al del externo, instalando igualmente Apache, PHP y MariaDB.

En este caso, en la carpeta /var/www/html/, volvemos a crear el directorio con el nombre que deseemos que tenga nuestra página, por ejemplo: ***“covid”***, y dentro de él copiamos el todos los archivos y carpetas descomprimidos, omitiendo *“Servidor externo”* que aquí no es necesario y por seguridad, conveniente que no esté.

En este caso, la base de datos que debemos instalar se encontrará en /var/www/html/covid/bdd/. En este directorio entramos en MySQL y ejecutamos:

Source covid\_usuario.sql;

Igualmente, para limpiar la base de datos de los registros de prueba, lo podemos hacer mediante:

Truncate table user;

#### Configuración

Para la correcta comunicación con el servidor externo, debemos hacer dos modificaciones:

En el archivo ./bddsx/config.php debemos cambiar el valor de la variable $ser\_ext, para ello eliminamos o comentamos cualquier definición que haya e incluimos la siguiente línea:

$ser\_ext=’http://xxx.xxx.xxx.xxx/covid/’;

Donde xxx.xxx.xxx.xxx es la ip del servidor externo que anotamos con anterioridad.

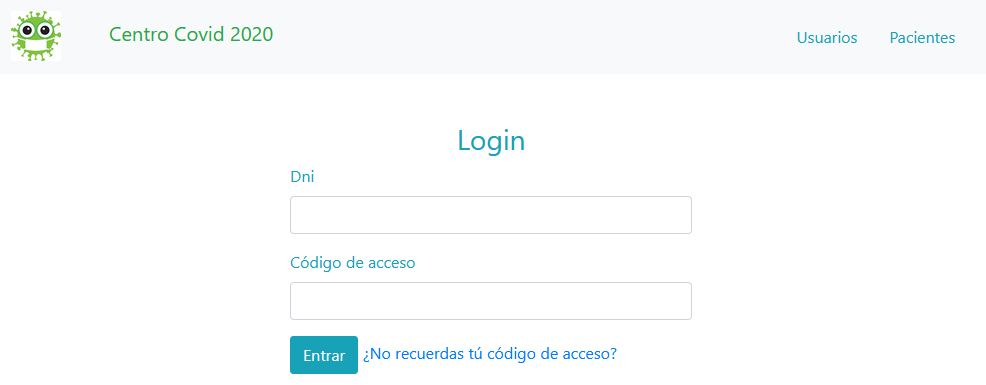
En caso de que el servidor externo esté en otra red, debemos sustituir esta dirección IP por el dominio en el cual se encuentre.

Igualmente, en el archivo ./js/cliente.js debemos hacer lo mismo con la variable ser\_ext:

ser\_ext=’http://xxx.xxx.xxx.xxx/covid/’;

## Descripción de funcionamiento

La página de entrada a la herramienta es un formulario de validación



Por defecto la página de acceso nos pide *DNI* y *Código de acceso*. Este es el acceso para los pacientes.

Si deseamos acceder como usuarios, pinchamos en la etiqueta *“Usuarios”*, y en este caso, se nos pedirá Email del sanitario y su contraseña.

### Acceso como Paciente

Para acceder como ciudadano (y si no se nos pide el DNI) debemos pinchar en la etiqueta *“Pacientes”* de la parte superior derecha.

Debemos introducir los siguientes datos para acceder:

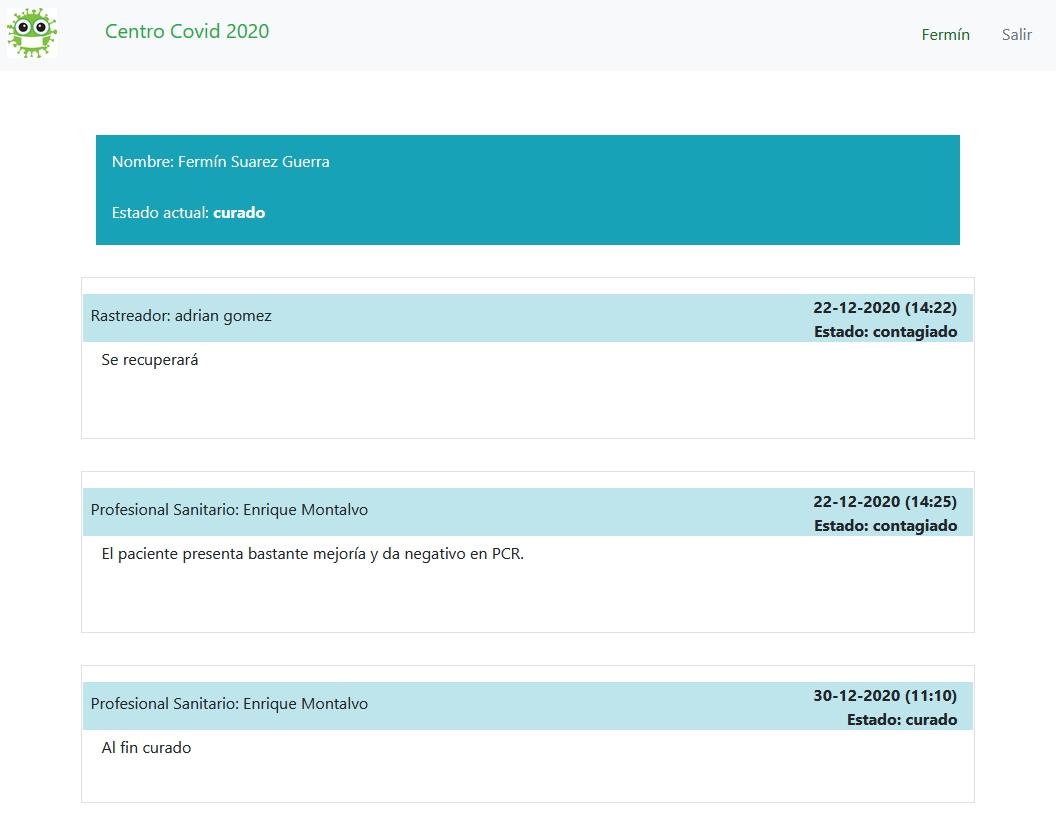
* **DNI:** con los números y letras seguidos, sin separar y sin guiones. Se aceptan numeraciones de españoles y de extranjeros con formatos *DDDDDDDDL* y *LDDDDDDDL*, donde *D* representa un dígito y *L* una letra.
* **Código de Acceso:** es una cadena de 8 caracteres compuesto por números y letras minúsculas, generadas por el sistema al crear el paciente y que éste debe conocer para poder acceder a su historial. El código de acceso sería enviado por correo electrónico al paciente una vez introducidos sus datos en el sistema.

Una vez dentro, nos encontramos lo siguiente:

En la parte superior derecha encontramos el nombre del paciente y la pestaña para salir del sistema y borrar la sesión.

A continuación tenemos el nombre completo del paciente y su estado actual. Este estado corresponde con el último registrado en el sistema.

Por último, una relación de todas las notas registradas en el sistema de ordenadas por fecha de creación, indicando en la cabecera el perfil del sanitario y su nombre con un apellido, y a la derecha la fecha y hora de creación y el estado registrado en ese momento. Como cuerpo, se indica la nota registrada.



