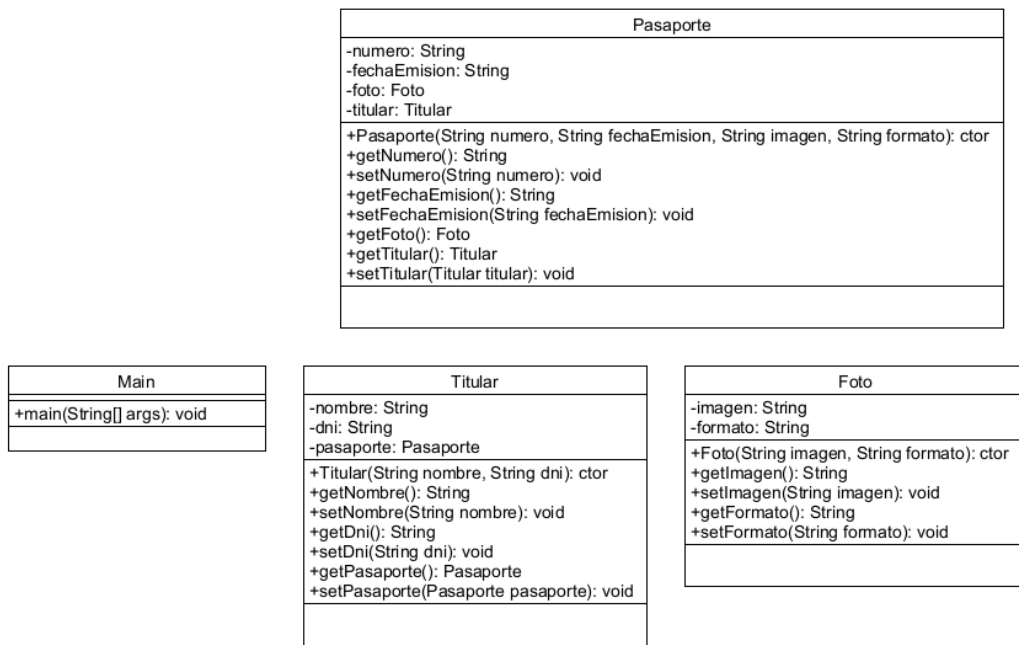


## Trabajo Práctico 5: Relaciones UML 1 a 1

### 1) Análisis de relaciones:

Composición: Pasaporte —◆— Foto

Asociación bidireccional: Pasaporte — Titular



```

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;
    }
}
  
