

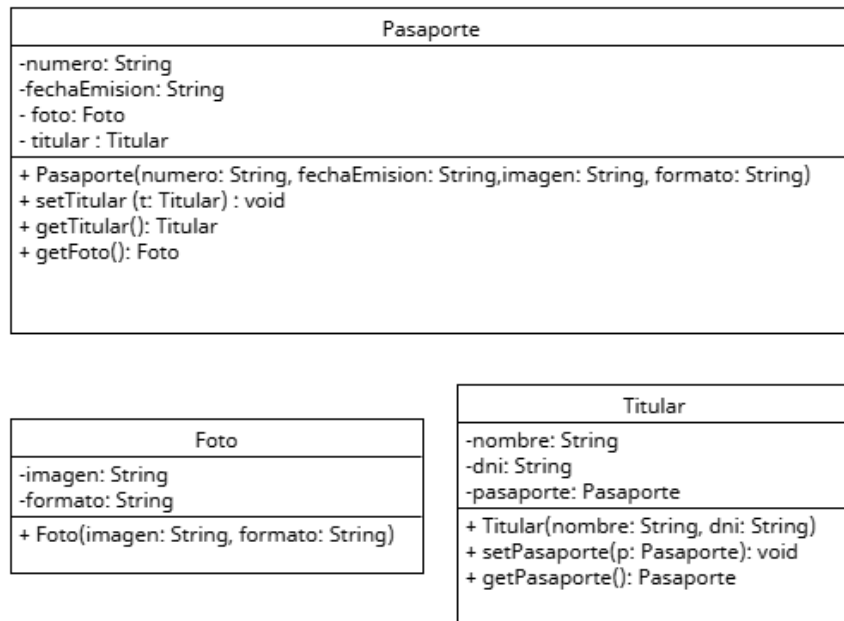
PROGRAMACIÓN II

TRABAJO PRÁCTICO 5: UML

Alumna: Garcia Galfione, Fiorella

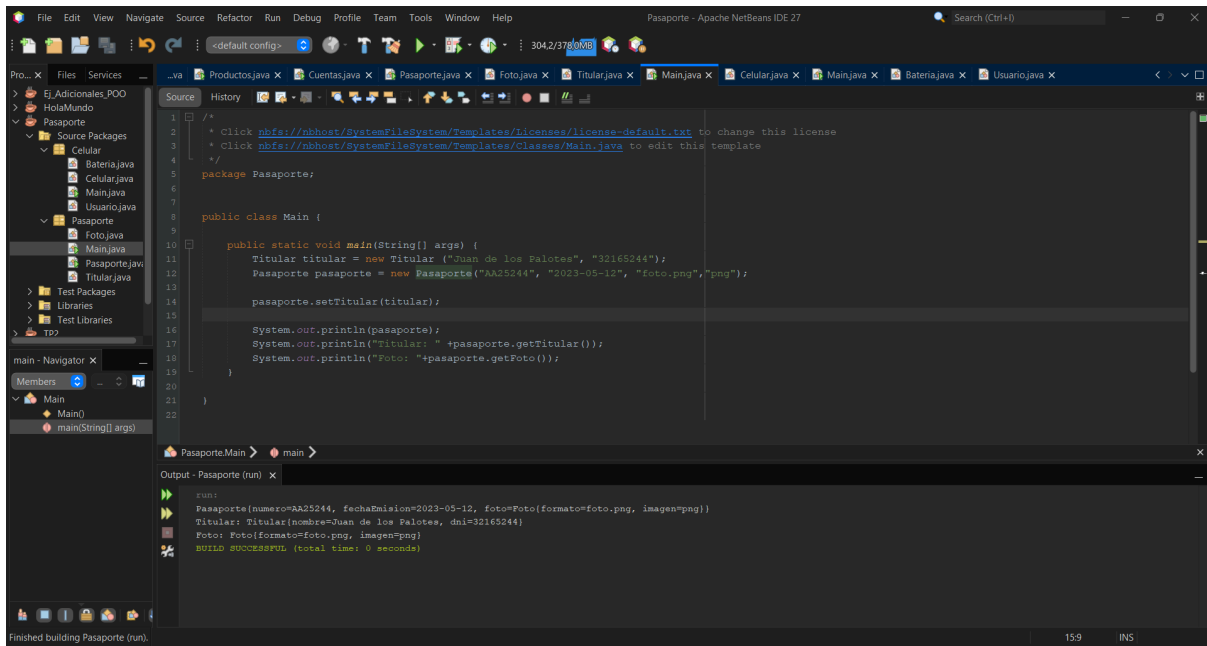
Enlace a repositorio en Github: <https://github.com/FioreGG/UTN-TUPaD-P2.git>

