

PROGRAMACIÓN II

Trabajo Práctico 5: Relaciones UML 1 a 1

Alumno: Chiavón, Cristian

Repositorio GitHub: <https://github.com/cridvon/UTN-TUPaD-P2>

Caso Práctico

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

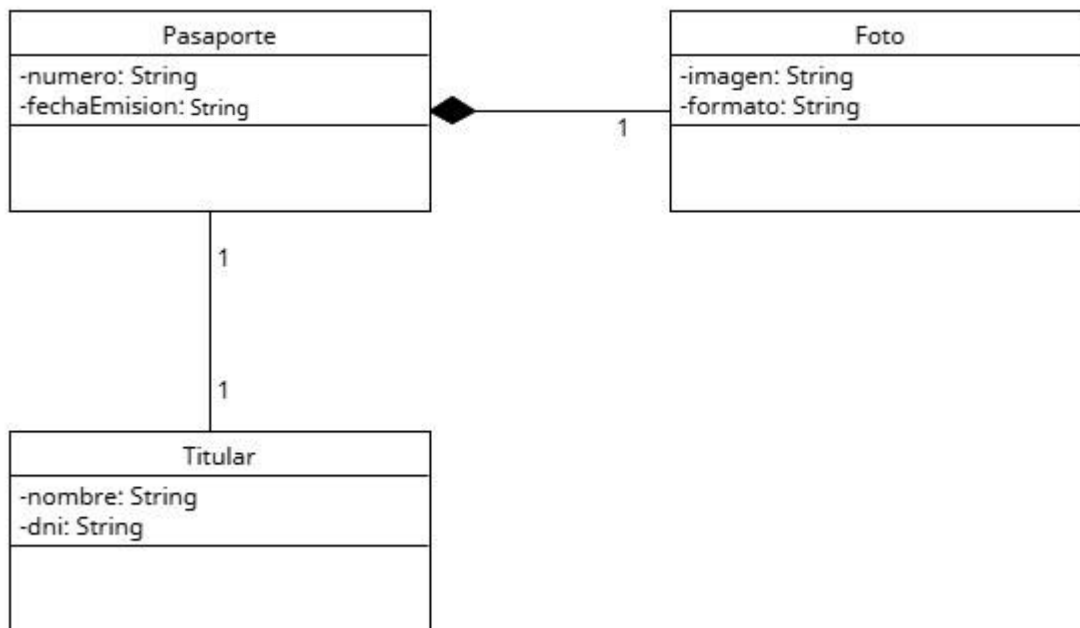
- Diagrama UML
- Tipo de relación (asociación, agregación, composición, dependencia)
- Dirección (unidireccional o bidireccional)
- Implementación de las clases con atributos y relaciones definidas

Método Main

```
1 package uml_basico;
2
3 /* IMPORTACIONES de clases -----
4 import uml_basico.Pasaporte_Foto_Titular.*;
5 import uml_basico.Celular_Bateria_Usuario.*;
6 import uml_basico.Libro_Autor_Editorial.*;
7 import uml_basico.TarjetaDeCrédito_Cliente_Banco.*;
8 import uml_basico.Computadora_PlacaMadre_Propietario.*;
9 import uml_basico.Reserva_Cliente_Mesa.*;
10 import uml_basico.Vehiculo_Motor_Conductor.*;
11 import uml_basico.Documento_FirmaDigital_Usuario.*;
12 import uml_basico.CitaMedica_Paciente_Profesional.*;
13 import uml_basico.CuentaBancaria_ClaveSeguridad_Titular.*;
14 import uml_basico.Reproductor_Cancion_Artista.*;
15 import uml_basico.Impuesto_Contribuyente_Calculadora.*;
16 import uml_basico.GeneradorQR_Usuario_CodigoQR.*;
17 import uml_basico.EditorVideo_Proyecto_Render.*;
18 ----- */
19 public class UML_Basico {
20
21     public static void Pasaporte_Foto_Titular() { ...14 lines }
22
23     public static void Celular_Bateria_Usuario() { ...2 lines }
24
25     public static void Libro_Autor_Editorial() { ...2 lines }
26
27     public static void TarjetaDeCrédito_Cliente_Banco() { ...2 lines }
28
29     public static void Computadora_PlacaMadre_Propietario() { ...2 lines }
30
31     public static void Reserva_Cliente_Mesa() { ...2 lines }
32
33     public static void Vehiculo_Motor_Conductor() { ...2 lines }
34
35     public static void Documento_FirmaDigital_Usuario() { ...2 lines }
36
37     public static void CitaMedica_Paciente_Profesional() { ...2 lines }
38
39     public static void CuentaBancaria_ClaveSeguridad_Titular() { ...2 lines }
40
41     public static void Reproductor_Cancion_Artista() { ...2 lines }
42
43     public static void Impuesto_Contribuyente_Calculadora() { ...2 lines }
44
45     public static void GeneradorQR_Usuario_CodigoQR() { ...2 lines }
46
47     public static void EditorVideo_Proyecto_Render() { ...2 lines }
48
49     public static void main(String[] args) {
50
51         Pasaporte_Foto_Titular();
52         Celular_Bateria_Usuario();
53         Libro_Autor_Editorial();
54         TarjetaDeCrédito_Cliente_Banco();
55         Computadora_PlacaMadre_Propietario();
56         Reserva_Cliente_Mesa();
57         Vehiculo_Motor_Conductor();
58         Documento_FirmaDigital_Usuario();
59         CitaMedica_Paciente_Profesional();
60         CuentaBancaria_ClaveSeguridad_Titular();
61         Reproductor_Cancion_Artista();
62         Impuesto_Contribuyente_Calculadora();
63         GeneradorQR_Usuario_CodigoQR();
64         EditorVideo_Proyecto_Render();
65
66     }
67 }
```

Ejercicios de Relaciones 1 a 1:

1. Pasaporte - Foto - Titular



Clase Pasaporte

```
package uml_basico.Pasaporte_Foto_Titular;

public class Pasaporte {

    private String numero;
    private String fechaEmision;
    private Foto foto;
    private Titular titular;

    public Pasaporte(String numero, String fechaEmision, String imagen, String formato) {
        this.numero = numero;
        this.fechaEmision = fechaEmision;
        this.foto = new Foto(imagen, formato);
    }

    public void setTitular(Titular titular) {
        this.titular = titular;
        if (titular != null && titular.getPasaporte() != this) {
            titular.setPasaporte(this);
        }
    }

    public Titular getTitular() {
        return this.titular;
    }

    public String getNombreTitular() {
        if (titular == null) {
            return "No tiene titular asignado";
        }
        return titular.getNombre();
    }

    public String getNumero() {
        return this.numero;
    }

    @Override
    public String toString() {
        return "Pasaporte{" + "numero=" + numero + ", fechaEmision=" + fechaEmision + ", foto=" + foto.getFoto() + ", titular=" + getNombreTitular() + '}';
    }
}
```

Clase Foto

```
public class Foto {  
  
    private String imagen;  
    private String formato;  
  
    public Foto(String imagen, String formato) {  
        this.imagen = imagen;  
        this.formato = formato;  
    }  
  
    public String getFoto() {  
        return this.imagen + "." + this.formato;  
    }  
  
}
```

Clase Titular

```
public 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;  
        if (pasaporte != null && pasaporte.getTitular() != this) {  
            pasaporte.setTitular(this);  
        }  
    }  
  
    public Pasaporte getPasaporte() {  
        return this.pasaporte;  
    }  
  
    public String getNumeroPasaporte() {  
        if (pasaporte == null) {  
            return "No tiene titular asignado";  
        }  
        return pasaporte.getNumero();  
    }  
  
    public String getNombre() {  
        if (pasaporte == null) {  
            return "No tiene titular asignado";  
        }  
        return this.nombre;  
    }  
  
    @Override  
    public String toString() {  
        return "Titular(" + "nombre=" + nombre + ", dni=" + dni + ", pasaporte=" + getNumeroPasaporte() + ')';  
    }  
}
```

