

PROGRAMACIÓN II

Trabajo Práctico 5: Relaciones UML 1 a 1

Enlace a Github: <https://github.com/MarianoAstorga/Programacion2>

1. Pasaporte - Foto - Titular

a. Composición: Pasaporte → Foto

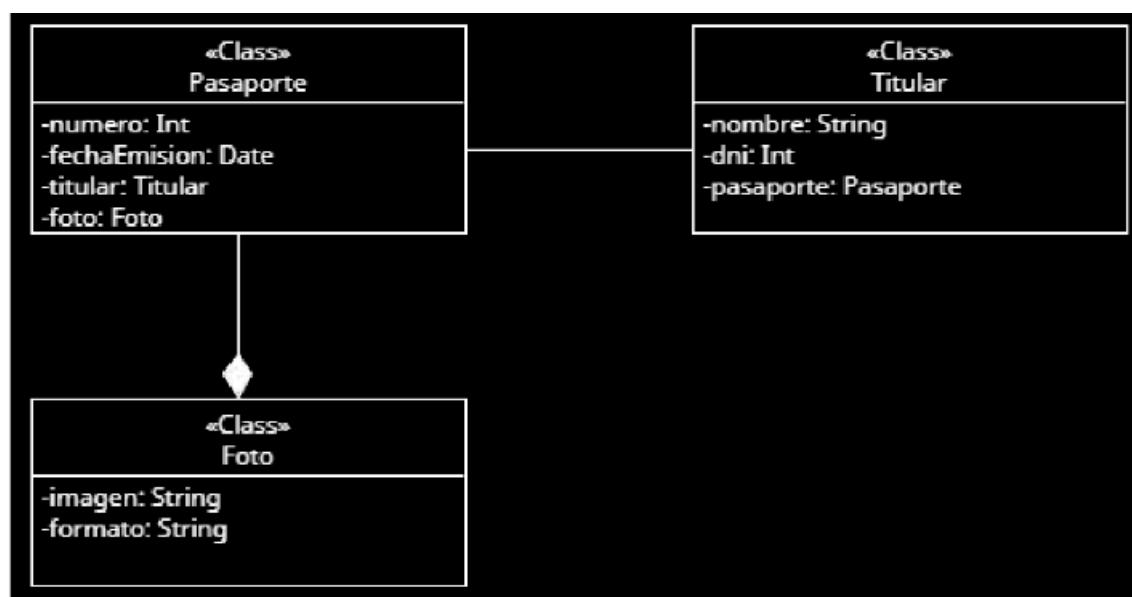
b. Asociación bidireccional: Pasaporte ↔ Titular

Clases y atributos:

i. Pasaporte: numero, fechaEmision

ii. Foto: imagen, formato

iii. Titular: nombre, dni



2. Celular - Batería - Usuario

a. Agregación: Celular → Batería

b. Asociación bidireccional: Celular ↔ Usuario

Clases y atributos:

- i. Celular: imei, marca, modelo
- ii. Batería: modelo, capacidad
- iii. Usuario: nombre, dni



3. Libro - Autor - Editorial

- Asociación unidireccional: Libro → Autor
- Agregación: Libro → Editorial

Clases y atributos:

- Libro: titulo, isbn
- Autor: nombre, nacionalidad
- Editorial: nombre, direccion

