

Trabajo Práctico 5: Relaciones UML 1 a 1

Nombre: Alfredo de Inocenti

Materia: Programación II

Carrera: Tecnicatura Universitaria en Programación

Repositorio GitHub: https://github.com/fonsecaFuentes/tps-Programacion_II/tree/main/Trabajo_Practico_5

OBJETIVO GENERAL

Modelar clases con relaciones 1 a 1 utilizando diagramas UML. Identificar correctamente el tipo de relación (asociación, agregación, composición, dependencia) y su dirección, y llevarlas a implementación en Java.

Ejercicios resueltos

❖ Ejercicio 1 – Pasaporte · Foto · Titular

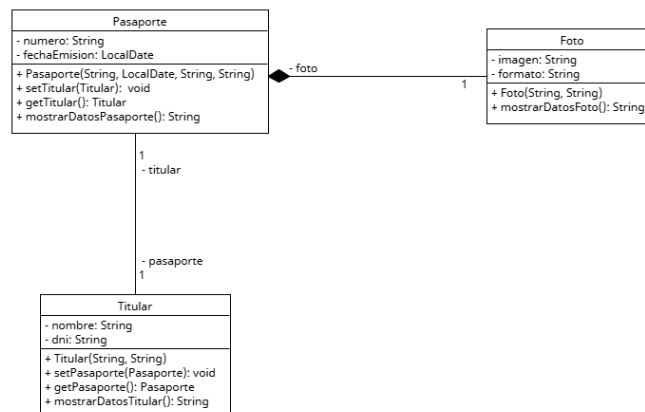
- **Consigna:**

Modelar **composición** Pasaporte → Foto y **asociación bidireccional** Pasaporte ↔ Titular.

Se implementaron setters para mantener la relación 1 ↔ 1.

- **Archivos principales:**

- Pasaporte.java · Foto.java · Titular.java
- Principal.java



❖ Ejercicio 2 – Celular · Batería · Usuario

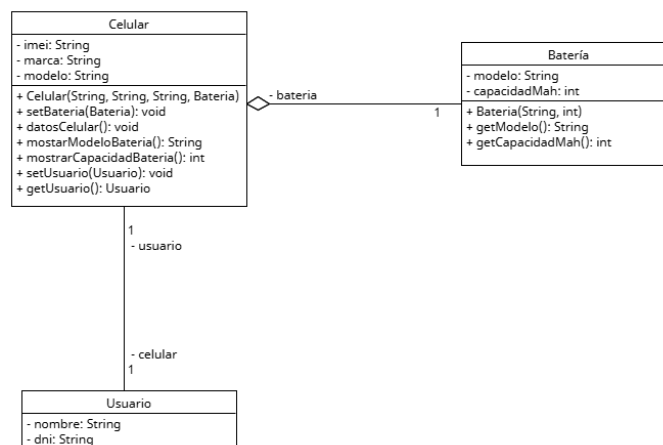
- **Consigna:**

Se modelo **agregación** Celular → Batería y **asociación bidireccional** Celular ↔ Usuario.

Mostrar datos del celular y su batería.

- **Archivos principales:**

- Celular.java · Bateria.java · Usuario.java
- Principal.java



❖ Ejercicio 3 – Libro · Autor · Editorial

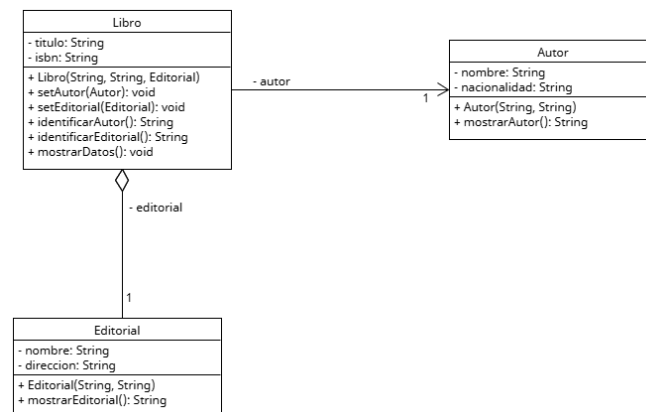
- **Consigna:**

Asociación unidireccional Libro → Autor y **agregación** con Libro → Editorial.

Imprimir ficha técnica del libro.

