

<https://github.com/Coman-Che/Programacion-II/tree/main/Unidad%205>

## PROGRAMACIÓN II

### Trabajo Práctico 5: Relaciones UML 1 a 1

#### 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.

#### MARCO TEÓRICO

Concepto	Aplicación en el proyecto
Asociación	Relación entre clases con referencia mutua o directa, puede ser uni o bidireccional
Agregación	Relación de "tiene un" donde los objetos pueden vivir independientemente
Composición	Relación fuerte de contención, el ciclo de vida del objeto contenido depende del otro
Dependencia de uso	Una clase usa otra como parámetro en un método, sin almacenarla como atributo
Dependencia de creación	Una clase crea otra en tiempo de ejecución, sin mantenerla como atributo

Asociación	Relación entre clases con referencia mutua o directa, puede ser uni o bidireccional
Agregación	Relación de "tiene un" donde los objetos pueden vivir independientemente

## Caso Práctico

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

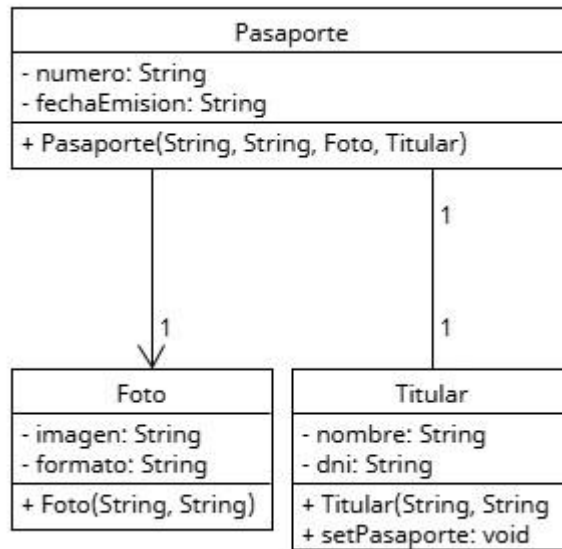
- Diagrama UML
- Tipo de relación (asociación, agregación, composición, dependencia)
- Dirección (unidireccional o bidireccional)
- Implementación de las clases con atributos y relaciones definidas

### 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



```
package Ejercicio_01;
```

```
public class Foto {
```

```
    private String imagen;
```

```
    private String formato;
```

```
    public Foto(String imagen, String formato) {
```

```
        this.imagen = imagen;        this.formato =
```

```
        formato;
```

```
    }
```

```
    @Override    public String toString() {        return "Foto{" +
```

```
        "imagen=" + imagen + ", formato=" + formato + '}';
```

```
    }
```

```
} package  
  
Ejercicio_01;  
  
public class Pasaporte {  
  
    private String numero;    private String  
    fechaEmision;    private Foto foto;    //  
    Composición    private Titular titular; // Asociación  
    bidireccional    public Pasaporte(String numero,  
    String fechaEmision, Foto foto, Titular titular) {  
        this.numero = numero;        this.fechaEmision =  
        fechaEmision;        this.foto = foto;        this.titular  
        = titular;        titular.setPasaporte(this); //  
        vinculación bidireccional  
    }  
  
    @Override    public  
    String toString() {  
        return "Pasaporte{" + "numero=" + numero + ", fechaEmision=" +  
        fechaEmision + ", foto=" + foto + ", titular=" + titular + '}';  
    }  
}
```

```
} package
```

```
Ejercicio_01;
```

```
public class Titular {
```

```
    private String nombre;    private String dni;    private
```

```
Pasaporte pasaporte; // Asociación bidireccional
```

```
    public Titular(String nombre, String dni) {
```

```
        this.nombre = nombre;
```

```
        this.dni = dni;
```

```
    }
```

```
    public void setPasaporte(Pasaporte pasaporte) {
```

```
        this.pasaporte = pasaporte;
```

```
    }
```

```
    @Override    public String toString() {        return "Titular{" +
```

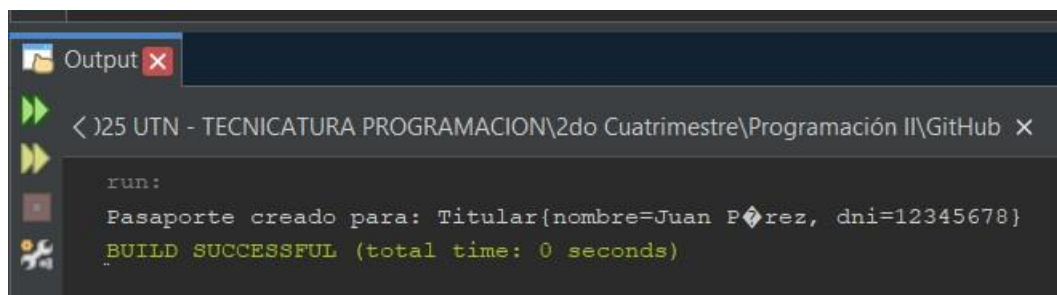
```
        "nombre=" + nombre + ", dni=" + dni + '}';
```

```
    }
```

```
} package
```

```
Ejercicio_01;
```

```
public class Main_Ejercicio_01 {  
  
    public static void main(String[] args) {  
  
        Foto foto = new Foto("foto.png", "PNG");  
  
        Titular titular = new Titular("Juan Pérez", "12345678");  
  
        Pasaporte pasaporte = new Pasaporte("A12345", "2025-01-01", foto,  
titular);  
  
        System.out.println("Pasaporte creado para: " + titular);  
  
    }  
  
}
```

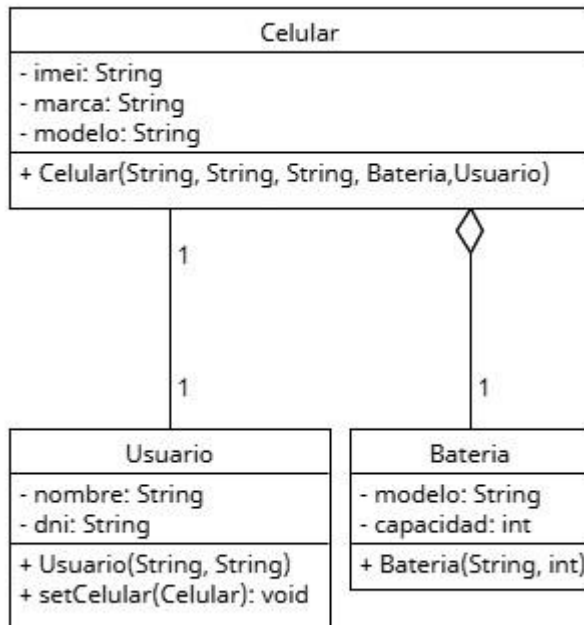


## 2. Celular - Batería - Usuario

- Agregación: **Celular** → **Batería**
- Asociación bidireccional: **Celular** ↔ **Usuario**

Clases y atributos:

- Celular: imei, marca, modelo
- Batería: modelo, capacidad
- Usuario: nombre, dni



```
package Ejercicio_02;
```

```
public class Bateria {
```

```
    private String modelo;
```

```
    private int capacidad; // en mAh
```

```
    public Bateria(String modelo, int capacidad)
```

```
    {        this.modelo = modelo;
```

```
    this.capacidad = capacidad;
```

```
    }
```

```
    @Override    public
```

```
    String toString() {
```

```
return "Bateria{" +  
    "modelo=" + modelo + "  
    capacidad=" + capacidad +  
    '}'  
};  
  
}
```

```
} package
```

```
Ejercicio_02;
```

```
public class Celular {
```

```
    private String imei;    private String marca;
```

```
    private String modelo;    private Bateria bateria;    //
```

```
    Agregación    private Usuario usuario;    // Asociación
```

```
    bidireccional
```

