UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

Trabajo Práctico N.º 5: Relaciones UML

Estudiante: Emilia Gómez Juárez

Objetivo:

Modelar clases con relaciones 1 a 1 utilizando diagramas UML. Identificar correctamente el tipo de relación (asociación, agregación, composición, dependencia) y su dirección, y llevarlas a implementación en Java.

Caso Práctico:

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

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

Ejercicios de relaciones 1 a 1:

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

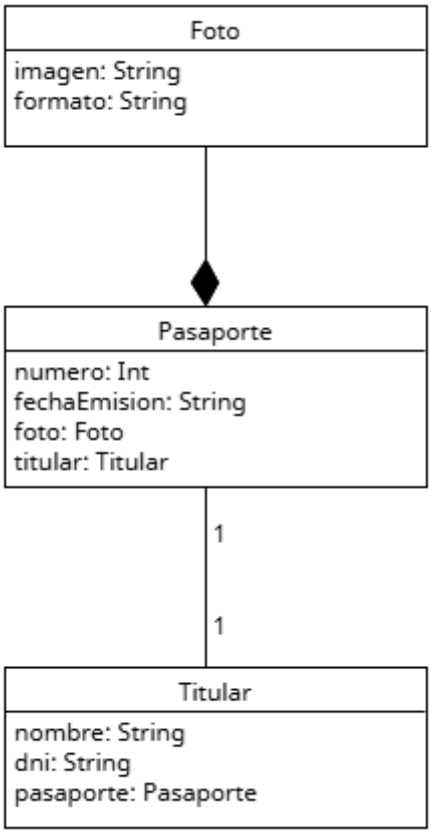
Clases y atributos:

I. Pasaporte: numero, fechaEmision
II. Foto: imagen, formato
III. Titular: nombre, dni

Diagrama UML - Tipo de relación y dirección

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II



Implementación de clases con atributos y relaciones definidas

Clase Foto:

Emilia Gómez Juárez
2025

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

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

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

    public String getImagen() {
        return imagen;
    }

    public void setImagen(String imagen) {
        this.imagen = imagen;
    }

    public String getFormato() {
        return formato;
    }

    public void setFormato(String formato) {
        this.formato = formato;
    }

    @Override
    public String toString() {
        return "Foto{" + "imagen=" + imagen +
                ", formato=" + formato + '}';
    }
}
```

Clase Pasaporte:

Emilia Gómez Juárez
2025

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

```java
public class Pasaporte {
    private int numero;
    private String fechaEmision;
    private Foto foto;
    private Titular titular;

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

    public int getNumero() {
        return numero;
    }

    public void setNumero(int numero) {
        this.numero = numero;
    }

    public String getFechaEmision() {
        return fechaEmision;
    }

    public void setFechaEmision(String fechaEmision) {
        this.fechaEmision = fechaEmision;
    }
```

```java
    public Foto getFoto() {
        return foto;
    }

    public void setFoto(Foto foto) {
        this.foto = foto;
    }

    public Titular getTitular() {
        return titular;
    }

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

    }

    @Override
    public String toString() {
        return "Pasaporte{" + "numero=" + numero +
                ", fechaEmision=" + fechaEmision +
                ", foto=" + foto +
                ", titular=" + (titular != null ? titular.getDni() : "null") + '}';
    }
```

Emilia Gómez Juárez
2025

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

Clase Titular:

```java
public class Titular {
    private String nombre;
    private String dni;
    private Pasaporte pasaporte;

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

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getDni() {
        return dni;
    }

    public void setDni(String dni) {
        this.dni = dni;
    }

```

```java
    public void setPasaporte(Pasaporte pasaporte) {
        if (this.pasaporte != pasaporte){
            this.pasaporte = pasaporte;
            if (pasaporte != null && this.pasaporte.getTitular() != this)
                pasaporte.setTitular(this);
            }

