

Actividad de Aprendizaje Contacto Docente N.º 2 Primer Parcial

Nombre: Leonardo Tipan

Nrc: 1323

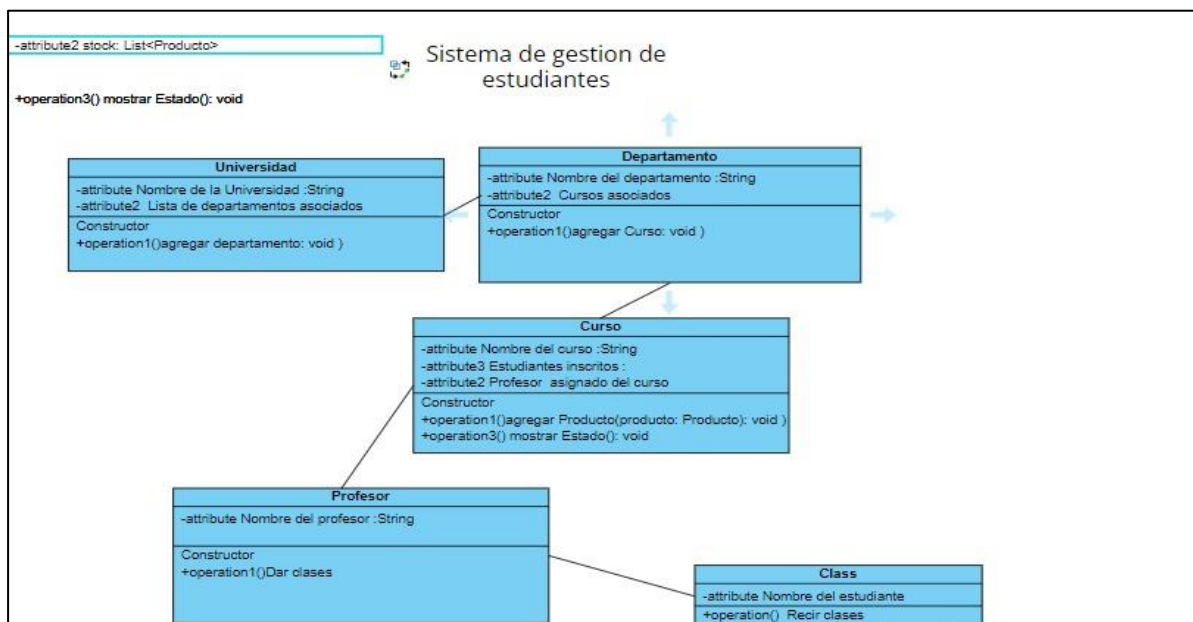
Descripción de la Actividad:

Diseñe 5 objetos diferentes con su correspondiente diagrama UML, asegurándose de mostrar las relaciones entre ellos.

Ejemplo:

- **Libro:** Representa los libros disponibles en una biblioteca, con atributos como título, autor, isbn, y métodos como prestar() y devolver().
- **Usuario:** Los usuarios pueden tener una lista de libros prestados. Se relaciona con Libro mediante composición.
- **Biblioteca:** Administra un catálogo de libros y se relaciona con Libro por asociación.
- **Empleado:** Gestiona usuarios y libros, y se relaciona con Usuario y Libro por asociación.
- **Transaccion:** Registra préstamos y devoluciones. Se relaciona con Usuario y Libro por agregación.

1. Sistema de Gestión de estudiantes.



Evidencia 1:

// Clase Universidad

```
class Universidad {  
    public String nombre;  
    public Departamento[] departamentos;  
  
    public Universidad(String nombre) {  
        this.nombre = nombre;  
        this.departamentos = new Departamento[0];  
    }  
  
    public void agregarDepartamento(Departamento departamento) {  
        Departamento[] nuevosDepartamentos = new  
Departamento[departamentos.length + 1];  
        System.arraycopy(departamentos, 0, nuevosDepartamentos, 0,  
departamentos.length);  
        nuevosDepartamentos[departamentos.length] = departamento;  
        departamentos = nuevosDepartamentos;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public Departamento[] getDepartamentos() {  
        return departamentos;  
    }  
}
```

