

PROGRAMACIÓN II

Trabajo Práctico 5: Relaciones UML 1 a 1

OBJETIVO GENERAL

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

MARCO TEÓRICO

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

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:

- Pasaporte: numero, fechaEmision
- Foto: imagen, formato
- Titular: nombre, dni

Clase Foto

```
package Ejercicio1;

public class Foto {
    //atributos
    private String imagen;
    private String formato;

    //Constructor
    public Foto(String imagen, String formato) {
        this.imagen = imagen;
        this.formato = formato;
    }
    //Getters y Setters
    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;
    }
}
```

Clase Pasaporte

```
package Ejercicio1;
public class Pasaporte {

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

    public Pasaporte(String numero, String fechaEmision, String imagen, String formato) {
        this.numero = numero;
        this.fechaEmision = fechaEmision;
        this.foto = new Foto(imagen, formato); // Composición,
        // Pasaporte posee y controla el ciclo de vida Foto
    }

    //Getters y Setters
    public String getNumero() {
        return numero;
    }

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

    public String getFechaEmision() {
        return fechaEmision;
    }

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

    public Foto getFoto() {
        return foto;
    }

    public Titular getTitular() {
        return titular;
    }

    //Relacion bidireccional con Titular
    public void setTitular(Titular titular) {
        this.titular = titular;
        if (titular != null && titular.getPasaporte() != this) {
            titular.setPasaporte(this);
        }
    }
}
```

```
package Ejercicio1;
public class Titular {

    //ATRIBUTOS
    private String nombre;
    private String dni;
    private Pasaporte pasaporte;

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

    // Getters y Setters
    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 Pasaporte getPasaporte() {
        return pasaporte;
    }
}
```

Clase Titular

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

Clase Main

```
package Ejercicio1;
public class Main {
    public static void main(String[] args) {
        Titular titular = new Titular("Yamila Fernandez", "39854714");
        Pasaporte pasaporte = new Pasaporte("AB123456", "2025-05-25", "foto.jpg", "JPEG");
        titular.setPasaporte(pasaporte);
        System.out.println("Pasaporte número: " + pasaporte.getNumero());
        System.out.println("Titular: " + pasaporte.getTitular().getNombre());
        System.out.println("Foto: " + pasaporte.getFoto().getImagen());
    }
}
```

```
run:
Pasaporte número: AB123456
Titular: Yamila Fernandez
Foto: foto.jpg
```

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

Clase Batería

```
package Ejercicio2;
public class Bateria {
    //atributos
    private String modelo;
    private int capacidad;

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

    //Getters y Setters
    public String getModelo() {
        return modelo;
    }
    public void setModelo(String modelo) {
        this.modelo = modelo;
    }
    public int getCapacidad() {
        return capacidad;
    }
    public void setCapacidad(int capacidad) {
        this.capacidad = capacidad;
    }
}
```

Clase Celular

```
package Ejercicio2;
public class Celular{
    //atributos
    private String marca;
    private String modelo;
    private String imei;
    private Bateria bateria;
    private Usuario usuario;
    //constructor
    public Celular(String marca, String modelo, String imei, Bateria bateria) {
        this.marca = marca;
        this.modelo = modelo;
        this.imei = imei;
        this.bateria = bateria;
    }
}
```

```
//Getters y Setters
public String getMarca() {
    return marca;
}
public void setMarca(String marca) {
    this.marca = marca;
}
public String getModelo() {
    return modelo;
}
public void setModelo(String modelo) {
    this.modelo = modelo;
}
public String getImei() {
    return imei;
}
public void setImei(String imei) {
    this.imei = imei;
}
public Bateria getBateria() {
    return bateria;
}
public void setBateria(Bateria bateria) {
    this.bateria = bateria;
}
public Usuario getUsuario() {
    return usuario;
}

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