    }

    public Pasaporte getPasaporte() {
        return pasaporte;
    }

    @Override
    public String toString() {
        return "Titular{" + "nombre=" + nombre +
                ", dni=" + dni +
                ", pasaporte="  + (pasaporte != null ? pasaporte.getNumero() : "null") +
            '}';
    }
}

```

Emilia Gómez Juárez
2025

Implementación en main:

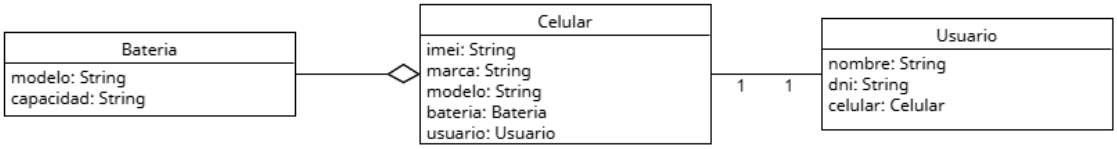
```java
public class MainEj1 {
    public static void main(String[] args) {
        Pasaporte pasaporte = new Pasaporte (12345678, "04/09/2010", "foto_titular.jpg");

        Titular titular = new Titular ("Pedro Perez", "12345678");

        titular.setPasaporte(pasaporte);

        System.out.println(titular.toString());

    }
}
```

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

Diagrama UML - Tipo de relación y dirección



Implementación de clases con atributos y relaciones definidas

Clase batería:

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

```java
public class Bateria {
    private String modelo;
    private String capacidad;

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

    @Override
    public String toString() {
        return "Bateria{" + "modelo=" + modelo +
                ", capacidad=" + capacidad + '}';
    }
}
```

Clase Celular:

Emilia Gómez Juárez
2025

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

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

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


    public Bateria getBateria() {
        return bateria;
    }

    public void setBateria(Bateria bateria) {
        this.bateria = bateria;
    }

    public String getImei() {
        return imei;
    }

    public void setImei(String imei) {
        this.imei = imei;
    }
```

Emilia Gómez Juárez
2025

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

```java
    public String getMarca() {
        return marca;
    }

    public void setMarca(String marca) {
        this.marca = marca;
    }

    public void setModelo(String modelo) {
        this.modelo = modelo;
    }

    public String getModelo() {
        return modelo;
    }

    public Usuario getUsuario() {
        return usuario;
    }

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

```java

    @Override
    public String toString() {
        return "Celular{" + "imei=" + imei +
                ", marca=" + marca +
                ", modelo=" + modelo +
                ", bateria=" + bateria +
                ", usuario=" + (usuario != null ? usuario.getNombre() : "null") + '}';
    }

```

Clase Usuario:

Emilia Gómez Juárez
2025

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

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

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

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getDni() {
        return dni;
    }

    public void setDni(String dni) {
        this.dni = dni;
    }

```

```java
    public Celular getCelular() {
        return celular;
    }

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

    @Override
    public String toString() {
        return "Usuario{" + "nombre=" + nombre +
                ", dni=" + dni +
                ", celular=" + (celular != null ? celular.getModelo() : "null") + '}';
    }
}
```

Implementación en main:

Emilia Gómez Juárez
2025

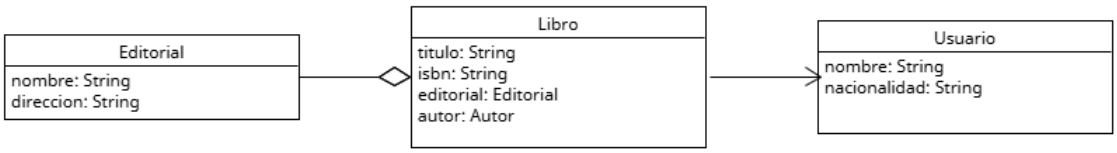
```java
public class MainEj2 {
    public static void main(String[] args) {
        Bateria bateria = new Bateria ("M6 Pro", "5000 mAh");

        Celular celular = new Celular("123456789", "Motorola", "MotoG22");
        celular.setBateria(bateria);

        Usuario usuario = new Usuario ("Emilia Gómez", "43153023");

        usuario.setCelular(celular);

        System.out.println(celular.toString());
    }
}
```

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

Diagrama UML - Tipo de relación y dirección



Implementación de clases con atributos y relaciones definidas

Clase Editorial

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

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

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

    public String getNombre() {return nombre;}

    public void setNombre(String nombre) {this.nombre = nombre;}

    public String getDireccion() {return direccion;}

    public void setDireccion(String direccion) {this.direccion = direccion;}

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

Clase Autor

Emilia Gómez Juárez
2025

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

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

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

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getNacionalidad() {
        return nacionalidad;
    }

    public void setNacionalidad(String nacionalidad) {
        this.nacionalidad = nacionalidad;
    }

