

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

PROGRAMACIÓN II

Trabajo Práctico 5: Relaciones UML 1 a 1

Alumno :

- González Federico

Profesor :

- Enferrel Ariel

Tutor :

- Ferro Tomás

GitHub:

- <https://github.com/fedeglz/UTN-TUPaD-P2.git>

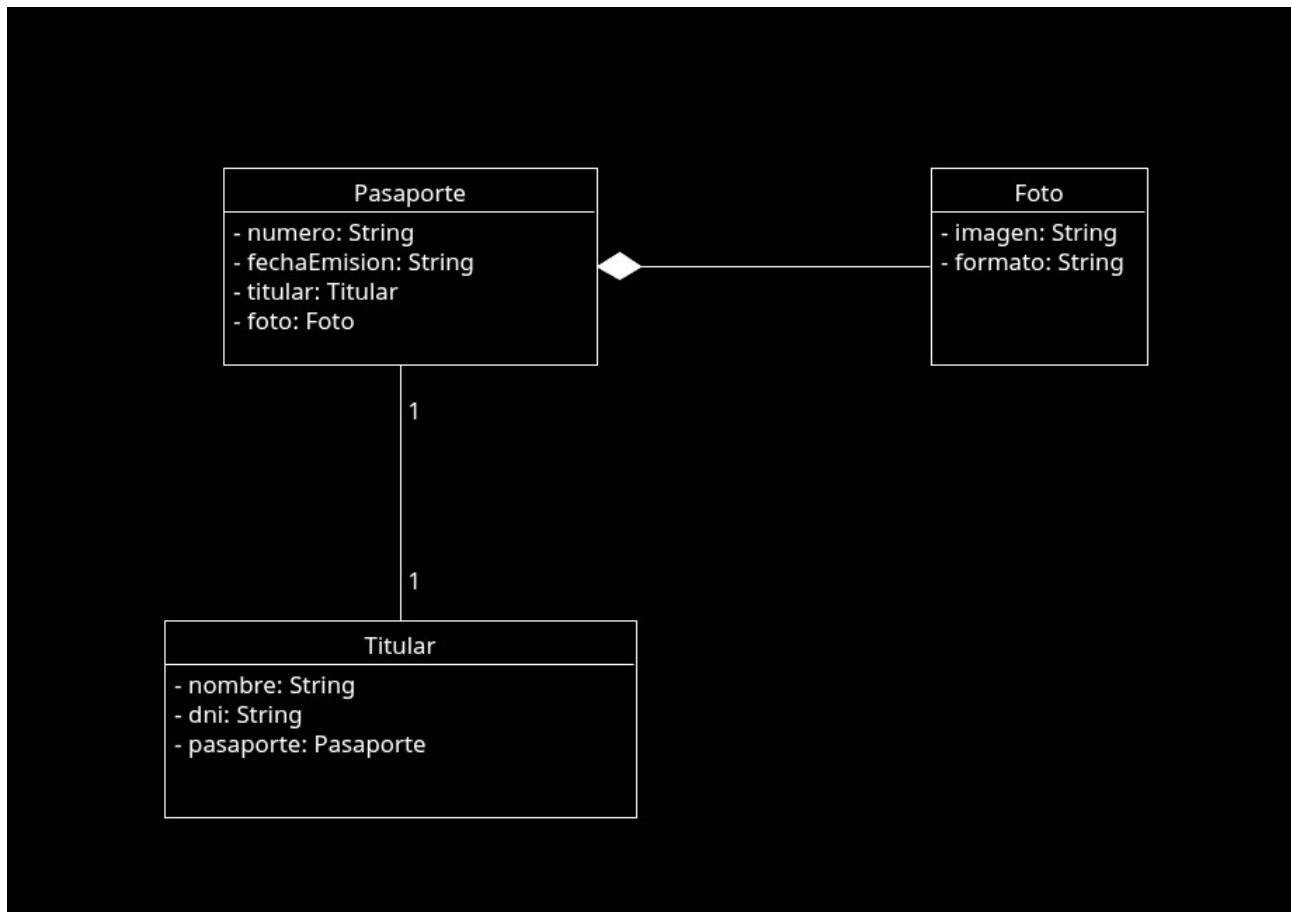
Ejercicios de Relaciones 1 a 1

1. Pasaporte - Foto - Titular

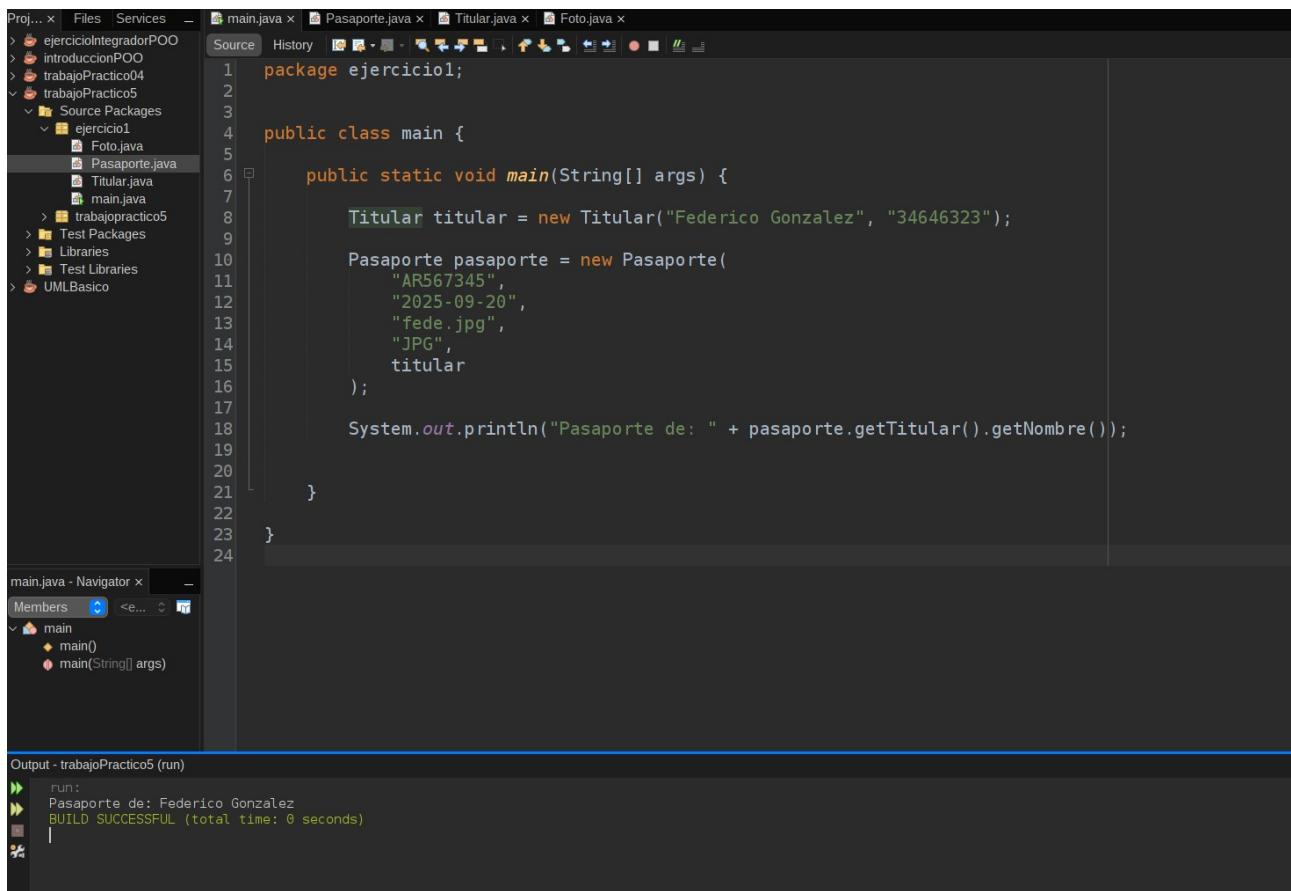
- a. Composición: Pasaporte → Foto
- b. Asociación bidireccional: Pasaporte ↔ Titular

Clases y atributos:

- i. Pasaporte: numero, fechaEmision
- ii. Foto: imagen, formato
- iii. Titular: nombre, dni



TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

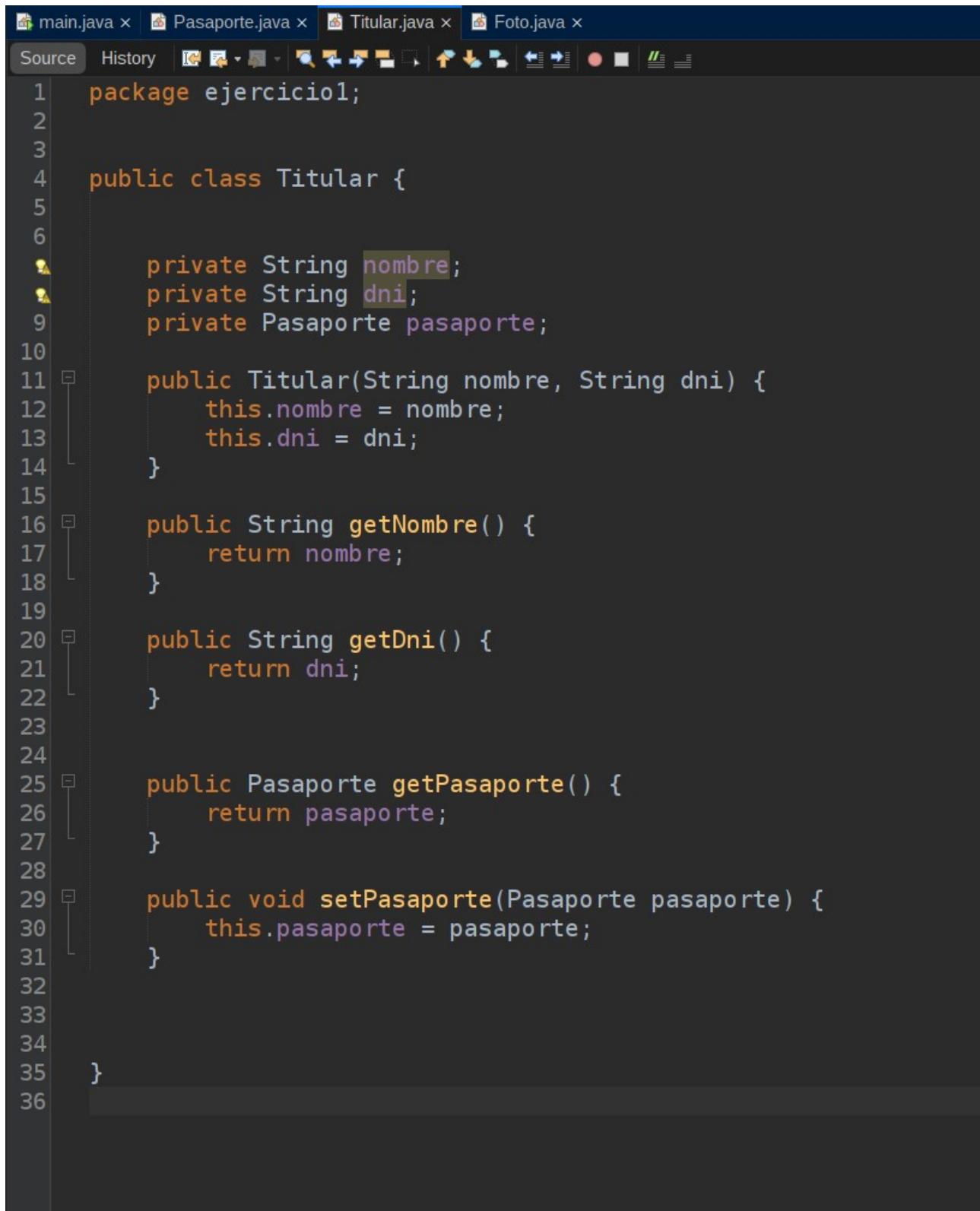


The screenshot shows an IDE interface with the following components:

- Project Explorer (Left):** Shows the project structure with packages like ejercicioIntegradorPOO, introduccionPOO, trabajoPractico04, trabajoPractico5, and UMLBasico. Inside the ejercicio1 package, there are Foto.java, Pasaporte.java, Titular.java, and main.java.
- Code Editor (Top Right):** Displays the main.java file containing Java code to create a Titular object and a Pasaporte object, then print the holder's name.
- Navigator (Bottom Left):** Shows the members of the main class: main() and main(String[] args).
- Output (Bottom):** Shows the run output: "run: Pasaporte de: Federico Gonzalez BUILD SUCCESSFUL (total time: 0 seconds)".

```
package ejercicio1;

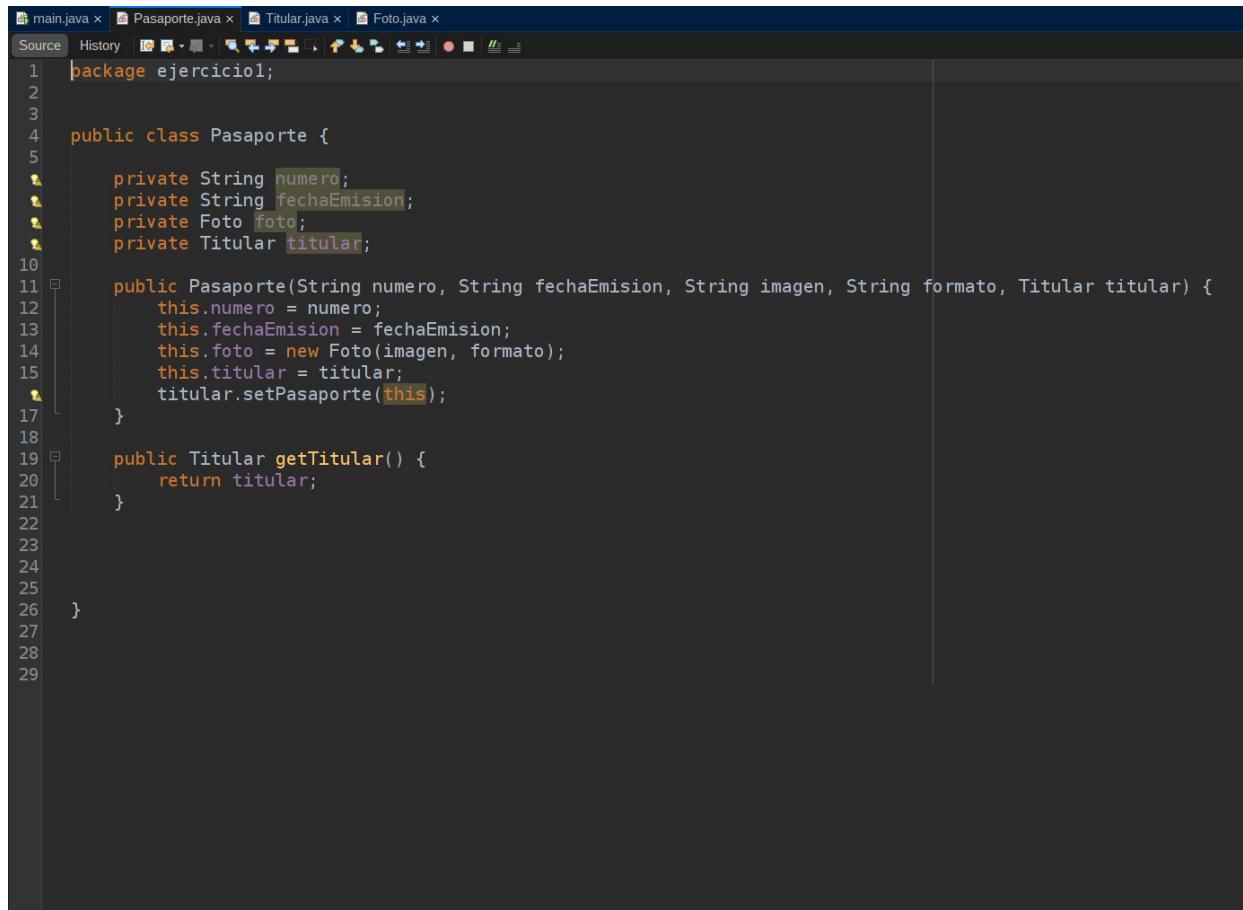
public class main {
    public static void main(String[] args) {
        Titular titular = new Titular("Federico Gonzalez", "34646323");
        Pasaporte pasaporte = new Pasaporte(
            "AR567345",
            "2025-09-20",
            "fede.jpg",
            "JPG",
            titular
        );
        System.out.println("Pasaporte de: " + pasaporte.getTitular().getNombre());
    }
}
```

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

The screenshot shows a Java code editor with the file `Titular.java` open. The code defines a class `Titular` with private attributes `nombre` and `dni`, and a private attribute `Pasaporte pasaporte`. It includes a constructor that takes `nombre` and `dni` as parameters and initializes the attributes. It also includes methods to get `nombre`, `dni`, and `pasaporte`, and a method to set `pasaporte`.

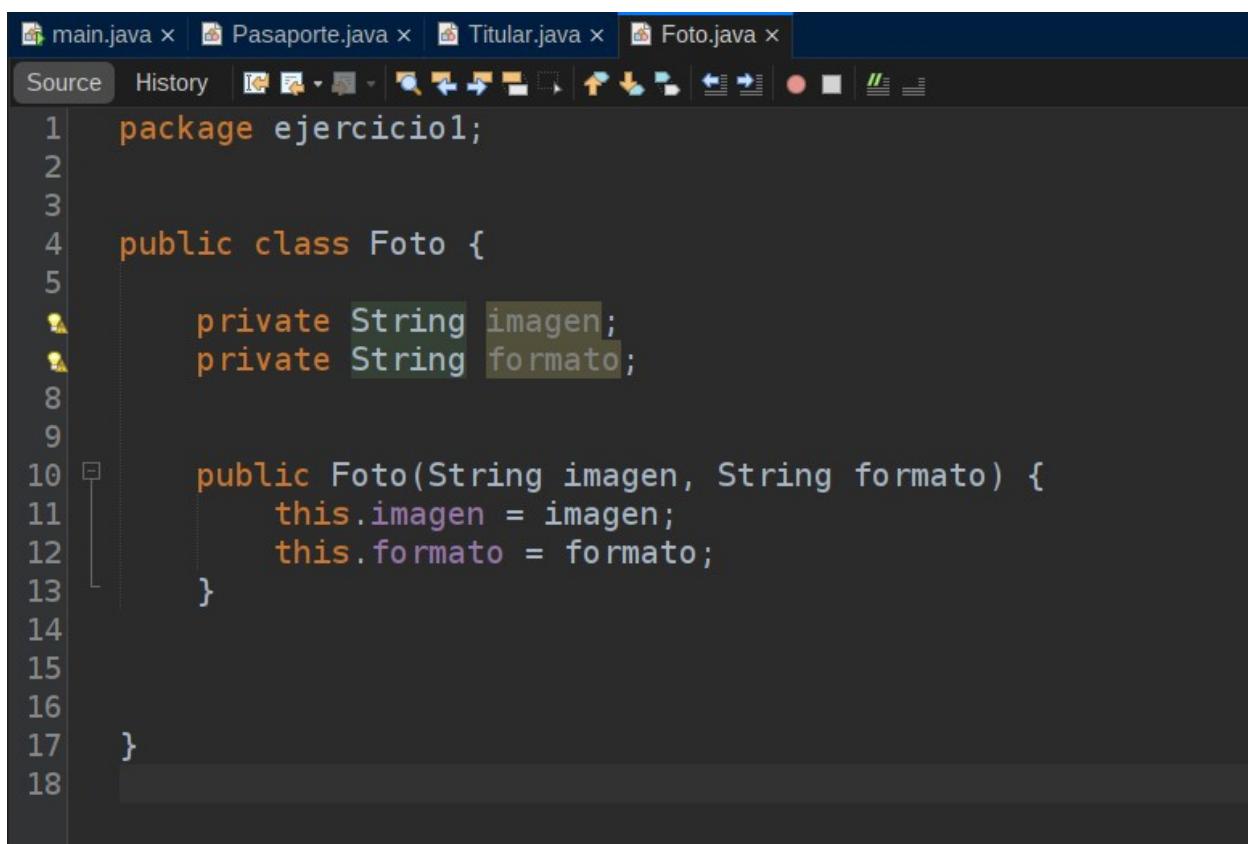
```
main.java x Pasaporte.java x Titular.java x Foto.java x
Source History | I C F D S M L G H P R E V N O P Z X Y
1 package ejercicio1;
2
3
4 public class Titular {
5
6
7     private String nombre;
8     private String dni;
9     private Pasaporte pasaporte;
10
11    public Titular(String nombre, String dni) {
12        this.nombre = nombre;
13        this.dni = dni;
14    }
15
16    public String getNombre() {
17        return nombre;
18    }
19
20    public String getDni() {
21        return dni;
22    }
23
24
25    public Pasaporte getPasaporte() {
26        return pasaporte;
27    }
28
29    public void setPasaporte(Pasaporte pasaporte) {
30        this.pasaporte = pasaporte;
31    }
32
33
34
35}
36
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



```

1 package ejercicio1;
2
3 public class Pasaporte {
4     private String numero;
5     private String fechaEmision;
6     private Foto foto;
7     private Titular titular;
8
9     public Pasaporte(String numero, String fechaEmision, String imagen, String formato, Titular titular) {
10        this.numero = numero;
11        this.fechaEmision = fechaEmision;
12        this.foto = new Foto(imagen, formato);
13        this.titular = titular;
14        titular.setPasaporte(this);
15    }
16
17    public Titular getTitular() {
18        return titular;
19    }
20
21 }
22
23
24
25
26 }
27
28
29
  
```



