

**PROGRAMACIÓN II.**  
**Modulo 5: UML Básico.**

**Trabajo Práctico 5: UML Básico.**

**Alumno:** LEPKA AGUSTIN

**Comisión:** 13

**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
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

**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

## Resultado:

### Clase tp5ej1 (contiene metodo Main):

```
package tp5ej1;

public class TP5EJ1 {

    public static void main(String[] args) {

        //se crea al titular
        Titular titular = new Titular("Pepe", "44786434");
        //se crea la foto del titular
        Foto foto = new Foto("imagenTitular.jpg", "JPG");
        //se crea el pasaporte, con foto y titular
        Pasaporte pasaporte = new Pasaporte("AR4747", "01/01/2025",
"imagenTitular.jpg", ".jpg", titular);
        //se imprimen los datos
        System.out.println(pasaporte);
        System.out.println("El titular del pasaporte: " +
titular.getPasaporte());

    }

}
```

### Clase Pasaporte

```
package tp5ej1;

public class Pasaporte {

    //Atributos
    private String numero;
    private String fechaEmision;
    private Foto foto; //Composicion y unidireccional, si se borra
Pasaporte, tambien Foto. Foto no sabe a que pasaporte pertenece
    private Titular titular; //Asociacion bidireccional

    //Constructor
    public Pasaporte(String numero, String fechaEmision, String
imagen, String formato, Titular titular) {
        this.numero = numero;
        this.fechaEmision = fechaEmision;
        this.foto = new Foto(imagen, formato);
        this.titular = titular;
        this.titular.setPasaporte(this); //se genera la asociacion
    }

    @Override
    public String toString() {
```

```

        return "Pasaporte numero:" + numero + ", fechaEmision:" +
fechaEmision +
        ", " + foto + ", " + titular;
    }
}

```

## **Clase Foto**

```

package tp5ej1;

public class Foto {

    //Atributos
    private String imagen;
    private String formato;

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

    //getters
    public String getImagen() {
        return imagen;
    }

    public String getFormato() {
        return formato;
    }

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

}

```

## **Clase Titular**

```

package tp5ej1;

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;
    }

    //setter

```

```

        public void setPasaporte(Pasaporte pasaporte) {
            this.pasaporte = pasaporte;
        }

        //getter
        public Pasaporte getPasaporte() {
            return pasaporte;
        }

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

```

## 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

### Resultado:

#### Clase tp5ej2 (contiene metodo Main):

```

package tp5ej2;

public class tp5ej2 {

    public static void main(String[] args) {

        //se crea un usuario
        Usuario usuario1 = new Usuario("Perez Fulano", "12345678");
        //Se crea una bateria
        Bateria bateria1 = new Bateria("Li-Ion", 5000);
        //se crea un celular con la bateria
        Celular celular1 = new Celular("32543243", "Samsung", "Galaxy
S24 ULTRA", bateria1, usuario1);

        //se imprime la informacion
        System.out.println(celular1);
        System.out.println("El usuario del celular: " +
usuario1.getCelular());
    }
}

```

#### Clase Usuario

```

package tp5ej2;

public class Usuario {
    private String nombre;

```

```

private String dni;
private Celular celular;//Asociación bidireccional

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

//setter
public void setCelular(Celular celular){
    this.celular = celular;
}
//getter
public Celular getCelular(){
    return celular;
}

@Override
public String toString() {
    return "Usuario: nombre: " + nombre + ", dni: " + dni;
}
}

```

## **Clase Celular**

```

package tp5ej2;

public class Celular {
    private String imei;
    private String marca;
    private String modelo;
    private Bateria bateria;//Agregacion unidireccional
    private Usuario usuario;//Asociacion bidireccional

    //Constructor
    public Celular(String imei, String marca, String modelo, Bateria
bateria, Usuario usuario) {
        this.imei = imei;
        this.marca = marca;
        this.modelo = modelo;
        this.bateria = bateria;
        this.usuario = usuario;
        this.usuario.setCelular(this);//se genera asociacion
    }

    @Override
    public String toString() {
        return "Celular: imei:" + imei + ", marca:" + marca + ",
modelo:" + modelo + ", bateria:" + bateria + ", usuario: " + usuario;
    }
}