```
package Ejercicio2;
public class Main {

    public static void main(String[] args) {
        Usuario usuario = new Usuario("Yamila Fernandez", "39456789");
        Bateria bateria = new Bateria("BTX-5000", 4000);
        Celular celular = new Celular("Samsung", "A52", "123456789012345", bateria);
        usuario.setCelular(celular);
        System.out.println("Usuario: " + celular.getUsuario().getNombre());
        System.out.println("Celular: " + celular.getMarca());
        System.out.println("Bateria: " + celular.getBateria().getModelo());
        System.out.println("Celular: " + celular.getMarca() + " Bateria: " + bateria.getCapacidad());
    }
}
```

```
Run:
Usuario: Yamila Fernandez
Celular: Samsung
Bateria: BTX-5000
Celular: Samsung Bateria: 4000
```

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

```
package Ejercicio3;
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 String getNacionalidad() {
        return nacionalidad;
    }
}
```

```
package Ejercicio3;
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 String getDireccion() {
        return direccion;
    }
}
```

```
package Ejercicio3;
public class Libro {

    private String titulo;
    private String isbn;
    private Autor autor; // Asociación unidireccional → el Libro conoce al Autor
    private Editorial editorial; // Agregación → el Libro está asociado a una Editorial

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

    public String getTitulo() {
        return titulo;
    }

    public String getIsbn() {
        return isbn;
    }

    public Autor getAutor() {
        return autor;
    }

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

```
package Ejercicio3;
public class Main {

    public static void main(String[] args) {
        Autor autor = new Autor("G. G. Márquez", "Colombiana");
        Editorial editorial = new Editorial("Sudamericana", "Calle Falsa 123");
        Libro libro = new Libro("Cien años de soledad", "978-3-16-148410-0", editorial);
        libro.setAutor(autor);

        System.out.println("Libro: " + libro.getTitulo());
        System.out.println("Autor: " + libro.getAutor().getNombre());
        System.out.println("Editorial: " + libro.getEditorial().getNombre());
    }
}
```

```
Libro: Cien años de soledad
Autor: G. G. Márquez
Editorial: Sudamericana
```

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

```
package Ejercicio4;
public class Banco {

    private String nombre;
    private String cuit;

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

    public String getNombre() {
        return nombre;
    }

    public String getCuit() {
        return cuit;
    }
}
```

```
package Ejercicio4;
public class TarjetaDeCredito {
    private String numero;
    private String fechaVencimiento;
    private Cliente cliente;
    private Banco banco;

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

    public String getNumero() {
        return numero;
    }

    public String getFechaVencimiento() {
        return fechaVencimiento;
    }

    public Cliente getCliente() {
        return cliente;
    }

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

    public Banco getBanco() {
        return banco;
    }
}
```

```
package Ejercicio4;
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 String getDni() {
        return dni;
    }

    public TarjetaDeCredito getTarjeta() {
        return tarjeta;
    }

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

```
package Ejercicio4;
public class Main {

    public static void main(String[] args) {
        Cliente cliente = new Cliente("Yamila Fernandez", "39306875");
        Banco banco = new Banco("Banco Nación", "30-99999999-9");
        TarjetaDeCredito tarjeta = new TarjetaDeCredito("1234-5678-9876-5432", "12/28", banco);
        cliente.setTarjeta(tarjeta);
        System.out.println("Cliente: " + tarjeta.getCliente().getNombre());
        System.out.println("Banco: " + tarjeta.getBanco().getNombre());
    }
}
```

```
Cliente: Yamila Fernandez
Banco: Banco Nación
```

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

```
package Ejercicio5;
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;
    }
}
```