```

1 package ejercicio1;
2
3
4 public class Foto {
5
6     private String imagen;
7     private String formato;
8
9
10    public Foto(String imagen, String formato) {
11        this.imagen = imagen;
12        this.formato = formato;
13    }
14
15
16
17 }
18
  
```

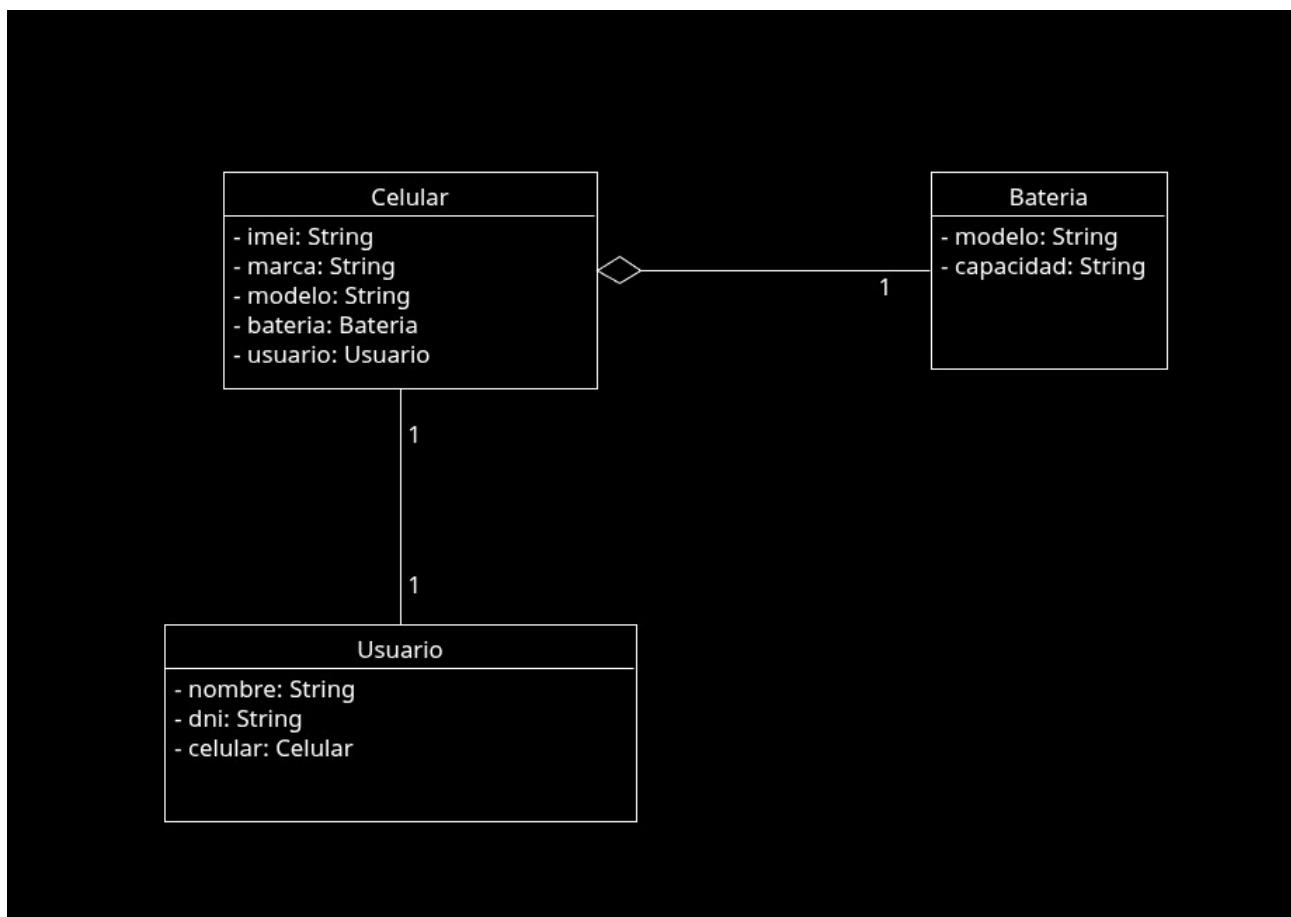
TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

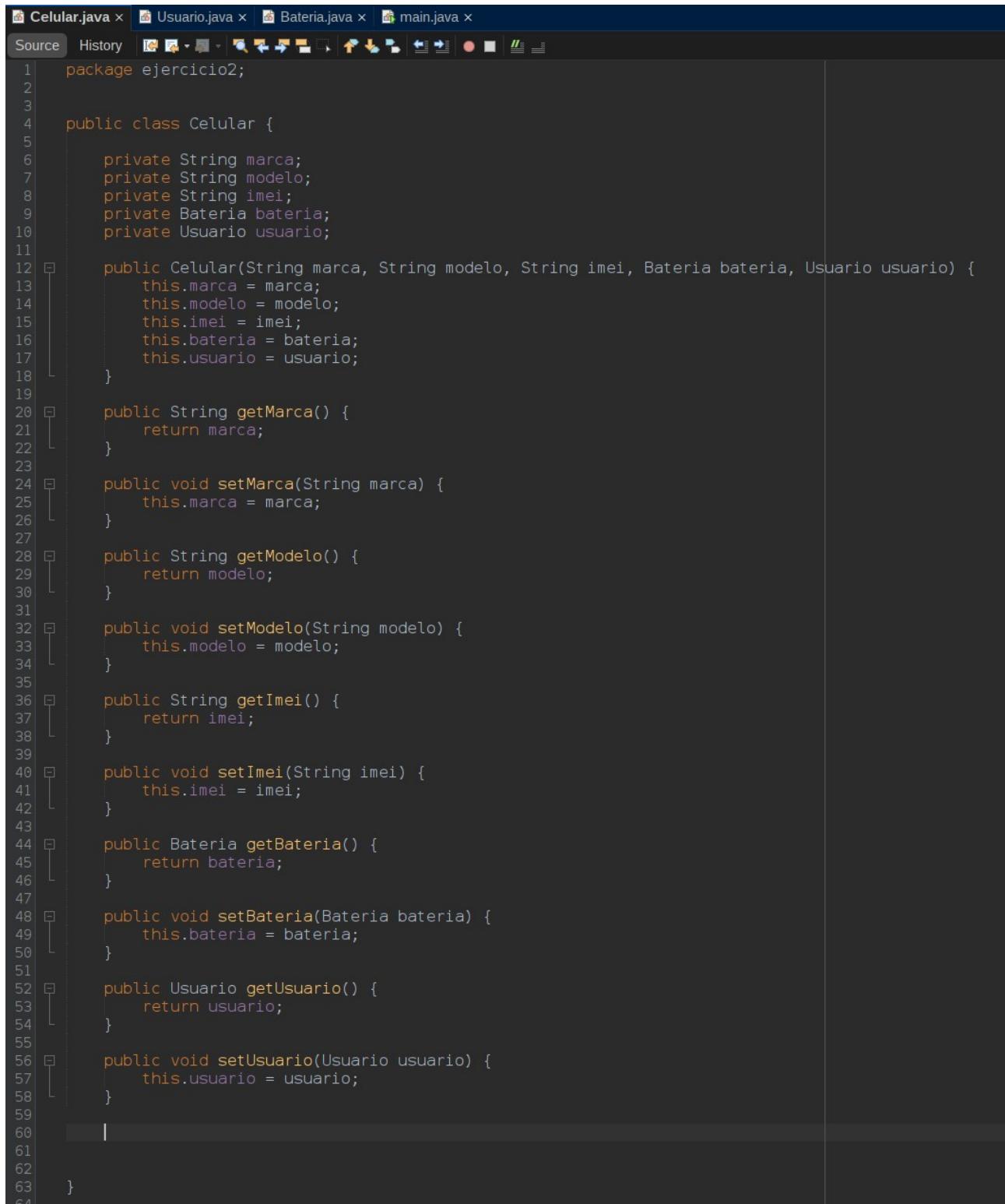
2. Celular - Batería - Usuario

- 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



TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

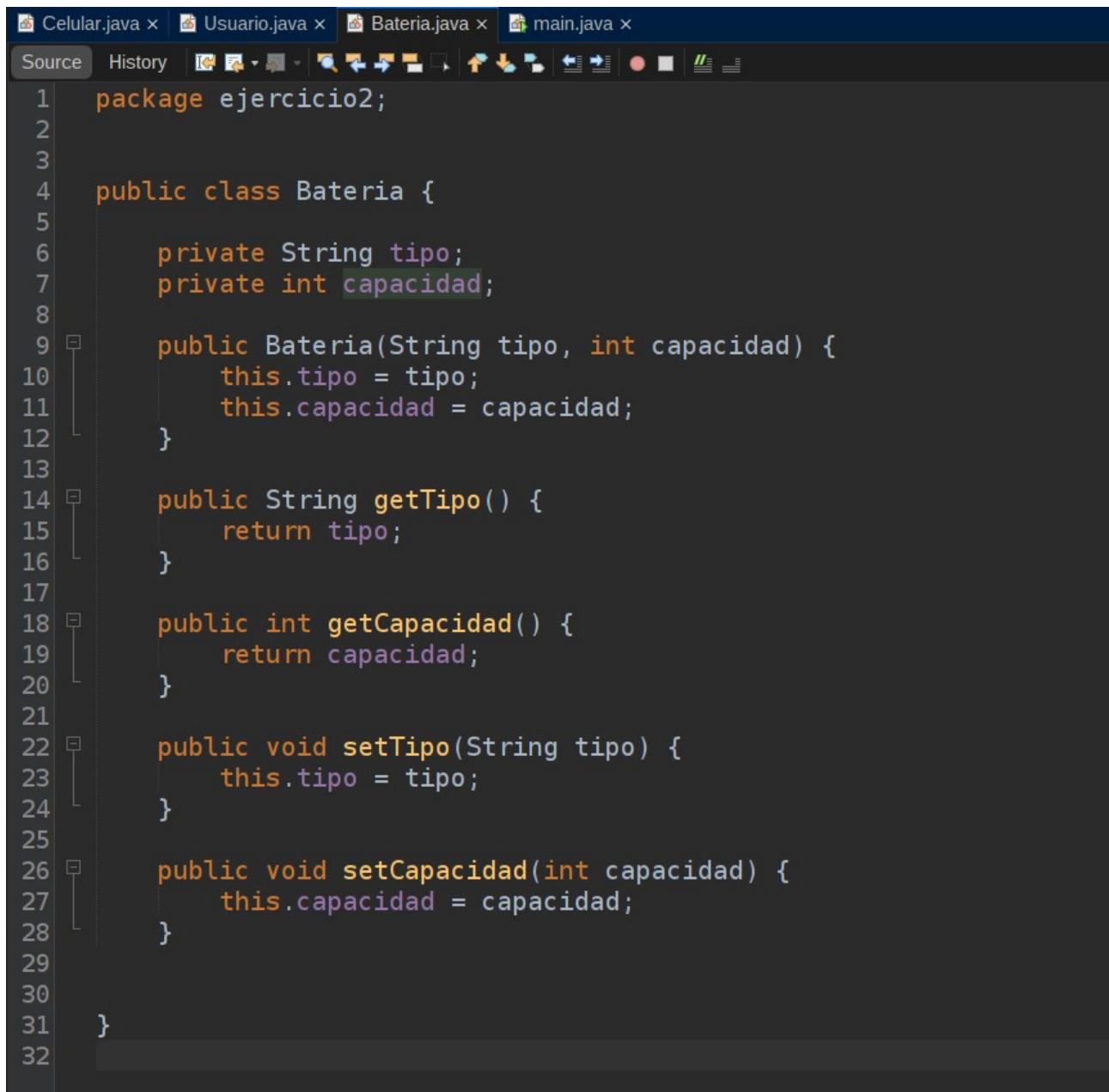
The screenshot shows a Java code editor with the file `Celular.java` open. The code defines a class `Celular` with private fields for marca, modelo, imei, Bateria, and Usuario. It includes a constructor, getters, and setters for these fields.

```
1 package ejercicio2;
2
3 public class Celular {
4
5     private String marca;
6     private String modelo;
7     private String imei;
8     private Bateria bateria;
9     private Usuario usuario;
10
11     public Celular(String marca, String modelo, String imei, Bateria bateria, Usuario usuario) {
12         this.marca = marca;
13         this.modelo = modelo;
14         this.imei = imei;
15         this.bateria = bateria;
16         this.usuario = usuario;
17     }
18
19     public String getMarca() {
20         return marca;
21     }
22
23     public void setMarca(String marca) {
24         this.marca = marca;
25     }
26
27     public String getModelo() {
28         return modelo;
29     }
30
31     public void setModelo(String modelo) {
32         this.modelo = modelo;
33     }
34
35     public String getImei() {
36         return imei;
37     }
38
39     public void setImei(String imei) {
40         this.imei = imei;
41     }
42
43     public Bateria getBateria() {
44         return bateria;
45     }
46
47     public void setBateria(Bateria bateria) {
48         this.bateria = bateria;
49     }
50
51     public Usuario getUsuario() {
52         return usuario;
53     }
54
55     public void setUsuario(Usuario usuario) {
56         this.usuario = usuario;
57     }
58
59     |
60
61
62
63 }
64 }
```

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

The screenshot shows a Java code editor with the file `Usuario.java` open. The code defines a class `Usuario` with private attributes `nombre`, `dni`, and `celular`. It includes a constructor, three getters, and three setters.

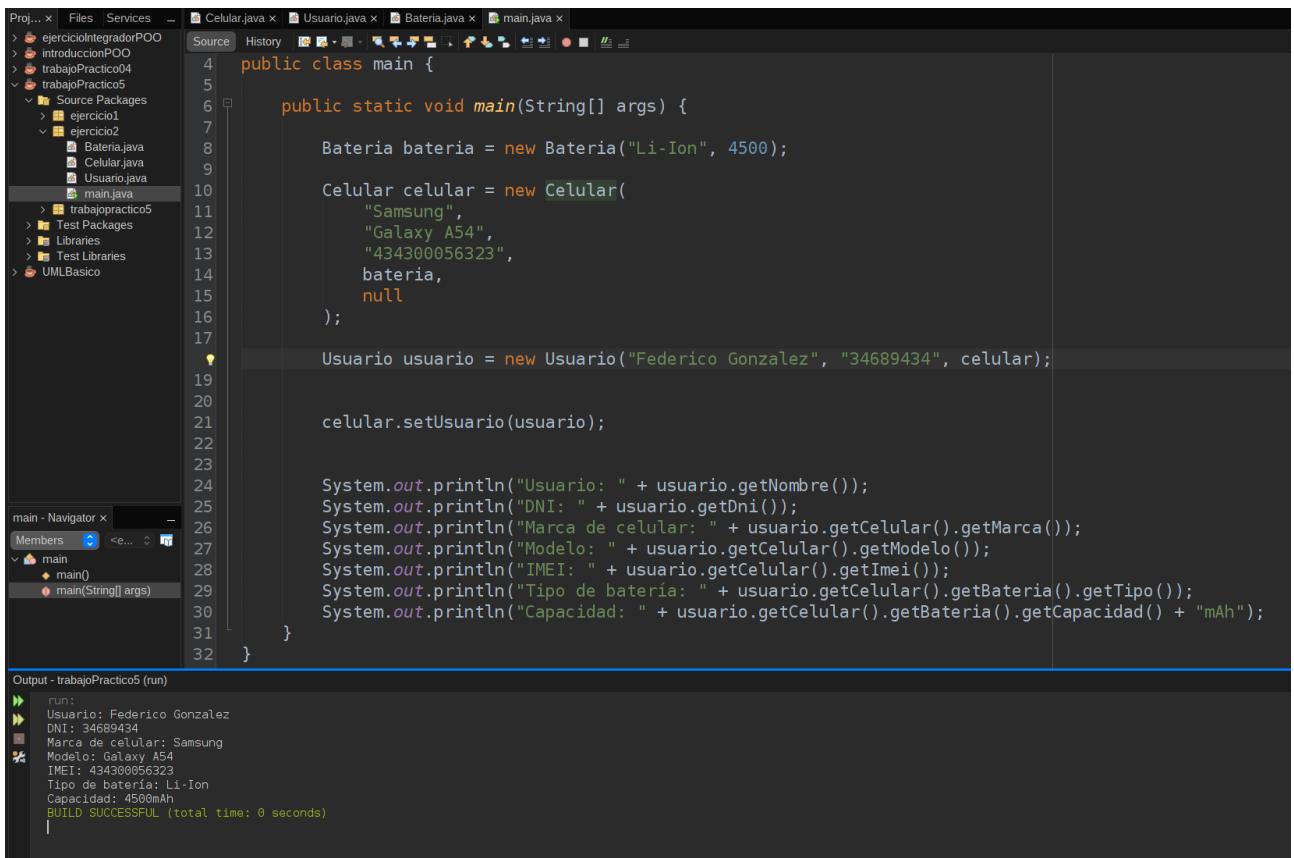
```
1 package ejercicio2;
2
3
4 public class Usuario {
5
6     private String nombre;
7     private String dni;
8     private Celular celular;
9
10    public Usuario(String nombre, String dni, Celular celular) {
11        this.nombre = nombre;
12        this.dni = dni;
13        this.celular = celular;
14    }
15
16
17    public String getNombre() {
18        return nombre;
19    }
20
21    public String getDni() {
22        return dni;
23    }
24
25    public Celular getCelular() {
26        return celular;
27    }
28
29    public void setNombre(String nombre) {
30        this.nombre = nombre;
31    }
32
33    public void setDni(String dni) {
34        this.dni = dni;
35    }
36
37    public void setCelular(Celular celular) {
38        this.celular = celular;
39    }
40
41
42
43 }
44
```

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

The screenshot shows a Java code editor with the file `Bateria.java` open. The code defines a class `Bateria` with private attributes `tipo` and `capacidad`, and methods to get and set these values. The code is part of a larger project with other files like `Celular.java`, `Usuario.java`, and `main.java` visible in the tabs.

```
1 package ejercicio2;
2
3
4 public class Bateria {
5
6     private String tipo;
7     private int capacidad;
8
9     public Bateria(String tipo, int capacidad) {
10         this.tipo = tipo;
11         this.capacidad = capacidad;
12     }
13
14     public String getTipo() {
15         return tipo;
16     }
17
18     public int getCapacidad() {
19         return capacidad;
20     }
21
22     public void setTipo(String tipo) {
23         this.tipo = tipo;
24     }
25
26     public void setCapacidad(int capacidad) {
27         this.capacidad = capacidad;
28     }
29
30
31 }
32 }
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



The screenshot shows an IDE interface with the following details:

- Project Explorer:** Shows packages like ejercicioIntegradorPOO, ejercicioPractico04, ejercicioPractico05, and Source Packages containing ejercicio1, ejercicio2, and main.java.
- Code Editor:** Displays the main.java file with the following code:


```

public class main {
    public static void main(String[] args) {
        Bateria bateria = new Bateria("Li-Ion", 4500);
        Celular celular = new Celular(
            "Samsung",
            "Galaxy A54",
            "434300056323",
            bateria,
            null
        );
        Usuario usuario = new Usuario("Federico Gonzalez", "34689434", celular);
        celular.setUsuario(usuario);

        System.out.println("Usuario: " + usuario.getNombre());
        System.out.println("DNI: " + usuario.getDni());
        System.out.println("Marca de celular: " + usuario.getCelular().getMarca());
        System.out.println("Modelo: " + usuario.getCelular().getModelo());
        System.out.println("IMEI: " + usuario.getCelular().getImei());
        System.out.println("Tipo de bateria: " + usuario.getCelular().getBateria().getTipo());
        System.out.println("Capacidad: " + usuario.getCelular().getBateria().getCapacidad() + "mAh");
    }
}
      
```
- Output Window:** Shows the run results:


```

run:
Usuario: Federico Gonzalez
DNI: 34689434
Marca de celular: Samsung
Modelo: Galaxy A54
IMEI: 434300056323
Tipo de bateria: Li-Ion
Capacidad: 4500mAh
BUILD SUCCESSFUL (total time: 0 seconds)
      
```

3. Libro - Autor - Editorial

a. Asociación unidireccional: Libro → Autor

b. Agregación: Libro → Editorial

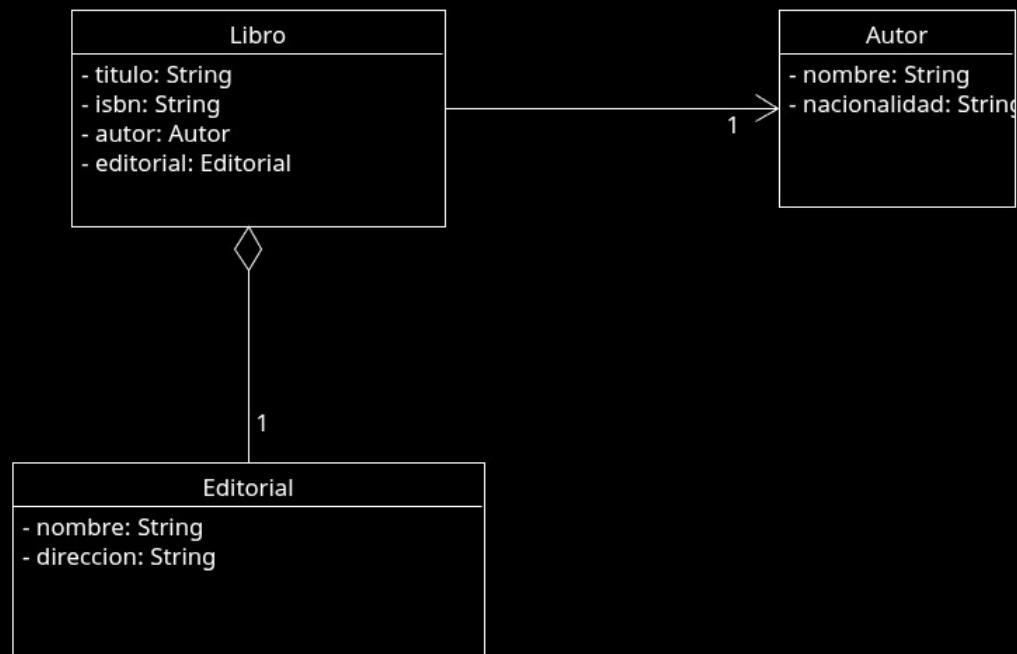
Clases y atributos:

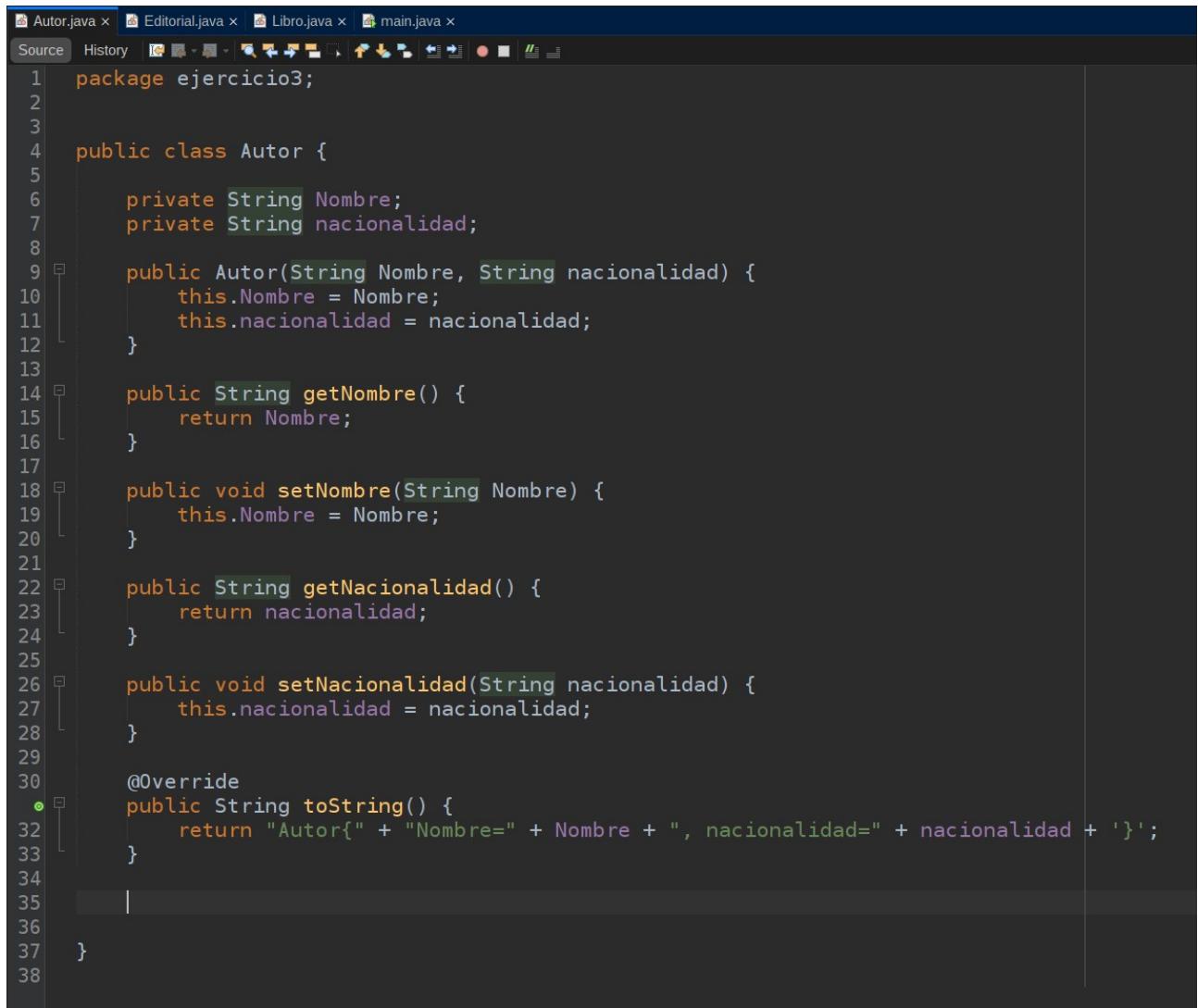
i. Libro: titulo, isbn

ii. Autor: nombre, nacionalidad

iii. Editorial: nombre, dirección

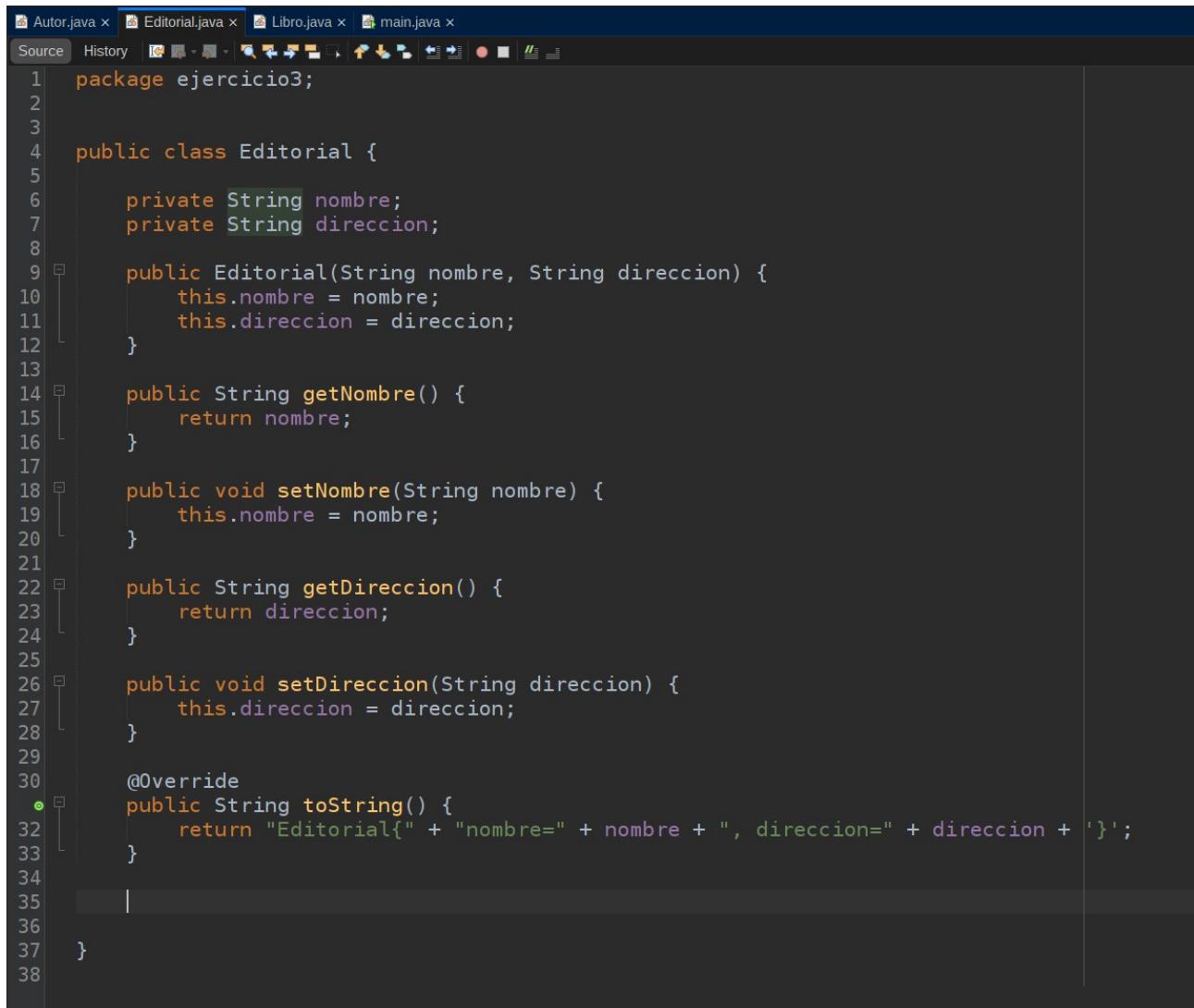
TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA



TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

The screenshot shows a Java code editor with the file `Autor.java` open. The code defines a class `Autor` with private attributes `Nombre` and `nacionalidad`, a constructor, and methods to get and set these values. It also overrides the `toString` method. The code is color-coded, and the editor interface includes tabs for other files like `Editorial.java` and `Libro.java`.