### Acceso como usuario

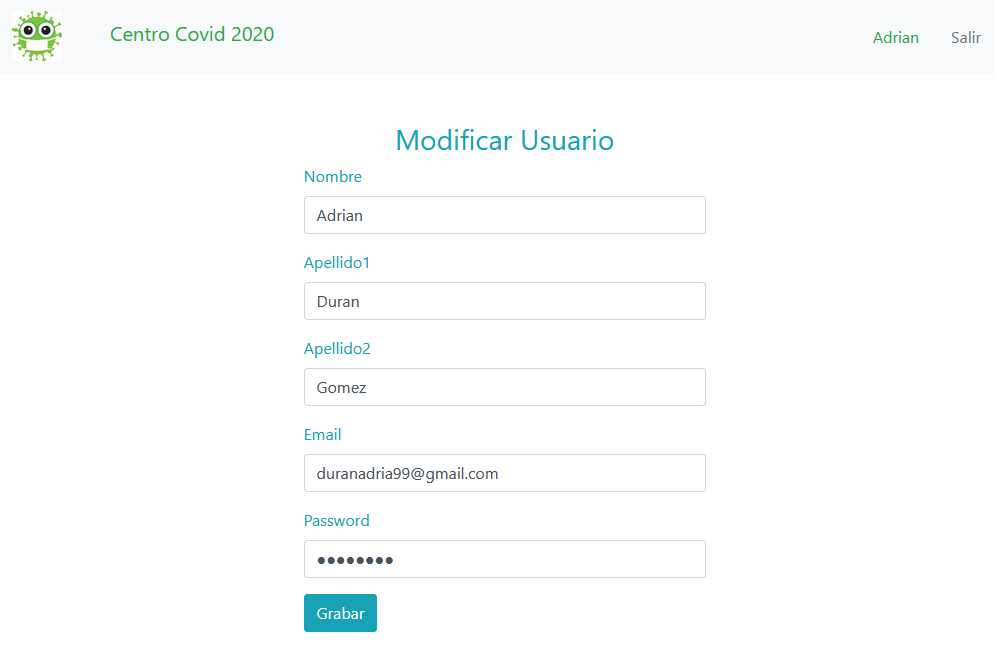
Según accedamos con un perfil de administrador, rastreador o médico se nos presentarán diferentes alternativas:

#### Administrador

En este caso, dispondrá de un listado de usuarios



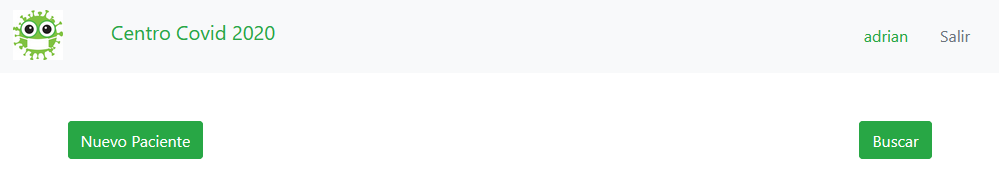
Igual que en la pantalla de paciente, en este caso, en la parte superior derecha encontramos el nombre del usuario, en este caso del administrador, y la etiqueta para salir del sistema. Sin embargo, aquí, pinchando en el nombre del usuario, éste puede modificar sus datos directamente a través de la siguiente pantalla:



Volviendo al listado de pacientes, la primera línea indica a que corresponde cada campo, y el primer formulario, con el botón verde, nos permite crear un nuevo usuario.

Los siguientes formularios muestran a cada uno de los usuarios y sus datos que pueden ser modificados en línea y pinchando “*modificar”* podemos actualizar el registro.