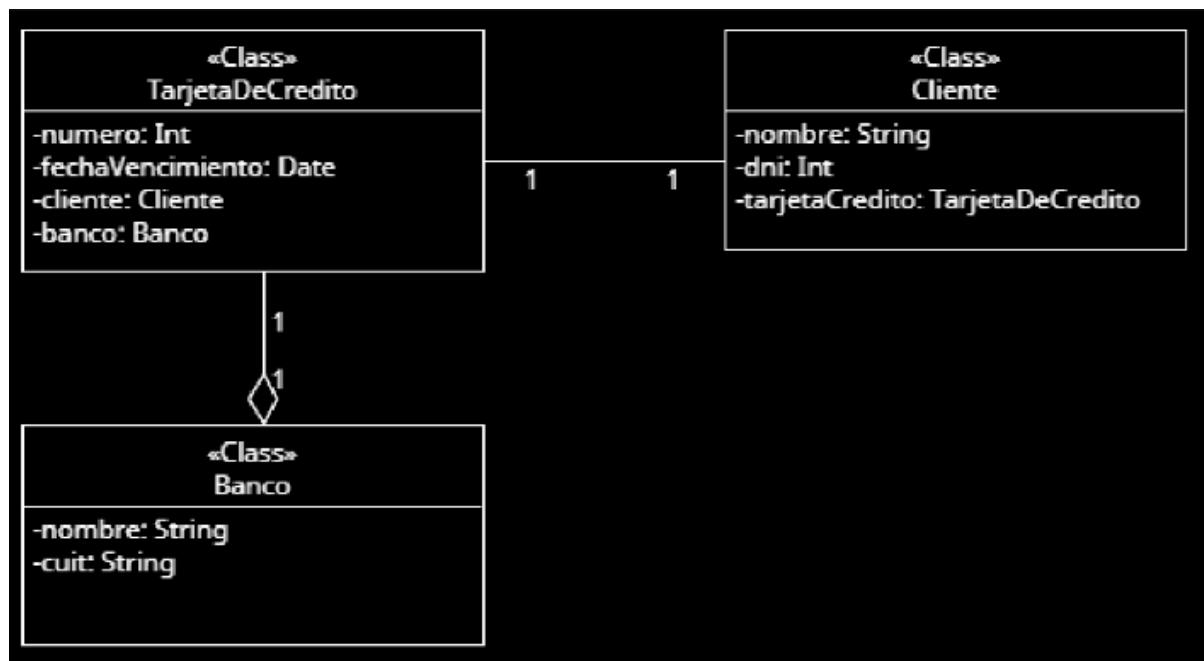


4. TarjetaDeCrédito - Cliente - Banco

- Asociación bidireccional: TarjetaDeCrédito ↔ Cliente
- Agregación: TarjetaDeCrédito → Banco

Clases y atributos:

- TarjetaDeCrédito: numero, fechaVencimiento
- Cliente: nombre, dni
- Banco: nombre, cuit

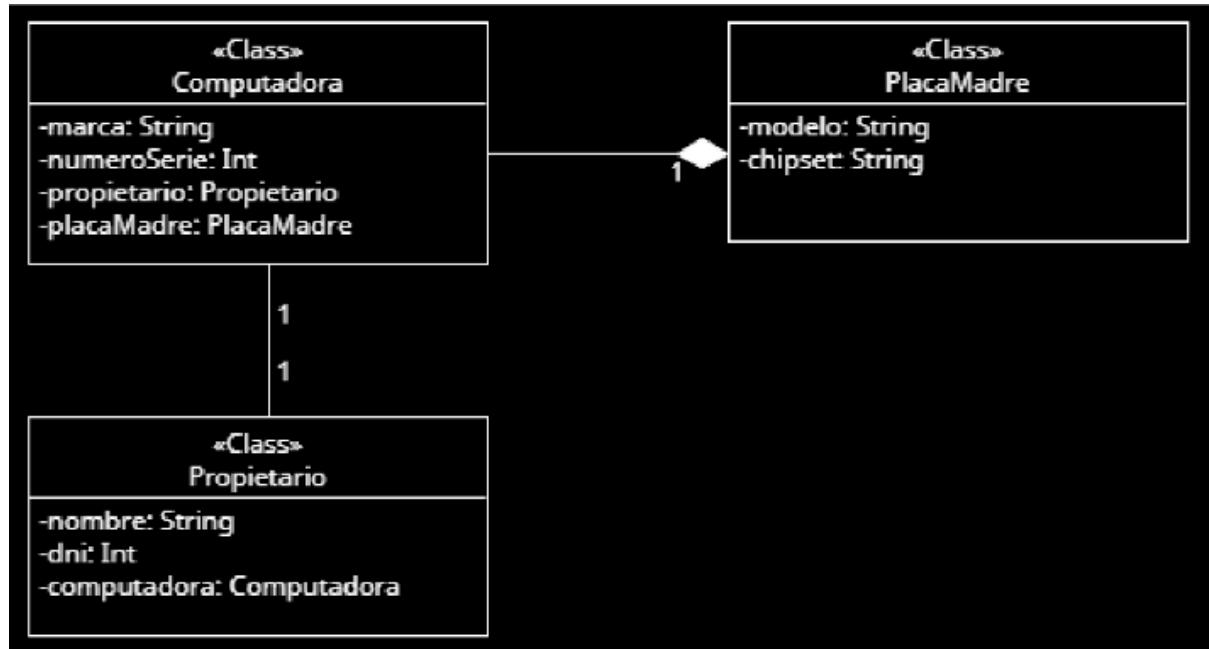


5. Computadora - PlacaMadre - Propietario

- a. Composición: Computadora → PlacaMadre
- b. Asociación bidireccional: Computadora ↔ Propietario