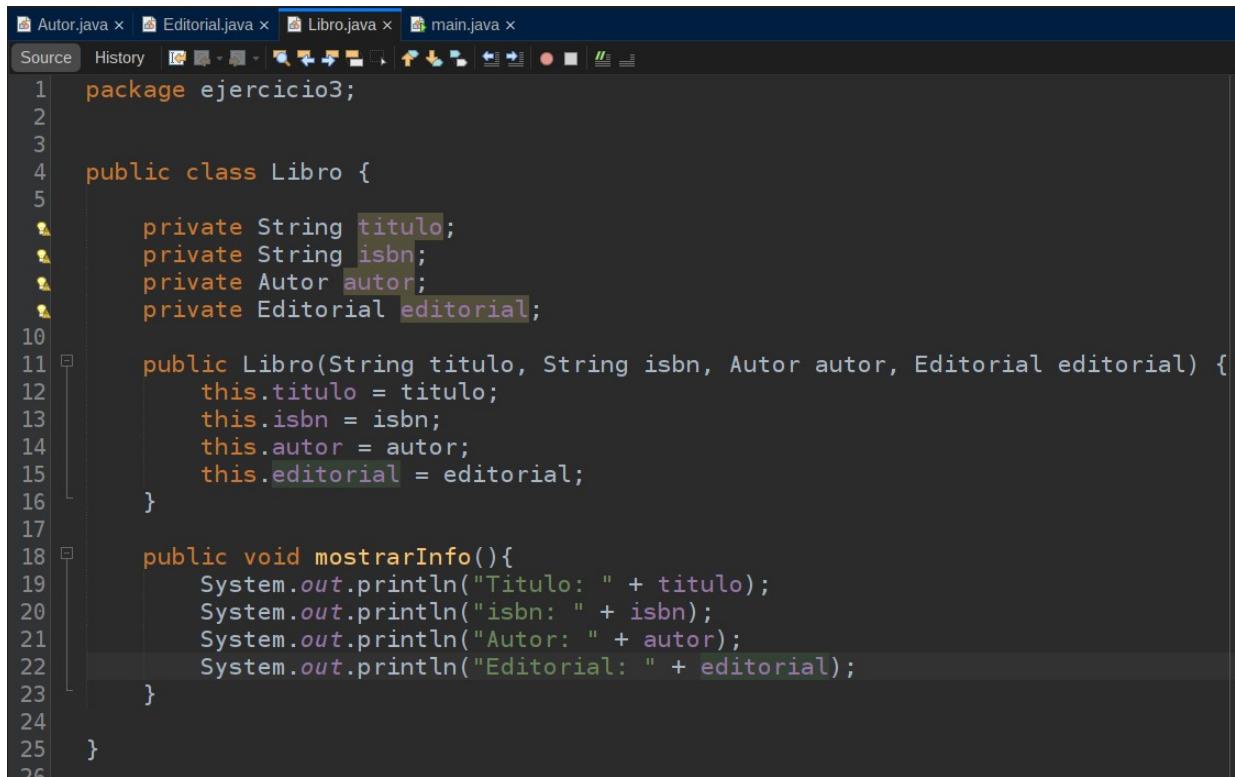
```
1 package ejercicio3;
2
3
4 public class Autor {
5
6     private String Nombre;
7     private String nacionalidad;
8
9     public Autor(String Nombre, String nacionalidad) {
10         this.Nombre = Nombre;
11         this.nacionalidad = nacionalidad;
12     }
13
14     public String getNombre() {
15         return Nombre;
16     }
17
18     public void setNombre(String Nombre) {
19         this.Nombre = Nombre;
20     }
21
22     public String getNacionalidad() {
23         return nacionalidad;
24     }
25
26     public void setNacionalidad(String nacionalidad) {
27         this.nacionalidad = nacionalidad;
28     }
29
30     @Override
31     public String toString() {
32         return "Autor{" + "Nombre=" + Nombre + ", nacionalidad=" + nacionalidad + '}';
33     }
34
35
36
37 }
38 }
```

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

The screenshot shows a Java code editor with the `Editorial.java` file open. The code defines a class `Editorial` with private attributes `nombre` and `direccion`, and methods for getting and setting these values, along with an overridden `toString` method.

```
1 package ejercicio3;
2
3
4 public class Editorial {
5
6     private String nombre;
7     private String direccion;
8
9     public Editorial(String nombre, String direccion) {
10         this.nombre = nombre;
11         this.direccion = direccion;
12     }
13
14     public String getNombre() {
15         return nombre;
16     }
17
18     public void setNombre(String nombre) {
19         this.nombre = nombre;
20     }
21
22     public String getDireccion() {
23         return direccion;
24     }
25
26     public void setDireccion(String direccion) {
27         this.direccion = direccion;
28     }
29
30     @Override
31     public String toString() {
32         return "Editorial{" + "nombre=" + nombre + ", direccion=" + direccion + '}';
33     }
34
35
36
37 }
38
```

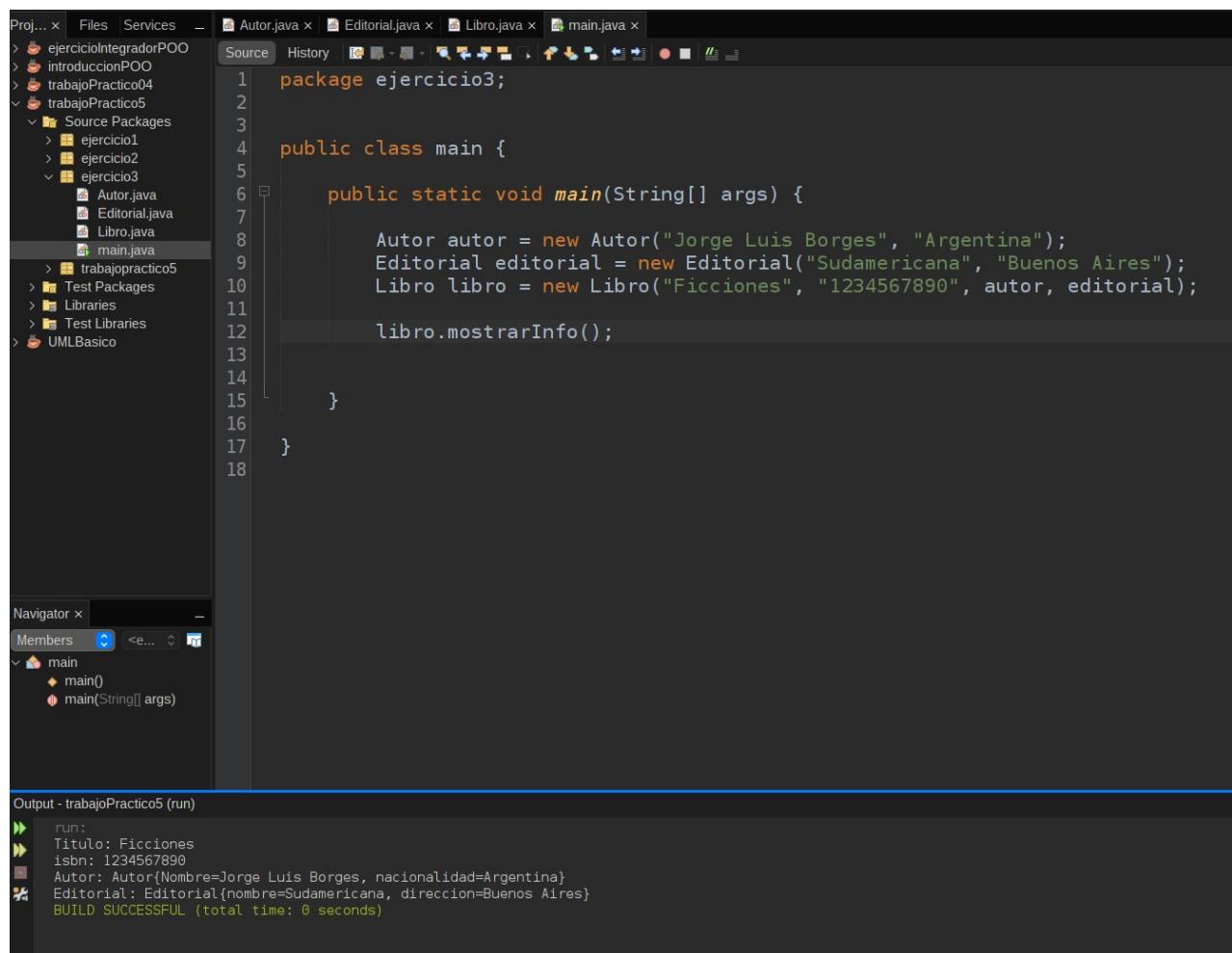
TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



```

 1 package ejercicio3;
 2
 3
 4 public class Libro {
 5
 6     private String titulo;
 7     private String isbn;
 8     private Autor autor;
 9     private Editorial editorial;
10
11     public Libro(String titulo, String isbn, Autor autor, Editorial editorial) {
12         this.titulo = titulo;
13         this.isbn = isbn;
14         this.autor = autor;
15         this.editorial = editorial;
16     }
17
18     public void mostrarInfo(){
19         System.out.println("Titulo: " + titulo);
20         System.out.println("isbn: " + isbn);
21         System.out.println("Autor: " + autor);
22         System.out.println("Editorial: " + editorial);
23     }
24
25 }
26

```



Project Explorer:

- > ejercicioIntegradorPOO
- > introducciónPOO
- > trabajoPráctico04
- < trabajoPráctico5
 - < Source Packages
 - > ejercicio1
 - > ejercicio2
 - < ejercicio3
 - Autor.java
 - Editorial.java
 - Libro.java
 - main.java
 - < Test Packages
 - < Libraries
 - < Test Libraries
- > UMLBasico

Source Editor (main.java):

```

 1 package ejercicio3;
 2
 3
 4 public class main {
 5
 6     public static void main(String[] args) {
 7
 8         Autor autor = new Autor("Jorge Luis Borges", "Argentina");
 9         Editorial editorial = new Editorial("Sudamericana", "Buenos Aires");
10         Libro libro = new Libro("Ficciones", "1234567890", autor, editorial);
11
12         libro.mostrarInfo();
13
14     }
15
16 }
17
18

```

Navigator:

- Members
 - main
 - main()
 - main(String[] args)

Output - trabajoPráctico5 (run)

```

 1 run:
 2 Titulo: Ficciones
 3 isbn: 1234567890
 4 Autor: Autor{Nombre=Jorge Luis Borges, nacionalidad=Argentina}
 5 Editorial: Editorial{nombre=Sudamericana, dirección=Buenos Aires}
 6 BUILD SUCCESSFUL (total time: 0 seconds)

```

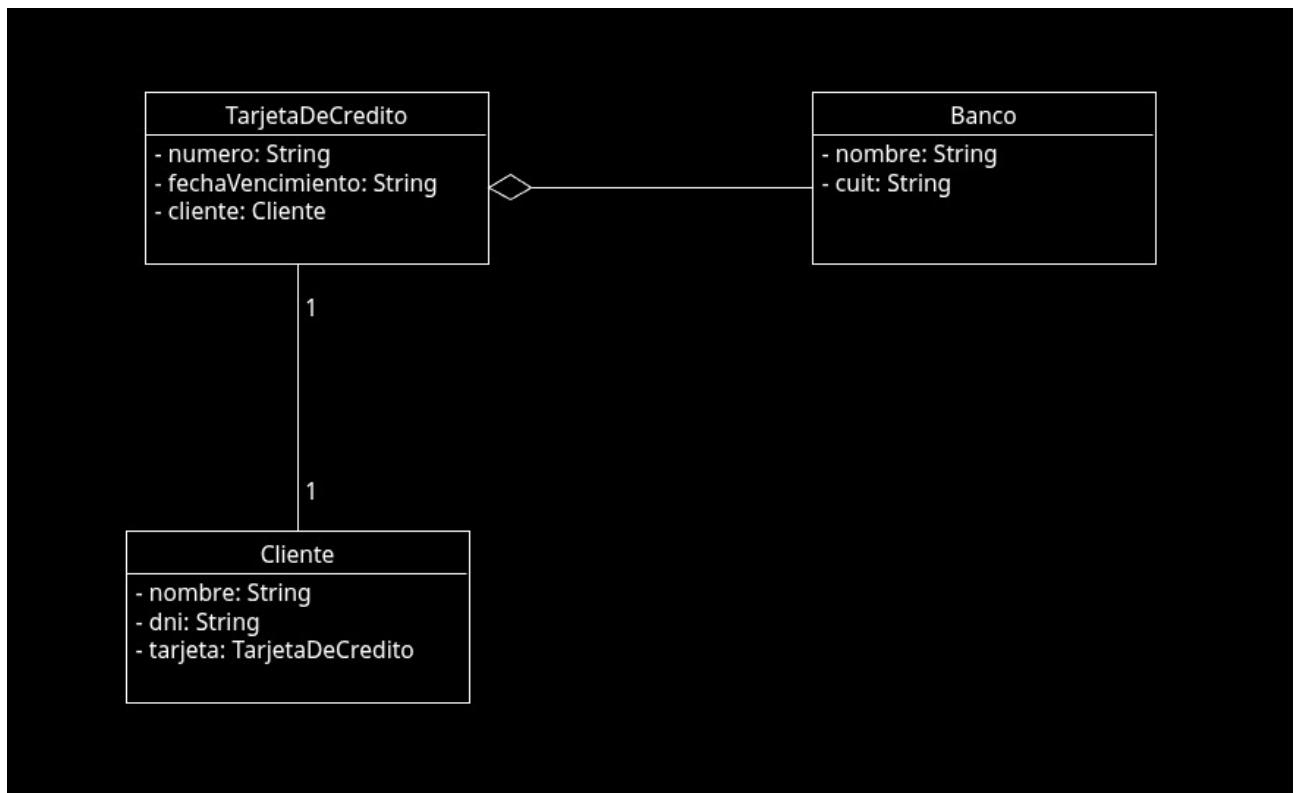
TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

4. TarjetaDeCrédito - Cliente - Banco

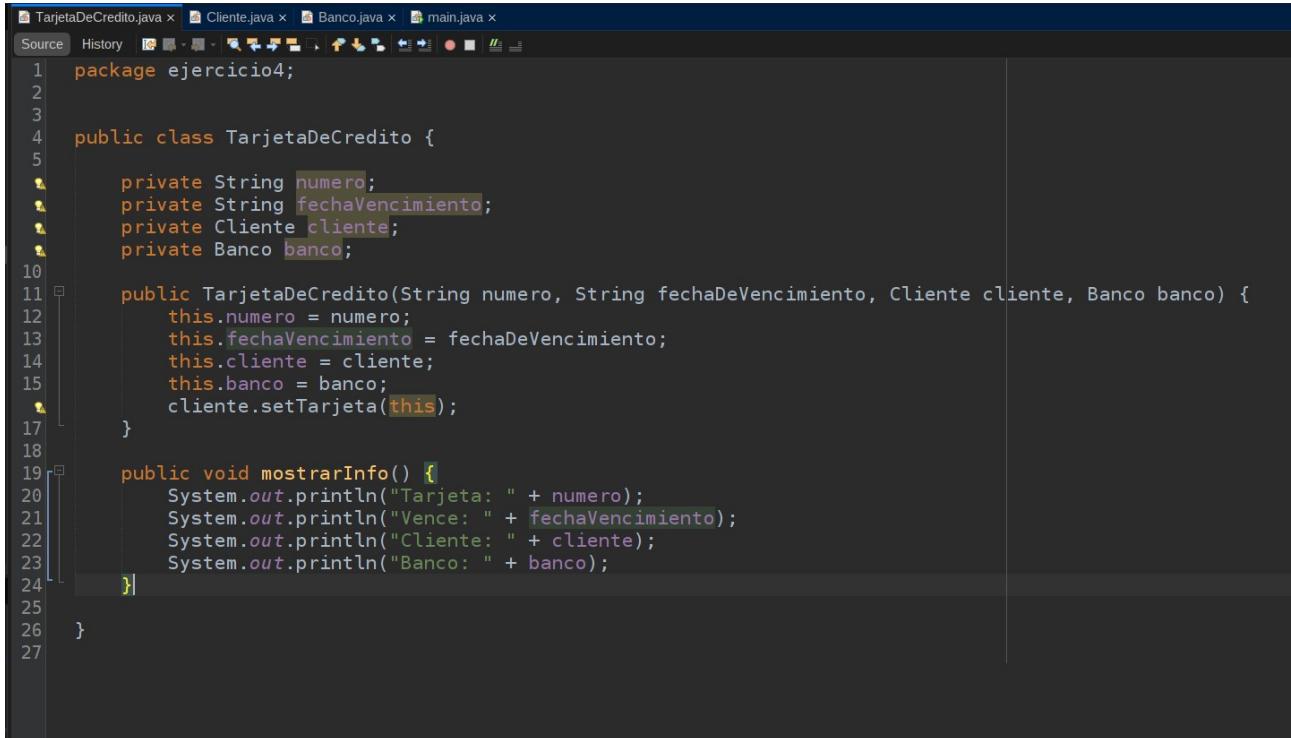
- a. Asociación bidireccional: TarjetaDeCrédito ↔ Cliente
- b. Agregación: TarjetaDeCrédito → Banco

Clases y atributos:

- i. TarjetaDeCrédito: numero, fechaVencimiento
- ii. Cliente: nombre, dni
- iii. Banco: nombre, cuit



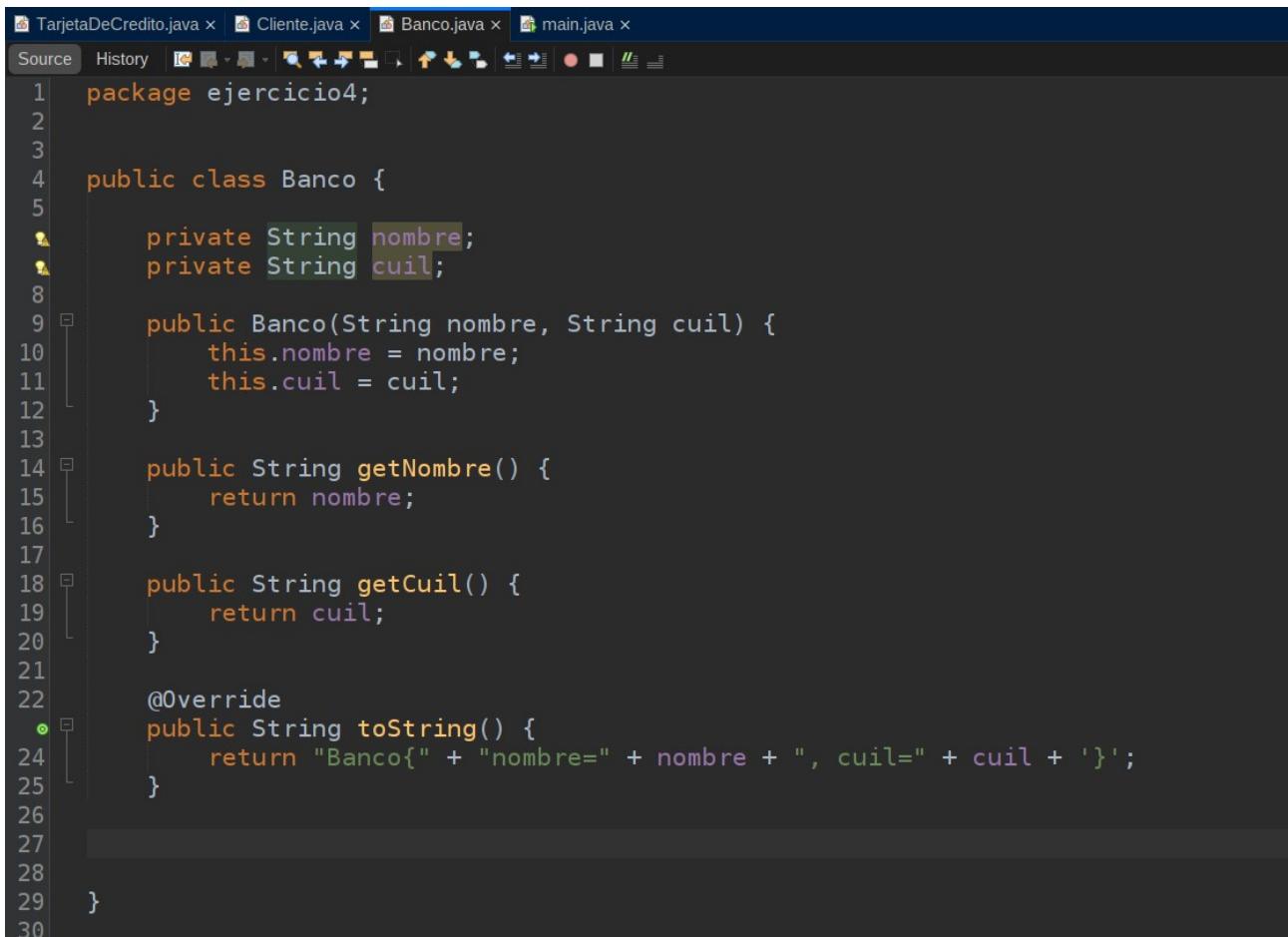
TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



```

1 package ejercicio4;
2
3
4 public class TarjetaDeCredito {
5
6     private String numero;
7     private String fechaVencimiento;
8     private Cliente cliente;
9     private Banco banco;
10
11    public TarjetaDeCredito(String numero, String fechaDeVencimiento, Cliente cliente, Banco banco) {
12        this.numero = numero;
13        this.fechaVencimiento = fechaDeVencimiento;
14        this.cliente = cliente;
15        this.banco = banco;
16        cliente.setTarjeta(this);
17    }
18
19    public void mostrarInfo() {
20        System.out.println("Tarjeta: " + numero);
21        System.out.println("Vence: " + fechaVencimiento);
22        System.out.println("Cliente: " + cliente);
23        System.out.println("Banco: " + banco);
24    }
25
26 }
27

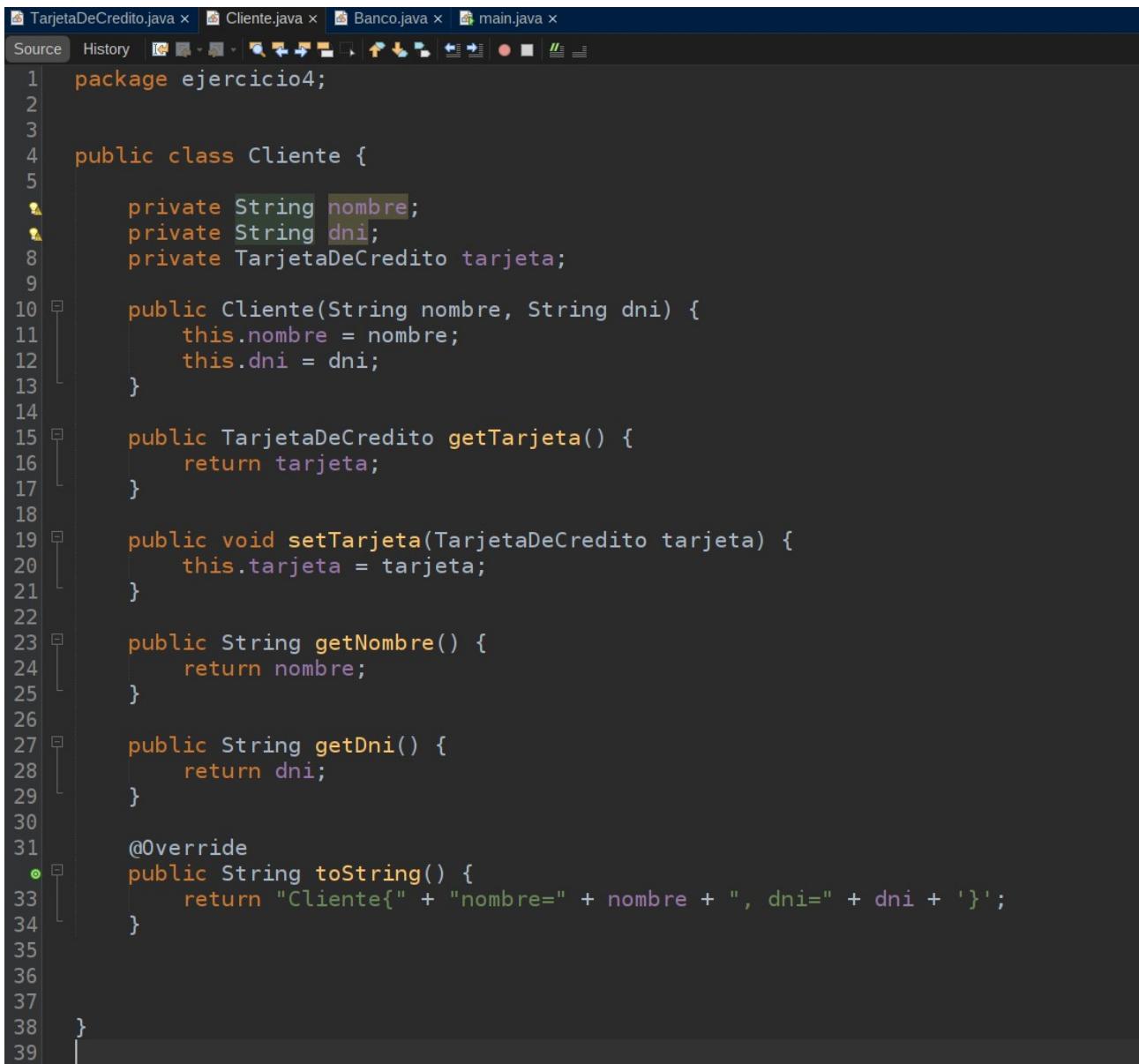
```



```

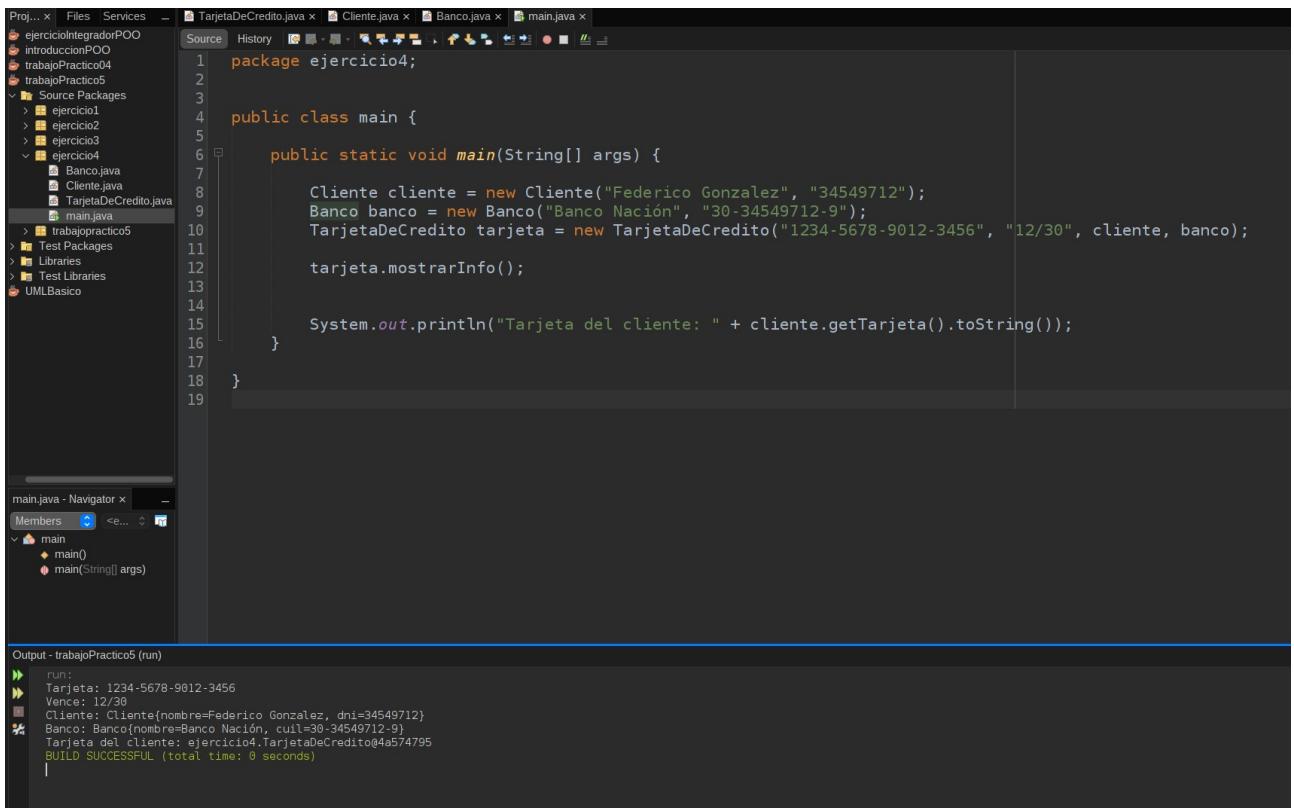
1 package ejercicio4;
2
3
4 public class Banco {
5
6     private String nombre;
7     private String cuil;
8
9    public Banco(String nombre, String cuil) {
10        this.nombre = nombre;
11        this.cuil = cuil;
12    }
13
14    public String getNombre() {
15        return nombre;
16    }
17
18    public String getCuil() {
19        return cuil;
20    }
21
22    @Override
23    public String toString() {
24        return "Banco{" + "nombre=" + nombre + ", cuil=" + cuil + '}';
25    }
26
27
28 }
29