```

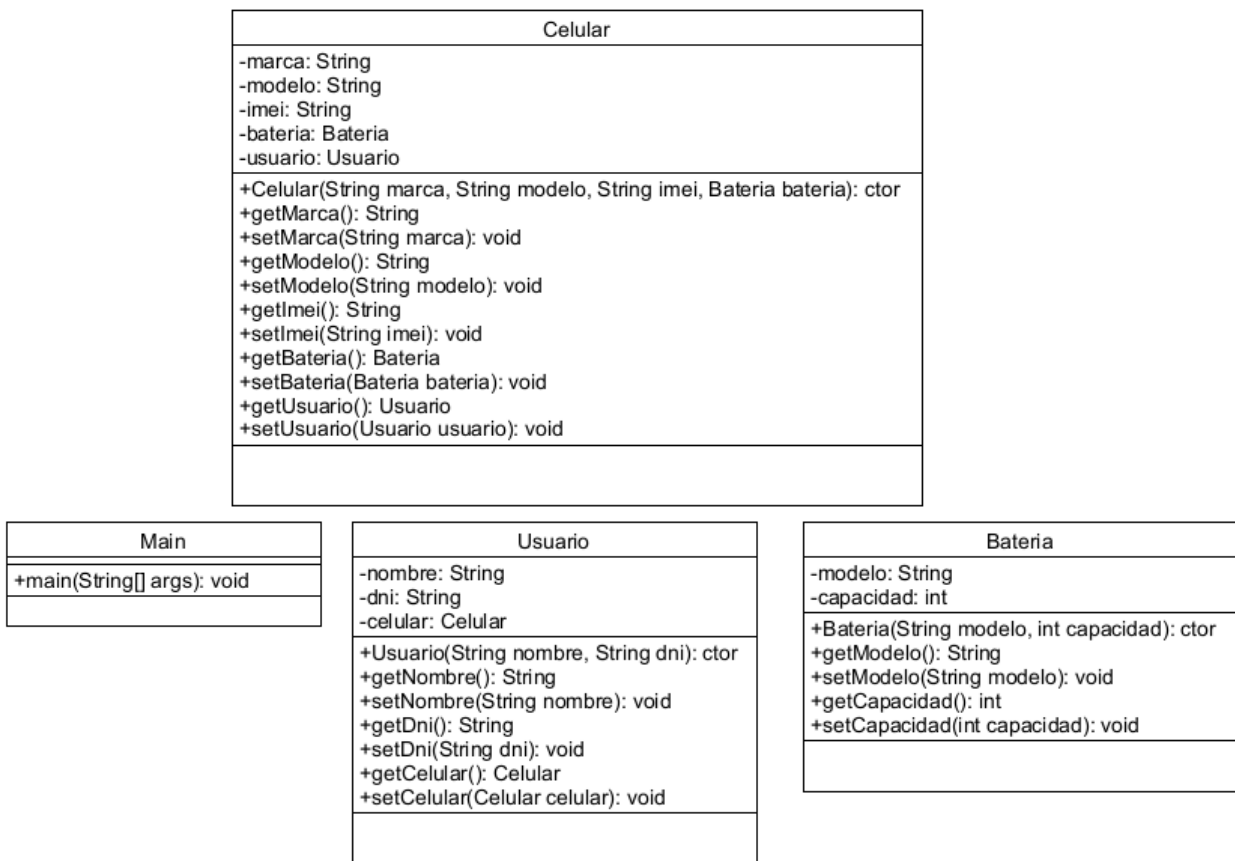
```
public class Titular {  
    private String nombre;  
    private String dni;  
    private Pasaporte pasaporte; // Asociación bidireccional  
  
    public Titular(String nombre, String dni) {  
        this.nombre = nombre;  
        this.dni = dni;  
    }  
  
    public void setPasaporte(Pasaporte pasaporte) {  
        this.pasaporte = pasaporte;  
    }  
  
    public Pasaporte getPasaporte() {  
        return pasaporte;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public String getDni() {  
        return dni;  
    }  
}
```

```
public class Pasaporte {  
    private String numero;  
    private String fechaEmision;  
    private Foto foto; // Composición  
    private Titular titular; // Asociación bidireccional  
  
    public Pasaporte(String numero, String fechaEmision, Foto foto, Titular titular) {  
        this.numero = numero;  
        this.fechaEmision = fechaEmision;  
        this.foto = foto;  
        this.titular = titular;  
        titular.setPasaporte(this); // Vinculo bidireccional  
    }  
  
    public String getNumero() {  
        return numero;  
    }  
  
    public String getFechaEmision() {  
        return fechaEmision;  
    }  
  
    public Foto getFoto() {  
        return foto;  
    }  
  
    public Titular getTitular() {  
        return titular;  
    }  
}
```

## 2) Análisis de relaciones:

Agregación: Celular —◇— Batería

Asociación bidireccional: Celular — Usuario



```

public class Bateria {
    private String modelo;
    private int capacidad; // en mAh

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

    public String getModelo() {
        return modelo;
    }

    public int getCapacidad() {
        return capacidad;
    }
}

```

```

public class Usuario {
    private String nombre;
    private String dni;
    private Celular celular; // Asociación bidireccional

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

    public void setCelular(Celular celular) {
        this.celular = celular;
    }

    public Celular getCelular() {
        return celular;
    }

    public String getNombre() {
        return nombre;
    }

    public String getDni() {
        return dni;
    }
}

```

```

public class Celular {
    private String imei;
    private String marca;
    private String modelo;
    private Bateria bateria; // Agregación
    private Usuario usuario; // Asociación bidireccional

    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); // vínculo bidireccional
    }

    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;
    }
}

```

## 3) Análisis de relaciones:

Asociación unidireccional: Libro → Autor

Agregación: Libro —◇— Editorial

Libro
-titulo: String -isbn: String -autor: Autor -editorial: Editorial
+Libro(String titulo, String isbn, Editorial editorial): ctor +getTitulo(): String +setTitulo(String titulo): void +getIsbn(): String +setIsbn(String isbn): void +getAutor(): Autor +setAutor(Autor autor): void +getEditorial(): Editorial +setEditorial(Editorial editorial): void

Main
+main(String[] args): void

Editorial
-nombre: String -direccion: String
+Editorial(String nombre, String direccion): ctor +getNombre(): String +setNombre(String nombre): void +getDireccion(): String +setDireccion(String direccion): void

Autor
-nombre: String -nacionalidad: String
+Autor(String nombre, String nacionalidad): ctor +getNombre(): String +setNombre(String nombre): void +getNacionalidad(): String +setNacionalidad(String nacionalidad): void

```

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;
    }
}