1. Resolución Pasaporte - Foto - Titular

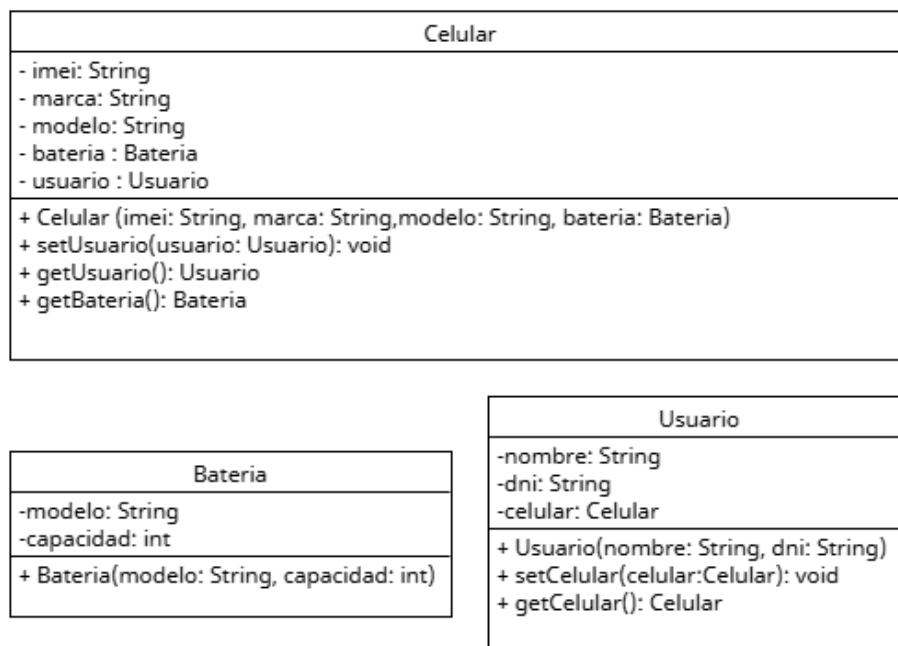


Pasaporte *-- Foto : composición

Pasaporte ↔ Titular : asociación



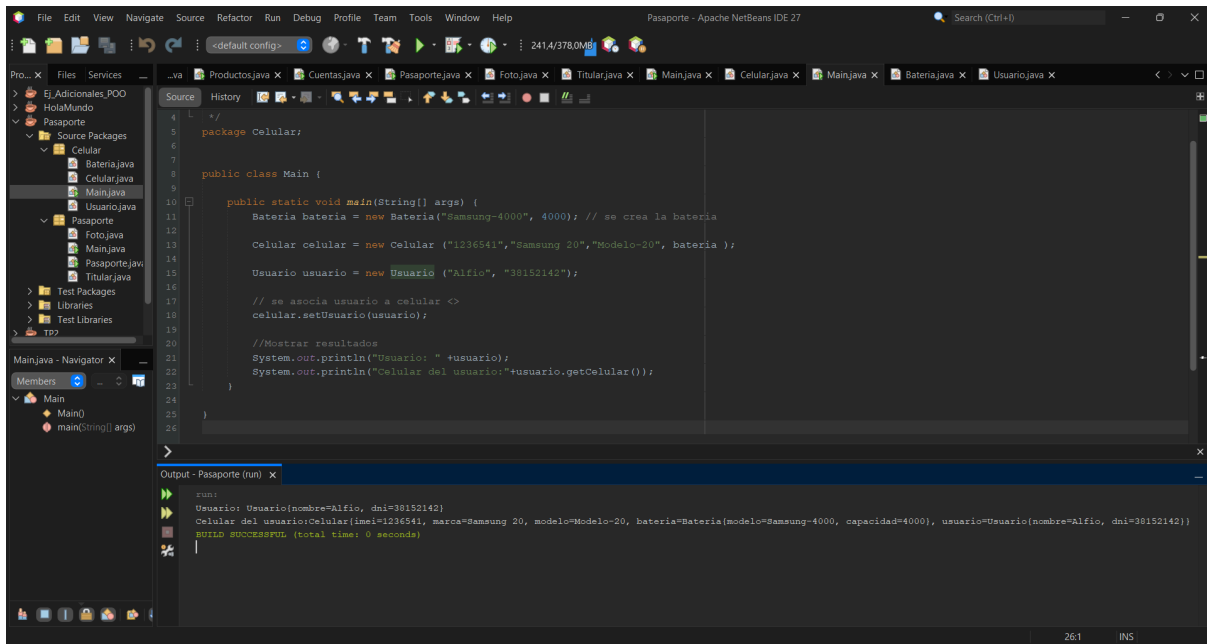
2. Resolucion Celular - Bateria - Usuario



Relaciones

Celular → Bateria : agregación

Celular ↔ Usuario : asociación bidireccional



3. Resolucion Libro - Autor - Editorial

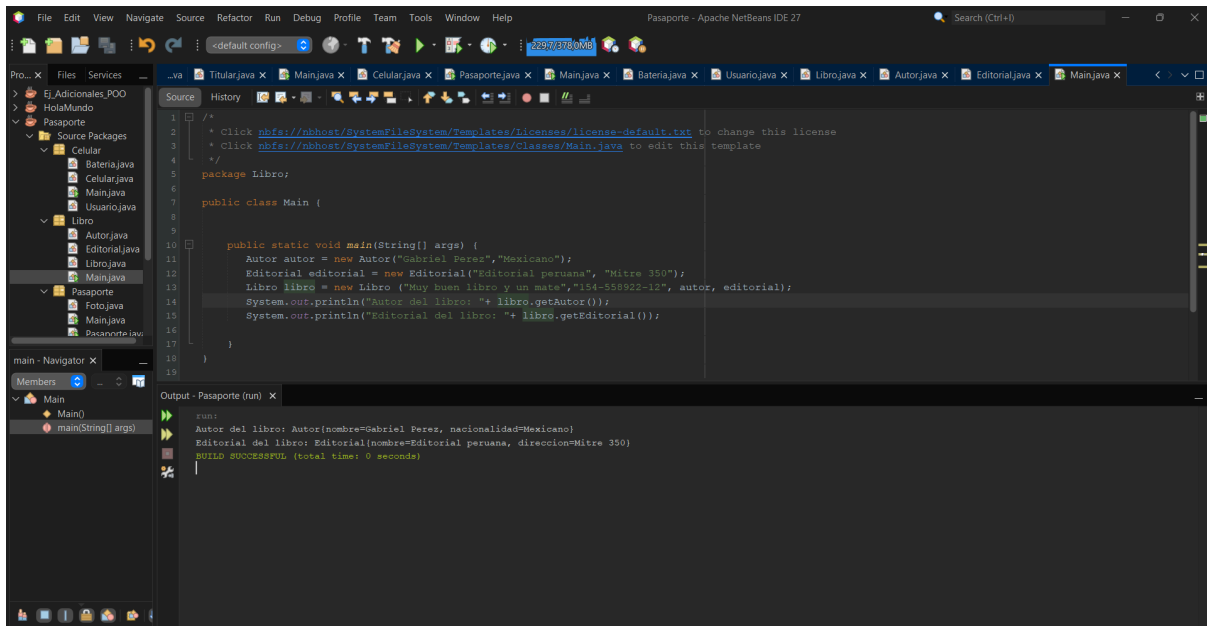
Libro
<ul style="list-style-type: none"> - titulo: String - isbn: String - autor: Autor - editorial: Editorial
<ul style="list-style-type: none"> + Libro (titulo: String, isbn: String, autor: Autor, editorial: Editorial) + getAutor(): Autor + getEditorial(): Editorial