```

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

```
1 package ejercicio4;
2
3
4 public class Cliente {
5
6     private String nombre;
7     private String dni;
8     private TarjetaDeCredito tarjeta;
9
10    public Cliente(String nombre, String dni) {
11        this.nombre = nombre;
12        this.dni = dni;
13    }
14
15    public TarjetaDeCredito getTarjeta() {
16        return tarjeta;
17    }
18
19    public void setTarjeta(TarjetaDeCredito tarjeta) {
20        this.tarjeta = tarjeta;
21    }
22
23    public String getNombre() {
24        return nombre;
25    }
26
27    public String getDni() {
28        return dni;
29    }
30
31    @Override
32    public String toString() {
33        return "Cliente{" + "nombre=" + nombre + ", dni=" + dni + '}';
34    }
35
36
37
38 }
39
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



```

package ejercicio4;

public class main {

    public static void main(String[] args) {
        Cliente cliente = new Cliente("Federico Gonzalez", "34549712");
        Banco banco = new Banco("Banco Nación", "30-34549712-9");
        TarjetaDeCredito tarjeta = new TarjetaDeCredito("1234-5678-9012-3456", "12/30", cliente, banco);

        tarjeta.mostrarInfo();

        System.out.println("Tarjeta del cliente: " + cliente.getTarjeta().toString());
    }
}

```

Output - trabajoPractico5 (run)

```

run:
Tarjeta: 1234-5678-9012-3456
Vence: 12/30
Cliente: Cliente(nombre=Federico Gonzalez, dni=34549712)
Banco: Banco(nombre=Banco Nación, cuit=30-34549712-9)
Tarjeta del cliente: ejercicio4.TarjetaDeCredito@4a574795
BUILD SUCCESSFUL (total time: 0 seconds)
|
```

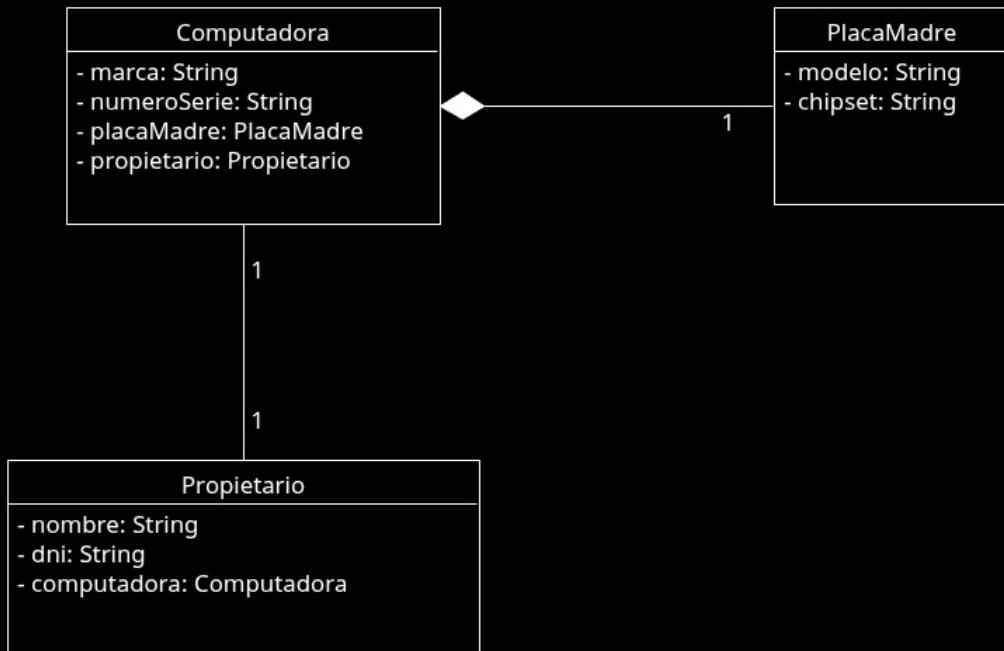
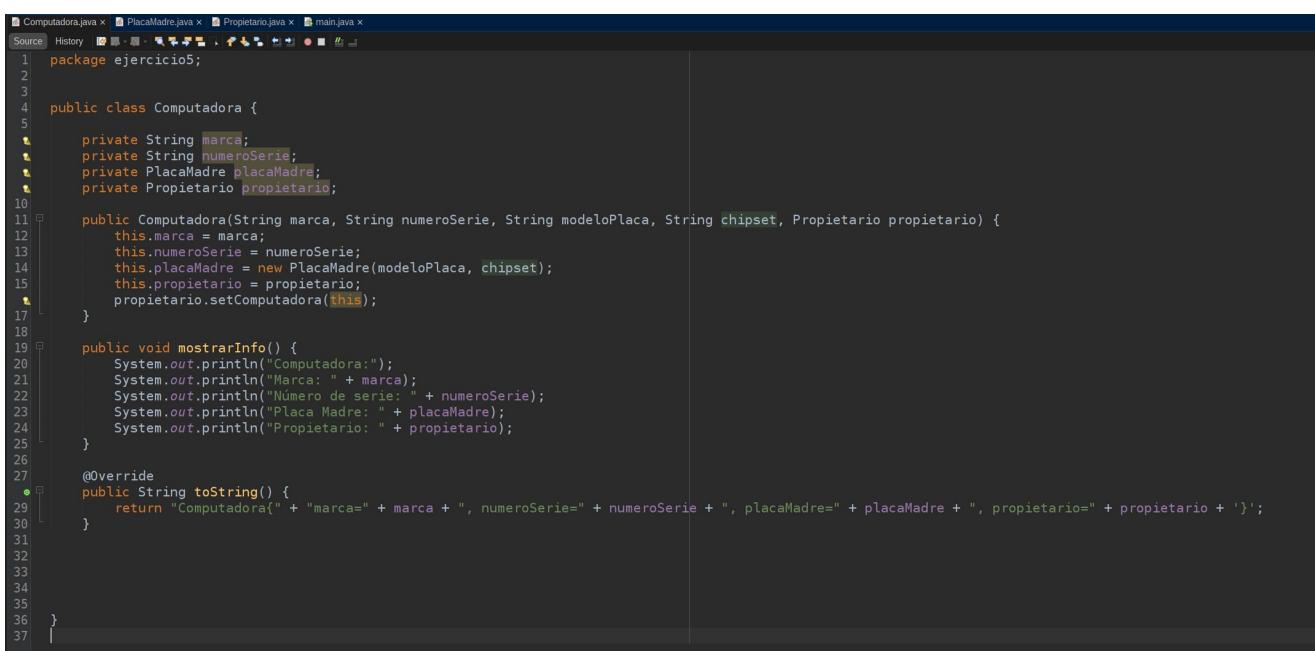
5. Computadora - PlacaMadre - Propietario

- a. Composición: Computadora → PlacaMadre
- b. Asociación bidireccional: Computadora ↔ Propietario

Clases y atributos:

- i. Computadora: marca, numeroSerie
- ii. PlacaMadre: modelo, chipset
- iii. Propietario: nombre, dni

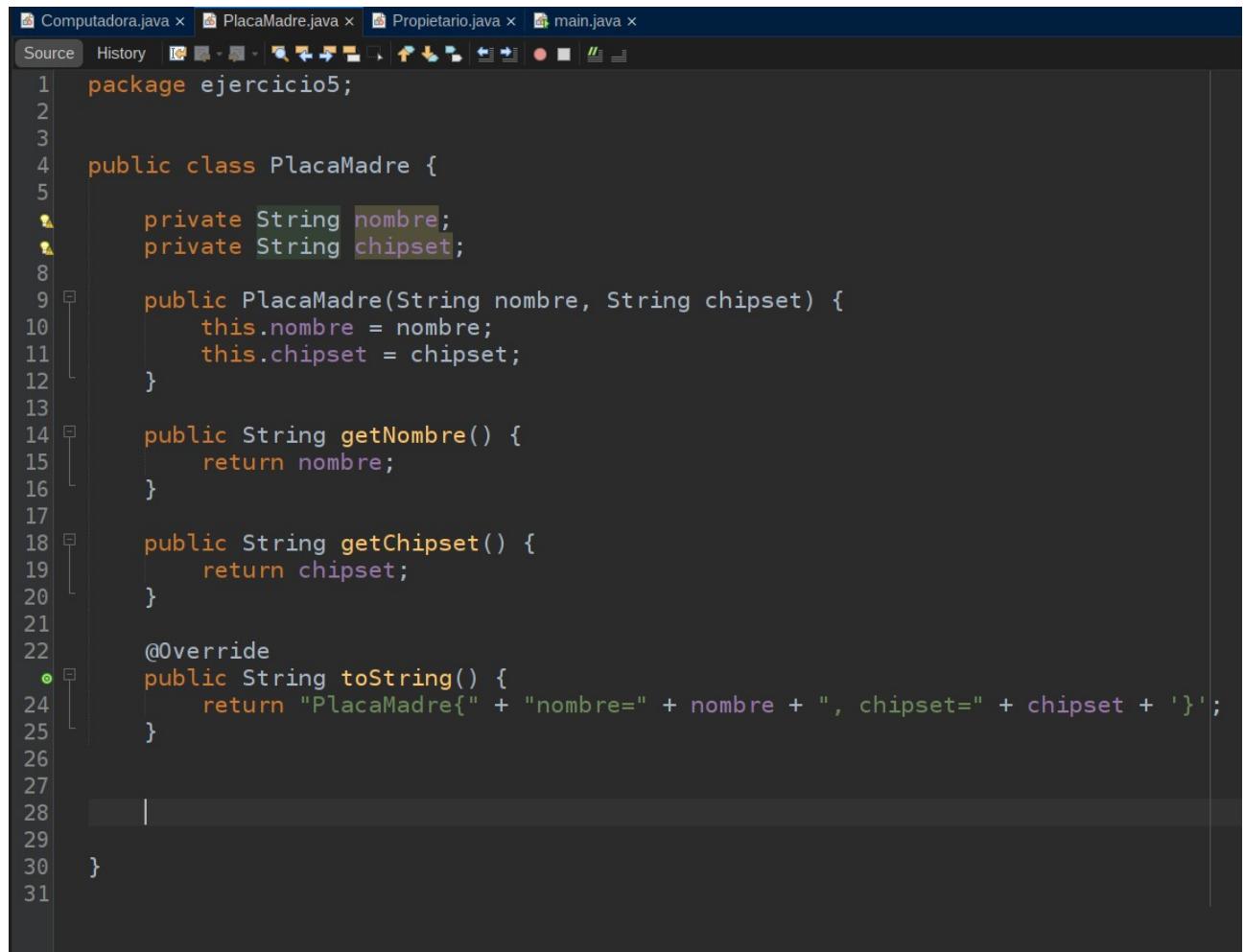
TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

```

1 package ejercicio5;
2
3
4 public class Computadora {
5
6     private String marca;
7     private String numeroSerie;
8     private PlacaMadre placaMadre;
9     private Propietario propietario;
10
11     public Computadora(String marca, String numeroSerie, String modeloPlaca, String chipset, Propietario propietario) {
12         this.marca = marca;
13         this.numeroSerie = numeroSerie;
14         this.placaMadre = new PlacaMadre(modeloPlaca, chipset);
15         this.propietario = propietario;
16         propietario.setComputadora(this);
17     }
18
19     public void mostrarInfo() {
20         System.out.println("Computadora:");
21         System.out.println("Marca: " + marca);
22         System.out.println("Número de serie: " + numeroSerie);
23         System.out.println("Placa Madre: " + placaMadre);
24         System.out.println("Propietario: " + propietario);
25     }
26
27     @Override
28     public String toString() {
29         return "Computadora{" + "marca=" + marca + ", numeroSerie=" + numeroSerie + ", placaMadre=" + placaMadre + ", propietario=" + propietario + '}';
30     }
31
32
33
34
35
36
37 }
  
```

The screenshot shows the Java code for the **Computadora** class. The class has private attributes for `marca`, `numeroSerie`, `placaMadre`, and `propietario`. It includes a constructor that initializes these attributes and sets the `Computadora` object as the `computadora` attribute of the `Propietario` object. The class also contains a `mostrarInfo()` method to print information about the computer and a `toString()` method to provide a string representation of the object.

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

The screenshot shows a Java code editor with the following code:

```
1 package ejercicio5;
2
3 public class PlacaMadre {
4
5     private String nombre;
6     private String chipset;
7
8     public PlacaMadre(String nombre, String chipset) {
9         this.nombre = nombre;
10        this.chipset = chipset;
11    }
12
13    public String getNombre() {
14        return nombre;
15    }
16
17    public String getChipset() {
18        return chipset;
19    }
20
21    @Override
22    public String toString() {
23        return "PlacaMadre{" + "nombre=" + nombre + ", chipset=" + chipset + '}';
24    }
25
26
27
28
29
30 }
31
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Screenshots of the Java code for the 'Propietario' class and the 'main' method.

```

 1 package ejercicio5;
 2
 3
 4 public class Propietario {
 5
 6     private String nombre;
 7     private String dni;
 8     private Computadora computadora;
 9
10    public Propietario(String nombre, String dni) {
11        this.nombre = nombre;
12        this.dni = dni;
13    }
14
15    public Computadora getComputadora() {
16        return computadora;
17    }
18
19    public void setComputadora(Computadora computadora) {
20        this.computadora = computadora;
21    }
22
23
24
25    @Override
26    public String toString() {
27        return "Propietario{" + "nombre=" + nombre + ", dni=" + dni + '}';
28    }
29
30
31
32 }
  
```

Screenshot of the Eclipse IDE interface showing the project structure and the output of the 'main' method execution.

Project Explorer:

- ejercicioIntegradorPOO
- introducciónPOO
- trabajoPráctico04
- trabajoPráctico5
- Source Packages
 - ejercicio1
 - ejercicio2
 - ejercicio3
 - ejercicio4
 - ejercicio5
 - Computadora.java
 - PlacaMadre.java
 - Propietario.java
 - main.java
- trabajoPráctico5
- Test Packages
- Libraries
- Test Libraries
- UMLBasico

Code Editor:

```

1 package ejercicio5;
2
3 public class main {
4
5     public static void main(String[] args) {
6
7         Propietario propietario = new Propietario("Federico González", "34692605");
8
9         Computadora compu = new Computadora("HP", "SN-987654", "Asus TUF B450", "AMD B450", propietario);
10
11         compu.mostrarInfo();
12
13         System.out.println("Computadora del propietario: " + propietario.getComputadora());
14
15     }
16
17 }
  
```

Navigator:

- main
 - main()
 - main(String[] args)

Output - trabajoPráctico5 (run)

```

run:
Computadora:
  Marca: HP
  Número de serie: SN-987654
  Placa Madre: PlacaMadre(nombre=Asus TUF B450, chipset=AMD B450)
  Propietario: Propietario(nombre=Federico González, dni=34692605)
  Computadora del propietario: Computadora(marca=HP, numeroSerie=SN-987654, placaMadre=PlacaMadre(nombre=Asus TUF B450, chipset=AMD B450), propietario=Propietario(nombre=Federico González, dni=34692605)}
BUILD SUCCESSFUL (total time: 0 seconds)
  
```

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

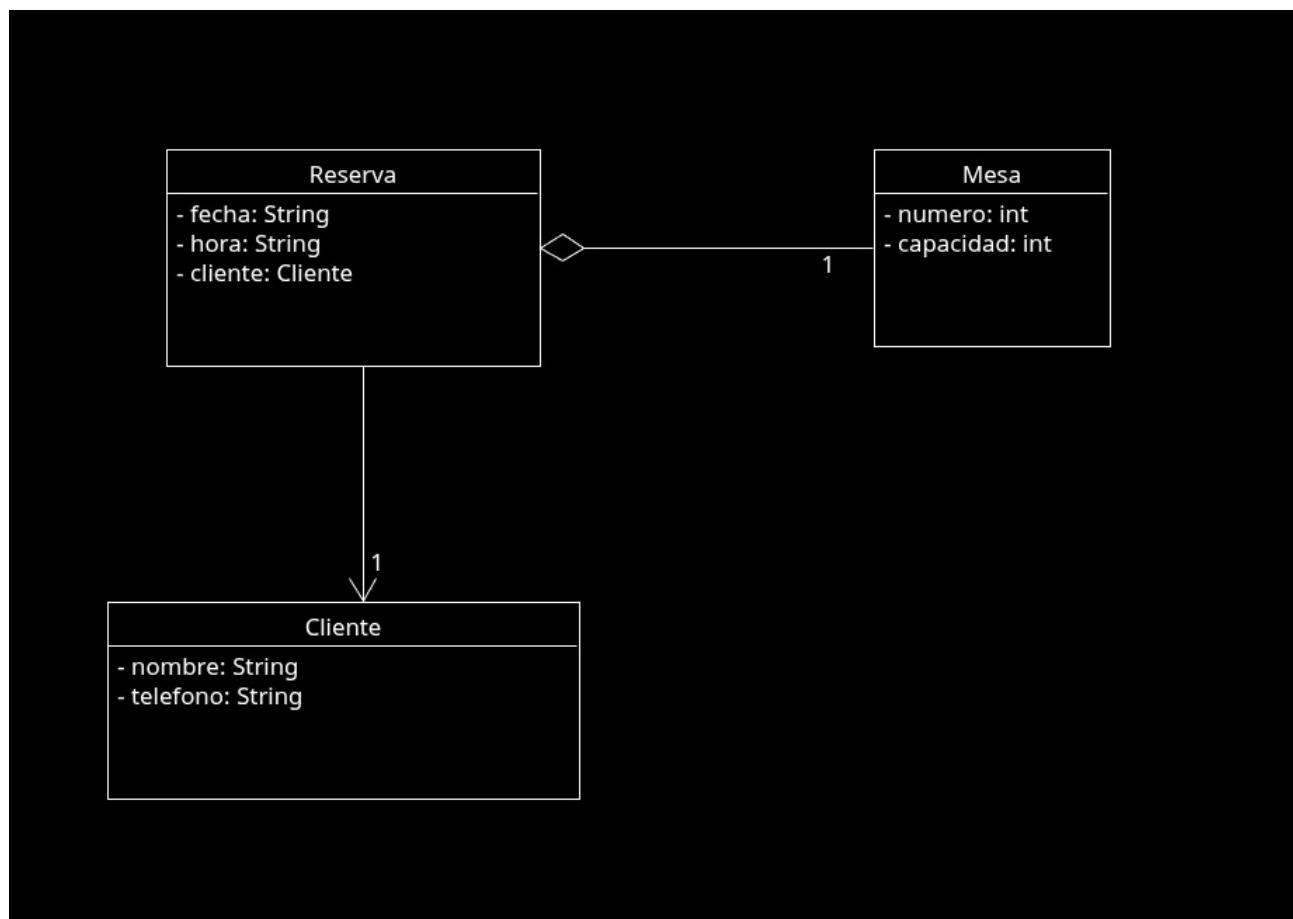
6. Reserva - Cliente - Mesa

a. Asociación unidireccional: Reserva → Cliente

b. Agregación: Reserva → Mesa

Clases y atributos:

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



TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Reserva.java x Cliente.java x Mesa.java x main.java x

Source History

```

1 package ejercicio6;
2
3
4 public class Reserva {
5
6     private String fecha;
7     private String hora;
8     private Cliente cliente;
9     private Mesa mesa;
10
11    public Reserva(String fecha, String hora, Cliente cliente, Mesa mesa) {
12        this.fecha = fecha;
13        this.hora = hora;
14        this.cliente = cliente;
15        this.mesa = mesa;
16    }
17
18    @Override
19    public String toString() {
20        return "Reserva{" + "fecha=" + fecha + ", hora=" + hora + ", cliente=" + cliente + ", mesa=" + mesa + '}';
21    }
22
23
24
25 }
26

```

Reserva.java x Cliente.java x Mesa.java x main.java x

Source History

```

1 package ejercicio6;
2
3
4 public class Cliente {
5
6     private String nombre;
7     private String telefono;
8
9     public Cliente(String nombre, String telefono) {
10         this.nombre = nombre;
11         this.telefono = telefono;
12     }
13
14     public String getNombre() {
15         return nombre;
16     }
17
18     public String getTelefono() {
19         return telefono;
20     }
21
22     @Override
23     public String toString() {
24         return "Cliente{" + "nombre=" + nombre + ", telefono=" + telefono + '}';
25     }
26
27
28
29 }
30

```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Reserva.java x Cliente.java x Mesa.java x main.java x

Source History

```

1 package ejercicio6;
2
3
4 public class Mesa {
5
6     private int numero;
7     private int capacidad;
8
9     public Mesa(int numero, int capacidad) {
10         this.numero = numero;
11         this.capacidad = capacidad;
12     }
13
14     public int getNumero() {
15         return numero;
16     }
17
18     public int getCapacidad() {
19         return capacidad;
20     }
21
22     @Override
23     public String toString() {
24         return "Mesa{" + "numero=" + numero + ", capacidad=" + capacidad + '}';
25     }
26
27
28 }
29
30

```

Proj... x Files Services - Reserva.java x Cliente.java x Mesa.java x main.java x

Source History

Project Explorer

- > ejercicioIntegradorPOO
- > introducciónPOO
- > trabajoPráctico04
- > trabajoPráctico5
- > ejercicio1
- > ejercicio2
- > ejercicio3
- > ejercicio4
- > ejercicio5
- > ejercicio6
 - ↳ ejercicio1
 - ↳ ejercicio2
 - ↳ ejercicio3
 - ↳ ejercicio4
 - ↳ ejercicio5
 - ↳ ejercicio6
 - ↳ main.java
- > trabajoPrácticos
- > Test Packages
- > Libraries
- > Test Libraries
- > UMLBasico

Source Packages

main

```

1 package ejercicio6;
2
3
4 public class main {
5
6     public static void main(String[] args) {
7
8         Cliente cliente = new Cliente("Federico González", "38712345099");
9         Mesa mesa = new Mesa(6, 2);
10        Reserva reserva = new Reserva("2025-09-21", "21:00", cliente, mesa);
11
12        System.out.println(reserva);
13    }
14
15 }
16
17
18

```

Navigator x

Members

main

- main()
- main(String[] args)

Output - trabajoPráctico5 (run)

run:

Reserva{fecha=2025-09-21, hora=21:00, cliente=Cliente{nombre=Federico González, telefono=38712345099}, mesa=Mesa{numero=6, capacidad=2}}

BUILD SUCCESSFUL (total time: 0 seconds)

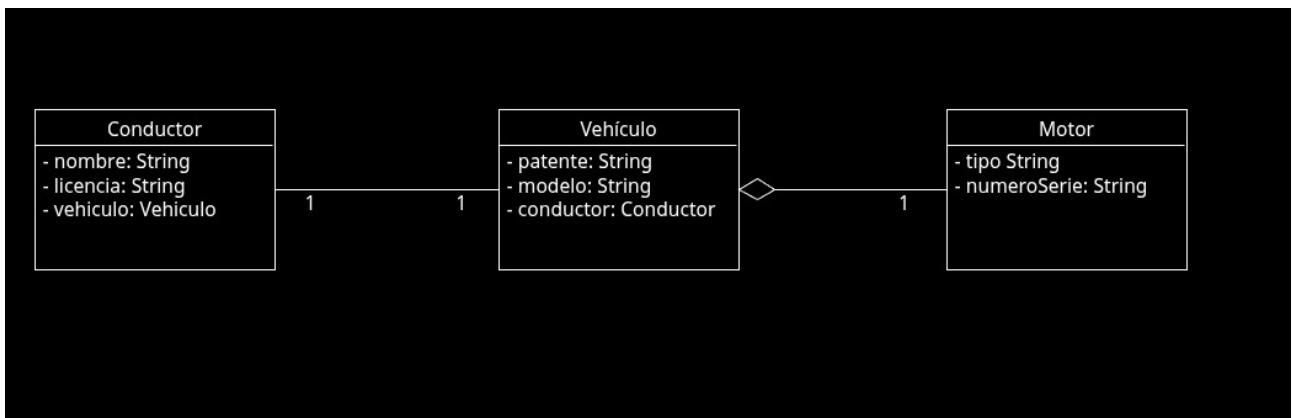
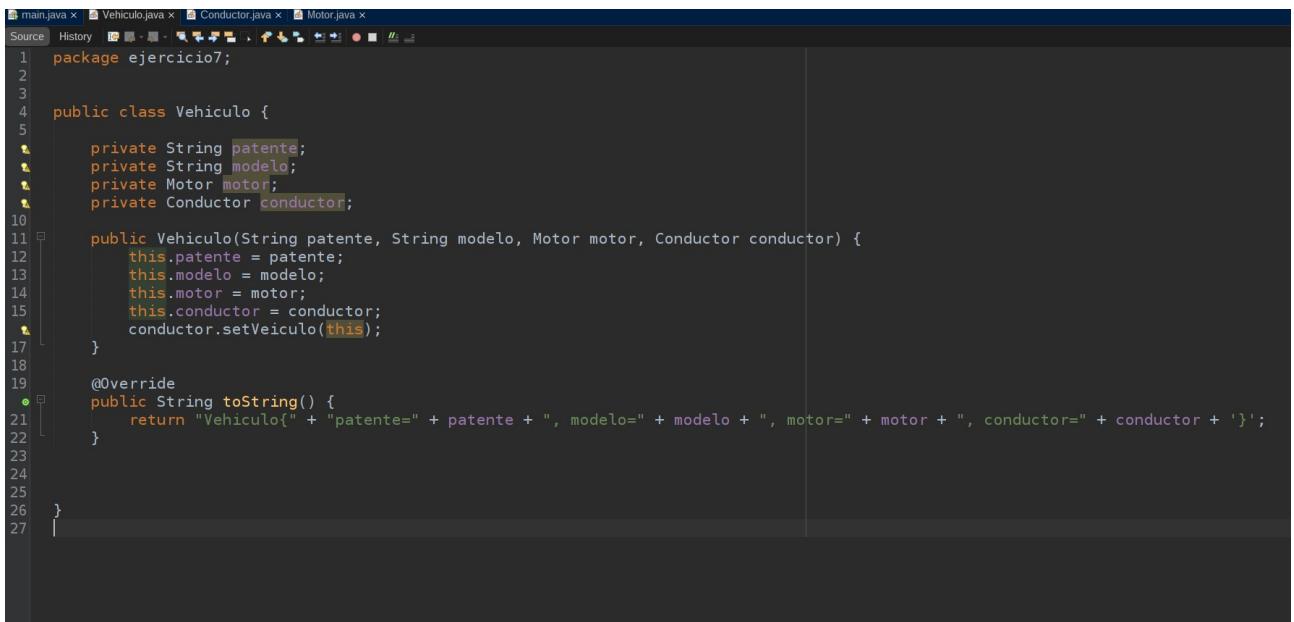
TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

