

TRABAJO PRÁCTICO RELACIONES

UML

AÑO: 2025

MATERIA: PROGRAMACIÓN II

PROFESORA: SOFIA RAIÁ

ALUMNO: LAUTARO ARIEL CEJAS

COMISIÓN: 2

Diagrama Ejercicio 1

Ejercicio 1: Pasaporte - Foto - Titular
Relaciones:
• Composición: Pasaporte — Foto (◆)
• Asociación Bidireccional: Pasaporte ↔ Titular

Asociación Bidireccional
Ambas clases se conocen mutuamente

Composición
La Foto se crea con el Pasaporte y depende de él

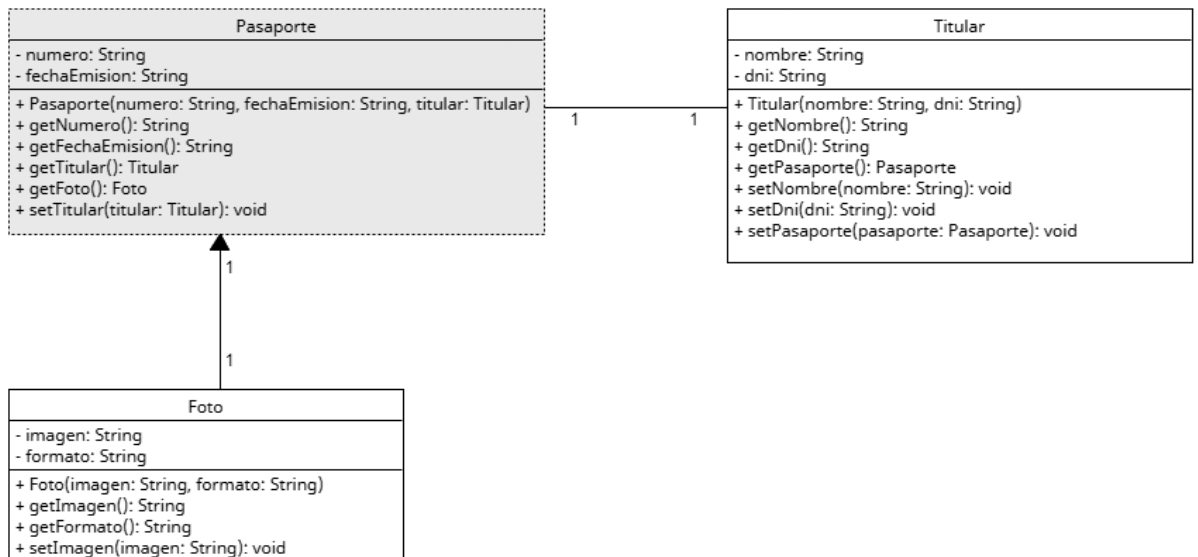


Diagrama Ejercicio 2

Ejercicio 2: Celular - Batería - Usuario
Relaciones:
• Agregación: Celular — Batería (◇)
• Asociación Bidireccional: Celular ↔ Usuario

Asociación Bidireccional
Ambas clases se conocen mutuamente

Agregación
La Batería puede existir independientemente del Celular

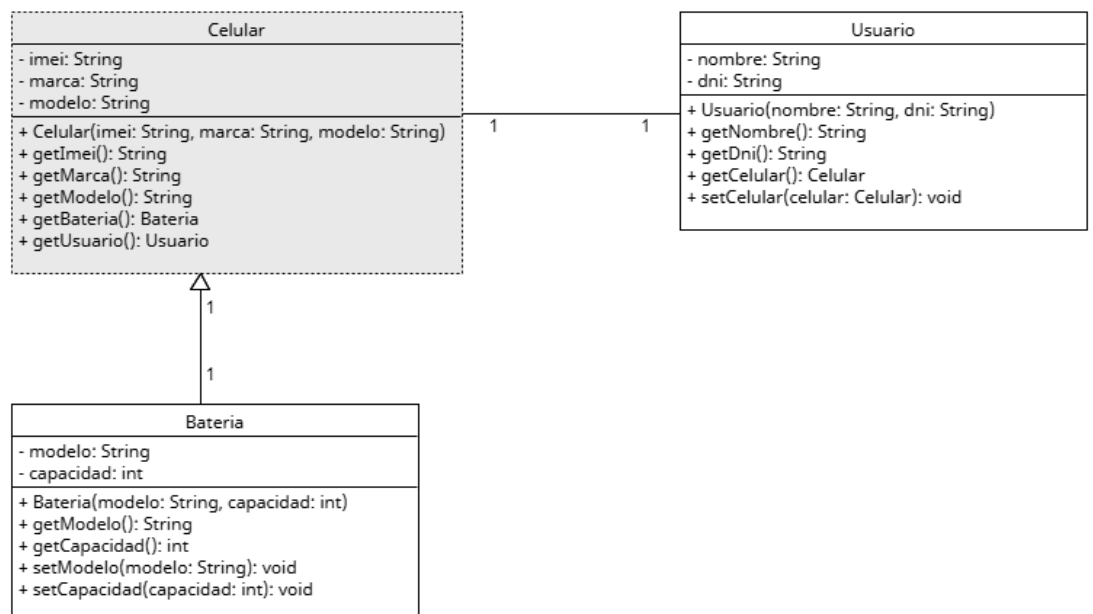


Diagrama Ejercicio 3

Ejercicio 3: Libro - Autor - Editorial

Relaciones:

