

PROGRAMACIÓN II

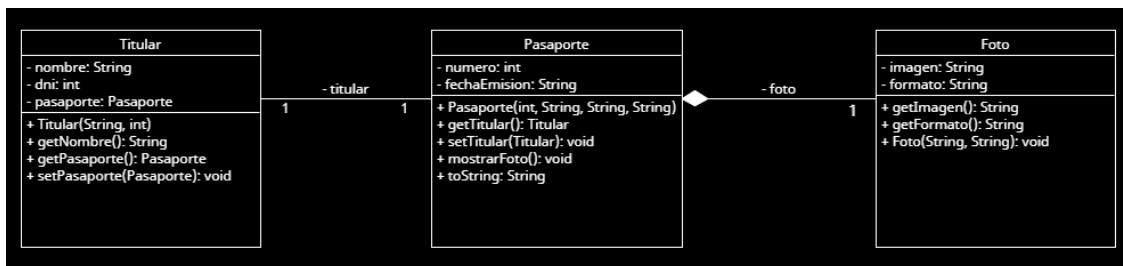
Trabajo Práctico 5: Relaciones UML 1 a 1

Alumno: Gonzalo Barrios.

GitHub: <https://github.com/Goonza88/Programacion-II>

1.

UML:



Clase Pasaporte:

```
public class Pasaporte {
    private int numero;
    private String fechaEmision;
    private Foto foto;
    private Titular titular;

    public Pasaporte(int numero, String fechaEmision, String imagen, String formato) {
        this.numero = numero;
        this.fechaEmision = fechaEmision;
        this.foto = new Foto(imagen, formato);
    }

    public Titular getTitular() {
        return titular;
    }

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

    public void mostrarFoto() {
        System.out.println("Imagen: " + foto.getImagen() + "\nFormato: " + foto.getFormato());
    }

    @Override
    public String toString() {
        return "Pasaporte{" + "Numero=" + numero + ", Fecha de Emision=" + fechaEmision + ", Titular=" + titular.getNombre() + "}";
    }
}
```

Clase Foto:

```
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;  
    }  
}
```

Clase Titular:

```
public class Titular {  
    private String nombre;  
    private int dni;  
    private Pasaporte pasaporte;  
  
    public Titular(String nombre, int dni) {  
        this.nombre = nombre;  
        this.dni = dni;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public Pasaporte getPasaporte() {  
        return pasaporte;  
    }  
  
    public void setPasaporte(Pasaporte pasaporte) {  
        this.pasaporte = pasaporte;  
        if (pasaporte != null && pasaporte.getTitular() != this) {  
            pasaporte.setTitular(this);  
        }  
    }  
}
```

Main y Resultados:

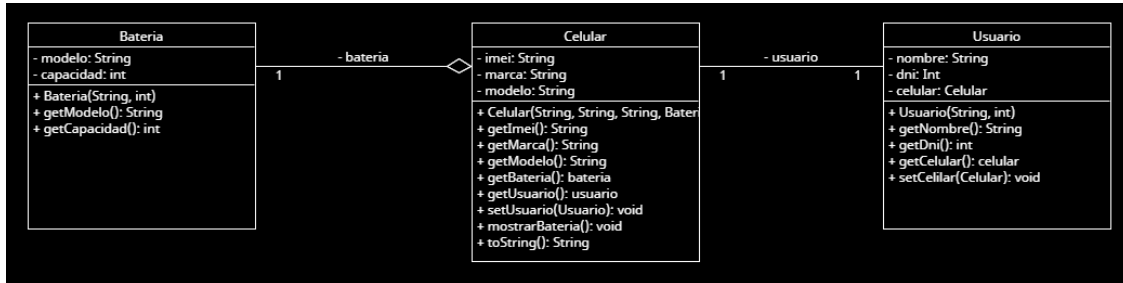
```
public class UMLBasico1 {  
    public static void main(String[] args) {  
        Titular titular = new Titular("Gonzalo Barrios", 43393592);  
        Pasaporte pasaporte = new Pasaporte(12345678, "04/04/2001", "imagen123", "PNG");  
        titular.setPasaporte(pasaporte);  
  
        System.out.println(pasaporte);  
        pasaporte.mostrarFoto();  
    }  
}
```

- UMLBasico1 (run) x

run:
Pasaporte(Numero=12345678, Fecha de Emision=04/04/2001, Titular=Gonzalo Barrios)
Imagen: imagen123
Formato: PNG
BUILD SUCCESSFUL (total time: 0 seconds)

2.

