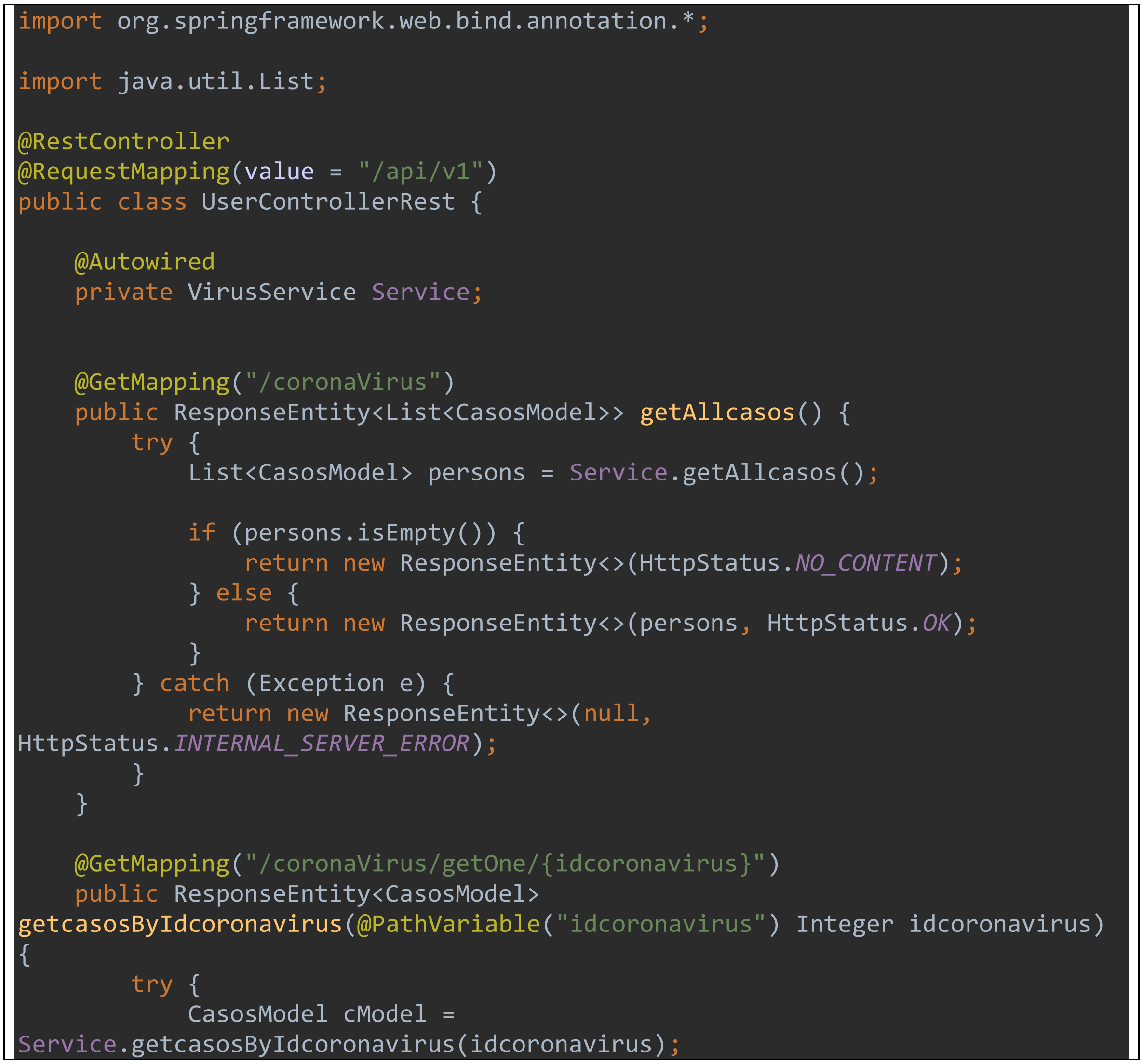
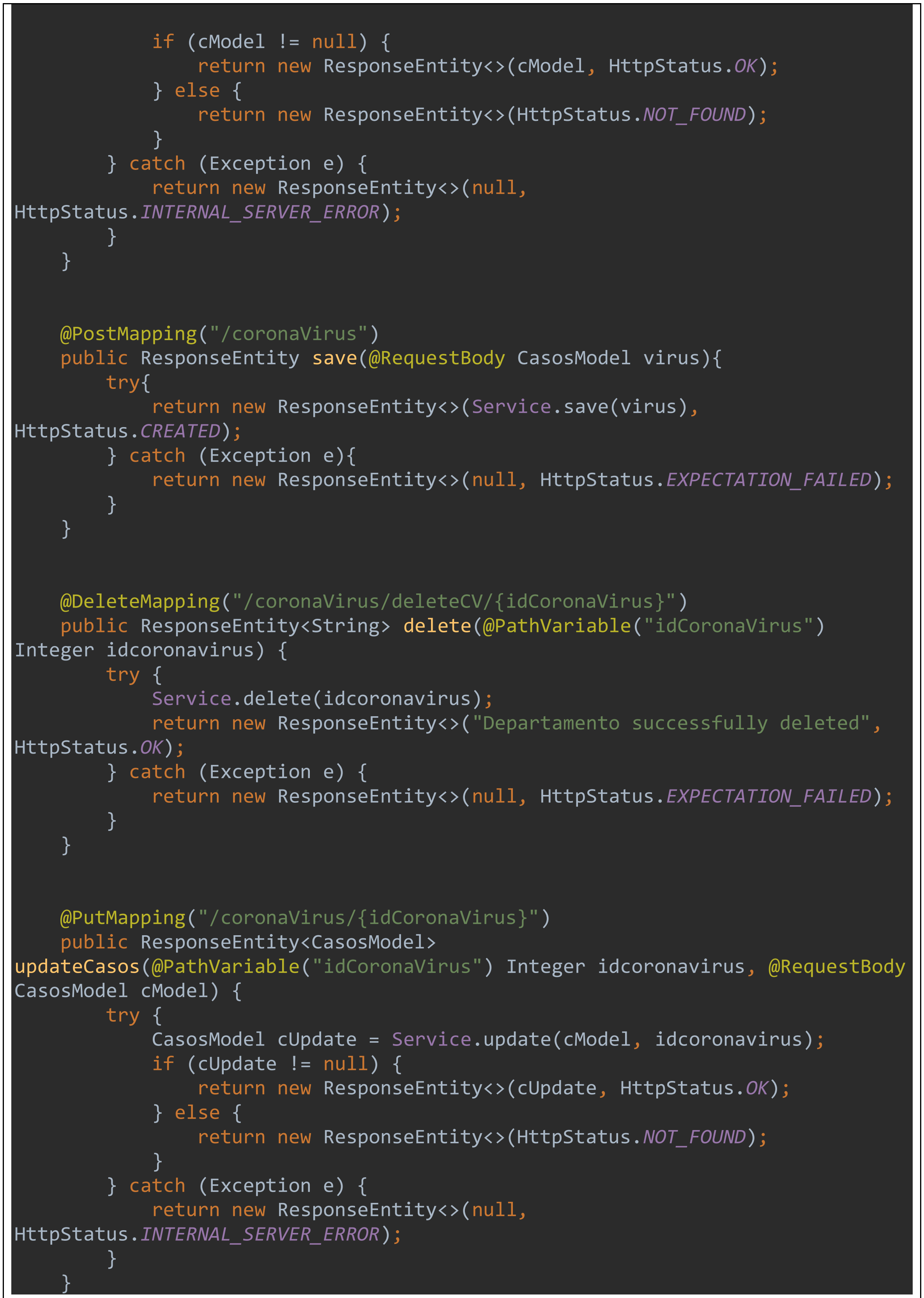
UNIVERSIDAD PRIVADA FRANZ TAMAYO

DEFENSA HITO 3 - TAREA FINAL

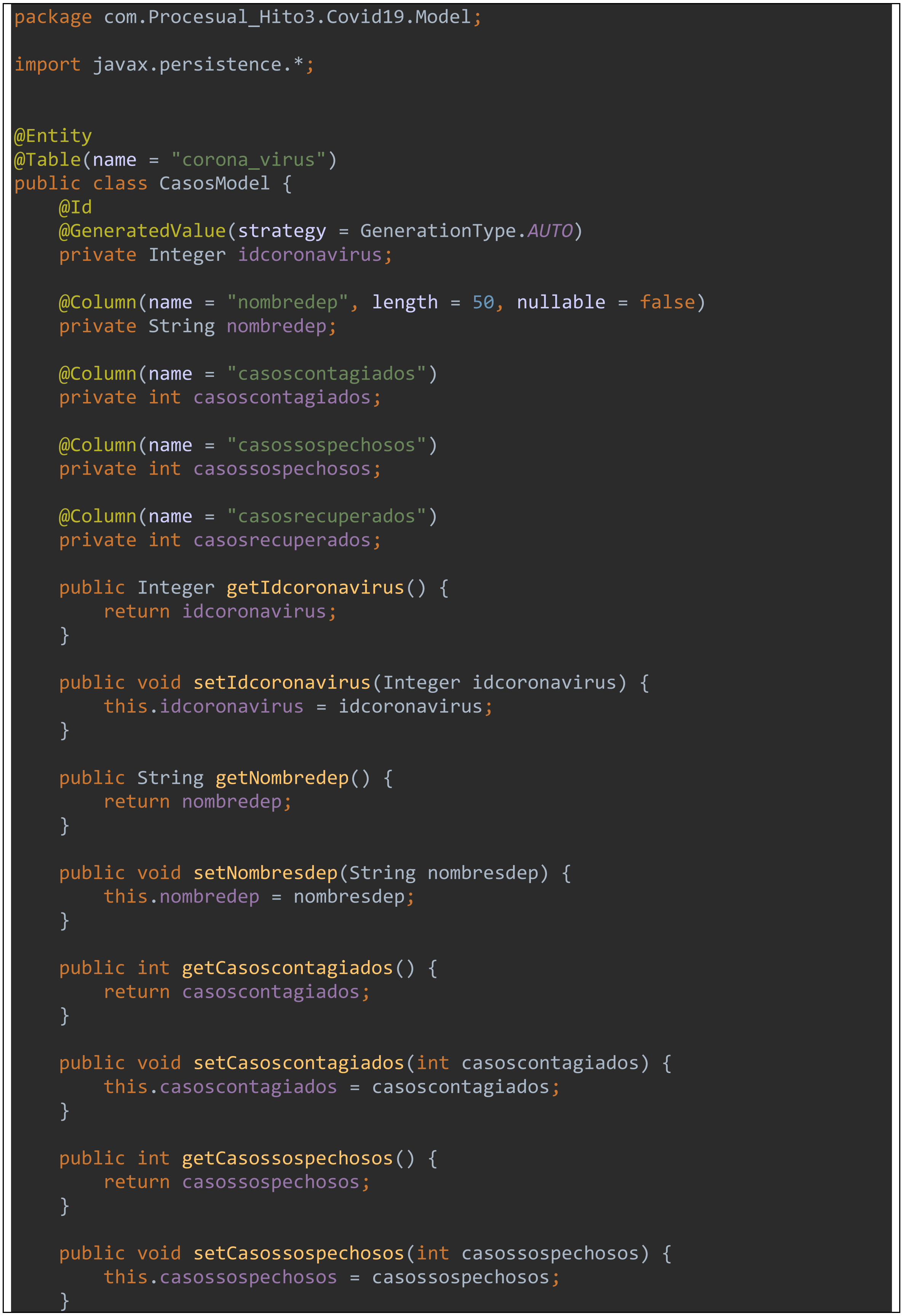
Nombre Completo: **Sergio Diego Cordova Davalos**
Asignatura: **PROGRAMACIÓN III**
Carrera: **INGENIERÍA DE SISTEMAS**
Paralelo: **PROG (1)**
Docente: **Lic. William R. Barra Paredes**
fecha: **11/05/2020**



```
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping(value = "/api/v1")
public class UserControllerRest {

    @Autowired
    private VirusService Service;


    @GetMapping("/coronaVirus")
    public ResponseEntity<List<CasosModel>> getAllcasos() {
        try {
            List<CasosModel> persons = Service.getAllcasos();

            if (persons.isEmpty()) {
                return new ResponseEntity<>(HttpStatus.NO_CONTENT);
            } else {
                return new ResponseEntity<>(persons, HttpStatus.OK);
            }
        } catch (Exception e) {
            return new ResponseEntity<>(null,
HttpStatus.INTERNAL_SERVER_ERROR);
        }
    }

    @GetMapping("/coronaVirus/getOne/{idcoronavirus}")
    public ResponseEntity<CasosModel>