- Asociación Unidireccional: Libro → Autor
- Agregación: Libro → Editorial (◇)

Asociación Unidireccional
Solo el Libro conoce
al Autor

Agregación
La Editorial puede existir
independientemente del Libro

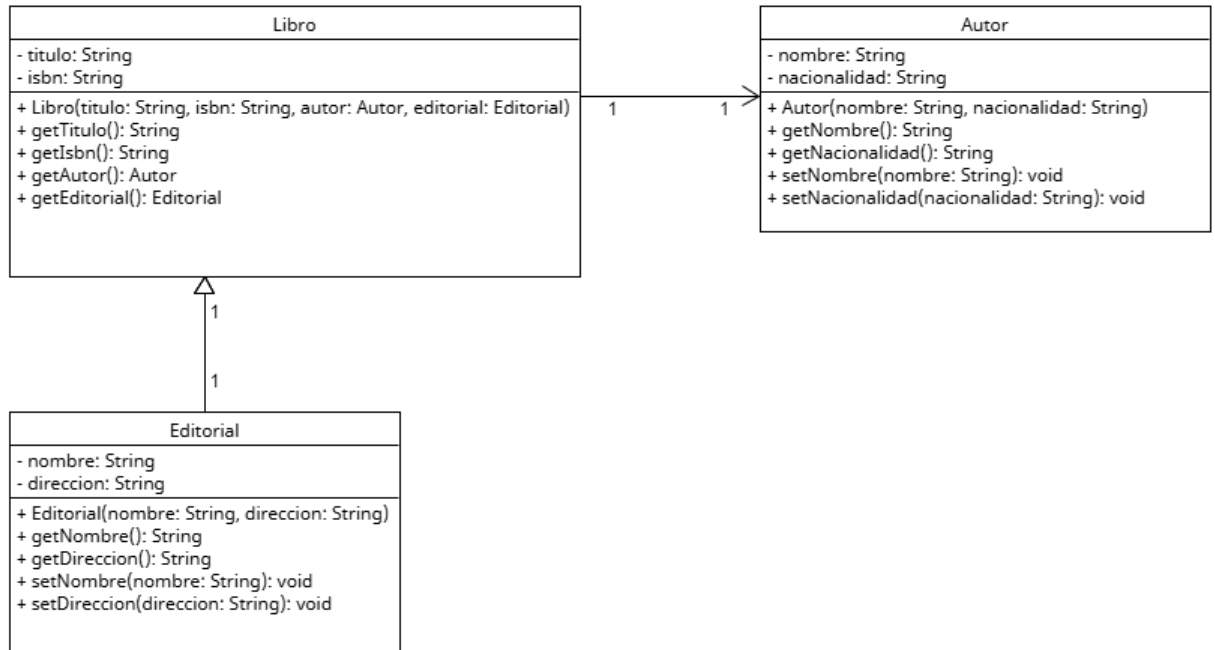


Diagrama Ejercicio 4

Ejercicio 4: TarjetaDeCrédito - Cliente - Banco

Relaciones:

- Asociación Bidireccional: TarjetaDeCrédito ↔ Cliente
- Agregación: TarjetaDeCrédito → Banco (◇)