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

```
    @Override    public  
    String toString() {
```



```
        return "Celular{" + "imei=" + imei + ", marca=" + marca + ", modelo=" + modelo  
+ ", bateria=" + bateria + ", usuario=" + usuario + '}';
```

```
    }
```

```
} package
```

```
Ejercicio_02;
```

```
public class Usuario {
```

```
    private String nombre;    private String dni;
```

```
    private Celular celular; // Asociación bidireccional
```

```
    public Usuario(String nombre, String dni)
```

```
    {        this.nombre = nombre;        this.dni =
```

```
    dni;
```

```
    }
```

```
    public void setCelular(Celular celular) {
```

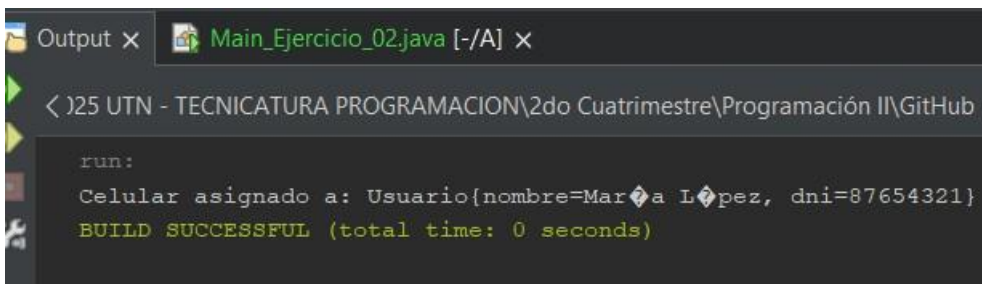
```
    this.celular = celular;
```

```
    }
```

```
    @Override    public String toString() {        return "Usuario{" +
```

```
    "nombre=" + nombre + ", dni=" + dni + '}';
```

```
}  
  
} package  
Ejercicio_02;  
  
public class Main_Ejercicio_02 {  
  
    public static void main(String[] args) {  
  
        Bateria bateria = new Bateria("Samsung-45W", 5000);  
        Usuario usuario = new Usuario("María López", "87654321");  
        Celular celular = new Celular("IMEI123", "Samsung", "S21", bateria, usuario);  
  
        System.out.println("Celular asignado a: " + usuario);  
  
    }  
  
}
```



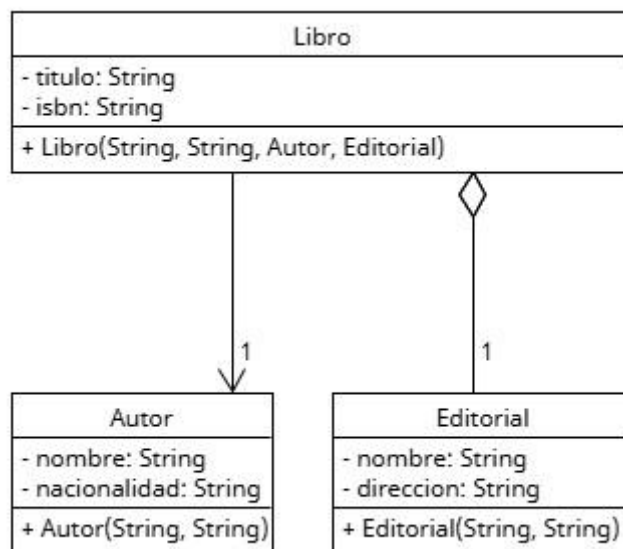
```
Output x Main_Ejercicio_02.java [-/A] x  
< >25 UTN - TECNICATURA PROGRAMACION\2do Cuatrimestre\Programación II\GitHub >  
run:  
Celular asignado a: Usuario{nombre=María López, dni=87654321}  
BUILD SUCCESSFUL (total time: 0 seconds)
```

### 3. Libro - Autor - Editorial

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

Clases y atributos:

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



```
package Ejercicio_03;
```

```
public class Autor {
```

```
    private String nombre;
```

```
    private String nacionalidad;
```

```
    public Autor(String nombre,
```

```
                String nacionalidad) {
```

```
        this.nombre = nombre;
```

```
        this.nacionalidad =
```

```
        nacionalidad;
```

```
}  
  
    @Override    public  
    String toString() {  
        return "Autor{" + "nombre=" + nombre + ", nacionalidad=" + nacionalidad  
+ '}';  
    }  
  
}  
  
package Ejercicio_03;  
  
public class Editorial {  
  
    private String nombre;  
    private String direccion;  
  
    public Editorial(String nombre, String direccion) {  
        this.nombre = nombre;  
        this.direccion = direccion;  
    }  
}
```

```
@Override public  
String toString() {  
    return "Editorial{" + "nombre=" + nombre + ", direccion=" + direccion +  
    '}';  
}
```

```
} package
```

```
Ejercicio_03;
```

```
public class Libro {
```

```
    private String titulo;    private String isbn;    private  
    Autor autor;    // Asociación unidireccional    private  
    Editorial editorial; // Agregación
```

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

```
@Override public  
String toString() {
```

```
        return "Libro{" + "titulo=" + titulo + ", isbn=" + isbn + ", autor=" + autor +  
        ", editorial=" + editorial + '}';
```

```
    }
```

```
} package
```

```
Ejercicio_03;
```

```
public class Main_Ejercicio_03 {
```

```
    public static void main(String[] args) {
```

```
        Autor autor = new Autor("Gabriel García Márquez", "Colombiano");
```

```
        Editorial editorial = new Editorial("Sudamericana", "Buenos Aires");
```

```
        Libro libro = new Libro("Cien Años de Soledad", "978-3-16-148410-0",  
        autor, editorial);
```

```
        System.out.println("Libro creado: " + libro);
```

```
    }
```

```
}
```

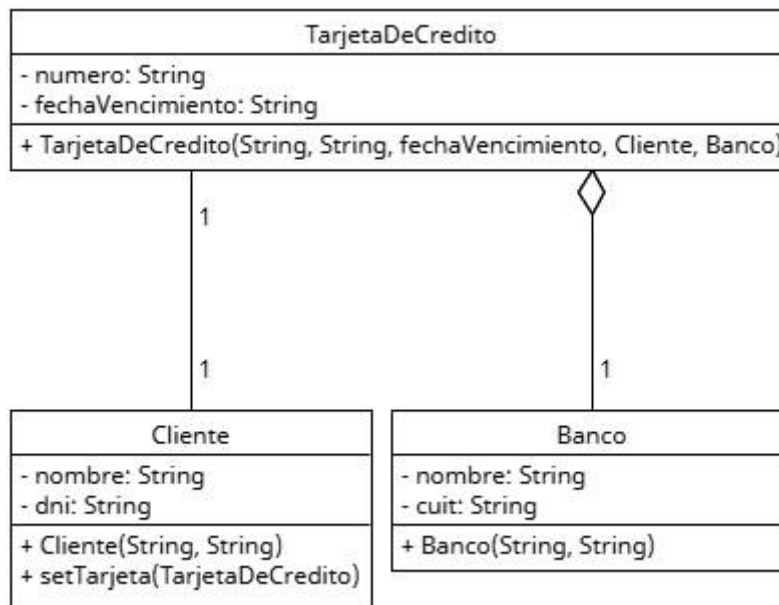
```
Output x Main_Ejercicio_02.java [-/A] x
< 125 UTN - TECNICATURA PROGRAMACION\2do Cuatrimestre\Programación II\GitHub x Jofre, Alan - Programación 2 - TP5 (run) x
run:
Libro creado: Libro{titulo=Cien Años de Soledad, isbn=978-3-16-148410-0, autor=Autor{nombre=Gabriel Gar
BUILD SUCCESSFUL (total time: 0 seconds)
```