Autor
<ul style="list-style-type: none"> -nombre: String -nacionalidad: String
<ul style="list-style-type: none"> + Autor(nombre: String, nacionalidad: String)

Editorial
<ul style="list-style-type: none"> -nombre: String -direccion: String
<ul style="list-style-type: none"> + Editorial(nombre: String, direccion: String)

Libro → Autor : asociación unidireccional

Libro → Editorial : agregación



4. Tarjeta de crédito - Cliente - Banco

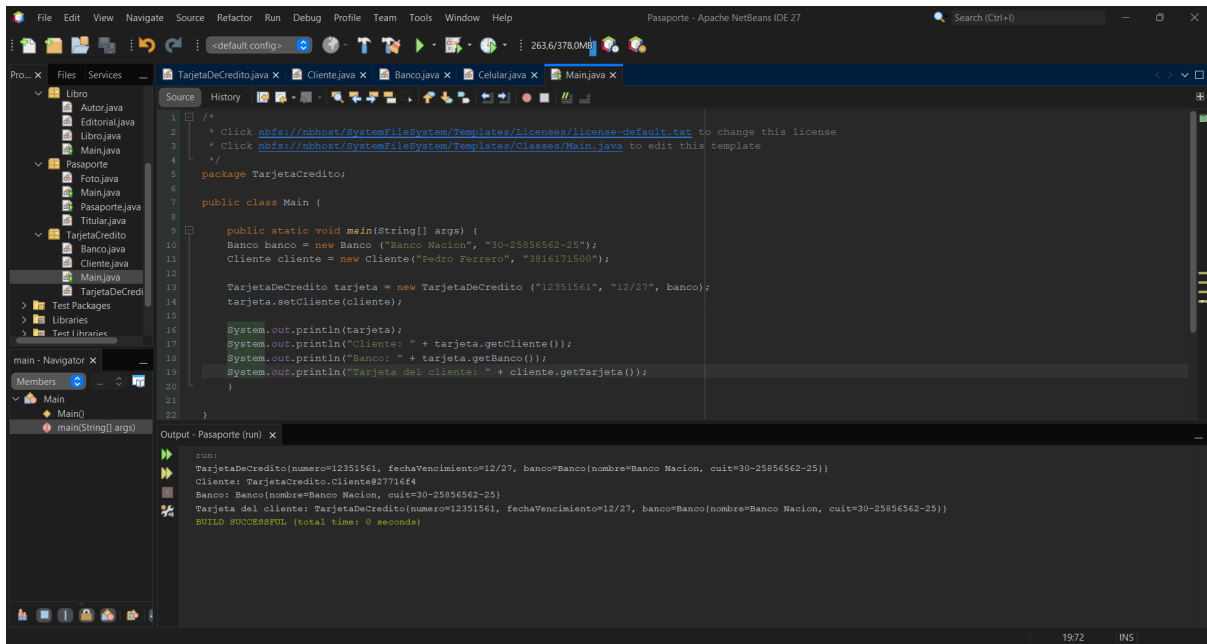
TarjetaDeCredito
<ul style="list-style-type: none"> - numero: String - fechaVencimiento: String - cliente: Cliente - banco: Banco
<ul style="list-style-type: none"> + TarjetaDeCredito(numero: String, fechaVencimiento: String, banco: Banco) + getCliente(): Cliente + getBanco(): Banco + setCliente(cliente: Cliente): void

Banco
<ul style="list-style-type: none"> - nombre: String - cuit: String
<ul style="list-style-type: none"> + Banco(nombre: String, cuit: String)

Cliente
<ul style="list-style-type: none"> - nombre: String - dni: String - tarjeta: TarjetaDeCredito
<ul style="list-style-type: none"> + Cliente(nombre: String, dni: String) + getTarjeta(): TarjetaDeCredito + setTarjeta(t: TarjetaDeCredito): void

