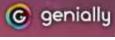
DEFENSA HITO 3

PROGRAMACION III - SPRING FRAMEWORK



ÍNDICE

01

SPRING FRAMEWORK

02

DEFENSA HITO3

03

CREAR EL MODELO
"CORONA VIRUS PACIENTE"

04

GENERAR UN SERVICIO REST-POST

05

LISTAR TODOS LOS PACIENTES Y UNO SOLO REST-GET 06

CREAR SERVICIO REST PUT PERMITA MODIFICAR

07

EVITAR LA INSERCION DE PACIENTES MAYORES A70 MAYOS

08

CREAR UN REST- DELETE QUE ELIMINE TODOS LOS REGISTROS

09

GRACIAS





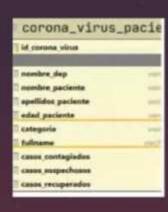


PROBLEMAS

DEFENSA HITO3

CREAR EL MODELO "CORONA VIRUS PACIENTE"

creara la tabal de nombre"corona virus paciente" con los atributos que especifica el enunciado



GENERAR UN SERVICIO REST-POST

crear un servicio rest-post para agregar un nuevo registro, las condiciones son: si es mayor a 20 se seteara ADULTO en el campo categoria, si es menor sera ADOLESCENTE y si la edad es menor a 10 sera NIÑO

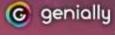
y para el campo FULLNAME se debera concatenar el nombre y el apellido del usuario

+ INFO

LISTAR TODOS LOS PACIENTES Y UNO SOLO REST-GET

usando el verbo GET debera leer todos los registros ingresados a la base de datos y asi tambien de ser capaz de leer un regsitro especifico

+ INFO



PROBLEMAS

DEFENSA HITO3

CREAR SERVICIO REST PUT PERMITA MODIFICAR

crear un servico utilizando el verbo PUT para modificar un registro especifico

EVITAR LA INSERCION DE PACIENTES MAYORES A70 MAYOS

al momento de agregar un nuevo registro de tomar en cuenta que no puede ingresar regist ros de personas mayores a 70 años

+ INFO

CREAR UN REST- DELETE QUE ELIMINE TODOS LOS **REGISTROS**

debe elimiar todos los registro que contenga la base de datos sin excepciones

+ INFO



creara la tabl a de nombre"corona virus paciente" con los atributos que especifica el enunciado

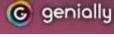
CODIGO DDL GENERADO EN LA BASE DE DATOS

-- auto-generated definition create table corona_virus

id_corona_virus integer not null constraint corona_virus_pkey primary key, casos_contagiados integer,

casos_recuperados integer, casos_sospechosos integer, nombre_dep varchar(50) not null

alter table corona_virus owner to aecovanrdqbkfr;



crear un servicio rest-post para agregar un nuevo registro, las condiciones son: si es mayor a 20 se seteara ADULTO en el campo categoria, si es menor sera ADOLESCENTE y si la edad es menor a 10 sera NIÑO

y para el campo FULLNAME se debera concatenar el nombre y el apellido del usuario

CODIGO PARA AGREGAR UN NUEVO REGISTRO DENTRO EL "CORONA SERVICE"

```
@Override
 public CoronaVirusModel save(CoronaVirusModel pModel) {
  if(pModel.getEdad() > 20)[
    pModel.setCategoria("ADULTO");
    String nombrecompleto = (pModel.getNombrepaciente() + " "+pModel.getApellidos()):
    pModel.setFullname(nombrecompleto);
  else if(pModel.getEdad() < 20 && pModel.getEdad() > 10 ){
    pModel.setCategoria("Adolecente");
    String nombrecompleto = (pModel.getNombrepaciente() + " "+pModel.getApellidos >
    pModel.setFullname(nombrecompleto);
   if(pModel.getEdad() < 10)[
    pModel.setCategoria("ninhio");
    String nombrecompleto = (pModel.getNombrepaciente() + " "+pModel.getApellidos());
    pModel_setFullname(nombrecompleto);
  else [
    pModel.setCategoria("ERROR");
  return coronaRepo.save(pModel);
```

CODIGO PARA AGREGAR UN NUEVO REGISTRO DENTRO DEL "USER CONTROLLER REST"

```
@PostMapping("/coronaVirusPaciente")
public ResponseEntity save(@RequestBody CoronaVirusModel persona) {
    try {
        if(persona.getEdad() > 70)
            persona= null;
        return new ResponseEntity<>(coronaservice.save(persona), HttpStatus.EXPECTATION_FAILED);
    } catch (Exception e) {
        return new ResponseEntity<>(null, HttpStatus.EXPECTATION_FAILED);
    }
}
```