#### 4. TarjetaDeCrédito - Cliente - Banco

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

Clases y atributos:

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



```
package Ejercicio_04;
```

```
public class Banco {
```

```
    private String nombre;
```

```
    private String cuit;
```

```
        public Banco(String nombre, String cuit) {  
            this.nombre = nombre;    this.cuit = cuit;  
        }  
  
        @Override    public String toString() {    return "Banco{" +  
            "nombre=" + nombre + ", cuit=" + cuit + '}';  
        }  
  
    } package  
Ejercicio_04;  
  
public class Cliente {  
  
    private String nombre;    private String dni;    private  
    TarjetaDeCredito tarjeta; // Asociación bidireccional    public  
    Cliente(String nombre, String dni) {    this.nombre =  
    nombre;    this.dni = dni;  
    }  
  
    public void setTarjeta(TarjetaDeCredito tarjeta) {  
        this.tarjeta = tarjeta;  
    }  
}
```



```
@Override public String toString() { return "Cliente{" +  
"nombre=" + nombre + ", dni=" + dni + '}';  
}  
  
} package  
Ejercicio_04;  
  
public class TarjetaDeCredito {  
  
    private String numero;    private String  
    fechaVencimiento;    private Cliente cliente; //  
    Asociación bidireccional    private Banco banco;  
    // Agregación  
  
    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); // Vinculación bidireccional  
    }  
  
    @Override public  
    String toString() {
```

```
        return "TarjetaDeCredito{" + "numero=" + numero + ",  
        fechaVencimiento=" + fechaVencimiento + '}';  
  
    }  
  
} package  
  
Ejercicio_04;  
  
public class Main_Ejercicio_04 {  
  
    public static void main(String[] args) {  
  
        Cliente cliente = new Cliente("Ana Torres", "44556677");  
  
        Banco banco = new Banco("Banco Nación", "30-12345678-9");  
  
        TarjetaDeCredito tarjeta = new TarjetaDeCredito("1234-5678-9012-3456",  
        "12/29", cliente, banco);  
  
        System.out.println("Tarjeta creada para cliente: " + cliente);  
  
    }  
  
}
```

```

Output x Main_Ejercicio_02.java [-/A] x
< 25 UTN - TECNICATURA PROGRAMACION\2do Cuatrimestre\Programación II\GitHub x Jofr
run:
Tarjeta creada para cliente: Cliente{nombre=Ana Torres, dni=44556677}
BUILD SUCCESSFUL (total time: 0 seconds)

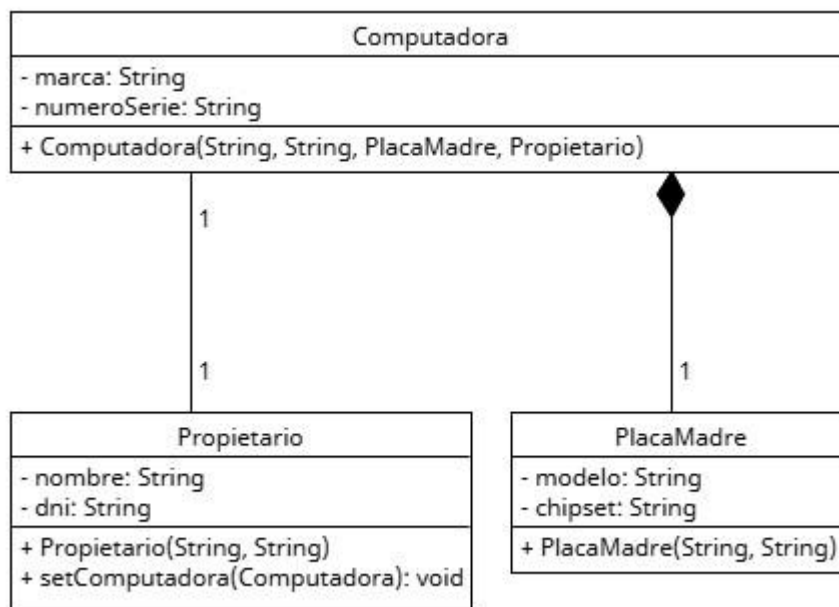
```

##### 5. Computadora - PlacaMadre - Propietario

- Composición: **Computadora** → **PlacaMadre**
- Asociación bidireccional: **Computadora** ↔ **Propietario**

Clases y atributos:

- Computadora: marca, numeroSerie
- PlacaMadre: modelo, chipset
- Propietario: nombre, dni



package Ejercicio\_05;

public class Computadora {

```
        private String marca;    private String numeroSerie;

        private PlacaMadre placaMadre; // Composición    private
        Propietario propietario; // Asociación bidireccional


        public Computadora(String marca, String numeroSerie, PlacaMadre
        placaMadre, Propietario propietario) {    this.marca = marca;
        this.numeroSerie = numeroSerie;    this.placaMadre = placaMadre;
        this.propietario = propietario;

        propietario.setComputadora(this); // Vinculación bidireccional
    }


    @Override    public
    String toString() {

        return "Computadora{" + "marca=" + marca + ", numeroSerie=" +
        numeroSerie + ", placaMadre=" + placaMadre + ", propietario=" + propietario
        + '}';

    }


} package

Ejercicio_05;


public class PlacaMadre {
```

```
        private String modelo;

        private String chipset;

        public PlacaMadre(String modelo, String chipset) {

            this.modelo = modelo;        this.chipset = chipset;

        }

        @Override

        public String toString() {        return "PlacaMadre{" + "modelo=" + modelo

+ ", chipset=" + chipset + '}';

        }

    } package

Ejercicio_05;

public class Propietario {

    private String nombre;    private String dni;    private

Computadora computadora; // Asociación bidireccional

    public Propietario(String nombre, String dni) {

        this.nombre = nombre;        this.dni = dni;

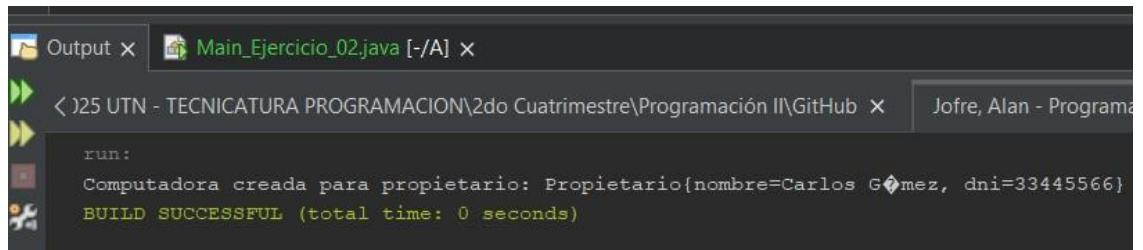
    }

}
```

```
        public void setComputadora(Computadora computadora) {  
  
            this.computadora = computadora;  
  
        }  
  
        @Override  
  
        public String toString() {        return "Propietario{" + "nombre=" +  
nombre + ", dni=" + dni + '}';  
  
        }  
  
    } package  
Ejercicio_05;  
  
public class Main_Ejercicio_05 {  
  
    public static void main(String[] args) {  
  
        PlacaMadre placa = new PlacaMadre("MSI Z490", "Intel Z490");  
  
        Propietario propietario = new Propietario("Carlos Gómez", "33445566");  
  
        Computadora pc = new Computadora("HP", "SN123456", placa,  
propietario);  
  
        System.out.println("Computadora creada para propietario: " + propietario);  
    }  
}
```

```
}
```

