

# Sistema de Transporte - Pseudocódigo

```
1 1. Inicio del programa
2 INICIAR sistema ← nuevo SistemaTransporte
3 LLAMAR menu_principal(sistema)
4 FIN
```

Listing 1: Programa Principal

```
1 2. Clase Ruta
2 CLASE Ruta:
3   ATRIBUTOS:
4     idRuta, destino, origen, paradas
5     horaSalida, horaLlegadaEstimada
6     conductorAsignado, busAsignado
7     ubicacionActual
8
9   METODO __init__():
10    guardar datos iniciales
11    ubicacionActual = origen
12
13  METODO asignarConductor(conductor):
14    conductorAsignado = conductor
15    conductor.asignarRuta(idRuta)
16
17  METODO asignarBus(bus):
18    busAsignado = bus
19    bus.asignarRuta(idRuta)
20
21  METODO actualizarUbicacion(nuevaUbicacion):
22    ubicacionActual = nuevaUbicacion
```

Listing 2: Clase Ruta

```
1 3. Clase Bus
2 CLASE Bus:
3   ATRIBUTOS:
4     idBus, placa, capacidad
5     ubicacionActual = "Terminal"
6     rutaAsignada, pasajerosActuales
7     conductor, viajeActivo
8
9   METODO asignarRuta(ruta):
10    rutaAsignada = ruta
11
12  METODO asignarConductor(conductor):
13    conductor = conductor
14
15  METODO actualizarUbicacion(ubicacion):
```

```

16     ubicacionActual = ubicacion
17
18     METODO obtenerCupoDisponible():
19         RETORNAR capacidad - pasajerosActuales

```

Listing 3: Clase Bus

```

1 4. Clase Conductor
2 CLASE Conductor:
3     ATRIBUTOS:
4         idConductor, nombre, cedula, telefono
5         rutaActual, totalGanancias
6
7     METODO asignarRuta(ruta):
8         rutaActual = ruta
9
10    METODO registrarGanancia(monto):
11        totalGanancias = totalGanancias + monto
12
13    METODO obtenerGanancias():
14        RETORNAR totalGanancias

```

Listing 4: Clase Conductor

```

1 5. Clase Pasajero
2 CLASE Pasajero:
3     ATRIBUTOS:
4         idPasajero, saldoTarjeta
5         ubicacionActual, enBus
6
7     METODO pagarPasaje():
8         SI saldoTarjeta ≥ TARIFA:
9             saldoTarjeta = saldoTarjeta - TARIFA
10            RETORNAR VERDADERO
11        SI NO:
12            RETORNAR FALSO
13
14     METODO registrarIngreso(bus):
15         SI bus.cupoDisponible > 0:
16             enBus = VERDADERO
17             bus.pasajerosActuales++
18             RETORNAR VERDADERO
19         SI NO:
20             RETORNAR FALSO
21
22     METODO registrarSalida(bus):
23         SI enBus ES VERDADERO:
24             enBus = FALSO
25             bus.pasajerosActuales--
26             RETORNAR VERDADERO
27         SI NO:
28             RETORNAR FALSO
29
30     METODO recargarTarjeta(monto):
31         saldoTarjeta = saldoTarjeta + monto

```

Listing 5: Clase Pasajero

```

1 6. Clase Viaje
2 CLASE Viaje:
3   ATRIBUTOS:
4     idViaje, ruta, bus, conductor
5     pasajerosTransportados
6     gananciaGenerada, ingresosBrutos
7     costos = COSTO_FIJO
8     fecha = fecha actual
9
10  METODO registrarPasajero():
11    pasajerosTransportados++
12
13  METODO cerrarViaje():
14    ingresosBrutos = pasajerosTransportados * TARIFA
15    gananciaGenerada = ingresosBrutos - costos
16
17  SI gananciaGenerada > 0:
18    conductor.registrarGanancia(gananciaGenerada)
19  SI NO:
20    conductor.registrarGanancia(0)
21
22  RETORNAR REPORTE con toda la informacion

```

Listing 6: Clase Viaje

```

1 7. Clase SistemaTransporte
2 CLASE SistemaTransporte:
3   ATRIBUTOS:
4     listas rutas, buses, conductores, pasajeros, viajes

```

Listing 7: Clase Sistema de Transporte

```

1 8. Registrar rutas
2 METODO registrarRuta():
3   pedir idRuta, origen, destino, paradas, horaSalida, horaLlegada
4   crear objeto Ruta
5   almacenar en rutas[idRuta]
6
7 9. Registrar bus
8 METODO registrarBus():
9   pedir idBus, placa
10  crear objeto Bus
11  almacenar en buses[idBus]
12
13 10. Registrar conductor
14 METODO registrarConductor():
15  pedir datos del conductor
16  crear objeto Conductor
17  guardar en conductores[id]
18
19 11. Registrar pasajero
20 METODO registrarPasajero():
21  pedir datos del pasajero
22  crear objeto Pasajero
23  agregar a pasajeros[id]
24
25 12. Iniciar viaje
26 METODO iniciarViaje():

```

```

27     pedir idViaje, idRuta, idBus, idConductor
28
29     SI alguno NO existe:
30         mostrar error y salir
31
32     crear nuevo Viaje
33     guardar en viajes[idViaje]
34
35     bus.viajeActivo ← idViaje
36     ruta.asignarBus(bus)
37     ruta.asignarConductor(conductor)
38
39     mostrar "Viaje iniciado"
40
41 13. Registrar ingreso de pasajero
42 METODO registrarIngreso():
43     pedir idPasajero, idBus
44
45     validar existencia de pasajero y bus
46
47     SI bus no tiene viajeActivo:
48         mostrar error y salir
49
50     SI pasajero NO puede pagar pasaje:
51         mostrar "Saldo insuficiente" y salir
52
53     SI registrarIngreso(bus) ES VERDADERO:
54         viaje.registrarPasajero()
55         mostrar "Ingreso registrado"
56     SI NO:
57         mostrar "Bus lleno"
58
59 14. Registrar salida
60 METODO registrarSalida():
61     pedir idPasajero, idBus
62
63     SI registrarSalida(bus) ES VERDADERO:
64         mostrar "Salida registrada"
65     SI NO:
66         mostrar "Pasajero no esta en el bus"
67
68 15. Consultar cupos
69 METODO consultarCuposBus():
70     pedir idBus
71     mostrar capacidad disponible
72
73 16. Consultar ubicacion
74 METODO consultarUbicacionBus():
75     pedir idBus
76     mostrar ubicacionActual
77
78 17. Cerrar viaje
79 METODO cerrarViaje():
80     pedir idViaje
81
82     obtener reporte = viaje.cerrarViaje()
83
84     limpiar viajeActivo del bus

```

```
85  
86     mostrar reporte completo
```

Listing 8: Metodos del Sistema

```
1 18. Menu principal del sistema  
2  
3 FUNCION menu_principal(sistema):  
4  
5 MIENTRAS VERDADERO:  
6     mostrar opciones  
7  
8     leer opcion  
9  
10    SEGUN opcion:  
11        1 = sistema.registrarRuta()  
12        2 = sistema.registrarBus()  
13        3 = sistema.registrarConductor()  
14        4 = sistema.registrarPasajero()  
15        5 = sistema.iniciarViaje()  
16        6 = sistema.registrarIngreso()  
17        7 = sistema.registrarSalida()  
18        8 = sistema.consultarCuposBus()  
19        9 = sistema.consultarUbicacionBus()  
20        10 = sistema.cerrarViaje()  
21        11 = salir del programa  
22        otro = mostrar "Opcion invalida"
```

Listing 9: Menu Principal