```

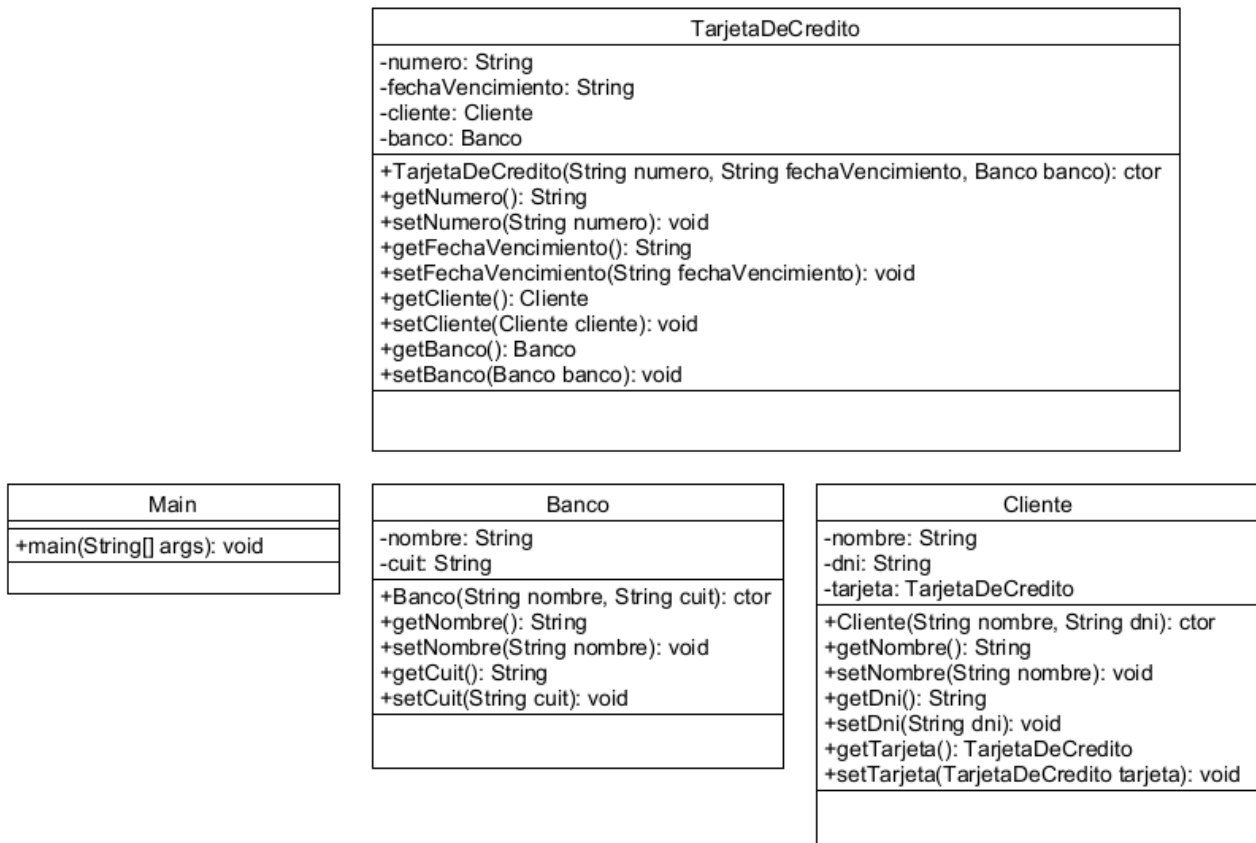
```
public class Editorial {  
    private String nombre;  
    private String direccion;  
  
    public Editorial(String nombre, String direccion) {  
        this.nombre = nombre;  
        this.direccion = direccion;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public String getDireccion() {  
        return direccion;  
    }  
}
```

```
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;  
    }  
}
```

## 4) Análisis de relaciones:

Asociación bidireccional: Tarjeta — Cliente

Agregación: Tarjeta — ◇ — Banco



```

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

    public TarjetaDeCredito getTarjeta() {
        return tarjeta;
    }
}

```

```

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; }
}

```

```

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

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

```

## 5)Análisis de relaciones

Composición: Computadora —◆— PlacaMadre

Asociación bidireccional: Computadora — Propietario

Main
+main(String[] args): void

Computadora
-marca: String -numeroSerie: String -placaMadre: PlacaMadre -propietario: Propietario
+Computadora(String marca, String numeroSerie, String modelo, String chipset): ctor +getMarca(): String +setMarca(String marca): void +getNumeroSerie(): String +setNumeroSerie(String numeroSerie): void +getPlacaMadre(): PlacaMadre +setPlacaMadre(PlacaMadre placaMadre): void +getPropietario(): Propietario +setPropietario(Propietario propietario): void

Propietario
-nombre: String -dni: String -computadora: Computadora
+Propietario(String nombre, String dni): ctor +getNombre(): String +setNombre(String nombre): void +getDni(): String +setDni(String dni): void +getComputadora(): Computadora +setComputadora(Computadora computadora): void

PlacaMadre
-modelo: String -chipset: String
+PlacaMadre(String modelo, String chipset): ctor +getModelo(): String +setModelo(String modelo): void +getChipset(): String +setChipset(String chipset): void



```

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

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

```

```

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 void setComputadora(Computadora computadora) {
        this.computadora = computadora;
    }
}

```

```

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

    public Computadora(String marca, String numeroSerie, PlacaMadre placaMadre, Propietario propietario) {
        this.marca = marca;
        this.numeroSerie = numeroSerie;
        this.placaMadre = placaMadre;
        this.propietario = propietario;
        propietario.setComputadora(this);
    }
}

```

## 6) Analisis de relaciones

Asociación unidireccional: Reserva → Cliente

Agregación: Reserva —◇— Mesa

Reserva		
-fecha: String -hora: String -cliente: Cliente -mesa: Mesa		
+Reserva(String fecha, String hora, Mesa mesa): ctor +getFecha(): String +setFecha(String fecha): void +getHora(): String +setHora(String hora): void +getCliente(): Cliente +setCliente(Cliente cliente): void +getMesa(): Mesa +setMesa(Mesa mesa): void		
Main	Mesa	Cliente
+main(String[] args): void	-numero: int -capacidad: int +Mesa(int numero, int capacidad): ctor +getNumero(): int +setNumero(int numero): void +getCapacidad(): int +setCapacidad(int capacidad): void	-nombre: String -telefono: String +Cliente(String nombre, String telefono): ctor +getNombre(): String +setNombre(String nombre): void +getTelefono(): String +setTelefono(String telefono): void

```

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

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

```

```

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

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

```

```

public class Reserva {
    private String fecha;
    private String hora;
    private ClienteReserva cliente;
    private Mesa mesa;

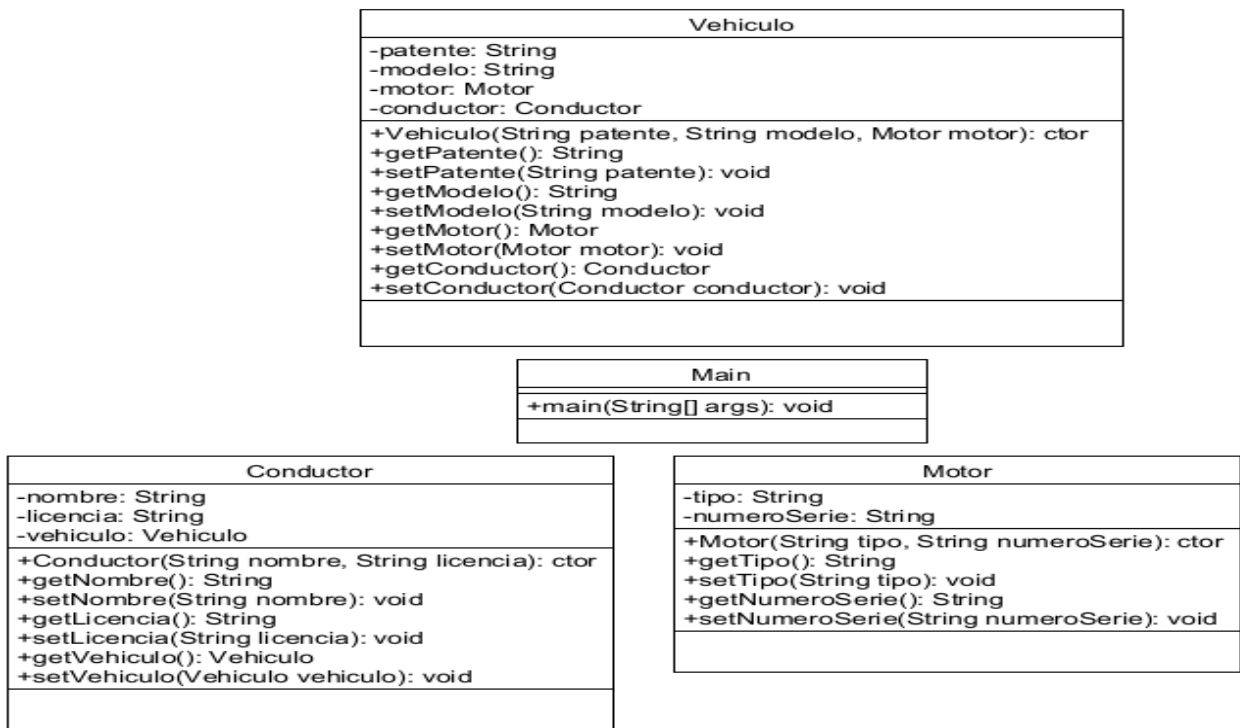
    public Reserva(String fecha, String hora, ClienteReserva cliente, Mesa mesa) {
        this.fecha = fecha;
        this.hora = hora;
        this.cliente = cliente;
        this.mesa = mesa;
    }
}

```

## 7) Analisis de relaciones:

Agregación: Vehículo —◇— Motor

Asociación bidireccional: Vehículo — Conducto



```

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

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

```

```

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

```

```

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

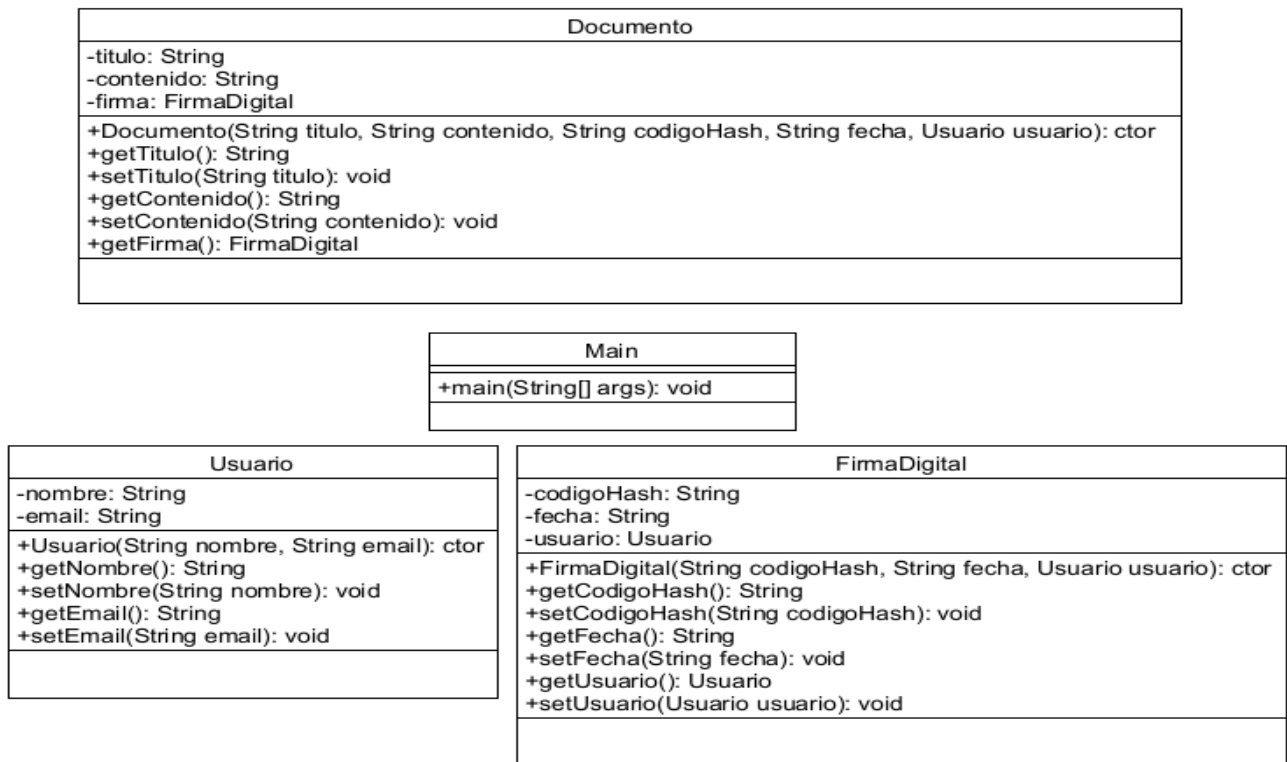
    public Vehiculo(String patente, String modelo, Motor motor, Conductor conductor) {
        this.patente = patente;
        this.modelo = modelo;
        this.motor = motor;
        this.conductor = conductor;
        conductor.setVehiculo(this);
    }
}

```

## 8) Análisis de relaciones:

Composición: Documento —◆— FirmaDigital

Agregación: FirmaDigital —◇— Usuario



```

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

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

```

```

public class FirmaDigital {
    private String codigoHash;
    private String fecha;
    private UsuarioDoc usuario;

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

```

```

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

    public Documento(String titulo, String contenido, FirmaDigital firma) {
        this.titulo = titulo;
        this.contenido = contenido;
        this.firma = firma;
    }
}

```

9) Asociación unidireccional: Cita → Paciente

Asociación unidireccional: Cita → Profesional

CitaMedica
-fecha: String -hora: String -paciente: Paciente -profesional: Profesional
+CitaMedica(String fecha, String hora): ctor +getFecha(): String +setFecha(String fecha): void +getHora(): String +setHora(String hora): void +getPaciente(): Paciente +setPaciente(Paciente paciente): void +getProfesional(): Profesional +setProfesional(Profesional profesional): void

Paciente
-nombre: String -obraSocial: String
+Paciente(String nombre, String obraSocial): ctor +getNombre(): String +setNombre(String nombre): void +getObraSocial(): String +setObraSocial(String obraSocial): void

Main
+main(String[] args): void

Profesional
-nombre: String -especialidad: String
+Profesional(String nombre, String especialidad): ctor +getNombre(): String +setNombre(String nombre): void +getEspecialidad(): String +setEspecialidad(String especialidad): void

```

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

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

```

```

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

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

```

```

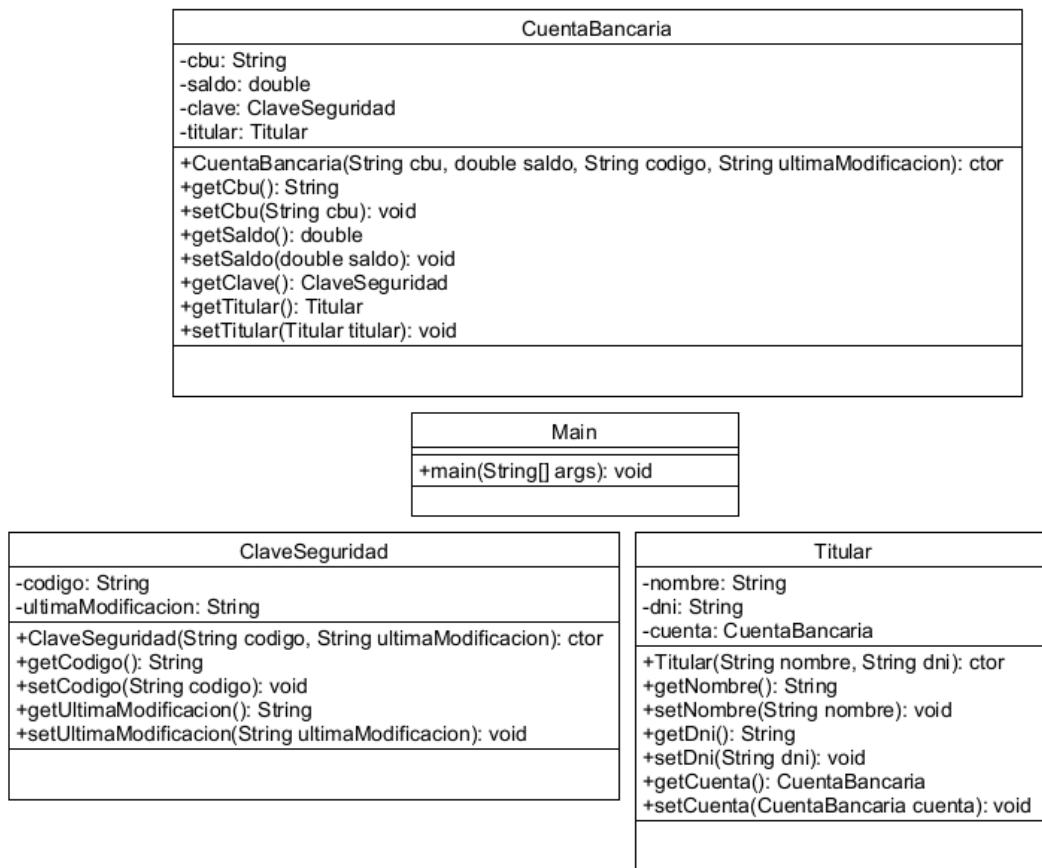
public class CitaMedica {
    private String fecha;
    private String hora;
    private Paciente paciente;
    private Profesional profesional;

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

```

## 10) Composición: Cuenta —◆— ClaveSeguridad

Asociación bidireccional: Cuenta — Titular



```

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

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

```

```

public class TitularCuenta {
    private String nombre;
    private String dni;
    private CuentaBancaria cuenta;

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

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

```

```

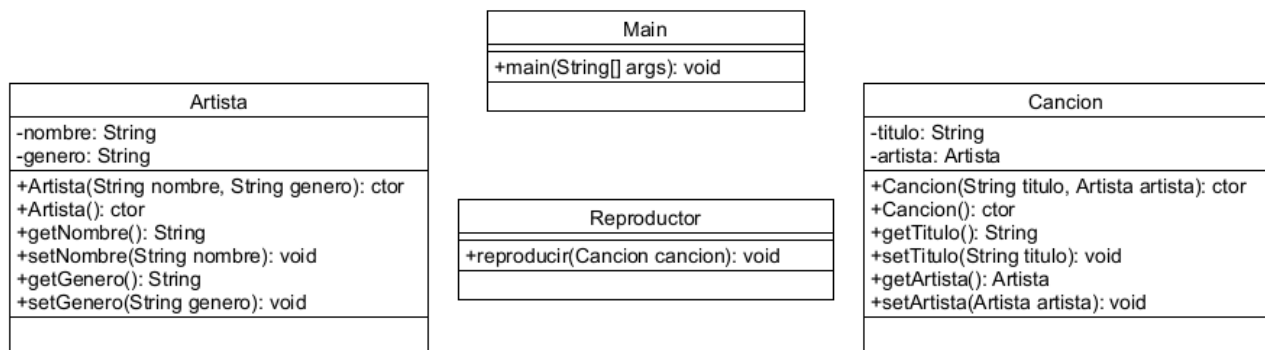
public class CuentaBancaria {
    private String cbu;
    private double saldo;
    private ClaveSeguridad clave;
    private TitularCuenta titular;

    public CuentaBancaria(String cbu, double saldo, ClaveSeguridad clave, TitularCuenta titular) {
        this.cbu = cbu;
        this.saldo = saldo;
        this.clave = clave;
        this.titular = titular;
        titular.setCuenta(this);
    }
}

```

## 11) Asociación unidireccional: Canción → Artista

Dependencia de uso: Reproductor ---> Canción



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

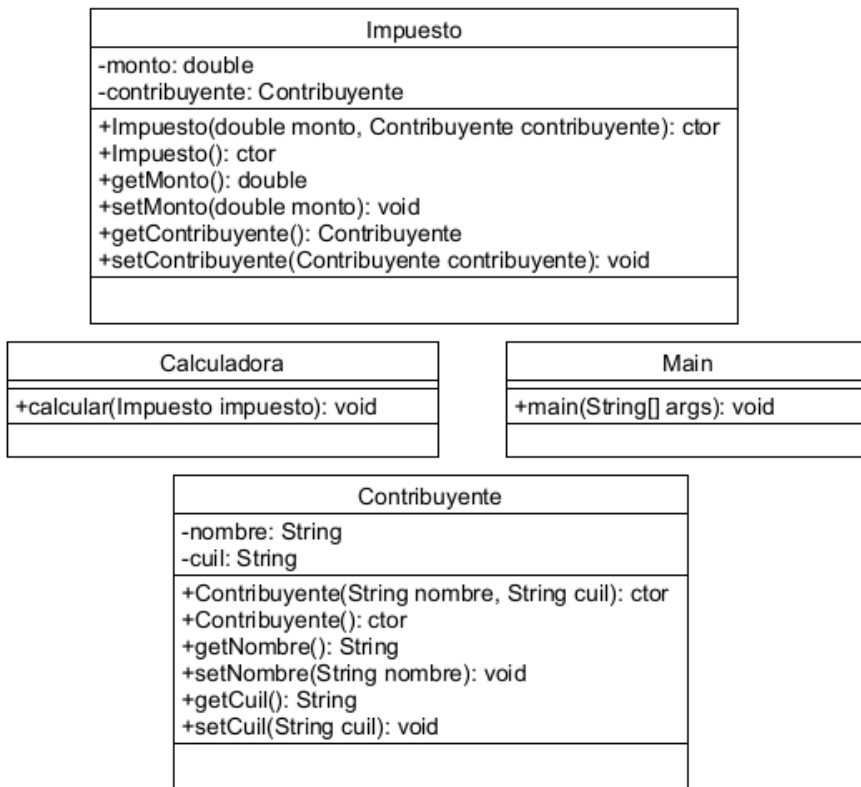
```
public class Cancion {  
    private String titulo;  
    private Artista artista;  
  
    public Cancion(String titulo, Artista artista) {  
        this.titulo = titulo;  
        this.artista = artista;  
    }  
  
    public String getTitulo() { return titulo; }  
    public Artista getArtista() { return artista; }  
}
```

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



## 12) Asociación unidireccional: Impuesto → Contribuyente

Dependencia de uso: Calculadora - - - &gt; Impuesto



```

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

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

```

```

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

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

    public double getMonto() { return monto; }
}

```

```
public class Calculadora {
    public void calcular(Impuesto impuesto) {
        System.out.println("Impuesto calculado: " + impuesto.getMonto());
    }
}
```

### 13) Asociación unidireccional: CódigoQR → Usuario

Dependencia de creación: GeneradorQR - - - > CódigoQR (<<create>>)

resolucion.tp5.Ejercicio13::Usuario
-nombre: String -email: String
+Usuario(): ctor +Usuario(String nombre, String email): ctor +getNombre(): String +setNombre(String nombre): void +getEmail(): String +setEmail(String email): void

resolucion.tp5.Ejercicio13::Main
+main(String[] args): void

resolucion.tp5.Ejercicio13::GeneradorQR
+generar(String valor, Usuario usuario): void

resolucion.tp5.Ejercicio13::CodigoQR
-valor: String -usuario: Usuario
+CodigoQR(): ctor +CodigoQR(String valor, Usuario usuario): ctor +getValor(): String +setValor(String valor): void +getUsuario(): Usuario +setUsuario(Usuario usuario): void

```
public class UsuarioQR {
    private String nombre;
    private String email;

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

```
public class CodigoQR {
    private String valor;
    private UsuarioQR usuario;

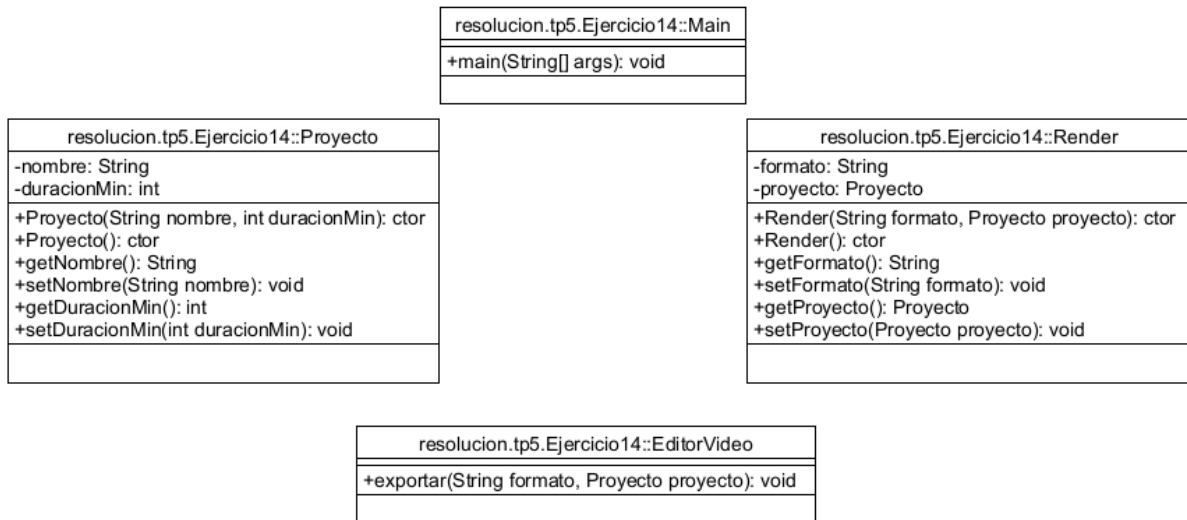
    public CodigoQR(String valor, UsuarioQR usuario) {
        this.valor = valor;
        this.usuario = usuario;
    }
}
```

```
public class GeneradorQR {
    public void generar(String valor, UsuarioQR usuario) {
        CodigoQR codigo = new CodigoQR(valor, usuario);
        System.out.println("QR generado para " + usuario.nombre + ": " + valor);
    }
}
```

14)

Asociación unidireccional: Render → Proyecto

Dependencia de creación: EditorVideo - - -&gt; Render (&lt;&lt;create&gt;&gt;)



```

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

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

```

```

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

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

```

```

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

```