

PROGRAMACIÓN II

Trabajo Práctico 5: Relaciones UML 1 a 1

Alumno: Mauro Zavatti

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

```
public class Pasaporte {
    private String numero;
    private LocalDate fechaEmision;

    // Composición: el Pasaporte crea/posee su Foto
    private Foto foto;

    // Asociación 1⇔1 con Titular (bidireccional)
    private Titular titular;

    public Pasaporte(String numero, LocalDate fechaEmision, String img, String formato, Titular titular) {
        this.numero = numero;
        this.fechaEmision = fechaEmision;
        this.foto = new Foto(img, formato); // composición 1⇔1
        setTitular(titular);               // establece la asociación
    }

    public String getNumero() { return numero; }
    public LocalDate getFechaEmision() { return fechaEmision; }
    public Foto getFoto() { return foto; }
    public Titular getTitular() { return titular; }

    // Mantener bidireccionalidad con Titular
    public void setTitular(Titular t) {
        if (this.titular = t) return;

        Titular anterior = this.titular;
        this.titular = t;

        if (anterior != null && anterior.getPasaporte() == this) {
            anterior.setPasaporte(null);
        }
        if (t != null && t.getPasaporte() != this) {
            t.setPasaporte(this);
        }
    }

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

```

public class Titular {
    private String nombre;
    private String dni;
    private Pasaporte pasaporte; // lado opuesto de la asociación

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

    public String getNombre() { return nombre; }
    public String getDni() { return dni; }
    public Pasaporte getPasaporte() { return pasaporte; }

    // Mantener la bidireccionalidad de forma segura
    public void setPasaporte(Pasaporte p) {
        // Evitar bucles y estados inconsistentes
        if (this.pasaporte == p) return;

        // Desvincular el anterior
        Pasaporte anterior = this.pasaporte;
        this.pasaporte = p;

        if (anterior != null && anterior.getTitular() == this) {
            anterior.setTitular(null); // corta el vínculo del otro lado
        }
        if (p != null && p.getTitular() != this) {
            p.setTitular(this); // asegura la relación en el otro lado
        }
    }

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

```

```

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

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

    public String getImagen() { return imagen; }
    public String getFormato() { return formato; }

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

```

```

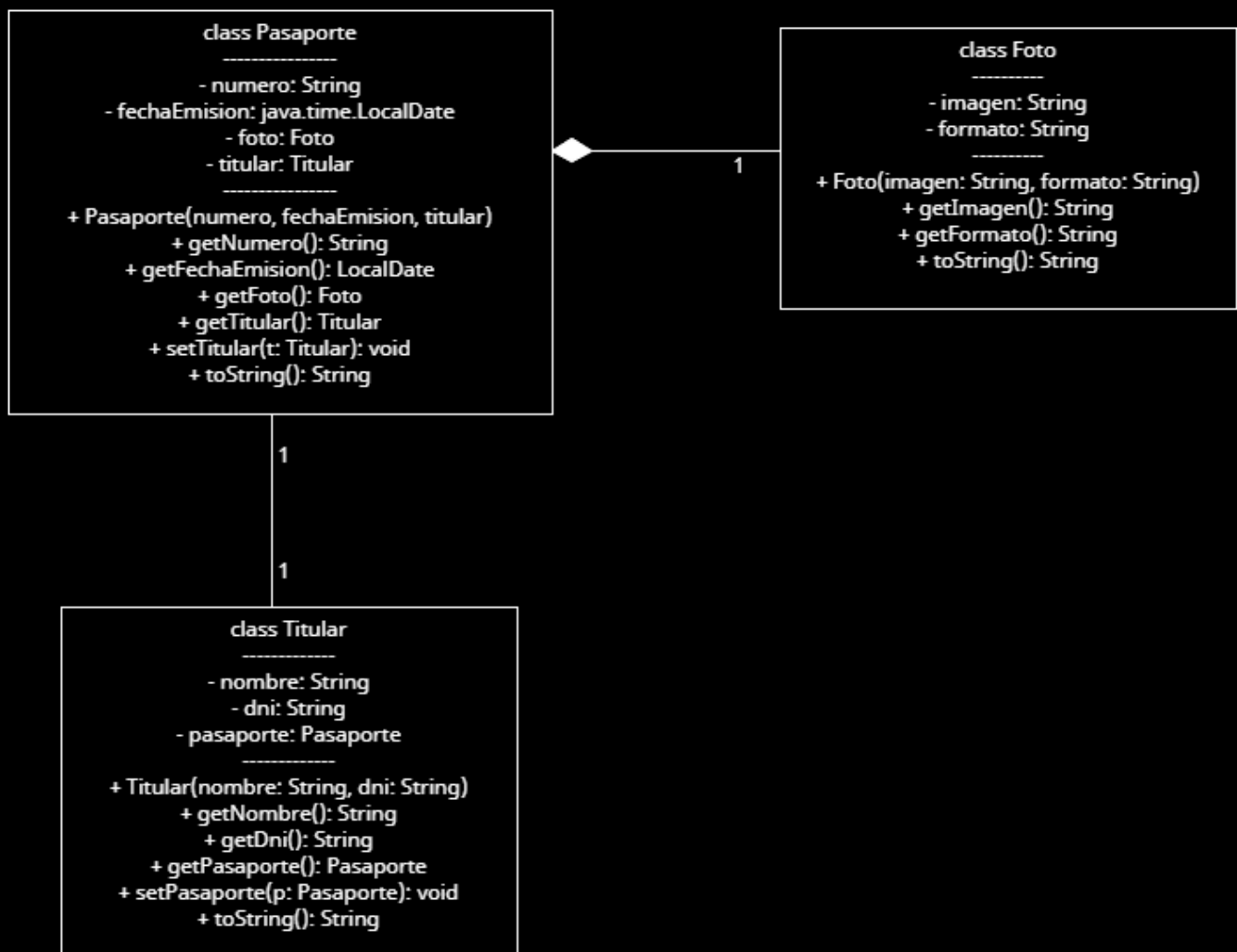
import java.time.LocalDate;

public class PasaporteMain {
    public static void main(String[] args) {
        Titular t = new Titular("Mauro Zavatti", "40545665");

        Pasaporte p = new Pasaporte(
            "AR-1234567",
            LocalDate.of(2025, 1, 10),
            "mauro.png",
            "PNG",
            t
        );

        System.out.println(p);
        System.out.println(t);
        System.out.println("¿Bidireccional? t.getPasaporte() == p ⇒ " + (t.getPasaporte() == p));
    }
}

```



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

```
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, Bateria bateria, Usuario usuario) {
        this.imei = imei;
        this.marca = marca;
        this.modelo = modelo;
        this.bateria = bateria;
        this.usuario = usuario;
        usuario.setCelular(this);
    }

    public String getImei() { return imei; }
    public String getMarca() { return marca; }
    public String getModelo() { return modelo; }
    public Bateria getBateria() { return bateria; }
    public Usuario getUsuario() { return usuario; }

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

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

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

    public String getModelo() { return modelo; }
    public int getCapacidad() { return capacidad; }