UML:



Clase Celular:

```

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) {
        this.imei = imei;
        this.marca = marca;
        this.modelo = modelo;
        this.bateria = bateria;
    }

    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;
    }

    public void setUsuario(Usuario usuario) {
        this.usuario = usuario;
        if (usuario != null && usuario.getCelular() != this) {
            usuario.setCelular(this);
        }
    }

    public void mostrarBateria() {
        System.out.println("Celular: " + marca + ", Bateria: " + bateria.getModelo() + ", Capacidad: " + bateria.getCapacidad() + "MHz.");
    }

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

```

Clase Batería:

```
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;
    }
}
```

Clase Usuario:

```
public class Usuario {
    private String nombre;
    private int dni;
    private Celular celular;

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

    public String getNombre() {
        return nombre;
    }

    public int getDni() {
        return dni;
    }

    public Celular getCelular() {
        return celular;
    }

    public void setCelular(Celular celular) {
        this.celular = celular;
        if (celular != null && celular.getUsuario() != this) {
            celular.setUsuario(this);
        }
    }
}
```

Main y Resultados:

```
public class UMLBasico2 {
    public static void main(String[] args) {
        Bateria B = new Bateria("Bateria12+", 5000);
        Usuario U = new Usuario("Gonzalo Barrios", 43393592);
        Celular C = new Celular("abc123def456ghi", "Motorola", "G88", B);
        U.setCelular(C);

        System.out.println(C);
        C.mostrarBateria();
    }
}
```

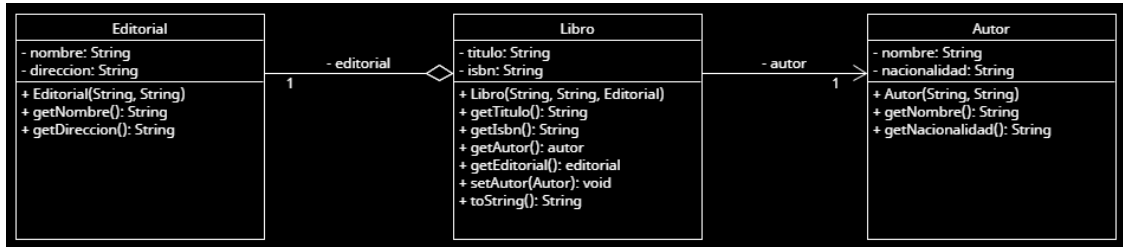
UMLBasico2 (run) X

Run:

Celular[imei=abc123def456ghi, marca=Motorola, modelo=G88, usuario=Gonzalo Barrios, dni=43393592]
Celular: Motorola, Bateria: Bateria12+, Capacidad: 5000mAh.
WUJID SUCCESSFUL (total time: 0 seconds)

3:

UML:



Clase Libro:

```

public class Libro {
    private String titulo;
    private String isbn;
    private Autor autor;
    private Editorial editorial;

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

    public String getTitulo() {
        return titulo;
    }

    public String getIsbn() {
        return isbn;
    }

    public Autor getAutor() {
        return autor;
    }

    public Editorial getEditorial() {
        return editorial;
    }

    public void setAutor(Autor autor) {
        this.autor = autor;
    }

    @Override
    public String toString() {
        return "Libro{" + "titulo=" + titulo + ", isbn=" + isbn + ", autor=" + autor.getNombre() +
            ", nacionalidad=" + autor.getNacionalidad() + ", editorial=" + editorial.getNombre() +
            ", direccion=" + editorial.getDireccion() + "}";
    }
}

```

Clase Autor:

```
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;  
    }  
}
```

Clase 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;  
    }  
}
```

Main y Resultados:

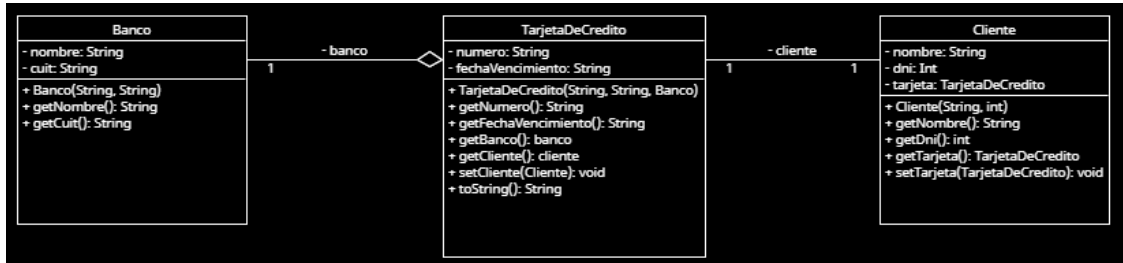
```
public class UMLBasico3 {  
    public static void main(String[] args) {  
        Autor A = new Autor("Jose Perez", "Mexico");  
        Editorial E = new Editorial("La Gran Editorial", "Ave, 9 de Julio 3871");  
        Libro L = new Libro("Libro sobre Nada", "032176572X", E);  
        L.setAutor(A);  
  
        System.out.println(L);  
    }  
}
```

- UML Basico (run) X

run:
Libro[titulo=Libro sobre Nada, isbn=032176572X, autor=Jose Perez, nacionalidad=Mexico, editorial=La Gran Editorial, direccion=Ave, 9 de Julio 3871]
BUILD SUCCESSFUL (total time: 0 seconds)

4.

UML:



Clase TarjetaDeCredito:

