

Examen del primer bimestre de Programación Orientada a Objetos

Desarrollo de un Sistema de Compra de Pasajes de Bus

La empresa de transportes “El Búho” desea desarrollar un sistema que permita la compra de tickets o pasajes de bus a varias regiones del país, para este proyecto usted ha sido contratado para que desarrolle dicho sistema.

La empresa tiene las siguientes rutas:

- Quito-Guayaquil (pasaje: \$20)
- Quito-Tulcán (pasaje: \$17,50)
- Quito-Puyo (pasaje: \$15)
- Quito-Riobamba (pasaje: \$17,50)

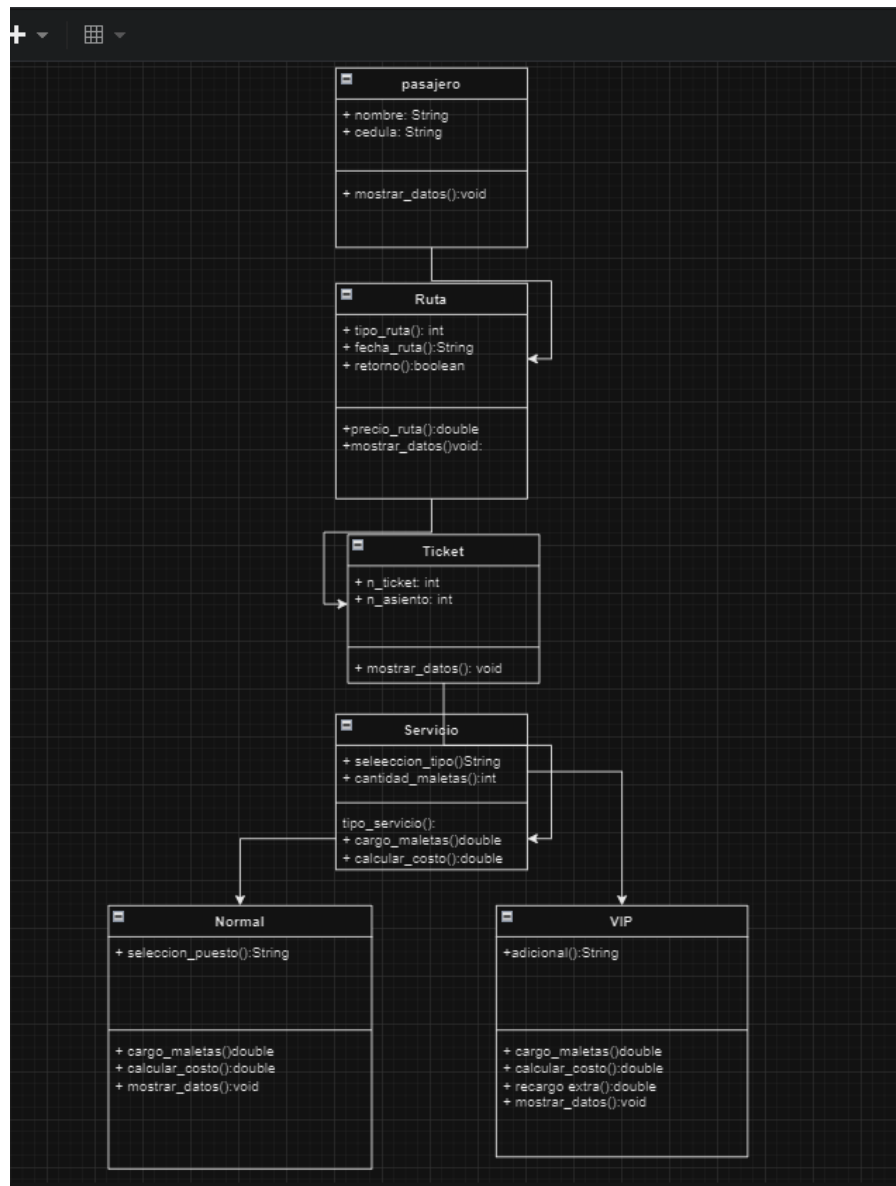
También tiene las rutas de retorno desde cada ciudad de destino, las cuales tienen el mismo precio.