TarjetaDeCrédito ↔ Cliente: Asociación bidireccional

TarjetaDeCrédito → Banco: Agregación



5. Computadora - PlacaMadre - Propietario

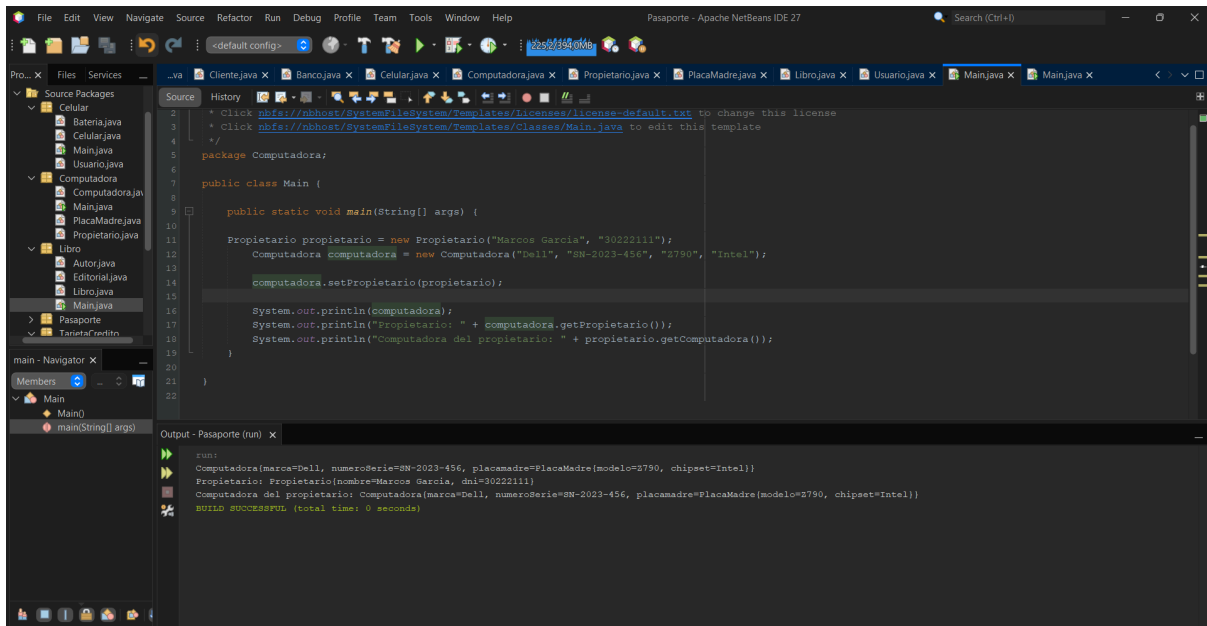
Computadora
<ul style="list-style-type: none"> - marca: String - numeroSerie: String - placamadre: PlacaMadre - propietario: Propietario
<ul style="list-style-type: none"> + Computadora(marca: String, numeroSerie: String, modelo: String, chipset: String) + getPropietario(): propietario + setPropietario(propietario: Propietario): void

PlacaMadre
<ul style="list-style-type: none"> - modelo: String - chipset: String
<ul style="list-style-type: none"> + PlacaMadre(modelo: String, chipset: String)

Propietario
<ul style="list-style-type: none"> - nombre: String - dni: String - computadora: Computadora
<ul style="list-style-type: none"> + Propietario(nombre: String, dni: String) + getComputadora(): computadora + setComputadora(computadora: Computadora): void

Composición: Computadora → PlacaMadre

Asociación bidireccional: Computadora ↔ Propietario



6. Reserva - Cliente - Mesa

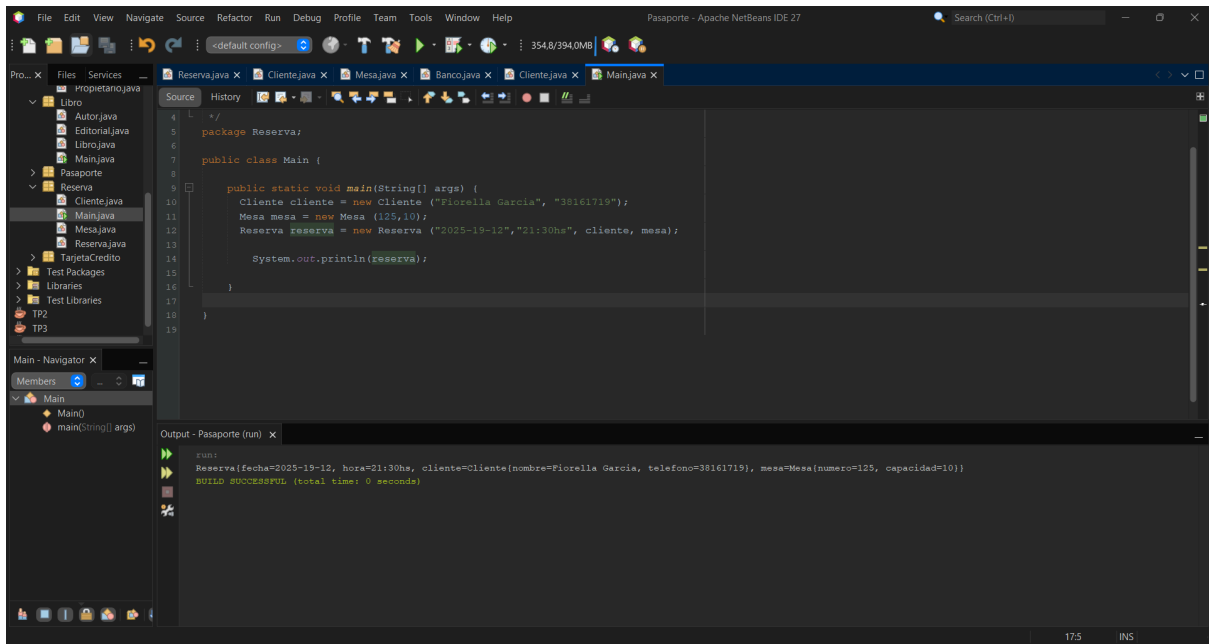
Asociación unidireccional: Reserva → Cliente

Agregación: Reserva → Mesa

Reserva
- fecha: String - hora: String - cliente: Cliente - mesa: Mesa
+ Reserva(fecha: String, hora: String, cliente: Cliente, mesa: Mesa)