Asociación Bidireccional
Ambas clases se conocen
mutuamente

Agregación
El Banco puede existir
independientemente de la Tarjeta

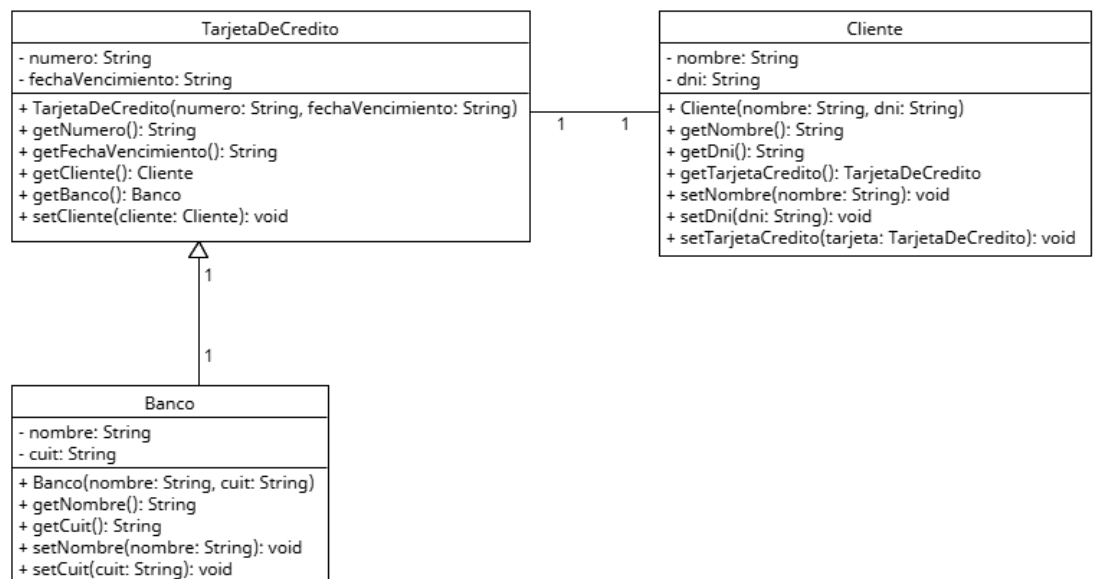


Diagrama Ejercicio 5

Ejercicio 5: Computadora - PlacaMadre - Propietario
Relaciones:
• Composición: Computadora — PlacaMadre (◆)
• Asociación Bidireccional: Computadora ↔ Propietario

Asociación Bidireccional
Ambas clases se conocen mutuamente

Composición
La PlacaMadre se crea con la Computadora y depende de ella

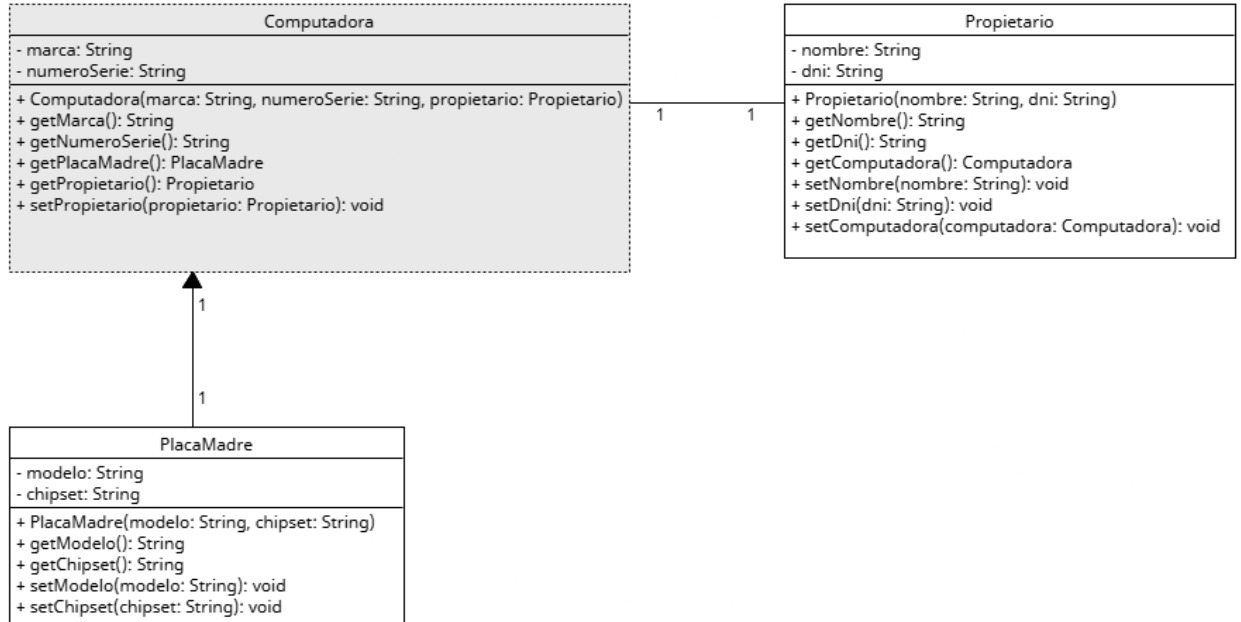


Diagrama Ejercicio 6

Ejercicio 6: Reserva - Cliente - Mesa
Relaciones:
• Asociación Unidireccional: Reserva → Cliente
• Agregación: Reserva — Mesa (◇)