    @Override
    public String toString() {
        return "Autor{" +
                "nombre='" + nombre + '\'' +
                ", nacionalidad='" + nacionalidad + '\'' +
                '\'';
```

Clase Libro

Emilia Gómez Juárez
2025

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

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

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

    public String getTitulo() {
        return titulo;
    }

    public void setTitulo(String titulo) {
        this.titulo = titulo;
    }

    public String getIsbn() {
        return isbn;
    }

    public void setIsbn(String isbn) {
        this.isbn = isbn;
    }

    public Editorial getEditorial() {
        return editorial;
    }
```

Emilia Gómez Juárez
2025

```
41
42      public void setEditorial(Editorial editorial) {
43          this.editorial = editorial;
44      }
45
46      public Autor getAutor() {
47          return autor;
48      }
49
50      public void setAutor(Autor autor) {
51          this.autor = autor;
52      }
53
54      @Override
        public String toString() {
56          return "Libro{" + "titulo=" + titulo +
57                  ", isbn=" + isbn +
58                  ", editorial=" + editorial +
59                  ", autor=" + autor + '}';
60      }
61  }
62
```

Implementación en Main:

```
11  public class MainEj3 {
12      public static void main(String[] args) {
13          Editorial editorial = new Editorial("Planeta", "Av. Corrientes 1234");
14
15          Autor autor = new Autor("Gabriel García Márquez", "Colombiano");
16
17          Libro libro = new Libro("Cien Años de Soledad", "978-84-376-0494-7");
18
19
20          libro.setEditorial(editorial);
21          libro.setAutor(autor);
22
23          System.out.println(libro);
24      }
25  }
26
```

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

Clases y atributos:

UNIVERSIDAD TECNOLÓGICA NACIONAL

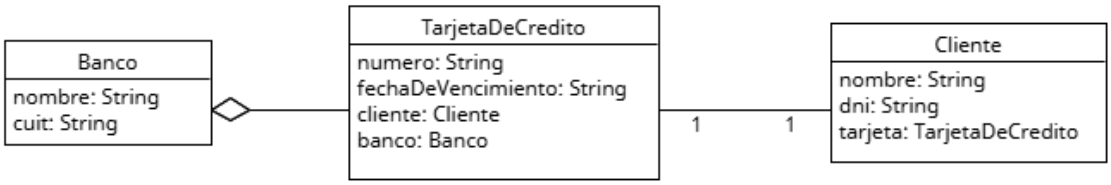
PROGRAMACIÓN II

I. TarjetaDeCrédito: numero, fechaVencimiento

II. Cliente: nombre, dni

III. Banco: nombre, cuit

Diagrama UML - Tipo de relación y dirección



Implementación de clases con atributos y relaciones definidas

Clase Banco

```
11  public class Banco {
12      private String nombre;
13      private String cuit;
14
15      public Banco(String nombre, String cuit) {
16          this.nombre = nombre;
17          this.cuit = cuit;
18      }
19
20      public String getNombre() { return nombre; }
21      public void setNombre(String nombre) { this.nombre = nombre; }
22
23      public String getCuit() { return cuit; }
24      public void setCuit(String cuit) { this.cuit = cuit; }
25
26      @Override
        public String toString() {
28          return "Banco{" +
29                  "nombre='" + nombre + '\'' +
30                  ", cuit='" + cuit + '\'' +
31                  '}';
32      }
33  }
```

Clase Cliente

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

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

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

    public String getNombre() { return nombre; }
    public void setNombre(String nombre) { this.nombre = nombre; }

    public String getDni() { return dni; }
    public void setDni(String dni) { this.dni = dni; }

    public TarjetaDeCredito getTarjeta() { return tarjeta; }

    public void setTarjeta(TarjetaDeCredito tarjeta) {
        if (this.tarjeta != tarjeta) {
            this.tarjeta = tarjeta;
            if (tarjeta != null && tarjeta.getCliente() != this) {
                tarjeta.setCliente(this);
            }
        }
    }

```