```

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

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

    public String getNumero() {
        return numero;
    }

    public String getFechaVencimiento() {
        return fechaVencimiento;
    }

    public Banco getBanco() {
        return banco;
    }

    public Banco getBanco() {
        return banco;
    }

    public Cliente getCliente() {
        return cliente;
    }

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

    @Override
    public String toString() {
        return "TarjetaDeCredito{" + "numero=" + numero + ", fechaVencimiento=" + fechaVencimiento +
            ", cliente=" + cliente.getNombre() + ", dni=" + cliente.getDni() +
            ", banco=" + banco.getNombre() + ", cuil=" + banco.getCuil() + "}";
    }
}

```

Clase Cliente:

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

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

    public String getNombre() {
        return nombre;
    }

    public int getDni() {
        return dni;
    }

    public TarjetaDeCredito getTarjeta() {
        return tarjeta;
    }

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

Clase Banco:

```
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;
    }
}
```

Main y Resultados:

```
public class UMLBasico4 {
    public static void main(String[] args) {
        Banco B = new Banco("Gran Banco", "30-12345678-9");
        TarjetaDeCredito T = new TarjetaDeCredito("1234-5678-9876-5432", "05/07/2030", B);
        Cliente C = new Cliente("Gonzalo Barrios", 43393592);
        T.setCliente(C);

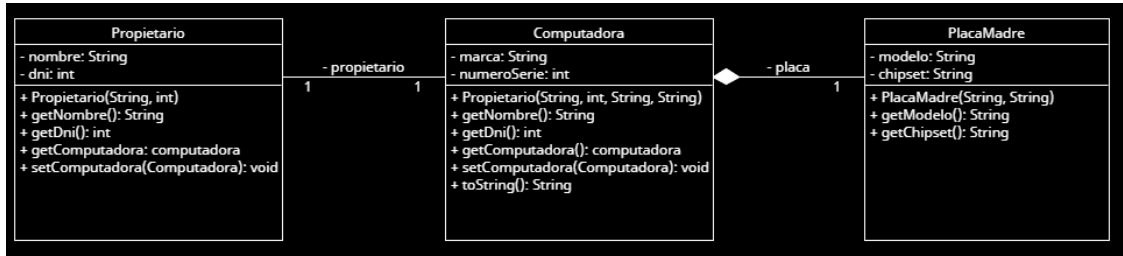
        System.out.println(T);
    }
}
```

- UML Basico (run) X

run:
TarjetaDeCredito[numero=1234-5678-9876-5432, fechaVencimiento=05/07/2030, cliente=Gonzalo Barrios, dni=43393592, banco=Gran Banco, cuit30-12345678-9]
BUILD SUCCESSFUL (total time: 0 seconds)

5.

UML:



Clase Computadora:

```

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

    public Computadora(String marca, int numeroSerie, String modelo, String chipset) {
        this.marca = marca;
        this.numeroSerie = numeroSerie;
        this.placa = new PlacaMadre(modelo, chipset);
    }

    public String getMarca() {
        return marca;
    }

    public int getNumeroSerie() {
        return numeroSerie;
    }

    public Propietario getPropietario() {
        return propietario;
    }

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

    @Override
    public String toString() {
        return "Computadora{" + "marca=" + marca + ", numeroSerie=" + numeroSerie +
            ", placa=" + placa.getModelo() + ", chipset=" + placa.getChipset() +
            ", propietario=" + propietario.getNombre() + ", dni" + propietario.getDni() + "}";
    }
}
    
```

Clase Propietario:

```
public class Propietario {
    private String nombre;
    private int dni;
    private Computadora computadora;

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

    public String getNombre() {
        return nombre;
    }

    public int getDni() {
        return dni;
    }

    public Computadora getComputadora() {
        return computadora;
    }

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

Clase PlacaMadre:

```
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;
    }
}
```

Main y Resultados:

```
public class UMLBasico5 {
    public static void main(String[] args) {
        Propietario P = new Propietario("Gonzalo Barrios", 43393592);
        Computadora C = new Computadora("Lenovo", 35012, "MSI MB 350", "AM3");
        C.setPropietario(P);

        System.out.println(C);
    }
}
```