#### Rastreador

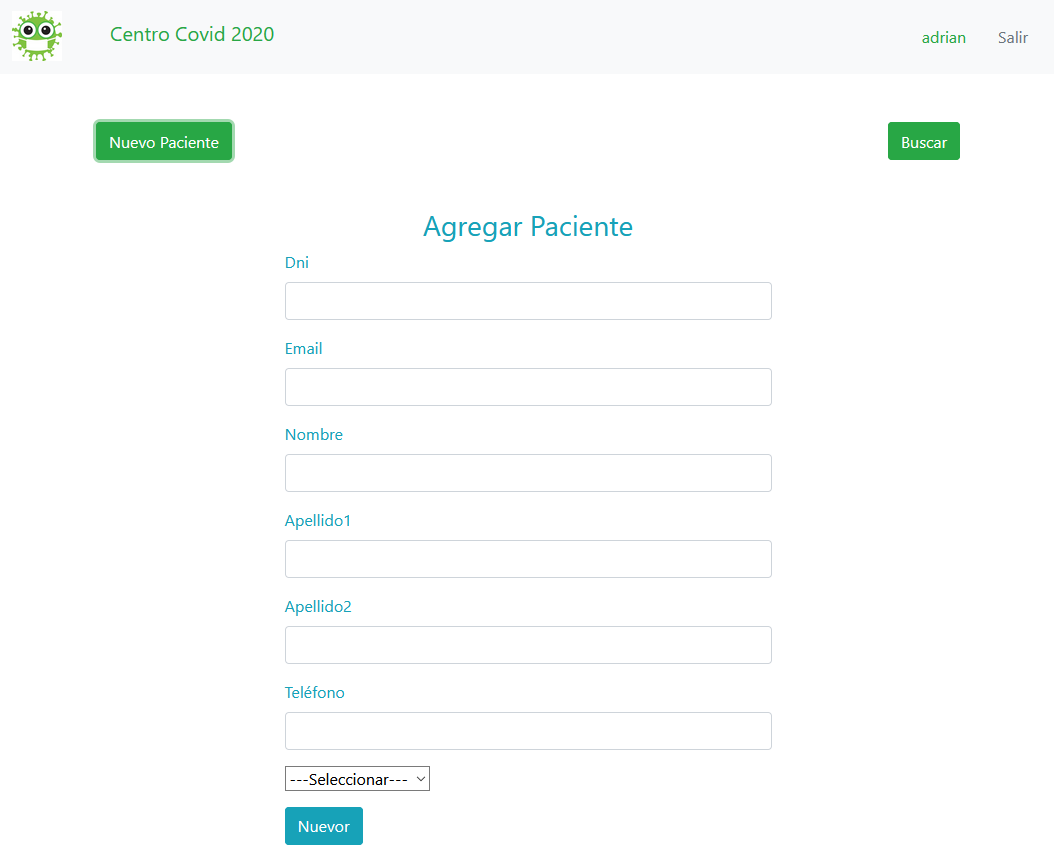


En la página de rastreador tenemos, al igual que antes, en la parte superior derecha una pestaña con el nombre del rastreador, que nos permite acceder al misma pantalla de modificación de sus datos, que el administrador, y a la pestaña *“Salir”* para desloguearse.

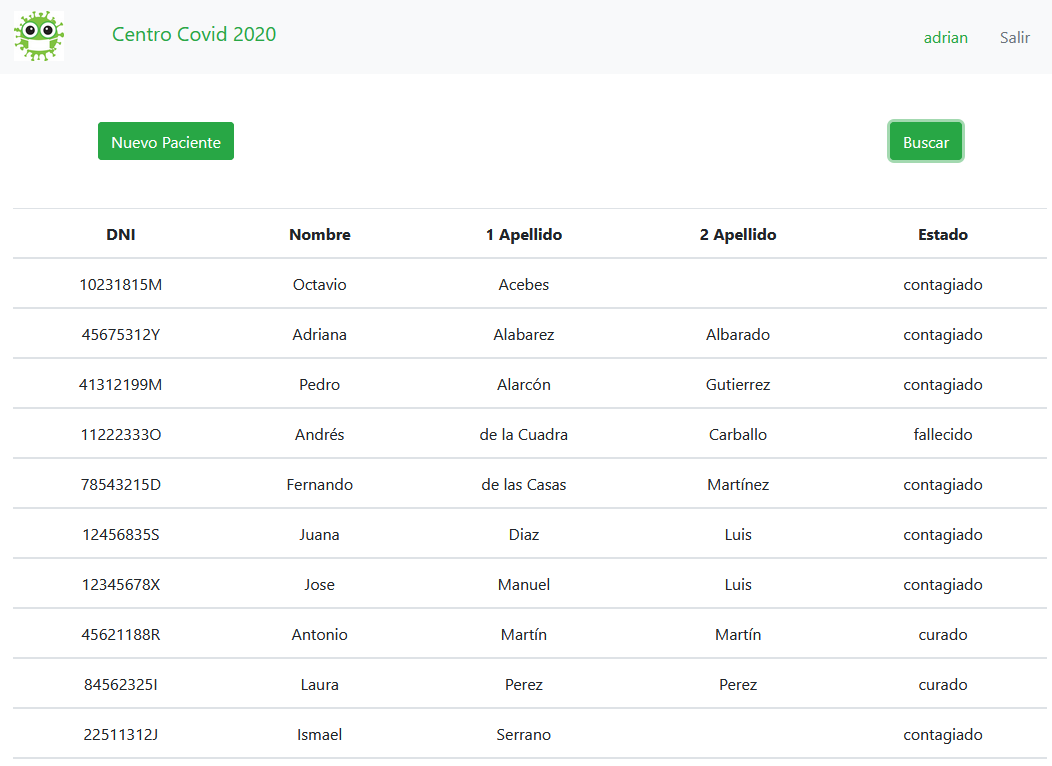
En el cuerpo de la página tiene dos opciones.