POST /coronaVirusPaciente Adds the new case CV in the store



usando el verbo GET debera leer todos los registros ingresados a la base de datos y asi tambien de ser capaz de leer un regsitro especifico

CODIGO PARA LEER TODOS LOS REGISTROS DENTRO EL "USER CONTROLLER REST"

```
@GetMapping("/coronaVirusPaciente")
 public ResponseEntity<List<CoronaVirusModel>>
getAllPersons() {
   try[
     List<CoronaVirusModel> registroCoronaVirus =
coronaservice.getAllPersons();
     if (registroCoronaVirus.isEmpty()) {
      return new ResponseEntity<>(HttpStatus.NO_CONTENT);
     ] else [
      return new ResponseEntity<>(registroCoronaVirus,
HttpStatus.OK);
   } catch (Exception e) {
     return new ResponseEntity<>(null.
HttpStatus.INTERNAL_SERVER_ERROR);
```

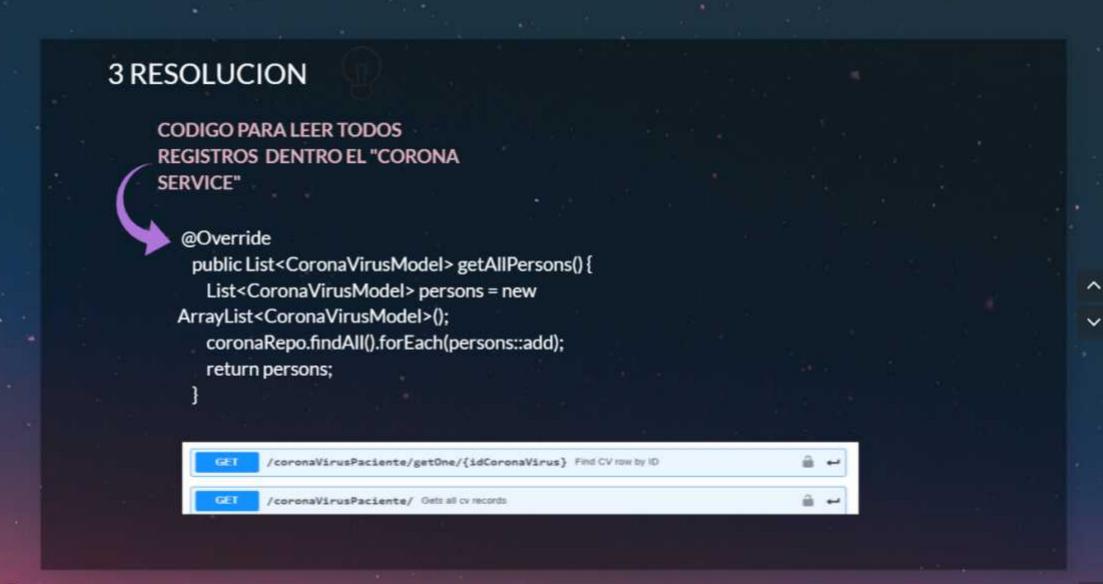


CODIGO PARA LEER UN REGISTRO ESPECIFICO DENTRO EL "CORONA SERVICE"

```
@Override
 public CoronaVirusModel getPersonByldPer(Integer idPer) {
   Optional < Corona Virus Model > person = corona Repo. find Byld (id Per);
   CoronaVirusModel pModel = null;
   if (person.isPresent()) {
    pModel = person.get();
  return pModel;
```

CODIGO PARA LEER UN REGISTRO ESPECIFICO DENTRO EL "USER C ONTROLLER REST"