getcasosByIdcoronavirus(@PathVariable("idcoronavirus") Integer idcoronavirus)
{
        try {
            CasosModel cModel =
Service.getcasosByIdcoronavirus(idcoronavirus);
```



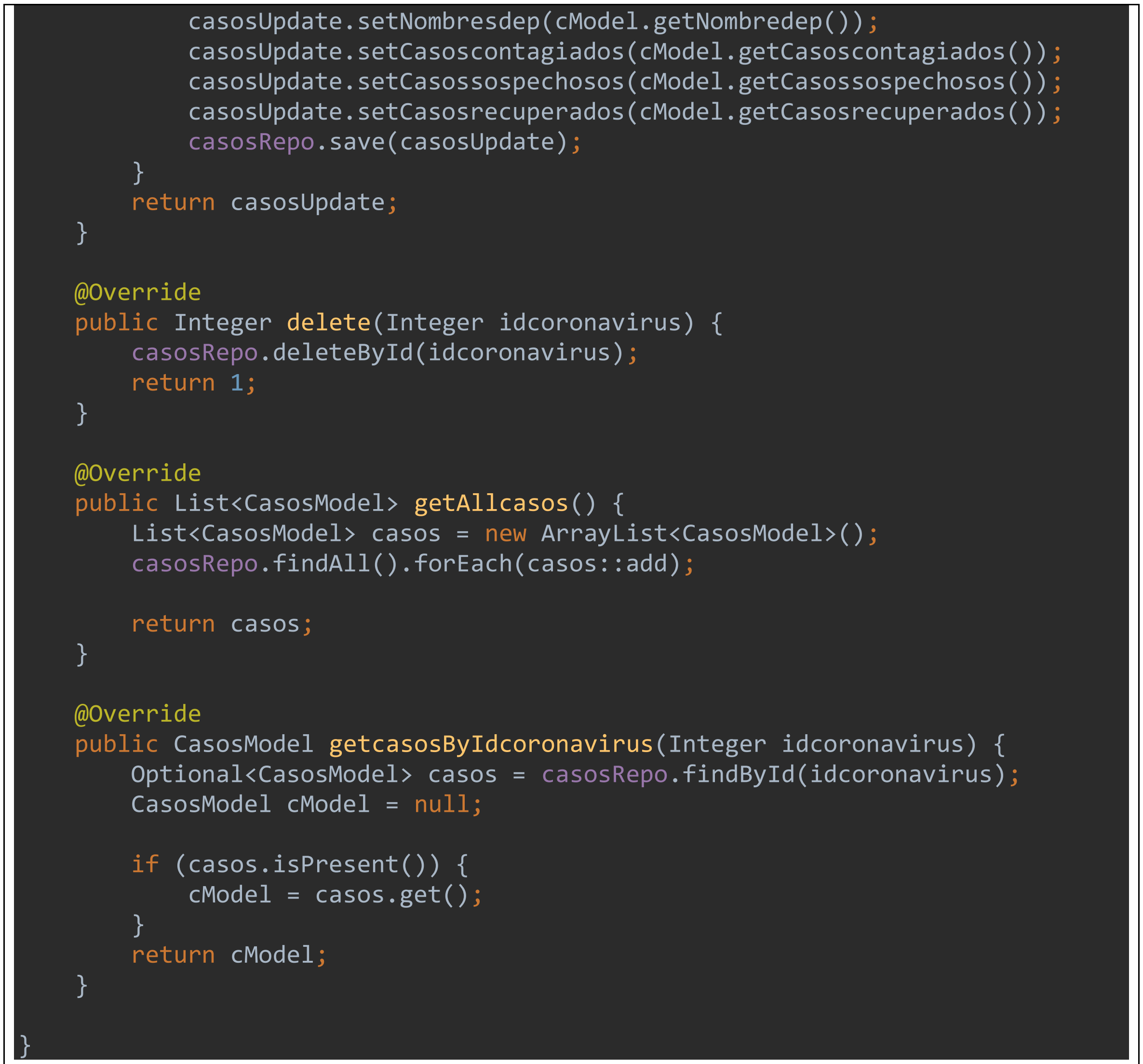
```
            if (cModel != null) {
                return new ResponseEntity<>(cModel, HttpStatus.OK);
            } else {
                return new ResponseEntity<>(HttpStatus.NOT_FOUND);
            }
        } catch (Exception e) {
            return new ResponseEntity<>(null,
HttpStatus.INTERNAL_SERVER_ERROR);
        }
    }


    @PostMapping("/coronaVirus")
    public ResponseEntity save(@RequestBody CasosModel virus){
        try{
            return new ResponseEntity<>(Service.save(virus),
HttpStatus.CREATED);
        } catch (Exception e){
            return new ResponseEntity<>(null, HttpStatus.EXPECTATION_FAILED);
        }
    }


    @DeleteMapping("/coronaVirus/deleteCV/{idCoronaVirus}")
    public ResponseEntity<String> delete(@PathVariable("idCoronaVirus")
Integer idcoronavirus) {
        try {
            Service.delete(idcoronavirus);
            return new ResponseEntity<>("Departamento successfully deleted",