Clases y atributos:

- i. Computadora: marca, numeroSerie
- ii. PlacaMadre: modelo, chipset
- iii. Propietario: nombre, dni

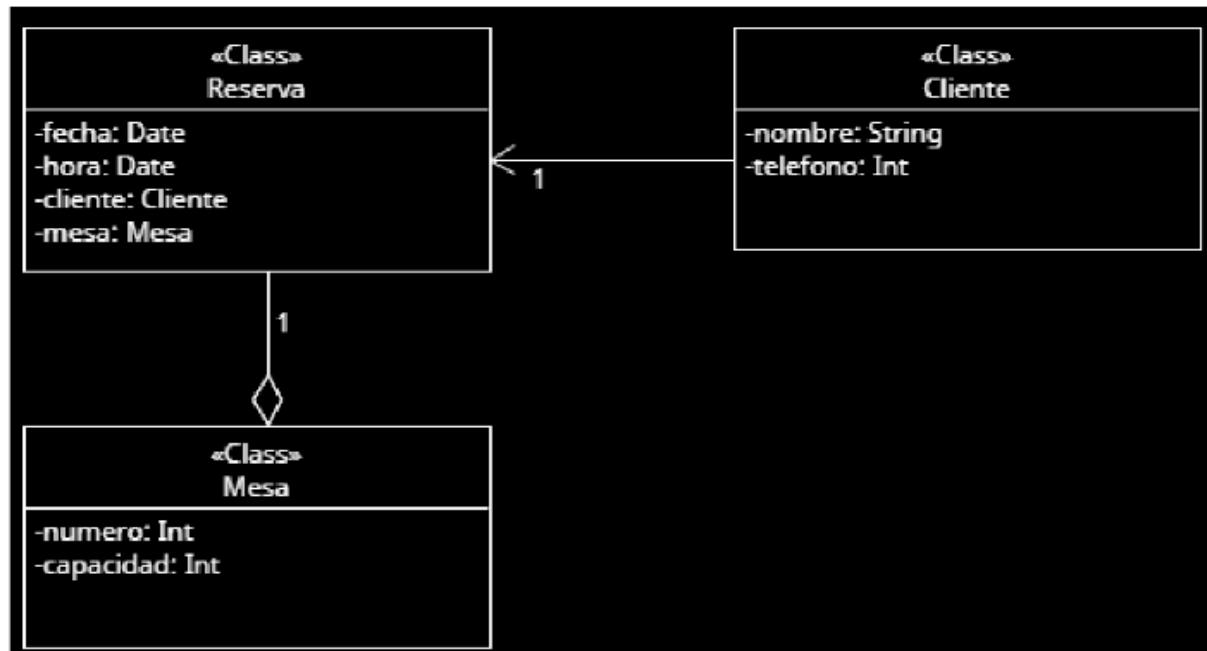


6. Reserva - Cliente - Mesa

- Asociación unidireccional: Reserva → Cliente
- Agregación: Reserva → Mesa

Clases y atributos:

- Reserva: fecha, hora
- Cliente: nombre, telefono
- Mesa: numero, capacidad

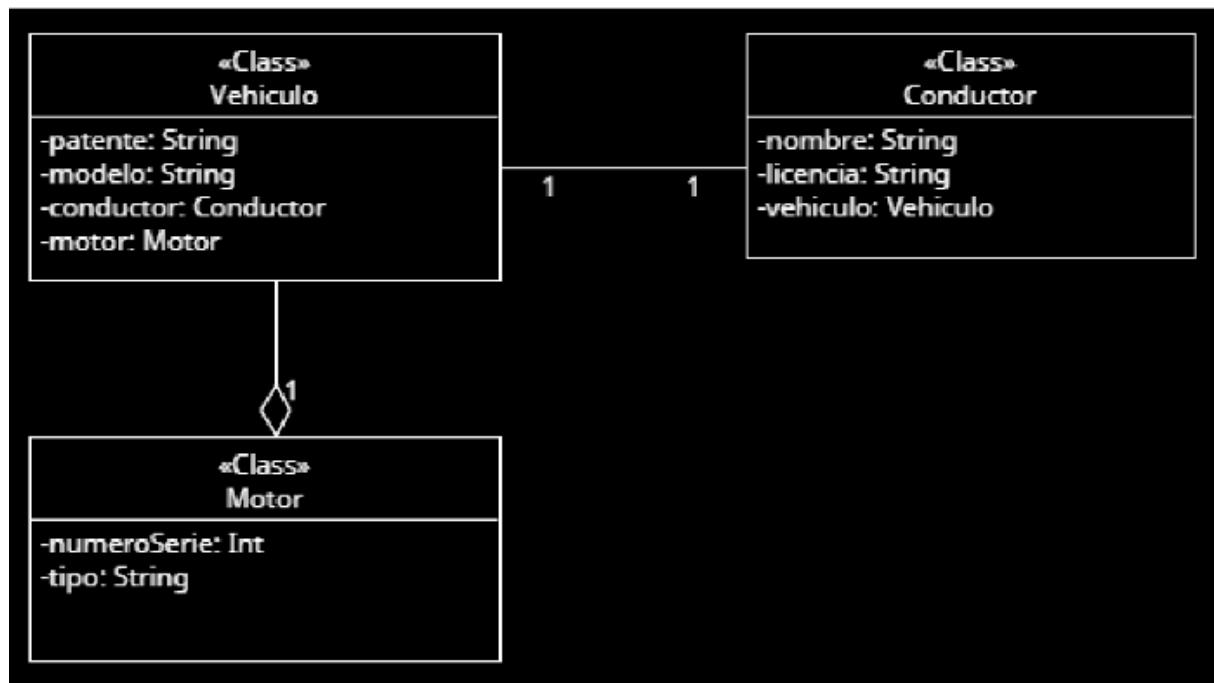


7. Vehículo - Motor - Conductor

- a. Agregación: Vehículo → Motor
- b. Asociación bidireccional: Vehículo ↔ Conductor

Clases y atributos:

- i. Vehículo: patente, modelo
- ii. Motor: tipo, numeroSerie
- iii. Conductor: nombre, licencia

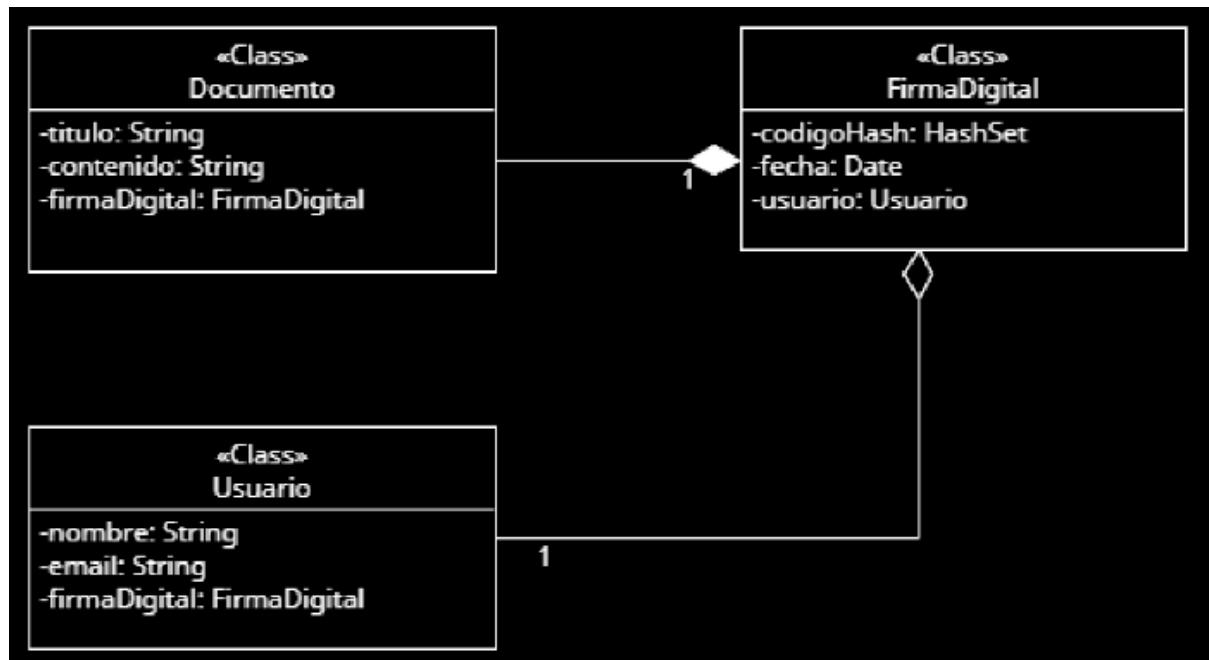


8. Documento - FirmaDigital - Usuario

- a. Composición: Documento → FirmaDigital
- b. Agregación: FirmaDigital → Usuario