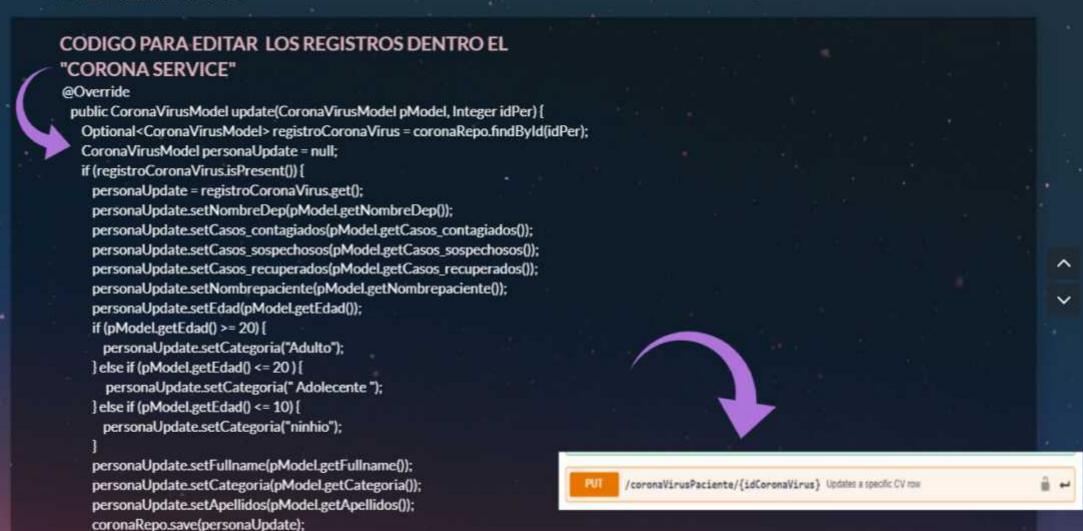
```
@GetMapping("/coronaVirusPaciente/getOne/{idCorona}")
 public ResponseEntity<CoronaVirusModel>
getPersonByIdPer(@PathVariable("idCorona") Integer idCorona) {
   try{
    CoronaVirusModel pModel =
coronaservice.getPersonByIdPer(idCorona);
    if (pModel != null) {
      return new ResponseEntity<>(pModel, HttpStatus.OK);
    ] else [
      return new ResponseEntity<>(HttpStatus.NOT_FOUND):
   } catch (Exception e) {
    return new ResponseEntity<>(null,
HttpStatus.INTERNAL_SERVER_ERROR);
```

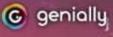


crear un servico utilizando el verbo PUT para modificar un registro especifico



```
@PutMapping("/coronaVirusPaciente/{idCorona}")
 public ResponseEntity<CoronaVirusModel>
updateMaterias(@PathVariable("idCorona") Integer idPer,
@RequestBody CoronaVirusModel pModel) [
   try{
     CoronaVirusModel pUpdate = coronaservice.update(pModel,
idPer);///
     if (pUpdate != null) {
      return new ResponseEntity<>(pUpdate, HttpStatus.OK);
     ] else [
      return new ResponseEntity<>(HttpStatus.NOT_FOUND);
   } catch (Exception e) {
     return new ResponseEntity<>(null.
HttpStatus.INTERNAL_SERVER_ERROR);
```





return personaUpdate;

al momento de agregar un nuevo registro de tomar en cuenta que no puede ingresar regist ros de personas mayores a 70 años

@PostMapping("/coronaVirusPaciente")
public ResponseEntity
nosavemayores(@RequestBody
CoronaVirusModel persona) {
 try {
 return new ResponseEntity<>
(coronaservice.save(persona),
HttpStatus.EXPECTATION_FAILED);
 } catch (Exception e) {
 return new ResponseEntity<>(null,
HttpStatus.EXPECTATION_FAILED);
 }
}

CODIGO QUE EVITA EL REGISTRO DE PERSONAS MAYORES A 70 AÑOS

```
//NO CREAR REGISTRO SI ES MAYOR A 70

@PostMapping("/coronaVirusPaciente")

public ResponseEntity save(@RequestBody CoronaVirusModel persona)

try {

if(persona.getEdad() > 70)

persona= null;

return new ResponseEntity<>(coronaservice.save(persona),

HttpStatus.EXPECTATION_FAILED);
} catch (Exception e) {

return new ResponseEntity<>(null, HttpStatus.EXPECTATION_FAILED)
}
```

debe elimiar todos los registro que contenga la base de datos sin excepciones

METODO QUE NO RECIBE NINGUN PARAMETRO DE ESTA MANERA ELIMINA TODO

```
@Override
public Integer deleteTodo() {
  return null;
}
```

CODIGO QUE ELIMINA TODOS LOS REGISTROS

```
@DeleteMapping("/CoronaVirusPaciente/deleteTodo")
public ResponseEntity<String> deleteTodo() {
    try {
        coronaservice.deleteTodo();
        return new ResponseEntity<>("info successfully deleted",
HttpStatus.OK);
    } catch (Exception e) {
        return new ResponseEntity<>(null,
HttpStatus.EXPECTATION_FAILED);
    }
}
```