7. Vehículo - Motor - Conductor

- a. Agregación: Vehículo → Motor
- b. Asociación bidireccional: Vehículo ↔ Conductor

Clases y atributos:

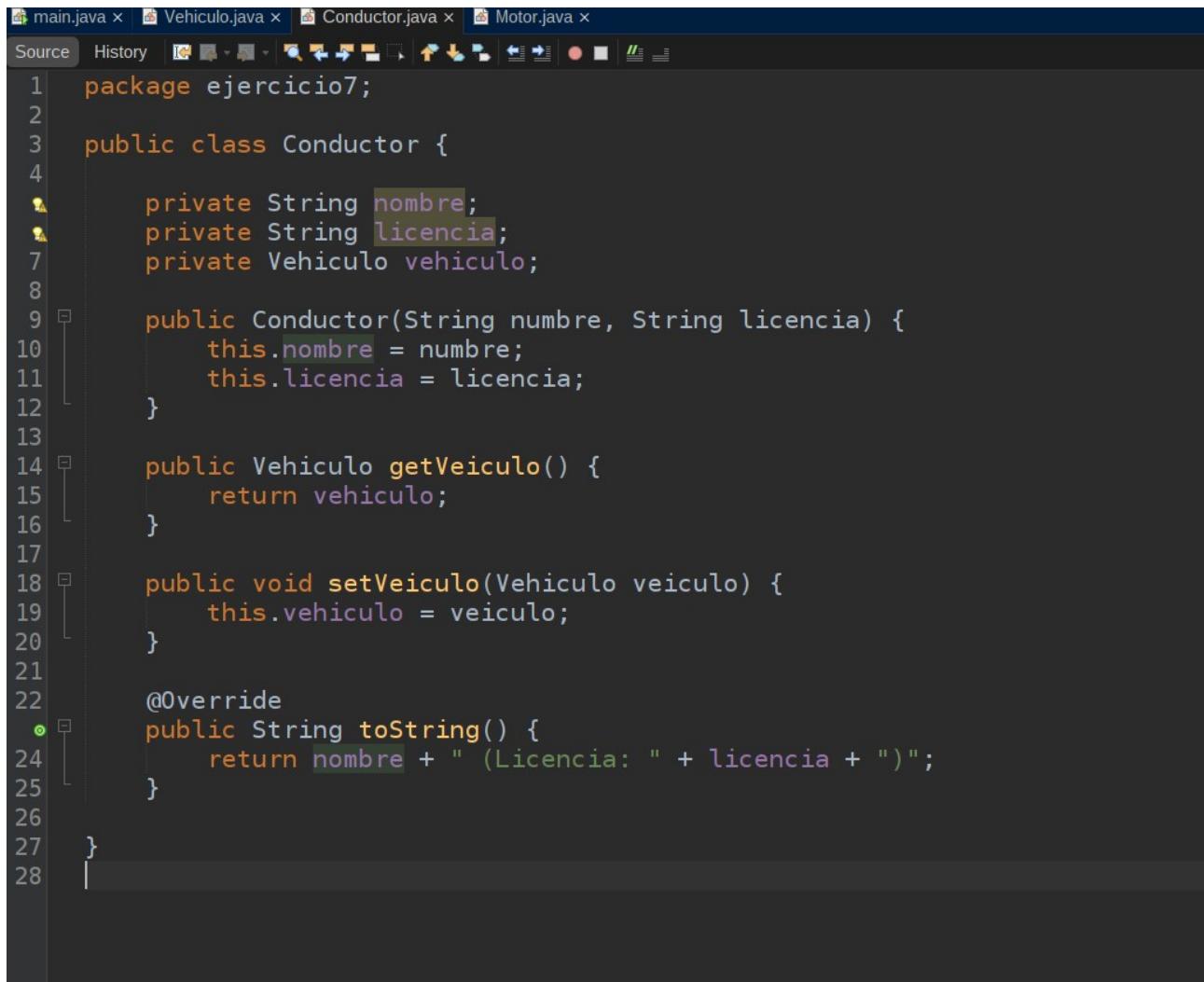
- i. Vehículo: patente, modelo
- ii. Motor: tipo, numeroSerie
- iii. Conductor: nombre, licencia

```

main.java X Vehiculo.java X Conductor.java X Motor.java X
Source History Tools Help
1 package ejercicio7;
2
3
4 public class Vehiculo {
5
6     private String patente;
7     private String modelo;
8     private Motor motor;
9     private Conductor conductor;
10
11     public Vehiculo(String patente, String modelo, Motor motor, Conductor conductor) {
12         this.patente = patente;
13         this.modelo = modelo;
14         this.motor = motor;
15         this.conductor = conductor;
16         conductor.setVehiculo(this);
17     }
18
19     @Override
20     public String toString() {
21         return "Vehiculo{" + "patente=" + patente + ", modelo=" + modelo + ", motor=" + motor + ", conductor=" + conductor + '}';
22     }
23
24
25
26 }
27
  
```

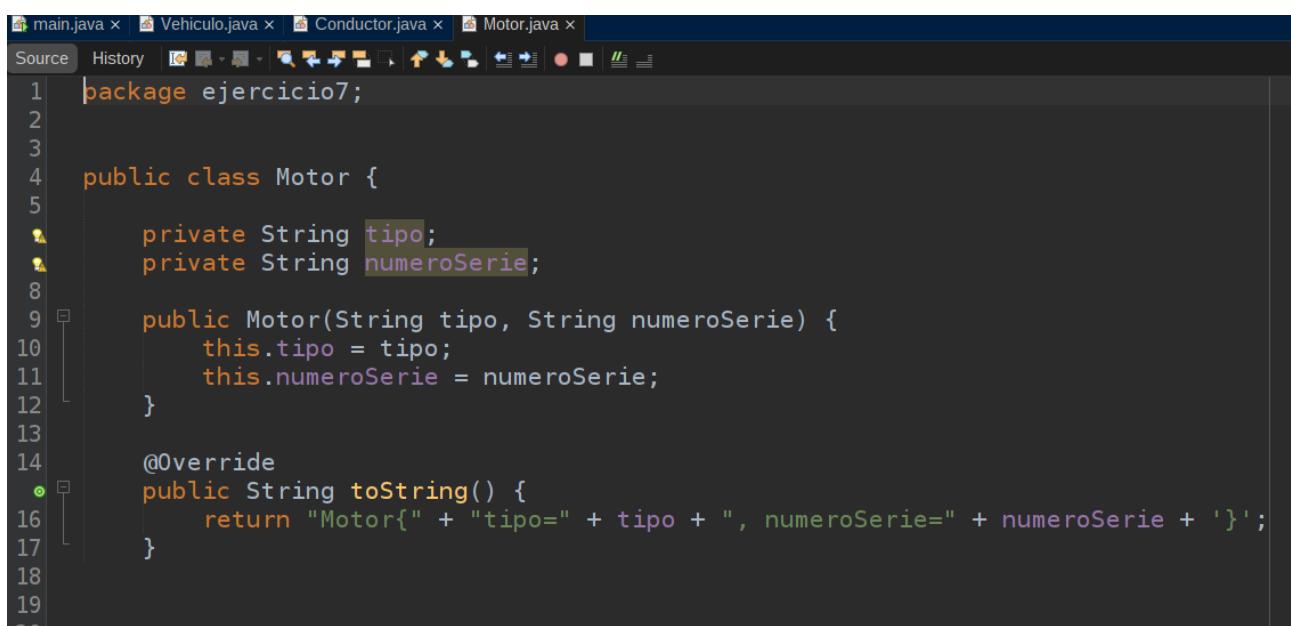
TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



The screenshot shows a Java code editor with the file `Conductor.java` open. The code defines a class `Conductor` with private attributes `nombre` and `licencia`, and a reference to a `Vehiculo` object. It includes a constructor, a getter for the vehicle, a setter for the vehicle, and an overridden `toString` method.

```

1 package ejercicio7;
2
3 public class Conductor {
4
5     private String nombre;
6     private String licencia;
7     private Vehiculo vehiculo;
8
9     public Conductor(String nombre, String licencia) {
10        this.nombre = nombre;
11        this.licencia = licencia;
12    }
13
14    public Vehiculo getVeiculo() {
15        return vehiculo;
16    }
17
18    public void setVeiculo(Vehiculo veiculo) {
19        this.vehiculo = veiculo;
20    }
21
22    @Override
23    public String toString() {
24        return nombre + " (Licencia: " + licencia + ")";
25    }
26
27 }
28
  
```

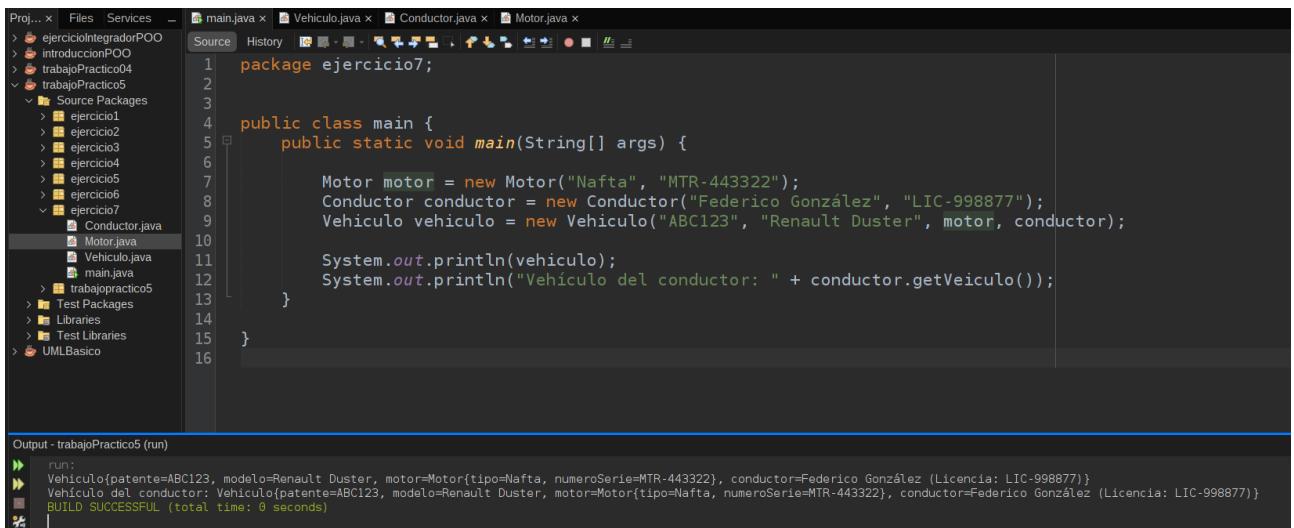


The screenshot shows a Java code editor with the file `Motor.java` open. The code defines a class `Motor` with private attributes `tipo` and `numeroSerie`. It includes a constructor and an overridden `toString` method.

```

1 package ejercicio7;
2
3
4 public class Motor {
5
6     private String tipo;
7     private String numeroSerie;
8
9     public Motor(String tipo, String numeroSerie) {
10        this.tipo = tipo;
11        this.numeroSerie = numeroSerie;
12    }
13
14    @Override
15    public String toString() {
16        return "Motor{" + "tipo=" + tipo + ", numeroSerie=" + numeroSerie + '}';
17    }
18
19
  
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



The screenshot shows an IDE interface with the following details:

- Project Structure:**
 - ejercicioIntegradorPOO
 - introduccionPOO
 - trabajoPractico04
 - trabajoPractico5 (selected)
 - Source Packages:
 - ejercicio1
 - ejercicio2
 - ejercicio3
 - ejercicio4
 - ejercicio5
 - ejercicio6
 - ejercicio7 (selected)
 - Conductor.java
 - Motor.java
 - Vehiculo.java
 - main.java
 - Test Packages
 - Libraries
 - Test Libraries
 - UMLBasico
- Code Editor:** Displays the main.java file with the following code:


```

1 package ejercicio7;
2
3
4 public class main {
5     public static void main(String[] args) {
6
7         Motor motor = new Motor("Nafta", "MTR-443322");
8         Conductor conductor = new Conductor("Federico González", "LIC-998877");
9         Vehiculo vehiculo = new Vehiculo("ABC123", "Renault Duster", motor, conductor);
10
11        System.out.println(vehiculo);
12        System.out.println("Vehiculo del conductor: " + conductor.getVeiculo());
13    }
14
15 }
      
```
- Output:** Shows the run results:


```

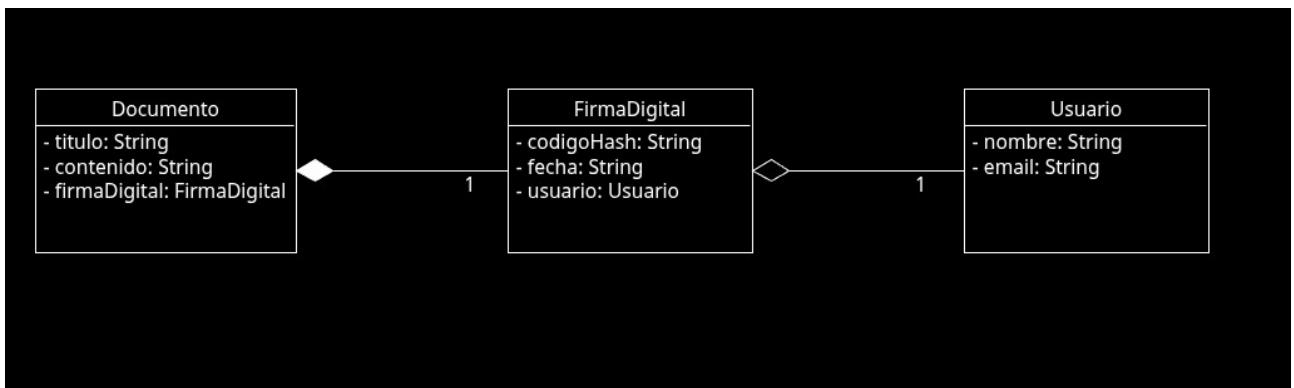
run:
> Vehiculo{patente=ABC123, modelo=Renault Duster, motor=Motor{tipo=Nafta, numeroSerie=MTR-443322}, conductor=Federico González (Licencia: LIC-998877)}
> Vehiculo del conductor: Vehiculo{patente=ABC123, modelo=Renault Duster, motor=Motor{tipo=Nafta, numeroSerie=MTR-443322}, conductor=Federico González (Licencia: LIC-998877)}
BUILD SUCCESSFUL (total time: 0 seconds)
      
```

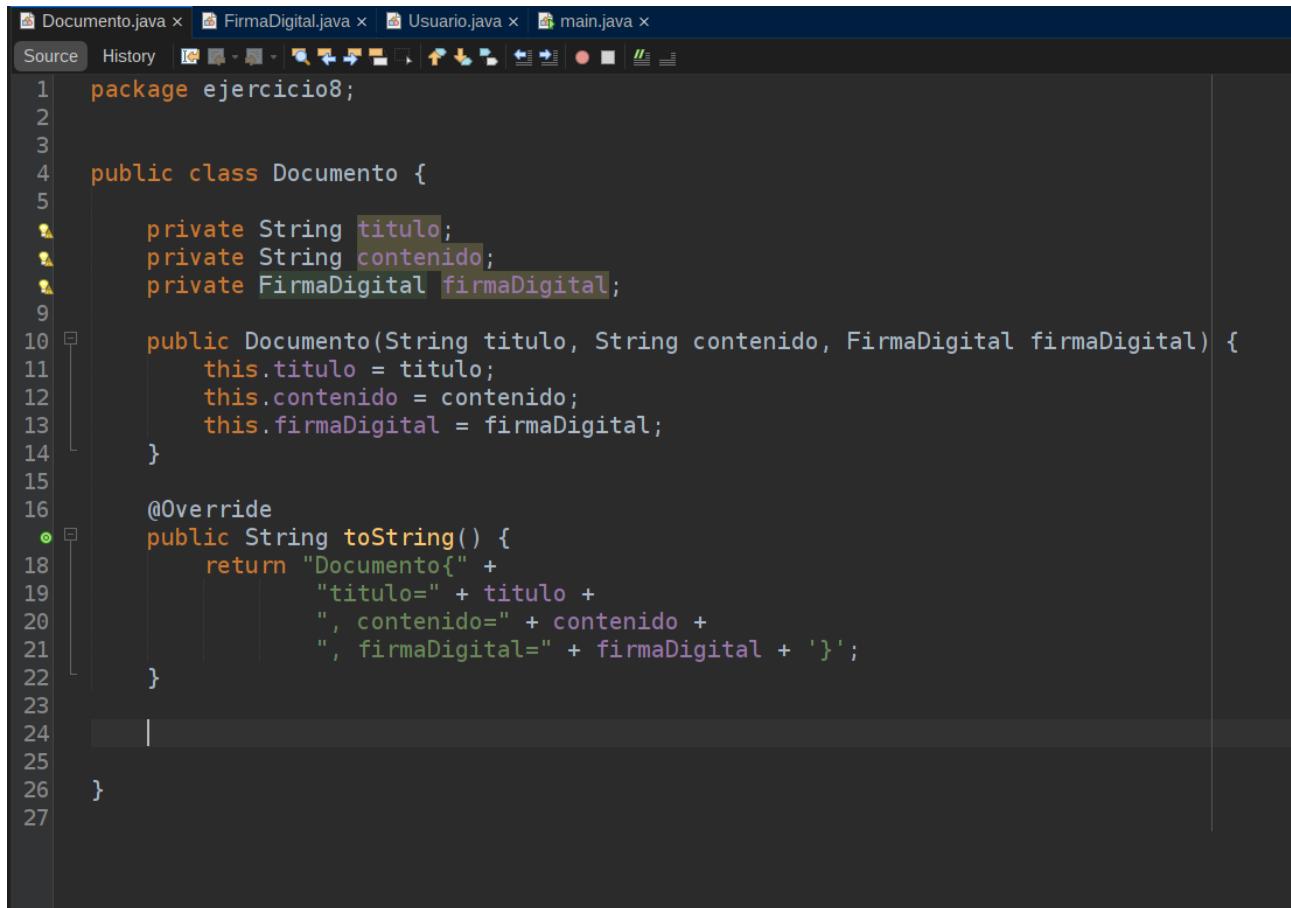
8. Documento - FirmaDigital - Usuario

- Composición: Documento → FirmaDigital
- Agregación: FirmaDigital → Usuario

Clases y atributos:

- Documento: titulo, contenido
- FirmaDigital: codigoHash, fecha
- Usuario: nombre, email



TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

The screenshot shows a Java code editor with the file `Documento.java` open. The code defines a class `Documento` with private attributes `titulo`, `contenido`, and `firmaDigital`. It includes a constructor and an overridden `toString` method. The code editor interface at the top shows tabs for other files: `Documento.java`, `FirmaDigital.java`, `Usuario.java`, and `main.java`. Below the tabs is a toolbar with various icons. The code itself is numbered from 1 to 27.

```
1 package ejercicio8;
2
3
4 public class Documento {
5
6     private String titulo;
7     private String contenido;
8     private FirmaDigital firmaDigital;
9
10    public Documento(String titulo, String contenido, FirmaDigital firmaDigital) {
11        this.titulo = titulo;
12        this.contenido = contenido;
13        this.firmaDigital = firmaDigital;
14    }
15
16    @Override
17    public String toString() {
18        return "Documento{" +
19            "titulo=" + titulo +
20            ", contenido=" + contenido +
21            ", firmaDigital=" + firmaDigital + '}';
22    }
23
24
25
26 }
27
```

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

The screenshot shows a Java code editor with the FirmaDigital.java file open. The code defines a class FirmaDigital with private attributes codigoHash, fecha, and usuario, and corresponding get and set methods. It also overrides the toString() method to return a string representation of the object's state.

```
3  public class FirmaDigital {  
4      private String codigoHash;  
5      private String fecha;  
6      private Usuario usuario;  
7  
8      public FirmaDigital(String codigoHash, String fecha, Usuario usuario) {  
9          this.codigoHash = codigoHash;  
10         this.fecha = fecha;  
11         this.usuario = usuario;  
12     }  
13  
14     public String getCodigoHash() {  
15         return codigoHash;  
16     }  
17  
18     public void setCodigoHash(String codigoHash) {  
19         this.codigoHash = codigoHash;  
20     }  
21  
22     public String getFecha() {  
23         return fecha;  
24     }  
25  
26     public void setFecha(String fecha) {  
27         this.fecha = fecha;  
28     }  
29  
30     public Usuario getUsuario() {  
31         return usuario;  
32     }  
33  
34     public void setUsuario(Usuario usuario) {  
35         this.usuario = usuario;  
36     }  
37  
38     @Override  
39     public String toString() {  
40         return "FirmaDigital{" +  
41             "codigoHash='" + codigoHash +  
42             "', fecha='" + fecha +  
43             "', usuario='" + usuario + "'}";  
44     }  
45  
46 }
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Screenshot of a Java code editor showing the `Usuario.java` file. The code defines a class `Usuario` with private attributes `nombre` and `email`, and methods for setting and getting these values. It also overrides the `toString` method.

```

1 package ejercicio8;
2
3
4 public class Usuario {
5
6
7     private String nombre;
8     private String email;
9
10    public Usuario(String nombre, String email) {
11        this.nombre = nombre;
12        this.email = email;
13    }
14
15
16    public String getNombre() {
17        return nombre;
18    }
19
20    public void setNombre(String nombre) {
21        this.nombre = nombre;
22    }
23
24    public String getEmail() {
25        return email;
26    }
27
28    public void setEmail(String email) {
29        this.email = email;
30    }
31
32
33    @Override
34    public String toString() {
35        return "Usuario{" + nombre + ", email=" + email + '}';
36    }
37
38 }
39

```

Screenshot of a Java code editor showing the `main.java` file. This is the entry point of the application, creating an `Usuario` object, a `FirmaDigital` object, and a `Documento` object, then printing the document's content.

```

1 package ejercicio8;
2
3
4 public class main {
5     public static void main(String[] args) {
6
7         Usuario usuario = new Usuario("Federico González", "fede@gmail.com");
8         FirmaDigital firma = new FirmaDigital("abc123hash", "2025-09-22", usuario);
9         Documento documento = new Documento("Contrato", "Contenido del contrato...", firma);
10
11         System.out.println(documento);
12     }
13
14 }
15

```

The left sidebar shows the project structure under `trabajoPractico5`, including source packages for exercises 1 through 8, and test packages for exercises 5 and 6.