UMLBasico5.UMLBasico5 > main >

UML Basico (run) x

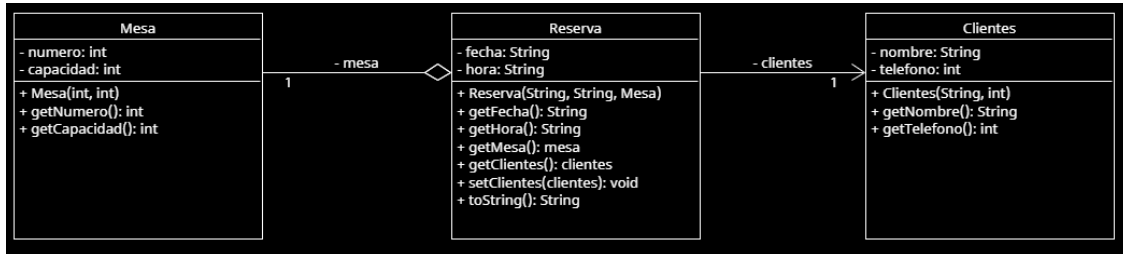
run:

Computadora{marca=Lenovo, numeroSerie=35012, placa=MSI MB 350, chipset=AM3, propietario=Gonzalo Barrios, dni43393592}

BUILD SUCCESSFUL (total time: 0 seconds)

6.

UML:



Clase Reserva:

```

public class Reserva {
    private String fecha;
    private String hora;
    private Clientes clientes;
    private Mesa mesa;

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

    public String getFecha() {
        return fecha;
    }

    public String getHora() {
        return hora;
    }

    public Mesa getMesa() {
        return mesa;
    }

    public Clientes getCientes() {
        return clientes;
    }

    public void setCientes(Clientes clientes) {
        this.clientes = clientes;
    }

    @Override
    public String toString() {
        return "Reserva{" + "fecha=" + fecha + ", hora=" + hora +
            ", clientes=" + clientes.getNombre() + ", telefono=" + clientes.getTelefono() +
            ", mesa=" + mesa.getNumero() + ", capacidad=" + mesa.getCapacidad() + "}";
    }
}
    
```

Clase Clientes (Esta en plural para diferenciarla de la Clase Cliente anterior):

```
public class Clientes {  
    private String nombre;  
    private int telefono;  
  
    public Clientes(String nombre, int telefono) {  
        this.nombre = nombre;  
        this.telefono = telefono;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public int getTelefono() {  
        return telefono;  
    }  
}
```

Clase 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;  
    }  
}
```

Main y Resultados:

```
public class UMLBasico6 {  
    public static void main(String[] args) {  
        Clientes C = new Clientes("Gonzalo Barrios", 15545243);  
        Mesa M = new Mesa(8, 3);  
        Reserva R = new Reserva("04/04/2026", "20:30", M);  
        R.setClientes(C);  
  
        System.out.println(R);  
    }  
}
```

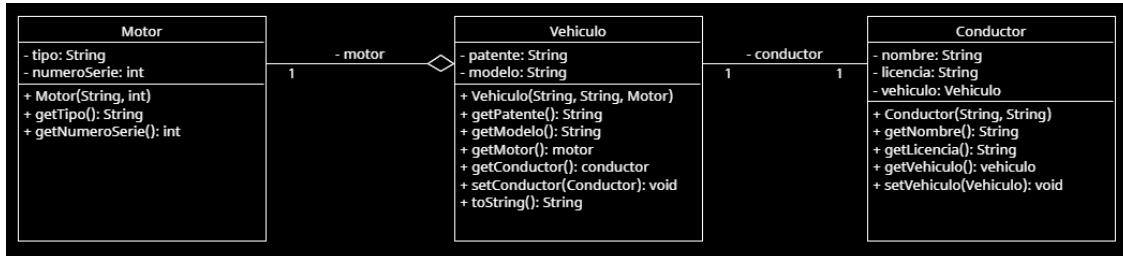
hl.basico.UMLBasico6 > main >

- UML Basico (run) x

run:
Reserva{fecha=04/04/2026, hora=20:30, clientes=Gonzalo Barrios, telefono=15545243, mesa=8, capacidad=3}
BUILD SUCCESSFUL (total time: 0 seconds)

7.

UML:



Clase Vehiculo:

```

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

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

    public String getPatente() {
        return patente;
    }

    public String getModelo() {
        return modelo;
    }

    public Motor getMotor() {
        return motor;
    }

    public Conductor getConductor() {
        return conductor;
    }

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

    @Override
    public String toString() {
        return "Vehiculo{" + "patente=" + patente + ", modelo=" + modelo +
            ", motor=" + motor.getTipo() + ", numeroSerie=" + motor.getNumeroSerie() +
            ", conductor=" + conductor.getNombre() + ", licencia=" + conductor.getLicencia() + "}";
    }
}

```

Clase Motor:

```
public class Motor {  
    private String tipo;  
    private int numeroSerie;  
  
    public Motor(String tipo, int numeroSerie) {  
        this.tipo = tipo;  
        this.numeroSerie = numeroSerie;  
    }  
  
    public String getTipo() {  
        return tipo;  
    }  
  
    public int getNumeroSerie() {  
        return numeroSerie;  
    }  
}
```

Clase Conductor:

```
public class Conductor {  
    private String nombre;  
    private String licencia;  
    private Vehiculo vehiculo;  
  
    public Conductor(String nombre, String licencia) {  
        this.nombre = nombre;  
        this.licencia = licencia;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public String getLicencia() {  
        return licencia;  
    }  
  
    public Vehiculo getVehiculo() {  
        return vehiculo;  
    }  
  
    public void setVehiculo(Vehiculo vehiculo) {  
        this.vehiculo = vehiculo;  
        if (vehiculo != null && vehiculo.getConductor() != this) {  
            vehiculo.setConductor(this);  
        }  
    }  
}
```

Main y Resultados:

```
public class UMLBasico7 {  
    public static void main(String[] args) {  
        Motor M = new Motor("v10", 4628);  
        Conductor C = new Conductor("Gonzalo Barrios", "A");  
        Vehiculo V = new Vehiculo("ABC123", "Corolla", M);  
        V.setConductor(C);  
  
        System.out.println(V);  
    }  
}
```

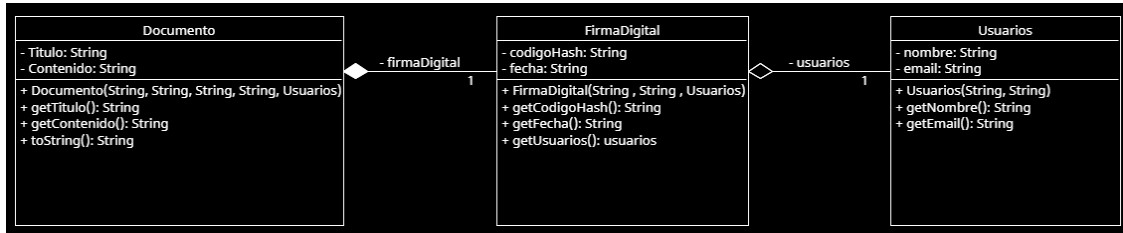
UMLBasico7 > main >

- UML Basico (run) x

run:
Vehiculo(patente=ABC123, modelo=Corolla, motor=v10, numeroSerie=4628, conductor=Gonzalo Barrios, licencia=A)
BUILD SUCCESSFUL (total time: 0 seconds)

8.

UML:



Clase Documento:

```

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

    public Documento(String titulo, String contenido, String codigoHash, String fecha, Usuarios usuarios) {
        this.titulo = titulo;
        this.contenido = contenido;
        this.firmaDigital = new FirmaDigital(codigoHash, fecha, usuarios);
    }

    public String getTitulo() {
        return titulo;
    }

    public String getContenido() {
        return contenido;
    }

    @Override
    public String toString() {
        return "Documento(" + "titulo=" + titulo + ", contenido=" + contenido +
            ", firmaDigital=" + firmaDigital.getCodigoHash() + ", fecha=" + firmaDigital.getFecha() +
            ", usuarios=" + firmaDigital.getUsuarios().getNombre() + ", email=" + firmaDigital.getUsuarios().getEmail() + ")";
    }
}

```

Clase FirmaDigital:

```

public class FirmaDigital {
    private String codigoHash;
    private String fecha;
    private Usuarios usuarios;

    public FirmaDigital(String codigoHash, String fecha, Usuarios usuarios) {
        this.codigoHash = codigoHash;
        this.fecha = fecha;
        this.usuarios = usuarios;
    }

    public String getCodigoHash() {
        return codigoHash;
    }

    public String getFecha() {
        return fecha;
    }

    public Usuarios getUsuarios() {
        return usuarios;
    }
}

```

Clase Usuarios (Esta en plural para diferenciarla de la Clase Usuario anterior):

```
public class Usuarios {  
    private String nombre;  
    private String email;  
  
    public Usuarios(String nombre, String email) {  
        this.nombre = nombre;  
        this.email = email;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public String getEmail() {  
        return email;  
    }  
}
```

Main y Resultados:

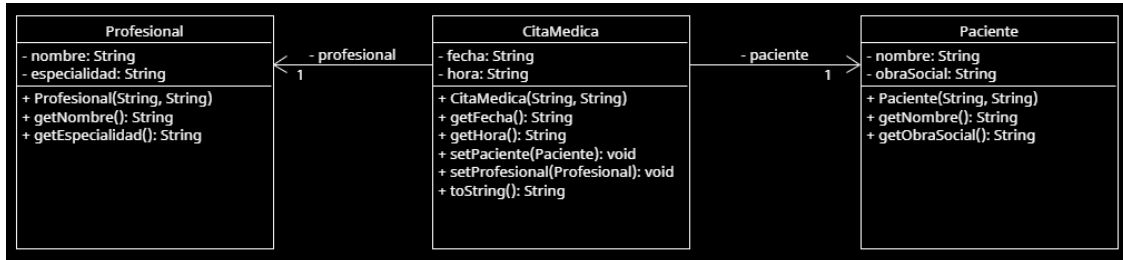
```
public class UMLBasico8 {  
    public static void main(String[] args) {  
        Usuarios U = new Usuarios("Gonzalo Barrios", "gonzabarrios@gmail.com");  
        Documento D = new Documento("Planos: Planta Nuclear", "Privado", "FalsoCodigoHash", "05/11/2025", U);  
  