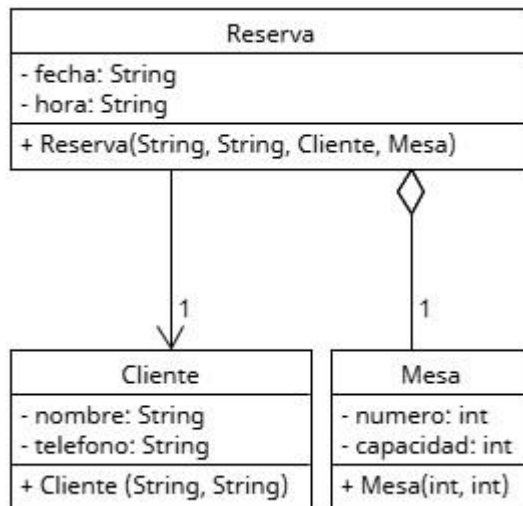
```
}
```



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

Clases y atributos:

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



```
package Ejercicio_06;
```

```
public class Cliente {
```

```
        private String nombre;

        private String telefono; public

        Cliente(String nombre, String

        telefono) {        this.nombre =

        nombre;        this.telefono =

        telefono;

        }

        @Override    public String toString() {        return "Cliente{" +

        "nombre=" + nombre + ", telefono=" + telefono + '}';

        }

    } package

    Ejercicio_06;

    public class Mesa {

        private int numero;

        private int capacidad;
```



```
        public Mesa(int numero, int capacidad) {  
            this.numero = numero;    this.capacidad  
            = capacidad;  
        }  
  
        @Override  
        public String toString() {    return "Mesa{" + "numero=" + numero + ",  
            capacidad=" + capacidad + '}';  
        }  
  
    } package  
Ejercicio_06;  
  
public class Reserva {  
  
    private String fecha;    private String hora;  
    private Cliente cliente; // Asociación unidireccional  
    private Mesa mesa;    // Agregación  
  
    public Reserva(String fecha, String hora, Cliente cliente, Mesa mesa) {  
        this.fecha = fecha;    this.hora = hora;    this.cliente = cliente;  
        this.mesa = mesa;  
    }  
}
```

@Override

```
public String toString() {  
  
    return "Reserva{" + "fecha=" + fecha + ", hora=" + hora + ", cliente=" +  
    cliente + ", mesa=" + mesa + '}';  
  
}
```

```
} package
```

Ejercicio\_06;

```
public class Main_Ejercicio_06 {
```

```
public static void main(String[] args) {
```

```
    Cliente cliente = new Cliente("Laura Méndez", "11223344");
```

```
    Mesa mesa = new Mesa(5, 4);
```

```
    Reserva reserva = new Reserva("2025-10-05", "21:00", cliente, mesa);
```

```
    System.out.println("Reserva creada para cliente: " + cliente);
```

```
}
```

```
}
```

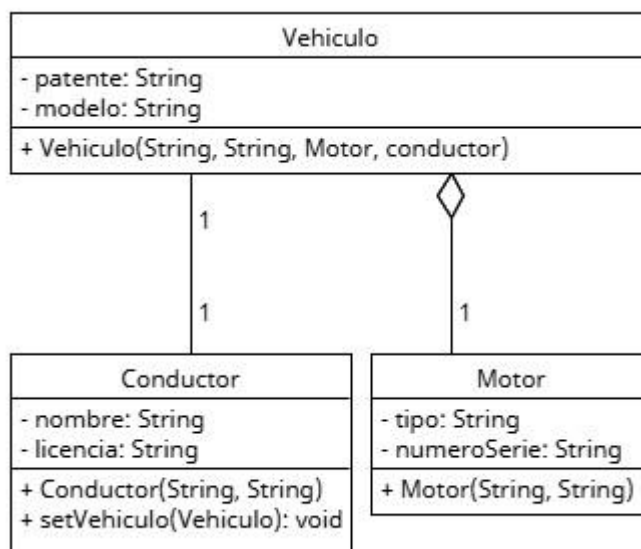
```
Output x Main_Ejercicio_02.java [-/A] x
< 25 UTN - TECNICATURA PROGRAMACION\2do Cuatrimestre\Programación II\GitHub x Jofre, Alan -
run:
Reserva creada para cliente: Cliente(nombre=Laura Mñdez, telefono=11223344)
BUILD SUCCESSFUL (total time: 0 seconds)
```

## 7. Vehículo - Motor - Conductor

- Agregación: **Vehículo** → **Motor**
- Asociación bidireccional: **Vehículo** ↔ **Conductor**

Clases y atributos:

- Vehículo: patente, modelo
- Motor: tipo, numeroSerie
- Conductor: nombre, licencia



```
package Ejercicio_07;
```

```
public class Conductor {
```

```
    private String nombre;    private String licencia;
```

```
    private Vehiculo vehiculo; // Asociación bidireccional
```

```
public Conductor(String nombre, String licencia) {  
  
    this.nombre = nombre;    this.licencia = licencia;  
  
}  
  
public void setVehiculo(Vehiculo vehiculo) {  
  
    this.vehiculo = vehiculo;  
  
}  
  
@Override    public  
String toString() {  
  
    return "Conductor{" + "nombre=" + nombre + ", licencia=" + licencia +  
    '}';  
  
}  
  
} package  
Ejercicio_07;  
  
public class Motor {  
  
    private String tipo;  
  
    private String numeroSerie;  
  
    public Motor(String tipo,  
  
    String numeroSerie) {
```

```
        this.tipo = tipo;

        this.numeroSerie = numeroSerie;

    }

    @Override public String toString() {        return "Motor{" + "tipo=" +
        tipo + ", numeroSerie=" + numeroSerie + '}';

    }

} package

Ejercicio_07;

public class Vehiculo {

    private String patente;    private String modelo;

    private Motor motor;    // Agregación    private
    Conductor conductor; // Asociación bidireccional

    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);

}

@Override public
String toString() {

    return "Vehiculo{" + "patente=" + patente + ", modelo=" + modelo + ",
motor=" + motor + ", conductor=" + conductor + '}';

}

} package

Ejercicio_07;

public class Main_Ejercicio_07 {

    public static void main(String[] args) {

        Motor motor = new Motor("V8", "MTR1234");

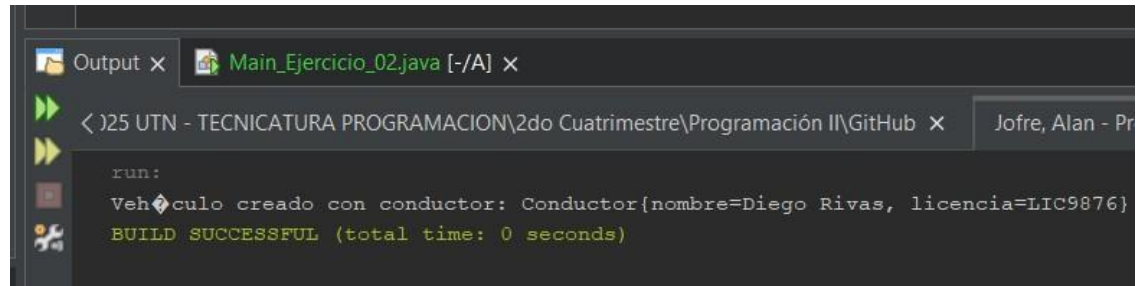
        Conductor conductor = new Conductor("Diego Rivas", "LIC9876");

        Vehiculo vehiculo = new Vehiculo("ABC123", "Toyota Hilux", motor,
conductor);

        System.out.println("Vehículo creado con conductor: " + conductor);
```

```
}
```