Output - TrabajoPractico5 (run)

```

run:
Documento{titulo=Contrato, contenido=Contenido del contrato..., firmaDigital=FirmaDigital{codigoHash='abc123hash', fecha='2025-09-22', usuario=Usuario{nombre='Federico González', email='fede@gmail.com'}}}
BUILD SUCCESSFUL (total time: 0 seconds)

```

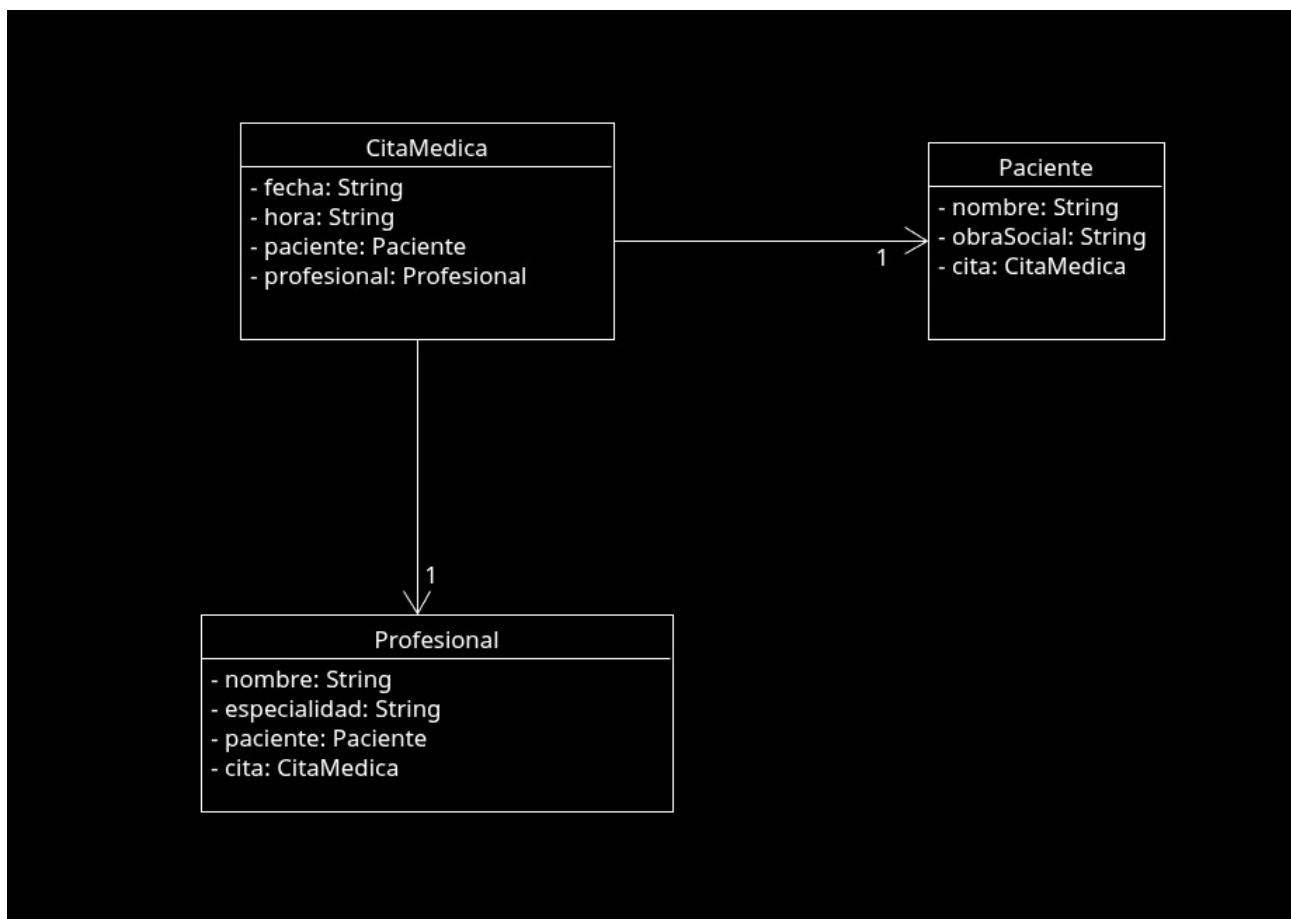
TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

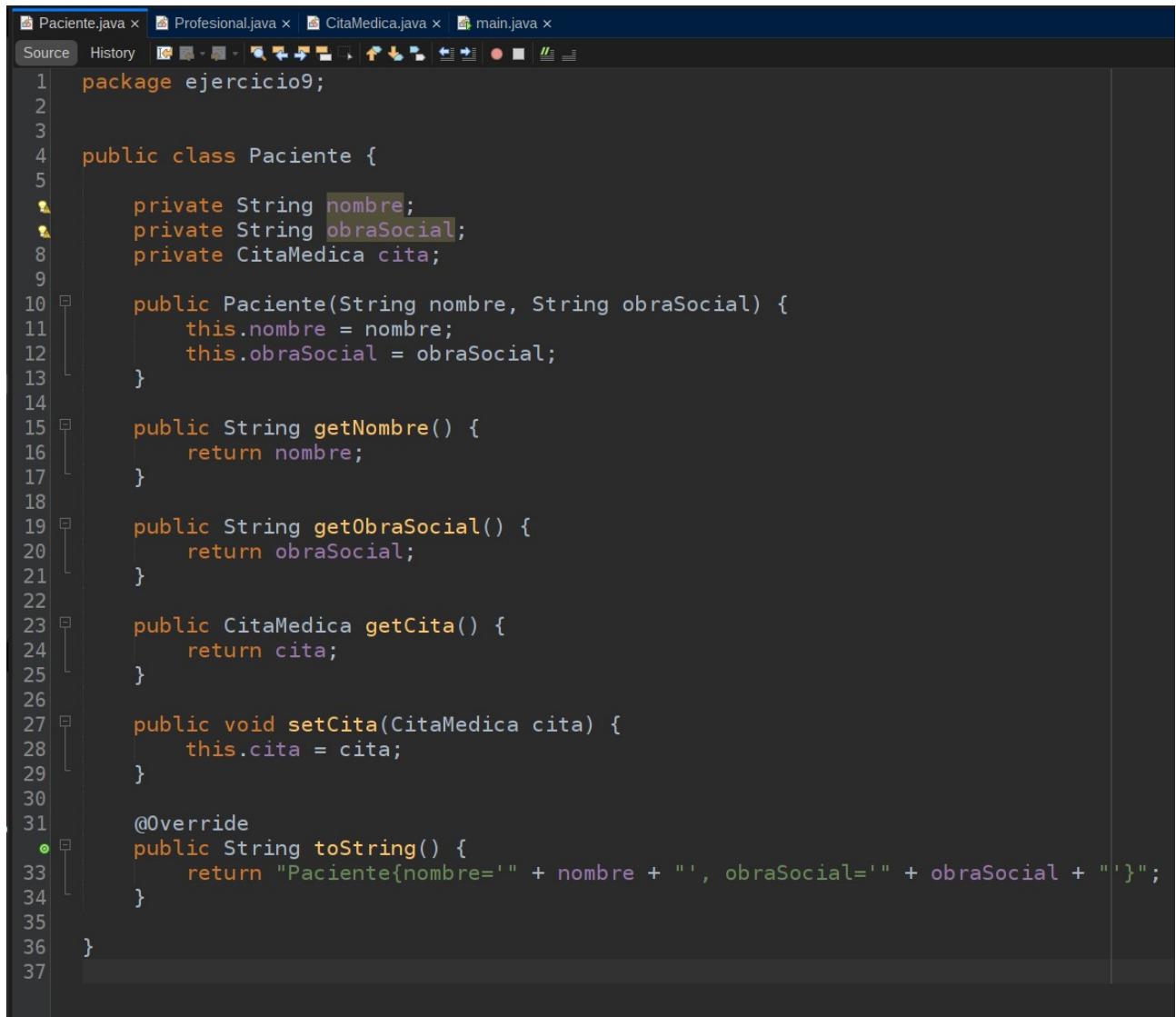
9. CitaMédica - Paciente - Profesional

- a. Asociación unidireccional: CitaMédica → Paciente,
- b. Asociación unidireccional: CitaMédica → Profesional

Clases y atributos:

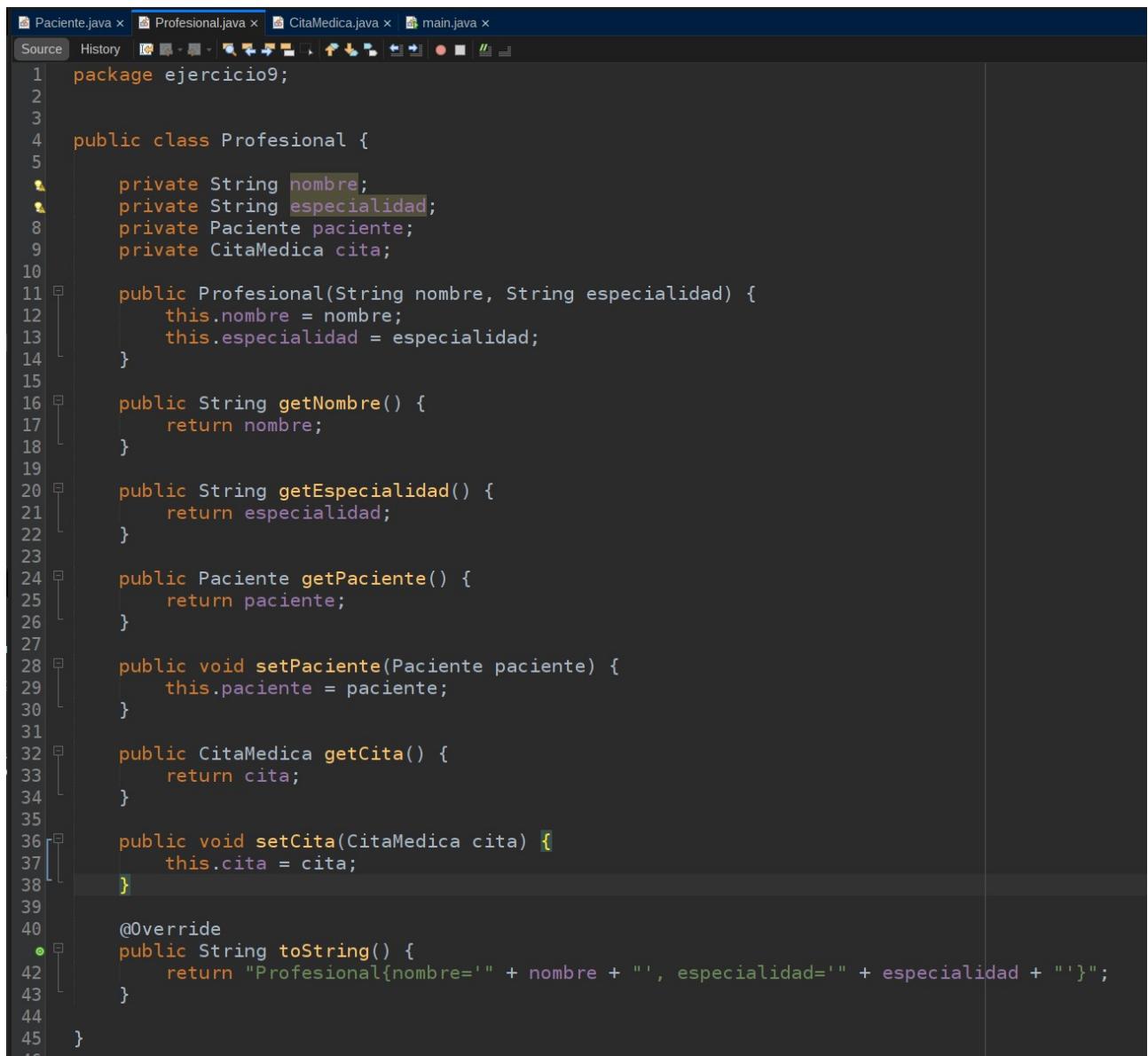
- i. CitaMédica: fecha, hora
- ii. Paciente: nombre, obraSocial
- iii. Profesional: nombre, especialidad



TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

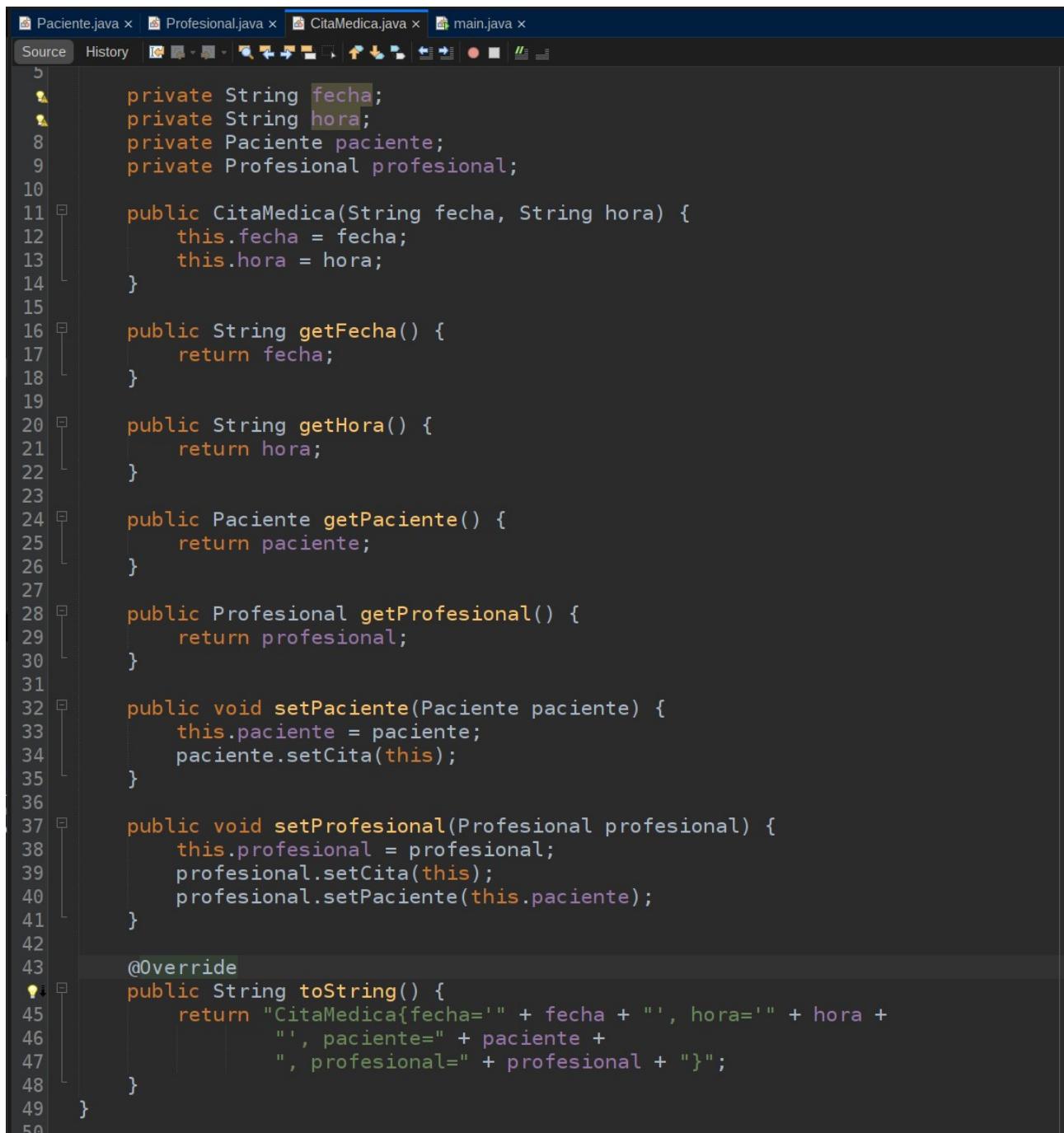
The screenshot shows a Java code editor with the file `Paciente.java` open. The code defines a class `Paciente` with private attributes `nombre` and `obraSocial`, and a reference to a `CitaMedica` object. It includes constructor, getter, and setter methods, and an overridden `toString` method.

```
1 package ejercicio9;
2
3
4 public class Paciente {
5
6     private String nombre;
7     private String obraSocial;
8     private CitaMedica cita;
9
10    public Paciente(String nombre, String obraSocial) {
11        this.nombre = nombre;
12        this.obraSocial = obraSocial;
13    }
14
15    public String getNombre() {
16        return nombre;
17    }
18
19    public String getObraSocial() {
20        return obraSocial;
21    }
22
23    public CitaMedica getCita() {
24        return cita;
25    }
26
27    public void setCita(CitaMedica cita) {
28        this.cita = cita;
29    }
30
31    @Override
32    public String toString() {
33        return "Paciente{" + nombre + ", obraSocial=" + obraSocial + '}';
34    }
35
36 }
37
```

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

The screenshot shows a Java code editor with the file `Profesional.java` open. The code defines a class `Profesional` with private attributes `nombre` and `especialidad`, and fields `paciente` and `cita`. It includes constructor, getters, setters, and an overridden `toString` method.

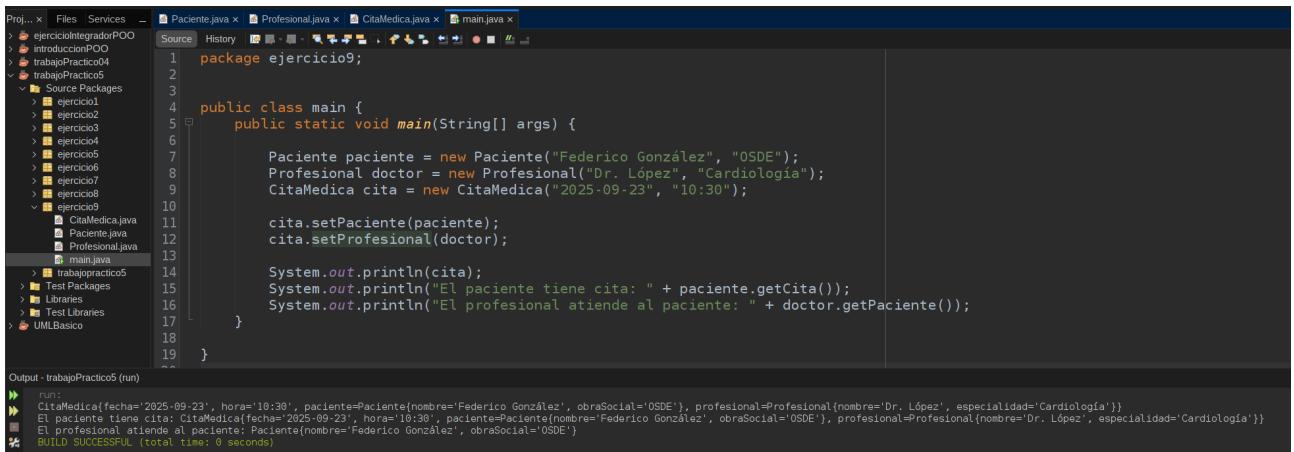
```
1 package ejercicio9;
2
3
4 public class Profesional {
5
6     private String nombre;
7     private String especialidad;
8     private Paciente paciente;
9     private CitaMedica cita;
10
11    public Profesional(String nombre, String especialidad) {
12        this.nombre = nombre;
13        this.especialidad = especialidad;
14    }
15
16    public String getNombre() {
17        return nombre;
18    }
19
20    public String getEspecialidad() {
21        return especialidad;
22    }
23
24    public Paciente getPaciente() {
25        return paciente;
26    }
27
28    public void setPaciente(Paciente paciente) {
29        this.paciente = paciente;
30    }
31
32    public CitaMedica getCita() {
33        return cita;
34    }
35
36    public void setCita(CitaMedica cita) {
37        this.cita = cita;
38    }
39
40    @Override
41    public String toString() {
42        return "Profesional{" + nombre + ", especialidad=" + especialidad + "}";
43    }
44}
45
46}
```

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

The screenshot shows a Java code editor with the file `CitaMedica.java` open. The code defines a class `CitaMedica` with attributes `fecha`, `hora`, `paciente`, and `profesional`. It includes constructors, getters, setters, and an overridden `toString` method. The code is color-coded for syntax highlighting.

```
5     private String fecha;
6     private String hora;
7     private Paciente paciente;
8     private Profesional profesional;
9
10    public CitaMedica(String fecha, String hora) {
11        this.fecha = fecha;
12        this.hora = hora;
13    }
14
15    public String getFecha() {
16        return fecha;
17    }
18
19    public String getHora() {
20        return hora;
21    }
22
23    public Paciente getPaciente() {
24        return paciente;
25    }
26
27    public Profesional getProfesional() {
28        return profesional;
29    }
30
31    public void setPaciente(Paciente paciente) {
32        this.paciente = paciente;
33        paciente.setCita(this);
34    }
35
36    public void setProfesional(Profesional profesional) {
37        this.profesional = profesional;
38        profesional.setCita(this);
39        profesional.setPaciente(this.paciente);
40    }
41
42    @Override
43    public String toString() {
44        return "CitaMedica{" + fecha + ", hora=" + hora +
45                  ", paciente=" + paciente +
46                  ", profesional=" + profesional + "}";
47    }
48}
49}
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



```

Proj... x Files Services - Paciente.java x Profesional.java x CitaMedica.java x main.java
Source History ... ...
1 package ejercicio9;
2
3
4 public class main {
5     public static void main(String[] args) {
6
7         Paciente paciente = new Paciente("Federico González", "OSDE");
8         Profesional doctor = new Profesional("Dr. López", "Cardiología");
9         CitaMedica cita = new CitaMedica("2025-09-23", "10:30");
10
11         cita.setPaciente(paciente);
12         cita.setProfesional(doctor);
13
14         System.out.println(cita);
15         System.out.println("El paciente tiene cita: " + paciente.getCita());
16         System.out.println("El profesional atiende al paciente: " + doctor.getpaciente());
17     }
18
19 }

```

Output - trabajoPractico5 (run)

```

▶ run:
▶ CitaMedica[fecha='2025-09-23', hora='10:30', paciente=Paciente(nombre='Federico González', obraSocial='OSDE'), profesional=Profesional(nombre='Dr. López', especialidad='Cardiología')]
El paciente tiene cita: CitaMedica[fecha='2025-09-23', hora='10:30', paciente=Paciente(nombre='Federico González', obraSocial='OSDE'), profesional=Profesional(nombre='Dr. López', especialidad='Cardiología')]
El profesional atiende al paciente: Paciente(nombre='Federico González', obraSocial='OSDE')
⌘ BUILD SUCCESSFUL (total time: 0 seconds)

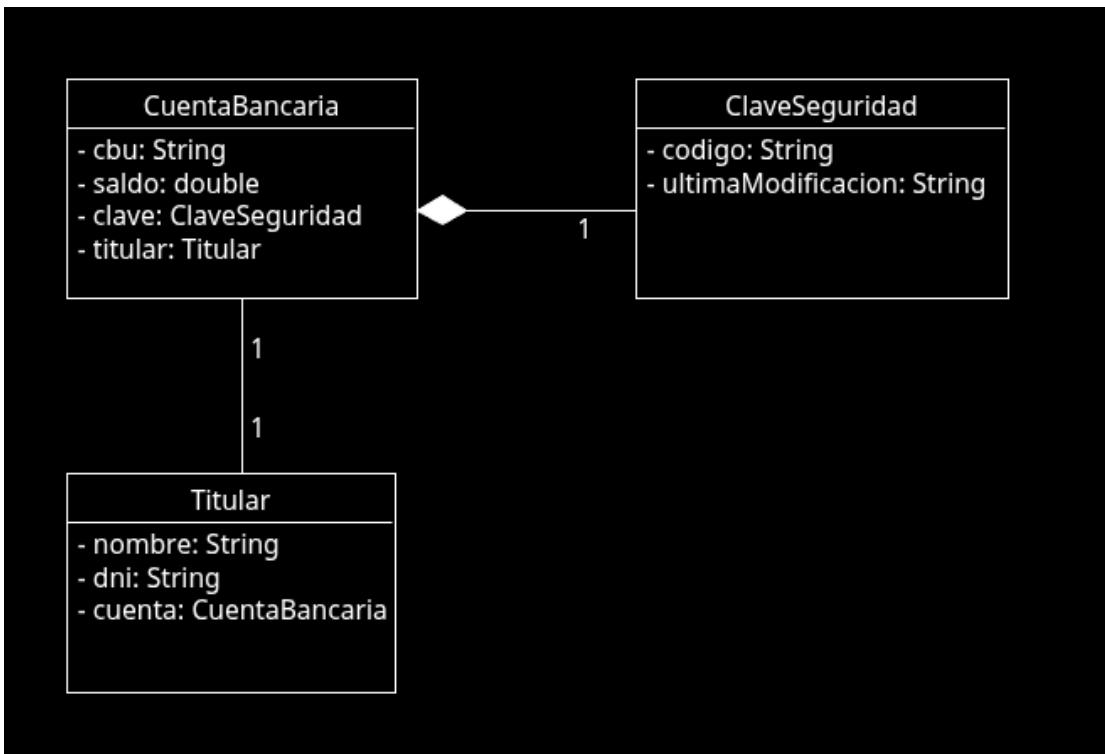
```

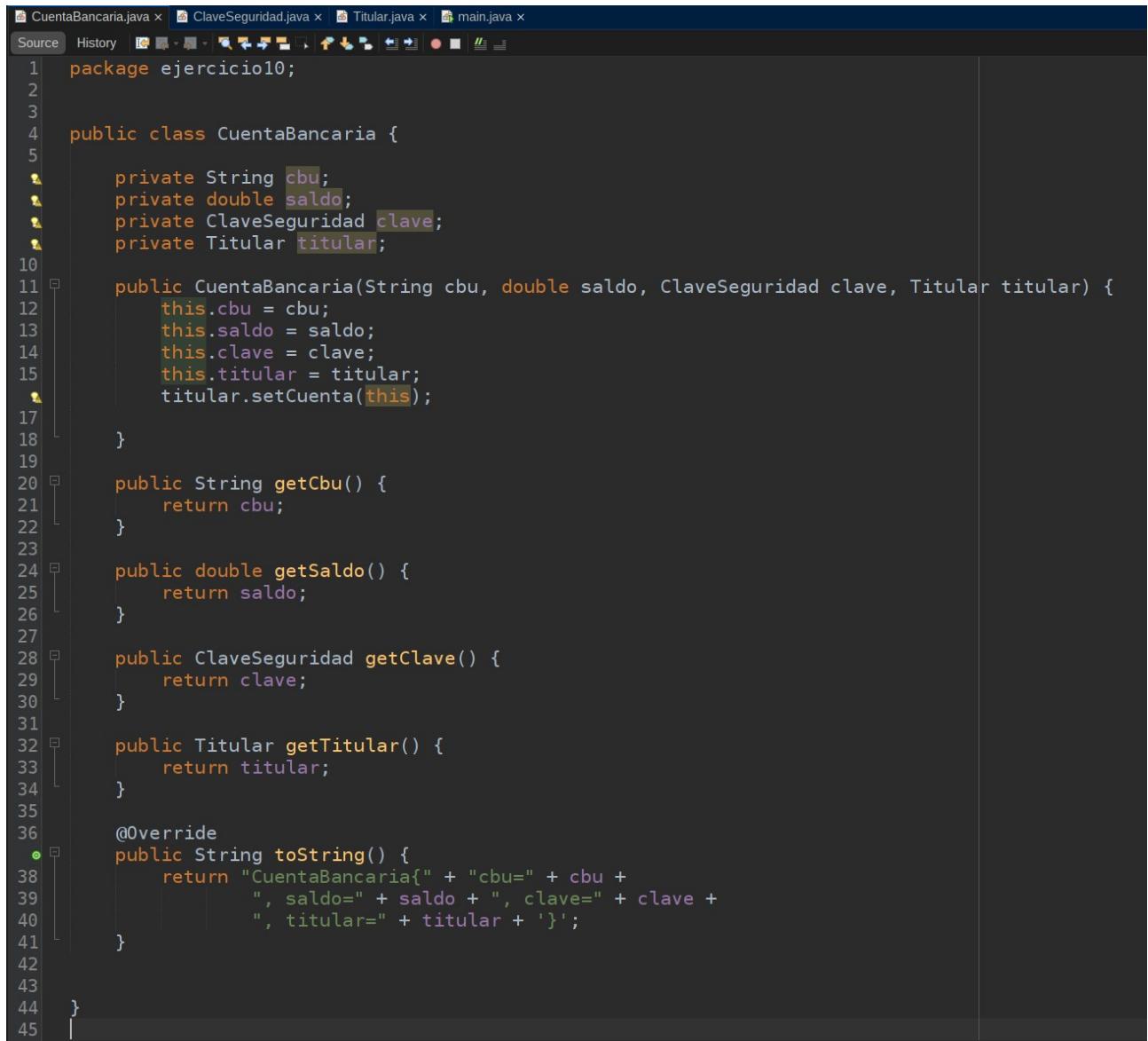
10. CuentaBancaria - ClaveSeguridad - Titular

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

Clases y atributos:

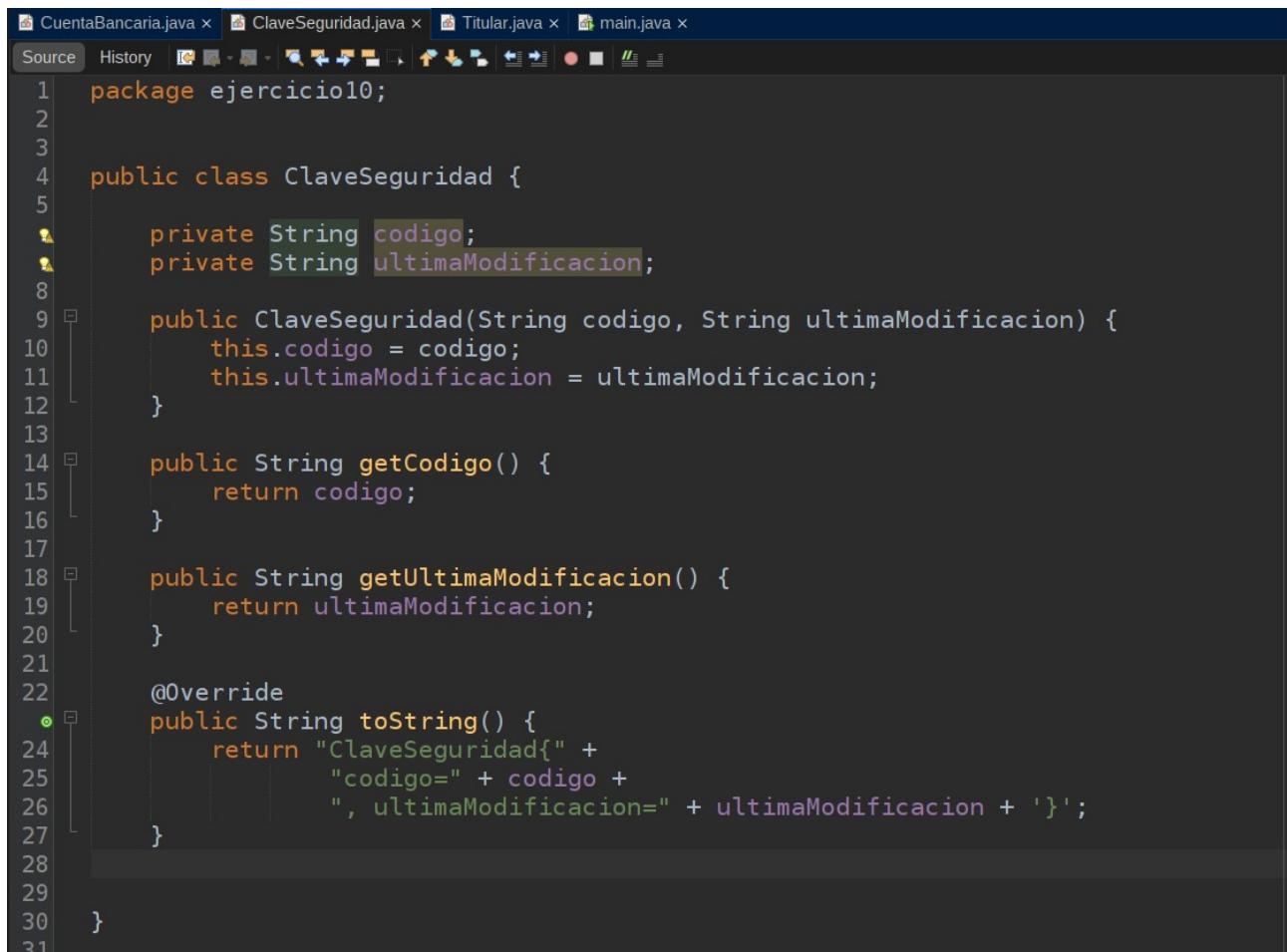
- CuentaBancaria: cbu, saldo
- ClaveSeguridad: codigo, ultimaModificacion
- Titular: nombre, dni.



TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

The screenshot shows a Java code editor with the file `CuentaBancaria.java` open. The code defines a class `CuentaBancaria` with fields `cbu`, `saldo`, `clave`, and `titular`. It includes a constructor, four getter methods, and an overridden `toString` method. The code editor interface at the top shows tabs for other files like `ClaveSeguridad.java` and `Titular.java`.