        System.out.println(D);  
    }  
}
```

- UML Basico (run) x

run:
Documento{titulo=Planos: Planta Nuclear, contenido=Privado, firmaDigital=FalsoCodigoHash, fecha=05/11/2025, usuarios=Gonzalo Barrios, email=gonzabarrios@gmail.com}
BUILD SUCCESSFUL (total time: 0 seconds)

9.

UML:



Clase CitaMedica:

```

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

    public CitaMedica(String fecha, String hora) {
        this.fecha = fecha;
        this.hora = hora;
    }

    public String getFecha() {
        return fecha;
    }

    public String getHora() {
        return hora;
    }

    public void setPaciente(Paciente paciente) {
        this.paciente = paciente;
    }

    public void setProfesional(Profesional profesional) {
        this.profesional = profesional;
    }

    @Override
    public String toString() {
        return "CitaMedica{" + "fecha=" + fecha + ", hora=" + hora +
            ", paciente=" + paciente.getNombre() + ", obraSocial=" + paciente.getObraSocial() +
            ", profesional=" + profesional.getNombre() + ", especialidad=" + profesional.getEspecialidad() + "}";
    }
}
    
```

Clase Paciente:

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

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

    public String getNombre() {
        return nombre;
    }

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

Clase Profesional:

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

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

    public String getNombre() {
        return nombre;
    }

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

Main y Resultados:

```
public class UMLBasico9 {
    public static void main(String[] args) {
        Paciente Pa = new Paciente ("Gonzalo Barrios", "Empleados de Comercio");
        Profesional Pr = new Profesional ("Rene Favaloro", "Cardiologo");
        CitaMedica C = new CitaMedica("16/08/2025", "16:00");
        C.setPaciente(Pa);
        C.setProfesional(Pr);

        System.out.println(C);
    }
}
```

UMLBasico9 > main > C >

UML Basico (run) X

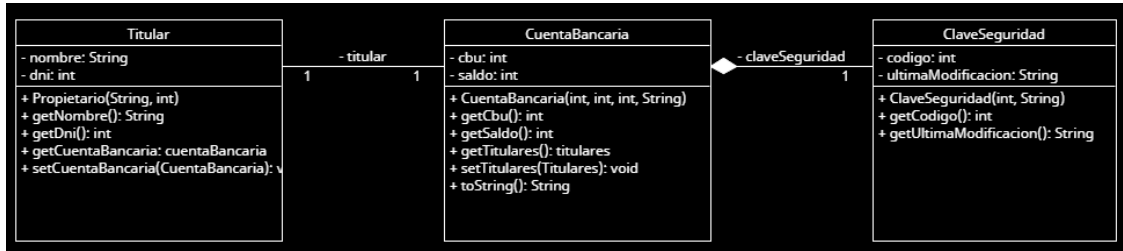
run:

CitaMedica[fecha=16/08/2025, hora=16:00, paciente=Gonzalo Barrios, obraSocial=Empleados de Comercio, profesional=Rene Favaloro, especialidad=Cardiologo]

BUILD SUCCESSFUL (total time: 0 seconds)

10.

UML:



Clase CuentaBancaria:

```

public class CuentaBancaria {
    private int cbu;
    private int saldo;
    private ClaveSeguridad claveSeguridad;
    private Titulares titulares;

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

    public int getCbu() {
        return cbu;
    }

    public int getSaldo() {
        return saldo;
    }

    public Titulares getTitulares() {
        return titulares;
    }

    public void setTitulares(Titulares titulares) {
        this.titulares = titulares;
        if (titulares != null && titulares.getCuentaBancaria() != this) {
            titulares.setCuentaBancaria(this);
        }
    }

    @Override
    public String toString() {
        return "CuentaBancaria{" + "cbu=" + cbu + ", saldo=" + saldo +
            ", claveSeguridad=" + claveSeguridad.getCodigo() + ", ultimaModificacion=" + claveSeguridad.getUltimaModificacion() +
            ", titulares=" + titulares.getNombre() + ", dni=" + titulares.getDni() + "}";
    }
}

```

Clase ClaveSeguridad:

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

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

    public int getCodigo() {
        return codigo;
    }

    public String getUltimaModificacion() {
        return ultimaModificacion;
    }
}
```

Clase Titulares:

```
public class Titulares {
    private String nombre;
    private int dni;
    private CuentaBancaria cuentaBancaria;

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

    public String getNombre() {
        return nombre;
    }

    public int getDni() {
        return dni;
    }

    public CuentaBancaria getCuentaBancaria() {
        return cuentaBancaria;
    }

    public void setCuentaBancaria(CuentaBancaria cuentaBancaria) {
        this.cuentaBancaria = cuentaBancaria;
        if (cuentaBancaria != null && cuentaBancaria.getTitulares() != this) {
            cuentaBancaria.setTitulares(this);
        }
    }
}
```