Mesa
- numero: int - capacidad: int
+ Mesa(numero: int, capacidad: int)

Cliente
- nombre: String - telefono: String
+ Cliente(nombre: String, telefono: String)



7. Vehículo - Conductor - Motor

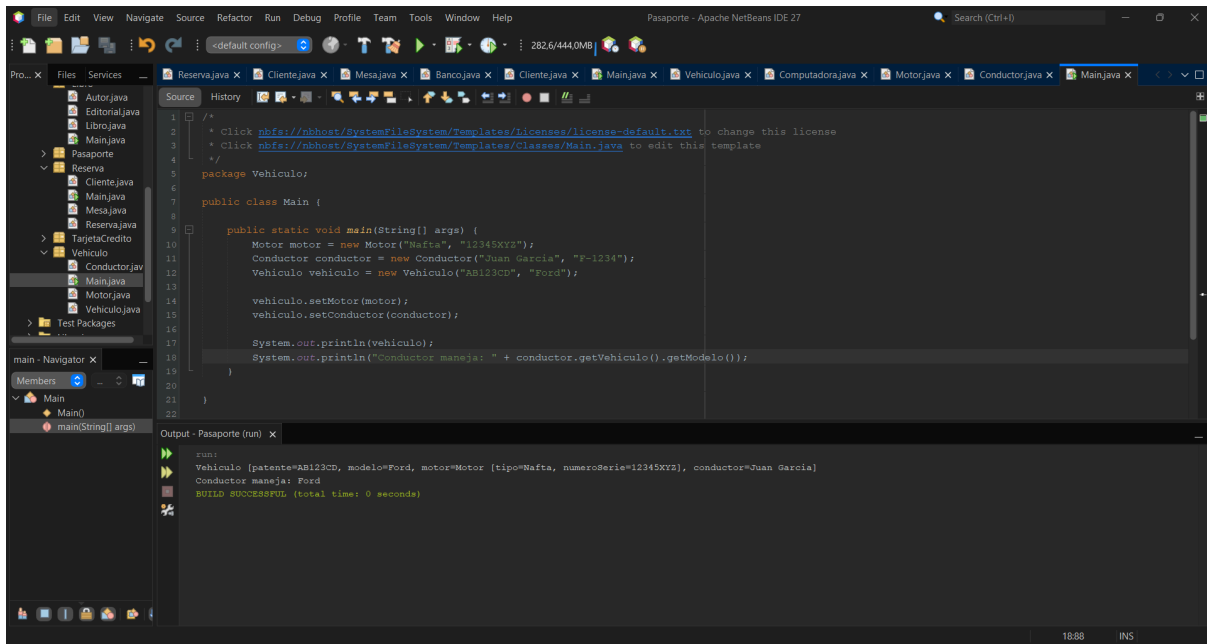
Agregación: Vehículo → Motor

Asociación bidireccional: Vehículo ↔ Conductor

Vehiculo
<ul style="list-style-type: none"> - patente: String - modelo: String - motor: Motor - conductor: Conductor
+ Vehiculo(patente: String, modelo: String, motor: Motor, conductor: Conductor)

Motor
<ul style="list-style-type: none"> - tipo: String - numeroSerie: String
+ Motor (tipo: String, numeroSerie: String)

Conductor
<ul style="list-style-type: none"> - nombre: String - licencia: String - vehiculo: Vehiculo
+ Clonductor(nombre: String, licencia: String) + setVehiculo(vehiculo: Vehiculo)



8. Documento - FirmaDigital - Usuario

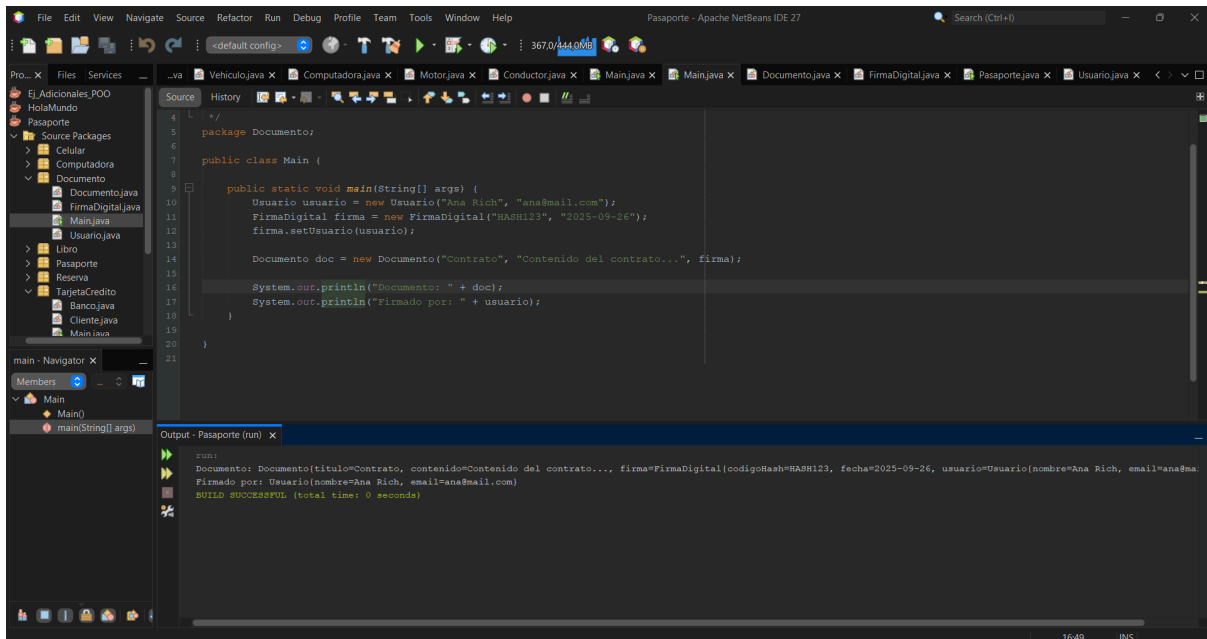
Composición: Documento → FirmaDigital

Agregación: FirmaDigital → Usuario

Documento
<ul style="list-style-type: none"> - titulo: String - contenido: String - FirmaDigital: firma
+ Documento(titulo: String, contenido: String, motor: Motor, FirmaDigital: firma)