Clases y atributos:

- i. Documento: titulo, contenido
- ii. FirmaDigital: codigoHash, fecha
- iii. Usuario: nombre, email

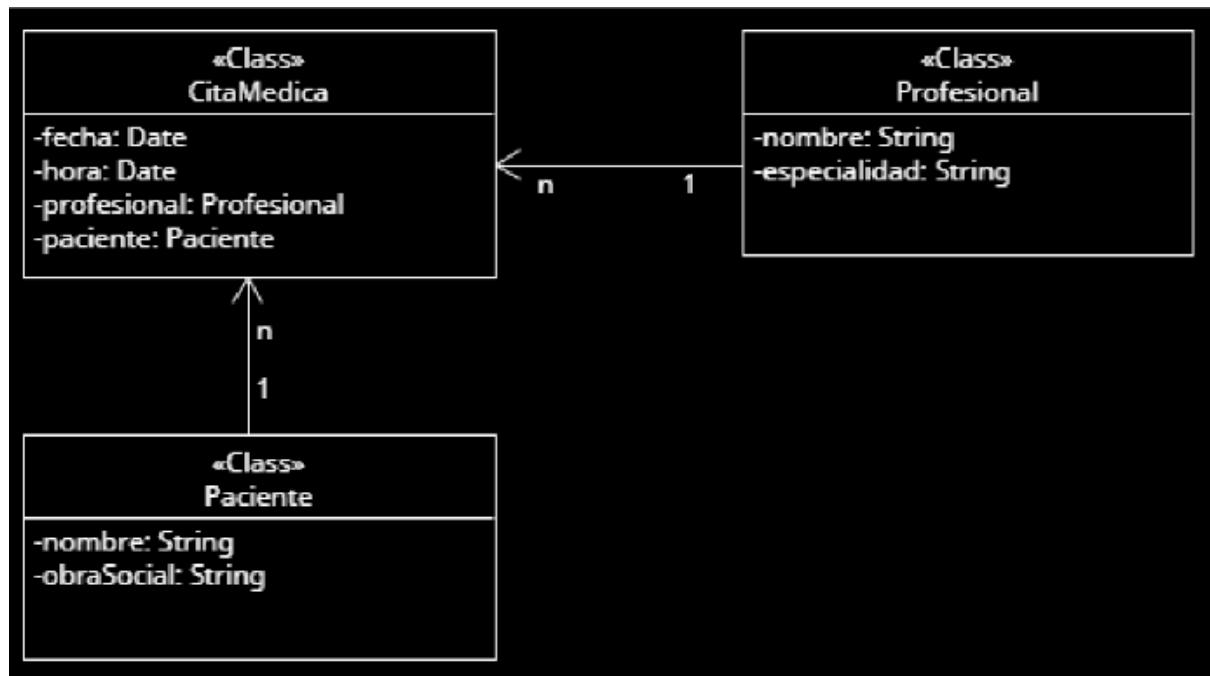


9. CitaMédica - Paciente - Profesional

- a. Asociación unidireccional: CitaMédica → Paciente,
- b. Asociación unidireccional: CitaMédica → Profesional

Clases y atributos:

- i. CitaMédica: fecha, hora
- ii. Paciente: nombre, obraSocial
- iii. Profesional: nombre, especialidad

