

TP 5 – Relaciones UML 1 a 1

Objetivo:

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.

Caso práctico:

Desarrollar los siguientes ejercicios en Java. Cada uno deberá incluir:

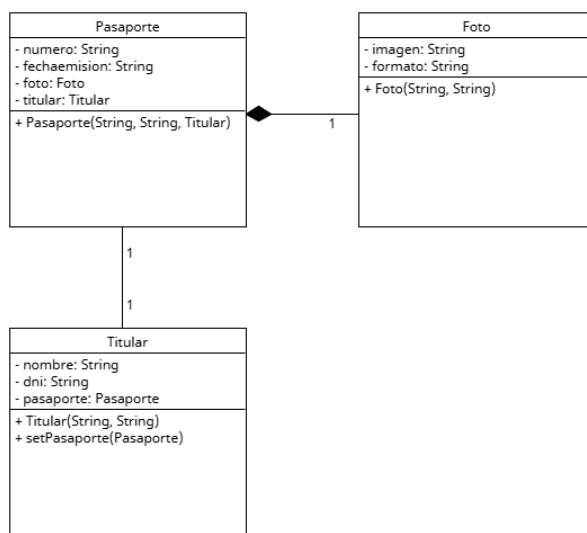
- Diagrama UML
- Tipo de relación (asociación, agregación, composición, dependencia)
- Dirección (unidireccional o bidireccional)
- Implementación de las clases con atributos y relaciones definidas

Ejercicios de Relaciones 1 a 1

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



```
class Foto {
    private String imagen;
    private String formato;

    public Foto(String imagen, String formato) {
        this.imagen = imagen;
        this.formato = formato;
    }
}

class Titular {
    private String nombre;
    private String dni;
    private Pasaporte pasaporte;

    public Titular(String nombre, String dni) {
        this.nombre = nombre;
        this.dni = dni;
    }

    public void setPasaporte(Pasaporte pasaporte) {
        this.pasaporte = pasaporte;
    }
}

class Pasaporte {
    private String numero;
    private String fechaEmision;
    private Foto foto;
    private Titular titular;

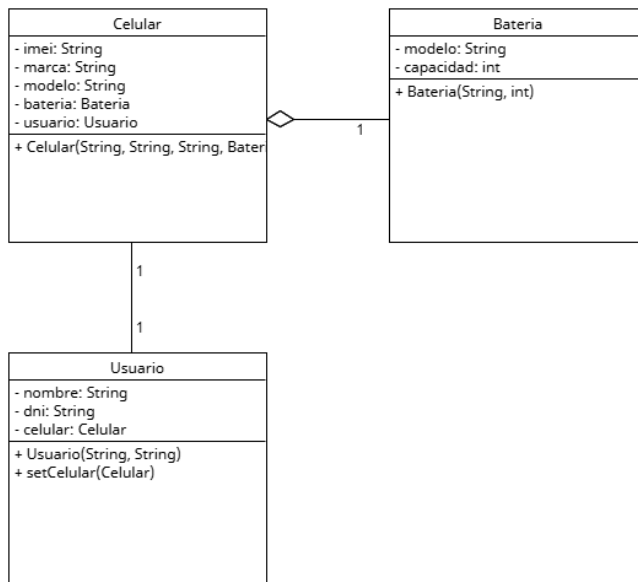
    public Pasaporte(String numero, String fechaEmision, Titular titular) {
        this.numero = numero;
        this.fechaEmision = fechaEmision;
        this.foto = new Foto("imagen.jpg", "jpg"); // composición
        this.titular = titular;
        titular.setPasaporte(this);
    }
}
```

