

Control de flujo

Main.java

```
27     int opcion = 0;
28     System.out.println("Bienvenido/a a Traza COVID\nSelecciona una de las siguientes opciones: ");
29     while(opcion != 8) {
30         menuPrincipal();
31
32         opcion = sc.nextInt();
33         if (opcion == 1) {
34             svc.agregarPersona(pacientes);
35         }
36         else if (opcion == 2) {
37             svc.listPersona(pacientes);
38         }
39         else if (opcion == 3) {
40             System.out.print("Ingrese id: ");
41             svc.getOne(pacientes, sc.nextInt());
42         }
43         else if (opcion == 4) {
44             System.out.print("Ingrese id: ");
45             svc.eliminarPersona(pacientes, sc.nextInt());
46         }
47         else if (opcion == 5) {
48             System.out.print("Ingrese id: ");
49             svc.modificarPersona(pacientes, sc.nextInt());
50         }
51         else if (opcion == 6) {
52             System.out.println("Implementar (tomar todos los datos)");
53         }
54         else if (opcion == 7) {
55             System.out.print("Ingrese id: ");
56             svc.getQRData(pacientes, sc.nextInt());
57         }
58     }
```

Clases

```
public class PersonaServiceImp implements PersonaService {

    public void listPersona(List<Paciente> pacientes) {
        // comentar en cuanto no sea explicable

        int size = pacientes.size();
        if (size == 0) {
            System.out.println("----Aún no existen registros----");
        } else {
            for (int i = 0; i < size; i++) {
                System.out.println(pacientes.get(i)); // Puede quedar mejor estilizado.
            }
        }
    }

    public Paciente agregarPersona(List<Paciente> pacientes) {
        @SuppressWarnings("resource")
        Scanner scan = new Scanner(System.in);
        @SuppressWarnings("resource")
        Scanner numberScan = new Scanner(System.in):
    }
```

```

public class Persona {

    private int id;
    private String rut;
    private String nombre;
    private String apellido;
    private int edad;
    public Persona(int id, String rut, String nombre, String apellido, int edad) {
        super();
        this.id = id;
        this.rut = rut;
        this.nombre = nombre;
        this.apellido = apellido;
        this.edad = edad;
    }
    public Persona() {
        super();
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {

package model;

import java.util.List;

public class Paciente extends Persona {

    private String estado;
    private String region;
    private String comuna;
    private List<Contacto> contactos;
    private Diagnostico diagnostico;

    public Paciente(int id, String rut, String nombre, String apellido, int edad, String estado, String region, String comuna, List<Contacto> contactos, Diagnostico diagnostico) {
        super(id, rut, nombre, apellido, edad);
        this.estado = estado;
        this.region = region;
        this.comuna = comuna;
        this.contactos = contactos;
        this.diagnostico = diagnostico;
    }

    public Paciente(int id, String rut, String nombre, String apellido, int edad, String estado, String region, String comuna, List<Contacto> contactos, Diagnostico diagnostico) {
        super(id, rut, nombre, apellido, edad);
        this.estado = estado;
        this.region = region;
        this.comuna = comuna;
        this.contactos = contactos;
        this.diagnostico = diagnostico;
    }

public class Diagnostico {

    private boolean previousPCR;
    private int duracionSintomas;
    private boolean enfermedadStatus;
    private String sospecha;
    private String medidas;

    public Diagnostico(boolean previousPCR, int duracionSintomas, boolean enfermedadStatus, String sospecha, String medidas) {
        super();
        this.previousPCR = previousPCR;
        this.duracionSintomas = duracionSintomas;
        this.enfermedadStatus = enfermedadStatus;
        this.sospecha = sospecha;
        this.medidas = medidas;
    }
    public Diagnostico() {
        super();
    }
    public boolean isPreviousPCR() {
        return previousPCR;
    }

```

```

public class Contacto extends Persona {

    private String correo;
    private String tipoContacto; //familia, trabajo, otros
    private String nexo; //hermano, padre, .colega, etc
    public Contacto(int id, String rut, String nombre, String apellido, int edad, String correo, String tipoContacto, String nexo) {
        super(id, rut, nombre, apellido, edad);
        this.correo = correo;
        this.tipoContacto = tipoContacto;
        this.nexo = nexo;
    }

    public Contacto(int id, String rut, String nombre, String apellido, int edad) {
        super(id, rut, nombre, apellido, edad);
    }

    public Contacto() {
        super();
    }

}

public class Main {

    public static void main(String[] args) throws IOException {
        @SuppressWarnings("resource")
        Scanner sc = new Scanner(System.in);
        PersonaServiceImp svc = new PersonaServiceImp();
        List<Paciente> pacientes = new ArrayList<Paciente>();
        // Inicialización de Scanner, clase servicio y listado de personas.
        // System.out.println(personas.size());
        // System.out.println("Siguiente" + svc.siguienteId(personas));
        Paciente paciente = new Paciente();
        pacientes.add(paciente);
    }
}

```

Constructor

```

public class Contacto extends Persona {

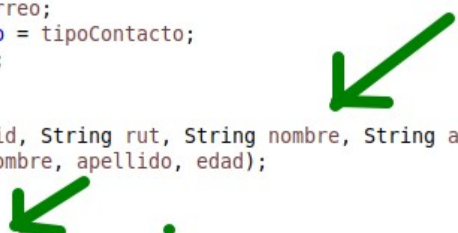
    private String correo;
    private String tipoContacto; //familia, trabajo, otros
    private String nexo; //hermano, padre, .colega, etc
    public Contacto(int id, String rut, String nombre, String apellido, int edad, String correo, String tipoContacto, String nexo) {
        super(id, rut, nombre, apellido, edad);
        this.correo = correo;
        this.tipoContacto = tipoContacto;
        this.nexo = nexo;
    }

    public Contacto(int id, String rut, String nombre, String apellido, int edad) {
        super(id, rut, nombre, apellido, edad);
    }

    public Contacto() {
        super();
    }

}

```



Herencia

```
public Contacto(int id, String rut, String nombre, String apellido, int edad) {  
    super(id, rut, nombre, apellido, edad);  
}  
  
public Contacto() {  
    super();  
}  
  
public String getCorreo() {  
    return correo;  
}  
public void setCorreo(String correo) {  
    this.correo = correo;  
}  
public String getTipoContacto() {  
    return tipoContacto;  
}  
public void setTipoContacto(String tipoContacto) {  
    this.tipoContacto = tipoContacto;  
}  
public String getNombre() {
```