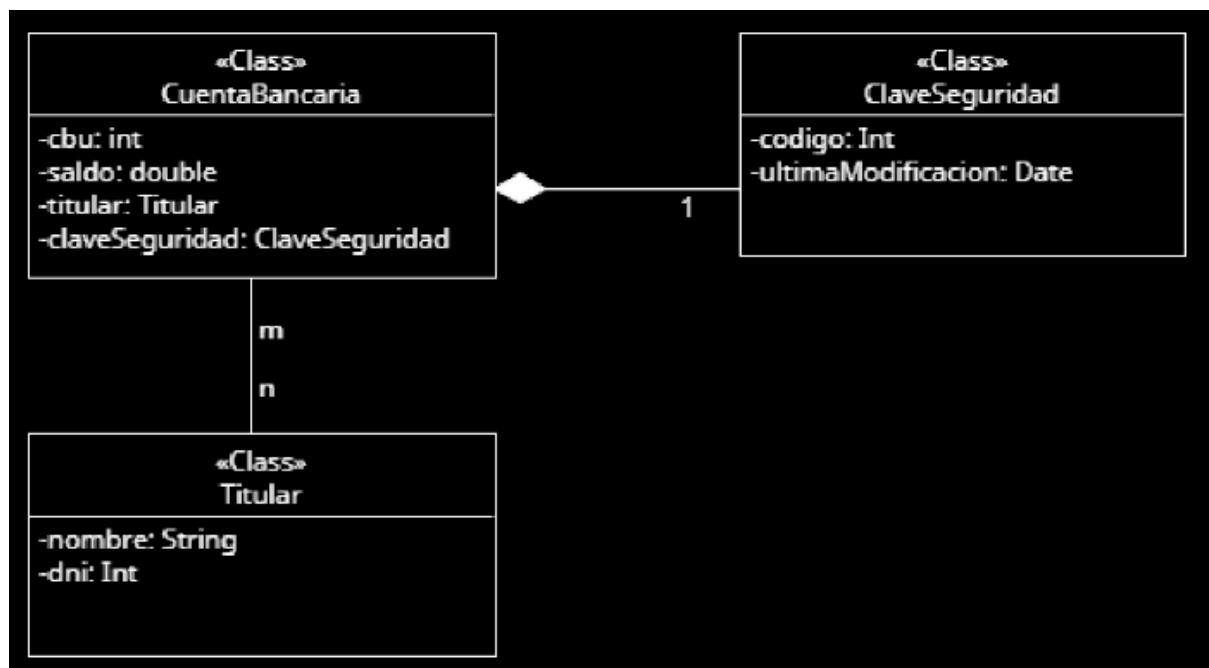


10. CuentaBancaria - ClaveSeguridad - Titular

- Composición: CuentaBancaria → ClaveSeguridad
- Asociación bidireccional: CuentaBancaria ↔ Titular

Clases y atributos:

- CuentaBancaria: cbu, saldo
- ClaveSeguridad: codigo, ultimaModificacion
- Titular: nombre, dni.