Cada ruta tiene dos tipos de servicios:

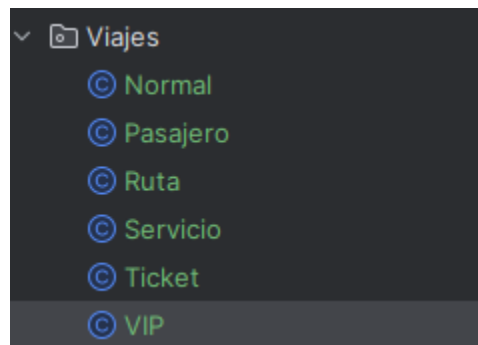
1. **Normal:** ○ Selección de asientos: ventana, pasillo, al final, adelante. ○ Espacio para una maleta.
 - Opción a una maleta adicional por costo extra.
2. **VIP** (tiene un recargo del 30% del valor del pasaje normal) ○ Televisión, internet y otros dos servicios adicionales a su consideración. ○ Espacio para dos maletas. ○ Opción a una maleta adicional por costo extra.

Una vez que ha sido contratado se solicita que utilice java y conceptos de programación orientada a objetos para iniciar el proyecto. Su misión es desarrollar los siguientes entregables:

1. **Diagrama UML** (2 puntos) este entregable lo puede adjuntar en el pdf entregable.
 - Crear un diagrama UML que incluya todas las clases necesarias, sus atributos y métodos.
 - Mostrar claramente las relaciones de herencia.



2. **Package y Clases** (2 puntos) ○ Crear un paquete en Java donde se incluirán todas las clases implementadas.



3. **Clases.**- se debe generar al menos las siguientes clases y relacionarlas de acuerdo al diseño de su diagrama: ○ Clase Ticket ○ Clases Normal y VIP

- Clase Ruta para manejar la información de las rutas.
- Clase Servicio para manejar los servicios adicionales.
- Clase Pasajero para manejar la información del pasajero.

Implementar constructores, setters, getters y métodos personalizados en cada clase. (8 puntos)

```
1 package org.example.Viajes;
2
3 public class Pasajero { 4 usages 3 inheritors new *
4     String nombre; 4 usages
5     String cedula; 4 usages
6
7     public Pasajero() {} 1 usage new *
8     public Pasajero(String nombre, String cedula) { 2 usages new *
9         this.nombre = nombre;
10        this.cedula = cedula;
11    }
12
13    public String getNombre() { 1 usage new *
14        return nombre;
15    }
16
17    public void setNombre(String nombre) { no usages new *
18        this.nombre = nombre;
19    }
20
21    public String getCedula() { 1 usage new *
22        return cedula;
23    }
24
25    public void setCedula(String cedula) { no usages new *
26        this.cedula = cedula;
27    }
28
29    //Metodo
30    public void mostrar_datos(){ 5 usages 3 overrides new *
31        System.out.println("Nombre: " + nombre);
32        System.out.println("Cedula: " + cedula);
33    }
34 }
```

```

1 package org.example.Viajes;
2
3 public class Ruta extends Pasajero{ 4 usages 2 inheritors new *
4     double precio; 12 usages
5     int tipo_ruta; 5 usages
6     String fecha_ruta; 2 usages
7     boolean retorno; 3 usages
8
9     public Ruta() {} 1 usage new *
10
11     public Ruta(String nombre, String cedula, double precio, int tipo_ruta, String fecha_ruta, boolean retorno) {
12         super(nombre, cedula);
13         this.precio = precio;
14         this.tipo_ruta = tipo_ruta;
15         this.fecha_ruta = fecha_ruta;
16         this.retorno = retorno;
17     }
18
19     //Metodo Personalizado
20     public void precio_ruta(){ 1 usage new *
21         if (tipo_ruta == 1 ){
22             System.out.println("Su ruta es de Quito-Guayaquil");
23             precio=20;
24             System.out.println("Su precio es:"+precio);
25         }else if (tipo_ruta == 2 ){
26             System.out.println("Su ruta es de Quito-Puyo");
27             precio=15;
28             System.out.println("Su precio es:"+precio);
29         }else if (tipo_ruta == 3 ){
30             System.out.println("Su ruta es de Quito-Tulcan");
31             precio=17.50;
32             System.out.println("Su precio es:"+precio);
33         }else if (tipo_ruta == 4 ){
34             System.out.println("Su ruta es de Quito-Riobamba");
35             precio=17.50;
36             System.out.println("Su precio es:"+precio);
37         }
38     }
39
40     public void retorno_v(){ 2 usages 1 override new *
41         if (retorno == true ){
42             precio=precio+precio;

