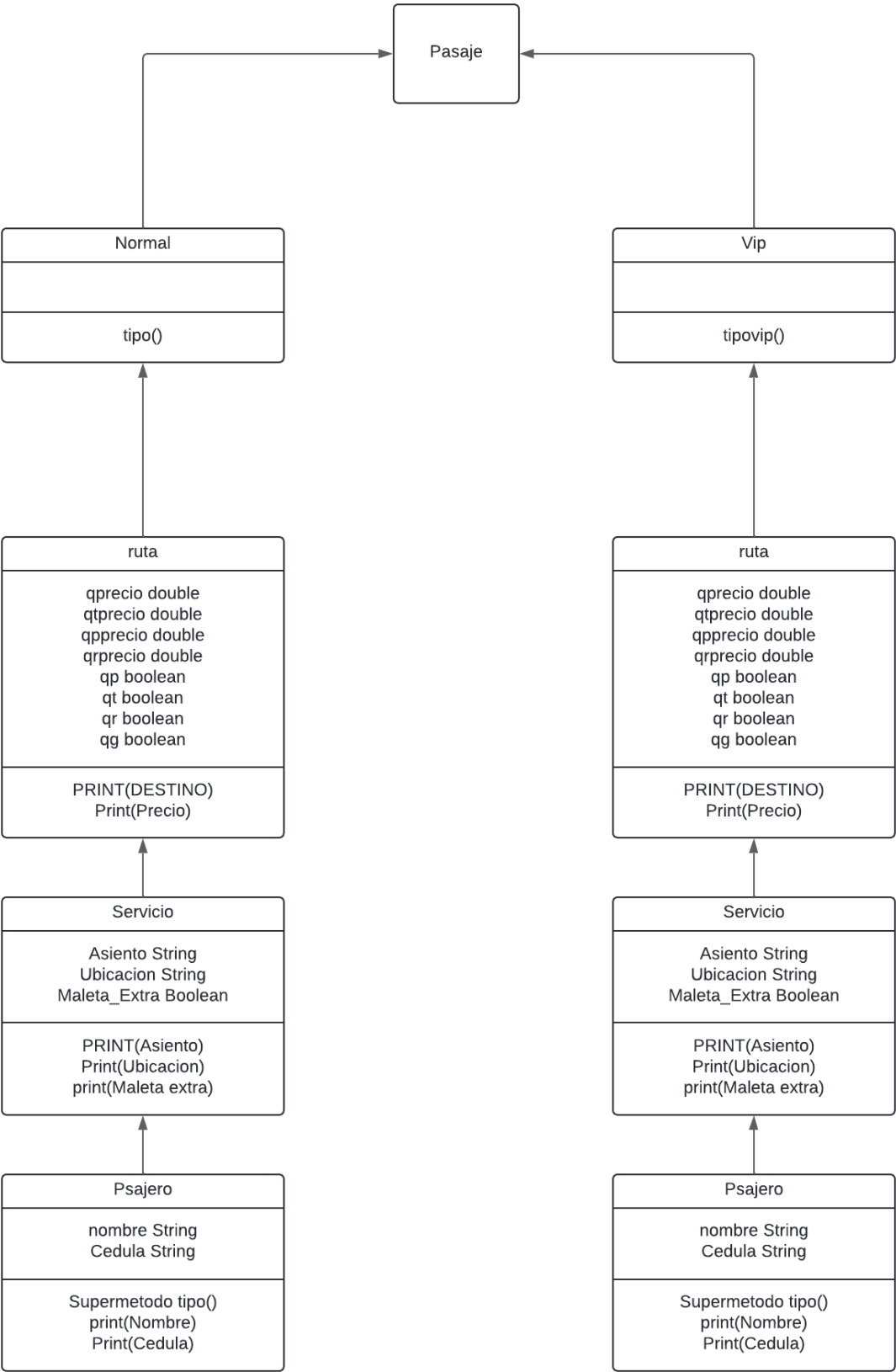


UML



## Capturas

```
Normal
Destino: Quito-Guayaquil
Precio: 20.0
Asiento: Ventana
Ubicacion: Atras
Maletaextra: false
Nombre: TADEO
Cedula: 0941077984
```

```
public class Main {  ▸ Adrian Cadena *
    public static void main(String[] args) {  ▸ Adrian Cadena *
        PasajeroNormal p1=new PasajeroNormal( normal: true, vip: false, qg: true, qt: false, qp: false, qr: false, ubicacion: "Atras", asien
        p1.tipo();
        PasajeroNormal p2=new PasajeroNormal( normal: true, vip: false, qg: false, qt: true, qp: false, qr: false, ubicacion: "Adelante", a
        p2.tipo();
        Psajero_VIP p3=new Psajero_VIP( normal: false, vip: true, qg: false, qt: false, qp: true, qr: false, television: true, dulces: true, c
        p3.tipovip();
        Psajero_VIP p4=new Psajero_VIP( normal: false, vip: true, qg: false, qt: false, qp: false, qr: true, television: false, dulces: true,
        p4.tipovip();
        Psajero_VIP p5=new Psajero_VIP( normal: false, vip: true, qg: false, qt: false, qp: true, qr: false, television: true, dulces: true, c
        p5.tipovip();
    }
```

```
9 Main.java  © Normal.java ×
1  @  public class Normal extends Pasaje{  1 usage  3 inheritors  ▸ Adrian Cadena
2      public Normal(Boolean normal, Boolean vip) {  1 usage  ▸ Adrian Cadena
3          super(normal, vip);
4      }
5      public Normal() {  1 usage  ▸ Adrian Cadena
6      }
7
8      @Override  7 usages  3 overrides  ▸ Adrian Cadena
9      public void tipo() {
10         super.tipo();
11     }
12 }
13
```

```
1 public class Pasaje { 2 usages 8 inheritors ⚡ Adrian Cadena *
2 public boolean getNormal() { no usages ⚡ Adrian Cadena
13 return Normal;
14 }
15 }
16
17 public void setNormal(Boolean normal) { no usages ⚡ Adrian Cadena
18 Normal = normal;
19 }
20
21 public Boolean getVip() { no usages ⚡ Adrian Cadena
22 return Vip;
23 }
24
25 public void setVip(Boolean vip) { no usages ⚡ Adrian Cadena
26 Vip = vip;
27 }
28 public void tipo(){ 7 usages 5 overrides ⚡ Adrian Cadena *
29 if (Normal==true && Vip==false){
30 System.out.println("Normal");
31 }
32 }
33 public void tipovip(){ 6 usages 3 overrides ⚡ Adrian Cadena *
34 if (Normal==false && Vip==true){
35 System.out.println("Vip");
```

```
1 public class PasajeroNormal extends Servicio{ 4 usages ⚡ Adrian Cadena *
2 String Nombre; 4 usages
3 String Cedula; 4 usages