```
package Ejercicio5;
public class Computadora {

    private String marca;
    private String numeroSerie;
    private PlacaMadre placaMadre; // Composición → La computadora "contiene" una placa madre
    private Propietario propietario; // Asociación bidireccional → Propietario ↔ Computadora

    public Computadora(String marca, String numeroSerie, String modelo, String chipset) {
        this.marca = marca;
        this.numeroSerie = numeroSerie;
        // Composición: se crea la PlacaMadre dentro del constructor
        this.placaMadre = new PlacaMadre(modelo, chipset);
    }

    public String getMarca() {
        return marca;
    }

    public String getNumeroSerie() {
        return numeroSerie;
    }

    public Propietario getPropietario() {
        return propietario;
    }

    public PlacaMadre getPlacaMadre() {
        return placaMadre;
    }

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

```
package Ejercicio5;
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 String getDni() {
        return dni;
    }

    public Computadora getComputadora() {
        return computadora;
    }

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

```
package Ejercicio5;
public class Main {

    public static void main(String[] args) {
        Propietario propietario = new Propietario("Yamila Fernandez", "22334455");
        Computadora pc = new Computadora("HP", "SN123456", "ASUS TUF", "B450");
        propietario.setComputadora(pc);

        System.out.println("Propietario: " + pc.getPropietario().getNombre());
        System.out.println("Placa Madre: " + pc.getPlacaMadre().getModelo());
    }
}
```

```
Propietario: Yamila Fernandez
Placa Madre: ASUS TUF
```

6. Reserva - Cliente - Mesa

a. Asociación unidireccional: Reserva → Cliente

b. Agregación: Reserva → Mesa

Clases y atributos:

i. Reserva: fecha, hora

ii. Cliente: nombre, telefono

Mesa: numero, capacidad

```
package Ejercicio6;
public class Cliente {

    private String nombre;
    private String telefono;

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

    public String getNombre() {
        return nombre;
    }

    public String getTelefono() {
        return telefono;
    }
}
```

```
package Ejercicio6;
public class Mesa {

    private int numero;
    private int capacidad;

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

    public int getNumero() {
        return numero;
    }

    public int getCapacidad() {
        return capacidad;
    }
}
```

```
package Ejercicio6;
public class Reserva {

    private String fecha;
    private String hora;
    private Cliente cliente; // Asociación unidireccional
    private Mesa mesa; // Agregación

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

    public String getFecha() {
        return fecha;
    }

    public String getHora() {
        return hora;
    }

    public Cliente getCliente() {
        return cliente;
    }

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

    public Mesa getMesa() {
        return mesa;
    }
}
```

```
package Ejercicio6;
public class Main {

    public static void main(String[] args) {
        Cliente cliente = new Cliente("Yamila Fernandez", "11332211");
        Mesa mesa = new Mesa(5, 4);
        Reserva reserva = new Reserva("2025-09-10", "21:00", mesa);
        reserva.setCliente(cliente);
        System.out.println("Cliente: " + reserva.getCliente().getNombre());
        System.out.println("Mesa número: " + reserva.getMesa().getNumero());
    }
}
```

```
Cliente: Yamila Fernandez
Mesa número: 5
```

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

```
package Ejercicio7;
public class Motor {

    private String tipo;
    private String numeroSerie;

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

    public String getTipo() {
        return tipo;
    }

    public void setTipo(String tipo) {
        this.tipo = tipo;
    }

    public String getNumeroSerie() {
        return numeroSerie;
    }

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

```
package Ejercicio7;
public class Vehiculo {

    private String patente;
    private String modelo;
    private Motor motor; // Asociación bidireccional (Conductor ↔ Vehiculo)
    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 String getModelo() {
        return modelo;
    }

    public Motor getMotor() {
        return motor;
    }

    public void setMotor(Motor motor) {
        this.motor = motor;
    }

    public Conductor getConductor() {
        return conductor;
    }

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

```
package Ejercicio7;
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 String getLicencia() {
        return licencia;
    }

    public Vehiculo getVehiculo() {
        return vehiculo;
    }

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

```
package Ejercicio7;
public class Main {

    public static void main(String[] args) {
        Conductor conductor = new Conductor("Yamila Fernandez", "LIC12345");
        Motor motor = new Motor("Diesel", "MTR0001");
        Vehiculo vehiculo = new Vehiculo("AB123CD", "Toyota Hilux", motor);
        conductor.setVehiculo(vehiculo);

        System.out.println("Conductor: " + vehiculo.getConductor().getNombre());
        System.out.println("Motor: " + vehiculo.getMotor().getTipo());
    }
}
```

```
Conductor: Yamila Fernandez
Motor: Diesel
```