```
1 package ejercicio10;
2
3
4 public class CuentaBancaria {
5
6     private String cbu;
7     private double saldo;
8     private ClaveSeguridad clave;
9     private Titular titular;
10
11    public CuentaBancaria(String cbu, double saldo, ClaveSeguridad clave, Titular titular) {
12        this.cbu = cbu;
13        this.saldo = saldo;
14        this.clave = clave;
15        this.titular = titular;
16        titular.setCuenta(this);
17    }
18
19    public String getCbu() {
20        return cbu;
21    }
22
23    public double getSaldo() {
24        return saldo;
25    }
26
27    public ClaveSeguridad getClave() {
28        return clave;
29    }
30
31    public Titular getTitular() {
32        return titular;
33    }
34
35
36    @Override
37    public String toString() {
38        return "CuentaBancaria{" + "cbu=" + cbu +
39                    ", saldo=" + saldo + ", clave=" + clave +
40                    ", titular=" + titular + '}';
41    }
42
43
44}
45
```

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

The screenshot shows a Java code editor with the following code:

```
1 package ejercicio10;
2
3
4 public class ClaveSeguridad {
5
6     private String codigo;
7     private String ultimaModificacion;
8
9     public ClaveSeguridad(String codigo, String ultimaModificacion) {
10         this.codigo = codigo;
11         this.ultimaModificacion = ultimaModificacion;
12     }
13
14     public String getCodigo() {
15         return codigo;
16     }
17
18     public String getUltimaModificacion() {
19         return ultimaModificacion;
20     }
21
22     @Override
23     public String toString() {
24         return "ClaveSeguridad{" +
25                 "codigo=" + codigo +
26                 ", ultimaModificacion=" + ultimaModificacion + '}';
27     }
28
29
30 }
31
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

```
1 package ejercicio10;
2
3
4 public class Titular {
5
6     private String nombre;
7     private String dni;
8     private CuentaBancaria cuenta;
9
10    public Titular(String nombre, String dni) {
11        this.nombre = nombre;
12        this.dni = dni;
13    }
14
15    public String getNombre() {
16        return nombre;
17    }
18
19    public String getDni() {
20        return dni;
21    }
22
23    public CuentaBancaria getCuenta() {
24        return cuenta;
25    }
26
27    public void setCuenta(CuentaBancaria cuenta) {
28        this.cuenta = cuenta;
29    }
30
31
32    @Override
33    public String toString() {
34        return "Titular{" + "nombre=" + nombre +
35                  ", dni=" + dni + '}';
36    }
37
38}
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Screenshot of an IDE showing Java code for a banking application. The code creates a ClaveSeguridad object, a Titular object, and a CuentaBancaria object, then prints their details.

```

Proj... x Files Services ... CuentaBancaria.java x ClaveSeguridad.java x Titular.java x main.java x
Source History ... - + ...
1 package ejercicio10;
2
3
4 public class main {
5   public static void main(String[] args) {
6
7     ClaveSeguridad clave = new ClaveSeguridad("123ABC", "2025-09-22");
8     Titular titular = new Titular("Federico González", "34460724");
9     CuentaBancaria cuenta = new CuentaBancaria("000123456789", 15000.50, clave, titular);
10
11    System.out.println(cuenta);
12    System.out.println("El titular conoce su cuenta: " + titular.getNombre());
13  }
14
15 }
16

```

Output - trabajoPractico5 (run)

```

run:
CuentaBancaria(cbu=000123456789, saldo=15000.5, clave=ClaveSeguridad{codigo=123ABC, ultimaModificacion=2025-09-22}, titular=Titular{nombre=Federico González, dni=34460724})
El titular conoce su cuenta: CuentaBancaria(cbu=000123456789, saldo=15000.5, clave=ClaveSeguridad{codigo=123ABC, ultimaModificacion=2025-09-22}, titular=Titular{nombre=Federico González, dni=34460724})
BUILD SUCCESSFUL (total time: 0 seconds)

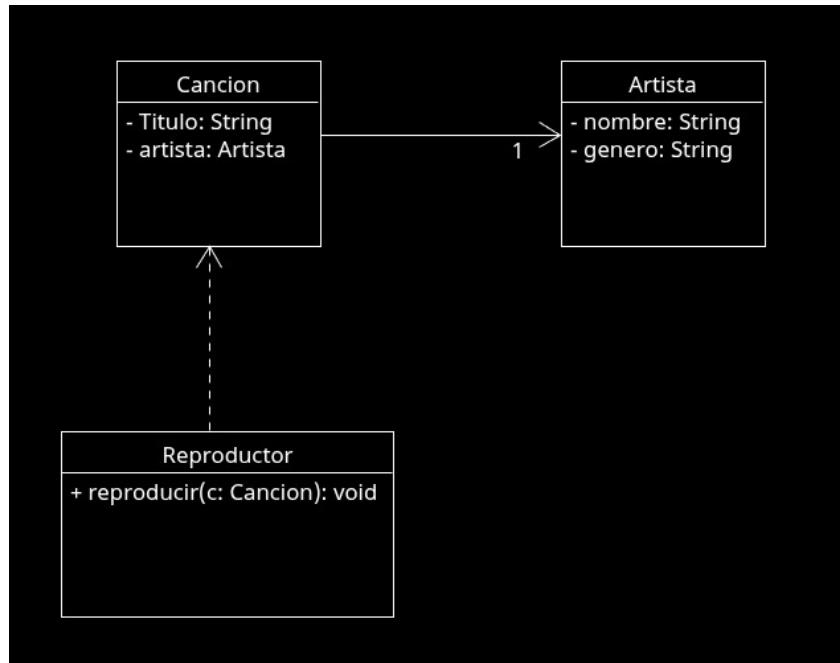
```

11. Reproductor - Canción - Artista

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

Clases y atributos:

- Canción: título.
- Artista: nombre, género.
- Reproductor->método: void reproducir(Cancion cancion)



TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Artista.java x Cancion.java x Reproductor.java x main.java x

Source History

```

1 package ejercicio11;
2
3
4 public class Artista {
5
6     private String nombre;
7     private String genero;
8
9     public Artista(String nombre, String genero) {
10         this.nombre = nombre;
11         this.genero = genero;
12     }
13
14     public String getNombre() {
15         return nombre;
16     }
17
18     public String getGenero() {
19         return genero;
20     }
21
22     @Override
23     public String toString() {
24         return "Artista{" + "nombre=" + nombre + ", genero=" + genero + '}';
25     }
26
27
28
29
30 }
```

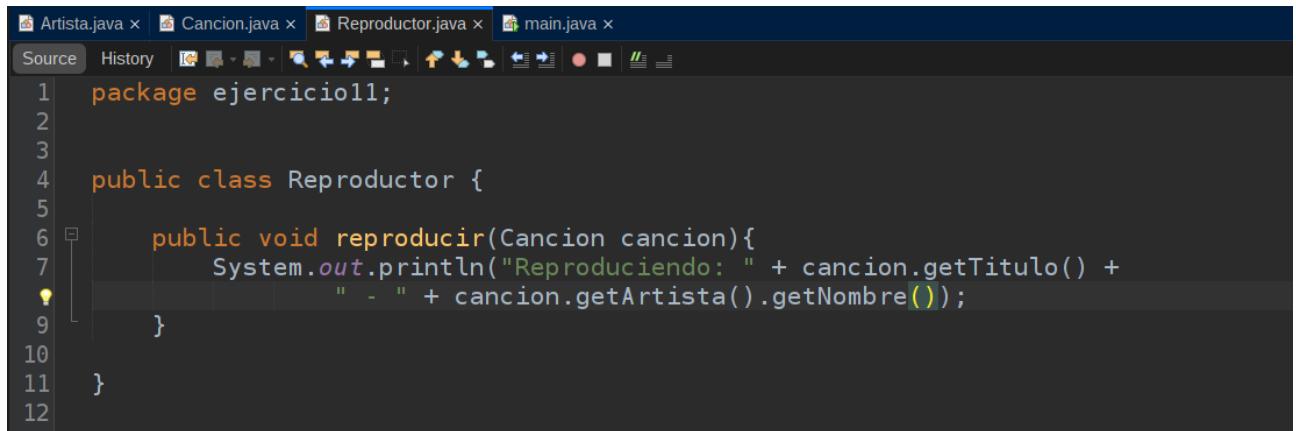
Artista.java x Cancion.java x Reproductor.java x main.java x

Source History

```

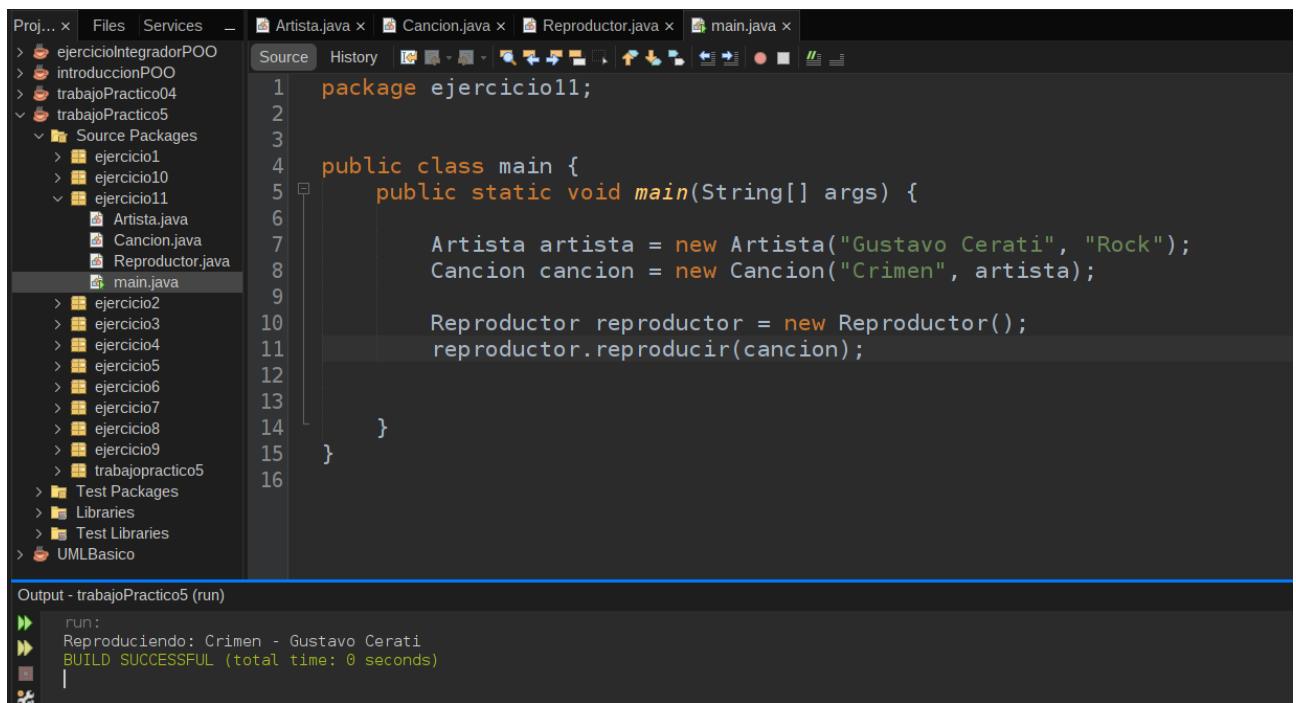
1 package ejercicio11;
2
3
4 public class Cancion {
5
6     private String titulo;
7     private Artista artista;
8
9     public Cancion(String titulo, Artista artista) {
10         this.titulo = titulo;
11         this.artista = artista;
12     }
13
14     public String getTitulo() {
15         return titulo;
16     }
17
18     public Artista getArtista() {
19         return artista;
20     }
21
22     @Override
23     public String toString() {
24         return "Cancion{" + "titulo=" + titulo + ", artista=" + artista + '}';
25     }
26
27
28
29
30 }
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



```

1 package ejercicio11;
2
3
4 public class Reproductor {
5
6     public void reproducir(Cancion cancion){
7         System.out.println("Reproduciendo: " + cancion.getTitulo() +
8                           " - " + cancion.getArtista().getNombre());
9     }
10
11 }
12
  
```



Project Explorer:

- ejercicioIntegradorPOO
- introduccionPOO
- trabajoPractico04
- trabajoPractico5
 - Source Packages
 - ejercicio1
 - ejercicio10
 - ejercicio11
 - Artista.java
 - Cancion.java
 - Reproductor.java
 - main.java
 - ejercicio2
 - ejercicio3
 - ejercicio4
 - ejercicio5
 - ejercicio6
 - ejercicio7
 - ejercicio8
 - ejercicio9
 - trabajopractico5
 - Test Packages
 - Libraries
 - Test Libraries
 - UMLBasico

Output - trabajoPractico5 (run)

```

run:
Reproduciendo: Crimen - Gustavo Cerati
BUILD SUCCESSFUL (total time: 0 seconds)
  
```

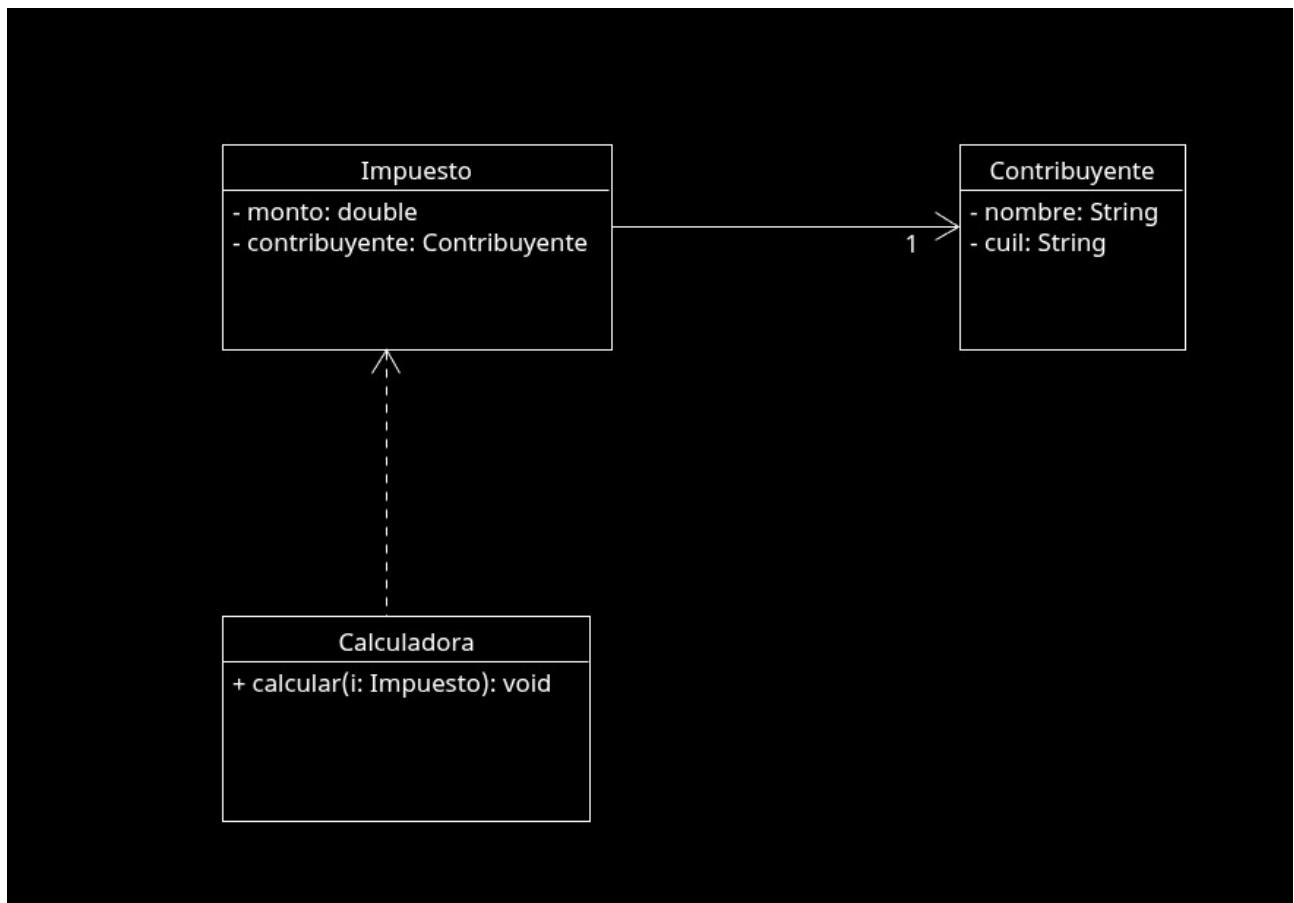
TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

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)



TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Contribuyente.java x Impuesto.java x Calculadora.java x main.java x

Source History

```

1 package ejercicio12;
2
3
4 public class Contribuyente {
5
6     private String nombre;
7     private String cuil;
8
9     public Contribuyente(String nombre, String cuil) {
10         this.nombre = nombre;
11         this.cuil = cuil;
12     }
13
14     public String getNombre() {
15         return nombre;
16     }
17
18     public String getCUIL() {
19         return cuil;
20     }
21
22     @Override
23     public String toString() {
24         return "Contribuyente{" + "nombre=" + nombre + ", cuil=" + cuil + '}';
25     }
26
27
28 }
29

```

Contribuyente.java x Impuesto.java x Calculadora.java x main.java x

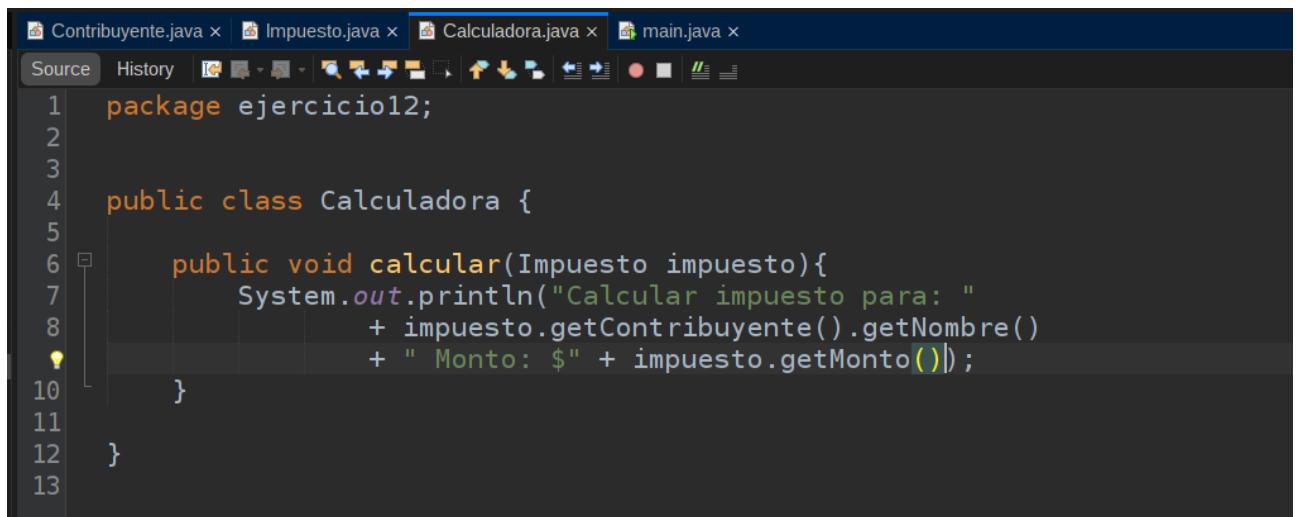
Source History

```

1 package ejercicio12;
2
3
4 public class Impuesto {
5
6     private double monto;
7     private Contribuyente contribuyente;
8
9     public Impuesto(double monto, Contribuyente contribuyente) {
10         this.monto = monto;
11         this.contribuyente = contribuyente;
12     }
13
14     public double getMonto() {
15         return monto;
16     }
17
18     public Contribuyente getContribuyente() {
19         return contribuyente;
20     }
21
22     @Override
23     public String toString() {
24         return "Impuesto{" + "monto=" + monto + ", contribuyente=" + contribuyente + '}';
25     }
26
27
28 }
29

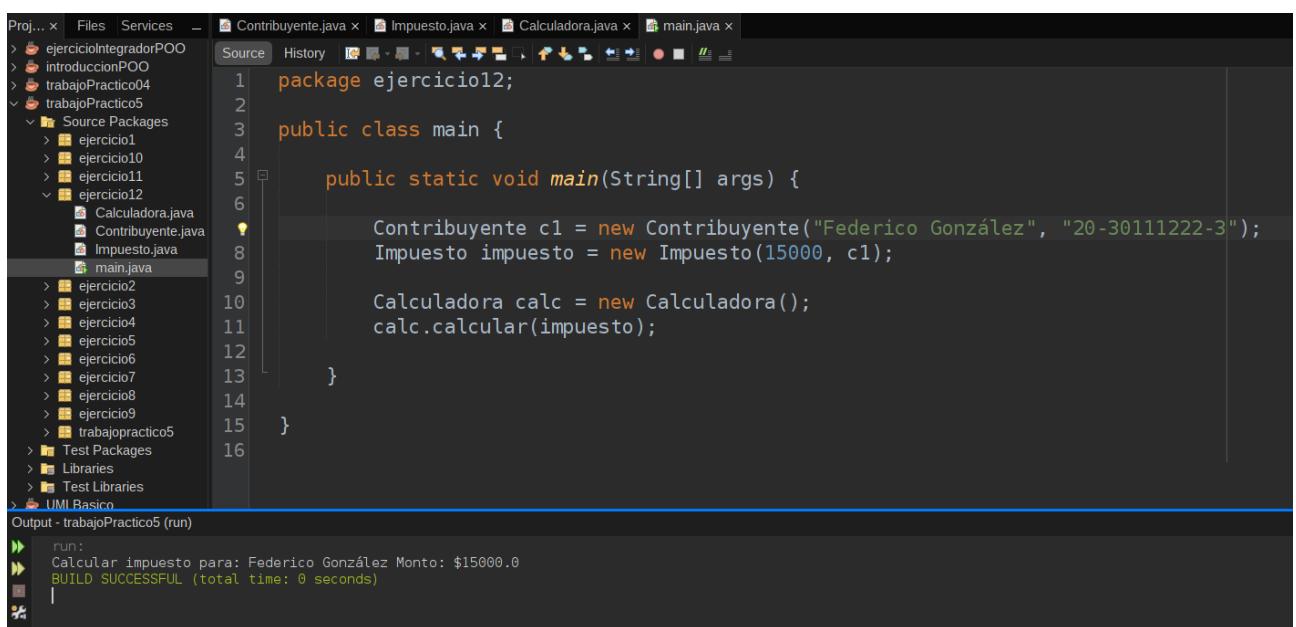
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