Asociación Unidireccional
Solo la Reserva conoce al Cliente

Agregación
La Mesa puede existir independientemente de la Reserva

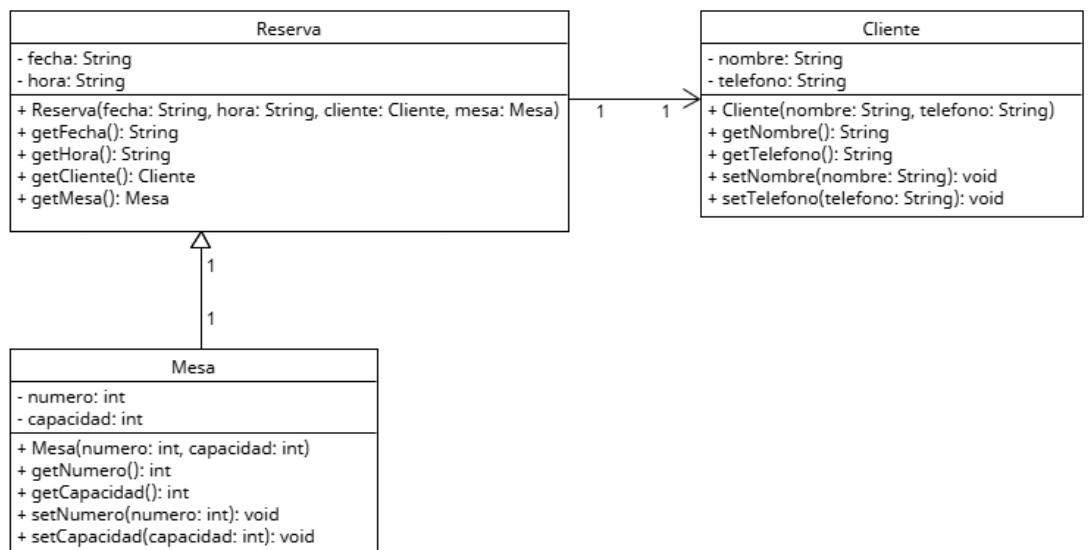


Diagrama Ejercicio 7

Ejercicio 7: Vehículo - Motor - Conductor

Relaciones:

- Agregación: Vehículo \dashv Motor (\diamond)
- Asociación Bidireccional: Vehículo \longleftrightarrow Conductor

Asociación Bidireccional
Ambas clases se conocen mutuamente

Agregación
El Motor puede existir independientemente del Vehículo

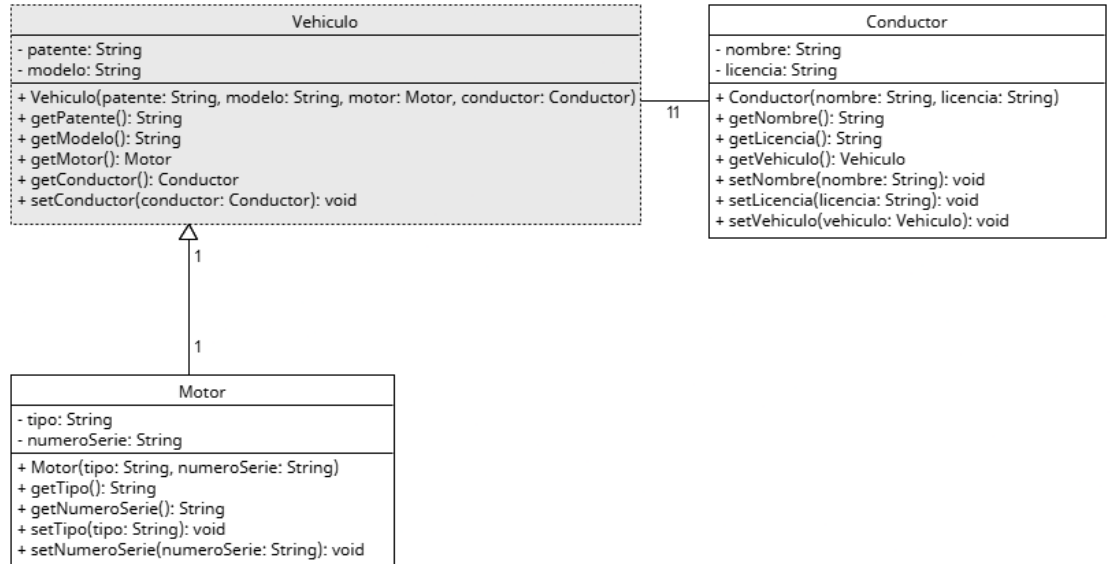


Diagrama Ejercicio 8

Ejercicio 8: Documento - FirmaDigital - Usuario

Relaciones:

- Composición: Documento \blacklozenge FirmaDigital (\blacklozenge)
- Agregación: FirmaDigital \dashv Usuario (\diamond)