FirmaDigital
<ul style="list-style-type: none"> - codigoHash: String - fecha: String - Usuario: usuario
+ FirmaDigital (codigoHash: String, fecha: String)
+ setUsuario(u: usuario)

Usuario
<ul style="list-style-type: none"> - nombre : String - email : String
+Usuario(nombre:String, email:String)



9. CitaMedica - Paciente - Profesional

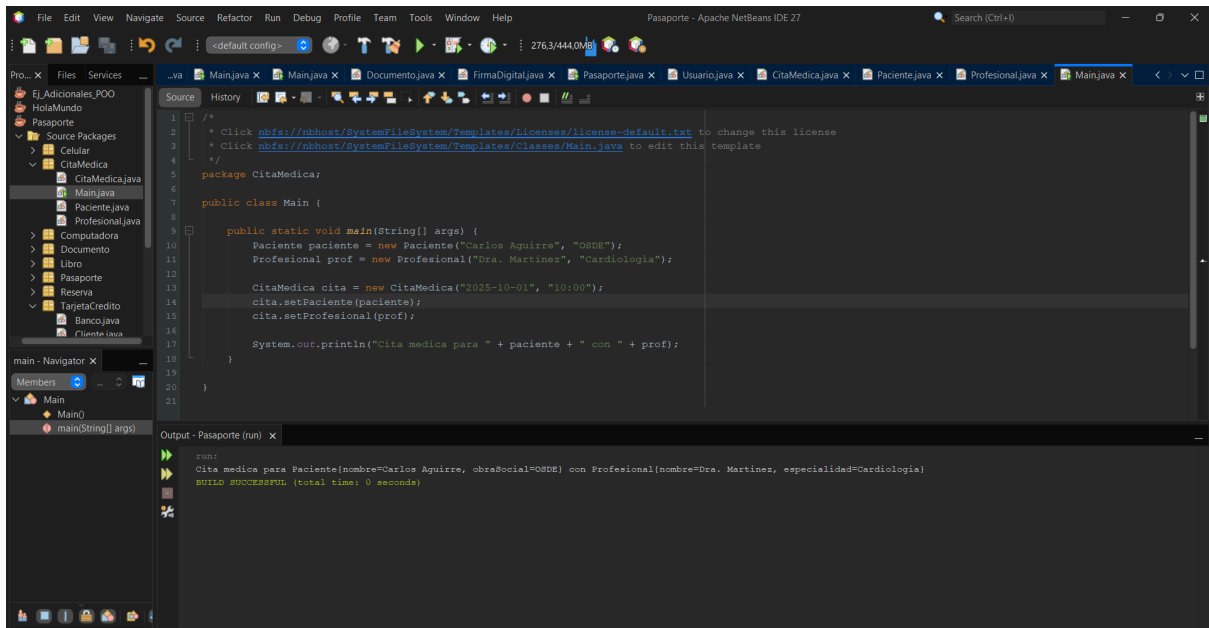
Asociación unidireccional: CitaMedica → Paciente,

Asociación unidireccional: CitaMedica → Profesional

CitaMedica
<ul style="list-style-type: none"> - fecha : String - hora : String - paciente : Paciente - profesional : Profesional
<ul style="list-style-type: none"> +CitaMedica(fecha:String, hora:String) +setPaciente(p:Paciente) +setProfesional(prof:Profesional)

Paciente
<ul style="list-style-type: none"> - nombre : String - obraSocial : String
<ul style="list-style-type: none"> + Paciente(nombre:String, obraSocial:String)

Profesional
<ul style="list-style-type: none"> - nombre : String - especialidad : String
<ul style="list-style-type: none"> +Profesional(nombre:String, especialidad:String)