- **Archivos principales:**

- Libro.java · Autor.java · Editorial.java
- Principal.java



❖ Ejercicio 4 – TarjetaDeCredito · Cliente · Banco

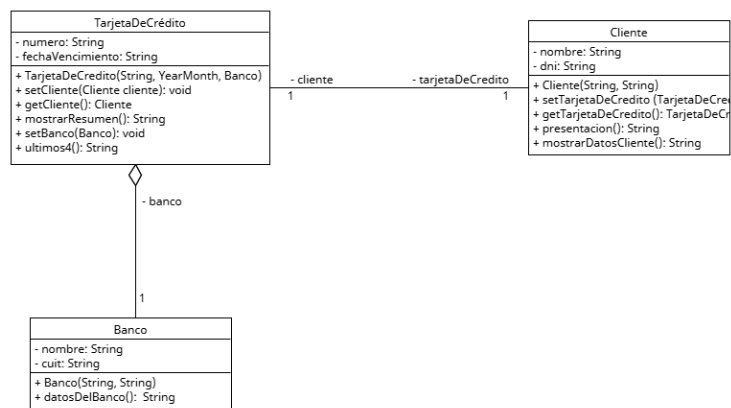
- **Consigna:**

Asociación bidireccional TarjetaDeCredito ↔ Cliente.

Relación **agregación** TarjetaDeCredito → Banco.

- **Archivos principales:**

- TarjetaDeCredito.java · Cliente.java · Banco.java
- Principal.java



❖ Ejercicio 5 – Computadora · PlacaMadre · Propietario

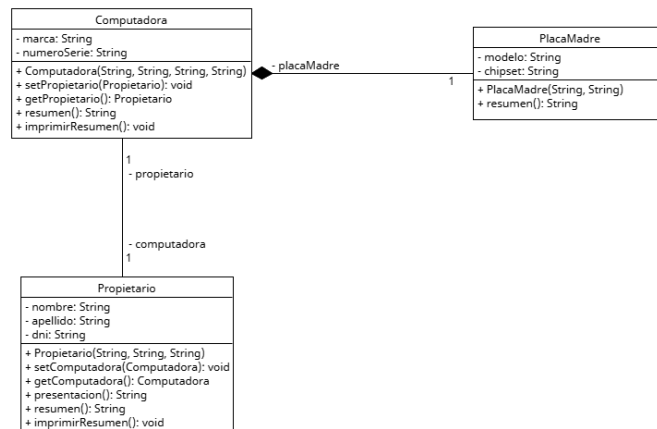
- **Consigna:**

Composición Computadora → PlacaMadre.

Asociación bidireccional Computadora ↔ Propietario.

- **Archivos principales:**

- Computadora.java · PlacaMadre.java · Propietario.java
- Principal.java



❖ Ejercicio 6 – Reserva · Cliente · Mesa

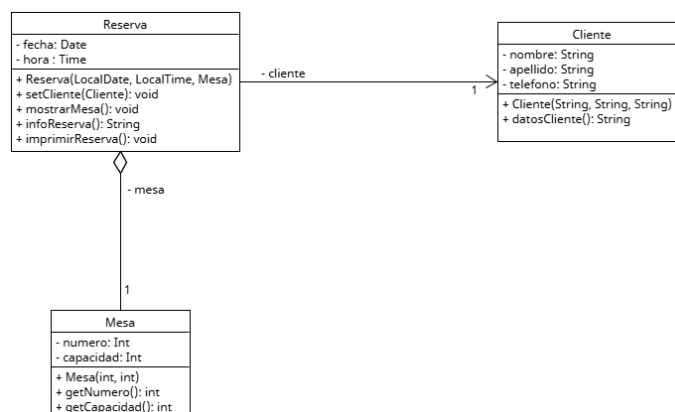
- **Consigna:**

Asociación unidireccional Reserva → Cliente y relación de **agregación** con Reserva → Mesa.

Mostrar resumen de reserva.

- **Archivos principales:**

- Reserva.java · Cliente.java · Mesa.java
- Principal.java



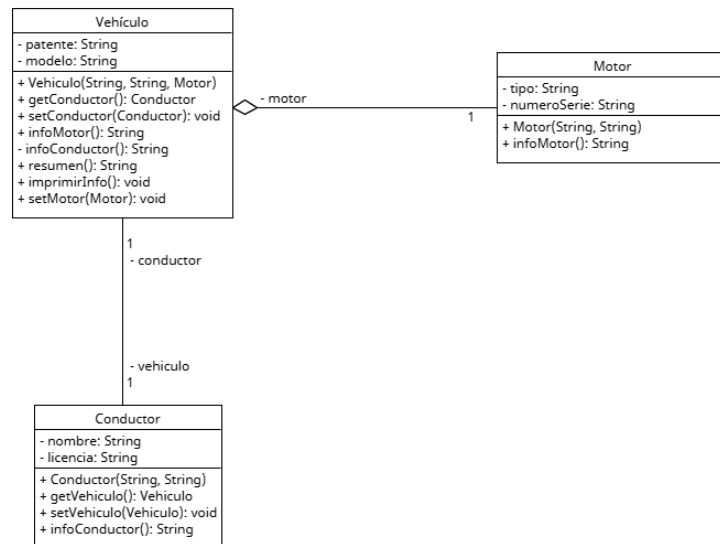
❖ Ejercicio 7 – Vehículo · Motor · Conductor

- **Consigna:**

Agregación Vehículo → Motor y **asociación bidireccional** Vehículo ↔ Conductor.