***Nuevo paciente***

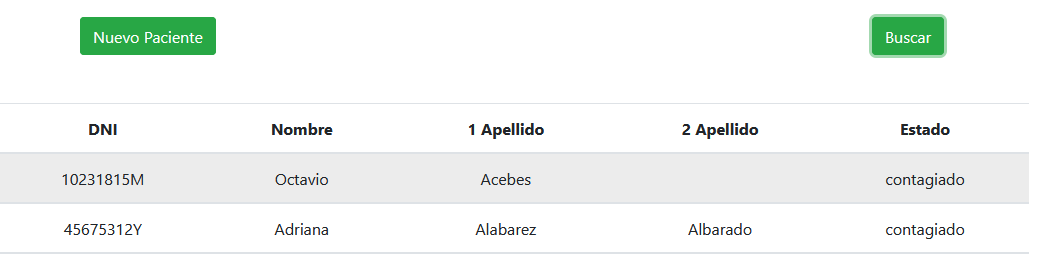
Donde podemos agregar todos los datos necesarios para introducir nuevos pacientes rastreados.



***Nuevo paciente***



Pinchando sobre el paciente, podemos acceder a su ficha y modificar sus datos.



#### Médico

## Diseño Front-End

El diseño consta de 9 páginas:

1. Ventana de bienvenida. Se entra al perfil de usuario o paciente
2. Ventana de validación paciente. Pide DNI (especificar formato) y código (indicar como puede obtener el código)
3. Datos del paciente. Muestra sus datos, su estado actual y las notas que ha introducido el médico.
4. Ventana de usuarios. Pide el email y su contraseña. También dispone de un link a otra ventana para solicitar datos y recuperar la contraseña.
5. Ventana del administrador. Dispondrá de un CRUD para la gestión de usuarios y una pestaña de listar usuarios.
6. Ventana de rastreador. Puede solicitar listados de pacientes, dar de alta pacientes y modificar dni, nombre, apellido 1, email y teléfono. Puede acceder a su perfil. Enviar email con código de acceso (se hace con un checkout al darlos de alta)
7. Ventana del médico. Puede solicitar listados de pacientes, ver su historial de notas y modificar estado y agregar notas. Cada nota del historial contiene la fecha de creación y el médico que la suscribe. Puede acceder a su perfil. Las solicitudes a administrador son referidas a notas que se envían al administrador para resolver problemas que con sus privilegios no puede, como modificar una nota, o solicitar el envío de un código.
8. Perfil de médico y rastreador. Para ver sus datos y modificar su contraseña.
9. Recuperación de contraseña. Pide el id y el DNI, (y pregunta secreta?)

## Bases de datos.

Falta el diagrama de Chen de las BBDD.