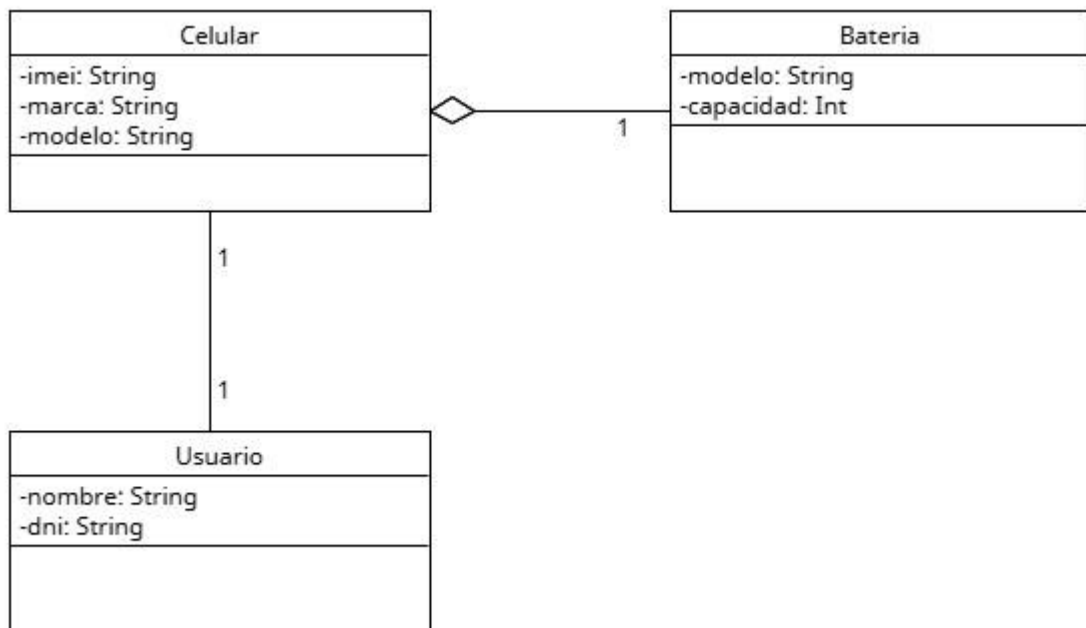
Aplicación del diagrama

```
public static void Pasaporte_Foto_Titular() {  
  
    Pasaporte pasaporte = new Pasaporte("321F", "15-2-25", "Foto Perfil", "png");  
    Titular titular = new Titular("Facu", "asdl23");  
  
    System.out.println("Datos guardados al iniciar los objetos----");  
    System.out.println(pasaporte);  
    System.out.println(titular);  
    pasaporte.setTitular(titular);  
    System.out.println("Datos guardados al setear un nuevo titular----");  
    System.out.println(pasaporte);  
    System.out.println(titular);  
  
}
```

Resultado obtenido

```
run:  
Datos guardados al iniciar los objetos----  
Pasaporte{numero=321F, fechaEmicion=15-2-25, foto=Foto Perfil.png, titular=No tiene titular asignado}  
Titular{nombre=Facu, dni=asdl23, pasaporte=No tiene pasaporte asignado}  
Datos guardados al setear un nuevo titular----  
Pasaporte{numero=321F, fechaEmicion=15-2-25, foto=Foto Perfil.png, titular=Facu}  
Titular{nombre=Facu, dni=asdl23, pasaporte=321F}  
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Celular - Batería - Usuario



Clase Celular

```
package uml_basico.Celular_Bateria_Usuario;

public class Celular {

    private String imei;
    private String marca;
    private String modelo;
    private Usuario usuario;
    private Bateria bateria;

    public Celular(String imei, String marca, String modelo, Bateria bateria) {
        this.imei = imei;
        this.marca = marca;
        this.modelo = modelo;
        this.bateria = bateria;
    }

    public void setUsuario(Usuario usuario) {
        this.usuario = usuario;
        if (usuario != null && usuario.getCelular() != this) {
            usuario.setCelular(this);
        }
    }

    public Usuario getUsuario() {
        return this.usuario;
    }

    public String getInfo() {
        return marca + " " + modelo;
    }

    public String getInfoUsuario() {
        if (this.usuario == null) {
            return "No tiene un usuario asignado";
        }
        return usuario.getInfo();
    }

    @Override
    public String toString() {
        return "Celular{" + "imei=" + imei + ", marca=" + marca + ", modelo=" + modelo + ", usuario=" + getInfoUsuario() + ", bateria=" + bateria.getInfoBateria() + '}';
    }
}
```


Clase Bateria

```
package uml_basico.Celular_Bateria_Usuario;

public class Bateria {

    private String modelo;
    private Integer capacidad;

    public Bateria(String modelo, Integer capacidad) {
        this.modelo = modelo;
        this.capacidad = capacidad;
    }

    public String getInfoBateria() {
        return modelo + " con capacidad de " + capacidad.toString();
    }

}
```

Clase Usuario

```
package uml_basico.Celular_Bateria_Usuario;

public 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;
        if (celular != null && celular.getUsuario() != this) {
            celular.setUsuario(this);
        }
    }

    public Celular getCelular() {
        return this.celular;
    }

    public String getInfo() {
        return nombre + " " + dni;
    }

    public String getInfoCelular() {
        if (this.celular == null) {
            return "No tiene un celular asignado";
        }
        return celular.getInfo();
    }

    @Override
    public String toString() {
        return "Usuario{" + "nombre=" + nombre + ", dni=" + dni + ", celular=" + getInfoCelular() + '}';
    }

}
```

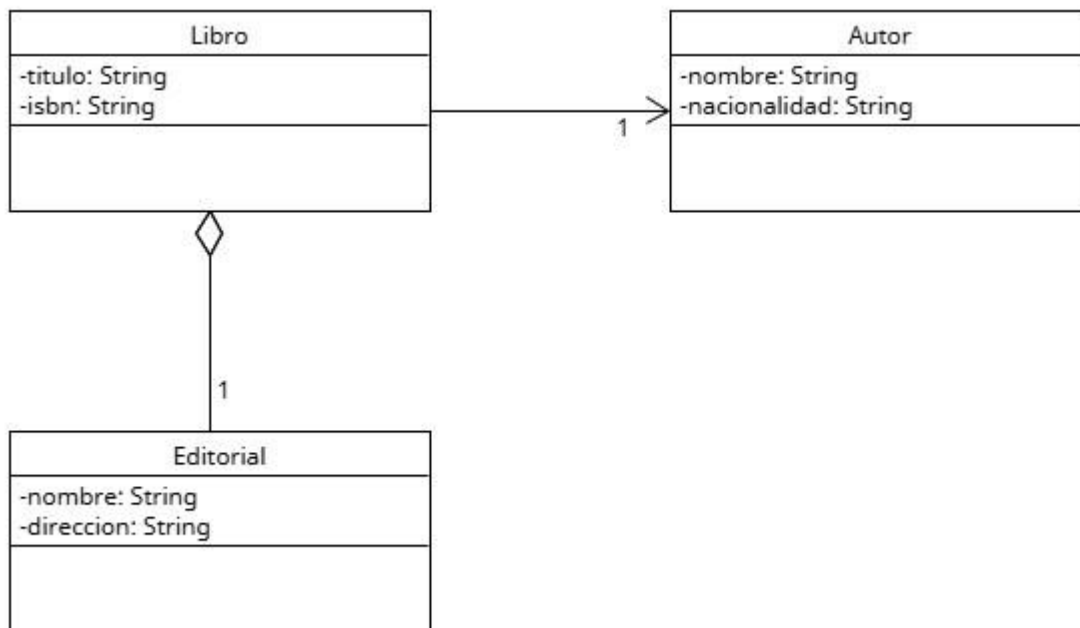
Aplicación del diagrama

```
public static void Celular_Bateria_Usuario() {  
    Bateria bateria = new Bateria("AJ32", 50);  
    Celular celular = new Celular("imei123", "Motorola", "G20", bateria); //Paso Bateria como parámetro  
    Usuario facu = new Usuario("Facu", "32325454");  
  
    System.out.println("Datos iniciales -----");  
    System.out.println(celular);  
    System.out.println(facu);  
    celular.setUsuario(facu); //Se asigna el usuario al celular  
    System.out.println("Datos asociados -----");  
    System.out.println(celular);  
    System.out.println(facu);  
}
```

Resultado obtenido

```
run:  
Datos iniciales -----  
Celular{imei=imei123, marca=Motorola, modelo=G20, usuario=No tiene un usuario asignado, bateria= AJ32 con capacidad de 50}  
Usuario{nombre=Facu, dni=32325454, celular=No tiene un celular asignado}  
Datos asociados -----  
Celular{imei=imei123, marca=Motorola, modelo=G20, usuario=Facu 32325454, bateria= AJ32 con capacidad de 50}  
Usuario{nombre=Facu, dni=32325454, celular=Motorola G20}  
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Libro - Autor - Editorial



Clase Libro

```
package uml_basico.Libro_Autor_Editorial;

public class Libro {

    private String titulo;
    private String isbn;
    private Autor autor;
    private Editorial editorial;

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

    public void setAutor(Autor autor) {
        this.autor = autor;
    }

    public void mostrarAutor() {
        if (this.autor != null) {
            System.out.println("El autor es: " + this.autor.getAutor());
        } else {
            System.out.println("Este libro no tiene asignado un autor aun");
        }
    }

    @Override
    public String toString() {
        return "Libro{" + "titulo=" + titulo + ", isbn=" + isbn + ", editorial=" + editorial + ", Autor= " + autor + '}';
    }

}
```