- **Archivos principales:**

- Vehiculo.java · Motor.java · Conductor.java
- Principal.java



❖ Ejercicio 8 – Documento · FirmaDigital · Usuario

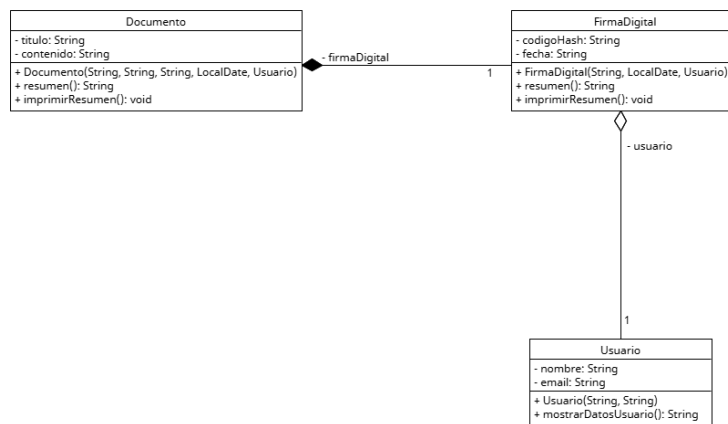
- **Consigna:**

Composición Documento → FirmaDigital.

Agregación con FirmaDigital → Usuario.

- **Archivos principales:**

- Documento.java · FirmaDigital.java · Usuario.java
- Principal.java



❖ Ejercicio 9 – CitaMédica · Paciente · Profesional

- **Consigna:**

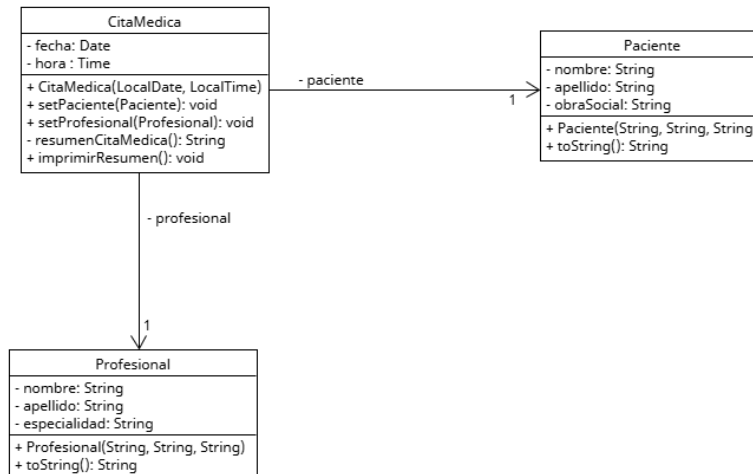
Asociación unidireccional desde CitaMédica → Paciente.

Misma relación CitaMédica → Profesional.

Formatear fecha/hora y mostrar el turno.

- **Archivos principales:**

- CitaMedica.java · Paciente.java · Profesional.java
- Principal.java



❖ Ejercicio 10 – CuentaBancaria · ClaveSeguridad · Titular

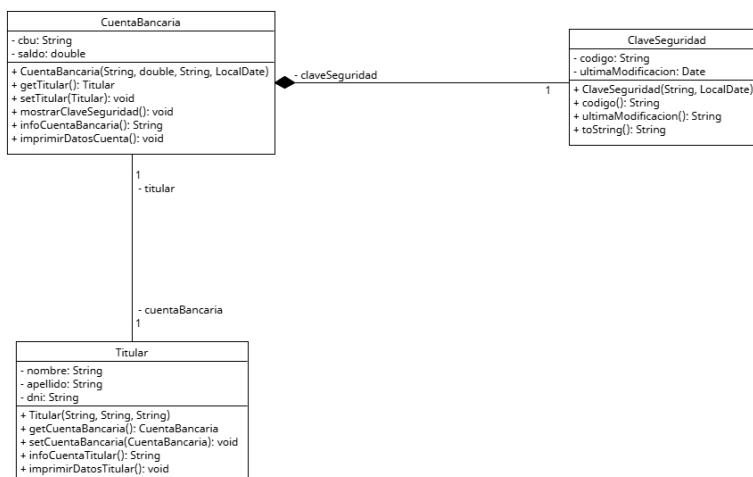
- **Consigna:**

Composición CuentaBancaria → ClaveSeguridad.

Asociación bidireccional CuentaBancaria ↔ Titular.