### Usuarios

#### Tabla usuario

Tiene los siguientes campos:

|  |  |  |  |
| --- | --- | --- | --- |
| Clave | tipo | Propiedad 1 | Propiedad 2 |
| Id\_usuario | Int | Primary key | Autoincremental |
| Nombre | Varchar(20) | Not Null |  |
| Apellido\_1 | Varchar(40) | Not Null |  |
| Apellido\_2 | Varchar(40) |  |  |
| Contraseña | Varchar(50) | Not Null |  |
| Email | Varchar(50) | Not Null |  |
| rol | Varchar(10) | Not Null |  |

El usuario puede tener varios roles, y la forma más sencilla es metiendo tres registros bool en cada uno de ellos para indicar cuales tiene.

#### Tabla notas administrador

Es una tabla opcional para incidencias dirigidas por los usuarios al administrador. Tiene su id único, la incidencia, quién la remite, la fecha y el estado que en principio es 0 (no resuelto) y el administrador puede pasar a 1(resuelto) para que no salte como alerta.

|  |  |  |  |
| --- | --- | --- | --- |
| Clave | tipo | Propiedad 1 | Propiedad 2 |
| Id\_solicitud | Int | Primary key | Autoincremental |
| Incidencia | Varchar | Not Null |  |
| Id\_usuario | int | Not Null |  |
| Fecha | datetime | Not Null |  |
| estado | Bool | Not Null |  |

### Pacientes

#### Tabla paciente

Consta de:

|  |  |  |  |
| --- | --- | --- | --- |
| Clave | tipo | Propiedad 1 | Propiedad 2 |
| DNI | Varchar(9) | Primary key |  |
| Codigo\_acceso | Varchar(8) | Not Null | unique |
| Email | Varchar(100) | Null |  |
| Nombre | Varchar(100) | Not Null |  |
| Apellido\_1 | Varchar(100) | Not Null |  |
| Apellido\_2, | Varchar(100) | Null |  |
| telefono | Varchar(12) | Not Null |  |
| estado | Varchar(12) |  |  |

#### Tabla nota

|  |  |  |  |
| --- | --- | --- | --- |
| Clave | tipo | Propiedad 1 | Propiedad 2 |
| Id | Int | Primary key | autoincremental |
| dni\_paciente | Varchar(9) | Not Null |  |
| fecha | datetime | Not Null |  |
| nota | Vachar(6000) | Not Null |  |
| Id\_doctor | Int | Not Null |  |

#### Perfiles

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Contraseña | Privilegios | |
| Datos | Estructura |
| paciente | paciente | Select |  |
| usuario | usuario | select, insert, update |  |
| admin | admin | select, insert, update | drop |

## Contraseñas DE rASPBERRY

Servidor pacientes:

Grupo: viernes\_care

User: usuario

Password: UsuVC

User: admin

Password: AdminVC

## Servicios

### Pacientes

1. Datos paciente: consulta que hace un paciente de sus datos (GET)
   1. Recibe: DNI, código\_acceso
   2. Devuelve: nombre, apellido\_1, estado, notas ordenadas por fecha
2. Nueva alta: alta de paciente (POST: alta)
   1. Recibe: nombre, apellido, apellido\_2, dni, email, teléfono, estado, id\_usuario
   2. Devuelve: nada
3. Lista pacientes: listado de pacientes, completo o filtrado (contagiado, fallecido y curado) (GET)
   1. Recibe: filtro (estado)
   2. Devuelve: listado con DNI, código\_acceso, nombre, apellido\_1, dni, teléfono, email, estado, numero de notas.
4. Datos paciente rastreador: (GET)
   1. Recibe: dni
   2. Devuelve: nombre, apellido\_1, apellido\_2, dni, email, teléfono
5. Modificar paciente rastreador (PUT)
   1. Recibe: nombre, apellido\_1, dni, email, teléfono
   2. Devuelve: nada
6. Historial paciente: (GET)
   1. Recibe: dni
   2. Devuelve: nombre, apellido\_1, email, teléfono, estado, listado de notas (nombre\_medico, fecha, nota)
7. Agregar nota (POST: nota)
   1. Recibe: dni, id\_usuario, nota, estado
   2. Devuelve: nada

## Diagrama de requisitos

Hay que preparar el diagrama de requisitos, más que nada por la batería de pruebas.