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

```

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 String getDni() { return dni; }
    public Celular getCelular() { return celular; }
    public void setCelular(Celular celular) { this.celular = celular; }

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

```

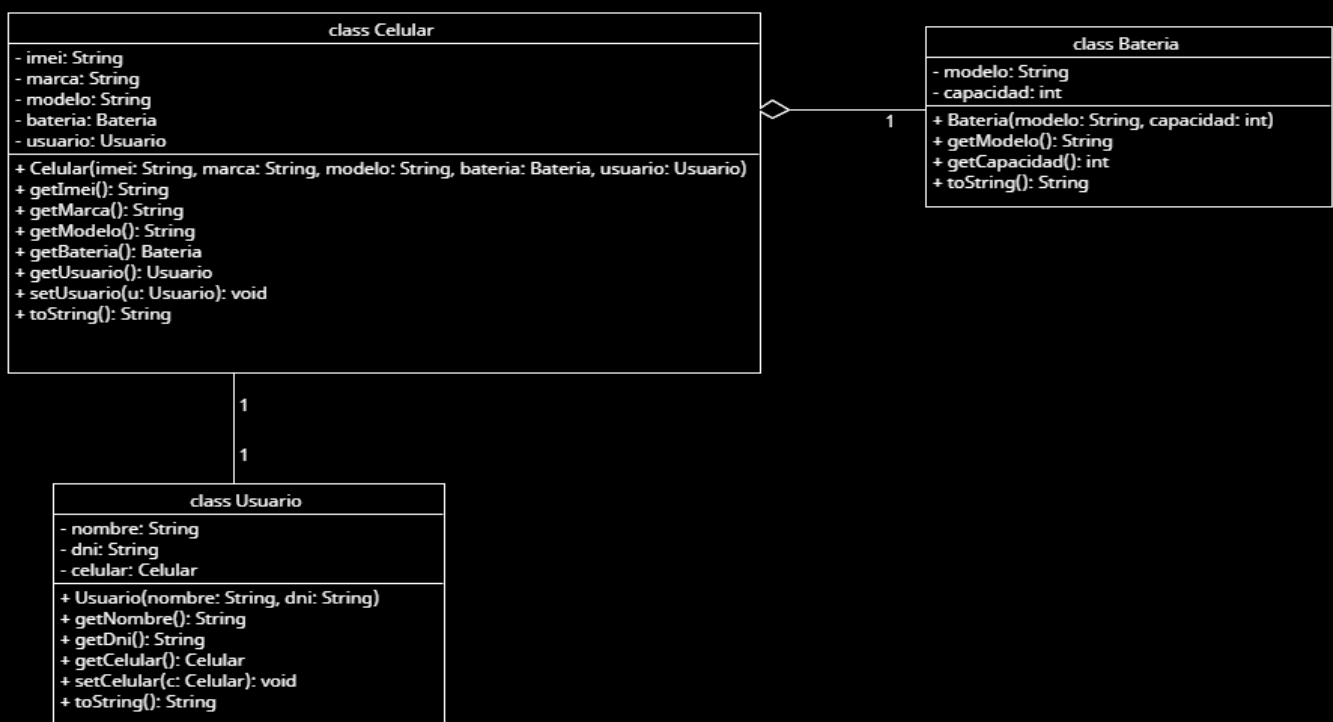
```

public class CelularMain {

    public static void main(String[] args) {
        Bateria b1 = new Bateria("Samsung-123", 4500);
        Usuario u1 = new Usuario("Mauro", "40555666");
        Celular c1 = new Celular("111222333", "Samsung", "S23", b1, u1);

        System.out.println(c1);
        System.out.println(u1.getCelular());
    }
}

```



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, direccion

```
public class Libro {
    private String titulo;
    private String isbn;
    private Autor autor; // Asociación unidireccional
    private Editorial editorial; // Agregación

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

    public String getTitulo() {
        return titulo;
    }

    public String getIsbn() {
        return isbn;
    }

    public Autor getAutor() {
        return autor;
    }

    public Editorial getEditorial() {
        return editorial;
    }

    @Override
    public String toString() {
        return "Libro{" +
            "titulo='" + titulo + '\'' +
            ", isbn='" + isbn + '\'' +
            ", autor=" + autor +
            ", editorial=" + editorial +
            '}';
    }
}
```

```

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;
    }

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

```

```

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;
    }

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

```

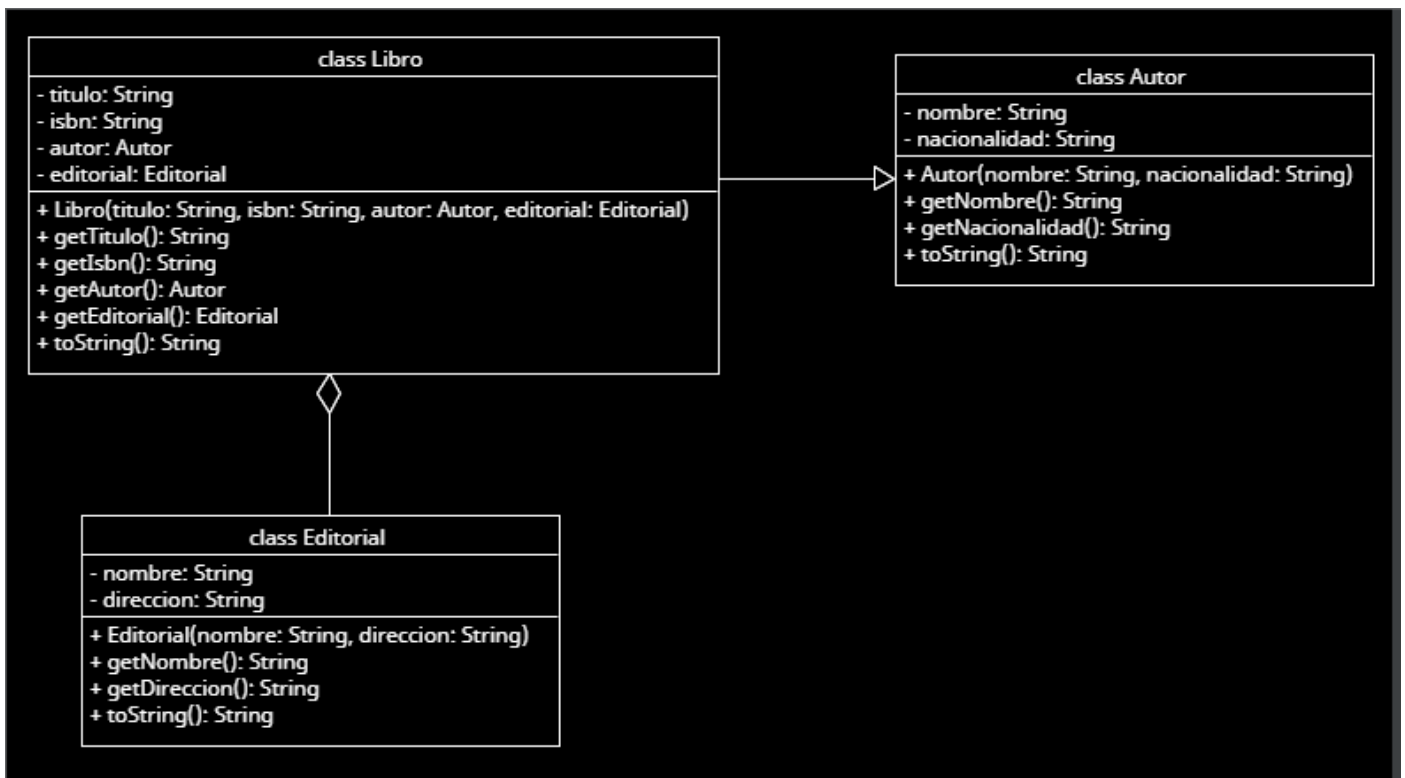
```

public class LibroMain {
    public static void main(String[] args) {
        Autor autor1 = new Autor("Gabriel García Márquez", "Colombiana");
        Editorial editorial1 = new Editorial("Sudamericana", "Buenos Aires");

        Libro libro1 = new Libro("Cien Años de Soledad", "978-987-1138", autor1, editorial1);

        System.out.println(libro1);
    }
}

