PROGRAMACIÓN II

TRABAJO PRÁCTICO Nº5: UML

CHRISTIAN EMMANUEL OLIVERO

https://github.com/CHRISTIAN2320/UTN-TUPAD-Programacion2

Ejercicios de Relaciones 1 a 1

1. Pasaporte - Foto - Titular

a. Composición: Pasaporte → Foto

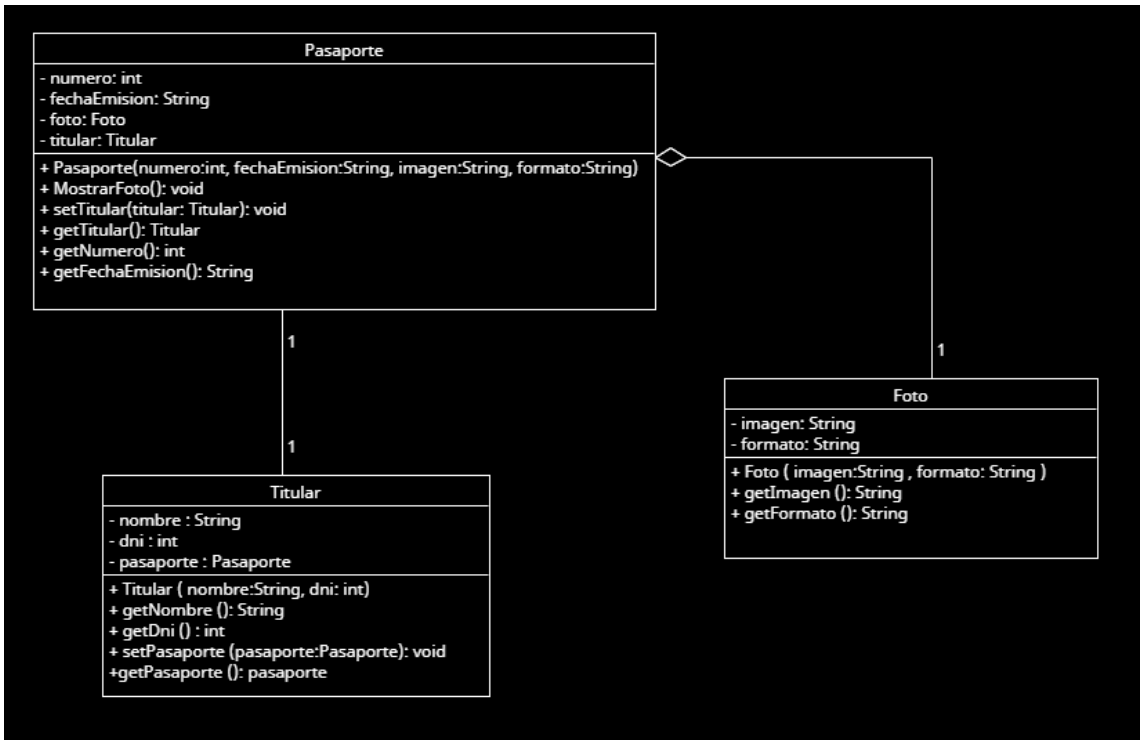
b. Asociación bidireccional: Pasaporte ↔ Titular

Clases y atributos:

i. Pasaporte: numero, fechaEmision

ii. Foto: imagen, formato

iii. Titular: nombre, dni



```java
public class Pasaporte {
    private int numero;
    private String fechaEmision;
    private Foto foto ;   // composicion
    private Titular titular; // Bidirecional
    //CONSTRUCTOR
    public Pasaporte(int numero, String fechaEmision, String imagen, String formato) {
        this.numero = numero;
        this.fechaEmision = fechaEmision;
        this.foto = new Foto(imagen, formato);
    }
    public void MostrarFoto (){
        System.out.println("La imagen del pasaporte es: " + foto.getImagen() +
                " y el formato es " + foto.getFormato() + "\n");
    }
    //Establecemos la relacion Bidireccional
    public void setTitular(Titular titular) {
        this.titular = titular;
        if (titular != null && titular.getPasaporte()!= this) {
            titular.setPasaporte(this);
        }
    }
```

```java
  public Titular getTitular() {
      return titular;
  }
  public int getNumero() {
      return numero;
  }
  public String getFechaEmision() {
      return fechaEmision;
  }
```

```java
public class Foto {
    private String imagen;
    private String formato;
    // CONSTRUCTOR
    public Foto(String imagen, String formato) {
        this.imagen = imagen;
        this.formato = formato;
    }
    public String getImagen() {
        return imagen;
    }
    public String getFormato() {
        return formato;
    }
}
```

```java
public class Titular {
    private String nombre;
    private int dni;
    private Pasaporte pasaporte; //Rel. Bidireccional
    //Constructor
    public Titular(String nombre, int dni) {
        this.nombre = nombre;
        this.dni = dni;
    }
    public String getNombre() {
        return nombre;
    }
    public int getDni() {
        return dni;
    }
    public void setPasaporte(Pasaporte pasaporte) {
        this.pasaporte = pasaporte;
        if (pasaporte != null && pasaporte.getTitular()!= this) {
            pasaporte.setTitular(this);
        }
    }
    public Pasaporte getPasaporte() {
        return pasaporte;
    }
}
```

2. Celular - Batería - Usuario

a. Agregación: Celular → Batería

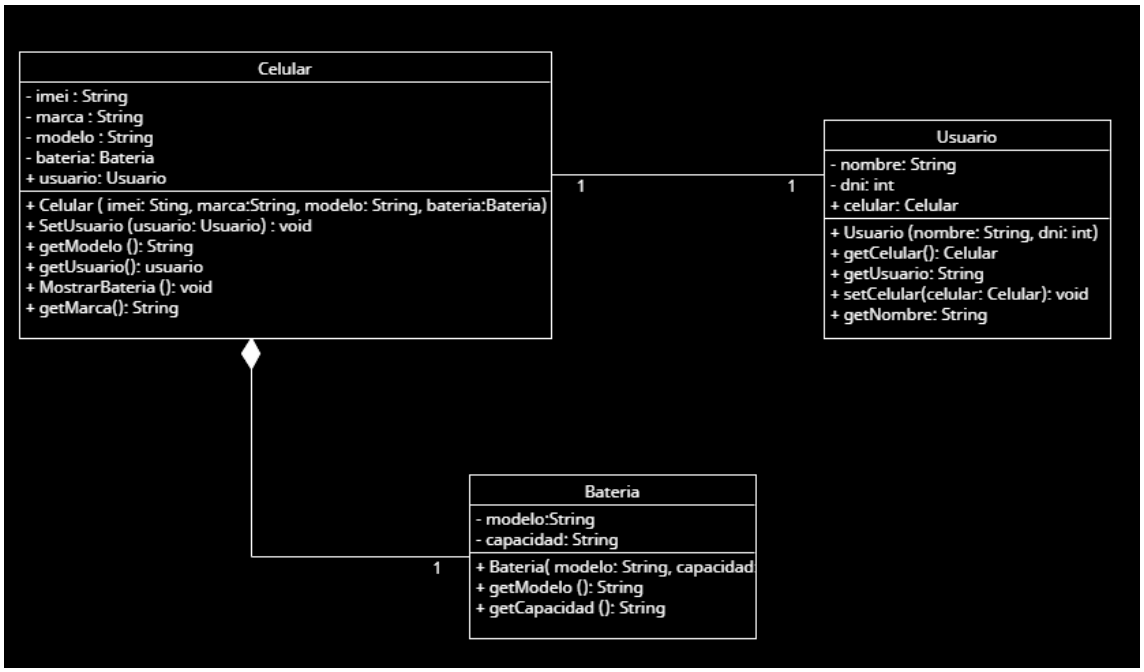
b. Asociación bidireccional: Celular ↔ Usuario