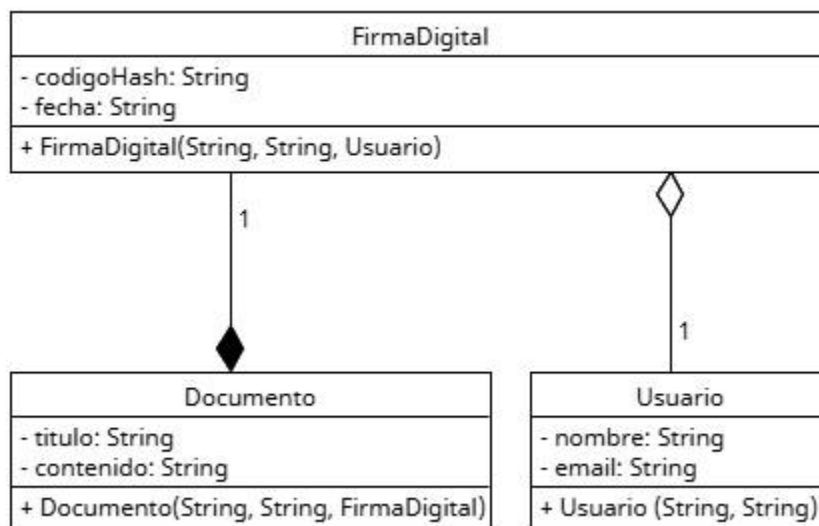
```
}
```



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



```
package Ejercicio_08;
```

```
public class Documento {
```

```
private String titulo; private String  
contenido; private FirmaDigital firma; //
```

Composición

```
public Documento(String titulo, String contenido, FirmaDigital firma) {  
    this.titulo = titulo;  
    this.contenido = contenido;  
    this.firma = firma;  
}
```

```
@Override public String toString() { return "Documento{" +  
"titulo=" + titulo + ", firma=" + firma + "}'";  
}
```

```
} package
```

Ejercicio\_08;

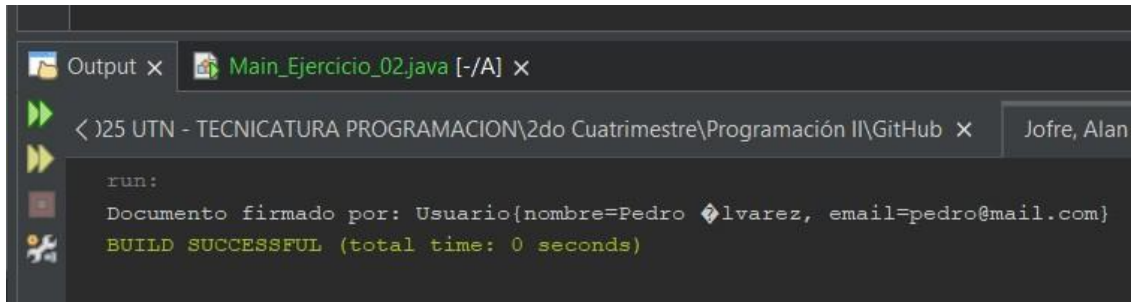
```
public class FirmaDigital {
```

```
private String codigoHash; private  
String fecha; private Usuario usuario;  
// Agregación
```



```
        public FirmaDigital(String codigoHash, String fecha, Usuario usuario) {  
  
            this.codigoHash = codigoHash;        this.fecha = fecha;        this.usuario =  
            usuario;  
  
        }  
  
        @Override    public  
        String toString() {  
  
            return "FirmaDigital{" + "codigoHash=" + codigoHash + ", fecha=" +  
            fecha + ", usuario=" + usuario + '}';  
  
        }  
  
    } package  
    Ejercicio_08;  
  
    public class Usuario {  
  
        private String nombre;  
  
        private String email;  
  
        public Usuario(String nombre, String email) {  
  
            this.nombre = nombre;        this.email = email;  
  
        }  
    }
```

```
@Override public String toString() {    return "Usuario{" +  
"nombre=" + nombre + ", email=" + email + '}';  
}  
  
} package  
Ejercicio_08;  
  
public class Main_Ejercicio_08 {  
  
    public static void main(String[] args) {  
  
        Usuario usuario = new Usuario("Pedro Álvarez", "pedro@mail.com");  
  
        FirmaDigital firma = new FirmaDigital("HASH123", "2025-10-01",  
usuario);  
  
        Documento doc = new Documento("Contrato", "Contenido del contrato...",  
firma);  
  
        System.out.println("Documento firmado por: " + usuario);  
  
    }  
  
}
```

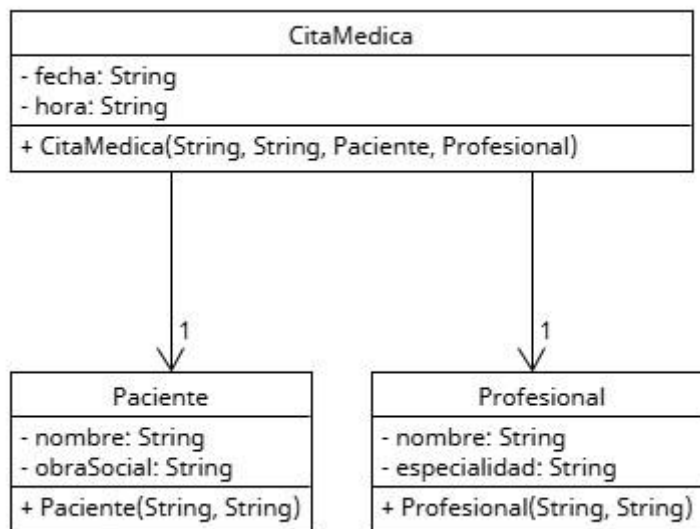


## 9. CitaMédica - Paciente - Profesional

- Asociación unidireccional: **CitaMédica** → **Paciente**,
- Asociación unidireccional: **CitaMédica** → **Profesional**

Clases y atributos:

- CitaMédica: fecha, hora
- Paciente: nombre, obraSocial
- Profesional: nombre, especialidad



```
package Ejercicio_09;
```

```
public 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;  
    }
```

```
    @Override    public  
String toString() {  
        return "CitaMedica{" + "fecha=" + fecha + ", hora=" + hora + ",  
paciente=" + paciente + ", profesional=" + profesional + '}';  
    }
```

```
}
```

```
package Ejercicio_09;
```

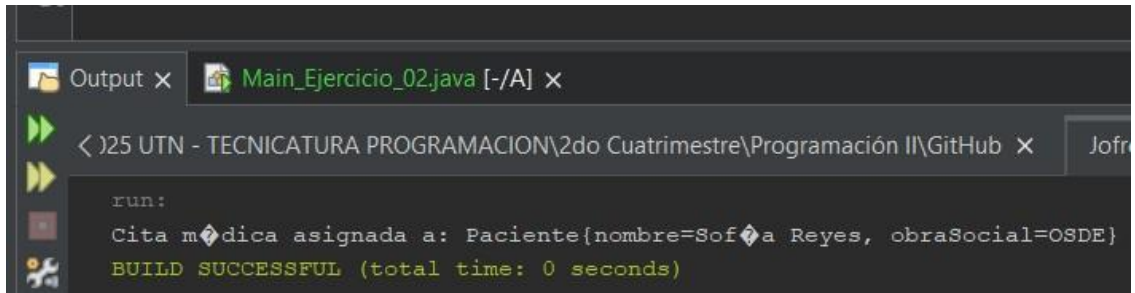
```
public class Paciente {
```

```
    private String nombre;
```

```
    private String obraSocial;
```

```
        public Paciente(String nombre, String obraSocial) {  
  
            this.nombre = nombre;        this.obraSocial =  
  
            obraSocial;  
  
        }  
  
        @Override    public  
  
        String toString() {  
  
            return "Paciente{" + "nombre=" + nombre + ", obraSocial=" + obraSocial  
            + '}';  
  
        }  
  
    } package  
  
    Ejercicio_09;  
  
    public class Profesional {  
  
        private String nombre;  
  
        private String especialidad;  
  
        public Profesional(String nombre, String especialidad) {  
  
            this.nombre = nombre;        this.especialidad =  
  
            especialidad;  
  
        }
```

```
@Override public  
String toString() {  
    return "Profesional{" + "nombre=" + nombre + ", especialidad=" +  
    especialidad + '}';  
}  
  
} package  
Ejercicio_09;  
  
public class Main_Ejercicio_09 {  
  
    public static void main(String[] args) {  
  
        Paciente paciente = new Paciente("Sofía Reyes", "OSDE");  
  
        Profesional medico = new Profesional("Dr. Luis", "Cardiología");  
        cita = new CitaMedica("2025-10-15", "10:00", paciente, medico);  
  
        System.out.println("Cita médica asignada a: " + paciente);  
  
    }  
  
}
```