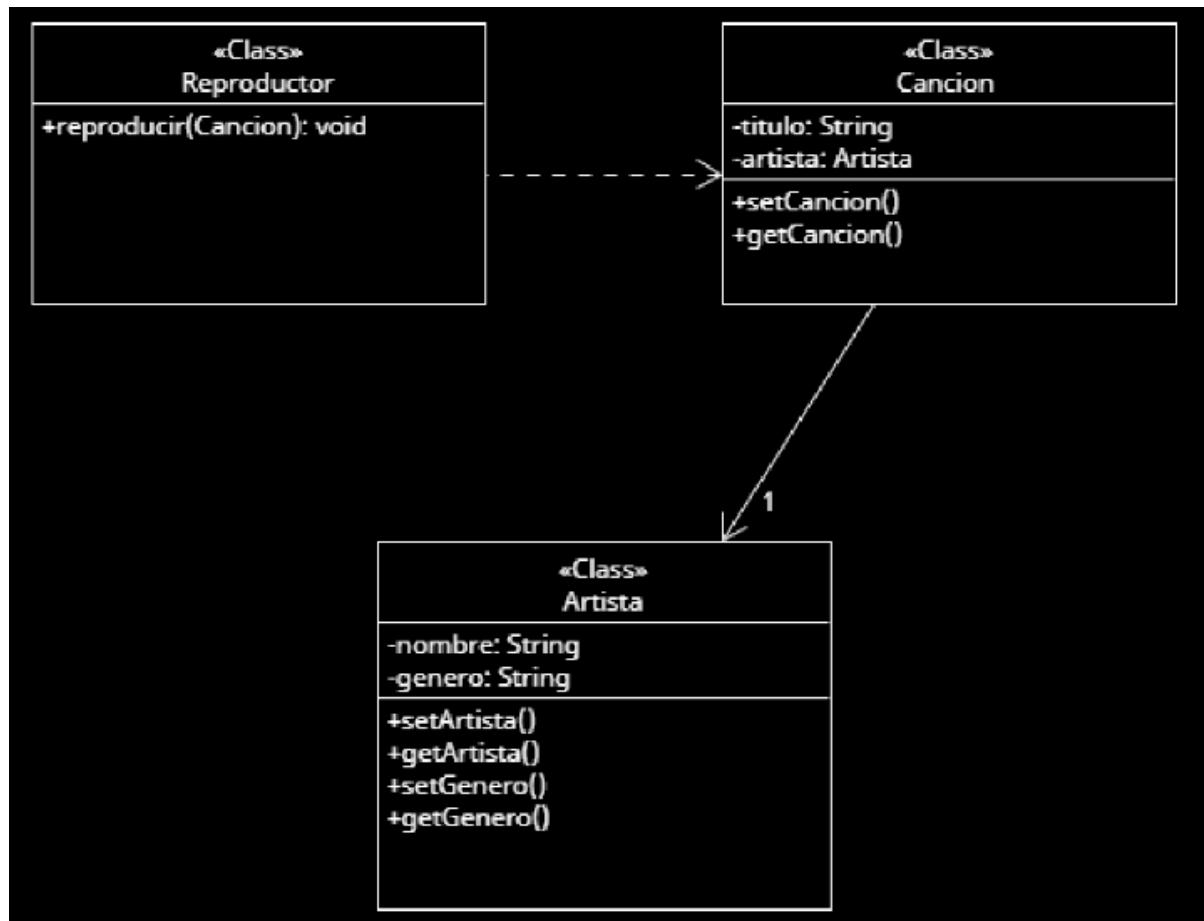


11. Reproductor - Canción - Artista

- Asociación unidireccional: Canción → Artista
- Dependencia de uso: Reproductor.reproducir(Cancion)

Clases y atributos:

- Canción: título.
- Artista: nombre, genero.
- Reproductor->método: void reproducir(Cancion cancion)

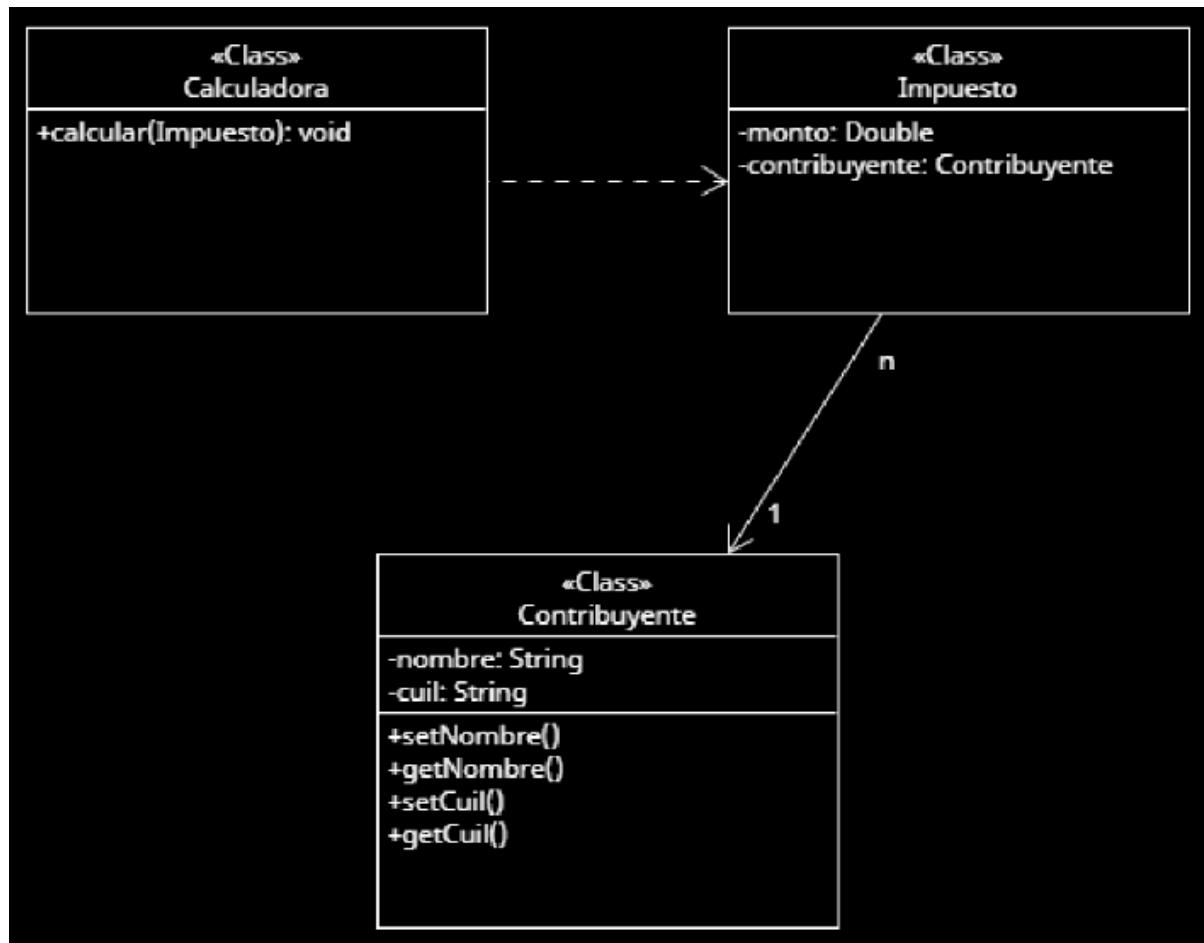


12. Impuesto - Contribuyente - Calculadora

- Asociación unidireccional: Impuesto → Contribuyente
- Dependencia de uso: Calculadora.calcular(Impuesto)