8. Documento - FirmaDigital - Usuario

a. Composición: Documento → FirmaDigital

b. Agregación: FirmaDigital → Usuario

Clases y atributos:

i. Documento: titulo, contenido

ii. FirmaDigital: codigoHash, fecha

iii. Usuario: nombre, email

```
package Ejercicio8;
public class Usuario {

    private String nombre;
    private String email;

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

    public String getNombre() {
        return nombre;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

}
```

```
package Ejercicio8;
public class FirmaDigital {

    private String codigoHash;
    private String fecha;
    private Usuario usuario; // Agregación: FirmaDigital → Usuario

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

    public String getCodigoHash() {
        return codigoHash;
    }

    public String getFecha() {
        return fecha;
    }

    public Usuario getUsuario() {
        return usuario;
    }

}
```

```
package Ejercicio8;
public class Documento {

    private String titulo;
    private String contenido;
    private FirmaDigital firma; // Composición: Documento → FirmaDigital

    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 String getTitulo() {
        return titulo;
    }

    public String getContenido() {
        return contenido;
    }

    public FirmaDigital getFirma() {
        return firma;
    }

}
```

```
package Ejercicio8;
public class Main {

    public static void main(String[] args) {
        // Crear usuario
        Usuario usuario = new Usuario("Yamila Fernandez", "feryami@mail.com");

        // Crear documento (composición con firma)
        Documento doc = new Documento("Contrato", "Contenido del contrato...", "abc123", "2025-08-01", usuario);

        System.out.println("Documento: " + doc.getTitulo());
        System.out.println("Usuario de la firma: " + doc.getFirma().getUsuario().getNombre());
        System.out.println("Hash de firma: " + doc.getFirma().getCodigoHash());
    }

}
```

```
Documento: Contrato
Usuario de la firma: Yamila Fernandez
Hash de firma: abc123
```

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

```
package Ejercicio9;
public class Paciente {

    private String nombre;
    private String obraSocial;

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

    public String getNombre() {
        return nombre;
    }
    public String getObraSocial() {
        return obraSocial;
    }
}
```

```
package Ejercicio9;
public class Profesional {

    private String nombre;
    private String especialidad;

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

    public String getNombre() {
        return nombre;
    }
    public String getEspecialidad() {
        return especialidad;
    }
}
```

```
package Ejercicio9;
public class CitaMedica {

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

    public CitaMedica(String fecha, String hora) {
        this.fecha = fecha;
        this.hora = hora;
    }
    public String getFecha() {
        return fecha;
    }
    public String getHora() {
        return 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;
    }
}
```

```

package Ejercicio9;
public class Main {

    public static void main(String[] args) {
        Paciente paciente = new Paciente("Andrea Flores", "OSDE");
        Profesional profesional = new Profesional("Dr. Carlos Soto", "Cardiología");
        CitaMedica cita = new CitaMedica("2025-10-15", "10:30");
        cita.setProfesional(profesional);
        cita.setPaciente(paciente);
        System.out.println("Paciente: " + cita.getPaciente().getNombre() + " Obra Social: " + paciente.getObraSocial());
        System.out.println("Profesional: " + cita.getProfesional().getNombre());
    }
}

```

```

Paciente: Andrea Flores Obra Social: OSDE
Profesional: Dr. Carlos Soto

```

10. CuentaBancaria - ClaveSeguridad - Titular

a. Composición: CuentaBancaria → ClaveSeguridad

b. Asociación bidireccional: CuentaBancaria ↔ Titular

Clases y atributos:

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

```

package Ejercicio10;
public class ClaveSeguridad {

    private String codigo;
    private String ultimaModificacion;

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

    public String getCodigo() {
        return codigo;
    }

    public String getUltimaModificacion() {
        return ultimaModificacion;
    }
}