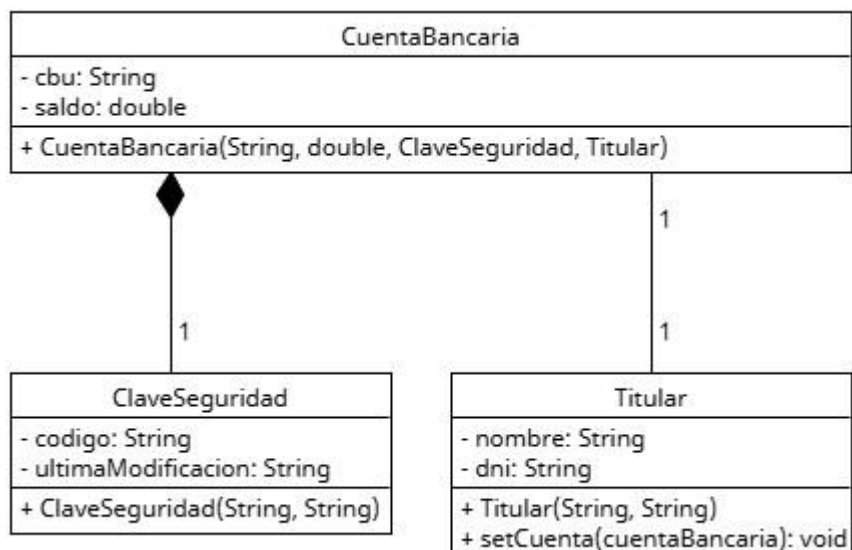
```
run:
Cita médica asignada a: Paciente{nombre=Sofoa Reyes, obraSocial=OSDE}
BUILD SUCCESSFUL (total time: 0 seconds)
```

#### 10. CuentaBancaria - ClaveSeguridad - Titular

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

Clases y atributos:

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



```
package Ejercicio_10;
```

```
public class ClaveSeguridad {
```

```
        private String codigo;    private

String ultimaModificacion;

        public ClaveSeguridad(String codigo, String ultimaModificacion) {

this.codigo = codigo;    this.ultimaModificacion =

ultimaModificacion;

        }

        @Override    public

String toString() {

        return "ClaveSeguridad{" + "codigo=" + codigo + ", ultimaModificacion="

+ ultimaModificacion + '}';

        }

    } package

Ejercicio_10;

public class CuentaBancaria {

        private String cbu;    private double saldo;

private ClaveSeguridad clave; // Composición

private Titular titular;    // Asociación bidireccional
```



```
        public CuentaBancaria(String cbu, double saldo, ClaveSeguridad clave,  
        Titular titular) {            this.cbu = cbu;            this.saldo = saldo;  
        this.clave = clave;            this.titular = titular;            titular.setCuenta(this);  
    }
```

```
        @Override    public  
        String toString() {  
        return "CuentaBancaria{"  
        + "cbu=" + cbu + "  
        saldo=" + saldo + "  
        clave=" + clave + "  
        titular=" + titular + '}';  
    }
```

```
} package
```

```
Ejercicio_10;
```

```
public class Titular {
```

```
        private String nombre;    private String dni;    private
```

```
CuentaBancaria cuenta; // Asociación bidireccional
```

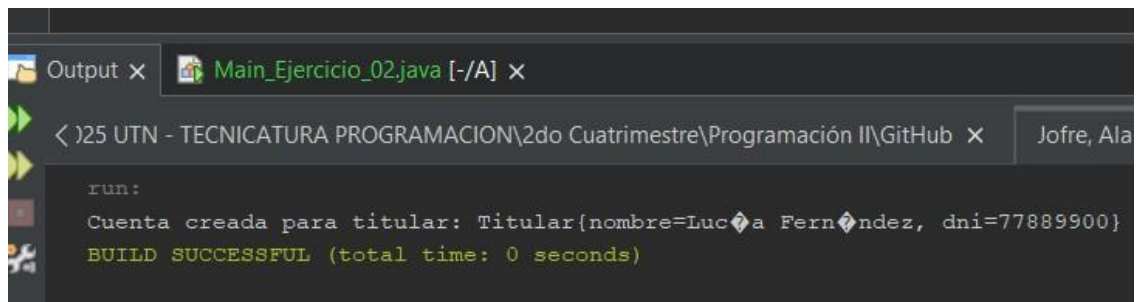
```
        public Titular(String nombre, String dni) {  
  
            this.nombre = nombre;        this.dni = dni;  
  
        }  
  
        public void setCuenta(CuentaBancaria cuenta) {  
  
            this.cuenta = cuenta;  
  
        }  
  
        @Override  
  
        public String toString() {        return "Titular{" + "nombre=" +  
nombre + ", dni=" + dni + '}';  
  
        }  
  
    }  
  
} package  
Ejercicio_10;  
  
public class Main_Ejercicio_10 {  
  
    public static void main(String[] args) {  
  
        ClaveSeguridad clave = new ClaveSeguridad("ABC123", "2025-09-30");  
  
        Titular titular = new Titular("Lucía Fernández", "77889900");  
  
    }  
}
```

```
CuentaBancaria cuenta = new CuentaBancaria("CBU123456789",  
15000.0, clave, titular);
```

```
System.out.println("Cuenta creada para titular: " + titular);
```

```
}
```

```
}
```



```
run:  
Cuenta creada para titular: Titular{nombre=Lucía Fernández, dni=77889900}  
BUILD SUCCESSFUL (total time: 0 seconds)
```

## DEPENDENCIA DE USO

La clase usa otra como **parámetro de un método**, pero **no la guarda como atributo**.

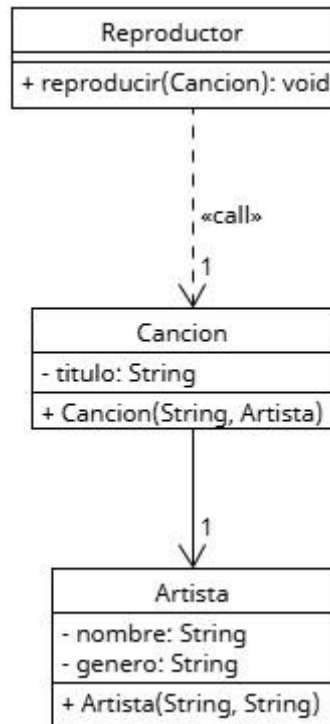
### Ejercicios de Dependencia de Uso

#### 11. Reproductor - Canción - Artista

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

Clases y atributos:

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



```
package Ejercicio_11;
```

```
public class Artista {
```

```
    private String nombre;
```

```
    private String genero;
```

```
    public Artista(String nombre, String genero) {
```

```
        this.nombre = nombre;    this.genero =
```

```
        genero;
```

```
    }
```

```
@Override public String toString() { return "Artista{" +  
"nombre=" + nombre + ", genero=" + genero + '}';  
}  
  
} package  
Ejercicio_11;  
  
public class Cancion {  
  
    private String titulo;  
  
    private Artista artista; // Asociación unidireccional  
  
    public Cancion(String titulo, Artista artista) {  
this.titulo = titulo;    this.artista = artista;  
    }  
  
    @Override public String toString() { return "Cancion{" +  
"titulo=" + titulo + ", artista=" + artista + '}';  
    }  
  
} package  
Ejercicio_11;
```

```
public class Reproductor {  
  
    public void reproducir(Cancion cancion) { // Dependencia de uso  
        System.out.println("Reproduciendo: " + cancion);  
    }  
  
}  
  
} package  
Ejercicio_11; public  
class  
Main_Ejercicio_11 {  
  
    public static void main(String[] args) {  
  
        Artista artista = new Artista("Shakira", "Pop");  
        Cancion cancion = new Cancion("Hips Don't Lie", artista);  
        Reproductor reproductor = new Reproductor();  
  
        reproductor.reproducir(cancion);  
  
    }  
  
}
```

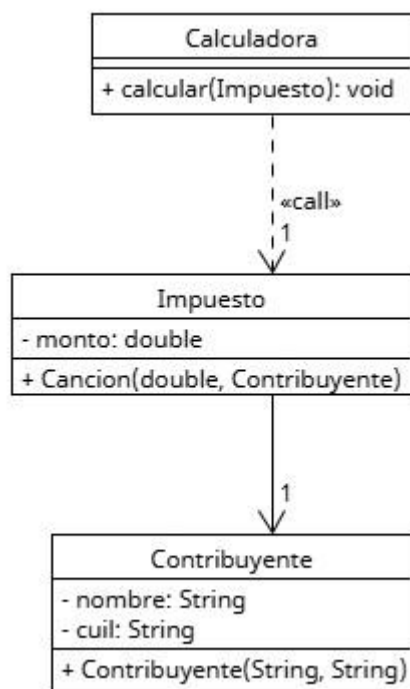
```
Output x Main_Ejercicio_02.java [-/A] x
< J25 UTN - TECNICATURA PROGRAMACION\2do Cuatrimestre\Programación II\GitHub x Jofre, Alan - Programación 2 -
run:
Reproduciendo: Cancion{titulo=Hips Don't Lie, artista=Artista{nombre=Shakira, genero=Pop}}
BUILD SUCCESSFUL (total time: 0 seconds)
```

## 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)



```
package Ejercicio_12;
```

```
public class Calculadora {
```

```
    public void calcular(Impuesto impuesto) { // Dependencia de uso
```

```
        System.out.println("Calculando impuesto: " + impuesto);
    }

} package

Ejercicio_12;

public class Contribuyente {

    private String nombre;

    private String cuil;

    public Contribuyente(String nombre, String cuil) {

this.nombre = nombre;    this.cuil = cuil;

    }

    @Override    public String toString() {    return "Contribuyente{" +

"nombre=" + nombre + ", cuil=" + cuil + '}';

    }

} package

Ejercicio_12;

public class Impuesto {
```



```
private double monto;    private Contribuyente contribuyente; //
```

Asociación unidireccional

```
public Impuesto(double monto, Contribuyente contribuyente) {  
  
this.monto = monto;      this.contribuyente = contribuyente;  
  
}  
  
@Override    public  
String toString() {  
  
    return "Impuesto{" + "monto=" + monto + ", contribuyente=" +  
contribuyente + '}';  
  
}
```

```
} package
```

```
Ejercicio_12;
```

```
public class Main_Ejercicio_12 {
```

```
public static void main(String[] args) {
```

```
    Contribuyente contribuyente = new Contribuyente("Roberto",  
"2012345678-9");
```

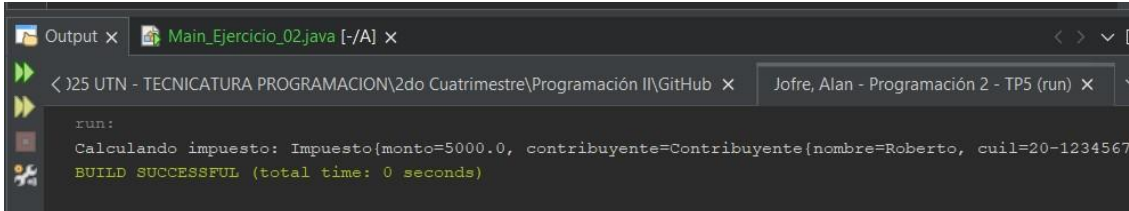
```
Impuesto impuesto = new Impuesto(5000.0, contribuyente);
```

```
Calculadora calc = new Calculadora();
```

```
calc.calcular(impuesto);
```

```
}
```

```
}
```



## DEPENDENCIA DE CREACIÓN

La clase crea otra dentro de un método, pero no la conserva como atributo..

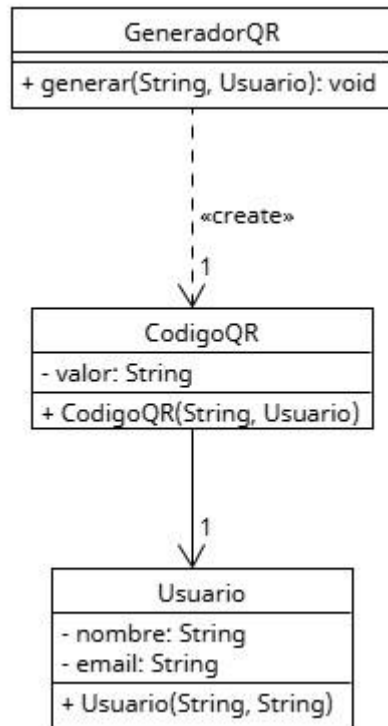
### Ejercicios de Dependencia de Creación

#### 13. GeneradorQR - Usuario - CódigoQR

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

Clases y atributos:

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



```
package Ejercicio_13;
```

```
public class CodigoQR {
```

```
    private String valor;    private Usuario usuario; //
```

```
Asociación unidireccional
```

```
    public CodigoQR(String valor, Usuario usuario) {
```

```
        this.valor = valor;        this.usuario = usuario;
```

```
    }
```

```
    @Override    public String toString() {        return "CodigoQR{" +
```

```
        "valor=" + valor + ", usuario=" + usuario + '}';
```

```
}
```

```
} package
```

```
Ejercicio_13;
```

```
public class GeneradorQR {
```

```
    public void generar(String valor, Usuario usuario) { // Dependencia de  
    creación
```

```
        CodigoQR qr = new CodigoQR(valor, usuario);
```

```
        System.out.println("QR generado para: " + usuario);
```

```
    }
```

```
} package
```

```
Ejercicio_13;
```

```
public class Usuario {
```

```
    private String nombre;
```

```
    private String email;
```

```
    public Usuario(String nombre, String email) {
```

```
        this.nombre = nombre;        this.email = email;
```