Clase Autor

```
package uml_basico.Libro_Autor_Editorial;

public class Autor {

    private String nombre;
    private String nacionalidad;

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

    public String getAutor() {
        return nombre + " de nacionalidad " + nacionalidad;
    }

}
```

Clase Editorial

```
package uml_basico.Libro_Autor_Editorial;

public class Editorial {

    private String nombre;
    private String direccion;

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

}
```

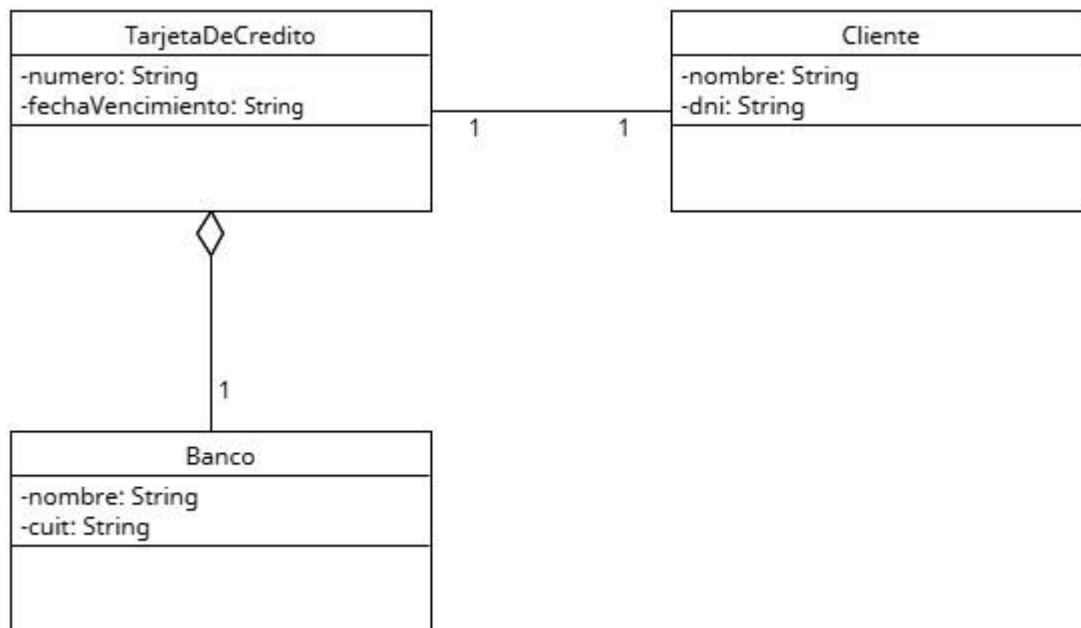
Aplicación del diagrama

```
public static void Libro_Autor_Editorial() {  
  
    Autor autor = new Autor("Edgar Alan Poe", "Ingles");  
    Editorial editorial = new Editorial("Saturno", "Siempre vivas 123");  
    Libro libro = new Libro("Sherlock Holmes", "9801", editorial);  
    System.out.println(libro);  
    libro.mostrarAutor();  
    libro.setAutor(author);  
    System.out.println(libro);  
    libro.mostrarAutor();  
  
}
```

Resultado obtenido

```
run:  
Libro(titulo=Sherlock Holmes, isbn=9801, editorial=uml_basico.Libro_Autor_Editorial.Editorial@2ff4acd0, Autor= null)  
Este libro no tiene asignado un autor aun  
Libro(titulo=Sherlock Holmes, isbn=9801, editorial=uml_basico.Libro_Autor_Editorial.Editorial@2ff4acd0, Autor= uml_basico.Libro_Autor_Editorial.Autor@27716f4)  
El autor es: Edgar Alan Poe de nacionalidad Ingles  
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. TarjetaDeCrédito - Cliente - Banco



Clase TarjetaDeCredito

```

package uml_basico.TarjetaDeCredito_Cliente_Banco;

public class TarjetaDeCredito {

    private String numero;
    private String fechaVencimiento;
    private Cliente cliente;
    private Banco banco;

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

    public String getInfo() {
        return "[" + numero + ", " + fechaVencimiento + ", " + banco.getInfo() + "]";
    }

    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
        if (cliente != null && cliente.getTarjeta() != this) {
            cliente.setTarjeta(this);
            System.out.println("Se asigna Tarjeta a cliente");
        }
    }

    public String mostrarCliente() {
        if (this.cliente != null) {
            return cliente.getInfo();
        }
        return "No tiene un cliente asignado";
    }

    public Cliente getCliente() {
        return cliente;
    }

    @Override
    public String toString() {
        return "TarjetaDeCredito[" + "numero=" + numero + ", fechaVencimiento=" + fechaVencimiento + ", cliente=" + mostrarCliente() + ", banco=" + banco.getInfo() + "]";
    }
}

```

Clase Cliente

```
package uml_basico.TarjetaDeCredito_Cliente_Banco;

public class Cliente {

    private String nombre;
    private String dni;
    private TarjetaDeCredito tarjeta;

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

    public String getInfo() {
        return "[" + nombre + ", " + dni + "]";
    }

    public void setTarjeta(TarjetaDeCredito tarjeta) {
        this.tarjeta = tarjeta;
        if (tarjeta != null && tarjeta.getCliente() != this) {
            tarjeta.setCliente(this);
            System.out.println("Se asigna Cliente a tarjeta");
        }
    }

    public TarjetaDeCredito getTarjeta() {
        return this.tarjeta;
    }

    public String mostrarTarjeta() {
        if (this.tarjeta != null) {
            return tarjeta.getInfo();
        }
        return "No tiene una tarjeta asignado";
    }

    @Override
    public String toString() {
        return "Cliente{" + "nombre=" + nombre + ", dni=" + dni + ", tarjeta=" + mostrarTarjeta() + '}';
    }
}
```

Clase Banco

```
package uml_basico.TarjetaDeCrédito_Cliente_Banco;

public class Banco {

    private String nombre;
    private String cuit;

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

    public String getInfo() {
        return "(" + nombre + ", " + cuit + ")";
    }

}
```

Aplicación del diagrama

```
public static void TarjetaDeCrédito_Cliente_Banco() {

    Cliente cliente = new Cliente("Facundo", "32324545");
    Banco banco = new Banco("ICBC", "123333411");
    TarjetaDeCredito tarjeta = new TarjetaDeCredito("1122", "12-12-22", banco);

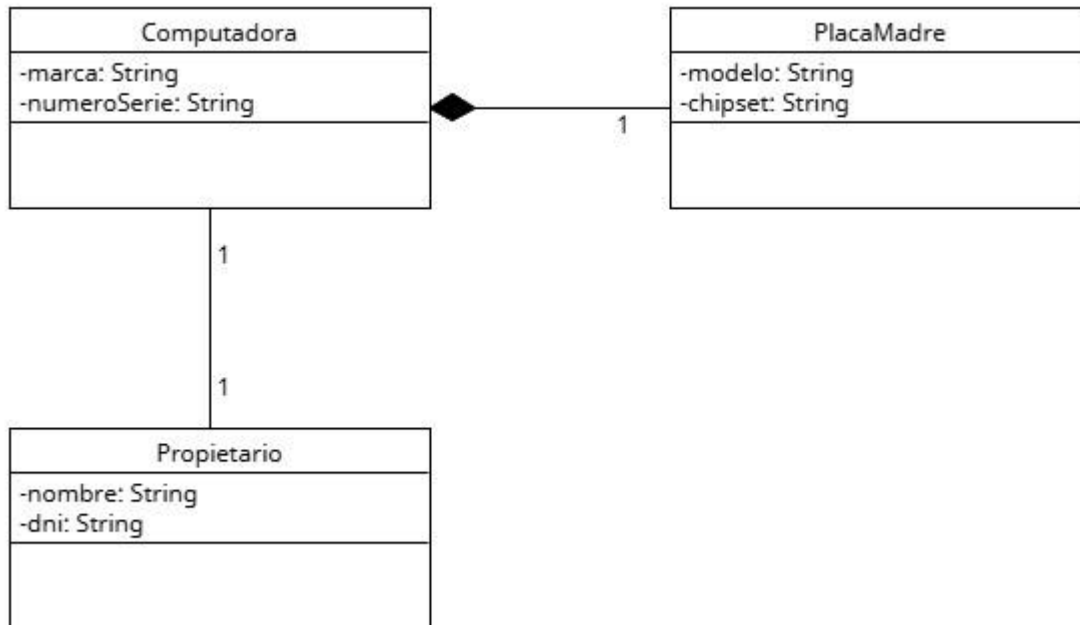
    System.out.println(tarjeta);
    System.out.println(cliente);
    tarjeta.setCliente(cliente);
    System.out.println(tarjeta);
    System.out.println(cliente);

