

## PROGRAMACIÓN II

### Trabajo Práctico 5: Relaciones UML 1 a 1

Enlace a Repositorio: <https://github.com/YamiiiFer/UTN-TUPaD-P2.git>

#### 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:

i. Pasaporte: numero, fechaEmision

ii. Foto: imagen, formato

iii. Titular: nombre, dni

### Explicación de símbolos

#### 1. Pasaporte → Foto

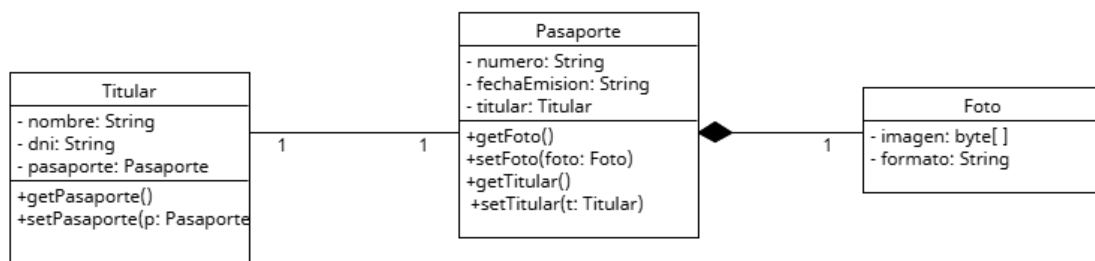
- **Rombo relleno (composición):** la foto **no puede existir** sin el pasaporte.
- En Java: `private Foto foto;` y se crea junto al pasaporte.

#### 2. Pasaporte ↔ Titular

- **Asociación bidireccional:** ambos se conocen.
- En Java: Pasaporte tiene `Titular titular` y Titular tiene `Pasaporte pasaporte`.

#### 3. Atributos

- Foto: imagen (`byte[]`) y formato (`String`)
- Pasaporte: numero y fechaEmision
- Titular: nombre y dni



```

package Ejercicio1;

public class Pasaaporte {
    //Atributos
    private String numero;
    private String fechaEmision;
    private Foto foto; //Composicion
    private Titular titular;//Asociacion bidireccional

    //Constructor

    public Pasaaporte(String numero, String fechaEmision, Foto foto) {
        this.numero = numero;
        this.fechaEmision = fechaEmision;
        this.foto = 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 void setFoto(Foto foto) {
    this.foto = foto;
}

```

```

public Titular getTitular() {
    return titular;
}

public void setTitular(Titular titular) {
    this.titular = titular;
    titular.setPasaaporte(this); // mantiene la bidireccionalidad
}

```

```

public void mostrarInfo() {
    System.out.println("Número: " + numero + ", Fecha de Emisión: " + fechaEmision);
    if (foto != null) {
        System.out.println("Formato foto: " + foto.getFormato());
    }
    if (titular != null) {
        System.out.println("Titular: " + titular.getNombre());
    }
}
}

```

```

package Ejercicio1;

public class Foto {
    private String nombre;
    private String formato;

    //Constructor

    public Foto(String nombre, String formato) {
        this.nombre = nombre;
        this.formato = formato;
    }

    // Getters y setters

    public String getNombre() {
        return nombre;
    }

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

    public String getFormato() {
        return formato;
    }

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

    public void mostrarInfo(){
        System.out.println("Nombre de la imagen: " + nombre + ", Formato: " + formato);
    }
}

```

```

package Ejercicio1;

public class Titular {

    private String nombre;
    private String dni;
    private Pasaporte pasaporte; // asociación bidireccional

    //Constructor
    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;
}

public Pasaporte getPasaporte() {
    return pasaporte;
}

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

public void mostrarInfo() {
    System.out.println("Nombre: " + nombre + ", DNI: " + dni);
    if (pasaporte != null) {
        System.out.println("Pasaporte asociado: " + pasaporte.getNumero());
    }
}
}

```

```

package Ejercicio1;

public class Principal {
    public static void main(String[] args) {
        // Crear la foto
        Foto nuevaImagen = new Foto("osito", "jpg");

        // Crear el pasaporte con la foto
        Pasaporte pasaporte = new Pasaporte("A1234567", "2025-10-19", nuevaImagen);

        // Crear el titular
        Titular titular = new Titular("Juan Pérez", "12345678");

        // Asociar pasaporte y titular (bidireccional)
        pasaporte.setTitular(titular);

        // Mostrar información
        pasaporte.mostrarInfo();
        System.out.println();
        titular.mostrarInfo();
    }
}

```

#### Output - Práctica UML (run)

```

run:
Número: A1234567, Fecha de Emisión: 2025-10-19
Formato foto: jpg
Titular: Juan Pérez

Nombre: Juan Pérez, DNI: 12345678
Pasaporte asociado: A1234567
BUILD SUCCESSFUL (total time: 0 seconds)

```

## 2. Celular - Batería - Usuario

a. Agregación: Celular → Batería

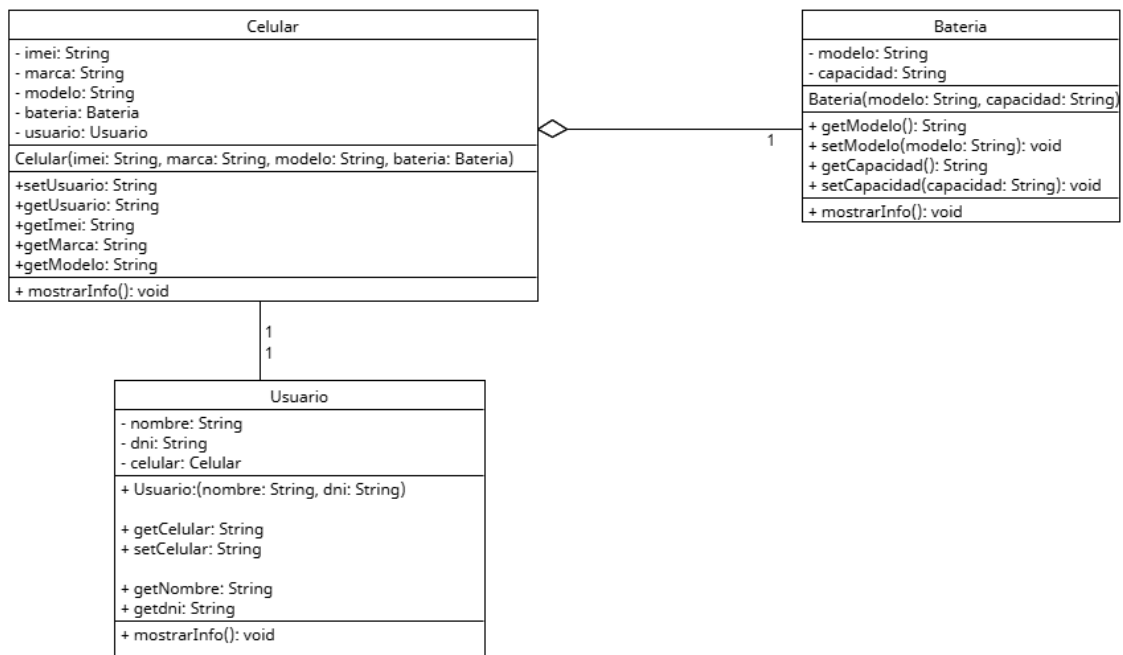
b. Asociación bidireccional: Celular ↔ Usuario

Clases y atributos:

i. Celular: imei, marca, modelo

ii. Batería: modelo, capacidad

iii. Usuario: nombre, dni



```
package Ejercicio2;

public class Batería {
    // Atributos
    private String modelo;
    private String capacidad;

    // Constructor
    public Batería(String modelo, String capacidad) {
        this.modelo = modelo;
        this.capacidad = capacidad;
    }
}
```

```
// Getters y Setters
public String getModelo() {
    return modelo;
}

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

public String getCapacidad() {
    return capacidad;
}

public void setCapacidad(String capacidad) {
    this.capacidad = capacidad;
}
```

```
// Método para mostrar información
public void mostrarInfo() {
    System.out.println("Modelo de batería: " + modelo);
    System.out.println("Capacidad: " + capacidad);
}
```