4
5 public PasajeroNormal(Boolean normal, Boolean vip, boolean qg, boolean qt, boolean qp, boolean qr, String ubicacion, String asiento, boolean
6 super(normal, vip, qg, qt, qp, qr, ubicacion, asiento, maletaextra);
7 Nombre = nombre;
8 Cedula = cedula;
9 }
10
11 public String getCedula() { no usages new *
12 return Cedula;
13 }
14
15 public void setCedula(String cedula) { no usages new *
16 Cedula = cedula;
17 }
18
19 public String getNombre() { no usages new *
20 return Nombre;
21 }
22
23 public void setNombre(String nombre) { no usages new *
24 Nombre = nombre;
```

```
Current File ▾ ▶ ⚙ ⋮ 👤 🔍 ⚙ - □ ×

Main.java Normal.java PasajeroNormal.java Psajero_VIP.java ×

1 public class Psajero_VIP extends ServicioVip { 6 usages new *
2     String Nombre; 4 usages
3     String Cedula; 4 usages
4
5     public Psajero_VIP(Boolean normal, Boolean vip, boolean qg, boolean qt, boolean qp, boolean qr, boolean television, boolean dulces, boolean
6         super(normal, vip, qg, qt, qp, qr, television, dulces, comida, internet);
7         Nombre = nombre;
8         Cedula = cedula;
9     }
10
11     public String getCedula() { no usages new *
12         return Cedula;
13     }
14
15     public void setCedula(String cedula) { no usages new *
16         Cedula = cedula;
17     }
18
19     public String getNombre() { no usages new *
20         return Nombre;
21     }
22
23     public void setNombre(String nombre) { no usages new *
24         Nombre = nombre;
25     }
26 }
```

```
Main.java Normal.java PasajeroNormal.java Psajero_VIP.java Ruta.java ×

2 public class Ruta extends Normal{ 1 usage 2 inheritors ▲ Adrian Cadena *
3
4     public void setPrecioqr(double precioqr) { this.precioqr = precioqr; }
5
6     @Override 7 usages 2 overrides ▲ Adrian Cadena
7     public void tipo() {
8         super.tipo();
9         if(qg==true){
10             System.out.println("Destino: Quito-Guayaquil");
11             System.out.println("Precio: "+precioqg);
12         }
13         if(qt==true){
14             System.out.println("Destino: Quito-Tulcan");
15             System.out.println("Precio: "+precioqt);
16         }
17         if(qp==true){
18             System.out.println("Destino: Quito-Puyo");
19             System.out.println("Precio: "+precioqp);
20         }
21         if(qr==true){
22             System.out.println("Destino: Quito-Riobamba");