```

## **Clase Bateria**

```

package tp5ej2;

public class Bateria {

```

```

private String modelo;
private int capacidad;

//Constructor
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;
}

@Override
public String toString() {
    return "Bateria: modelo:" + modelo + ", capacidad:" +
capacidad + "mAh";
}
}

```

### 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

#### Resultado:

#### Clase tp5ej3 (contiene metodo Main):

```

package tp5ej3;

public class tp5ej3 {

    public static void main(String[] args) {
        //se crea el autor
        Autor autor1 = new Autor("J. R. R. Tolkien", "Britanico");

        //se crea la editorial
        Editorial editorial1 = new Editorial("HarperCollins", "UK");
    }
}

```

```

        //se crea el libro con asociación a autor y agregación a
editorial
        Libro libro1 = new Libro("El señor de los anillos",
"9123092132", autor1, editorial1);

        //se muestra la información
        System.out.println("Información del libro");
        System.out.println(libro1.toString());

    }

}

```

## **Clase Editorial**

```

package tp5ej3;

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

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

    @Override
    public String toString() {
        return "Editorial nombre:" + nombre + ", direccion:" +
direccion;
    }
}

```

## **Clase Libro**

```

package tp5ej3;

public class Libro {
    private String titulo;
    private String isbn;
    private Autor autor; //Asociacion unidireccional
    private Editorial editorial; //Agregacion

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

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

```

```
}
```

## **Clase Autor**

```
package tp5ej3;
```

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

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

    //Getters y setters
    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;
    }
}
```

## **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

## **Resultado:**

### **Clase tp5ej4 (contiene metodo Main):**

```
package tp5ej4;
```



```

public class tp5ej4 {

    public static void main(String[] args) {
        //se crea el banco
        Banco banco1 = new Banco("Banco Nacion", "23-12345678-5");

        //se crea el cliente
        Cliente cliente1 = new Cliente("Jorge perez", "1332234");

        //se crea tarjeta con el banco
        TarjetaDeCredito tarjeta1 = new TarjetaDeCredito("1266-5458-
9322-3322", "12/27", banco1);

        //se asocia la tarjeta al cliente (bidireccional)
        cliente1.setTarjeta(tarjeta1);

        //se muestra la información de la tarjeta
        System.out.println("Datos de la tarjeta");
        System.out.println(tarjeta1.getDatosTarjeta());
    }

}

```

## Clase Banco

```

package tp5ej4;

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

    public String getCuit() {
        return cuit;
    }

    public void setCuit(String cuit) {
        this.cuit = cuit;
    }

    @Override
    public String toString() {
        return "Banco: nombre:" + nombre + ", cuit:" + cuit;
    }
}

```

## Clase Cliente

```

package tp5ej4;

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

    public Cliente(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 TarjetaDeCredito getTarjeta() {
        return tarjeta;
    }

    //Asociación bidireccional: asignar tarjeta
    public void setTarjeta(TarjetaDeCredito tarjeta) {
        this.tarjeta = tarjeta;
        if (tarjeta.getClientes() != this) {
            tarjeta.setCliente(this);
        }
    }

    @Override
    public String toString() {
        return "Cliente: nombre:" + nombre + ", dni:" + dni;
    }
}

```

## Clase TarjetaDeCredito

```

package tp5ej4;

public class TarjetaDeCredito {
    private String numero;
    private String fechaVencimiento;
    private Cliente cliente; //Asociacion bidireccional
    private Banco banco; //Agregacion

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

```

```

        this.banco = banco;
    }

    //Getters y setters
    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) {
        this.cliente = cliente;
        if (cliente.getTarjeta() != this) {
            cliente.setTarjeta(this);
        }
    }

    public Banco getBanco() {
        return banco;
    }

    public void setBanco(Banco banco) {
        this.banco = banco;
    }

    //Getter que devuelve todos los datos de la tarjeta
    public String getDatosTarjeta() {
        return "Número: " + numero + "\n" +
            "Fecha de Vencimiento: " + fechaVencimiento + "\n" +
            "Cliente: " + cliente.getNombre() + " (DNI: " +
cliente.getDni() + ") \n" +
            "Banco: " + banco.getNombre() + " - CUIT: " +
banco.getCuit();
    }