}
```

Resultado obtenido

```
run:
TarjetaDeCredito[numero=1122, fechaVencimiento=12-12-22, cliente=No tiene un cliente asignado, banco={ICBC, 123333411}]
Cliente[nombre=Facundo, dni=32324545, tarjeta=No tiene una tarjeta asignado]
Se asigna Tarjeta a cliente
TarjetaDeCredito[numero=1122, fechaVencimiento=12-12-22, cliente={Facundo, 32324545}, banco={ICBC, 123333411}]
Cliente[nombre=Facundo, dni=32324545, tarjeta={1122, 12-12-22, {ICBC, 123333411}}]
BUILD SUCCESSFUL (total time: 0 seconds)
```


5. Computadora - PlacaMadre - Propietario



Clase Computadora

```

package uml_basico.Computadora_PlacaMadre_Propietario;

public class Computadora {

    private String marca;
    private String numeroSerie;
    private PlacaMadre placaMadre;
    private Propietario propietario;

    public Computadora(String marca, String numeroSerie, String modeloPlacaMadre, String chipsetPlacaMadre) {
        this.marca = marca;
        this.numeroSerie = numeroSerie;
        this.placaMadre = new PlacaMadre(modeloPlacaMadre, chipsetPlacaMadre);
    }

    public String getInfo() {
        return "(" + marca + ", " + numeroSerie + ", Placa Madre=" + placaMadre.getInfo() + ")";
    }

    public void setPropietario(Propietario propietario) {
        this.propietario = propietario;
        if (propietario != null && propietario.getComputadora() != this) {
            propietario.setComputadora(this);
            System.out.println("Se asigna Cliente a tarjeta");
        }
    }

    public Propietario getPropietario() {
        return this.propietario;
    }

    public String mostrarPropietario() {
        if (this.propietario != null) {
            return propietario.getInfo();
        }
        return "No tiene un propietario asignado";
    }

    @Override
    public String toString() {
        return "Computadora(" + "marca=" + marca + ", numeroSerie=" + numeroSerie + ", placaMadre=" + placaMadre.getInfo() + ", propietario=" + mostrarPropietario() + ")";
    }
}

```

Clase Placa Madre

```
package uml_basico.Computadora_PlacaMadre_Propietario;

public class PlacaMadre {

    private String modelo;
    private String chipset;

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

    public String getInfo() {
        return "{" + modelo + ", " + chipset + "}";
    }
}
```

Clase Propietario

```
package uml_basico.Computadora_PlacaMadre_Propietario;

public class Propietario {

    private String nombre;
    private String dni;
    private Computadora computadora;

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

    public String getInfo() {
        return "{" + nombre + ", " + dni + "}";
    }

    public void setComputadora(Computadora computadora) {
        this.computadora = computadora;
        if (computadora != null && computadora.getPropietario() != this) {
            computadora.setPropietario(this);
            System.out.println("Se asigna Cliente a tarjeta");
        }
    }

    public Computadora getComputadora() {
        return this.computadora;
    }

    public String mostrarComputadora() {
        if (this.computadora != null) {
            return computadora.getInfo();
        }
        return "No tiene una computadora asignado";
    }

    @Override
    public String toString() {
        return "Propietario{" + "nombre=" + nombre + ", dni=" + dni + ", computadora=" + mostrarComputadora() + '}';
    }
}
```

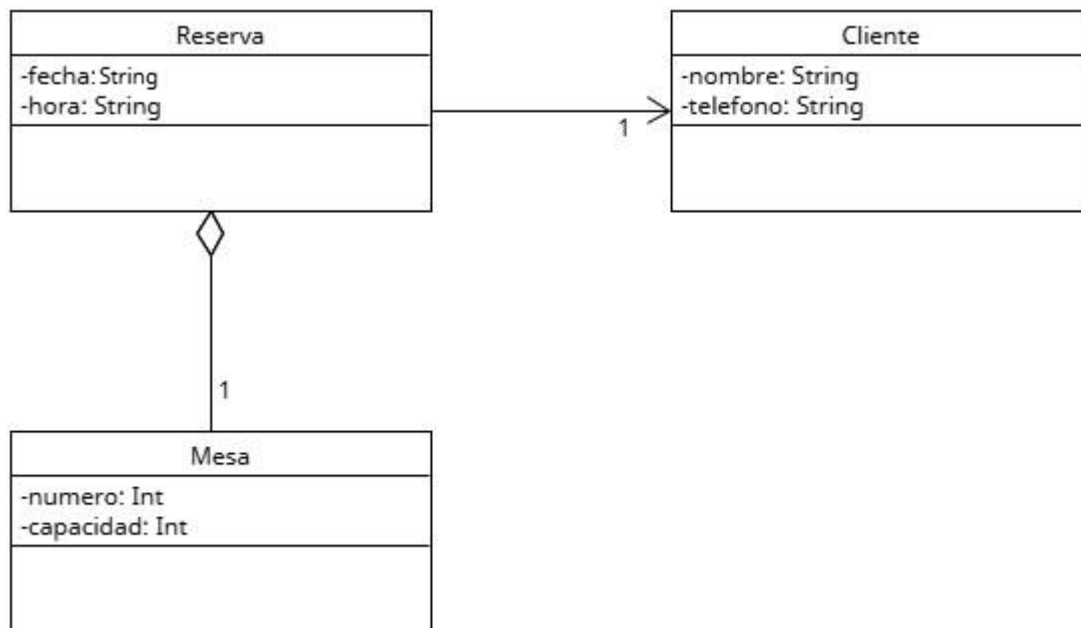
Aplicación del diagrama

```
public static void Computadora_PlacaMadre_Propietario() {  
  
    Computadora computadora = new Computadora("HP", "232425T", "A520M Pro", "Intel I5");  
    Propietario propietario = new Propietario("Joaquin", "45121212");  
  
    System.out.println(computadora);  
    System.out.println(propietario);  
    propietario.setComputadora(computadora);  
    System.out.println("Se asigna computadora a propietario");  
    System.out.println(computadora);  
    System.out.println(propietario);  
  
}
```

Resultado obtenido

```
run:  
Computadora{marca=HP, numeroSerie=232425T, placaMadre={A520M Pro, Intel I5}, propietario=No tiene un propietario asignado}  
Propietario{nombre=Joaquin, dni=45121212, computadora=No tiene una computadora asignado}  
Se asigna Cliente a tarjeta  
Se asigna computadora a propietario  
Computadora{marca=HP, numeroSerie=232425T, placaMadre={A520M Pro, Intel I5}, propietario={Joaquin, 45121212}}  
Propietario{nombre=Joaquin, dni=45121212, computadora={HP, 232425T, Placa Madre= {A520M Pro, Intel I5}}}  
BUILD SUCCESSFUL (total time: 0 seconds)
```

6. Reserva - Cliente - Mesa



Clase Reserva

```
package uml_basico.Reserva_Cliente_Mesa;

public class Reserva {

    private String fecha;
    private String hora;
    private Cliente cliente; //Agregación 1:1
    private Mesa mesa;

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

    public String getInfo() {
        return "{" + fecha + ", " + hora + ", Cliente = " + mostrarCliente() + "}";
    }

    public String mostrarCliente() {
        return cliente.getInfo();
    }

    public void setMesa(Mesa mesa) {
        this.mesa = mesa;
        if (mesa != null && mesa.getReserva() != this) {
            mesa.setReserva(this);
        }
    }

    public Mesa getMesa() {
        return this.mesa;
    }

    public String mostrarMesa() {
        if (this.mesa != null) {
            return mesa.getInfo();
        }
        return "No hay mesa asignada";
    }

    @Override
    public String toString() {
        return "Reserva{" + "fecha=" + fecha + ", hora=" + hora + ", cliente=" + mostrarCliente() + ", mesa=" + mostrarMesa() + "}";
    }
}
```

Clase Cliente

```
package uml_basico.Reserva_Cliente_Mesa;

public class Cliente {

    private String nombre;
    private String telefono;

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

    public String getInfo() {
        return "{" + nombre + ", " + telefono + "}";
    }
}
```


Clase Mesa

```
package uml_basico.Reserva_Cliente_Mesa;

public class Mesa {

    private Integer numero;
    private Integer capacidad;
    private Reserva reserva;

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

    public String getInfo() {
        return "[" + numero + ", " + capacidad + "]";
    }

    public void setReserva(Reserva reserva) {
        this.reserva = reserva;
        if (reserva != null && reserva.getMesa() != this) {
            reserva.setMesa(this);
        }
    }

    public Reserva getReserva() {
        return this.reserva;
    }

    public String mostrarReserva() {
        if (this.reserva != null) {
            return reserva.getInfo();
        }
        return "No hay reserva asignada";
    }