Main y Resultados:

```
public class UMLBasico10 {
    public static void main(String[] args) {
        Titulares T = new Titulares("Gonzalo Barrios", 43393592);
        CuentaBancaria C = new CuentaBancaria(481837541, 350000, 7412, "15/05/2025");
        C.setTitulares(T);
        System.out.println(C);
    }
}
```

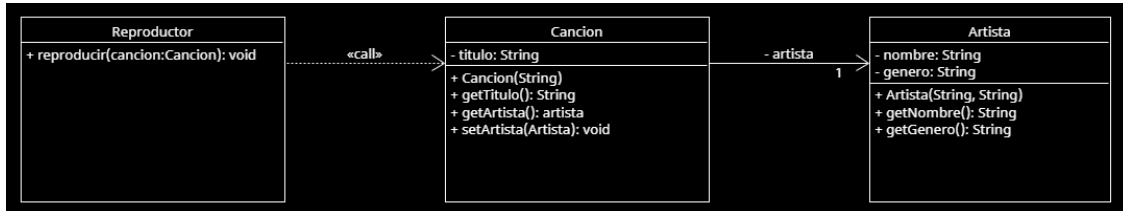
UMLBasico10 > main >

- UML Basico (run) x

run:
CuentaBancaria(cbu=481837541, saldo=350000, claveSeguridad=7412, ultimaModificacion=15/05/2025, titulares=Gonzalo Barrios, dni=43393592)
BUILD SUCCESSFUL (total time: 0 seconds)

11.

UML:



Clase Reproductor:

```
public class Reproductor {
    public void reproducir(Cancion cancion) {
        System.out.println("Reproduciendo: " + cancion.getTitulo() +
            ", del artista: " + cancion.getArtista().getNombre() +
            ", del genero: " + cancion.getArtista().getGenero());
    }
}
```

Clase Cancion:

```
public class Cancion {
    private String titulo;
    private Artista artista;

    public Cancion(String titulo) {
        this.titulo = titulo;
    }

    public String getTitulo() {
        return titulo;
    }

    public Artista getArtista() {
        return artista;
    }

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

Clase Artista:

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

Main y Resultados:

```
public class UMLBasico11 {  
    public static void main(String[] args) {  
        Artista A = new Artista("Duki", "Trap");  
        Cancion C = new Cancion("Givenchi");  
        Reproductor R = new Reproductor();  
        C.setArtista(A);  
  
        R.reproducir(C);  
    }  
}
```

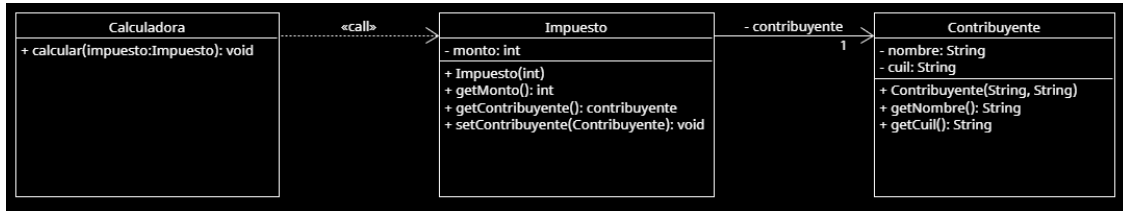
nl.basico.UMLBasico11 > main >

- UML Basico (run) x

run:
Reproduciendo: Givenchi, del artista: Duki, del genero: Trap
BUILD SUCCESSFUL (total time: 0 seconds)

12.

UML:



Clase Calculadora:

```

public class Calculadora {
    public void calcular(Impuesto impuesto) {
        System.out.println("El contribuyente " + impuesto.getContribuyente().getNombre() +
            "(cuil " + impuesto.getContribuyente().getCuil() +
            ") tiene que pagar $" + impuesto.getMonto());
    }
}
    
```

Clase Impuesto:

```

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

    public Impuesto(int monto) {
        this.monto = monto;
    }

    public int getMonto() {
        return monto;
    }

    public Contribuyente getContribuyente() {
        return contribuyente;
    }

    public void setContribuyente(Contribuyente contribuyente) {
        this.contribuyente = contribuyente;
    }
}
    
```

Clase Contribuyente:

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

Main y Resultados:

```
public class UMLBasico12 {  
    public static void main(String[] args) {  
        Contribuyente C = new Contribuyente("Gonzalo Barrios", "2-43393592-7");  
        Impuesto I = new Impuesto(250000);  
        Calculadora Calc = new Calculadora();  
        I.setContribuyente(C);  
  