HttpStatus.OK);
        } catch (Exception e) {
            return new ResponseEntity<>(null, HttpStatus.EXPECTATION_FAILED);
        }
    }


    @PutMapping("/coronaVirus/{idCoronaVirus}")
    public ResponseEntity<CasosModel>
updateCasos(@PathVariable("idCoronaVirus") Integer idcoronavirus, @RequestBody
CasosModel cModel) {
        try {
            CasosModel cUpdate = Service.update(cModel, idcoronavirus);
            if (cUpdate != null) {
                return new ResponseEntity<>(cUpdate, HttpStatus.OK);
            } else {
                return new ResponseEntity<>(HttpStatus.NOT_FOUND);
            }
        } catch (Exception e) {
            return new ResponseEntity<>(null,
HttpStatus.INTERNAL_SERVER_ERROR);
        }
    }
```



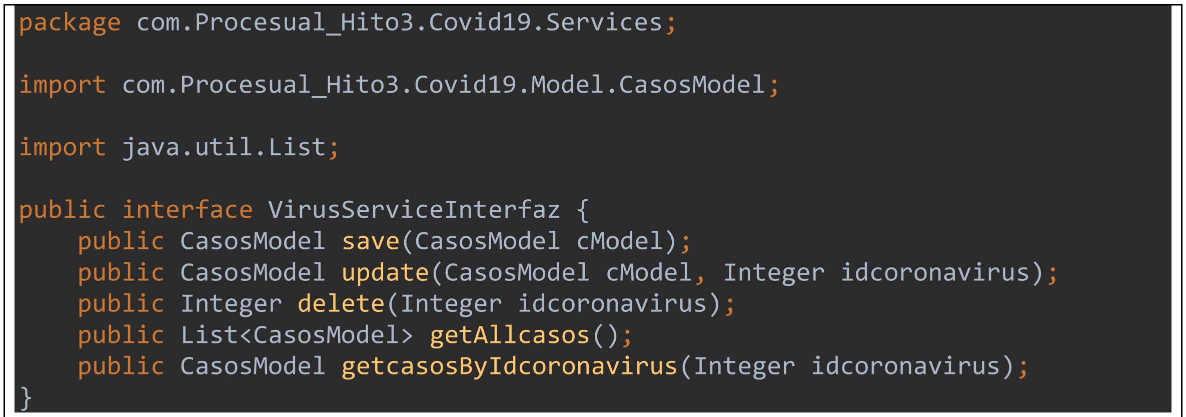
```java
package com.Procesual_Hito3.Covid19.Model;

import javax.persistence.*;


@Entity
@Table(name = "corona_virus")
public class CasosModel {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer idcoronavirus;

    @Column(name = "nombredep", length = 50, nullable = false)
    private String nombredep;