10. CuentaBancaria - ClaveSeguridad - Titular

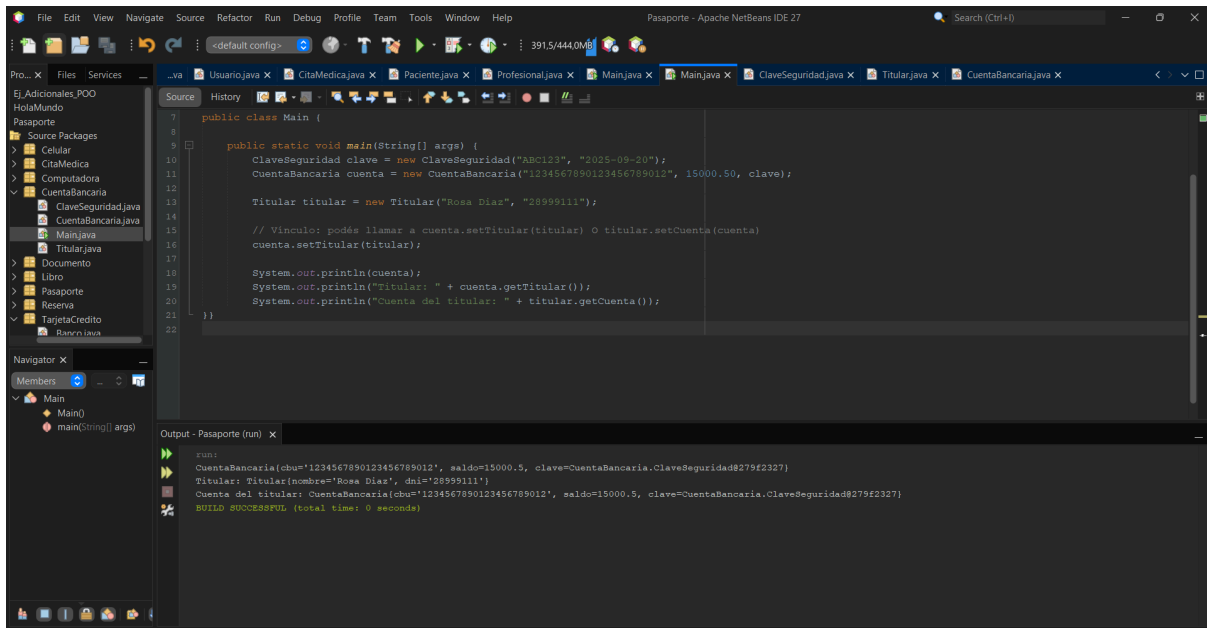
Composición: CuentaBancaria → ClaveSeguridad

Asociación bidireccional: CuentaBancaria ↔ Titular

CuentaBancaria
- cbu : String - saldo : double - clave : ClaveSeguridad - titular : Titular
+CuentaBancaria(cbu:String, saldo:double, clave:ClaveSeguridad, titular:Titular) +setTitular(t:Titular)

ClaveSeguridad
-codigo : String -ultimaModificacion : String
+ClaveSeguridad(codigo:String, ultima:String)

Titular
- nombre : String - dni : String - cuenta : CuentaBancaria
+Titular(nombre:String, dni:String) +setCuenta(c:CuentaBancaria)



11. Reproductor - Canción - Artista

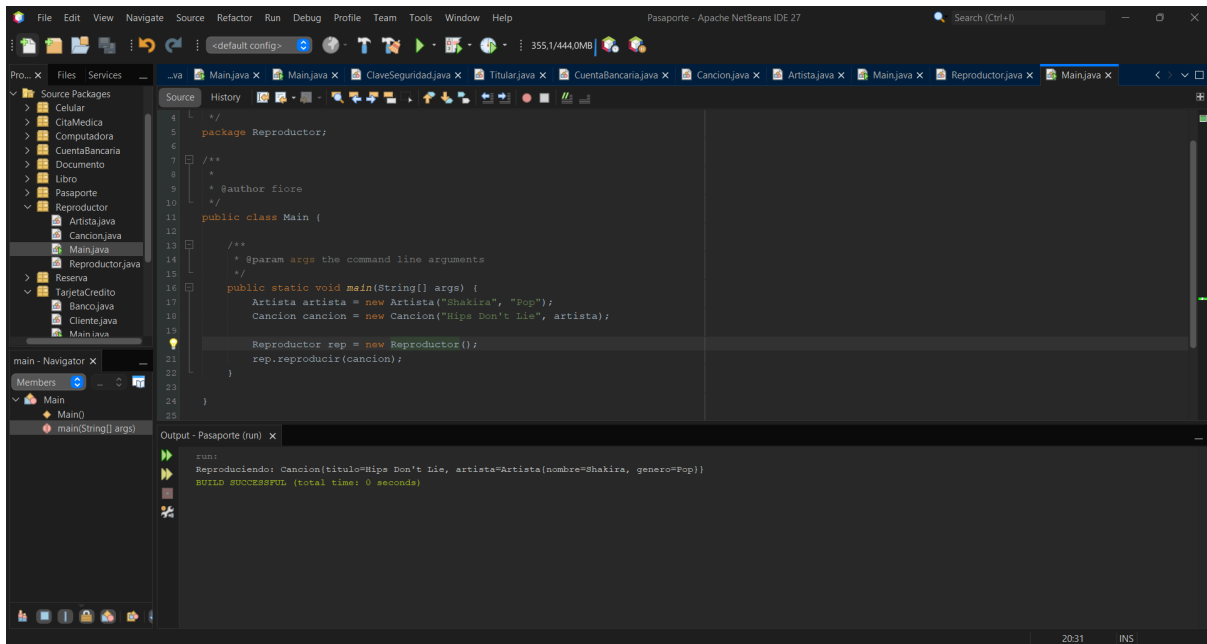
Asociación unidireccional: Canción → Artista

Dependencia de uso: Reproductor.reproducir(Cancion)

Cancion
- titulo : String
- artista : Artista
+Cancion(titulo:String, artista:Artista)

Artista
- nombre : String
- genero : String
+Artista(nombre:String, genero:String)

Reproductor
+ reproducir (c:Cancion)