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



```
class Batería {
    private String modelo;
    private int capacidad;

    public Batería(String modelo, int capacidad) {
        this.modelo = modelo;
        this.capacidad = capacidad;
    }
}
```

```
class Usuario {
    private String nombre;
    private String dni;
    private Celular celular;

    public Usuario(String nombre, String dni) {
        this.nombre = nombre;
        this.dni = dni;
    }
}
```

```
}

    public void setCelular(Celular celular) {
        this.celular = celular;
    }
}

class Celular {
    private String imei;
    private String marca;
    private String modelo;
    private Bateria bateria;
    private Usuario usuario;

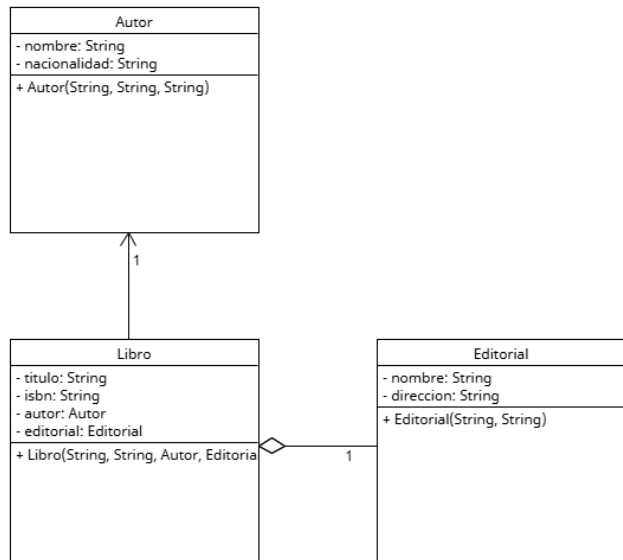
    public Celular(String imei, String marca, String modelo, Bateria bateria, Usuario usuario) {
        this.imei = imei;
        this.marca = marca;
        this.modelo = modelo;
        this.bateria = bateria; // agregación
        this.usuario = usuario;
        usuario.setCelular(this);
    }
}
```

3. Libro - Autor – Editorial

- a. Asociación unidireccional: **Libro → Autor**
- b. Agregación: **Libro → Editorial**

Clases y atributos:

- i. Libro: titulo, isbn
- ii. Autor: nombre, nacionalidad
- iii. Editorial: nombre, dirección



```
class Autor {
    private String nombre;
    private String nacionalidad;

    public Autor(String nombre, String nacionalidad) {
        this.nombre = nombre;
        this.nacionalidad = nacionalidad;
    }
}
```

```
class Editorial {
    private String nombre;
    private String direccion;

    public Editorial(String nombre, String direccion) {
        this.nombre = nombre;
        this.direccion = direccion;
    }
}
```

```
class Libro {
    private String titulo;
    private String isbn;
    private Autor autor;
    private Editorial editorial;
```

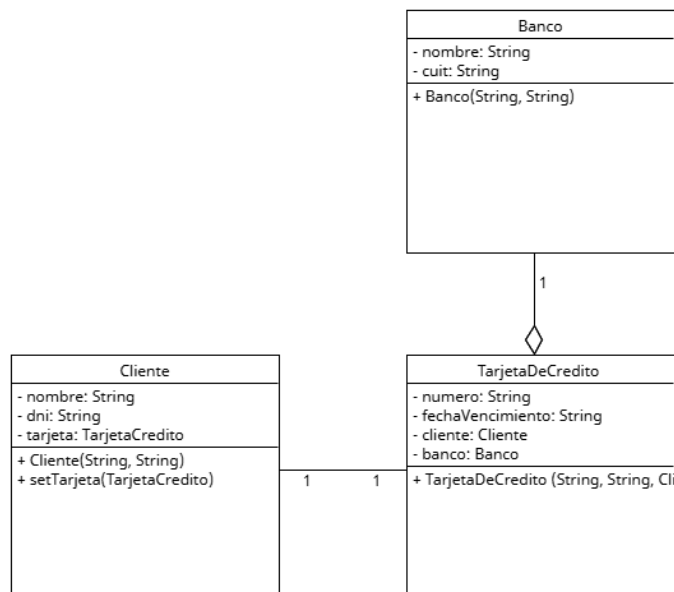
```
public Libro(String titulo, String isbn, Autor autor, Editorial editorial) {
    this.titulo = titulo;
    this.isbn = isbn;
    this.autor = autor;
    this.editorial = editorial;
}
}
```

4. TarjetaDeCrédito - Cliente – Banco

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

Clases y atributos:

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



```
class Banco {
    private String nombre;
    private String cuit;

    public Banco(String nombre, String cuit) {
        this.nombre = nombre;
        this.cuit = cuit;
    }
}
```

```
class Cliente {
    private String nombre;
    private String dni;
    private TarjetaDeCredito tarjeta;

    public Cliente(String nombre, String dni) {
        this.nombre = nombre;
        this.dni = dni;
    }

    public void setTarjeta(TarjetaDeCredito tarjeta) {
        this.tarjeta = tarjeta;
    }
}

class TarjetaDeCredito {
    private String numero;
    private String fechaVencimiento;
    private Cliente cliente;
    private Banco banco;

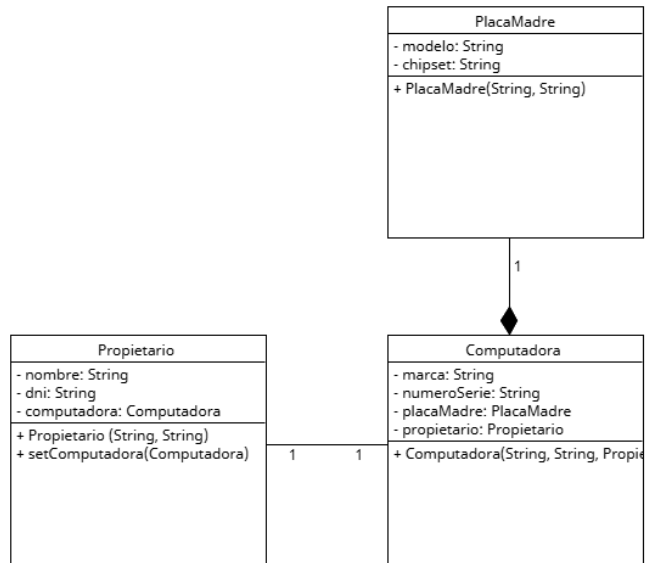
    public TarjetaDeCredito(String numero, String fechaVencimiento, Cliente cliente, Banco
banco) {
        this.numero = numero;
        this.fechaVencimiento = fechaVencimiento;
        this.cliente = cliente;
        this.banco = banco;
        cliente.setTarjeta(this);
    }
}
```

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



```
class PlacaMadre {
    private String modelo;
    private String chipset;

    public PlacaMadre(String modelo, String chipset) {
        this.modelo = modelo;
        this.chipset = chipset;
    }
}

class Propietario {
    private String nombre;
    private String dni;
    private Computadora computadora;

    public Propietario(String nombre, String dni) {
        this.nombre = nombre;
        this.dni = dni;
    }

    public void setComputadora(Computadora computadora) {
        this.computadora = computadora;
    }
}
```

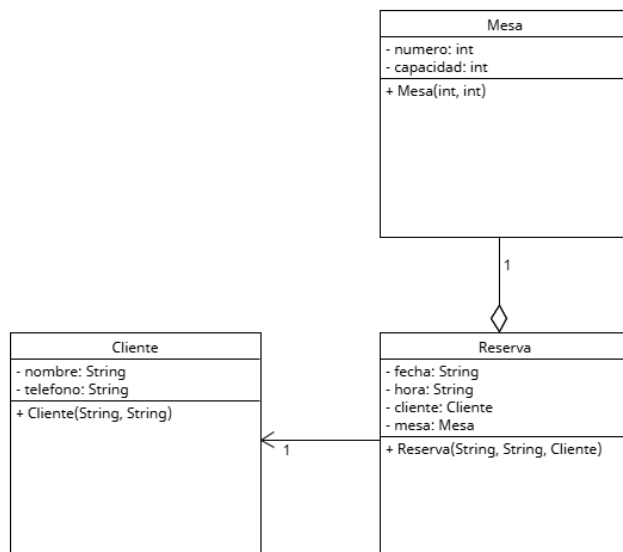


```
class Computadora {  
    private String marca;  
    private String numeroSerie;  
    private PlacaMadre placaMadre;  
    private Propietario propietario;  
  
    public Computadora(String marca, String numeroSerie, Propietario propietario) {  
        this.marca = marca;  
        this.numeroSerie = numeroSerie;  
        this.placaMadre = new PlacaMadre("MSI Z790", "Intel Z790"); // composición  
        this.propietario = propietario;  
        propietario.setComputadora(this);  
    }  
}
```

6. Reserva - Cliente – Mesa
- a. Asociación unidireccional: **Reserva** → **Cliente**
 - b. Agregación: **Reserva** → **Mesa**

Clases y atributos:

- i. Reserva: fecha, hora
- ii. Cliente: nombre, teléfono
- iii. Mesa: numero, capacidad



```
class Cliente {  
    private String nombre;  
    private String telefono;
```

```
public Cliente(String nombre, String telefono) {
    this.nombre = nombre;
    this.telefono = telefono;
}

class Mesa {
    private int numero;
    private int capacidad;

    public Mesa(int numero, int capacidad) {
        this.numero = numero;
        this.capacidad = capacidad;
    }
}

class Reserva {
    private String fecha;
    private String hora;
    private Cliente cliente;
    private Mesa mesa;

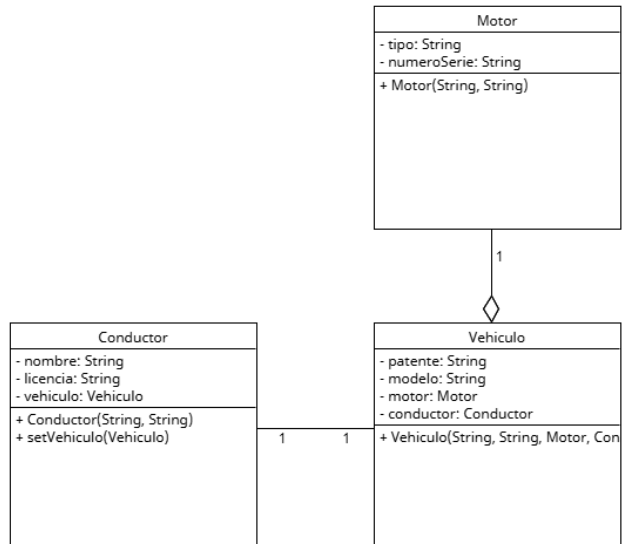
    public Reserva(String fecha, String hora, Cliente cliente, Mesa mesa) {
        this.fecha = fecha;
        this.hora = hora;
        this.cliente = cliente;
        this.mesa = mesa;
    }
}
```

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



```

class Motor {
    private String tipo;
    private String numeroSerie;

    public Motor(String tipo, String numeroSerie) {
        this.tipo = tipo;
        this.numeroSerie = numeroSerie;
    }
}

class Conductor {
    private String nombre;
    private String licencia;
    private Vehiculo vehiculo;

    public Conductor(String nombre, String licencia) {
        this.nombre = nombre;
        this.licencia = licencia;
    }

    public void setVehiculo(Vehiculo vehiculo) {
        this.vehiculo = vehiculo;
    }
}

class Vehiculo {

```

```
private String patente;  
private String modelo;  
private Motor motor;  
private Conductor conductor;  
  
public Vehiculo(String patente, String modelo, Motor motor, Conductor conductor) {  
    this.patente = patente;  
    this.modelo = modelo;  
    this.motor = motor;  
    this.conductor = conductor;  
    conductor.setVehiculo(this);  
}  
}
```

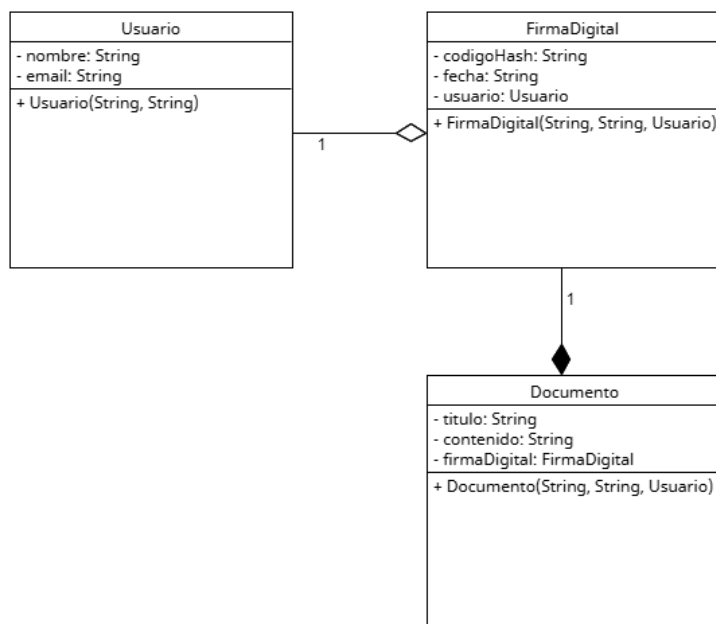
8. Documento - FirmaDigital – Usuario

a. Composición: **Documento** → **FirmaDigital**

b. Agregación: **FirmaDigital** → **Usuario**

Clases y atributos:

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



```
class Usuario {  
    private String nombre;  
    private String email;
```

```
public Usuario(String nombre, String email) {
    this.nombre = nombre;
    this.email = email;
}

class FirmaDigital {
    private String codigoHash;
    private String fecha;
    private Usuario usuario;

    public FirmaDigital(String codigoHash, String fecha, Usuario usuario) {
        this.codigoHash = codigoHash;
        this.fecha = fecha;
        this.usuario = usuario;
    }
}

class Documento {
    private String titulo;
    private String contenido;
    private FirmaDigital firmaDigital;

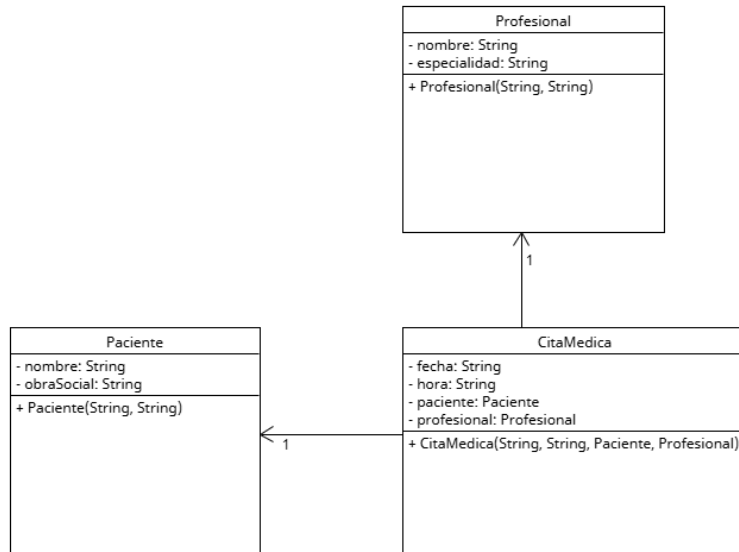
    public Documento(String titulo, String contenido, Usuario usuario) {
        this.titulo = titulo;
        this.contenido = contenido;
        this.firmaDigital = new FirmaDigital("ABC123", "2025-10-22", usuario);
    }
}
```

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



```
class Paciente {
    private String nombre;
    private String obraSocial;

    public Paciente(String nombre, String obraSocial) {
        this.nombre = nombre;
        this.obraSocial = obraSocial;
    }
}

class Profesional {
    private String nombre;
    private String especialidad;

    public Profesional(String nombre, String especialidad) {
        this.nombre = nombre;
        this.especialidad = especialidad;
    }
}

class CitaMedica {
    private String fecha;
    private String hora;
    private Paciente paciente;
    private Profesional profesional;
}
```

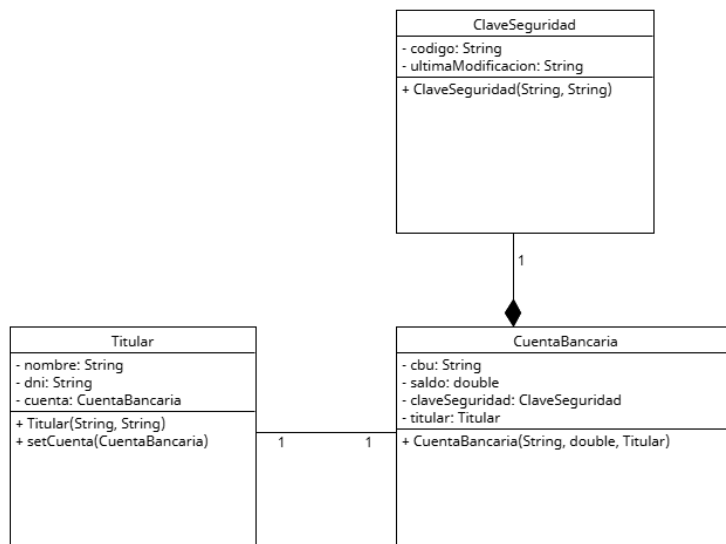
```
public CitaMedica(String fecha, String hora, Paciente paciente, Profesional profesional) {  
    this.fecha = fecha;  
    this.hora = hora;  
    this.paciente = paciente;  
    this.profesional = profesional;  
}  
}
```

10. CuentaBancaria - ClaveSeguridad – Titular

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

Clases y atributos:

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



```
class ClaveSeguridad {  
    private String codigo;  
    private String ultimaModificacion;  
  
    public ClaveSeguridad(String codigo, String ultimaModificacion) {  
        this.codigo = codigo;  
        this.ultimaModificacion = ultimaModificacion;  
    }  
}  
  
class Titular {
```

```
private String nombre;
private String dni;
private CuentaBancaria cuenta;

public Titular(String nombre, String dni) {
    this.nombre = nombre;
    this.dni = dni;
}

public void setCuenta(CuentaBancaria cuenta) {
    this.cuenta = cuenta;
}
}

class CuentaBancaria {
    private String cbu;
    private double saldo;
    private ClaveSeguridad claveSeguridad;
    private Titular titular;

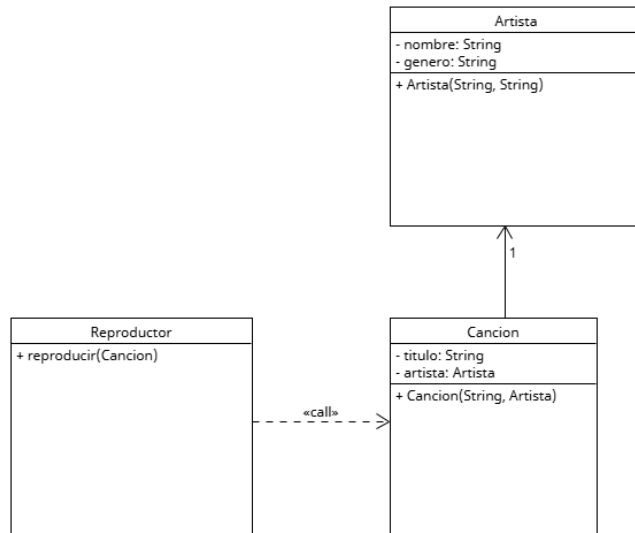
    public CuentaBancaria(String cbu, double saldo, Titular titular) {
        this.cbu = cbu;
        this.saldo = saldo;
        this.claveSeguridad = new ClaveSeguridad("XYZ999", "2025-10-22");
        this.titular = titular;
        titular.setCuenta(this);
    }
}
```

11. Reproductor - Canción – Artista

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

Clases y atributos:

- i. Canción: titulo.
- ii. Artista: nombre, genero.
- iii. Reproductor->método: void reproducir(Cancion cancion)



```

class Artista {
    private String nombre;
    private String genero;

    public Artista(String nombre, String genero) {
        this.nombre = nombre;
        this.genero = genero;
    }
}

class Cancion {
    private String titulo;
    private Artista artista;

    public Cancion(String titulo, Artista artista) {
        this.titulo = titulo;
        this.artista = artista;
    }
}

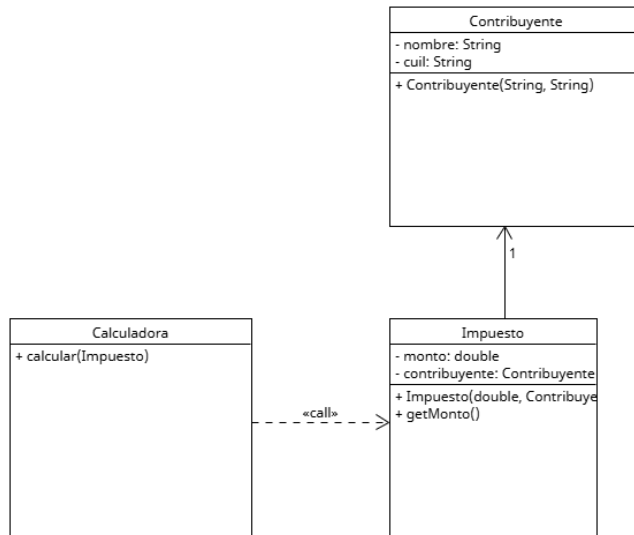
class Reproductor {
    public void reproducir(Cancion cancion) {
        System.out.println("Reproduciendo: " + cancion);
    }
}
    
```

12. Impuesto - Contribuyente – Calculadora

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

Clases y atributos:

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



```
class Contribuyente {
    private String nombre;
    private String cuil;

    public Contribuyente(String nombre, String cuil) {
        this.nombre = nombre;
        this.cuil = cuil;
    }
}

class Impuesto {
    private double monto;
    private Contribuyente contribuyente;

    public Impuesto(double monto, Contribuyente contribuyente) {
        this.monto = monto;
        this.contribuyente = contribuyente;
    }

    public double getMonto() {
```

```
        return monto;
    }
}

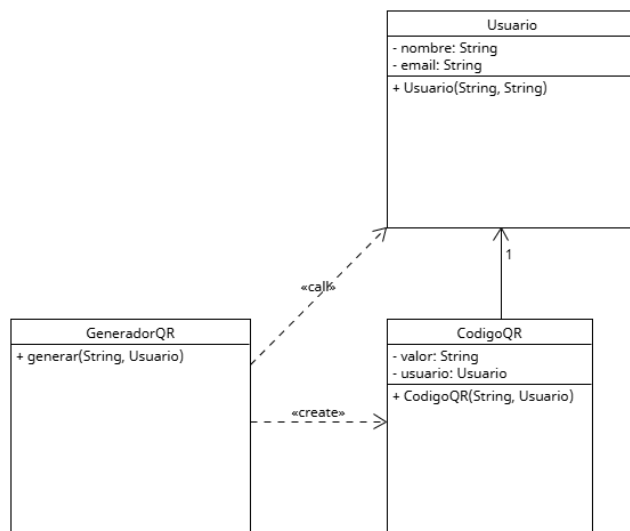
class Calculadora {
    public void calcular(Impuesto impuesto) {
        System.out.println("Calculando impuesto: $" + impuesto.getMonto());
    }
}
```

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)



```
class CódigoQR {
    private String valor;
    private Usuario usuario;

    public CódigoQR(String valor, Usuario usuario) {
        this.valor = valor;
        this.usuario = usuario;
    }
}
```

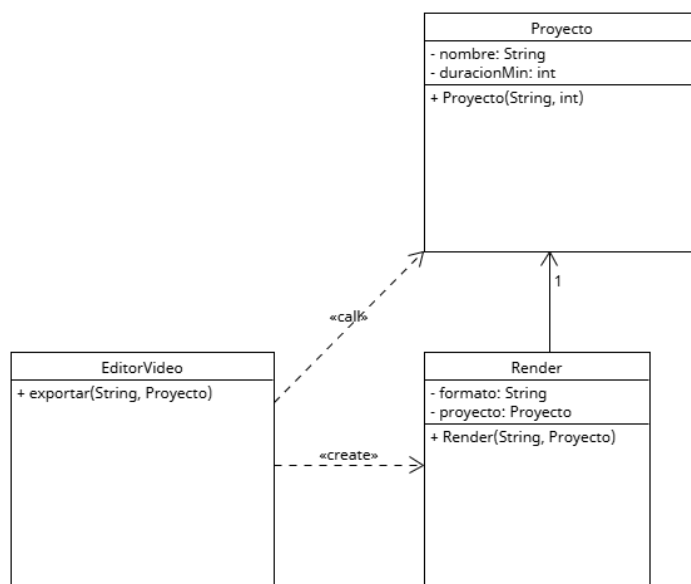
```
class Usuario {  
    private String nombre;  
    private String email;  
  
    public Usuario(String nombre, String email) {  
        this.nombre = nombre;  
        this.email = email;  
    }  
}  
  
class GeneradorQR {  
    public void generar(String valor, Usuario usuario) {  
        CodigoQR codigo = new CodigoQR(valor, usuario);  
        System.out.println("QR generado para " + usuario + " con valor " + valor);  
    }  
}
```

14. EditorVideo - Proyecto – Render

- Asociación unidireccional: **Render → Proyecto**
- Dependencia de creación: **EditorVideo.exportar(String, Proyecto)**

Clases y atributos:

- Render: formato.
- Proyecto: nombre, duracionMin.
- EditorVideo->método: void exportar(String formato, Proyecto proyecto)



```
class Proyecto {
    private String nombre;
    private int duracionMin;

    public Proyecto(String nombre, int duracionMin) {
        this.nombre = nombre;
        this.duracionMin = duracionMin;
    }
}

class Render {
    private String formato;
    private Proyecto proyecto;

    public Render(String formato, Proyecto proyecto) {
        this.formato = formato;
        this.proyecto = proyecto;
    }
}

class EditorVideo {
    public void exportar(String formato, Proyecto proyecto) {
        Render render = new Render(formato, proyecto);
        System.out.println("Render exportado en formato: " + formato);
    }
}
```