```

1 package ejercicio12;
2
3
4 public class Calculadora {
5
6     public void calcular(Impuesto impuesto){
7         System.out.println("Calcular impuesto para: "
8                         + impuesto.getContribuyente().getNombre()
9                         + " Monto: $" + impuesto.getMonto());
10    }
11
12 }
13
  
```



Project Explorer:

- ejercicioIntegradorPOO
- introduccionPOO
- trabajoPractico04
- trabajoPractico5
 - Source Packages
 - ejercicio1
 - ejercicio10
 - ejercicio11
 - ejercicio12
 - Calculadora.java
 - Contribuyente.java
 - Impuesto.java
 - main.java
 - ejercicio2
 - ejercicio3
 - ejercicio4
 - ejercicio5
 - ejercicio6
 - ejercicio7
 - ejercicio8
 - ejercicio9
 - trabajopractico5
 - Test Packages
 - Libraries
 - Test Libraries
- UML Basico

Output - trabajoPractico5 (run)

```

run:
Calcular impuesto para: Federico González Monto: $15000.0
BUILD SUCCESSFUL (total time: 0 seconds)
| 
```

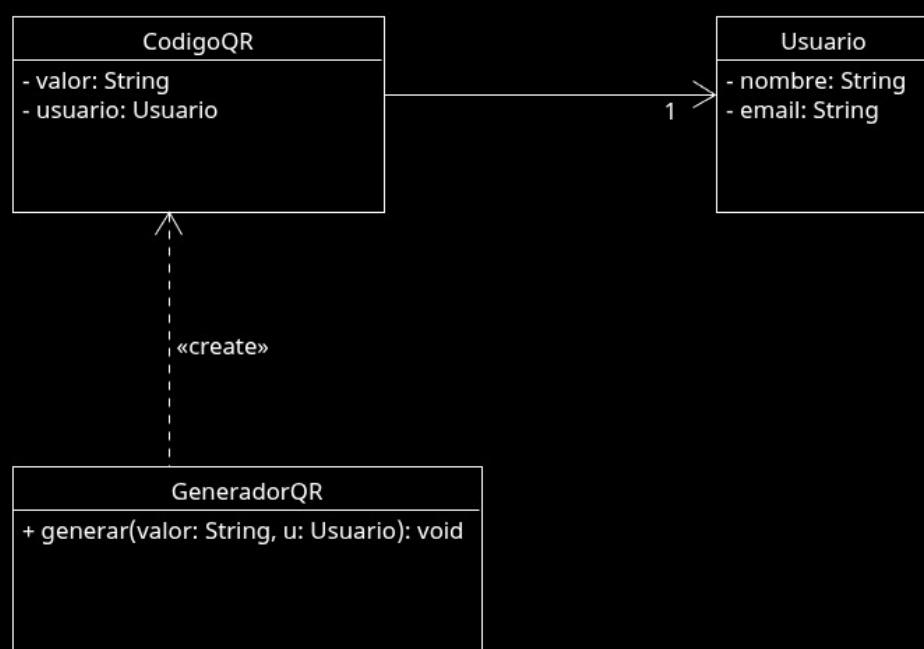
TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

13. GeneradorQR - Usuario - CódigoQR

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

Clases y atributos:

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



TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Usuario.java x CódigoQR.java x GeneradorQR.java x main.java x

Source History

```

1 package ejercicio13;
2
3
4 public class Usuario {
5
6     private String nombre;
7     private String email;
8
9     public Usuario(String nombre, String email) {
10         this.nombre = nombre;
11         this.email = email;
12     }
13
14     public String getNombre() {
15         return nombre;
16     }
17
18     public String getEmail() {
19         return email;
20     }
21
22     @Override
23     public String toString() {
24         return "Usuario{" + "nombre=" + nombre + ", email=" + email + '}';
25     }
26
27
28
29 }
```

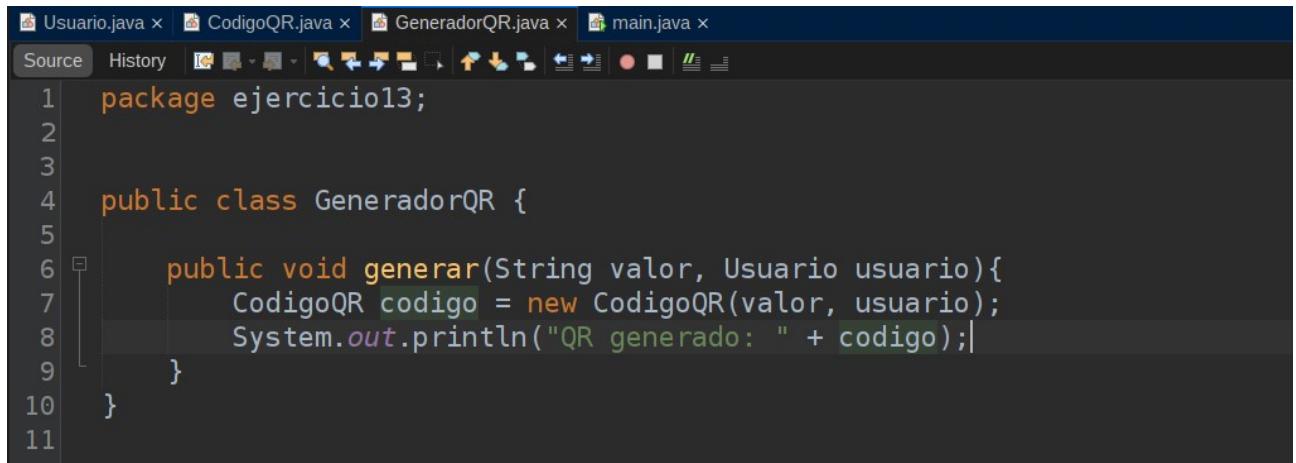
Usuario.java x CódigoQR.java x GeneradorQR.java x main.java x

Source History

```

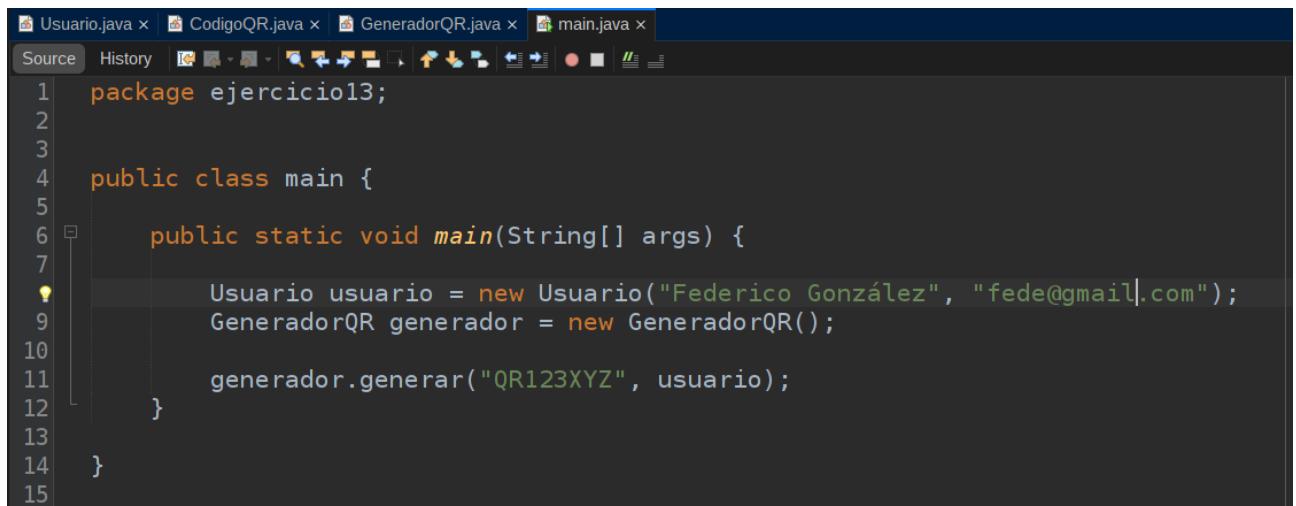
1 package ejercicio13;
2
3
4 public class CódigoQR {
5
6     private String valor;
7     private Usuario usuario;
8
9     public CódigoQR(String valor, Usuario usuario) {
10         this.valor = valor;
11         this.usuario = usuario;
12     }
13
14     public String getValor() {
15         return valor;
16     }
17
18     public Usuario getUsuario() {
19         return usuario;
20     }
21
22     @Override
23     public String toString() {
24         return "CódigoQR{" + "valor=" + valor + ", usuario=" + usuario + '}';
25     }
26
27
28
29 }
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



```

1 package ejercicio13;
2
3
4 public class GeneradorQR {
5
6     public void generar(String valor, Usuario usuario){
7         CódigoQR codigo = new CódigoQR(valor, usuario);
8         System.out.println("QR generado: " + codigo);
9     }
10 }
11
  
```

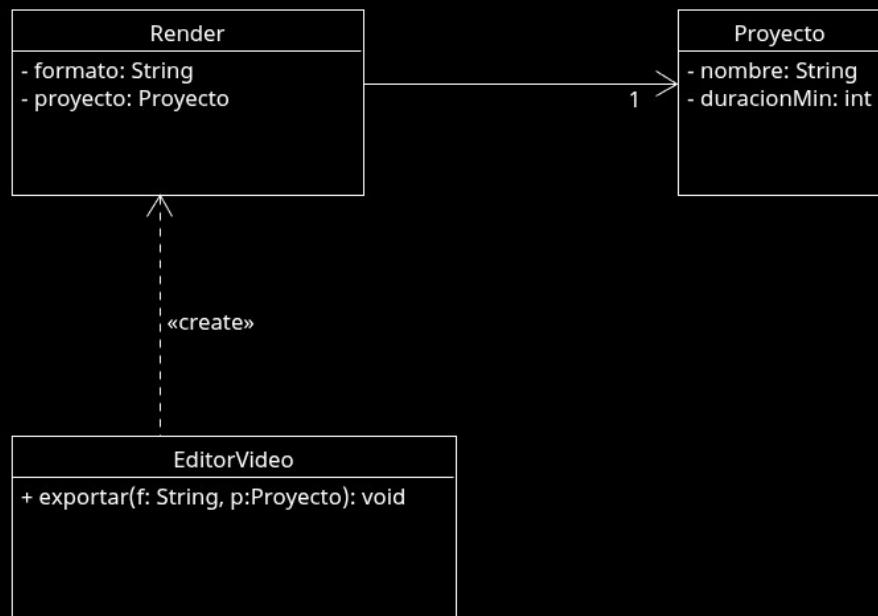
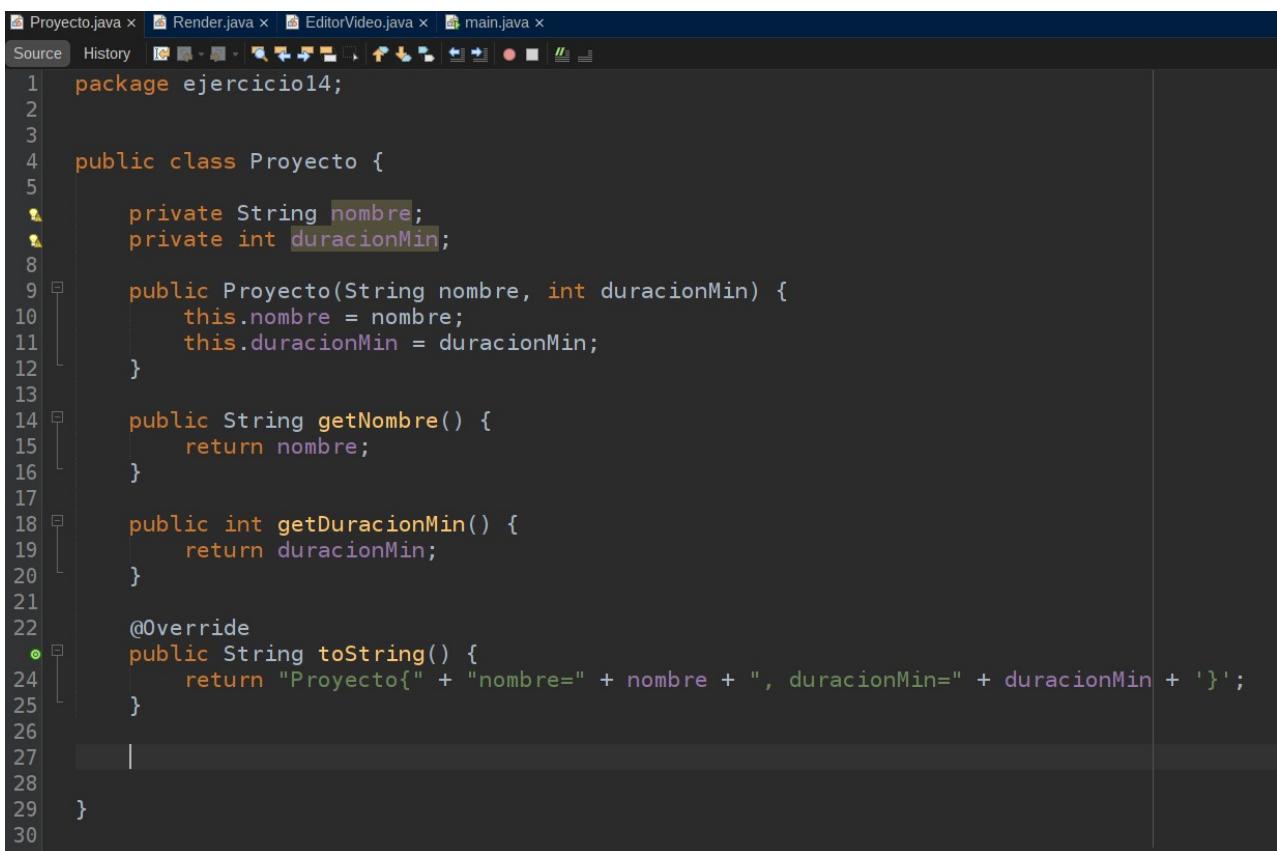


```

1 package ejercicio13;
2
3
4 public class main {
5
6     public static void main(String[] args) {
7
8         Usuario usuario = new Usuario("Federico González", "fede@gmail.com");
9         GeneradorQR generador = new GeneradorQR();
10
11         generador.generar("QR123XYZ", usuario);
12     }
13
14 }
15
  
```

14. EditorVideo - Proyecto - Render

- a. Asociación unidireccional: Render → Proyecto
- b. Dependencia de creación: EditorVideo.exportar(String, Proyecto)
- c. Clases y atributos:
 - i. Render: formato.
 - ii. Proyecto: nombre, duracionMin.
 - iii. Reproductor → EditorVideo->método: void exportar(String formato, Proyecto proyecto)

TECNICATURA UNIVERSITARIA EN
 PROGRAMACIÓN A DISTANCIA



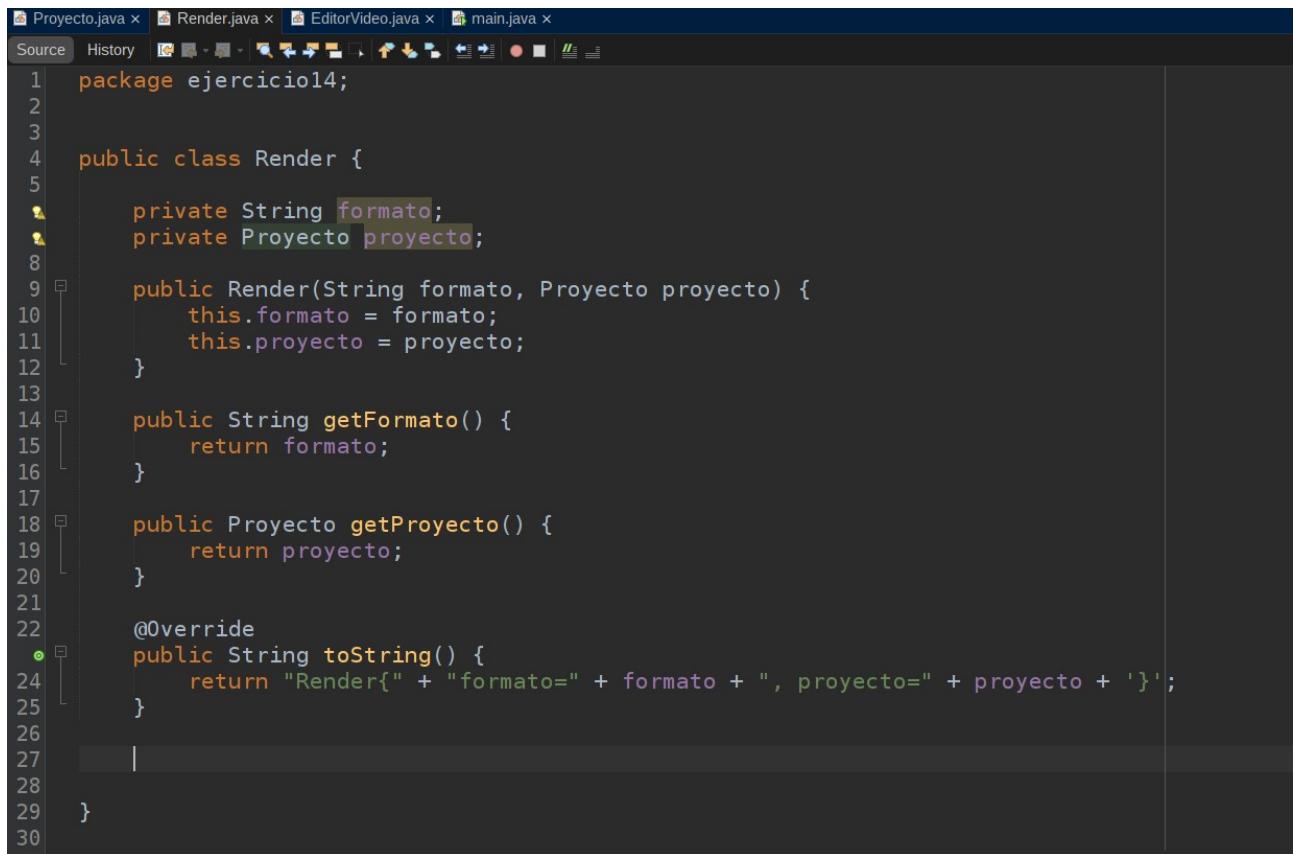
```

package ejercicio14;

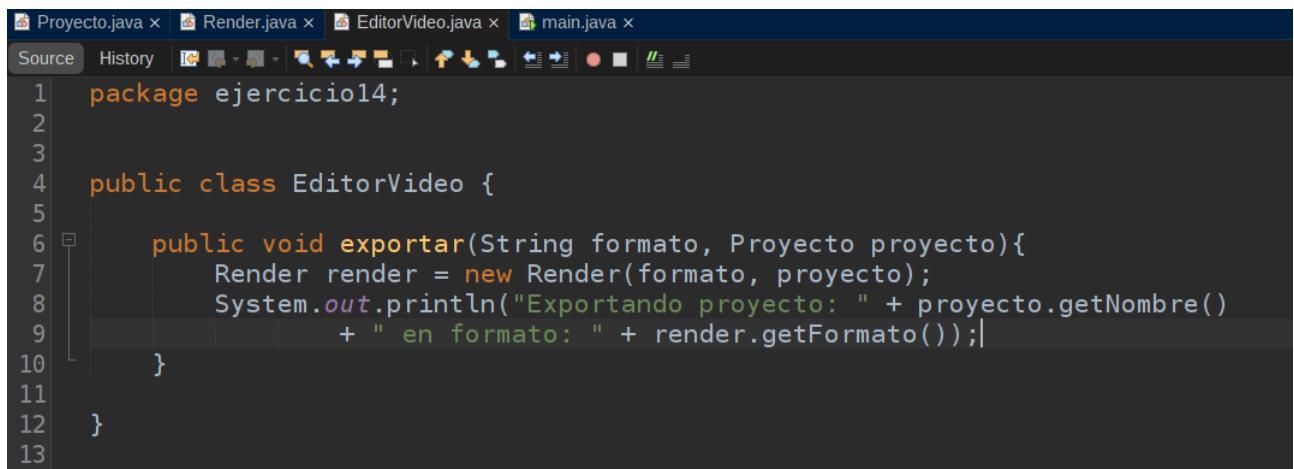
public class Proyecto {
    private String nombre;
    private int duracionMin;
    public Proyecto(String nombre, int duracionMin) {
        this.nombre = nombre;
        this.duracionMin = duracionMin;
    }
    public String getNombre() {
        return nombre;
    }
    public int getDuracionMin() {
        return duracionMin;
    }
    @Override
    public String toString() {
        return "Proyecto{" + "nombre=" + nombre + ", duracionMin=" + duracionMin + '}';
    }
}
  
```

The screenshot shows a Java code editor with the `Projeto.java` file open. The code defines a class `Proyecto` with private attributes `nombre` and `duracionMin`, a constructor, and methods to get each attribute. It also overrides the `toString` method to return a string representation of the object.

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

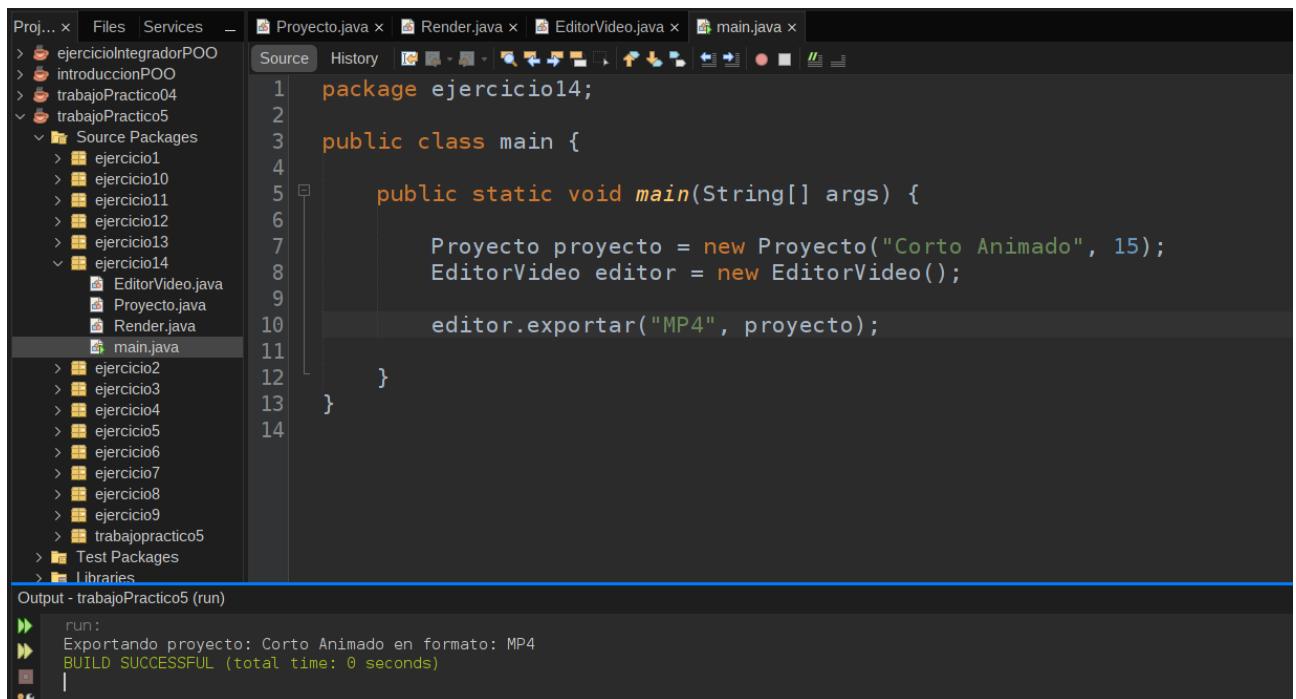


```
1 package ejercicio14;
2
3
4 public class Render {
5
6     private String formato;
7     private Proyecto proyecto;
8
9     public Render(String formato, Proyecto proyecto) {
10         this.formato = formato;
11         this.proyecto = proyecto;
12     }
13
14     public String getFormato() {
15         return formato;
16     }
17
18     public Proyecto getProyecto() {
19         return proyecto;
20     }
21
22     @Override
23     public String toString() {
24         return "Render{" + "formato=" + formato + ", proyecto=" + proyecto + '}';
25     }
26
27
28 }
29 }
```



```
1 package ejercicio14;
2
3
4 public class EditorVideo {
5
6     public void exportar(String formato, Proyecto proyecto){
7         Render render = new Render(formato, proyecto);
8         System.out.println("Exportando proyecto: " + proyecto.getNombre()
9                             + " en formato: " + render.getFormato());
10    }
11
12 }
13 }
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



The screenshot shows a Java development environment with the following details:

- Project Explorer (Left):** Shows a project structure with packages like ejercicioIntegradorPOO, introduccionPOO, trabajoPractico04, and trabajoPractico5. Under trabajoPractico5, there are Source Packages (ejercicio1 through ejercicio14) and Test Packages.
- Code Editor (Center):** Displays the main.java file with the following code:

```
package ejercicio14;
public class main {
    public static void main(String[] args) {
        Proyecto proyecto = new Proyecto("Corto Animado", 15);
        EditorVideo editor = new EditorVideo();
        editor.exportar("MP4", proyecto);
    }
}
```
- Output Window (Bottom):** Shows the results of a run operation:

```
run:
Exportando proyecto: Corto Animado en formato: MP4
BUILD SUCCESSFUL (total time: 0 seconds)
```

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA