

DEFENSA EXAMEN H3

PROGRAMA III

NOMBRE: ROGER CAMACHO VALLEJOS

MATERIA: PROGRAMACION III

1.- CREAR EL ENTITY PARA EL MODELO CORONAVIRUSPACIENTE.

MODEL:

```
1 package com.CVPaciente.DefensaH3.MODEL;
2
3 import javax.persistence.*;
4
5 @Entity
6 @Table(name = "coronaVpacienteDefensa")
7 public class DCMModel {
8     @Id
9     @GeneratedValue(strategy = GenerationType.AUTO)
10     private Integer IDCoronaV;
11     @Column(name = "nombreDe", length = 50, nullable = false)
12     private String nombreDe;
13     @Column(name = "nombrePaciente", length = 50, nullable = false)
14     public String nombrePaciente;
15     @Column(name = "apellidosPaciente", length = 50, nullable = false)
16     public String apellidosPaciente;
17     @Column(name = "edadPaciente")
18     public Integer edadPaciente;
19     @Column(name = "categoria", length = 50, nullable = false)
20     public String categoria;
21     @Column(name = "fullname", length = 100, nullable = false)
22     public String fullname;
23     @Column(name = "casosContagiados")
24     private Integer casosContagiados;
25     @Column(name = "casosSospechosos")
26     private Integer casosSospechosos;
27     @Column(name = "casosRecuperados")
28     private Integer casosRecuperados;
29 }
```

Creacion de la Tabla
y las Columnas

SETTERS AND GETTERS

```
30 public Integer getIDCoronaV() { return IDCORONA; }
33
34 public void setIDCoronaV(Integer IDCORONA) { this.IDCORONA = IDCORONA; }
37
38 public String getNombreDe() { return nombreDe; }
41
42 public void setNombreDe(String nombreDe) { this.nombreDe = nombreDe; }
45
46 public String getNombrePaciente() { return nombrePaciente; }
49
50 public void setNombrePaciente(String nombrePaciente) { this.nombrePaciente = nombrePaciente; }
53
54 public String getApellidosPaciente() { return apellidosPaciente; }
57
58 public void setApellidosPaciente(String apellidosPaciente) { this.apellidosPaciente = apellidosPaciente; }
61
62 public Integer getEdadPaciente() { return edadPaciente; }
65
66 public void setEdadPaciente(Integer edadPaciente) { this.edadPaciente = edadPaciente; }
69
70 public String getCategoria() { return categoria; }
73
74 public void setCategoria(String categoria) { this.categoria = categoria; }
77
78 public String getFullname() { return fullname; }
81
82 public void setFullname(String fullname) { this.fullname = fullname; }
85
86 public Integer getCasosContagiados() { return casosContagiados; }
89
90 public void setCasosContagiados(Integer casosContagiados) { this.casosContagiados = casosContagiados; }
93
94 public Integer getCasosSospechosos() { return casosSospechosos; }
97
98 public void setCasosSospechosos(Integer casosSospechosos) { this.casosSospechosos = casosSospechosos; }
101
102 public Integer getCasosRecuperados() { return casosRecuperados; }
105
106 public void setCasosRecuperados(Integer casosRecuperados) { this.casosRecuperados = casosRecuperados; }
```

2.-GENERAR EL SERVICIO REST-POST PARA PODER CREAR UN NUEVO CASO INCLUIDOS CATEGORIA FULLNAME. REST-POST

```
@PostMapping("/coronaVirusPaciente")
public ResponseEntity save(@RequestBody DCVModel dcvModel){
    try {
        return new ResponseEntity(dcvService.save(dcvModel), HttpStatus.CREATED);
    }catch (Exception e){
        return new ResponseEntity( headers: null, HttpStatus.EXPECTATION_FAILED);
    }
}
```

CATEGORIA Y FULLNAME: DEFINIDOS EN EL SERVICE (SAVE)

```
18
19     @Override
20     public DCVModel save(DCVModel pModel) {
21         if (pModel.getEdadPaciente()>20){
22             pModel.categoria = "Adulto";
23         } else if (pModel.getEdadPaciente()<20 && pModel.getEdadPaciente()>10){
24             pModel.categoria = "Adolescente";
25         } else if (pModel.getEdadPaciente()<10){
26             pModel.categoria = "Niño";
27         }
28
29         pModel.fullname = pModel.nombrePaciente+" " + " "+pModel.apellidosPaciente;
30         return dcvRepo.save(pModel);
31     }
32
```

3.-CREAR LOS SERVICIOS PARA PODER LISTAR TODOS LOS PACIENTES Y EN SU CASO UNO SOLO.

REST-GET

```
61
62     @GetMapping("/coronaVirusPaciente/")
63     public ResponseEntity<List<DCVModel>> GetAllPaciente(){
64         try {
65             List<DCVModel> patients = dcvService.GetAllPaciente();
66             if (patients.isEmpty()){
67                 return new ResponseEntity<>(HttpStatus.NO_CONTENT);
68             }else {
69                 return new ResponseEntity<>(patients, HttpStatus.OK);
70             }
71         }catch (Exception e){
72             return new ResponseEntity<>(null, HttpStatus.INTERNAL_SERVER_ERROR);
73         }
74     }
75
```

```
76
77     @GetMapping("/coronaVirusPaciente/getOne/{IDCoronaV}")
78     public ResponseEntity<DCVModel> GetPacienteByIDCoronaV(@PathVariable("IDCoronaV") Integer IDCoronaV){
79         try {
80             DCVModel pModel = dcvService.GetPacienteByIDCoronaV(IDCoronaV);
81             if (pModel != null){
82                 return new ResponseEntity<>(pModel, HttpStatus.OK);
83             }else {
84                 return new ResponseEntity<>(HttpStatus.NOT_FOUND);
85             }
86         }catch (Exception e) {
87             return new ResponseEntity<>(null, HttpStatus.INTERNAL_SERVER_ERROR);
88         }
89     }
90
```

4.-CREAR UN SERVICIO REST-PUT QUE PERMITA MODIFICAR UN REGISTRO CVP REST-PUT

```
47 @PutMapping("/coronaVirusPaciente/{IDCoronaV}")
48 public ResponseEntity<DCVModel> update(@PathVariable("IDCoronaV") Integer IDCoronaV, @RequestBody DCVModel pacientModel){
49     try {
50         DCVModel pUdat = dcvService.update(pacientModel, IDCoronaV);
51         if (pUdat != null){
52             return new ResponseEntity<>(pUdat, HttpStatus.OK);
53         }else{
54             return new ResponseEntity<>(HttpStatus.NOT_FOUND);
55         }
56     }catch (Exception e){
57         return new ResponseEntity<>(null, HttpStatus.INTERNAL_SERVER_ERROR);
58     }
59 }
60
```


5.-EVITAR INSERTAR EN LA BASE DE DATOS NUEVOS CASOS CVP SI LA EDAD DE EL PACIENTE ES MAYOR 70. (DEBE DE CREAR OTRO SERVICIO REST POST)

```
29
30     @PostMapping("/coronaVirusPaciente1")
31     public ResponseEntity save(@RequestBody DCVModel cvModel){
32         try{
33             if (cvModel.getEdadPaciente()>70) {
34                 return new ResponseEntity( headers: null, HttpStatus.EXPECTATION_FAILED);
35             } else {
36                 return new ResponseEntity<>(dcvService.save(cvModel),HttpStatus.CREATED);
37             }
38         }catch (Exception e){
39             return new ResponseEntity( headers: null, HttpStatus.EXPECTATION_FAILED);
40         }
41     }
42
43
```

6.-CREAR UN SERVICIO REST-DELETE QUE ELIMINA TODOS LOS REGISTROS DE LA BASE DE DATOS.

```
54      @Override
55      public Integer delete(Integer IDCoronaV) {
56          dcvRepo.deleteAll();
57          return 1;
58      }
```

```
59
```

```
91     @DeleteMapping("/paciente/{IDCoronaV}")
92     public ResponseEntity<String> delete(@PathVariable("IDCoronaV") Integer IDCoronaV){
93         try {
94             dcvService.delete(IDCoronaV);
95             return new ResponseEntity<>( body: "DataBase sccesfully deleted", HttpStatus.OK);
96         }catch (Exception e){
97             return new ResponseEntity<>(null, HttpStatus.EXPECTATION_FAILED);
98         }
99     }
```