```java
    @Override
    public String toString() {
        return "Cliente{" +
                "nombre='" + nombre + '\'' +
                ", dni='" + dni + '\'' +
                ", tarjetaNumero=" + (tarjeta != null ? tarjeta.getNumero() : "null") +
                '}';
    }
}
```

Clase TarjetaDeCredito

Emilia Gómez Juárez
2025

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

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

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

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

    public String getFechaVencimiento() { return fechaVencimiento; }
    public void setFechaVencimiento(String fechaVencimiento) { this.fechaVencimiento = fechaVencimiento; }

    public Cliente getCliente() { return cliente; }

    public void setCliente(Cliente cliente) {
        if (this.cliente != cliente) {
            this.cliente = cliente;
            if (cliente != null && cliente.getTarjeta() != this) {
                cliente.setTarjeta(this);
            }
        }
    }
```

```java
    public Banco getBanco() { return banco; }
    public void setBanco(Banco banco) { this.banco = banco; }

    @Override
    public String toString() {
        return "TarjetaDeCredito{" +
                "numero='" + numero + '\'' +
                ", fechaVencimiento='" + fechaVencimiento + '\'' +
                ", cliente=" + (cliente != null ? cliente.getNombre() : "null") +
                ", banco=" + (banco != null ? banco.getNombre() : "null") +
                '}';
    }
}
```

Implementación en Main

```java
public class MainEj4 {
    public static void main(String[] args) {
        Banco banco = new Banco("Banco Nación", "30-12345678-9");

        Cliente cliente = new Cliente("Juan Pérez", "12345678");

        TarjetaDeCredito tarjeta = new TarjetaDeCredito("4567-8901-2345-6789", "12/2027");


        tarjeta.setBanco(banco);
        tarjeta.setCliente(cliente);

        System.out.println(tarjeta);
    }
}
```

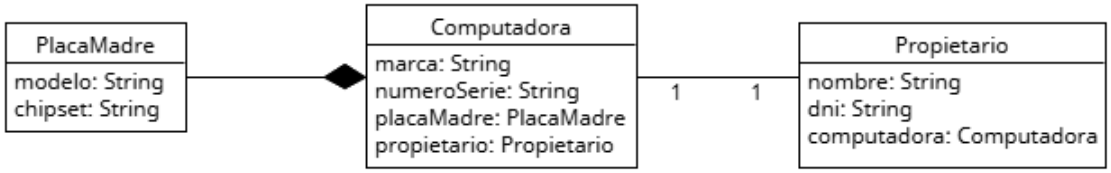
Emilia Gómez Juárez
2025

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

Diagrama UML - Tipo de relación y dirección



Implementación de clases con atributos y relaciones definidas

Clase PlacaMadre

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

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

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

    public String getModelo() { return modelo; }
    public void setModelo(String modelo) { this.modelo = modelo; }

    public String getChipset() { return chipset; }
    public void setChipset(String chipset) { this.chipset = chipset; }

    @Override
    public String toString() {
        return "PlacaMadre{" +
                "modelo='" + modelo + '\'' +
                ", chipset='" + chipset + '\'' +
                '}';
    }
}
```

Clase Computadora

```java
public class Computadora {
    private String marca;
    private String numeroSerie;
    private PlacaMadre placaMadre;
    private Propietario propietario;

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

    public String getMarca() { return marca; }
    public void setMarca(String marca) { this.marca = marca; }

    public String getNumeroSerie() { return numeroSerie; }
    public void setNumeroSerie(String numeroSerie) { this.numeroSerie = numeroSerie; }

    public PlacaMadre getPlacaMadre() { return placaMadre; }

    public Propietario getPropietario() { return propietario; }

    public void setPropietario(Propietario propietario) {
        if (this.propietario != propietario) {
            this.propietario = propietario;
            if (propietario != null && propietario.getComputadora() != this) {
                propietario.setComputadora(this);
            }
        }
    }
}
```

Emilia Gómez Juárez
2025

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

```java
    @Override
    public String toString() {
        return "Computadora{" +
                "marca='" + marca + '\'' +
                ", numeroSerie='" + numeroSerie + '\'' +
                ", placaMadre=" + placaMadre +
                ", propietario=" + (propietario != null ? propietario.getNombre() : "null") +
                '}';
    }