    @Override
    public String toString() {
        return "Mesa{" + "numero=" + numero + ", capacidad=" + capacidad + ", reserva=" + mostrarReserva() + "}";
    }
}
```

Aplicación del diagrama

```
public static void Reserva_Cliente_Mesa() {

    Cliente cliente = new Cliente("Pedro", "11225544");
    Reserva reserva = new Reserva("12-12-25", "14:00", cliente);
    Mesa mesa = new Mesa(5, 2);

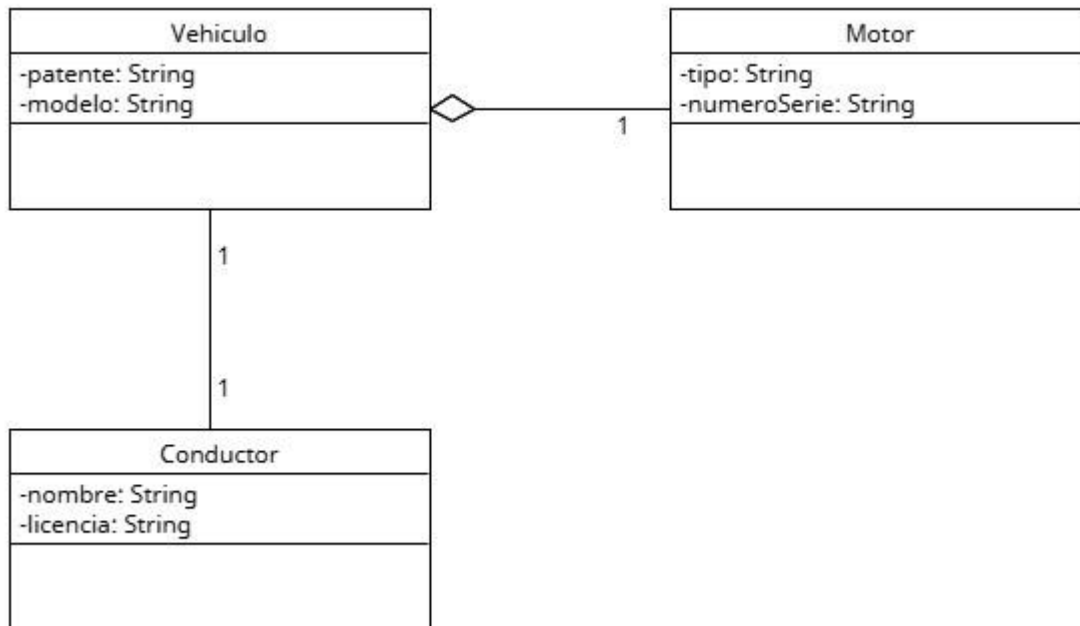
    System.out.println(reserva);
    System.out.println(mesa);
    System.out.println("Se asigna reserva a Mesa");
    mesa.setReserva(reserva);
    System.out.println(reserva);
    System.out.println(mesa);

}
```

Resultado obtenido

```
run:
Reserva{fecha=12-12-25, hora=14:00, cliente={Pedro, 11225544}, mesa=No hay mesa asignada}
Mesa{numero=5, capacidad=2, reserva=No hay reserva asignada}
Se asigna reserva a Mesa
Reserva{fecha=12-12-25, hora=14:00, cliente={Pedro, 11225544}, mesa={5, 2}}
Mesa{numero=5, capacidad=2, reserva={12-12-25, 14:00, Cliente = {Pedro, 11225544}}}}
BUILD SUCCESSFUL (total time: 0 seconds)
```

7. Vehículo - Motor - Conductor



Clase Vehiculo

```
package uml_basico.Vehiculo_Motor_Conductor;

public class Vehiculo {

    private String patente;
    private String modelo;
    private Motor motor;
    private Conductor conductor;

    public Vehiculo(String patente, String modelo, Motor motor) {
        this.patente = patente;
        this.modelo = modelo;
        this.motor = motor;
    }

    public String getInfo() {
        return "{" + patente + ", " + modelo + ", Motor=" + motor.getInfo() + "}";
    }

    public void setConductor(Conductor conductor) {
        this.conductor = conductor;
        if (conductor != null && conductor.getVehiculo() != this) {
            conductor.setVehiculo(this);
        }
    }

    public Conductor getConductor() {
        return this.conductor;
    }

    public String mostrarConductor() {
        if (this.conductor != null) {
            return conductor.getInfo();
        }
        return "No hay conductor asociado";
    }

    @Override
    public String toString() {
        return "Vehiculo{" + "patente=" + patente + ", modelo=" + modelo + ", motor=" + motor.getInfo() + ", Conductor=" + mostrarConductor() + "}";
    }
}
```

Clase Motor

```
package uml_basico.Vehiculo_Motor_Conductor;

public class Motor {

    private String tipo;
    private String numeroSerie;

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

    public String getInfo() {
        return "{" + tipo + ", " + numeroSerie + "}";
    }
}
```

Clase Conductor

```
package uml_basico.Vehiculo_Motor_Conductor;

public class Conductor {

    private String nombre;
    private String licencia;
    private Vehiculo vehiculo;

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

    public String getInfo() {
        return "[" + nombre + ", " + licencia + "]";
    }

    public void setVehiculo(Vehiculo vehiculo) {
        this.vehiculo = vehiculo;
        if (vehiculo != null && vehiculo.getConductor() != this) {
            vehiculo.setConductor(this);
        }
    }

    public Vehiculo getVehiculo() {
        return this.vehiculo;
    }

    public String mostrarVehiculo() {
        if (this.vehiculo != null) {
            return vehiculo.getInfo();
        }
        return "No hay vehiculo asociado";
    }

    @Override
    public String toString() {
        return "Conductor[" + "nombre=" + nombre + ", licencia=" + licencia + ", vehiculo=" + mostrarVehiculo() + ']';
    }
}
```

Aplicación del diagrama

```
public static void Vehiculo_Motor_Conductor() {

    Motor v8 = new Motor("V8", "2222F");
    Vehiculo vehiculo = new Vehiculo("12AA23", "Nissan", v8);
    Conductor conductor = new Conductor("Rodrigo", "124578");

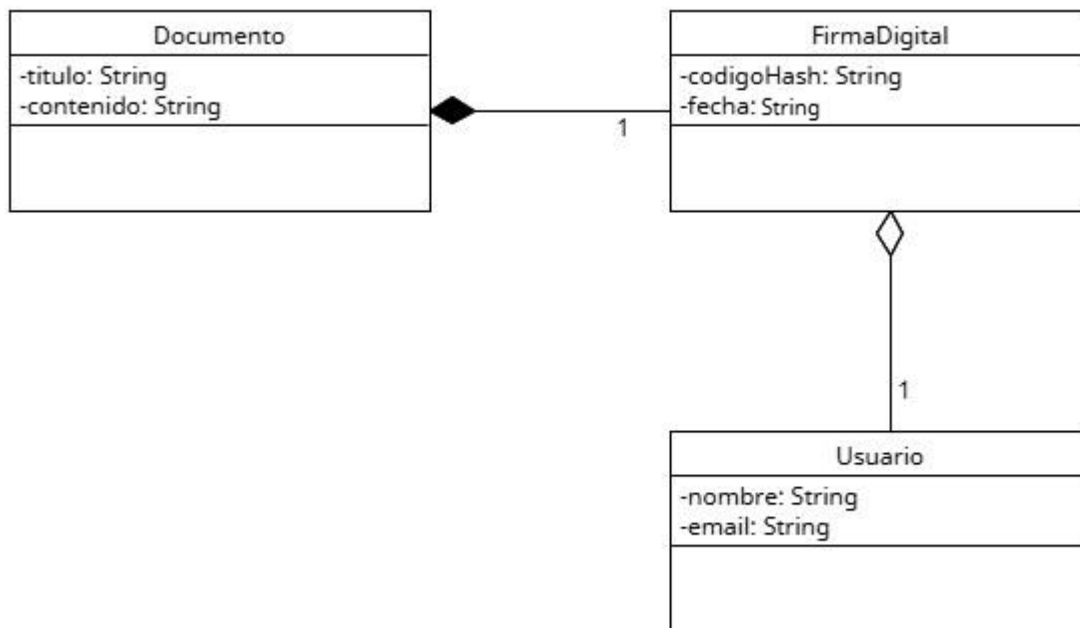
    System.out.println(vehiculo);
    System.out.println(conductor);
    System.out.println("Asigno vehiculo a Conductor");
    conductor.setVehiculo(vehiculo);
    System.out.println(vehiculo);
    System.out.println(conductor);

}
```

Resultado obtenido

```
run:
Vehiculo{patente=l2AA23, modelo=Nissan, motor={V8, 2222F}, Conductor=No hay conductor asociado}
Conductor{nombre=Rodrigo, licencia=l24578, vehiculo=No hay vehiculo asociado}
Asigno vehiculo a Conductor
Vehiculo{patente=l2AA23, modelo=Nissan, motor={V8, 2222F}, Conductor={Rodrigo, l24578}}
Conductor{nombre=Rodrigo, licencia=l24578, vehiculo={l2AA23, Nissan, Motor={V8, 2222F}}}}
BUILD SUCCESSFUL (total time: 0 seconds)
```

8. Documento - FirmaDigital - Usuario



Clase Documento

```
package uml_basico.Documento_FirmaDigital_Usuario;

public class Documento {

    private String titulo;
    private String contenido;

    private FirmaDigital firmaDigital;

    public Documento(String titulo, String contenido, String codigoHash, String fecha, Usuario usuario) {
        this.titulo = titulo;
        this.contenido = contenido;
        this.firmaDigital = new FirmaDigital(codigoHash, fecha, usuario);
    }

    @Override
    public String toString() {
        return "Documento{" + "titulo=" + titulo + ", contenido=" + contenido + ", firmaDigital=" + firmaDigital.getInfo() + '}';
    }

}
```


Clase FirmaDigital

```
package uml_basico.Documento_FirmaDigital_Usuario;

public 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;
    }

    public String getInfo() {
        return "{CodigoHash= " + codigoHash + ", fecha= " + fecha + ", Usuario= " + usuario.getInfo() + "}";
    }

}
```

Clase Usuario

```
package uml_basico.Documento_FirmaDigital_Usuario;

public class Usuario {

    private String nombre;
    private String email;

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

    public String getInfo() {
        return "{nombre= " + nombre + ", email= " + email + "}";
    }

}
```

Aplicación del diagrama

```
public static void Documento_FirmaDigital_Usuario() {

    Usuario usuario = new Usuario("Jaime", "jaimito@gmail.com");
    Documento documento = new Documento("Contrato de Venta", "Contenido del contrato", "#225", "25/9/25", usuario);

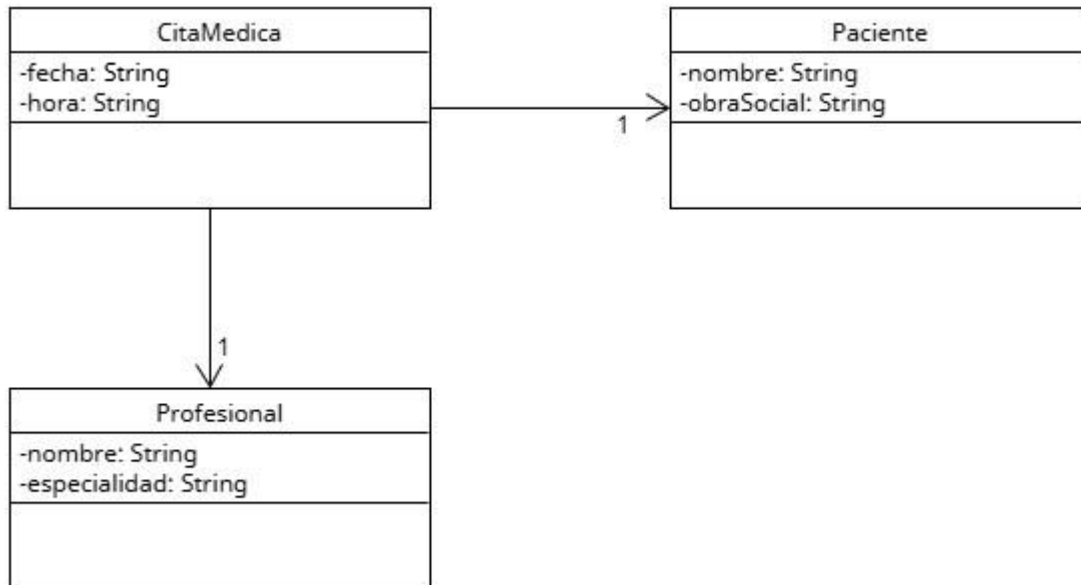
    System.out.println(documento);

}
```

Resultado obtenido

```
run:
Documento[titulo=Contrato de Venta, contenido=Contenido del contrato, firmaDigital={CodigoHash= #225, fecha= 25/9/25, Usuario= {nombre= Jaime, email= jaimito@gmail.com}}]
BUILD SUCCESSFUL (total time: 0 seconds)
```

9. CitaMédica - Paciente - Profesional



Clase CitaMedica

```
package uml_basico.CitaMedica_Paciente_Profesional;

public class CitaMedica {

    private String fecha;
    private String hora;
    private Profesional profesional;
    private Paciente paciente;

    public CitaMedica(String fecha, String hora) {
        this.fecha = fecha;
        this.hora = hora;
    }

    public void setProfesional(Profesional profesional) {
        this.profesional = profesional;
    }

    public void setPaciente(Paciente paciente) {
        this.paciente = paciente;
    }

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

}
```

Clase Paciente

```
package uml_basico.CitaMedica_Paciente_Profesional;

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 + '}';
    }

}
```

Clase Profesional

```
package uml_basico.CitaMedica_Paciente_Profesional;

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 + '}';
    }

}
```

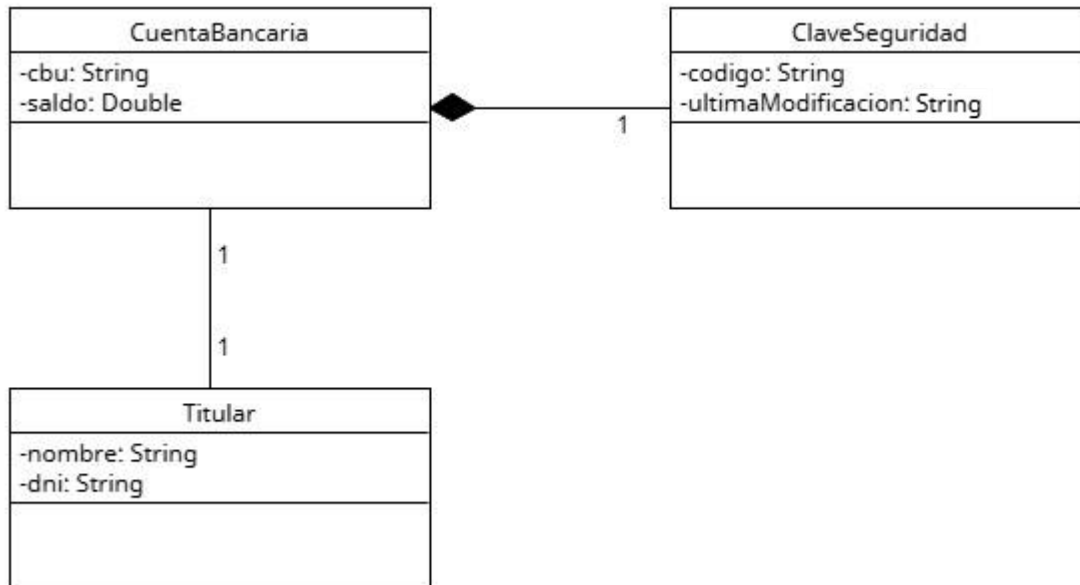
Aplicación del diagrama

```
public static void CitaMedica_Paciente_Profesional() {  
  
    Profesional profesional = new Profesional("Dr. Ricardo", "Odontologo");  
    Paciente paciente = new Paciente("Carlos", "Galeno");  
    CitaMedica citaMedica = new CitaMedica("12-4-25", "11:00hs");  
    System.out.println(citaMedica);  
    System.out.println(paciente);  
    System.out.println(profesional);  
    System.out.println("Se asignan paciente y profesional a cita medica");  
    citaMedica.setPaciente(paciente);  
    citaMedica.setProfesional(profesional);  
    System.out.println(citaMedica);  
  
}
```

Resultado obtenido

```
run:  
CitaMedica[fecha=12-4-25, hora=11:00hs, profesional=null, paciente=null]  
Paciente[nombre=Carlos, obraSocial=Galeno]  
Profesional[nombre=Dr. Ricardo, especialidad=Odontologo]  
Se asignan paciente y profesional a cita medica  
CitaMedica[fecha=12-4-25, hora=11:00hs, profesional=Profesional[nombre=Dr. Ricardo, especialidad=Odontologo], paciente=Paciente[nombre=Carlos, obraSocial=Galeno]]  
BUILD SUCCESSFUL (total time: 0 seconds)
```

10. CuentaBancaria - ClaveSeguridad - Titular



Clase CuentaBancaria

```
package uml_basico.CuentaBancaria_ClaveSeguridad_Titular;

public class CuentaBancaria {

    private String cbu;
    private Double saldo;
    private ClaveSeguridad claveSeguridad;
    private Titular titular;

    public CuentaBancaria(String cbu, Double saldo, String codigo, String ultimaModificacion) {
        this.cbu = cbu;
        this.saldo = saldo;
        this.claveSeguridad = new ClaveSeguridad(codigo, ultimaModificacion);
    }

    public String getInfo() {
        return "cbu= " + cbu;
    }

    public void setTitular(Titular titular) {
        this.titular = titular;
        if (titular != null && titular.getCuentaBancaria() != this) {
            titular.setCuentaBancaria(this);
        }
    }

    public Titular getTitular() {
        return this.titular;
    }

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

}
```


Clase ClaveSeguridad

```
package uml_basico.CuentaBancaria_ClaveSeguridad_Titular;

public class ClaveSeguridad {

    private String codigo;
    private String ultimaModificación;

    public ClaveSeguridad(String codigo, String ultimaModificación) {
        this.codigo = codigo;
        this.ultimaModificación = ultimaModificación;
    }

    @Override
    public String toString() {
        return "ClaveSeguridad{" + "codigo=" + codigo + ", ultimaModificaci\u00f3n=" + ultimaModificación + '}';
    }

}
```

Clase Titular

```
package uml_basico.CuentaBancaria_ClaveSeguridad_Titular;

public class Titular {

    private String nombre;
    private String dni;
    private CuentaBancaria cuentaBancaria;

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

    public void setCuentaBancaria(CuentaBancaria cuentaBancaria) {
        this.cuentaBancaria = cuentaBancaria;
        if (cuentaBancaria != null && cuentaBancaria.getTitular() != this) {
            cuentaBancaria.setTitular(this);
        }
    }

    public CuentaBancaria getCuentaBancaria() {
        return this.cuentaBancaria;
    }

    public String mostrarCuentaBancaria() {
        if (this.cuentaBancaria != null) {
            return cuentaBancaria.getInfo();
        }
        return null;
    }

    @Override
    public String toString() {
        return "Titular{" + "nombre=" + nombre + ", dni=" + dni + ", cuentaBancaria= {" + mostrarCuentaBancaria() + '}';
    }

}
```

Aplicación del diagrama

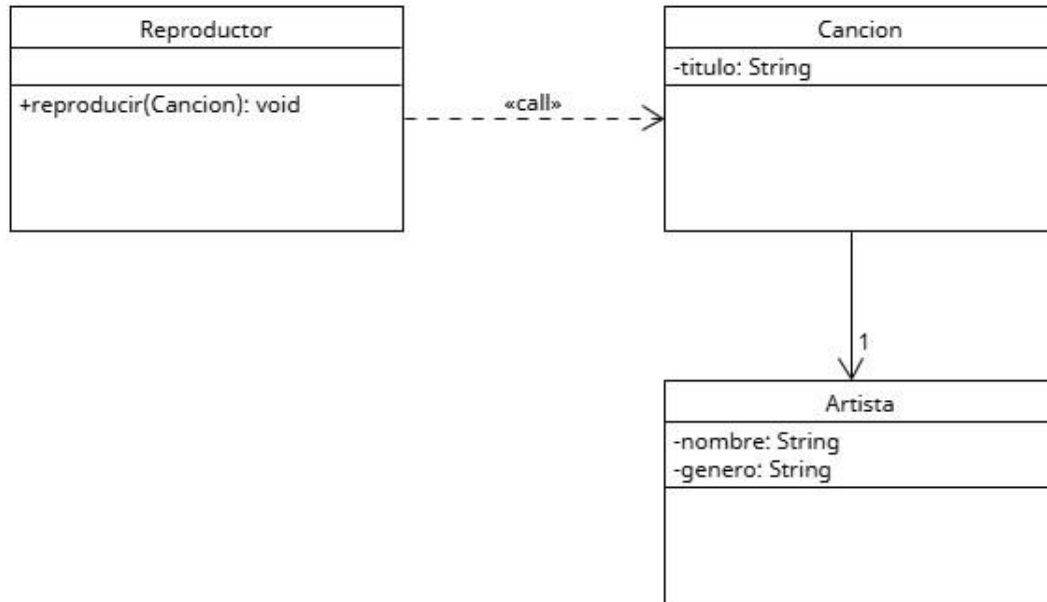
```
public static void CuentaBancaria_ClaveSeguridad_Titular() {  
  
    CuentaBancaria cuentaBancaria = new CuentaBancaria("11225454", 500.00, "1234", "20/9/2025");  
    Titular titular = new Titular("Facundo", "32111111");  
  
    System.out.println(cuentaBancaria);  
    System.out.println(titular);  
    cuentaBancaria.setTitular(titular);  
    System.out.println("Se compone la cuenta bancaria con el titular");  
    System.out.println(cuentaBancaria);  
    System.out.println(titular);  
  
}
```

Resultado obtenido

```
run:  
CuentaBancaria(cbu=11225454, saldo=500.0, claveSeguridad=ClaveSeguridad[codigo=1234, ultimaModificaci  
Titular(nombre=Facundo, dni=32111111, cuentaBancaria= [null])  
Se compone la cuenta bancaria con el titular  
CuentaBancaria(cbu=11225454, saldo=500.0, claveSeguridad=ClaveSeguridad[codigo=1234, ultimaModificaci  
Titular(nombre=Facundo, dni=32111111, cuentaBancaria= [cbu= 11225454])  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ejercicios de Dependencia de Uso

11. Reproductor - Canción - Artista



Clase Reproductor

```
package uml_basico.Reproductor_Cancion_Artista;

public class Reproductor {

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

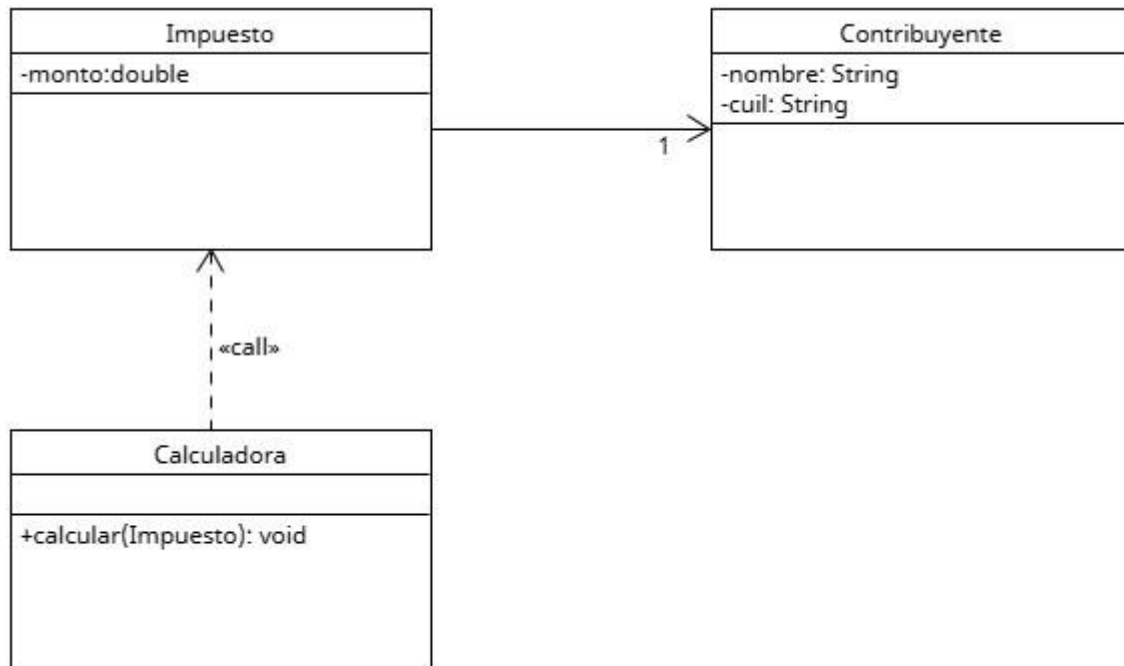
}
```

Clase Cancion

```
package uml_basico.Reproductor_Cancion_Artista;  
  
public class Cancion {  
  
    private String titulo;  
    private Artista artista;  
  
    public Cancion(String titulo) {  
        this.titulo = titulo;  
    }  
  
    public void setArtista(Artista artista) {  
        this.artista = artista;  
    }  
  
    @Override  
    public String toString() {  
        return "Cancion{" + "titulo=" + titulo + ", artista=" + artista + '}';  
    }  
}
```

Clase Artista

12. Impuesto - Contribuyente - Calculadora



Clase Impuesto

```
package uml_basico.Impuesto_Contribuyente_Calculadora;

public class Impuesto {

    private Double monto;
    private Contribuyente contribuyente;

    public Impuesto(Double monto) {
        this.monto = monto;
    }

    public void setContribuyente(Contribuyente contribuyente) {
        this.contribuyente = contribuyente;
    }

    @Override
    public String toString() {
        return "Impuesto{" + "monto=" + monto + ", contribuyente=" + contribuyente + '}';
    }

}
```

Clase Contribuyente

```
package uml_basico.Impuesto_Contribuyente_Calculadora;

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 + '}';
    }

}
```

Clase Calculadora

```
package uml_basico.Impuesto_Contribuyente_Calculadora;

public class Calculadora {

    public void calcular(Impuesto impuesto) {
        System.out.println("Se aplica impuesto: " + impuesto);
    }

}
```

Aplicación del diagrama

```
public static void Impuesto_Contribuyente_Calculadora() {
    Contribuyente contribuyente = new Contribuyente("Javier", "20-23232323-7");
    Impuesto impuesto = new Impuesto(700.00);
    Calculadora calculadora = new Calculadora();

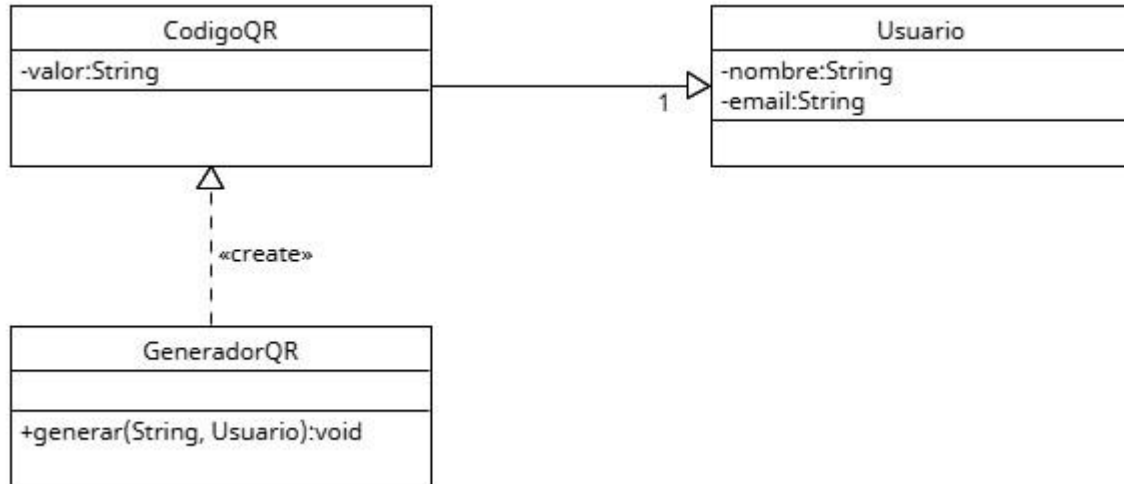
    System.out.println(impuesto);
    System.out.println("Se asocia contribuyente a impuesto");
    impuesto.setContribuyente(contribuyente);
    System.out.println(impuesto);
    System.out.println("Se utiliza calculadora");
    calculadora.calcular(impuesto);
}
```

Resultado obtenido

```
run:
Impuesto{monto=700.0, contribuyente=null}
Se asocia contribuyente a impuesto
Impuesto{monto=700.0, contribuyente=Contribuyente{nombre=Javier, cuil=20-23232323-7}}
Se utiliza calculadora
Se aplica impuesto: Impuesto{monto=700.0, contribuyente=Contribuyente{nombre=Javier, cuil=20-23232323-7}}
BUILD SUCCESSFUL (total time: 0 seconds)
```


Ejercicios de Dependencia de Creación

13. GeneradorQR - Usuario - CódigoQR



Clase GeneradorQR

```
package uml_basico.GeneradorQR_Usuario_CodigoQR;

public class GeneradorQR {

    private CodigoQR codigoQR;

    public void generar(String valor, Usuario usuario) {
        this.codigoQR = new CodigoQR(valor);
        System.out.println("Se crea codigoQR: " + codigoQR);
        this.codigoQR.setUsuario(usuario);
        System.out.println("Se asocia usuario: " + codigoQR);
    }

}
```

Clase Usuario

```
package uml_basico.GeneradorQR_Usuario_CodigoQR;

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 + '}';
    }

}
```

Clase CodigoQR

```
package uml_basico.GeneradorQR_Usuario_CodigoQR;

public class CodigoQR {

    private String valor;
    private Usuario usuario;

    public CodigoQR(String valor) {
        this.valor = valor;
    }

    public void setUsuario(Usuario usuario) {
        this.usuario = usuario;
    }

    @Override
    public String toString() {
        return "CodigoQR{" + "valor=" + valor + ", usuario=" + usuario + '}';
    }

}
```

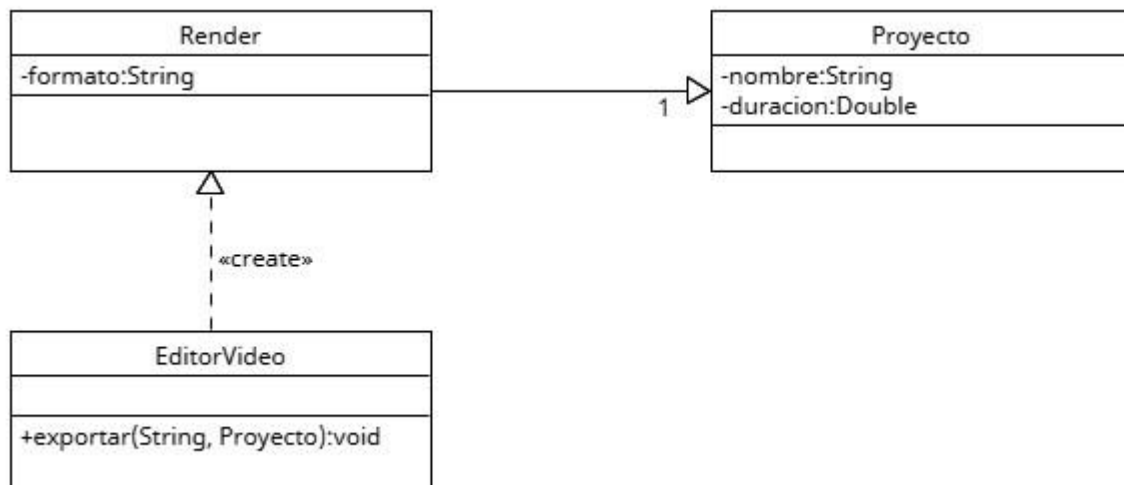
Aplicación del diagrama

```
public static void GeneradorQR_Usuario_CodigoQR() {  
    Usuario usuario = new Usuario("Facu", "farvon@gmail.com");  
    GeneradorQR generadorQR = new GeneradorQR();  
  
    generadorQR.generar("00101100", usuario);  
}
```

Resultado obtenido

```
run:  
Se crea codigoQR: CodigoQR{valor=00101100, usuario=null}  
Se asocia usuario: CodigoQR{valor=00101100, usuario=Usuario{nombre=Facu, email=farvon@gmail.com}}  
BUILD SUCCESSFUL (total time: 0 seconds)
```

14. EditorVideo - Proyecto - Render



Clase EditorVideo

```
package uml_basico.EditorVideo_Proyecto_Render;

public class EditorVideo {

    Render render;

    public void exportar(String formato, Proyecto proyecto) {
        this.render = new Render(formato);
        System.out.println("Genero el render");
        System.out.println(render);
        System.out.println("Asocio el proyecto al render");
        render.setProyecto(proyecto);
        System.out.println(render);
    }

}
```

Clase Proyecto

```
package uml_basico.EditorVideo_Proyecto_Render;

public class Proyecto {

    private String nombre;
    private Double duracion;

    public Proyecto(String nombre, Double duracion) {
        this.nombre = nombre;
        this.duracion = duracion;
    }

    @Override
    public String toString() {
        return "Proyecto{" + "nombre=" + nombre + ", duracion=" + duracion + '}';
    }

}
```

Clase Render

```
package uml_basico.EditorVideo_Proyecto_Render;

public class EditorVideo {

    Render render;

    public void exportar(String formato, Proyecto proyecto) {
        this.render = new Render(formato);
        System.out.println("Genero el render");
        System.out.println(render);
        System.out.println("Asocio el proyecto al render");
        render.setProyecto(proyecto);
        System.out.println(render);
    }

}
```

Aplicación del diagrama

```
public static void EditorVideo_Proyecto_Render() {
    Proyecto proyecto = new Proyecto("Proyecto Final", 15.00);
    EditorVideo editorVideo = new EditorVideo();

    editorVideo.exportar("mp4", proyecto);
}
```

Resultado obtenido

```
run:
Genero el render
Render{formato=mp4, proyecto=null}
Asocio el proyecto al render
Render{formato=mp4, proyecto=Proyecto{nombre=Proyecto Final, duracion=15.0}}
BUILD SUCCESSFUL (total time: 0 seconds)
```