    @Column(name = "casoscontagiados")
    private int casoscontagiados;

    @Column(name = "casossospechosos")
    private int casossospechosos;

    @Column(name = "casosrecuperados")
    private int casosrecuperados;

    public Integer getIdcoronavirus() {
        return idcoronavirus;
    }

    public void setIdcoronavirus(Integer idcoronavirus) {
        this.idcoronavirus = idcoronavirus;
    }

    public String getNombredep() {
        return nombredep;
    }

    public void setNombresdep(String nombresdep) {
        this.nombredep = nombresdep;
    }

    public int getCasoscontagiados() {
        return casoscontagiados;
    }

    public void setCasoscontagiados(int casoscontagiados) {
        this.casoscontagiados = casoscontagiados;
    }

    public int getCasossospechosos() {
        return casossospechosos;
    }

    public void setCasossospechosos(int casossospechosos) {
        this.casossospechosos = casossospechosos;
    }
    public int getCasosrecuperados() {
        return casosrecuperados;
    }

    public void setCasosrecuperados(int casosrecuperados) {
        this.casosrecuperados = casosrecuperados;
    }
}
```

```
package com.Procesual_Hito3.Covid19.Repo;

import com.Procesual_Hito3.Covid19.Model.CasosModel;
import org.springframework.data.jpa.repository.JpaRepository;

public interface CasosRepo  extends JpaRepository<CasosModel, Integer> {

}
```

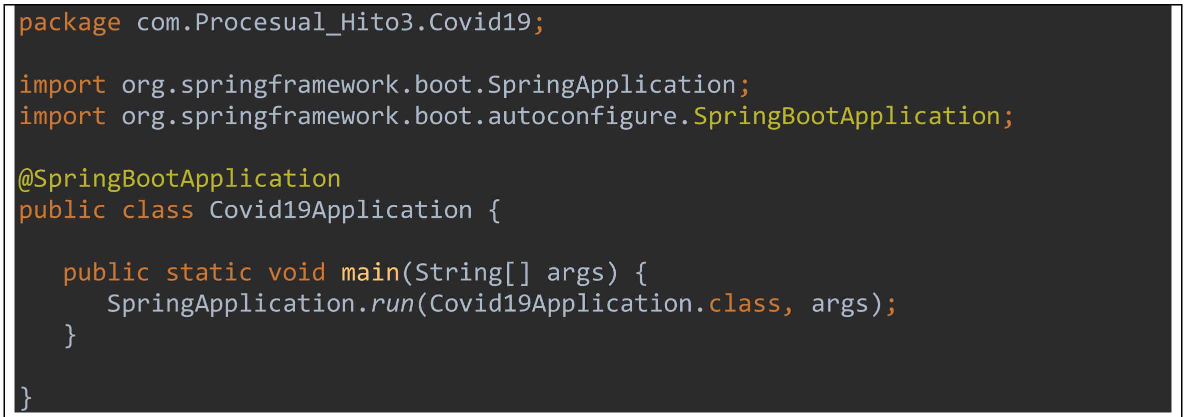
```
package com.Procesual_Hito3.Covid19.Services;

import com.Procesual_Hito3.Covid19.Model.CasosModel;
import com.Procesual_Hito3.Covid19.Repo.CasosRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.ArrayList;
import java.util.List;
import java.util.Optional;

 @Service
  public class VirusService implements VirusServiceInterfaz{
    @Autowired
    private CasosRepo casosRepo;

    @Override
    public CasosModel save(CasosModel cModel) {
        return casosRepo.save(cModel);
    }

    @Override
    public CasosModel update(CasosModel cModel, Integer idcoronavirus) {
        Optional<CasosModel> casos = casosRepo.findById(idcoronavirus);
        CasosModel casosUpdate = null;

        if (casos.isPresent()) {
            casosUpdate = casos.get();
```



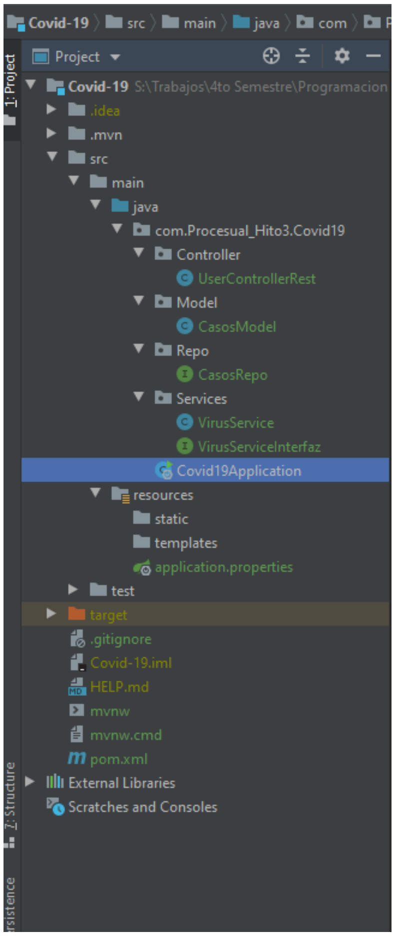
```java
            casosUpdate.setNombresdep(cModel.getNombredep());
