Programación OO y Estructurada

Integrantes:

- Rea, Gabriel Alejandro
- Díaz, Lucas
- Sosa, Matías
- Bruno Vera, Leonel
- Coronel, Maximiliano

Proyecto: SITE

Versión Orientada a Objetos.

El siguiente ejemplo muestra un bloque de código orientado a objetos utilizando Java, que es el lenguaje utilizado en nuestro proyecto. Este bloque de código es un servicio de guardado de una denuncia, la cual carga y vincula personas y a su vez cargando archivos.

```
    @Override
    public void guardarDenuncia(DenunciaDTO denunciaDTO,
List<MultipartFile> files) {



        if (denunciaDTO == null) throw new IllegalArgumentException("La
denuncia no puede ser nula.");
        if (files == null) throw new IllegalArgumentException("Los
archivos no pueden ser nulos.");
        if (denunciaDTO.getPersonas() == null) throw new
IllegalArgumentException("Las personas no deben ser nulas.");



        try {
            // Aquí se crea la denuncia
            Denuncia denuncia = new Denuncia();
            denuncia.setDescripcion(denunciaDTO.getDescripcion());
            denuncia.setObjeto(denunciaDTO.getObjeto());
            denuncia.setMotivo(denunciaDTO.getMotivo());
            denunciaRepository.save(denuncia);



            // Aquí se crean (si es que no existen) y vinculan las
personas a la denuncia
```

```
            List<Persona> personasPersistidas =
personaService.guardarPersonas(denunciaDTO.getPersonas());

            // 3. Mapear roles a personas persistidas

            List<PersonaRolDTO> personasRolPersistidas = new
ArrayList<>();

            for (int i = 0; i < denunciaDTO.getPersonas().size(); i++) {

                PersonaRolDTO original = denunciaDTO.getPersonas().get(i);

                Persona personaPersistida = personasPersistidas.get(i);

                PersonaRolDTO dto = new PersonaRolDTO();

                dto.setPersona(personaPersistida);

                dto.setRol(original.getRol());

                dto.setNombreDelegado(original.getNombreDelegado());

                dto.setApellidoDelegado(original.getApellidoDelegado());

                dto.setDniDelegado(original.getDniDelegado());

                personasRolPersistidas.add(dto);

            }


dPersonaService.vincularPersonaDenuncia(personasRolPersistidas, denuncia);



            // Aquí se guardan los documentos, sigue igual

            documentoService.guardarDocumentos(files, denuncia);


        } catch (Exception e) {

            e.getMessage();
```

```
            throw new RuntimeException("Ocurrió un error al guardar una
nueva denuncia");

        }

    }
```

Versión Estructurada.

Este otro ejemplo muestra un bloque de código estructurado utilizando el código anteriormente utilizado. Este bloque de código es un servicio de guardado de una denuncia, la cual carga y vincula personas y a su vez cargando archivos.

```
public class Main {

    public static void main(String[] args) {

        String descripcion = "Descripción";

        String objeto = "Objeto";

        String motivo = "Motivo";



        Denuncia denuncia = crearDenuncia(descripcion, objeto, motivo);

        guardarEnBD(denuncia);



        List<Persona> personas = cargarPersonas();

        for (Persona p : personas) {

            guardarPersona(p);
```

```
            vincularPersonaADenuncia(p, denuncia);
        }
        List<Archivo> archivos = cargarArchivos();
        for (Archivo a : archivos) {
            guardarArchivo(a, denuncia);
        }
        System.out.println("Denuncia registrada.");
    }

    static Denuncia crearDenuncia(String d, String o, String m) {
        return new Denuncia(d, o, m);
    }

    static void guardarEnBD(Denuncia d) {}
    static void guardarPersona(Persona p) {}
    static void vincularPersonaADenuncia(Persona p, Denuncia d) {}
    static void guardarArchivo(Archivo a, Denuncia d) {}
    static List<Persona> cargarPersonas() { return new ArrayList<>(); }
    static List<Archivo> cargarArchivos() { return new ArrayList<>(); }
}
```