```

```

package Ejercicio10;
public class Titular {

    private String nombre;
    private String dni;
    private CuentaBancaria cuenta; // Asociación bidireccional

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

    public String getNombre() {
        return nombre;
    }

    public String getDni() {
        return dni;
    }

    public CuentaBancaria getCuenta() {
        return cuenta;
    }

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

```

```

package Ejercicio10;
public class CuentaBancaria {

    private String cbu;
    private double saldo;
    private ClaveSeguridad clave; // Composición (CuentaBancaria → ClaveSeguridad)
    private Titular titular; // Asociación bidireccional (CuentaBancaria ↔ Titular)

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

    public String getCbu() {
        return cbu;
    }

    public double getSaldo() {
        return saldo;
    }

    public ClaveSeguridad getClave() {
        return clave;
    }

    public Titular getTitular() {
        return titular;
    }

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

```

```
package Ejercicio10;
public class Main {

    public static void main(String[] args) {
        Titular titular = new Titular("Yamila Fernandez", "33445566");
        ClaveSeguridad cs = new ClaveSeguridad("seg123", "2025-08-01");
        CuentaBancaria cuenta = new CuentaBancaria("2220001133445566", 150000.0, "seg123", "2025-08-01");
        titular.setCuenta(cuenta);
        System.out.println("Titular: " + cuenta.getTitular().getNombre());
        System.out.println("Saldo: " + cuenta.getSaldo());
        System.out.println("Clave: " + cuenta.getClave().getCodigo());
    }
}
```

```
Titular: Yamila Fernandez
Saldo: 150000.0
Clave: seg123
```

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:

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

```
package Ejercicio11;
public class Artista {

    private String nombre;
    private String genero;

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

    public Artista() {
    }

    public String getNombre() {
        return nombre;
    }

    public String getGenero() {
        return genero;
    }
}
```

```
package Ejercicio11;
public class Cancion {

    private String titulo;
    private Artista artista;

    public Cancion(String titulo, Artista artista) {
        this.titulo = titulo;
        this.artista = artista; // Asociación unidireccional (Canción → Artista)
    }

    public Cancion() {
    }

    public String getTitulo() {
        return titulo;
    }

    public Artista getArtista() {
        return artista;
    }

    public void setArtista(Artista artista) {
        this.artista = artista;
    }
}
```

```
package Ejercicio11;
public class Reproductor {

    // Dependencia de uso: el método usa una Cancion como parámetro,
    // pero no la guarda como atributo.

    public void reproducir(Cancion cancion) {
        System.out.println("Reproduciendo: " + cancion.getTitulo()
            + " por " + cancion.getArtista().getNombre());
    }
}
```

```
package Ejercicio11;
public class Main {

    public static void main(String[] args) {
        Artista artista = new Artista("Shakira", "Pop");
        Cancion cancion = new Cancion("Antología", artista);
        Reproductor r = new Reproductor();
        r.reproducir(cancion); // Dependencia de uso
    }
}
```

```
run:
Reproduciendo: Antología por Shakira
```

12. Impuesto - Contribuyente - Calculadora

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

Clases y atributos:

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

```
package Ejercicio12;
public class Contribuyente {

    private String nombre;
    private String cuil;

    public Contribuyente(String nombre, String cuil) {
        this.nombre = nombre;
        this.cuil = cuil;
    }
    public Contribuyente() {
    }
    public String getNombre() {
        return nombre;
    }
    public String getCuil() {
        return cuil;
    }
}
```

```
package Ejercicio12;
public class Impuesto {

    private double monto;
    private Contribuyente contribuyente; // Asociación unidireccional (Impuesto → Contribuyente)

    public Impuesto(double monto, Contribuyente contribuyente) {
        this.monto = monto;
        this.contribuyente = contribuyente;
    }
    public Impuesto() {
    }
    public double getMonto() {
        return monto;
    }
    public Contribuyente getContribuyente() {
        return contribuyente;
    }
    public void setContribuyente(Contribuyente contribuyente) {
        this.contribuyente = contribuyente;
    }
}
```

```
package Ejercicio12;
public class Calculadora {

    // Dependencia de uso: el método usa un objeto Impuesto como parámetro
    // pero no lo guarda como atributo
    public void calcular(Impuesto impuesto) {
        System.out.println("Cálculo de impuesto de: " + impuesto.getContribuyente().getNombre());
        System.out.println("Monto: $" + impuesto.getMonto());
    }
}
```

```
package Ejercicio12;
public class Main {

    public static void main(String[] args) {
        Contribuyente contribuyente = new Contribuyente("Carlos Peralta", "20-12345678-3");
        Impuesto impuesto = new Impuesto(50000.0, contribuyente);
        Calculadora calc = new Calculadora();
        calc.calcular(impuesto); // dependencia de uso
    }
}
```

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:

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

```
package Ejercicio13;
public class Usuario {

    private String nombre;
    private String email;

    public Usuario() {
    }
    public Usuario(String nombre, String email) {
        this.nombre = nombre;
        this.email = email;
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public String getEmail() {
        return email;
    }
}
```

```
package Ejercicio13;
public class CódigoQR {

    private String valor;
    private Usuario usuario; // Asociación unidireccional (CódigoQR → Usuario)

    public CódigoQR() {
    }

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

    public String getValor() {
        return valor;
    }

    public Usuario getUsuario() {
        return usuario;
    }

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

```
package Ejercicio13;
public class GeneradorQR {
    // Dependencia de creación: el método crea un objeto CódigoQR dentro de él
    public void generar(String valor, Usuario usuario) {
        CódigoQR qr = new CódigoQR(valor, usuario);
        System.out.println("QR generado para: " + qr.getUsuario().getNombre() + " - Valor: " + qr.getValor());
    }
}
```

```
package Ejercicio13;
public class Main {

    public static void main(String[] args) {
        Usuario user = new Usuario("Elena Soto", "elena@gmail.com");
        GeneradorQR gen = new GeneradorQR();
        gen.generar("ABC-123", user); // dependencia de creación
    }
}
```

```
run:
QR generado para: Elena Soto - Valor: ABC-123
```

14. EditorVideo - Proyecto - Render

- Asociación unidireccional: Render → Proyecto
- Dependencia de creación: EditorVideo.exportar(String, Proyecto)
- Clases y atributos:
 - Render: formato.
 - Proyecto: nombre, duracionMin.
 - EditorVideo->método: void exportar(String formato, Proyecto proyecto)

```
package Ejercicio14;
public class Proyecto {

    private String nombre;
    private int duracionMin;

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

    public Proyecto() {
    }

    public String getNombre() {
        return nombre;
    }

    public int getDuracionMin() {
        return duracionMin;
    }

}
```

```
package Ejercicio14;
public class Render {

    private String formato;
    private Proyecto proyecto; // Asociación unidireccional (Render → Proyecto)

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

    public Render() {
    }

    public String getFormato() {
        return formato;
    }

    public Proyecto getProyecto() {
        return proyecto;
    }

    public void setProyecto(Proyecto proyecto) {
        this.proyecto = proyecto;
    }

}
```

```
package Ejercicio14;
public class EditorVideo {

    // Dependencia de creación:
    // Crea un Render dentro del método, pero no lo conserva como atributo
    public void exportar(String formato, Proyecto proyecto) {
        Render render = new Render(formato, proyecto);
        System.out.println("Exportando proyecto: " + render.getProyecto().getNombre() + " en formato " + render.getFormato());
    }

}
```

```
package Ejercicio14;
public class Main {

    public static void main(String[] args) {
        Proyecto proyecto = new Proyecto("Promo institucional", 3);
        EditorVideo editor = new EditorVideo();
        editor.exportar("MP4", proyecto); // dependencia de creación
    }

}
```