- **Archivos principales:**

- CuentaBancaria.java · ClaveSeguridad.java · Titular.java
- Principal.java



Dependencia de uso

❖ Ejercicio 11 – Reproductor · Canción · Artista

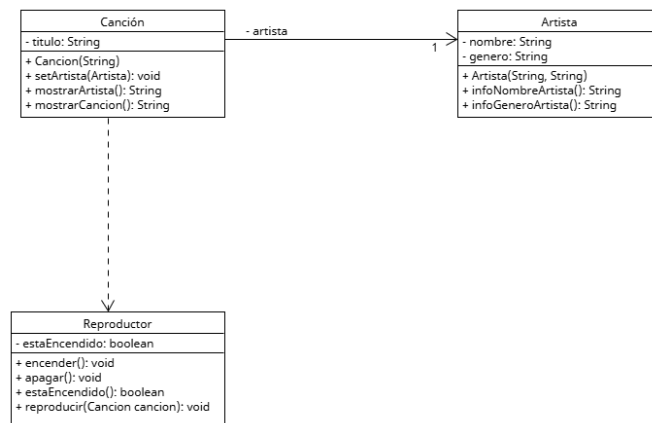
- **Consigna:**

Asociación unidireccional Canción → Artista.

Dependencia de uso: Reproductor.reproducir(Canción) recibe la canción por parámetro.

- **Archivos principales:**

- Reproductor.java · Cancion.java · Artista.java
- Principal.java



❖ Ejercicio 12 – Impuesto · Contribuyente · Calculadora

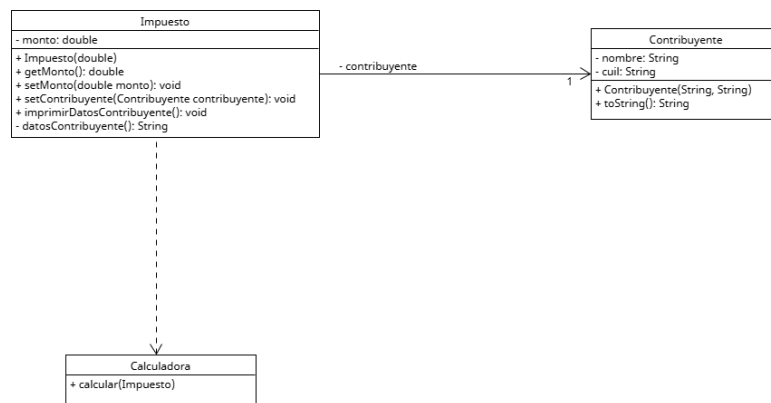
- **Consigna:**

Asociación unidireccional Impuesto → Contribuyente.

Dependencia de uso: Calculadora.calcular(Impuesto) usa el objeto sin almacenarlo.

- **Archivos principales:**

- Impuesto.java · Contribuyente.java · Calculadora.java
- Principal.java



Dependencia de creación

❖ Ejercicio 13 – GeneradorQR · Usuario · CódigoQR

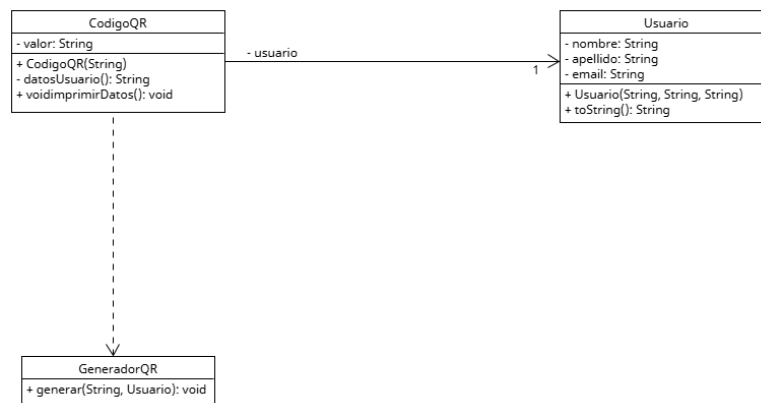
- **Consigna:**

Asociación unidireccional CódigoQR → Usuario.

Dependencia de creación: GeneradorQR.generar(String, Usuario) **crea** un CódigoQR y no lo conserva.

- **Archivos principales:**

- CódigoQR.java · Usuario.java · GeneradorQR.java
- Principal.java



❖ Ejercicio 14 – EditorVideo · Proyecto · Render

- **Consigna:**

Asociación unidireccional Render → Proyecto.

Dependencia de creación: EditorVideo.exportar(Formato, Proyecto) crea un Render (usa enum Formato) y no lo guarda.

- **Archivos principales:**

- EditorVideo.java · Proyecto.java · Render.java · Formato.java
- Principal.java