```



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, cuil


```

public class TarjetaDeCredito {
    private String numero;
    private LocalDate fechaVencimiento;
    private Cliente cliente; // Asociación bidireccional
    private Banco banco; // Agregación

    public TarjetaDeCredito(String numero, LocalDate fechaVencimiento, Cliente cliente, Banco banco) {
        this.numero = numero;
        this.fechaVencimiento = fechaVencimiento;
        this.cliente = cliente;
        this.banco = banco;
        cliente.setTarjeta(this); // vínculo bidireccional
    }

    public String getNumero() {
        return numero;
    }

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

```

```

public class Cliente {
    private String nombre;
    private String dni;
    private TarjetaDeCredito tarjeta; // Asociación bidireccional

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

    public String getNombre() {
        return nombre;
    }

    public String getDni() {
        return dni;
    }

    public void setTarjeta(TarjetaDeCredito tarjeta) {
        this.tarjeta = tarjeta;
    }

    public TarjetaDeCredito getTarjeta() {
        return tarjeta;
    }

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

```

```

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;
    }

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

```

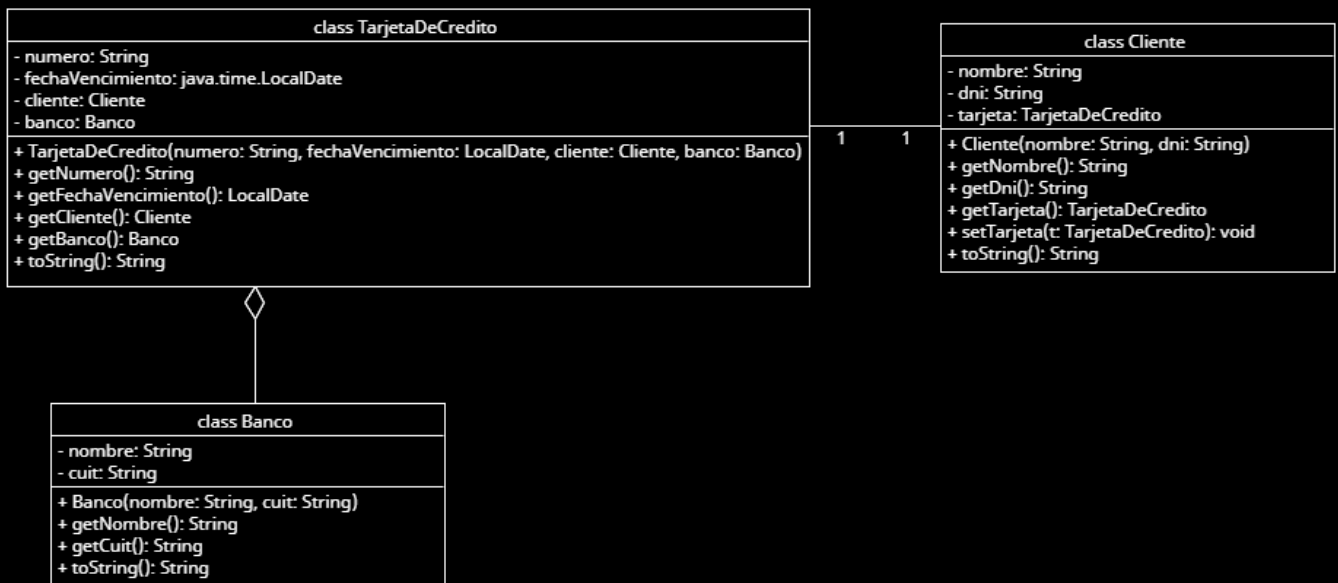
```

public class TarjetaDeCreditoMain {
    public static void main(String[] args) {
        Banco banco = new Banco("Banco Nación", "30-12345678-9");
        Cliente cliente = new Cliente("Mauro Zavatti", "40545665");

        TarjetaDeCredito tarjeta = new TarjetaDeCredito(
            "1234-5678-9876-5432",
            LocalDate.of(2027, 5, 31),
            cliente,
            banco
        );

        System.out.println(tarjeta);
        System.out.println("La tarjeta de " + cliente.getNombre() + " pertenece a: " + cliente.getTarjeta().getNumero());
    }
}

```



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

```
//
public class Computadora {
    private String marca;
    private String numeroSerie;
    private PlacaMadre placaMadre; // composición
    private Propietario propietario; // asociación bidireccional

    public Computadora(String marca, String numeroSerie, String modeloPM, String chipset, Propietario propietario) {
        this.marca = marca;
        this.numeroSerie = numeroSerie;
        this.placaMadre = new PlacaMadre(modeloPM, chipset); // composición
        this.propietario = propietario;
        if (propietario != null) propietario.setComputadora(this);
    }

    public String getMarca() { return marca; }
    public String getNumeroSerie() { return numeroSerie; }
    public PlacaMadre getPlacaMadre() { return placaMadre; }
    public Propietario getPropietario() { return propietario; }

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

```
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 String getChipset() { return chipset; }

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

```

public class Propietario {
    private String nombre;
    private String dni;
    private Computadora computadora; // asociación bidireccional

    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;
    }

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

```

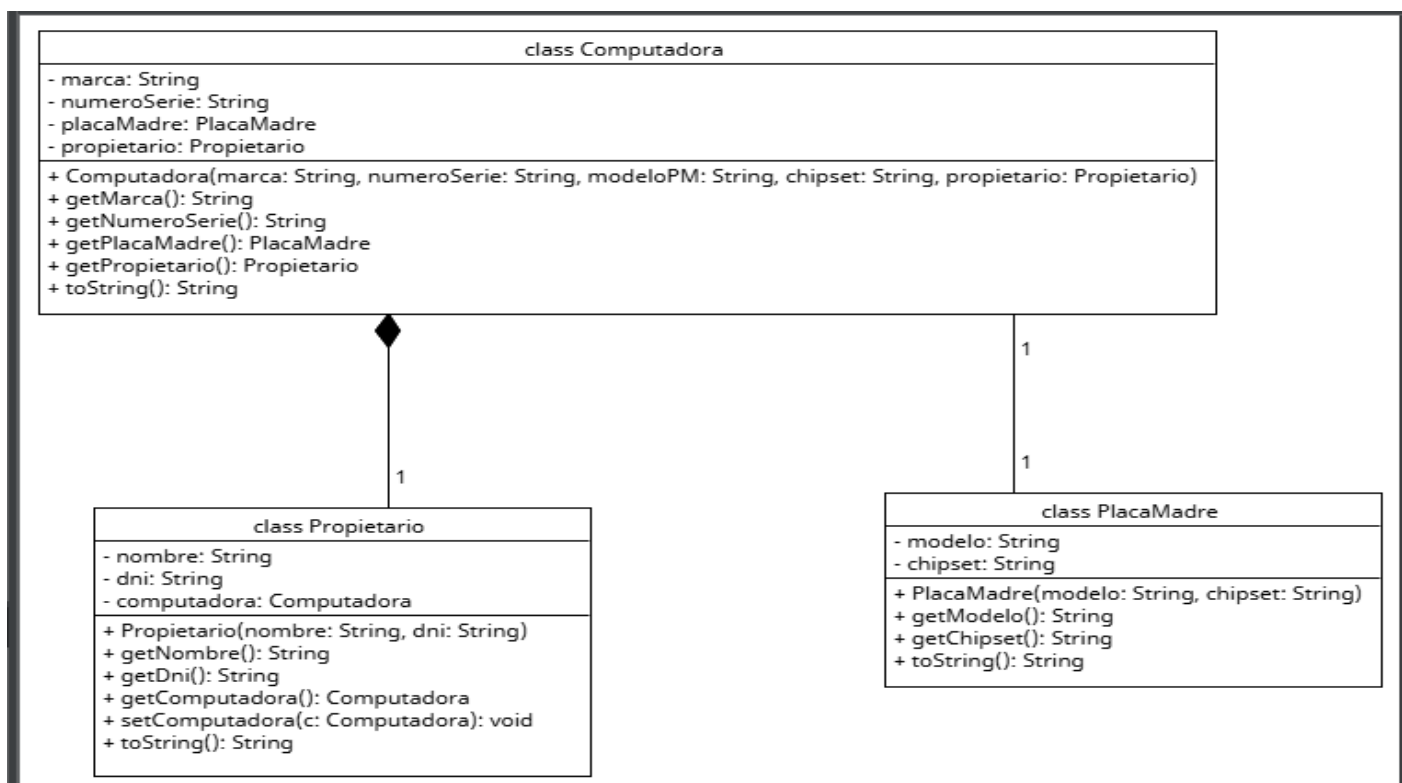
```

public class ComputadoraMain {
    public static void main(String[] args) {
        Propietario propietario = new Propietario("Mauro Zavatti", "40545665");

        Computadora compu = new Computadora(
            "Lenovo",
            "SN123456789",
            "B450M-A",
            "AMD B450",
            propietario
        );

        System.out.println(compu);
        System.out.println("La computadora pertenece a: " + propietario.getNombre());
        System.out.println("Placa madre: " + compu.getPlacaMadre().getModelo());
    }
}

```



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

```
public class Reserva {
    private String fecha;
    private String hora;
    private Cliente cliente; // Asociación unidireccional
    private Mesa mesa;      // Agregación

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

    public String getFecha() { return fecha; }
    public String getHora() { return hora; }
    public Cliente getCliente() { return cliente; }
    public Mesa getMesa() { return mesa; }

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

```
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; }

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

```

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; }

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

```

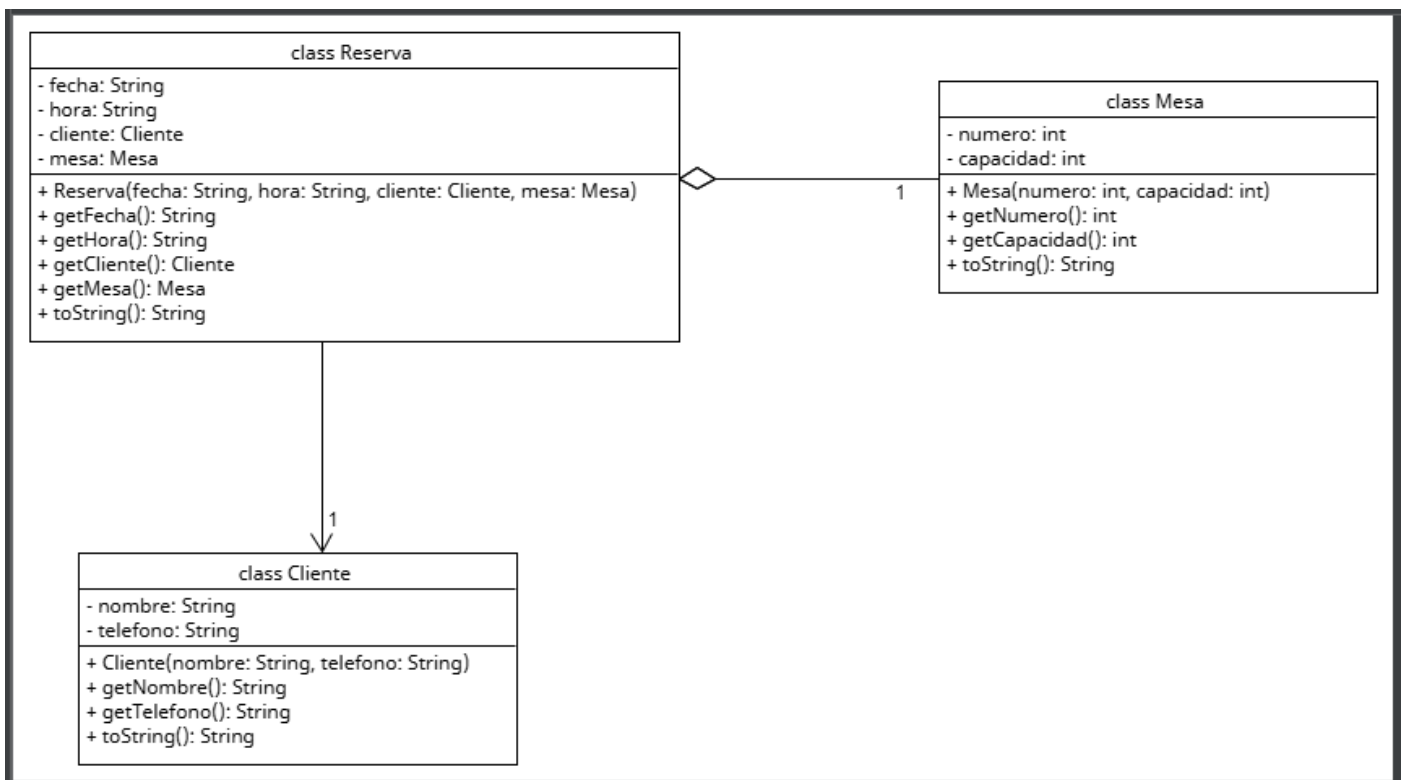
```

public class ReservaMain {
    public static void main(String[] args) {
        Cliente cliente = new Cliente("Mauro Zavatti", "3412345678");
        Mesa mesa = new Mesa(5, 4);

        Reserva reserva = new Reserva("2025-09-15", "21:00", cliente, mesa);

        System.out.println(reserva);
        System.out.println("Reserva a nombre de: " + reserva.getClient().getNombre());
    }
}

```



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

```
public class Vehiculo {
    private String patente;
    private String modelo;
    private Motor motor; // agregación
    private Conductor conductor; // asociación bidireccional

    public Vehiculo(String patente, String modelo, Motor motor, Conductor conductor) {
        this.patente = patente;
        this.modelo = modelo;
        this.motor = motor;
        this.conductor = conductor;
        if (conductor != null) conductor.setVehiculo(this); // mantener bidireccionalidad
    }

    @Override
    public String toString() {
        return "Vehiculo{patente='" + patente + "', modelo='" + modelo +
            "', motor='" + motor +
            "', conductor='" + (conductor != null ? conductor.toString() : "sin conductor") +
            "'}";
    }
}
```

```
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 + "'}";
    }
}
```

```

public class Conductor {
    private String nombre;
    private String licencia;
    private Vehiculo vehiculo; // asociación bidireccional

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

    public void setVehiculo(Vehiculo vehiculo) {
        this.vehiculo = vehiculo;
    }

    public Vehiculo getVehiculo() {
        return vehiculo;
    }

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

```

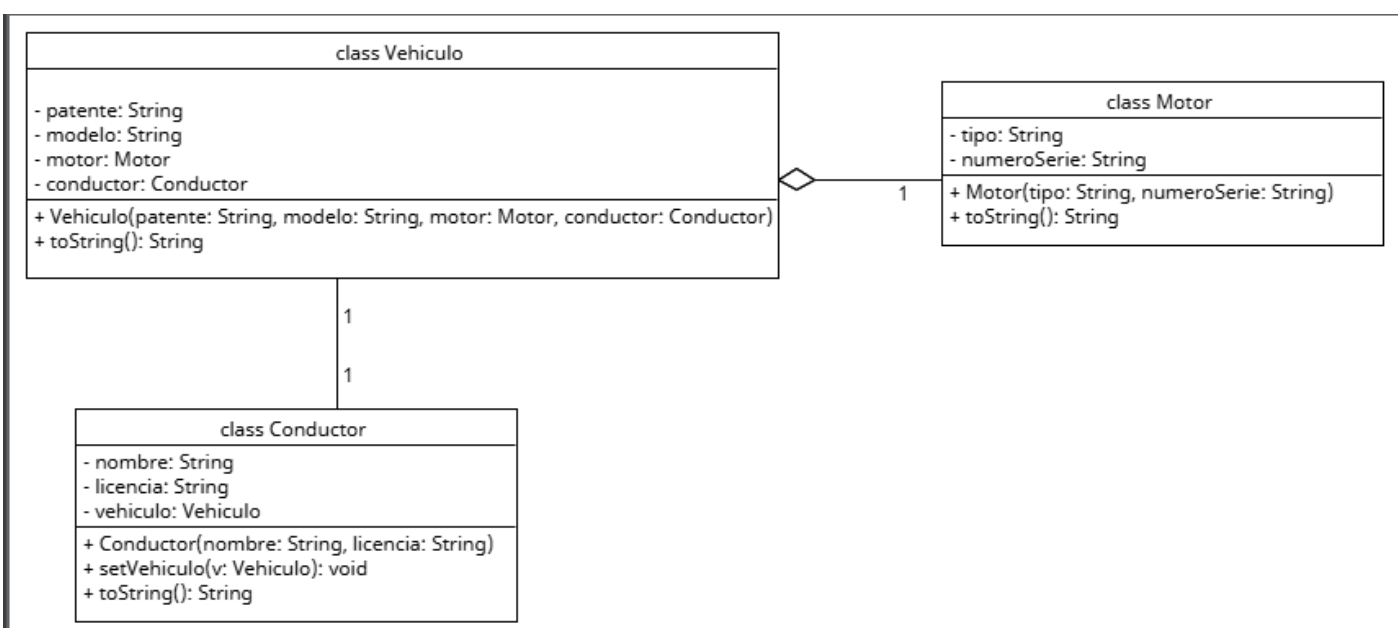
```

public class VehiculoMain {
    public static void main(String[] args) {
        Motor motor = new Motor("Nafta", "SN-12345");
        Conductor conductor = new Conductor("Mauro Zavatti", "LIC-98765");

        Vehiculo vehiculo = new Vehiculo("AB123CD", "Toyota Corolla", motor, conductor);

        System.out.println(vehiculo);
        System.out.println("El vehiculo de " + conductor.toString() + " es: " + conductor.getVehiculo());
    }
}

```



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

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

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

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

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

    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 + "}";
    }
}
```

```

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 + "'}";
    }
}

```

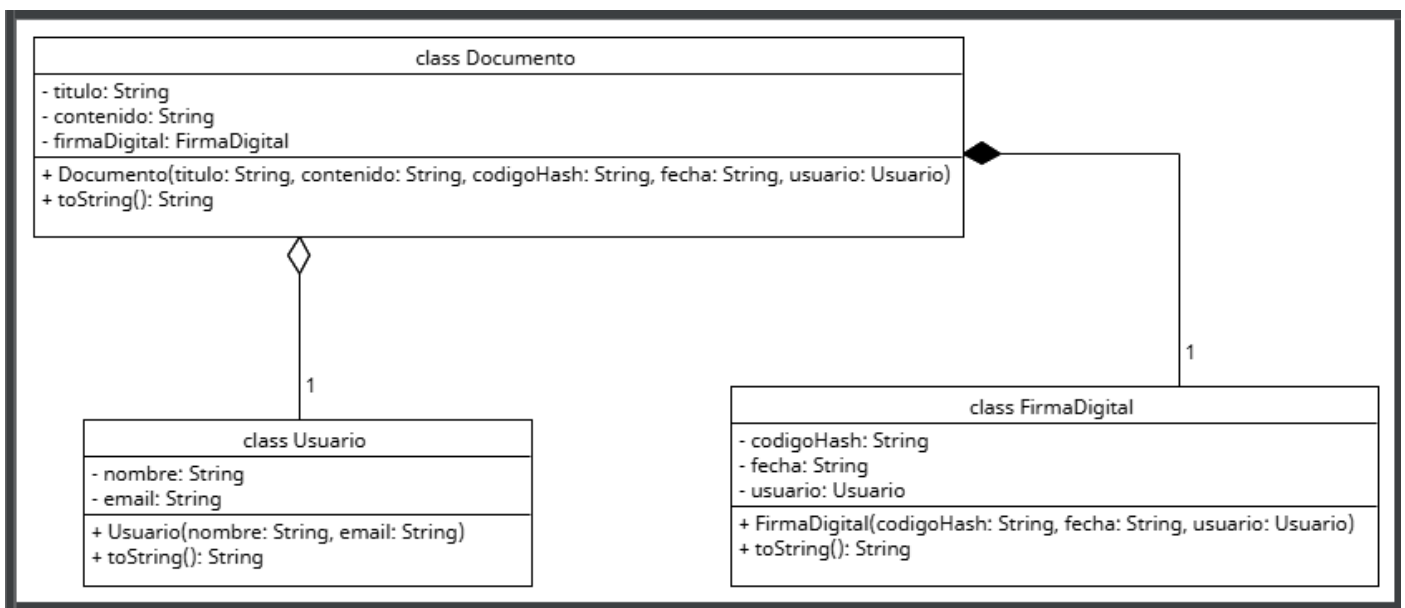
```

public class DocumentoMain {
    public static void main(String[] args) {
        Usuario usuario = new Usuario("Mauro Zavatti", "mauro@mail.com");

        Documento doc = new Documento(
            "Contrato de Servicio",
            "Contenido del contrato ... ",
            "HASH123ABC",
            "2025-09-15",
            usuario
        );

        System.out.println(doc);
    }
}

```



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

```
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, 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 + "'}";
    }
}
```

```
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 + "'}";
    }
}
```

```
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 + "'}";
    }
}
```

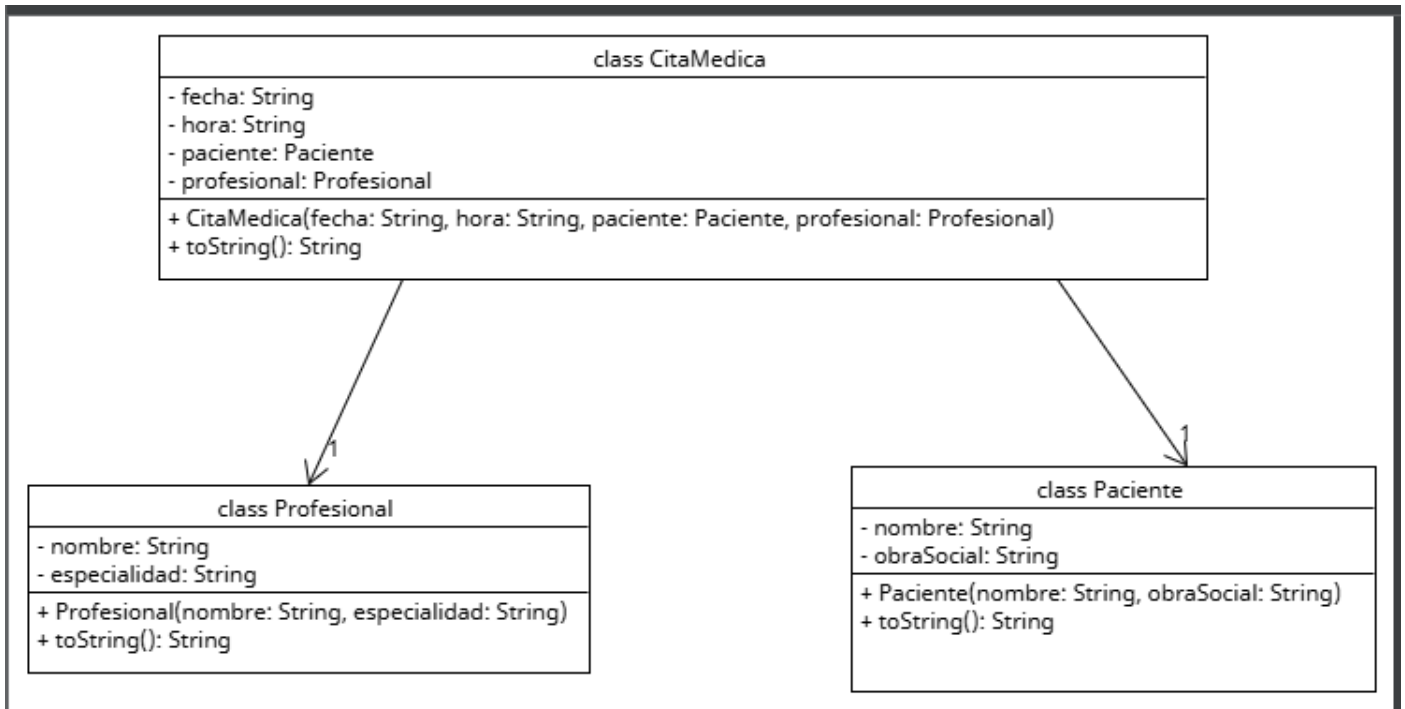
```

public class CitaMedicaMain {
    public static void main(String[] args) {
        Paciente paciente = new Paciente("Mauro Zavatti", "OSDE");
        Profesional profesional = new Profesional("Dra. Laura Pérez", "Cardiología");

        CitaMedica cita = new CitaMedica("2025-09-20", "15:30", paciente, profesional);

        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.

```

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

    public CuentaBancaria(String cbu, double saldo, String codigo, String ultimaModificacion, Titular titular) {
        this.cbu = cbu;
        this.saldo = saldo;
        this.claveSeguridad = new ClaveSeguridad(codigo, ultimaModificacion); // composición
        this.titular = titular;
        if (titular != null) titular.setCuenta(this); // mantener bidireccionalidad
    }

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

```

```

public class ClaveSeguridad {
    private String codigo;
    private String ultimaModificacion;

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

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

```

```

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 void setCuenta(CuentaBancaria cuenta) {
        this.cuenta = cuenta;
    }

    public CuentaBancaria getCuenta() {
        return cuenta;
    }

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

```

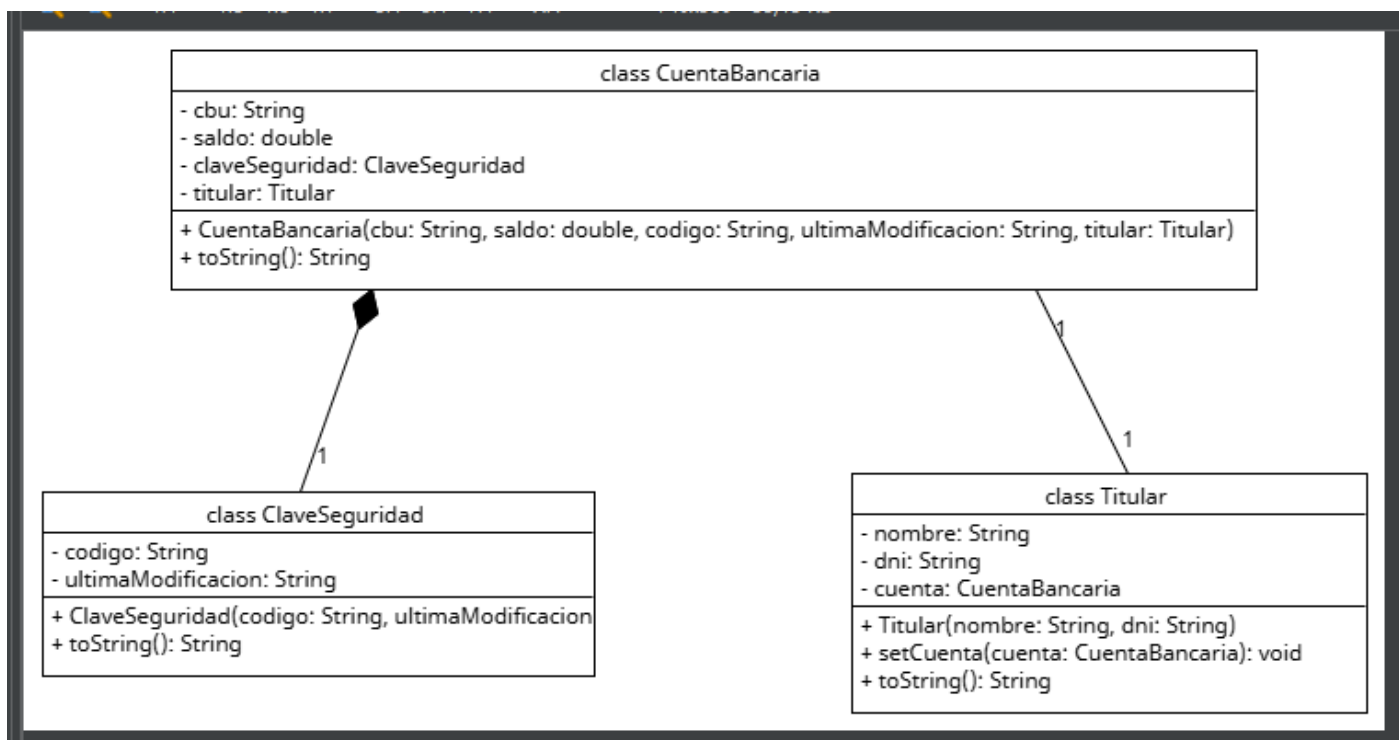
```

public class CuentaBancariaMain {
    public static void main(String[] args) {
        Titular titular = new Titular("Mauro Zavatti", "40545665");

        CuentaBancaria cuenta = new CuentaBancaria(
            "1234567890001112223334",
            150000.50,
            "SEG-98765",
            "2025-09-15",
            titular
        );

        System.out.println(cuenta);
        System.out.println("La cuenta de " + titular + " es: " + titular.getCuenta());
    }
}

```



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)

```

public class Reproductor {
    // Dependencia de uso: recibe Cancion como parámetro
    void reproducir(Cancion cancion) {
        System.out.println("Reproduciendo: " + cancion.titulo
            + " de " + cancion.artista.nombre);
    }
}
  
```

```

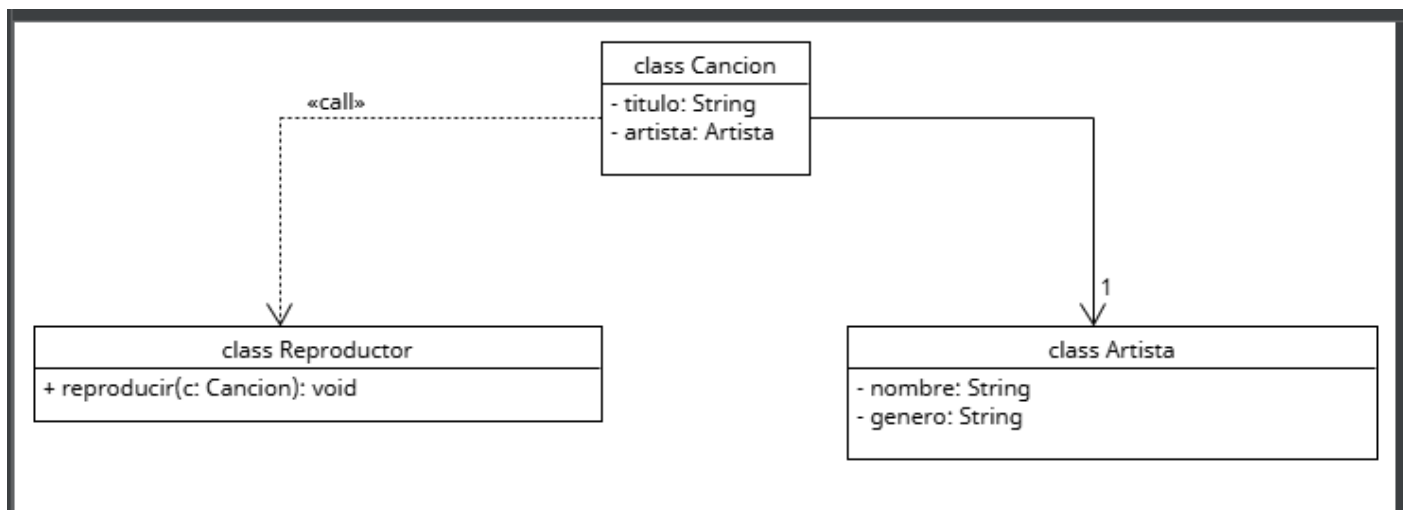
public class Cancion {
    String titulo;
    Artista artista; // Asociación unidireccional hacia Artista
}
  
```

```
public class Artista {
    String nombre;
    String genero;
}
```

```
public class ReproductorMain {
    public static void main(String[] args) {
        Artista artista = new Artista();
        artista.nombre = "Babasonicos";
        artista.genero = "Rock alternativo";

        Cancion cancion = new Cancion();
        cancion.titulo = "Posesion del Tercer Tipo";
        cancion.artista = artista;

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



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)`

```

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

    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 + "}";
    }
}

```

```

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 + "'}";
    }
}

```

```

public class Calculadora {
    // Dependencia de uso
    public void calcular(Impuesto impuesto) {
        System.out.println("Calculando impuesto para " + impuesto.getContribuyente());
        System.out.println("Monto del impuesto: $" + impuesto.getMonto());
        double total = impuesto.getMonto() * 1.21; // por ejemplo, con IVA del 21%
        System.out.println("Total con IVA: $" + total);
    }
}