12. Impuesto - Contribuyente - Calculadora

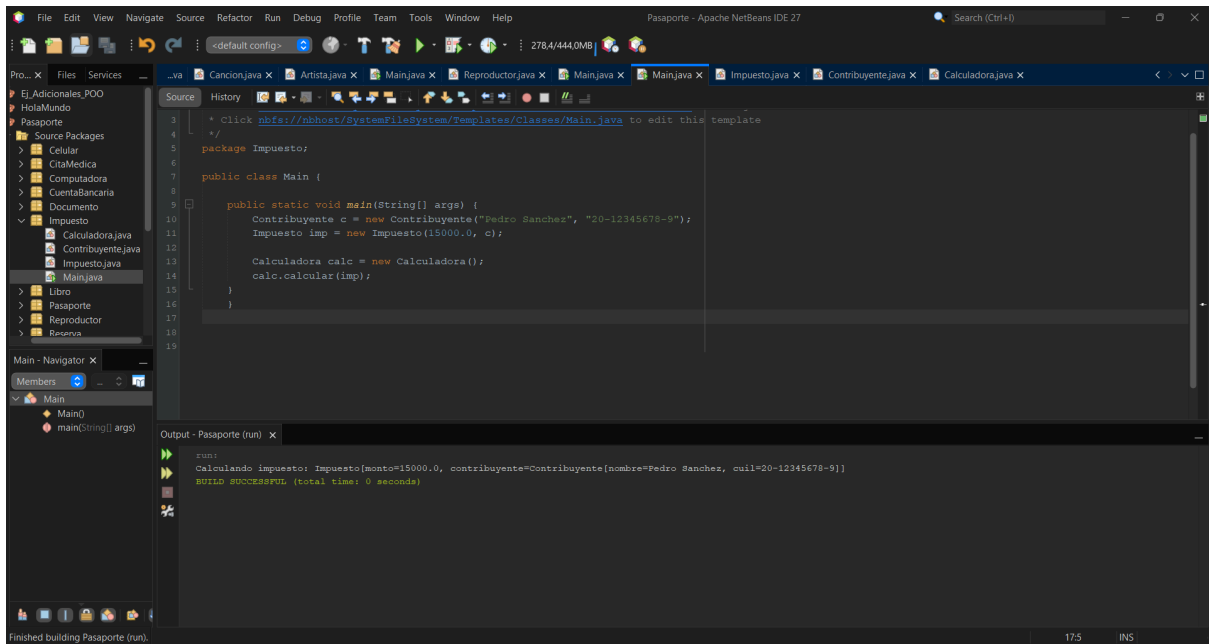
Asociación unidireccional: Impuesto → Contribuyente

Dependencia de uso: Calculadora.calcular(Impuesto)

Impuesto
- monto : double
- contribuyente : Contribuyente
+Impuesto(monto:double, c:Contribuyente)

Contribuyente
- nombre : String
- cuil : String
+Contribuyente(nombre:String, cuil:String)

Calculadora
+calcular(i:Impuesto)



13. GeneradorQR - Usuario - CódigoQR

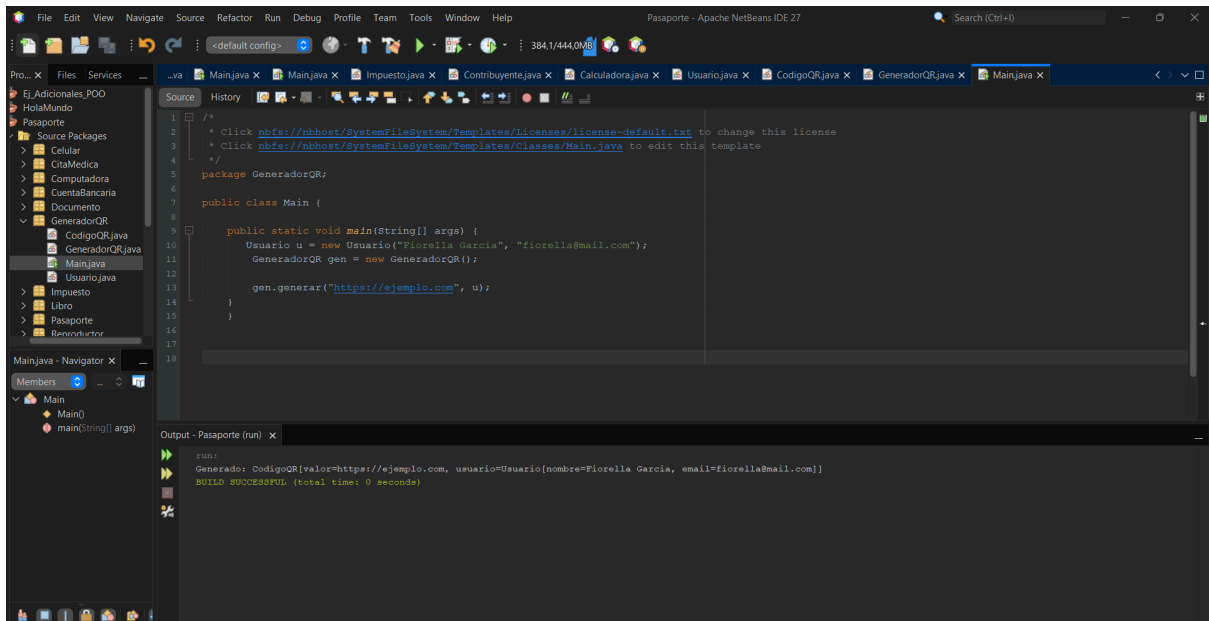
Asociación unidireccional: CódigoQR → Usuario

Dependencia de creación: GeneradorQR.generar(String, Usuario)

CódigoQR
- valor : String
- usuario : Usuario
+ CódigoQR(valor:String, usuario:Usuario)

Usuario
- nombre : String
- email : String
+ Usuario(nombre:String, email:String)

GeneradorQR
+ generar(valor:String, u:Usuario)



14. EditorVideo - Proyecto - Render

Asociación unidireccional: Render → Proyecto

Dependencia de creación: EditorVideo.exportar(String, Proyecto)

Render
- formato : String
- proyecto : Proyecto
+ Render(formato:String, proyecto:Proyecto)

Proyecto
- nombre : String
- duracionMin : int
+ Proyecto(nombre:String, duracionMin:int)

EditorVideo
+ exportar(formato:String, p:Proyecto)