        Calc.calcular(I);  
    }  
}
```

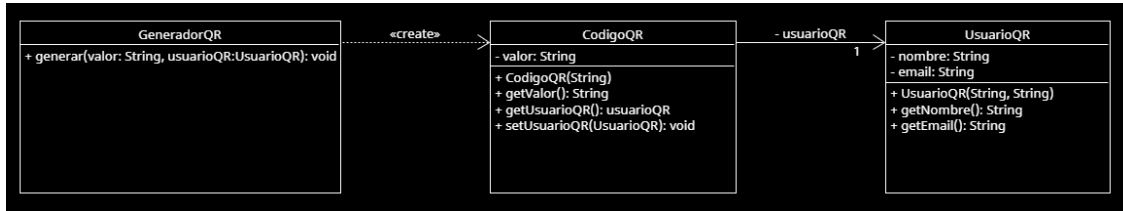
UMLBasico12 > main >

- UML Basico (run) x

run:
El contribuyente Gonzalo Barrios(cuil 2-43393592-7) tiene que pagar \$250000
BUILD SUCCESSFUL (total time: 0 seconds)

13.

UML:



Clase GeneradorQR:

```
public class GeneradorQR {
    public void generar(String valor, UsuarioQR usuarioQR) {
        CodigoQR C = new CodigoQR(valor);
        C.setUsuarioQR(usuarioQR);

        System.out.println("El usuario " + C.getUsuarioQR().getNombre() +
            " (email: " + C.getUsuarioQR().getEmail() +
            ") tiene el siguiente QR: " + C.getValor());
    }
}
```

Clase CodigoQR:

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

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

    public String getValor() {
        return valor;
    }

    public UsuarioQR getUsuarioQR() {
        return usuarioQR;
    }

    public void setUsuarioQR(UsuarioQR usuarioQR) {
        this.usuarioQR = usuarioQR;
    }
}
```

Clase UsuarioQR(Esta con QR para diferenciarla de las Clases Cliente anteriores):

```
public class UsuarioQR {  
    private String nombre;  
    private String email;  
  
    public UsuarioQR(String nombre, String email) {  
        this.nombre = nombre;  
        this.email = email;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public String getEmail() {  
        return email;  
    }  
}
```

Main y Resultados:

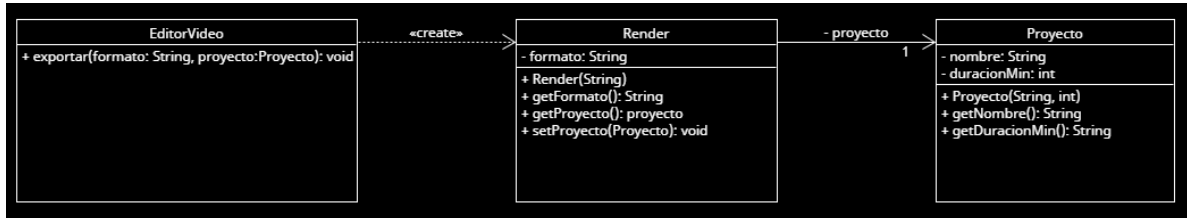
```
public class UMLBasico13 {  
    public static void main(String[] args) {  
        UsuarioQR U = new UsuarioQR("Gonzalo Barrios", "gonzabarrios@gmail.com");  
        GeneradorQR G = new GeneradorQR();  
        G.generar("FX3451s8fas8", U);  
    }  
}
```

UML Basico (run) x

run:
El usuario Gonzalo Barrios (email: gonzabarrios@gmail.com) tiene el siguiente QR: FX3451s8fas8
BUILD SUCCESSFUL (total time: 0 seconds)

14.

UML:



Clase EditorVideo:

```
public class EditorVideo {
    public void exportar(String formato, Proyecto proyecto) {
        Render R = new Render(formato);
        R.setProyecto(proyecto);

        System.out.println("El video del proyecto: " + R.getProyecto().getNombre() +
            " de " + R.getProyecto().getDuracionMin() +
            " minutos de duracion se exporto en formato " + R.getFormato());
    }
}
```

Clase Render:

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

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

    public String getFormato() {
        return formato;
    }

    public Proyecto getProyecto() {
        return proyecto;
    }

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

Clase Proyecto:

```
public class Proyecto {  
    private String nombre;  
    private int duracionMin;  
  
    public Proyecto(String nombre, int duracionMin) {  
        this.nombre = nombre;  
        this.duracionMin = duracionMin;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public int getDuracionMin() {  
        return duracionMin;  
    }  
}
```

Main y Resultados:

```
public class UMLBasico14 {  
    public static void main(String[] args) {  
        Proyecto P = new Proyecto("Malvinas", 125);  
        EditorVideo E = new EditorVideo();  
        E.exportar("MP4", P);  
    }  
}
```

- UML Basico (run) ×

run:
El video del proyecto: 'Malvinas' de 125 minutos de duracion se exporto en formato MP4
BUILD SUCCESSFUL (total time: 0 seconds)