# Viernes Care Covid

## descripción

## características

Empresa: Viernes Care

## VC

## Consultas

* Clave de 20 caracteres para consumir servicios: un código único para hacer las consultas.
* Cómo hacemos el GIT

## Tecnologías

Emplearemos

* HTML
* CSS, BOOTSTRAP
* JS
* PHP
* MYSQL
* Editor de código, VisualCode, SublimeText

## 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.