    @Override
    public String toString() {
        return "TarjetaDeCredito numero:" + numero + ",
fechaVencimiento:" + fechaVencimiento + ", banco:" + banco;
    }
}

```

## 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

### Resultado:

#### Clase tp5ej5 (contiene metodo Main):

```
package tp5ej5;

public class tp5ej5 {

    public static void main(String[] args) {
        //se crea al propietario
        Propietario propietario1 = new Propietario("Agustín Pérez",
"13542334");

        //se crea la computadora (composicion con placa madre)
        Computadora computadoral = new Computadora("ASUS", "SN235423",
"ASUS ROG", "Z790", propietario1);

        //se muestra la informacion completa
        System.out.println("Datos de la computadora");
        System.out.println(computadoral.toString());
    }
}
```

#### Clase Propietario

```
package tp5ej5;

public class Propietario {
    private String nombre;
    private String dni;

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

    //Getters
    public String getNombre() {
        return nombre;
    }

    public String getDni() {
        return dni;
    }

    @Override
    public String toString() {
        return "Propietario nombre:" + nombre + ", dni:" + dni;
    }
}
```

## Clase Computadora

```
package tp5ej5;

public class Computadora {
    private String marca;
    private String numeroSerie;
    private PlacaMadre placaMadre; //Composicion
    private Propietario propietario; //Asociacion bidireccional

    public Computadora(String marca, String numeroSerie, String
modeloPlaca, String chipset, Propietario propietario) {
        this.marca = marca;
        this.numeroSerie = numeroSerie;
        this.placaMadre = new PlacaMadre(modeloPlaca, chipset);
//Composicion
        this.propietario = propietario; //Asociacion bidireccional
    }

    @Override
    public String toString() {
        return "Computadora marca:" + marca + ", numeroSerie:" +
numeroSerie + ", Placa madre: " + placaMadre + ", propietario: " +
propietario;
    }
}
```

## Clase PlacaMadre

```
package tp5ej5;

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

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

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

## 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

### **Resultado:**

#### **Clase tp5ej6 (contiene metodo Main):**

```
package tp5ej6;

public class tp5ej6 {

    public static void main(String[] args) {
        //se crea al cliente
        Cliente cliente1 = new Cliente("Jorge lopez", "3242342");

        //se crea la mesa
        Mesa mesa1 = new Mesa(5, 4);

        //se crea la reserva (asociacion unidireccional con cliente y
        //agregacion con mesa)
        Reserva reserva1 = new Reserva("10/5/25", "22:35", cliente1,
        mesa1);

        //se muestra la informacion completa
        System.out.println("Datos de la reserva");
        System.out.println(reserva1);

    }

}
```

#### **Clase Reserva**

```
package tp5ej6;

public class Reserva {
    private String fecha;
    private String hora;
    private Cliente cliente;//Asociacion unidireccional
    private Mesa mesa;//Agregacion

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

    @Override
    public String toString() {
        return "Reserva fecha:" + fecha + ", hora:" + hora + ",
        cliente: " + cliente + ", mesa:" + mesa;
    }

}
```

## Clase Mesa

```
package tp5ej6;

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

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

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

## Clase Cliente

```
package tp5ej6;

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

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

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

## 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

### Resultado:

#### Clase tp5ej7 (contiene metodo Main):

```
package tp5ej7;

public class tp5ej7 {
```

```

        public static void main(String[] args) {
            //se crea al conductor
            Conductor conductor1 = new Conductor("Jorge Fulano", "L-
123123");

            //se crea el motor
            Motor motor1 = new Motor("V4", "SN6234");

            //se crea el vehiculo (agregaciin con motor y asociacion
            bidireccional con conductor)
            Vehiculo vehiculo1 = new Vehiculo("AD1234WF", "Fiat 600",
            motor1, conductor1);

            //se muestra la informacion completa
            System.out.println(conductor1.getVehiculo());
            System.out.println(vehiculo1.getConductor());

        }
    }
}

```

## Clase Vehiculo

```

package tp5ej7;

public class Vehiculo {
    private String patente;
    private String modelo;
    private Motor motor; //Agregacion
    private Conductor conductor; //Asociacion bidireccional

    //constructor
    public Vehiculo(String patente, String modelo, Motor motor,
    Conductor conductor) {
        this.patente = patente;
        this.modelo = modelo;
        this.motor = motor;
        this.conductor = conductor;
        this.conductor.setVehiculo(this);
    }

    //getter
    public Conductor getConductor(){
        return conductor;
    }

    @Override
    public String toString() {
        return "Vehículo patente:" + patente + ", modelo:" + modelo +
        ", Motor: " + motor + ", Conductor: " + conductor;
    }
}

```

## Clase Motor

```

package tp5ej7;

public class Motor {
    private String tipo;
    private String numeroSerie;
}

```



```

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

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

```

## Clase Conductor

```

package tp5ej7;

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

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

    //setter
    public void setVehiculo(Vehiculo vehiculo){
        this.vehiculo = vehiculo;
    }
    //getter
    public Vehiculo getVehiculo(){
        return vehiculo;
    }

    @Override
    public String toString() {
        return "Conductor nombre:" + nombre + ", licencia:" +
licencia;
    }
}

```

## 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

### Resultado:

**Clase tp5ej8 (contiene metodo Main):**

```

package tp5ej8;

public class tp5ej8 {

    public static void main(String[] args) {
        //se crea el usuario
        Usuario usuario1 = new Usuario("Jorge Fulano",
        "Jorge@gmail.com");

        //se crea el documento (composición con firma digital)
        Documento documento1 = new Documento("Contrato 123",
        "Condiciones empleado", "AB5323", "9/2/25", usuario1);

        //se muestra la informacion completa
        System.out.println("Datos del documento");
        System.out.println(documento1);
    }
}

```

## Clase Usuario

```

package tp5ej8;

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

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

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

```

## Clase FirmaDigital

```

package tp5ej8;

public class FirmaDigital {
    private String codigoHash;
    private String fecha;
    private Usuario usuario; //Agregacion

    //Constructor
    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 Documento

```

package tp5ej8;

public class Documento {
    private String titulo;
    private String contenido;
    private FirmaDigital firmaDigital; //Composicion

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

    @Override
    public String toString() {
        return "Documento titulo:" + titulo + ", contenido:" +
contenido + ", firmadigital: " + firmaDigital;
    }
}

```

## 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

### Resultado:

#### Clase tp5ej9 (contiene metodo Main):

```

package tp5ej9;

public class tp5ej9 {

    public static void main(String[] args) {
        //se crea el paciente
        Paciente paciente1 = new Paciente("Jorge Fulano", "Swiss");

        //se crea el profesional
        Profesional profesional1 = new Profesional("Dr. Gomez",
"Cirugia");

        //se crea la cita medica
    }
}

```

```

        CitaMedica cita1 = new CitaMedica("01/10/24", "8:15",
paciente1, profesional1);

        //se muestra la informacion completa
        System.out.println("Datos de la cita medica");
        System.out.println(cita1);

    }
}

```

## Clase Profesional

```

package tp5ej9;

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

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

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

```

## Clase Paciente

```

package tp5ej9;

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

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

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

```

## Clase CitaMedica

```

package tp5ej9;

public class CitaMedica {
    private String fecha;
    private String hora;
    private Paciente paciente;//Asociacion unidireccional
}

```

```

        private Profesional profesional;//Asociacion unidireccional

        //constructor
        public CitaMedica(String fecha, String hora, Paciente paciente,
        Profesional profesional) {
            this.fecha = fecha;
            this.hora = hora;
            this.paciente = paciente;
            this.profesional = profesional;
        }

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

```

## 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.

### Resultado:

#### Clase tp5ej10 (contiene metodo Main):

```

package tp5ej10;

public class tp5ej10 {

    public static void main(String[] args) {
        //se crea al titular
        Titular titular1 = new Titular("Jorge Fulano", "132423");
        //se crea la cuenta bancaria
        CuentaBancaria cuental = new
        CuentaBancaria("1234243243243234243", 10350, "PASSWORD", "01/02/25",
        titular1);

        //se imprimen los datos
        System.out.println("Datos de la cuenta bancaria");
        System.out.println(cuental);

    }

}

```

#### Clase Titular

```

package tp5ej10;

public class Titular {
    private String nombre;
    private String dni;
    private CuentaBancaria cbu;
}

```

```

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

//setter
public void setCuentaBancaria(CuentaBancaria cbu) {
    this.cbu = cbu;
}

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

```

## Clase Cuenta Bancaria

```

package tp5ej10;

public class CuentaBancaria {
    private String cbu;
    private double saldo;
    private ClaveSeguridad claveSeguridad; //composicpn
    private Titular titular; //Asociacion bidireccional

    public CuentaBancaria(String cbu, double saldo, String
codigoClave, String fechaModificacion, Titular titular) {
        this.cbu = cbu;
        this.saldo = saldo;
        this.claveSeguridad = new ClaveSeguridad(codigoClave,
fechaModificacion); //Composicion
        this.claveSeguridad.setCuentaBancaria(this);
        this.titular = titular; //Asociacion bidireccional
        this.titular.setCuentaBancaria(this);
    }
    //getter
    public Titular getTitular(){
        return titular;
    }
    public ClaveSeguridad getClaveSeguridad(){
        return claveSeguridad;
    }

    @Override
    public String toString() {
        return "CuentaBancaria cbu:" + cbu + ", saldo:" + saldo + ",
clave: " + claveSeguridad + "Titular: " + titular;
    }
}

```

## Clase ClaveSeguridad

```

package tp5ej10;

public class ClaveSeguridad {
    private String codigo;
}

```

```

private String ultimaModificacion;
private CuentaBancaria cbu;

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

//setter
public void setCuentaBancaria(CuentaBancaria cbu) {
    this.cbu = cbu;
}

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

```

## 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

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

#### Clases y atributos:

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

#### Resultado:

#### Clase tp5ej11 (contiene metodo Main):

```

package tp5ej11;

public class tp5ej11 {

    public static void main(String[] args) {

        //se crea el artista
        Artista artista = new Artista("Shakira", "Pop");
        //se crea la cancion
        Cancion cancion = new Cancion("Hips Don't Lie", artista);
        //se crea el reproductoir
        Reproductor reproductor = new Reproductor();

        //ejecucion del reproductor
        reproductor.reproducir(cancion); //Dependencia de uso
    }
}

```

```
    }  
}
```

## Clase Reproductor

```
package tp5ej11;  
  
public class Reproductor {  
  
    //Dependencia de uso: recibe la clase Cancion como parametro y no  
    la guarda  
  
    public void reproducir(Cancion cancion) {  
        System.out.println("Reproduciendo: " + cancion);  
    }  
}
```

## Clase Cancion

```
package tp5ej11;  
  
public class Cancion {  
    private String titulo;  
    private Artista artista;//Asociacion unidireccional  
  
    //constructor  
    public Cancion(String titulo, Artista artista) {  
        this.titulo = titulo;  
        this.artista = artista;  
    }  
  
    //getters  
    public String getTitulo() {  
        return titulo;  
    }  
  
    public Artista getArtista() {  
        return artista;  
    }  
  
    @Override  
    public String toString() {  
        return "Canción: " + titulo + " artista: " + artista;  
    }  
}
```

## Clase Artista

```
package tp5ej11;  
  
public class Artista {  
    private String nombre;  
    private String genero;  
  
    //constructor  
    public Artista(String nombre, String genero) {  
        this.nombre = nombre;  
        this.genero = genero;  
    }  
}
```



```

@Override
public String toString() {
    return "Artista: " + nombre + " - Género: " + genero;
}
}

```

## 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)

### Resultado:

#### Clase tp5ej12 (contiene metodo Main):

```

package tp5ej12;

public class tp5ej12 {

    public static void main(String[] args) {
        //creacion de contribuyente
        Contribuyente contribuyente = new Contribuyente("Jorge", "32-1234232-5");
        //creacion de impuesto
        Impuesto impuesto = new Impuesto(5000.0, contribuyente);
        //creacion mde calculadora
        Calculadora calc = new Calculadora();
        calc.calcular(impuesto); //Dependencia de uso
    }

}

```

#### Clase Impuesto

```

package tp5ej12;

public class Impuesto {
    private double monto;
    private Contribuyente contribuyente; //Asociacion unidireccional

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

    //getter
    public double getMonto() {
        return monto;
    }

    @Override
    public String toString() {

```

```

        return "Impuesto: $" + monto + " - " + contribuyente;
    }
}

```

## Clase Contribuyente

```

package tp5ej12;

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

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

    @Override
    public String toString() {
        return "Contribuyente: " + nombre + " CUIL: " + cuil;
    }
}

```

## Clase Calculadora

```

package tp5ej12;

public class Calculadora {

    //Dependencia de uso, recibe Impuesto como parametro y no lo
    guarda
    public void calcular(Impuesto impuesto) {
        System.out.println("Calculando impuesto: " + impuesto);
        // Ejemplo de cálculo (simple)
        double total = impuesto.getMonto() * 1.21; //ejemplo con el
        IVA del 21%
        System.out.println("Monto total con impuesto: $" + total);
    }
}

```

## 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

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

### Clases y atributos:

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

### Resultado:

### Clase tp5ej13 (contiene metodo Main):

```

package tp5ej13;

```

```

public class tp5ej13 {

    public static void main(String[] args) {
        //creacion del usuario
        Usuario usuario = new Usuario("Jorge", "Jorge@gmail.com");

        //creacion del generador de QR
        GeneradorQR generador = new GeneradorQR();
        //inicio del generador
        generador.generar("AW12321X", usuario); //Dependencia de
        creacion

    }

}

```

## Clase Usuario

```

package tp5ej13;

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

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

    @Override
    public String toString() {
        return "Usuario: " + nombre + " - Email: " + email;
    }
}

```

## Clase GeneradorQR

```

package tp5ej13;

public class GeneradorQR {

    //Dependencia de creacion, se crea un CodigoQR dentro del metodo

    public void generar(String valor, Usuario usuario) {
        CodigoQR qr = new CodigoQR(valor, usuario); //se crea el
        generador
        System.out.println("Generado: " + qr);
    }
}

```

## Clase CodigoQR

```

package tp5ej13;

public class CodigoQR {
    private String valor;
    private Usuario usuario; //Asociacion unidireccional

    //constructor
}

```

```

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

    @Override
    public String toString() {
        return "CódigoQR: " + valor + " - " + usuario;
    }
}

```

#### 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)

#### Resultado:

#### Clase tp5ej14 (contiene metodo Main):

```

package tp5ej14;

public class tp5ej14 {

    public static void main(String[] args) {

        //creacion del proyecto
        Proyecto proyecto = new Proyecto("Corto Animado", 15);

        //creacion del editor
        EditorVideo editor = new EditorVideo();
        //ejecuciones de las exportaciones
        editor.exportar("MP4", proyecto);
        editor.exportar("AVI", proyecto);
    }

}

```

#### Clase Render

```

package tp5ej14;

public class Render {
    private String formato;
    private Proyecto proyecto; //Asociacion unidireccional

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

    @Override

```

```

        public String toString() {
            return "Render en formato " + formato + " del " + proyecto;
        }
    }

```

## Clase Proyecto

```

package tp5ej14;

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

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

    @Override
    public String toString() {
        return "Proyecto: " + nombre + " (" + duracionMin + " min)";
    }
}

```

## Clase EditorVideo

```

package tp5ej14;

public class EditorVideo {

    //Dependencia de creacion
    public void exportar(String formato, Proyecto proyecto) {
        Render render = new Render(formato, proyecto); //se crea el
render
        System.out.println("Exportación realizada: " + render);
    }
}

```