Composición
La FirmaDigital se crea con el Documento y depende de él

Agregación
El Usuario puede existir independientemente de la Firma

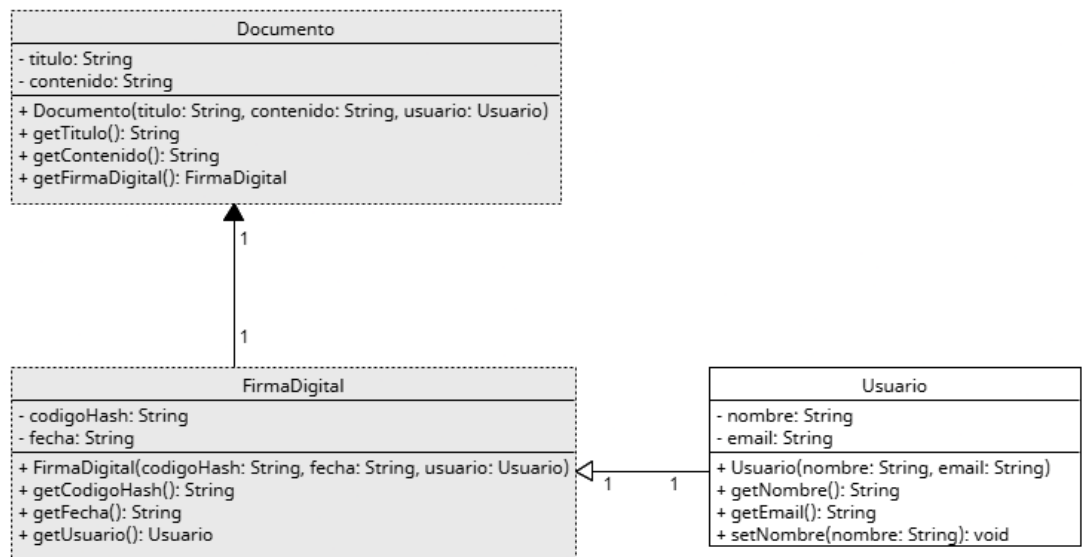


Diagrama Ejercicio 9

Ejercicio 9: CitaMédica - Paciente - Profesional
Relaciones:

- Asociación Unidireccional: CitaMédica → Paciente
- Asociación Unidireccional: CitaMédica → Profesional

Asociación Unidireccional
Solo la CitaMédica conoce
al Paciente y Profesional

Dos relaciones independientes
La cita conecta paciente
y profesional

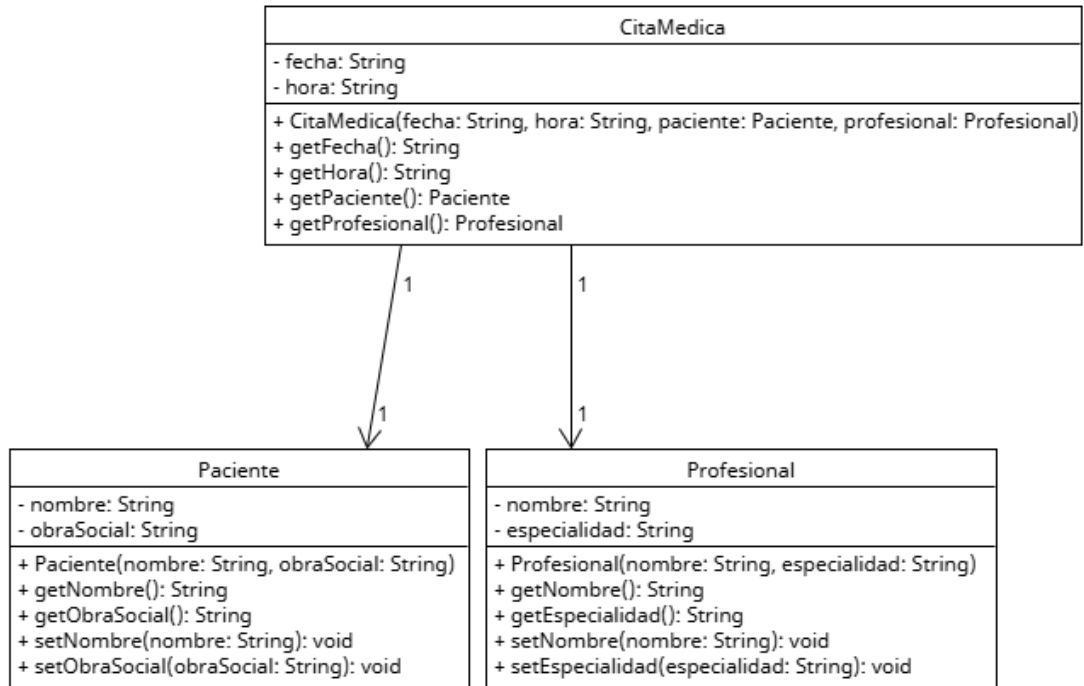


Diagrama Ejercicio 10

Ejercicio 10: CuentaBancaria - ClaveSeguridad - Titular
Relaciones:

- Composición: CuentaBancaria → ClaveSeguridad (◆)
- Asociación Bidireccional: CuentaBancaria ↔ Titular

Asociación Bidireccional
Ambas clases se conocen
mutuamente

Composición
La ClaveSeguridad se crea con
la Cuenta y depende de ella

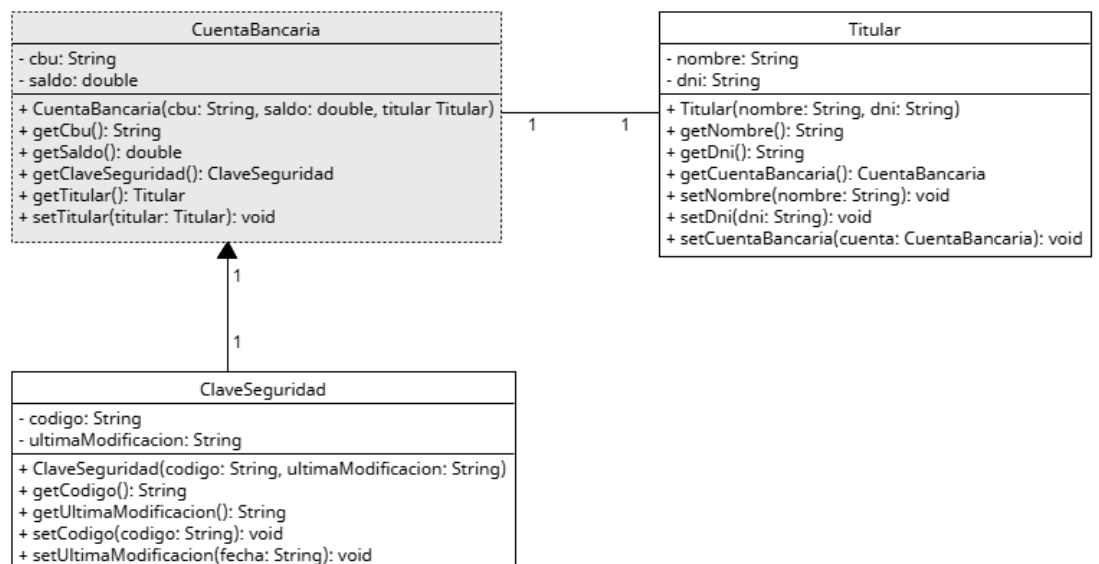


Diagrama Ejercicio 11

Ejercicio 11: Reproductor - Canción - Artista

Relaciones:

- Asociación Unidireccional: Canción → Artista
- Dependencia de Uso: Reproductor → Canción

Dependencia de Uso

El Reproductor usa Canción como parámetro solamente

Asociación Unidireccional

Solo la Canción conoce al Artista

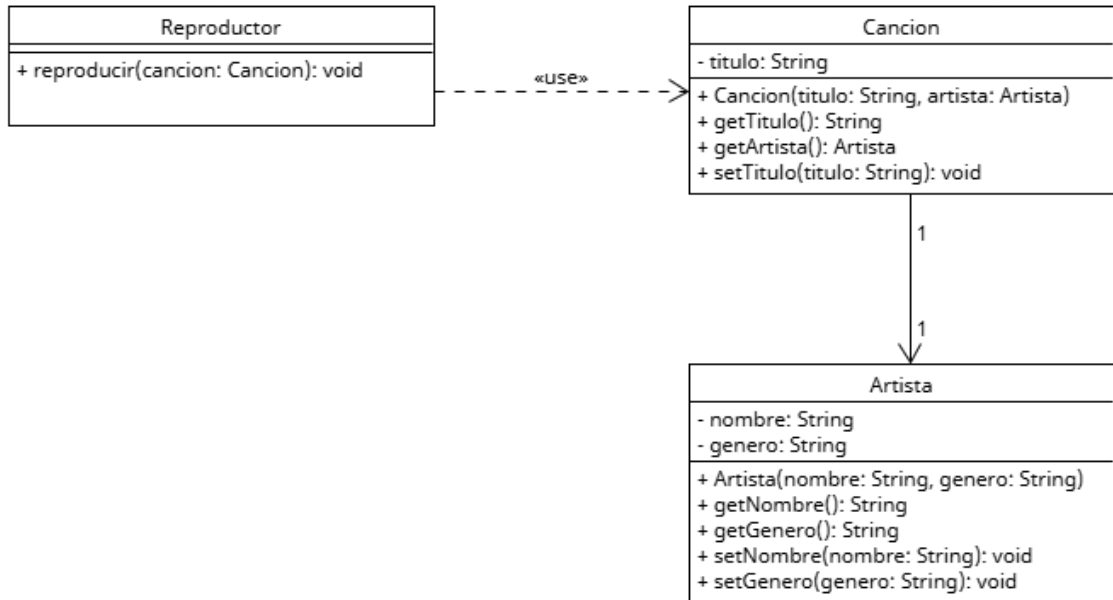


Diagrama Ejercicio 12

Ejercicio 12: Calculadora - Impuesto - Contribuyente

Relaciones:

- Asociación Unidireccional: Impuesto → Contribuyente (Impuesto conoce a Contribuyente)
- Dependencia de Uso: Calculadora → Impuesto (Calculadora usa Impuesto como parámetro)

Dependencia de Uso

Calculadora no tiene atributo Impuesto, solo lo recibe temporalmente en parámetros de métodos

Asociación Unidireccional

Solo Impuesto tiene referencia a Contribuyente, no al revés

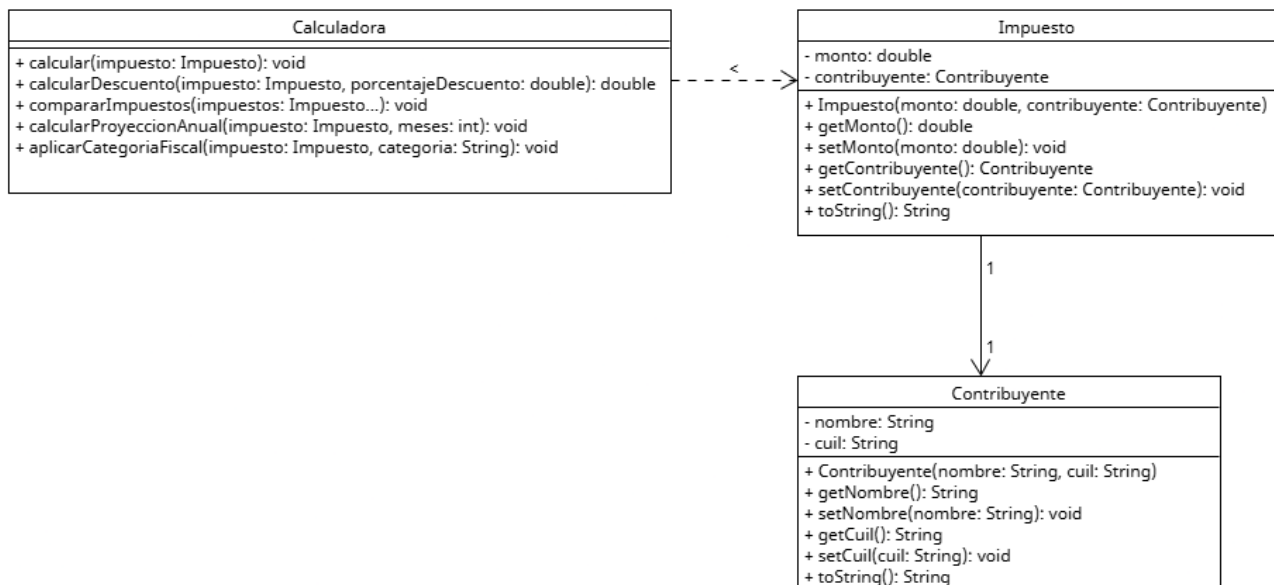


Diagrama Ejercicio 13

Ejercicio 13: GeneradorQR - CódigoQR - Usuario

Relaciones detalladas:

- Asociación Unidireccional: CódigoQR → Usuario (CódigoQR mantiene referencia persistente a Usuario via atributo privado)
- Dependencia de Creación: GeneradorQR → CódigoQR (GeneradorQR crea instancias de CódigoQR en tiempo de ejecución dentro de sus métodos, sin almacenarlas como atributos - las instancias se descartan al finalizar el método)
- Dependencia de Uso: GeneradorQR → Usuario (GeneradorQR recibe Usuario como parámetro en múltiples métodos, lo usa temporalmente sin almacenarlo)

Dependencia de Creación Detallada

- No atributo en GeneradorQR para CódigoQR
- Creación local en métodos como generar(), generarConOpciones(), etc.
- Ciclo de vida de CódigoQR limitado al scope del método
- Demuestra uso temporal sin contención persistente

Asociación Unidireccional Detallada

- CódigoQR tiene atributo 'usuario: Usuario' (referencia persistente)
- Usuario no tiene referencia de vuelta a CódigoQR
- Dirección: Solo CódigoQR conoce a Usuario
- Multiplicidad 1:1 (un CódigoQR pertenece a un Usuario)

Dependencia de Uso Detallada

- Usuario pasado como parámetro en todos los métodos
- No almacenado como atributo en GeneradorQR
- Acceso temporal para operaciones como impresión o asociación