}

```

Clase Propietario

```java
public class Propietario {
     private String nombre;
    private String dni;
    private Computadora computadora;

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

    public String getNombre() { return nombre; }
    public void setNombre(String nombre) { this.nombre = nombre; }

    public String getDni() { return dni; }
    public void setDni(String dni) { this.dni = dni; }

    public Computadora getComputadora() { return computadora; }

    public void setComputadora(Computadora computadora) {
        if (this.computadora != computadora) {
            this.computadora = computadora;
            if (computadora != null && computadora.getPropietario() != this) {
                computadora.setPropietario(this);
            }
        }
    }

```

```java
    @Override
    public String toString() {
        return "Propietario{" +
                "nombre='" + nombre + '\'' +
                ", dni='" + dni + '\'' +
                ", computadoraSerie=" + (computadora != null ? computadora.getNumeroSerie() : "null") +
                '}';
    }
}

```

Implementación en Main

Emilia Gómez Juárez
2025

```java
public class MainEj5 {
    public static void main(String[] args) {
        Propietario propietario = new Propietario("Ana Gómez", "30123456");

        Computadora compu = new Computadora("Dell", "SN12345", "Z490", "Intel");

        compu.setPropietario(propietario);

        System.out.println(compu);
        System.out.println(propietario);
    }
}

```

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

Clases y atributos:

I. Reserva: fecha, hora
II. Cliente: nombre, telefono
III. Mesa: numero, capacidad

Diagrama UML - Tipo de relación y dirección

Implementación de clases con atributos y relaciones definidas

Clase Mesa

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

```java
public class Mesa {
    private int numero;
    private int capacidad;

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

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

```

Clase Cliente

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

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


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

```

Clase Reserva

Emilia Gómez Juárez
2025

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

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

    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
    }

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

Implementación en Main:

```java
public class MainEj6 {
    public static void main(String[] args) {
        Cliente cliente1 = new Cliente("Ana Pérez", "2611234567");
        Mesa mesa1 = new Mesa(5, 4);
        Reserva reserva1 = new Reserva("24/09", "21:00", mesa1);

        reserva1.setCliente(cliente1);

        System.out.println(reserva1);
    }
}
```

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

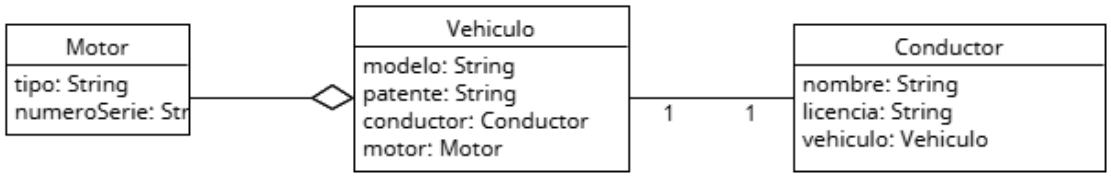
UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

Clases y atributos:

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

Diagrama UML - Tipo de relación y dirección



Implementación de clases con atributos y relaciones definidas

Clase Motor

```java
public class Motor {
    private String tipo;
    private String numeroSerie;

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

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

Clase Conductor

Emilia Gómez Juárez
2025

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

```java
public class Conductor {
    private String nombre;
    private String licencia;
    private Vehiculo vehiculo;

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

    public String getNombre() {
        return nombre;
    }

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

    @Override
    public String toString() {
        return "Conductor{nombre='" + nombre + "', licencia='" + licencia +
                "', vehiculo=" + (vehiculo != null ? vehiculo.getPatente() : "null") + "}";
    }
}
```

Clase Vehiculo

```java
public class Vehiculo {
    private String patente;
    private String modelo;
    private Motor motor;
    private Conductor conductor;

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

    public String getPatente() {
        return patente;
    }

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

Emilia Gómez Juárez
2025

```
37      @Override
38      public String toString() {
39          return "Vehiculo{patente='" + patente + "', modelo='" + modelo +
40                  "', motor=" + motor +
41                  ", conductor=" +  (conductor != null ? conductor.getNombre() : "null") + "}";
42      }
43  }
44
```