            casosUpdate.setCasoscontagiados(cModel.getCasoscontagiados());
            casosUpdate.setCasossospechosos(cModel.getCasossospechosos());
            casosUpdate.setCasosrecuperados(cModel.getCasosrecuperados());
            casosRepo.save(casosUpdate);
        }
        return casosUpdate;
    }

    @Override
    public Integer delete(Integer idcoronavirus) {
        casosRepo.deleteById(idcoronavirus);
        return 1;
    }

    @Override
    public List<CasosModel> getAllcasos() {
        List<CasosModel> casos = new ArrayList<CasosModel>();
        casosRepo.findAll().forEach(casos::add);

        return casos;
    }

    @Override
    public CasosModel getcasosByIdcoronavirus(Integer idcoronavirus) {
        Optional<CasosModel> casos = casosRepo.findById(idcoronavirus);
        CasosModel cModel = null;

        if (casos.isPresent()) {
            cModel = casos.get();
        }
        return cModel;
    }

}
```



```java
package com.Procesual_Hito3.Covid19.Services;

import com.Procesual_Hito3.Covid19.Model.CasosModel;

import java.util.List;

public interface VirusServiceInterfaz {
    public CasosModel save(CasosModel cModel);
    public CasosModel update(CasosModel cModel, Integer idcoronavirus);
    public Integer delete(Integer idcoronavirus);
    public List<CasosModel> getAllcasos();
    public CasosModel getcasosByIdcoronavirus(Integer idcoronavirus);
}
```



```
package com.Procesual_Hito3.Covid19;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Covid19Application {

   public static void main(String[] args) {
      SpringApplication.run(Covid19Application.class, args);
   }

}
```



```
▼ Covid-19  S:\Trabajos\4to Semestre\Programacion
  ▶ .idea
  ▶ .mvn
  ▼ src
    ▼ main
      ▼ java
        ▼ com.Procesual_Hito3.Covid19
          ▼ Controller
              UserControllerRest
          ▼ Model
              CasosModel
          ▼ Repo
              CasosRepo
          ▼ Services
              VirusService
              VirusServiceInterfaz
            Covid19Application
      ▼ resources
          static
          templates
          application.properties
    ▶ test
  ▶ target
    .gitignore
    Covid-19.iml
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
▶ External Libraries
  Scratches and Consoles
```