Clases y atributos:

- Impuesto: monto.
- Contribuyente: nombre, cuil.
- Calculadora->método: void calcular(Impuesto impuesto)

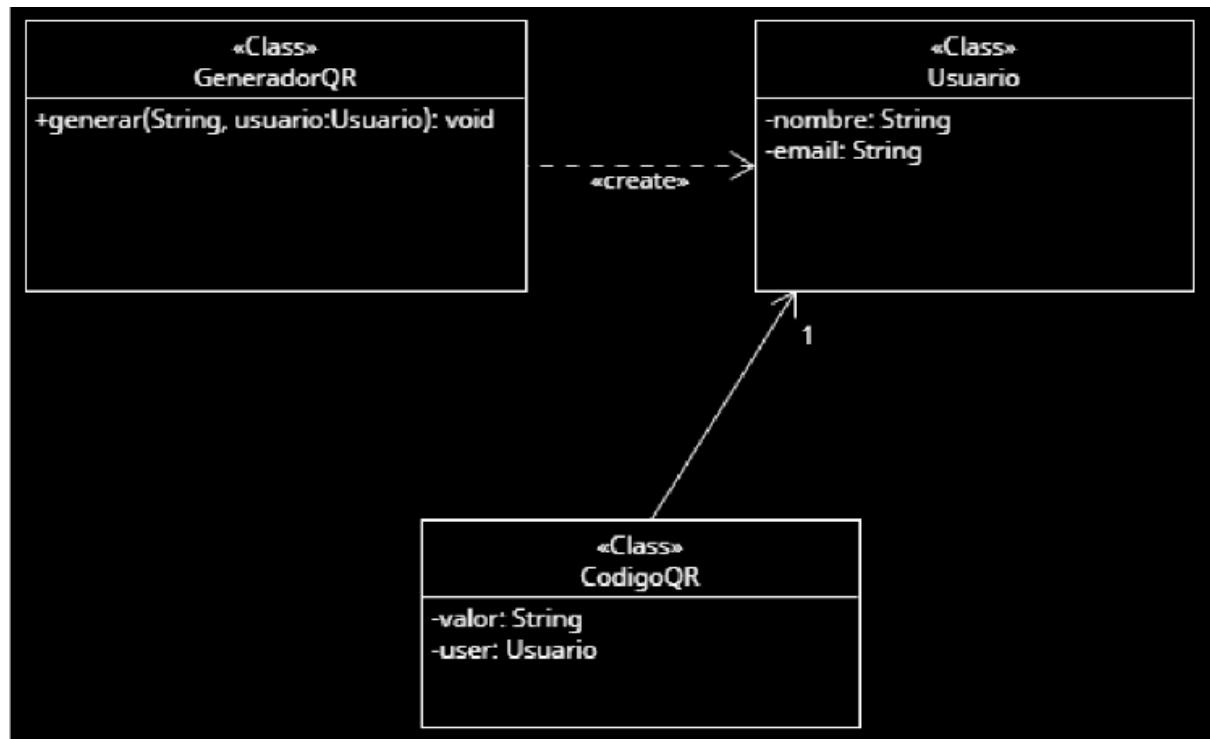


13. GeneradorQR - Usuario - CódigoQR

- a. Asociación unidireccional: CódigoQR → Usuario
- b. Dependencia de creación: GeneradorQR.generar(String, Usuario)

Clases y atributos:

- i. CódigoQR: valor.
- ii. Usuario: nombre, email.
- iii. GeneradorQR->método: void generar(String valor, Usuario usuario)



14. EditorVideo - Proyecto - Render

- a. Asociación unidireccional: Render → Proyecto
- b. Dependencia de creación: EditorVideo.exportar(String, Proyecto)
- c. Clases y atributos:
 - i. Render: formato.
 - ii. Proyecto: nombre, duracionMin.
 - iii. EditorVideo->método: void exportar(String formato, Proyecto proyecto)