Implementación en Main

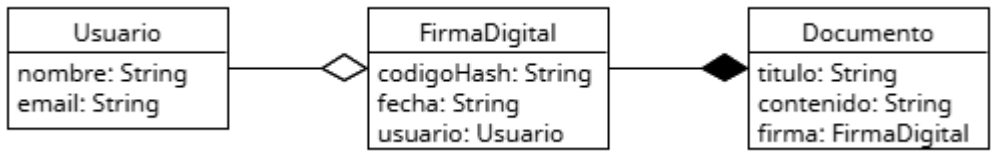
```
11  public class MainEj7 {
12      public static void main(String[] args) {
13          Motor motor1 = new Motor("Nafta", "SN12345");
14          Vehiculo vehiculo1 = new Vehiculo("ABC123", "Toyota Corolla", motor1);
15          Conductor conductor1 = new Conductor("Juan Pérez", "LIC98765");
16
17          vehiculo1.setConductor(conductor1);
18
19          System.out.println(vehiculo1);
20      }
21  }
22
```

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

Diagrama UML – Tipo de relación y dirección



Implementación de clases con atributos y relaciones definidas

Clase FirmaDigital

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

```java
public class FirmaDigital {
    private String codigoHash;
    private String fecha;
    private Usuario usuario;

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

    @Override
    public String toString() {
        return "FirmaDigital{codigoHash='" + codigoHash + "', fecha=" + fecha +
                ", usuario=" + usuario + "}";
    }
}
```

Clase Usuario

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

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

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

```

Clase Documento

```java
public class Documento {
    private String titulo;
    private String contenido;
    private FirmaDigital firma;

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

    public FirmaDigital getFirma() { return firma; }

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

Emilia Gómez Juárez
2025

Implementación en Main

```java
public class MainEj8 {
    public static void main(String[] args) {
        Usuario usuario1 = new Usuario("Ana López", "ana@ejemplo.com");
        Documento doc1 = new Documento("Contrato de Alquiler", "Contenido legal del contrato...", "HASH12345", "24/09", usuario1);

        System.out.println(doc1);
    }
}
```

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

Diagrama UML – Tipo de relación y dirección

Implementación de clases con atributos y relaciones definidas

Clase Paciente

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

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

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

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

Clase Profesional

```java
public class Profesional {
    private String nombre;
    private String especialidad;

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

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

Clase CitaMedica

```java
public class CitaMedica {
    private String fecha;
    private String hora;
    private Paciente paciente;       // asociación unidireccional
    private Profesional profesional; // asociación unidireccional

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

    public Paciente getPaciente() {return paciente;}

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

    public Profesional getProfesional() {return profesional;}

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

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

Implementación en Main:

Emilia Gómez Juárez
2025

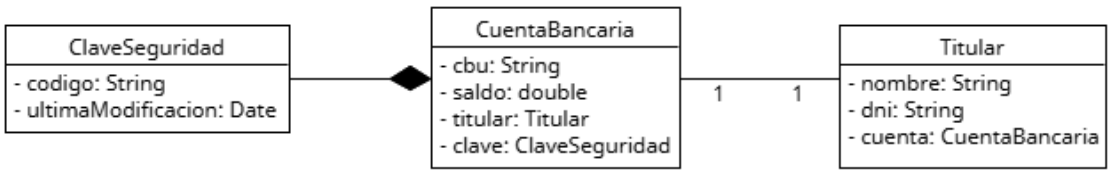
```java
public class MainEj9 {
    public static void main(String[] args) {
        Paciente pacientel = new Paciente("Juan Pérez", "OSDE");
        Profesional profesionall= new Profesional("Dra. García", "Cardiología");

        CitaMedica cita = new CitaMedica("29/09", "10:30");
        cita.setPaciente(pacientel);
        cita.setProfesional(profesionall);

        System.out.println(cita);
    }
}

```

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.

Diagrama UML – Tipo de relación y dirección



Implementación de clases con atributos y relaciones definidas

Clase Titular

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

```java
public class Titular {
    private String nombre;
    private String dni;
    private CuentaBancaria cuenta;

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

    public String getNombre() { return nombre; }

    public CuentaBancaria getCuenta() { return cuenta; }
    public void setCuenta(CuentaBancaria cuenta) {
        if (this.cuenta != cuenta) {
            this.cuenta = cuenta;
            if (cuenta != null && cuenta.getTitular() != this) {
                cuenta.setTitular(this);
            }
        }
    }

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

Clase ClaveSeguridad

```java
public class ClaveSeguridad {
    private String codigo;
    private Date ultimaModificacion;

    public ClaveSeguridad(String codigo) {
        this.codigo = codigo;
        this.ultimaModificacion = new Date();
    }

    @Override
    public String toString() {
        return "ClaveSeguridad{codigo='" + codigo + "', ultimaModificacion=" + ultimaModificacion + "}";
    }
}
```

Clase CuentaBancaria

Emilia Gómez Juárez
2025

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

```java
public class CuentaBancaria {
      private String cbu;
    private double saldo;
    private ClaveSeguridad clave;
    private Titular titular;

    public CuentaBancaria(String cbu, double saldo, String claveSeguridad) {
        this.cbu = cbu;
        this.saldo = saldo;
        this.clave = new ClaveSeguridad(claveSeguridad);
    }

    public String getCbu() { return cbu; }
    public void setCbu(String cbu) { this.cbu = cbu; }

    public double getSaldo() { return saldo; }
    public void setSaldo(double saldo) { this.saldo = saldo; }

    public ClaveSeguridad getClave() { return clave; }

    public Titular getTitular() { return titular; }
    public void setTitular(Titular titular) {
        if (this.titular != titular) {
            this.titular = titular;
            if (titular != null && titular.getCuenta() != this) {
                titular.setCuenta(this);
            }
        }
    }
```

```java
    @Override
    public String toString() {
        return "CuentaBancaria{cbu='" + cbu + "', saldo=" + saldo +
                ", clave=" + clave +
                ", titular=" + (titular != null ? titular.getNombre() : "null") +
                "}";
    }


}
```

Implementación en Main

```java
public class MainEj10 {
    public static void main(String[] args) {
        CuentaBancaria cuenta = new CuentaBancaria("1234567890123456789012", 50000.0, "ABC123");
        Titular titular = new Titular("María Gómez", "30111222");

        cuenta.setTitular(titular);

        System.out.println(cuenta);
    }
}
```

Emilia Gómez Juárez
2025

DEPENDENCIA DE USO: La clase usa otra como parámetro de un método, pero no la guarda como atributo.

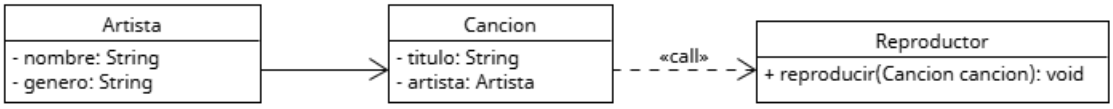
Ejercicios de Dependencia de Uso:

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)

Diagrama UML – Tipo de relación y dirección



Implementación de clases con atributos y relaciones definidas

Clase Artista

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

```java
public class Artista {
    private String nombre;
    private String genero;

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

    @Override
    public String toString() {
        return "Artista{" +
                "nombre='" + nombre + '\'' +
                ", genero='" + genero + '\'' +
                '}';
    }
}
```

Clase Cancion

```java
public class Cancion {
    private String titulo;
    private Artista artista;

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

    public Artista getArtista() {
        return artista;
    }

    @Override
    public String toString() {
        return "Cancion{" +
                "titulo='" + titulo + '\'' +
                ", artista=" + artista +
                '}';
    }
}
```

Clase Reproductor

Emilia Gómez Juárez
2025

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

Implementación en Main:

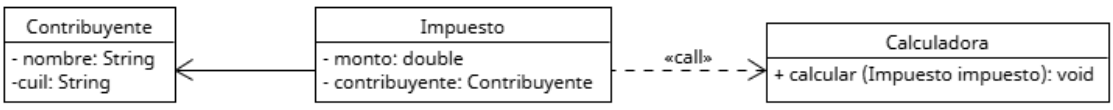
```java
public class MainEj11 {
    public static void main(String[] args) {
        Artista artista = new Artista("Gustavo Cerati", "Rock");
        Cancion cancion = new Cancion("Crimen", artista);