Clases y atributos:

i. Celular: imei, marca, modelo

ii. Batería: modelo, capacidad

iii. Usuario: nombre, dni




```java
public class Celular {

    private String imei;
    private String marca;
    private String modelo;
    private Bateria bateria; // AGREGACION 1:1
    public Usuario usuario; // rel Bidireccional
    //CONSTRUCTOR
    public Celular(String imei, String marca, String modelo, Bateria bateria) {
        this.imei = imei;
        this.marca = marca;
        this.modelo = modelo;
        this.bateria= bateria;
    }
    // REFERENCIA DE USUARIO
    public void setUsuario(Usuario usuario) {
        this.usuario = usuario;
        if (usuario != null && usuario.getCelular() != this) {
            usuario.setCelular(this);
        }
```

```java
public String getModelo() {
    return modelo;
}
public Usuario getUsuario() {
    return usuario;
}
public void MostrarBateria(){
    System.out.println("La bateria que tiene el celular es " + bateria.getModelo()
            + " y tiene una capacidad de " + bateria.getCapacidad() + "\n" );
}
public String getMarca() {
    return marca;
}
```

```java
public class Usuario {
    private String nombre ;
    private int dni;
    public Celular celular;
    //CONSTRUCTOR
    public Usuario(String nombre, int dni) {
        this.nombre = nombre;
        this.dni = dni;
    }
   public Celular getCelular() {
        return celular;
    }
    public String getUsuario() {
        return nombre;
    }
    // REFERENCIA DE CELULAR
    public void setCelular(Celular celular) {
        this.celular = celular;
        if (celular != null && celular.getUsuario()!= this) {
            celular.setUsuario(this);
        }
    }
    public String getNombre() {
        return nombre;
    }
}
```

```java
public class Bateria {
    private String modelo;
    private String capacidad;
    //Constructor
    public Bateria(String modelo, String capacidad) {
        this.modelo = modelo;
        this.capacidad = capacidad;
    }
    public String getModelo() {
        return modelo;
    }
    public String getCapacidad() {
        return capacidad;
    }
```

```java
public class Ejercicio2 {

    public static void main(String[] args) {
        // Agregacion
        Bateria bateria = new Bateria("AA55", "6000mA");
        Celular celular = new Celular("ABC_123", "Samsung", "A54", bateria);
        celular.MostrarBateria();
        // Bidireccionalidad
        Usuario usuario = new Usuario("Christian", 40123456);
        celular.setUsuario(usuario);
        System.out.println("El dueño del celuar es: " + celular.getUsuario().getNombre());
        System.out.println("Los datos datos del celular son: \n" +
                " Marca: " + usuario.getCelular().getMarca()
                + "\n Modelo: " +usuario.getCelular().getModelo() );
    }
}
```

```
La bateria que tiene el celular es AA55 y tiene una capacidad de 6000mA

El due□o del celuar es: Christian
Los datos datos del celular son:
 Marca: Samsung
 Modelo: A54
```

3. Libro - Autor - Editorial

a. Asociación unidireccional: Libro → Autor

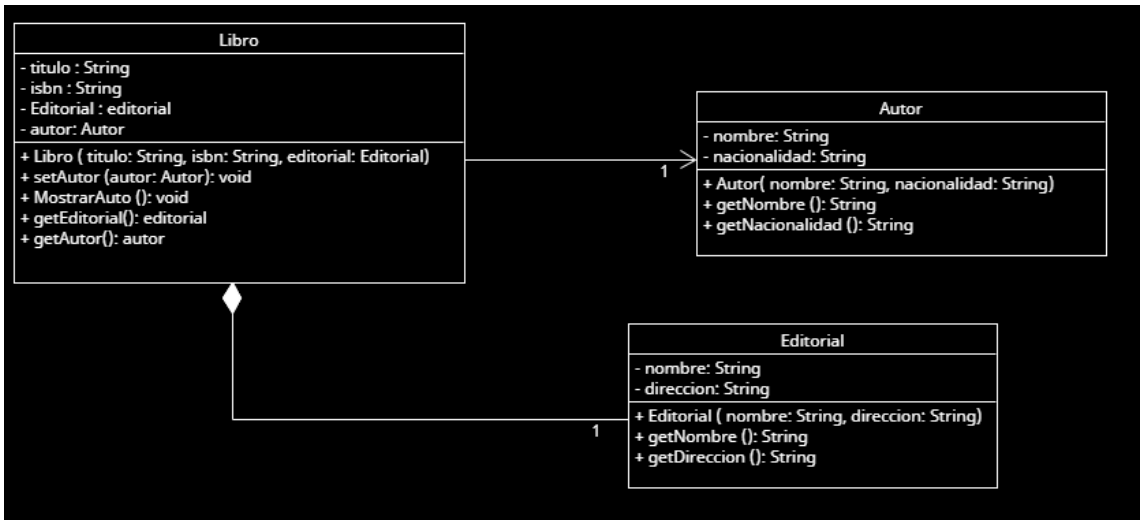
b. Agregación: Libro → Editorial

Clases y atributos:

i. Libro: titulo, isbn

ii. Autor: nombre, nacionalidad

iii. Editorial: nombre, dirección




```java
public class Libro {
    private String titulo;
    private String isbn;
    private Editorial editorial; // ASOCIACION DE AGREGACION 1:1
    private Autor autor; // ASOCIACION UNIDIRECCIONAL 1:1
    //Constructor
    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 (){
        System.out.println("El nombre del autor es:" + autor.getNombre());
    }
    public Editorial getEditorial() {
        return editorial;
    }

    public Autor getAutor() {
        return autor;
    }
```

```java
public class Autor {
    private String nombre;
    private String nacionalidad;
    //Constructor
    public Autor(String nombre, String nacionalidad) {
        this.nombre = nombre;
        this.nacionalidad = nacionalidad;
    }
    public String getNombre() {
        return nombre;
    }
    public String getNacionalidad() {
        return nacionalidad;
    }
}
```

```java
public class Editorial {
    private String nombre;
    private String direccion;
    //Constructor
    public Editorial(String nombre, String direccion) {
        this.nombre = nombre;
        this.direccion = direccion;
    }
    public String getNombre() {
        return nombre;
    }
    public String getDireccion() {
        return direccion;
    }
}
```

```java
public static void main(String[] args) {
    //Agregacion
    Editorial editorial = new Editorial("Planeta", "Av.Siempre viva");
    Libro libro = new Libro("DUNE", "ABC123", editorial);
    System.out.println("La editorial del libro es: " + libro.getEditorial().getNombre());
    //Asociacion
    Autor autor = new Autor("Frank Herbert", "EEUU");
    libro.setAutor(autor);
    System.out.println("El autor del libro es " + libro.getAutor().getNombre()
            + " de nacionalidad " + libro.getAutor().getNacionalidad());
}
```

```
La editorial del libro es: Planeta
El autor del libro es Frank Herbert de nacionalidad EEUU
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. TarjetaDeCrédito - Cliente - Banco

a. Asociación bidireccional: **TarjetaDeCrédito ↔ Cliente**

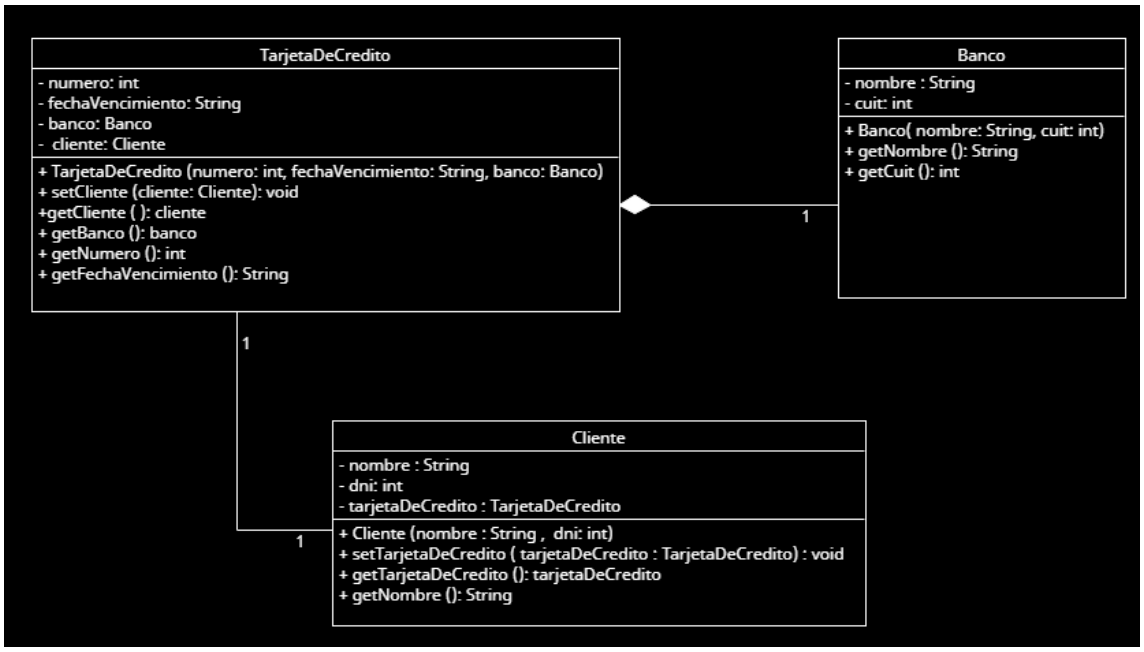
b. Agregación: **TarjetaDeCrédito → Banco**

Clases y atributos:

i. TarjetaDeCrédito: numero, fechaVencimiento

ii. Cliente: nombre, dni

iii. Banco: nombre, cuit