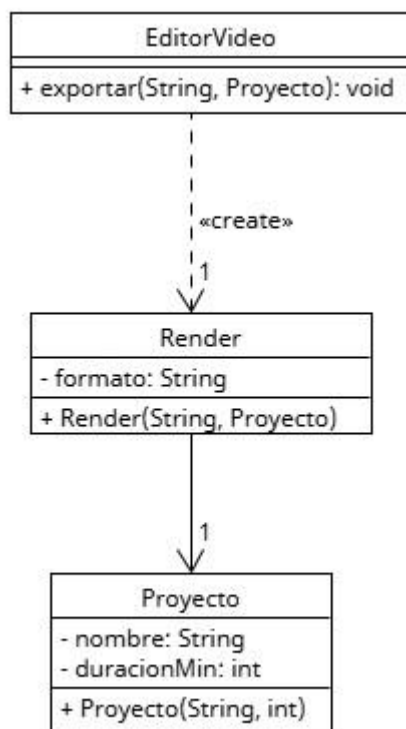
```
}  
  
    @Override    public String toString() {        return "Usuario{" +  
"nombre=" + nombre + ", email=" + email + "}";  
    }  
  
} package  
Ejercicio_13;  
  
public class Main_Ejercicio_13 {  
  
    public static void main(String[] args) {  
  
        Usuario usuario = new Usuario("Martina", "martina@mail.com");  
        GeneradorQR generador = new GeneradorQR();  
  
        generador.generar("https://miweb.com", usuario);  
  
    }  
  
}
```

```
Output x Main_Ejercicio_02.java [-/A] x
< 25 UTN - TECNICATURA PROGRAMACION\2do Cuatrimestre\Programación II\GitHub x

run:
QR generado para: Usuario{nombre=Martina, email=martina@mail.com}
BUILD SUCCESSFUL (total time: 0 seconds)
```

#### 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)



```
package Ejercicio_14;
```

```
public class EditorVideo {
```

```
    public void exportar(String formato, Proyecto proyecto) { // Dependencia de
creación

        Render render = new Render(formato, proyecto);

        System.out.println("Proyecto exportado en formato " + formato);

    }

}

package Ejercicio_14;

public class Proyecto {

    private String nombre;

    private int duracionMin;

    public Proyecto(String nombre, int duracionMin) {

        this.nombre = nombre;        this.duracionMin =
duracionMin;

    }

    @Override    public
String toString() {

        return "Proyecto{" + "nombre=" + nombre + ", duracionMin=" +
duracionMin + '}';

    }

}
```

```
} package
```

```
Ejercicio_14;
```

```
public class Render {
```

```
    private String formato;    private Proyecto proyecto; //
```

Asociación unidireccional

```
    public Render(String formato, Proyecto proyecto) {
```

```
        this.formato = formato;        this.proyecto = proyecto;
```

```
    }
```

```
    @Override    public String toString() {        return "Render{" + "formato="
```

```
+ formato + ", proyecto=" + proyecto + '}';
```

```
    }
```

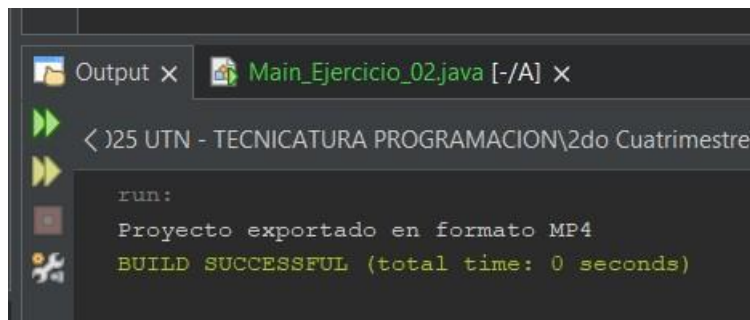
```
} package
```

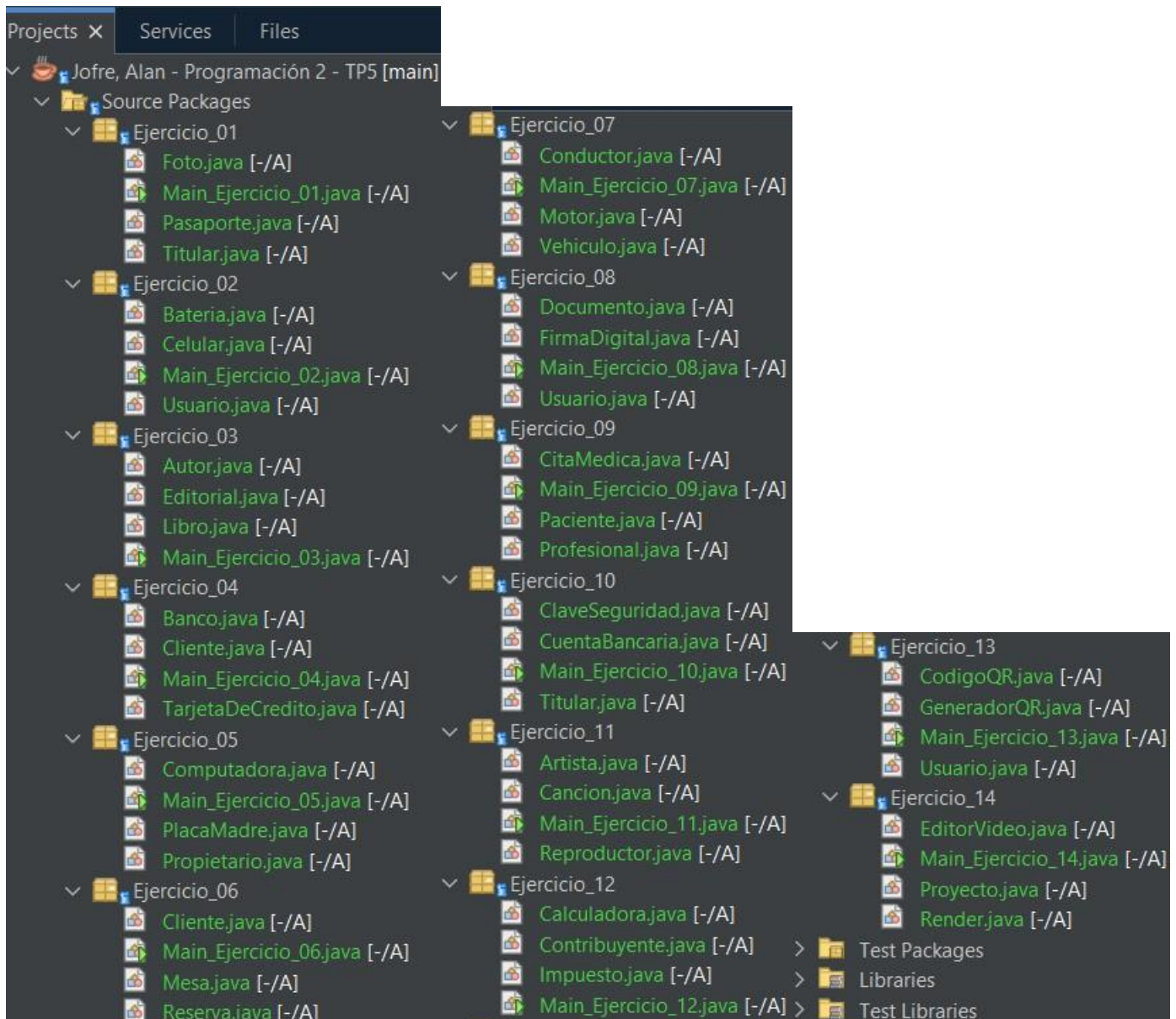
```
Ejercicio_14;
```

```
public class Main_Ejercicio_14 {
```



```
public static void main(String[] args) {  
  
    Proyecto proyecto = new Proyecto("Corto Animado", 15);  
  
    EditorVideo editor = new EditorVideo();  
  
    editor.exportar("MP4", proyecto);  
  
}  
  
}
```





## CONCLUSIONES ESPERADAS

- Diferenciar claramente los tipos de relaciones entre clases (asociación, agregación, composición).
- Representar las relaciones con la dirección adecuada en diagramas UML.
- Comprender e implementar dependencias de uso y de creación.
- Aplicar relaciones 1 a 1 en el diseño e implementación de clases en Java.
- Reforzar el análisis de modelos orientados a objetos y la capacidad de abstracción.