        Reproductor reproductor = new Reproductor();
        reproductor.reproducir(cancion);
    }
}
```

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

Clases y atributos:

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

Diagrama UML – Tipo de relación y dirección



Implementación de clases con atributos y relaciones definidas

Clase Impuesto

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

```java
public class Impuesto {
    private double monto;
    private Contribuyente contribuyente;

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

    public double getMonto() {
        return monto;
    }

    public Contribuyente getContribuyente() {
        return contribuyente;
    }

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

Clase Contribuyente

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

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

    @Override
    public String toString() {
        return "Contribuyente{" +
                "nombre='" + nombre + '\'' +
                ", cuil='" + cuil + '\'' +
                '}';
    }
}
```

Emilia Gómez Juárez
2025

Clase Calculadora

```java
public class Calculadora {
    public void calcular(Impuesto impuesto) {
        System.out.println(" Calculando impuesto de " + impuesto.getContribuyente() +
                " por un monto de $" + impuesto.getMonto());
    }
}

```

Implementación en Main

```java
public class MainEj12 {
    public static void main(String[] args) {
        Contribuyente c1 = new Contribuyente("María López", "20-12345678-9");
        Impuesto impl = new Impuesto(15000.75, c1);

        Calculadora calc = new Calculadora();
        calc.calcular(impl);
    }
}

```

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

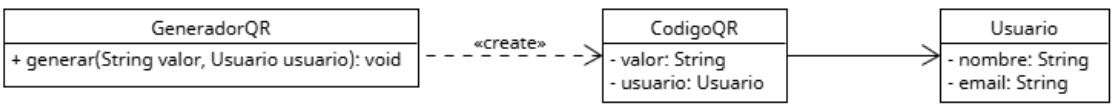
Ejercicios de Dependencia de Creación:

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)

Diagrama UML – Tipo de relación y dirección



UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

Implementación de clases con atributos y relaciones definidas

Clase Usuario

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

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

    @Override
    public String toString() {
        return "Usuario{" +
                "nombre='" + nombre + '\'' +
                ", email='" + email + '\'' +
                '}';
    }
}
```

Clase CodigoQR

```java
public class CodigoQR {
    private String valor;
    private Usuario usuario;

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

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

Clase GeneradorQR

Emilia Gómez Juárez
2025

```java
public class GeneradorQR {
    public void generar(String valor, Usuario usuario) {
        CodigoQR qr = new CodigoQR(valor, usuario);
        System.out.println("Código QR generado: " + qr);
    }
}
```

Implementación en Main

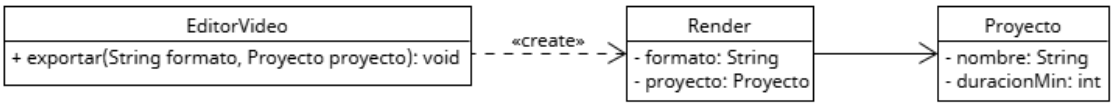
```java
public class MainEj13 {
    public static void main(String[] args) {
        Usuario u1 = new Usuario("Ana Pérez", "ana@ejemplo.com");

        GeneradorQR generador = new GeneradorQR();
        generador.generar("ABC123XYZ", u1);
    }
}
```

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)

Diagrama UML – Tipo de relación y dirección



Implementación de clases con atributos y relaciones definidas

Clase Proyecto

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

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

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

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

Clase Render

```java
public class Render {
    private String formato;
    private Proyecto proyecto;

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

    @Override
    public String toString() {
        return "Render{" +
                "formato='" + formato + '\'' +
                ", proyecto=" + proyecto +
                '}';
    }
}
```

Clase EditorVideo

Emilia Gómez Juárez
2025

UNIVERSIDAD TECNOLÓGICA NACIONAL

PROGRAMACIÓN II

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

```

Implementación en Main

```java
public class mainEj14 {
    public static void main(String[] args) {
        Proyecto proyecto = new Proyecto("Corto Documental", 15);

        EditorVideo editor = new EditorVideo();
        editor.exportar("MP4", proyecto);
    }
}
```

REPOSITORIO REMOTO: https://github.com/GomezJEmilia/UTN-Programacion2-TPs-EmiliaGJ/tree/a09c6fecf7e72a7ba17ed8c9399c371fdc7f89e6/05%20UML

Emilia Gómez Juárez
2025