```java
public class TarjetaDeCredito {
    private int numero;
    private String fechaVencimiento;
    private Banco banco; // AGREGACION 1:1
    private Cliente cliente ; // ASOC. BIDIRECCIONAL 1:1
    //Constructor
    public TarjetaDeCredito(int numero, String fechaVencimiento, Banco banco) {
        this.numero = numero;
        this.fechaVencimiento = fechaVencimiento;
        this.banco = banco;
    }
    // ASOCIACION
    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
        if (cliente != null && cliente.getTarjetaDeCredito()!= this ) {
            cliente.setTarjetaDeCredito(this);
        }
    }
```

```java
    public Cliente getCliente() {
        return cliente;
    }
    public Banco getBanco() {
        return banco;
    }
    public int getNumero() {
        return numero;
    }
    public String getFechaVencimiento() {
        return fechaVencimiento;
    }
}
```

```java
public class Cliente {
    private String nombre;
    private int dni;
    private TarjetaDeCredito tarjetaDeCredito ;
    // CONSTRUCTOR
    public Cliente(String nombre, int dni) {
        this.nombre = nombre;
        this.dni = dni;
    }
    public void setTarjetaDeCredito(TarjetaDeCredito tarjetaDeCredito) {
        this.tarjetaDeCredito = tarjetaDeCredito;
        if (tarjetaDeCredito != null && tarjetaDeCredito.getCliente() != this) {
            tarjetaDeCredito.setCliente(this);
        }
    }
    public TarjetaDeCredito getTarjetaDeCredito() {
        return tarjetaDeCredito;
    }
    public String getNombre() {
        return nombre;
    }
}
```

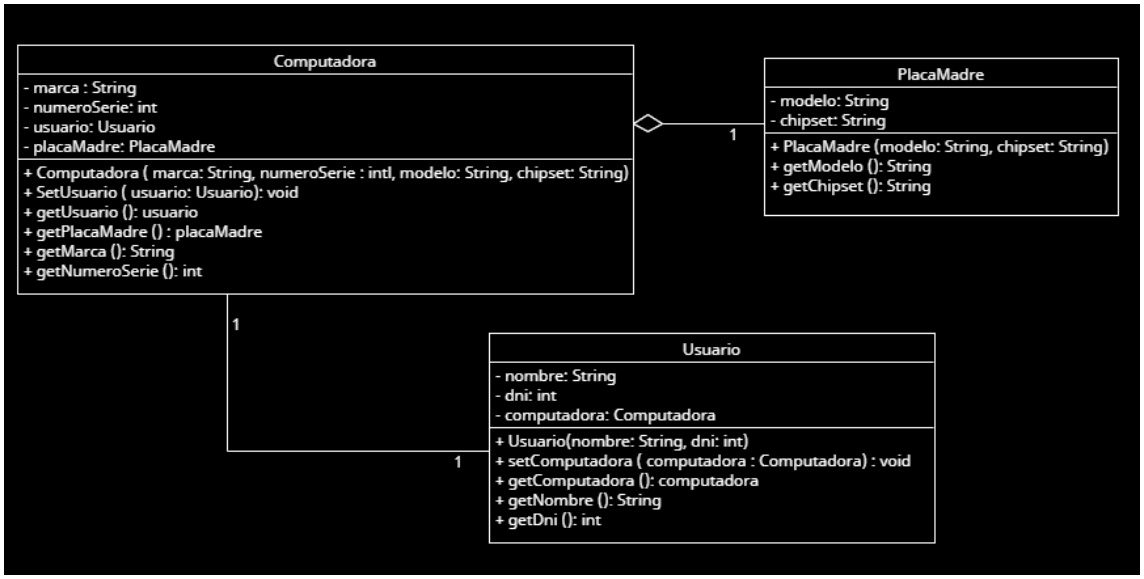
```java
public class Banco {
    private String nombre;
    private int cuit;
    // Constructor
    public Banco(String nombre, int cuit) {
        this.nombre = nombre;
        this.cuit = cuit;
    }
    public String getNombre() {
        return nombre;
    }
    public int getCuit() {
        return cuit;
    }
}
```

```java
public class Ejercicio4 {
    public static void main(String[] args) {
        // AGREGACION
        Banco banco = new Banco("Galicia", 789456);
        TarjetaDeCredito tarjeta= new TarjetaDeCredito(123456, "29/09/2027", banco);
        System.out.println("El banco de la tarjeta es: " + tarjeta.getBanco().getNombre() + "\n");
        // ASOC. BIDIRECCIONAL:
        Cliente cliente = new Cliente("Christian", 123456);
        tarjeta.setCliente(cliente);
        System.out.println("El propietario de la tarjeta es: " + tarjeta.getCliente().getNombre() +"\n");
        System.out.println("Los datos de la tarjeta son: "+ "\nNumero:" + cliente.getTarjetaDeCredito().getNumero()
        + "\nFecha de vencimiento: " + cliente.getTarjetaDeCredito().getFechaVencimiento());
    }
```

```
El banco de la tarjeta es: Galicia

El propietario de la tarjeta es: Christian

Los datos de la tarjeta son:
Numero:123456
Fecha de vencimiento: 29/09/2027
BUILD SUCCESSFUL (total time: 0 seconds)
```

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




```java
public class Computadora {
    private String marca;
    private int numeroSerie;
    private Usuario usuario; //Rel BIDIRECCIONAL 1:1
    private PlacaMadre placaMadre;  // COMPOSICION
    //CONSTRUCTOR
    public Computadora(String marca, int numeroSerie, String modelo, String chipset) {
        this.marca = marca;
        this.numeroSerie = numeroSerie;
        this.placaMadre = new PlacaMadre(modelo, chipset);
    }
    // RELACION
    public void setUsuario(Usuario usuario) {
        this.usuario = usuario;
        if (usuario != null && usuario.getComputadora()!=this) {
            usuario.setComputadora(this);
        }
    }
    public Usuario getUsuario() {
        return usuario;
    }
    public PlacaMadre getPlacaMadre() {
        return placaMadre;
    }
    public String getMarca() {
        return marca;
    }
    public int getNumeroSerie() {
        return numeroSerie;
    }
```

```java
public class PlacaMadre {
    private String modelo;
    private String chipset;
    //Constructor
    public PlacaMadre(String modelo, String chipset) {
        this.modelo = modelo;
        this.chipset = chipset;
    }
    public String getModelo() {
        return modelo;
    }
    public String getChipset() {
        return chipset;
    }
```

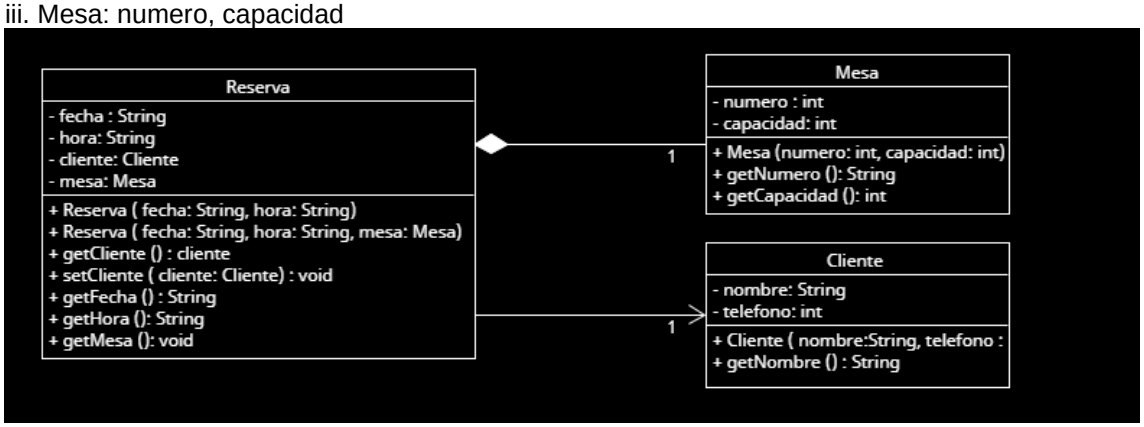
```java
public class Usuario {
    private String nombre;
    private int dni;
    private Computadora computadora; // Rel. BIDIRECCIONAL
    //Constructor
    public Usuario(String nombre, int dni) {
        this.nombre = nombre;
        this.dni = dni;
    }
    // Relacion
    public void setComputadora(Computadora computadora) {
        this.computadora = computadora;
        if (computadora != null && computadora.getUsuario() != this) {
            computadora.setUsuario(this);
        }
    }
    public Computadora getComputadora() {
        return computadora;
    }
    public String getNombre() {
        return nombre;
    }
    public int getDni() {
        return dni;
    }
}
```

```java
public static void main(String[] args) {
    //Composicion
    Computadora computadora = new Computadora("ASUS", 123456, "Zenbook Duo", "r9");
    System.out.println("El chipset de la computadora es: " + computadora.getPlacaMadre().getChipset()
    + " y su modelo es " + computadora.getPlacaMadre().getModelo());
    // Asoc. Bidireccional
    Usuario usuario = new Usuario("Christian", 40123456);
    computadora.setUsuario(usuario);
    System.out.println("El propietario de la computadora es "+ computadora.getUsuario().getNombre());
    System.out.println("Los datos de la computadora son: \nMarca: " + usuario.getComputadora().getMarca()
    + "\nNumero de Serie: " + usuario.getComputadora().getNumeroSerie());
}
```

```
El chipset de la computadora es: r9 y su modelo es Zenbook Duo
El propietario de la computadora es Christian
Los datos de la computadora son:
Marca: ASUS
Numero de Serie: 123456
BUILD SUCCESSFUL (total time: 0 seconds)
```

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




```java
public class Reserva {
    private String fecha;
    private String hora ;
    private Cliente cliente;  // Asociacion 1:1
    private Mesa mesa; // Agregacion
    // Constructor
    public Reserva(String fecha, String hora) {
        this.fecha = fecha;
        this.hora = hora;
    }
    public Reserva(String fecha, String hora, Mesa mesa) {
        this.fecha = fecha;
        this.hora = hora;
        this.mesa = mesa;
    }
    public Cliente getCliente() {
        return cliente;
    }
    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
    }
    public String getFecha() {
        return fecha;
    }
    public String getHora() {
        return hora;
    }
```

```
public void getMesa() {
        System.out.println("La Reserva de la fecha " + getFecha() +
                "; hora: " + getHora() + " tiene una mesa numero " + mesa.getNumero() +
                " de una capacidad de " + mesa.getCapacidad() + " personas");
}
@Override
public String toString() {
    return "Reserva{" +
            "\nfecha=" + fecha +
            ", \nhora=" + hora +
            ", \n" + cliente + '}';
}
```

```
public class Cliente {
    private String nombre;
    private int telefono;
    //private Reserva reserva;

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

    public String getNombre() {
        return nombre;
    }

    @Override
    public String toString() {
        return "Cliente{" + "\nnombre=" + nombre +
                " , \ntelefono=" + telefono + '}';
    }
}
```

```java
public class Mesa {
    private int numero;
    private int capacidad;
    //Constructor
    public Mesa(int numero, int capacidad) {
        this.numero = numero;
        this.capacidad = capacidad;
    }
    public int getNumero() {
        return numero;
    }
    public int getCapacidad() {
        return capacidad;
    }
}
```

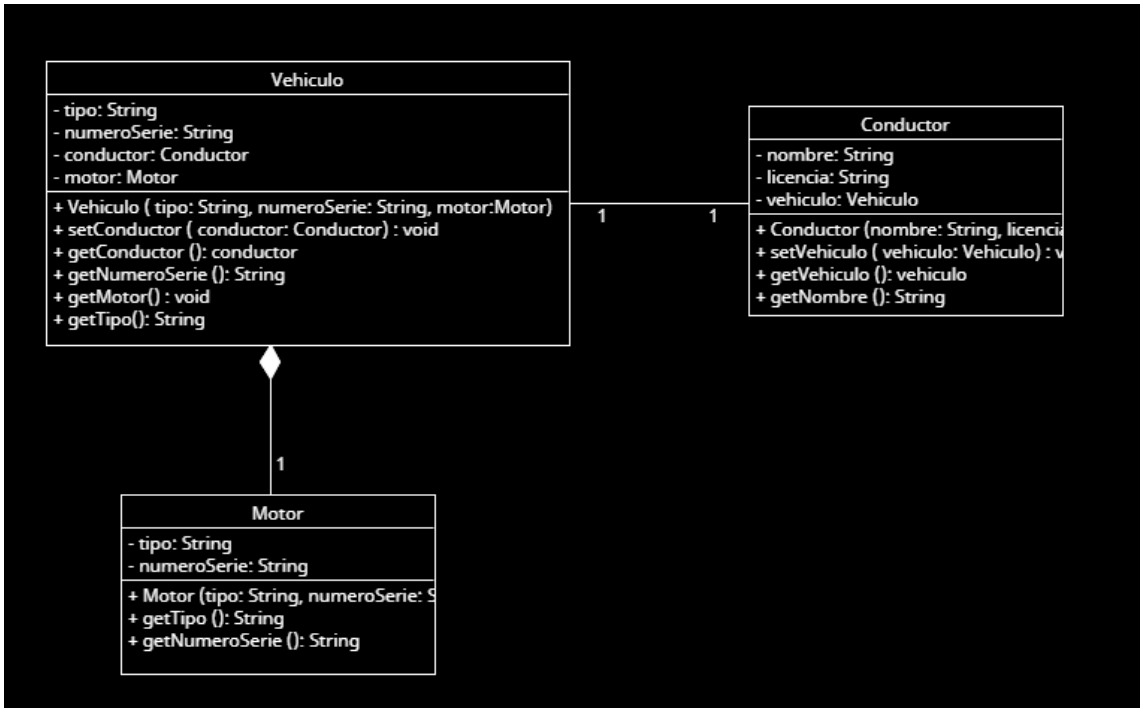
```java
package ejercicio6;
public class Ejercicio6 {
    public static void main(String[] args) {
        // Instancia de la clase Reserva - Cliente : Rel. Unidireccional
        Cliente c1 = new Cliente("Christian", 264123456);
        Reserva r1 = new Reserva("26/09/2025", "12:00");
        r1.setCliente(c1); // relacionamos ambas clases
        System.out.println(r1);

        // Instancia Clase  Reserva - Mesa : Agregación
        System.out.println("");
        Mesa m1 = new Mesa(10, 4);
        Reserva r2= new Reserva("26/09/2025", "14:00", m1);
        r2.getMesa(); // Utilizamos la relación
    }
}
```

```
Reserva{
fecha=26/09/2025,
hora=12:00,
Cliente{
nombre=Christian ,
telefono=264123456}}

La Reserva de la fecha 26/09/2025; hora: 14:00 tiene una mesa numero 10 de una capacidad de 4 personas
BUILD SUCCESSFUL (total time: 0 seconds)
```

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




```java
public class Vehiculo {
    private String tipo;
    private String numeroSerie ;
    private Conductor conductor; //Asoc Bidireccional
    private Motor motor; // Agregacion
    // CONSTRUCTOR
    public Vehiculo(String tipo, String numeroSerie) {
        this.tipo = tipo;
        this.numeroSerie = numeroSerie;
    }
    // 2º Constructor (Agregacion)
    public Vehiculo(String tipo, String numeroSerie, Motor motor) {
        this.tipo = tipo;
        this.numeroSerie = numeroSerie;
        this.motor = motor;
    }
```

```java
//Relacion
public void setConductor(Conductor conductor) {
    this.conductor = conductor;
    if (conductor != null && conductor.getVehiculo() != this ) {
        conductor.setVehiculo(this);
    }
}
public Conductor getConductor() {
    return conductor;
}
public String getNumeroSerie() {
    return numeroSerie;
}
public void getMotor() {
    System.out.println("El vehiculo tipo:" + getTipo() +" numero de serie: "+
            getNumeroSerie() +" tiene un motor: " + motor.getTipo() + " y un numero de serie: "+
            motor.getNumeroSerie());
}
   public String getTipo() {
       return tipo;
   }
  @Override
   public String toString() {
       return "{" + "tipo=" + tipo + ", numeroSerie=" + numeroSerie + ", conductor=" + conductor.getNombre() + '}';
   }
}
```

```java
public class Motor {
    private String tipo;
    private String numeroSerie ;
    //Constructor
    public Motor(String tipo, String numeroSerie) {
        this.tipo = tipo;
        this.numeroSerie = numeroSerie;
    }
    public String getTipo() {
        return tipo;
    }
    public String getNumeroSerie() {
        return numeroSerie;
    }
}
```

```java
public class Conductor {
    private String nombre;
    private String licencia;
    private Vehiculo vehiculo; // R. Bidireccional
    // CONSTRUCTOR
    public Conductor(String nombre, String licencia) {
        this.nombre = nombre;
        this.licencia = licencia;
    }
    // Relacion
    public void setVehiculo(Vehiculo vehiculo) {
        this.vehiculo = vehiculo;
        if (vehiculo != null && vehiculo.getConductor() != this ) {
            vehiculo.setConductor(this);
        }
    }
    public Vehiculo getVehiculo() {
        return vehiculo;
    }
    public String getNombre() {
        return nombre;
    }
     @Override
    public String toString() {
        return "{" + "nombre=" + nombre + ", licencia=" + licencia +", vehiculo=" +
                vehiculo.getNumeroSerie() + '}';
    }
}
```

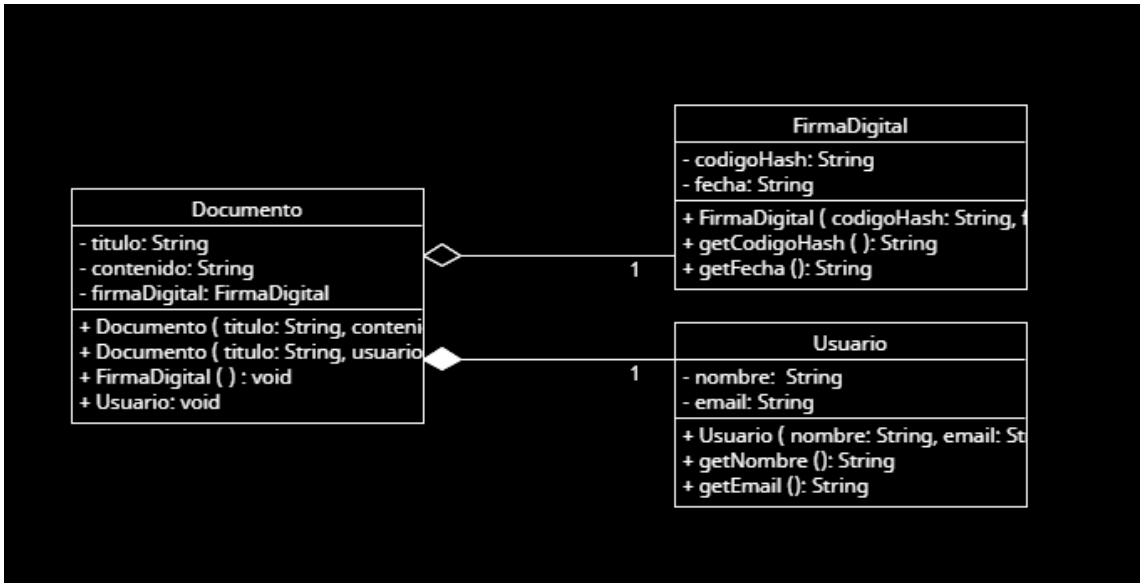
```java
package ejercicio7;
public class Ejercicio7 {
    public static void main(String[] args) {
       // Instancia R. Bidireccional VEHÍCULO - CONDUCTOR
       Vehiculo v1 = new Vehiculo("auto", "ABC123");
       Conductor c1 = new Conductor("Christian", "AB_autos");
       v1.setConductor(c1); // Establecer la relación bidireccional
       // Mostrar el estado de cada objeto
       System.out.println("Vehiculo -> " + v1);
       System.out.println("Conductor -> " + c1 + "\n");
       // Verificar consistencia: cada uno conoce al otro
       System.out.println("Conductor del vehiculo: " + v1.getConductor().getNombre());
       System.out.println("Vehiculo del conductor: " + c1.getVehiculo().getNumeroSerie() + "\n");

       // Instancia R. Agregacion VEHICULO - MOTOR
       Motor m1 = new Motor("1.8", "JKL_987");
       Vehiculo v2= new Vehiculo("Camioneta", "FGH_456", m1);
       v2.getMotor();
    }
}
```

```
Vehiculo -> {tipo=auto, numeroSerie=ABC123, conductor=Christian}
Conductor -> {nombre=Christian, licencia=AB_autos, vehiculo=ABC123}

Conductor del vehiculo: Christian
Vehiculo del conductor: ABC123

El vehiculo tipo:Camioneta numero de serie: FGH_456 tiene un motor: 1.8 y un numero de serie: JKL_987
BUILD SUCCESSFUL (total time: 0 seconds)
```

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




```java
public class Documento {
    private String titulo;
    private String contenido;
    private FirmaDigital firmaDigital; // Composicion 1:1
    private Usuario usuario; // Agregacion 1:1
    // CONSTRUCTOR
    public Documento(String titulo, String codigoHash, String fecha) {
        this.titulo = titulo;
        this.contenido = contenido;
        this.firmaDigital = new FirmaDigital(codigoHash, fecha);
    }
    // CONSTRUCTOR 2
    public Documento(String titulo, Usuario usuario) {
        this.titulo = titulo;
        this.usuario = usuario;
    }
    public void FirmaDigital() {
            System.out.println("La firma digital del libro es: " +
                    firmaDigital.getCodigoHash() + "\n y su fecha es: " +
                    firmaDigital.getFecha() + "\n" );
    }
    public void Usuario (){
        System.out.println("El nombre del usuario es: " + usuario.getNombre() +
                "\n y su email es: " + usuario.getEmail());
    }
```

```java
public class FirmaDigital {
    private String codigoHash;
    private String fecha ;
    //CONSTRUCTOR
    public FirmaDigital(String codigoHash, String fecha) {
        this.codigoHash = codigoHash;
        this.fecha = fecha;
    }
    public String getCodigoHash() {
        return codigoHash;
    }
    public String getFecha() {
        return fecha;
    }
```

```java
public class Usuario {
    private String  nombre;
    private String  email ;
    // CONSTRUCTOR
    public Usuario(String nombre, String email) {
        this.nombre = nombre;
        this.email = email;
    }
    public String getNombre() {
        return nombre;
    }
    public String getEmail() {
        return email;
    }
```

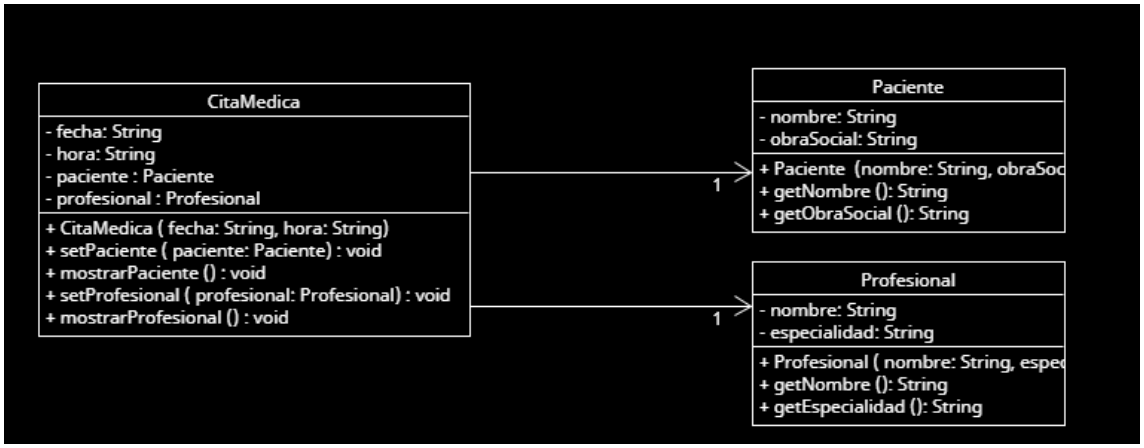
```java
public static void main(String[] args) {
    // COMPOSICION:
    Documento Documento1 = new Documento("Programacio II", "ABC_567", "27/09/2025");
    Documento1.FirmaDigital();
    // AGREGACION
    Usuario U1 = new Usuario("Christian", "Christian@gmail.com");
    Documento Documento2= new Documento("Lbro 2", U1);
    Documento2.Usuario();
}
```

```
 La firma digital del libro es: ABC_567
  y su fecha es: 27/09/2025

 El nombre del usuario es: Christian
  y su email es: Christian@gmail.com
 BUILD SUCCESSFUL (total time: 0 seconds)
```

9. CitaMédica - Paciente - Profesional

a. Asociación unidireccional: **CitaMédica → Paciente,**
b. Asociación unidirecciona: **CitaMédica → Profesional**

Clases y atributos:
i. CitaMédica: fecha, hora
ii. Paciente: nombre, obraSocial
iii. Profesional: nombre, especialidad




```java
public class CitaMedica {
    private String fecha;
    private String hora ;
    private Paciente paciente; // Asoc. Unidireccional 1:1
    private Profesional profesional; // Asoc. Unidireccional 1:1
    //Constructor
    public CitaMedica(String fecha, String hora) {
        this.fecha = fecha;
        this.hora = hora;
    }
    public void setPaciente(Paciente paciente) {
        this.paciente = paciente;
    }
    public void mostrarPaciente() {
        if (paciente != null) {
            System.out.println("El paciente cargado es: " + paciente.getNombre() +
                    " y su obra sociales es: " + paciente.getObraSocial() + "\n");
        } else {
            System.out.println("No se ha cargado paciente");
        }
    }
    public void setProfesional(Profesional profesional) {
        this.profesional = profesional;
    }
    public void MostrarProfesional (){
        if (profesional != null) {
            System.out.println("El profesional cargado es: " + profesional.getNombre() +
                    " y su especialidad es: " + profesional.getEspecialidad() + "\n");
        } else {
            System.out.println("No se ha cargado profesional");
        }
    }
}
```

```java
public class Paciente {
    private String nombre;
    private String obraSocial ;
    //Constructor
    public Paciente(String nombre, String obraSocial) {
        this.nombre = nombre;
        this.obraSocial = obraSocial;
    }
    public String getNombre() {
        return nombre;
    }
    public String getObraSocial() {
        return obraSocial;
    }
}
public class Profesional {
    private String nombre;
    private String especialidad ;
    // Constructor
    public Profesional(String nombre, String especialidad) {
        this.nombre = nombre;
        this.especialidad = especialidad;
    }
    public String getNombre() {
        return nombre;
    }
    public String getEspecialidad() {
        return especialidad;
    }
}
```

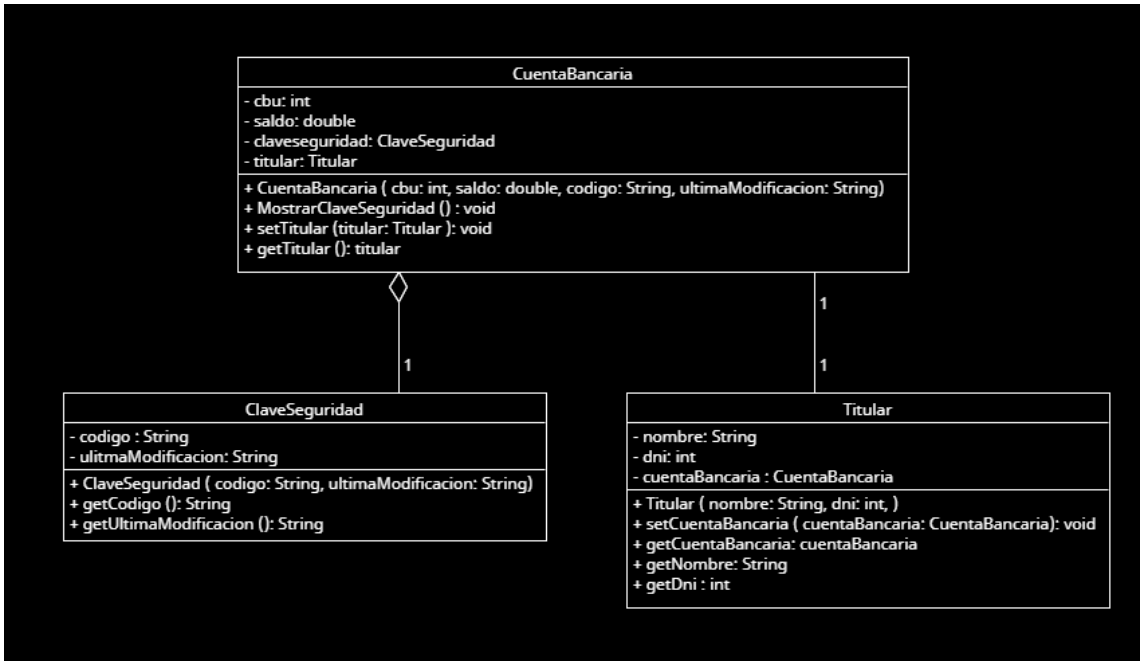
```java
public class Ejercicio9 {
    public static void main(String[] args) {
        // Relacion Unidireccional CitaMedica -> Paciente
        CitaMedica Cita1 = new CitaMedica("28/09/2025", "12:00");
        Paciente paciente1 = new Paciente("Christian", "OSDE");
        Cita1.setPaciente(paciente1);           //Se establece la relacion
        Cita1.mostrarPaciente();
        // Relacion Unidireccional CitaMedica -> Profesional
        Profesional profesional1 = new Profesional("Emmanuel", "Neurocirujano");
        Cita1.setProfesional(profesional1);     //Se establece la relacion
        Cita1.MostrarProfesional();
    }
}
```

```
El paciente cargado es: Christian y su obra sociales es: OSDE

El profesional cargado es: Emmanuel y su especialidad es: Neurocirujano

BUILD SUCCESSFUL (total time: 0 seconds)
```

10. CuentaBancaria - ClaveSeguridad - Titular

a. Composición: **CuentaBancaria → ClaveSeguridad**
b. Asociación bidireccional: **CuentaBancaria ↔ Titular**

Clases y atributos:
i. CuentaBancaria: cbu, saldo
ii. ClaveSeguridad: codigo, ultimaModificacion
iii. Titular: nombre, dni.




```java
public class CuentaBancaria {
    private int cbu;
    private double saldo;
    private ClaveSeguridad claveseguridad; // Composicion 1:1
    private Titular titular; // Rel. Bidireccional 1:1
    // Constructor
    public CuentaBancaria(int cbu, double saldo, String codigo, String ultimaModificaion) {
        this.cbu = cbu;
        this.saldo = saldo;
        this.claveseguridad = new ClaveSeguridad(codigo, ultimaModificaion);
    }
    public void MostrarClaveSeguridad(){
        System.out.println("La clave de seguridad es: "+ claveseguridad.getCodigo()
                + " y su ultima modificacion es: " + claveseguridad.getUltimaModificacion() + "\n");
    }
    public void setTitular(Titular titular) {
        this.titular = titular;
        if (titular != null &&  titular.getCuentabancaria() != this ) {
            titular.setCuentabancaria(this);
        }
    }
     public Titular getTitular() {
         return titular;
     }
     @Override
     public String toString() {
         return "CuentaBancaria{" + "cbu=" + cbu + ", saldo=" + saldo + '}';
     }
}
```

```java
public class ClaveSeguridad {
    private String codigo;
    private String ultimaModificacion;
    // Constructor
    public ClaveSeguridad(String codigo, String ultimaModificacion) {
        this.codigo = codigo;
        this.ultimaModificacion = ultimaModificacion;
    }
    public String getCodigo() {
        return codigo;
    }
    public String getUltimaModificacion() {
        return ultimaModificacion;
    }
}
```

```java
public class Titular {
    private String nombre;
    private int dni ;
    private CuentaBancaria cuentabancaria;
    // Constructor
    public Titular(String nombre, int 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 cuentabancaria;
    }
    public String getNombre() {
        return nombre;
    }
    public int getDni() {
        return dni;
    }
    @Override
    public String toString() {
        return "Titular{" + "nombre=" + nombre + ", dni=" + dni + '}';
    }
}
```

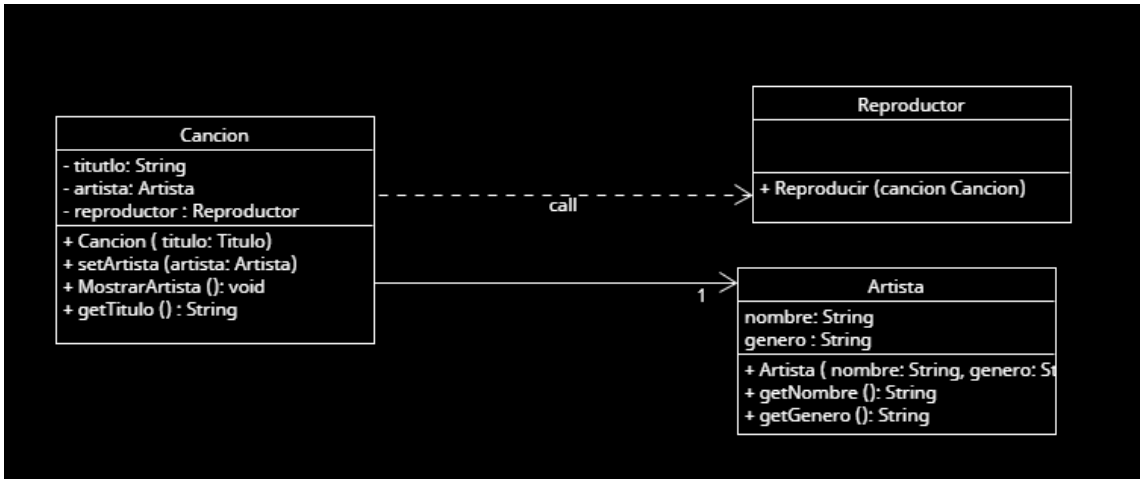
```java
package ejercicio10;

public class Ejercicio10 {

    public static void main(String[] args) {
        // Composicion CuentaBancaria -> ClaveSeguridad
       CuentaBancaria Cuenta1 = new CuentaBancaria(250000453, 400, "ABC_123", "28/09/2025");
       Cuenta1.MostrarClaveSeguridad();
       // Bidireccional CuentaBancaria -> Titular
       Titular titular1 = new Titular("Christian", 4026531);
       Cuenta1.setTitular(titular1); //Establecemos la relacion Bidireccional
       System.out.println("Cuenta Bancaria : " + Cuenta1.getTitular());
       System.out.println("Titular : " + Cuenta1.getTitular().getNombre());
       System.out.println("Cuenta asociada a " + titular1.getNombre() + "\n");

    }

}
```

```
La clave de seguridad es: ABC_123 y su ultima modificacion es: 28/09/2025

Cuenta Bancaria : Titular{nombre=Christian, dni=4026531}
Titular : Christian
Cuenta asociada Christian
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
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)




```java
public class Cancion {
    private String titulo;
    private Artista artista; // Asociacion 1:1
    private Reproductor reproductor;
    //Constructor
    public Cancion(String titulo) {
        this.titulo = titulo;
    }
    public void setArtista(Artista artista) {
        this.artista = artista;
    }
    public void MostrarArtista(){
        if (artista != null) {
            System.out.println("El artista de la cancion \" " + getTitulo() +"\", es "
                    + artista.getNombre()
                    + " y su genero es " + artista.getGenero() + "\n");
        } else {
            System.out.println("No se ha cargado un artista");
        }
    }
    public String getTitulo() {
        return titulo;
    }
}
```

```java
public class Artista {
    private String nombre;
    private String genero;
    //Constructor
    public Artista(String nombre, String genero) {
        this.nombre = nombre;
        this.genero = genero;
    }
    public String getNombre() {
        return nombre;
    }
    public String getGenero() {
        return genero;
    }
```

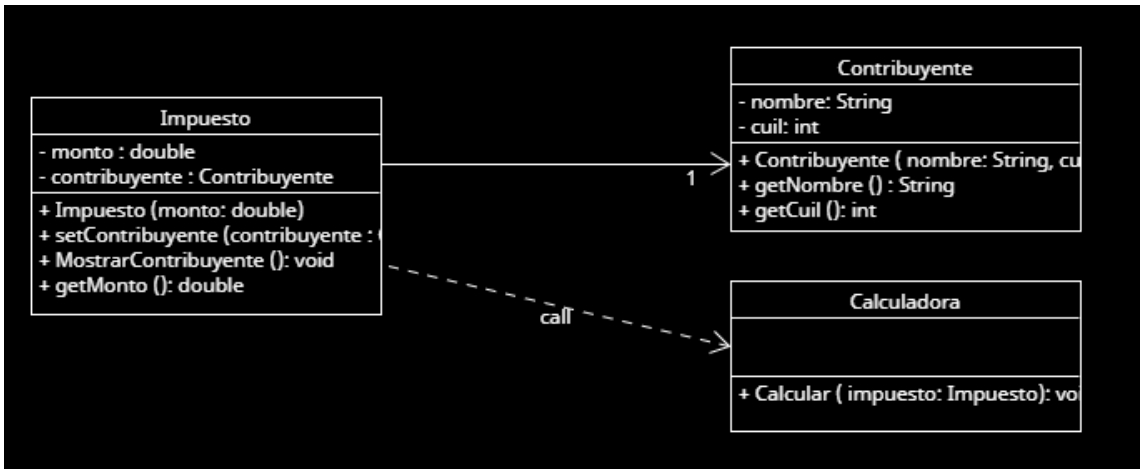
```java
public class Reproductor {
    public static void Reproducir(Cancion cancion){
        System.out.println("Reproduciento: " + cancion.getTitulo());
    }
}
```

```java
public class Ejercicio11 {
    public static void main(String[] args) {
        Cancion cancion1 = new Cancion("Ich tu dir weh");
        Artista artista1= new Artista("Rammstein", "Metal industrial");
        cancion1.setArtista(artista1); //Establecemos la relación
        cancion1.MostrarArtista();
        Reproductor.Reproducir(cancion1); // Reproducimios la cancion
    }
}
```

```
El artista de la cancion " Ich tu dir weh", es Rammstein y su genero es Metal industrial

Reproduciento: Ich tu dir weh
BUILD SUCCESSFUL (total time: 0 seconds)
```

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)




```java
public class Impuesto {
    private double monto;
    private Contribuyente contribuyente; // Asociacion 1:1
    // Cosntructor
    public Impuesto(double monto) {
        this.monto = monto;
    }
    public void setContribuyente(Contribuyente contribuyente) {
        this.contribuyente = contribuyente;
    }
    public void MostrarContribuyente (){
        if (contribuyente != null) {
            System.out.println("El contribuyente es: " + contribuyente.getNombre()
            +" y su cuil es: " + contribuyente.getCuil());
        } else {
            System.out.println("No se ha cargago contribuyente");
        }
    }
    public double getMonto() {
        return monto;
    }
}
```

```java
public class Contribuyente {
    private String nombre;
    private int cuil;
    //Constructor
    public Contribuyente(String nombre, int cuil) {
        this.nombre = nombre;
        this.cuil = cuil;
    }
    public String getNombre() {
        return nombre;
    }
    public int getCuil() {
        return cuil;
    }
```

```java
public class Calculadora {
    public static void Calcular (Impuesto impuesto){
        System.out.println("El impuesto es: " + impuesto.getMonto());
    }
}
```

```java
public static void main(String[] args) {
    Contribuyente C1 = new Contribuyente("Christian", 123124546);
    Impuesto I1= new Impuesto(100);
    I1.setContribuyente(C1); //establecemos la relacion
    I1.MostrarContribuyente();
    Calculadora.Calcular(I1);
}
```

```
El contribuyente es: Christian y su cuil es: 123124546
El impuesto es: 100.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

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

a. Asociación unidireccional: **CódigoQR → Usuario**

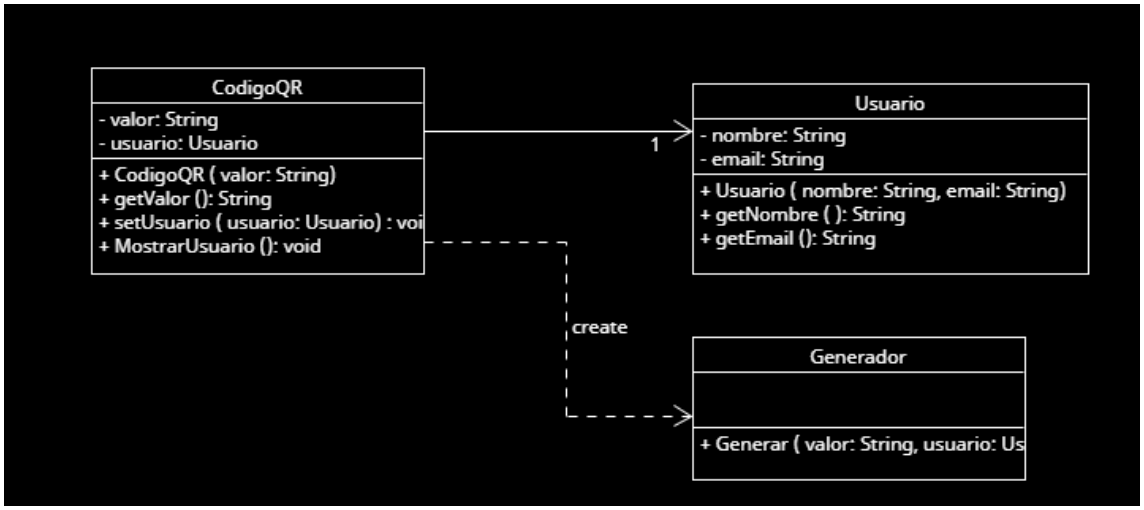
b. Dependencia de creación: **GeneradorQR.generar(String, Usuario)**

Clases y atributos:

i. CodigoQR: valor.

ii. Usuario: nombre, email.

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




```java
public class Usuario {
    private String nombre;
    private String email;
    // Constructor
    public Usuario(String nombre, String email) {
        this.nombre = nombre;
        this.email = email;
    }
    public String getNombre() {
        return nombre;
    }
    public String getEmail() {
        return email;
    }
}
```

```java
public class CodigoQR {
    private String valor;
    private Usuario usuario; // Asociacion
    // Constructor
    public CodigoQR(String valor) {
        this.valor = valor;
    }
    public String getValor() {
        return valor;
    }
    public void setUsuario(Usuario usuario) {
        this.usuario = usuario;
    }
    public void MostrarUsuario (){
        if (usuario != null) {
            System.out.println("El usuario asociado es " + usuario.getNombre());
        } else {
            System.out.println("No se ha cargado un usuario");
        }
    }
    @Override
    public String toString() {
        return " CodigoQR : {" + "valor=" + valor + '}';
    }
}
```

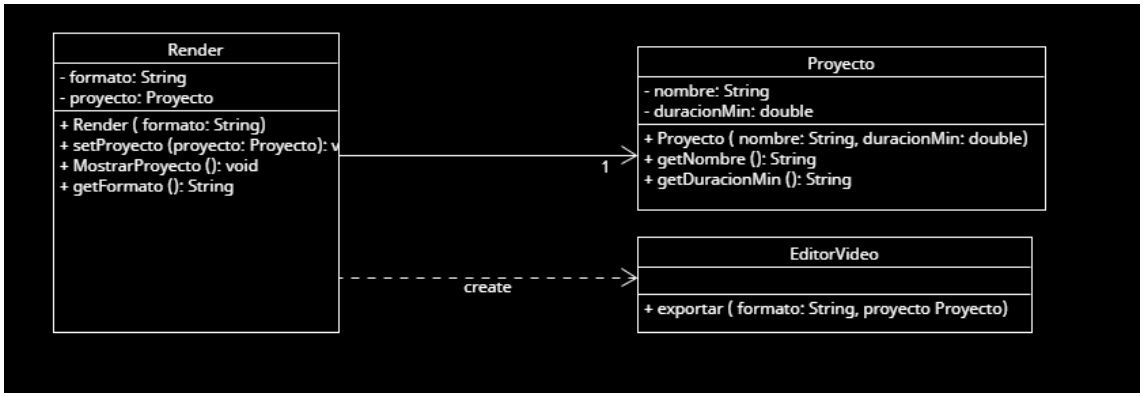
```java
public class Generador {
    public static void generar (String valor , Usuario usuario ) {
        CodigoQR codigo = new CodigoQR(valor);
        codigo.setUsuario(usuario);
        codigo.MostrarUsuario();
        System.out.println(codigo.getValor());
    }
```

```java
public static void main(String[] args) {
    Usuario usuario = new Usuario("Christian", "qwerty@gmail.com");
    CodigoQR codigo = new CodigoQR("Codigo 1 ");
    codigo.setUsuario(usuario);// Establecemos relacion
    codigo.MostrarUsuario();
    System.out.println(codigo.getValor());
    Usuario usuario2= new Usuario("Emmanuel", "asdfklj@gmail");
    Generador.generar("Codigo2", usuario2);

}
```

```
El usuario asociado es Christian
Codigo 1
El usuario asociado es Emmanuel
Codigo2
BUILD SUCCESSFUL (total time: 0 seconds)
```

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

Clases y atributos:
i. Render: formato.
ii. Proyecto: nombre, duracionMin.
iii. EditorVideo->método: void exportar(String formato, Proyecto proyecto)




```java
public class Render {
    private String formato;
    private Proyecto proyecto;
    // Constructor
    public Render(String formato) {
        this.formato = formato;
    }
    public void setProyecto(Proyecto proyecto) {
        this.proyecto = proyecto;
    }
    public void MostrarProyecto(){
        System.out.println("El proyecto se llama " + proyecto.getNombre()
        + " y tiene una duracion de: " +proyecto.getDuracionMin() + " min \n"
        + "renderizado en formato. " +getFormato() );
    }
    public String getFormato() {
        return formato;
    }
}
```

```java
public class Proyecto {
    private String nombre;
    private double duracionMin ;
    //Constructor
    public Proyecto(String nombre, double duracionMin) {
        this.nombre = nombre;
        this.duracionMin = duracionMin;
    }
    public String getNombre() {
        return nombre;
    }
    public double getDuracionMin() {
        return duracionMin;
    }
```

```java
public class EditorVideo {
    public static void exportar (String formato, Proyecto proyecto){
        Render render1 = new Render(formato);
        render1.setProyecto(proyecto);
        render1.MostrarProyecto();
    }
```

```
El proyecto se llama Christian y tiene una duracion de: 142.0 min
renderizado en formato. mp4
El proyecto se llama Emmanuel y tiene una duracion de: 160.0 min
renderizado en formato. mkv
```