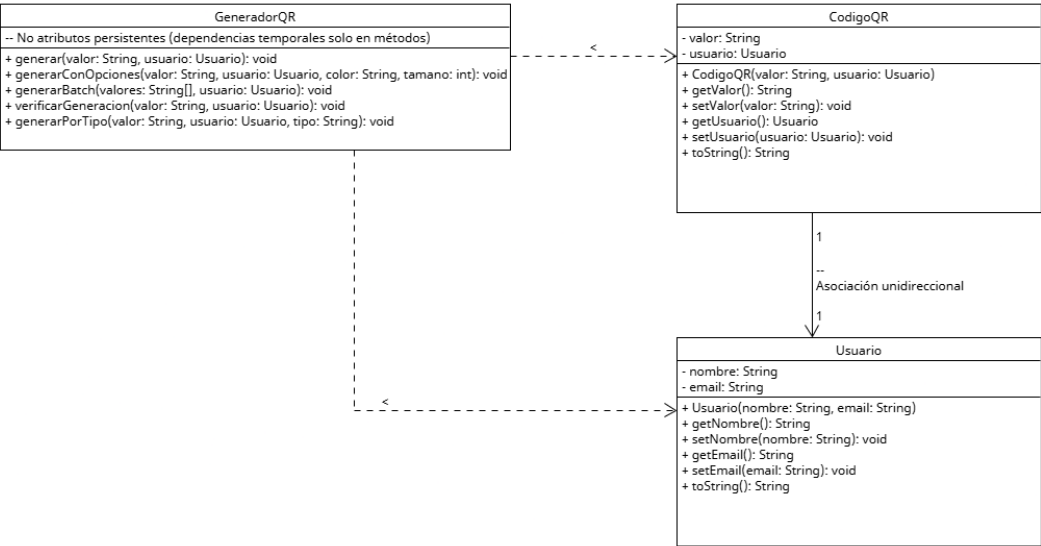


Diagrama Ejercicio 14

Ejercicio 14

- Asociación Unidireccional: Render → Proyecto
- Dependencia de Creación: EditorVideo → Render
- Dependencia de Uso: EditorVideo → Proyecto

Dependencia de Creación

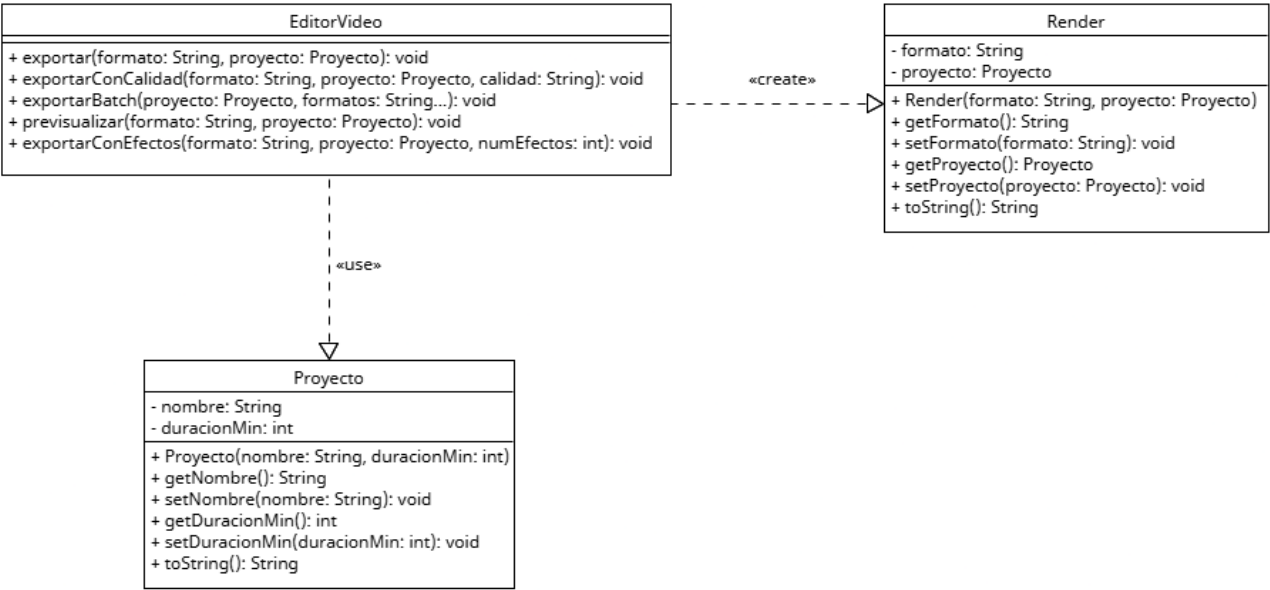
- EditorVideo crea Render temporalmente

Dependencia de Uso

- Usa Proyecto como parámetro

Asociación

- Unidireccional 1:1



Repositorio en GitHub (clic en la imagen)