23             System.out.println("Precio: "+precioqr);
24         }
25     }
26 }
```

```
Main.java Normal.java PasajeroNormal.java Psajero_VIP.java Ruta.java Ruta_Vip.java ×

1 public class Ruta_Vip extends Vip { 1 usage 2 inheritors new *
2
3     @Override 6 usages 2 overrides new *
4     public void tipovip() {
5         super.tipovip();
6         if (qg == true) {
7             System.out.println("Destino: Quito-Guayaquil");
8             System.out.println("Precio: " + precioqgvip);
9         }
10         if (qt == true) {
11             System.out.println("Destino: Quito-Tulcan");
12             System.out.println("Precio: " + precioqtvip);
13         }
14         if (qp == true) {
15             System.out.println("Destino: Quito-Puyo");
16             System.out.println("Precio: " + precioqpvip);
17         }
18         if (qr == true) {
19             System.out.println("Destino: Quito-Riobamba");
20             System.out.println("Precio: " + precioqrvip);
21         }
22     }
23 }
```

```
Current File
Main.java Normal.java PasajeroNormal.java Psajero_VIP.java Ruta.java Ruta_Vip.java Servicio.java x
1 public class Servicio extends Ruta { 1 usage 1 inheritor Adrian Cadena *
2     String Asiento; 4 usages
3     String Ubicacion; 4 usages
4     boolean Maletaextra; 4 usages
5
6     public Servicio() { no usages Adrian Cadena
7     }
8
9     public Servicio(Boolean normal, Boolean vip, boolean qq, boolean qt, boolean qp, boolean qr, String ubicacion, String asiento, boolean )
10         super(normal, vip, qq, qt, qp, qr);
11         Ubicacion = ubicacion;
12         Asiento = asiento;
13         Maletaextra = maletaextra;
14     }
15
16     public String getAsiento() { no usages new *
17         return Asiento;
18     }
19
20     public void setAsiento(String asiento) { no usages new *
21         Asiento = asiento;
22     }
23
24     public boolean isMaletaextra() { no usages new *
```

Cedula: 1720559684  
Vip  
Destino: Quito-Riobamba  
Precio: 22.75  
Television: false  
Dulces: true  
Comida: false  
Internet: true  
Nombre: Amaralis  
Cedula: 12125478963  
Vip  
Destino: Quito-Puyo  
Precio: 19.5  
Television: true

Ubicacion: Adelante  
Maletaextra: true  
Nombre: Madeline  
Cedula: 0941077978  
Vip  
Destino: Quito-Puyo  
Precio: 19.5  
Television: true  
Dulces: true  
Comida: true  
Internet: true  
Nombre: Sofia  
Cedula: 1720559684