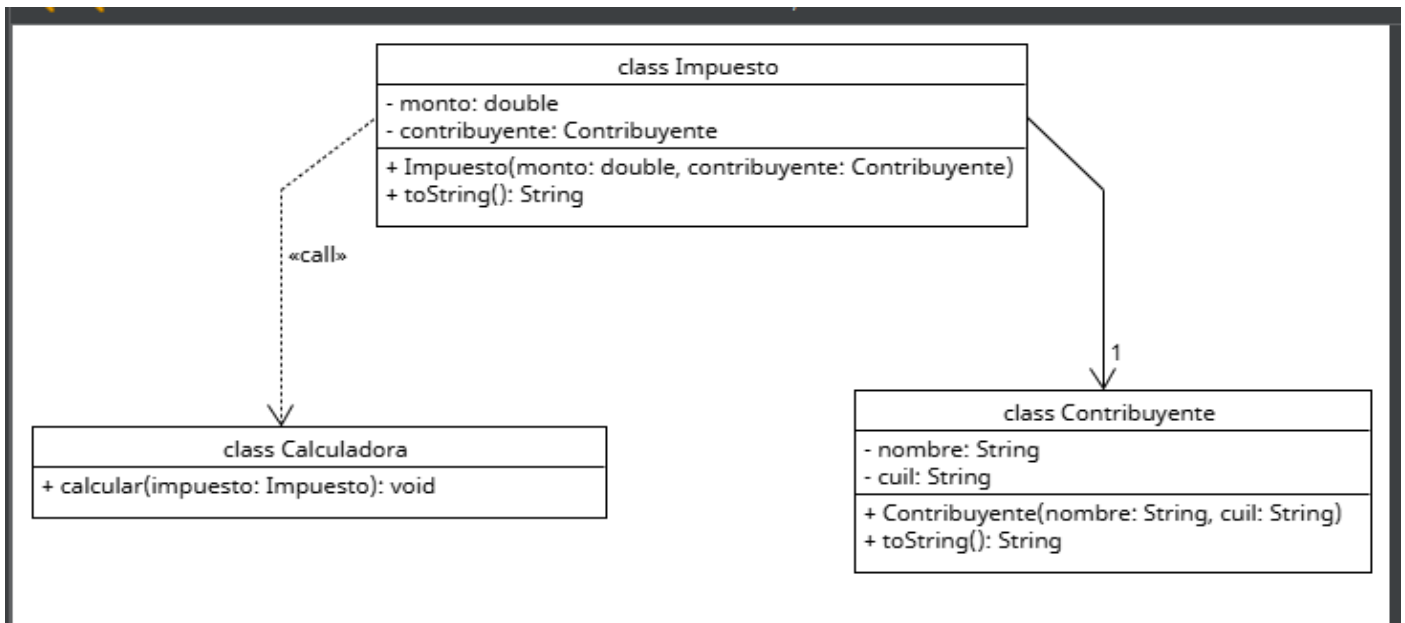
```

```

public class ImpuestoMain {
    public static void main(String[] args) {
        Contribuyente contribuyente = new Contribuyente("Mauro Zavatti", "20-40545665-3");
        Impuesto impuesto = new Impuesto(10000, contribuyente);

        Calculadora calculadora = new Calculadora();
        calculadora.calcular(impuesto);
    }
}

```

Ejercicios de Dependencia de Creación

13. GeneradorQR - Usuario - CódigoQR

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

Clases y atributos:

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

```

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

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

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

```

public class GeneradorQR {
    // Dependencia de creación: genera un nuevo objeto CodigoQR
    public CodigoQR generar(String valor, Usuario usuario) {
        return new CodigoQR(valor, usuario);
    }
}

```

```

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 + "'}";
    }
}

```

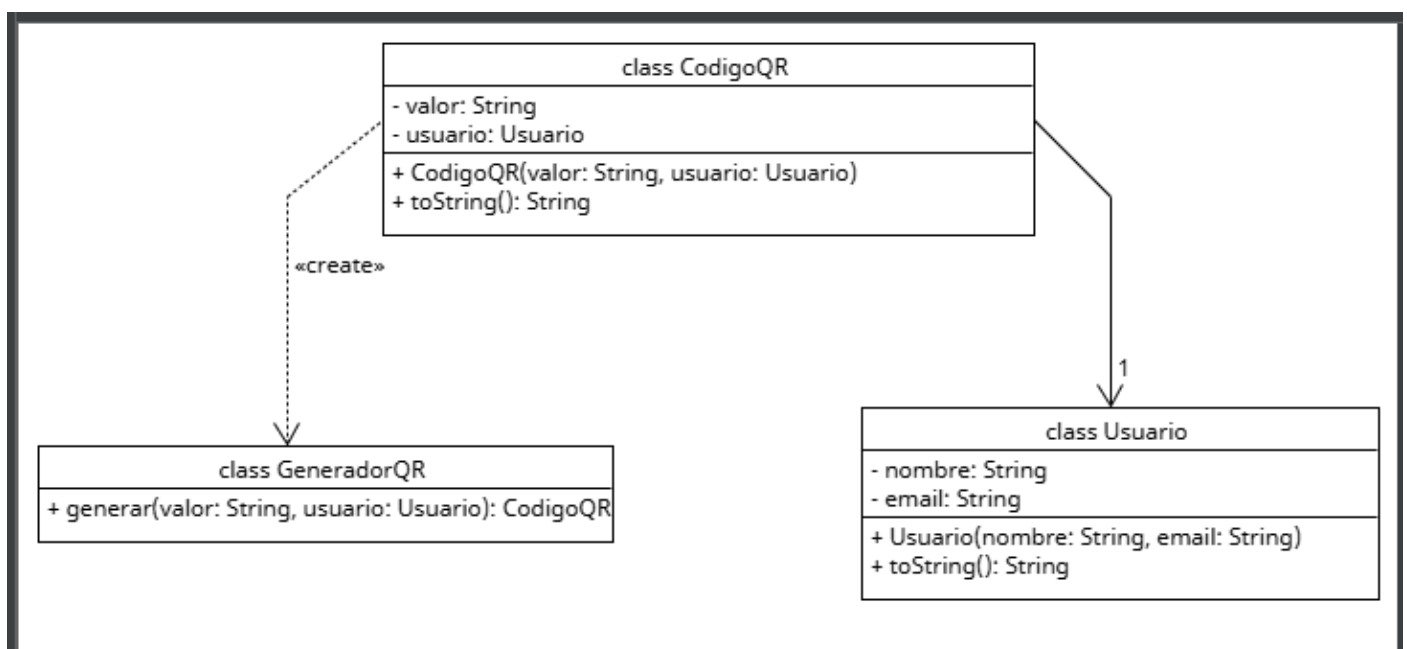
```

public class CodigoQRMain {
    public static void main(String[] args) {
        Usuario usuario = new Usuario("Mauro Zavatti", "mauro@exaplemail.com");

        GeneradorQR generador = new GeneradorQR();
        CodigoQR codigo = generador.generar("QR-ABC-123", usuario);

        System.out.println(codigo);
    }
}

```



14. EditorVideo - Proyecto - Render

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

```
public class EditorVideo {  
    // Dependencia de creación: genera un objeto Render a partir de un Proyecto  
    public Render exportar(String formato, Proyecto proyecto) {  
        System.out.println("Exportando proyecto '" + proyecto + "' en formato " + formato);  
        return new Render(formato, proyecto);  
    }  
}
```

```
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 + "}";  
    }  
}
```

```
public class Render {  
    private String formato;  
    private Proyecto proyecto; // asociación unidireccional  
  
    public Render(String formato, Proyecto proyecto) {  
        this.formato = formato;  
        this.proyecto = proyecto;  
    }  
  
    @Override  
    public String toString() {  
        return "Render{formato='" + formato + "', proyecto=" + proyecto + "}";  
    }  
}
```

```
public class EditorVideoMain {  
    public static void main(String[] args) {  
        Proyecto proyecto = new Proyecto("Documental Naturaleza", 120);  
  
        EditorVideo editor = new EditorVideo();  
        Render render = editor.exportar("MP4", proyecto);  
  
        System.out.println(render);  
    }  
}
```