```

Main.java Pasajero.java Ruta.java Ticket.java x Servicio.java VIP.java Normal.java

```

package org.example.Viajes;

public class Ticket extends Ruta { 3 usages 1 inheritor new *
    int n_ticket; 4 usages
    int n_asiento; 4 usages
    public Ticket() {} no usages new *

    public Ticket(String nombre, String cedula, double precio, int tipo_ruta, String fecha_ruta, boolean retorno, int n_ticket, int n_asiento) {
        super(nombre, cedula, precio, tipo_ruta, fecha_ruta, retorno);
        this.n_ticket = n_ticket;
        this.n_asiento = n_asiento;
    }

    public int getN_ticket() { no usages new *
        return n_ticket;
    }

    public void setN_ticket(int n_ticket) { no usages new *
        this.n_ticket = n_ticket;
    }

    public int getN_asiento() { no usages new *
        return n_asiento;
    }

    public void setN_asiento(int n_asiento) { no usages new *
        this.n_asiento = n_asiento;
    }
}

```

//Metodos

```

@Override 2 usages new *
public void retorno_v() {
    super.retorno_v();
}
@Override 2 usages 1 override new *

```

```

Main.java Pasajero.java Ruta.java Ticket.java Servicio.java x VIP.java N
1 package org.example.Viajes;
2
3 public class Servicio extends Ticket{ no usages new *
4     int seleccion_tipo; 4 usages
5     int cantidad_maleta; 4 usages
6     public Servicio() {} no usages new *
7
8
9     public Servicio(String nombre, String cedula, double precio, int tipo_ruta, String fecha
10         super(nombre, cedula, precio, tipo_ruta, fecha_ruta, retorno, n_ticket, n_asiento);
11         this.seleccion_tipo = seleccion_tipo;
12         this.cantidad_maleta = cantidad_maleta;
13     }
14
15     public int getSeleccion_tipo() { no usages new *
16         return seleccion_tipo;
17     }
18
19     public void setSeleccion_tipo(int seleccion_tipo) { no usages new *
20         this.seleccion_tipo = seleccion_tipo;
21     }
22
23     public int getCantidad_maleta() { no usages new *
24         return cantidad_maleta;
25     }
26
27     public void setCantidad_maleta(int cantidad_maleta) { no usages new *
28         this.cantidad_maleta = cantidad_maleta;
29     }
30     //Metodo
31
32     @Override 5 usages new *
33     public void mostrar_datos() {
34         super.mostrar_datos();
35         System.out.println("Servicio seleccionado: " + seleccion_tipo);
36         System.out.println("Cantidad maleta: " + cantidad_maleta);
37     }
38 }

```

4. **Implementar Herencia Multinivel** (2 puntos) ○ Crear una clase base y al menos dos niveles de herencia adicionales.
5. **Métodos Personalizados** (2 puntos) ○ Crear métodos para calcular el costo del pasaje según la ruta y el tipo de servicio seleccionado.
 - Crear métodos personalizados para imprimir datos del objeto.
6. **Crear Objetos** (2 puntos) ○ Instanciar al menos 5 objetos con valores nulos y posteriormente setear algunos de sus atributos. Utilizar setters con valores quemados, no es necesario el ingreso de datos por consola.
 - Instanciar al menos 5 objetos con valores iniciales.
 - Para ambos casos imprimir la información de los objetos.
7. **Uso de GitHub** (2 puntos) ○ Crear un repositorio en GitHub.
 - Realizar al menos 2 commit y push que reflejen el progreso del proyecto. ○ Proveer una captura de pantalla del código funcionando correctamente y guardarla en formato PDF.

Indicaciones adicionales:

- En el aula virtual debe subir el pdf con las capturas de pantalla y el link de github.
- Por favor vayan guardando cualquier cambio ya sea local o en su repositorio para evitar cualquier pérdida en el caso de falla de su equipo.
- **NO es necesario crear un menú ni ingresar los datos por consola.**
- No pueden utilizar chatgpt o similares.
- En caso de copia se retirará el examen.