// Clase Departamento

```
class Departamento {  
    public String nombre;  
    public Curso[] cursos;  
  
    public Departamento(String nombre) {  
        this.nombre = nombre;  
        this.cursos = new Curso[0];  
    }  
  
    public void agregarCurso(Curso curso) {  
        Curso[] nuevosCursos = new Curso[cursos.length + 1];
```



```
        System.arraycopy(cursos, 0, nuevosCursos, 0, cursos.length);
        nuevosCursos[cursos.length] = curso;
        cursos = nuevosCursos;
    }

    public String getNombre() {
        return nombre;
    }

    public Curso[] getCursos() {
        return cursos;
    }
}

// Clase Curso
class Curso {
    public String nombre;
    public Profesor profesor;
    public Estudiante[] estudiantes;

    public Curso(String nombre, Profesor profesor) {
        this.nombre = nombre;
        this.profesor = profesor;
        this.estudiantes = new Estudiante[0];
    }

    public void agregarEstudiante(Estudiante estudiante) {
        Estudiante[] nuevosEstudiantes = new Estudiante[estudiantes.length + 1];
        System.arraycopy(estudiantes, 0, nuevosEstudiantes, 0, estudiantes.length);
        nuevosEstudiantes[estudiantes.length] = estudiante;
        estudiantes = nuevosEstudiantes;
    }

    public String getNombre() {
        return nombre;
    }

    public Profesor getProfesor() {
        return profesor;
    }
}
```

```
public Estudiante[] getEstudiantes() {
    return estudiantes;
}
}
// Clase Profesor
class Profesor {
    private String nombre;

    public Profesor(String nombre) {
        this.nombre = nombre;
    }

    public String getNombre() {
        return nombre;
    }
}

// Clase Estudiante
class Estudiante {
    public String nombre;

    public Estudiante(String nombre) {
        this.nombre = nombre;
    }

    public String getNombre() {
        return nombre;
    }
}

// Ejemplo de uso
public class Main {
    public static void main(String[] args) {
        Universidad universidad = new Universidad("Universidad XYZ");
        Departamento departamentoCS = new Departamento("Ciencias de la
Computación");
        Profesor profesor = new Profesor("Dra. Martínez");
        Curso curso = new Curso("Programación 101", profesor);
        Estudiante estudiante = new Estudiante("Juan Pérez");
```

```
universidad.agregarDepartamento(departamentoCS);  
departamentoCS.agregarCurso(curso);  
curso.agregarEstudiante(estudiante);
```

```
System.out.println("Universidad: " + universidad.getNombre());  
System.out.println("Departamento: " + departamentoCS.getNombre());  
System.out.println("Curso: " + curso.getNombre());  
System.out.println("Profesor: " + profesor.getNombre());  
System.out.println("Estudiante: " + estudiante.getNombre());
```

```
}  
}
```

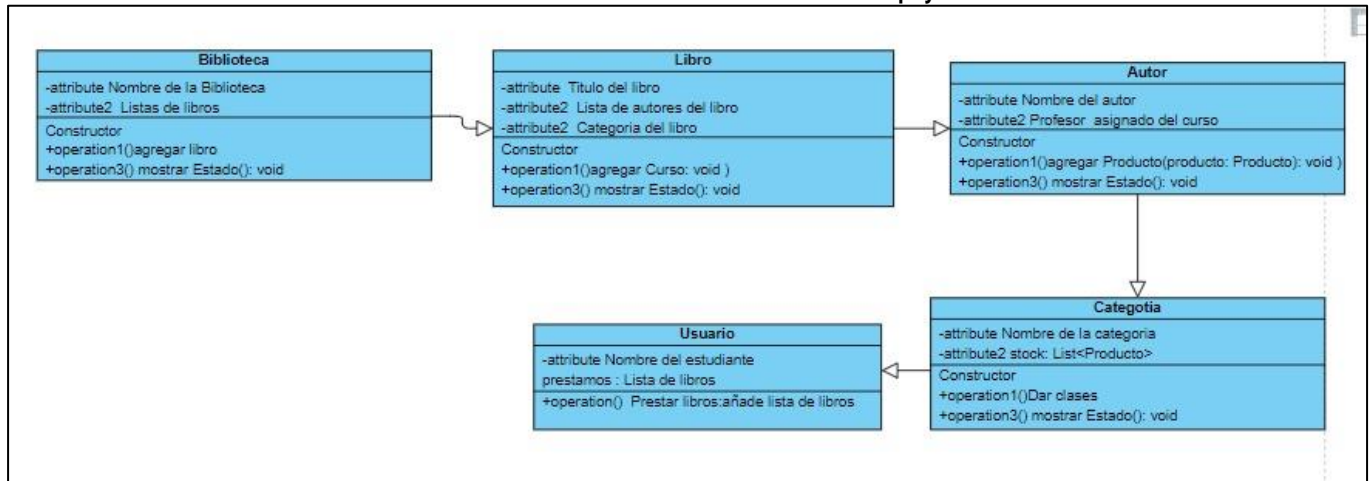
```
111 // Ejemplo de uso  
112 public class Main {  
113     public static void main(String[] args) {  
114         Universidad universidad = new Universidad("Universidad XYZ");  
115         Departamento departamentoCS = new Departamento("Ciencias de la Computación");  
116         Profesor profesor = new Profesor("Dra. Martínez");  
117         Curso curso = new Curso("Programación 101", profesor);  
118         Estudiante estudiante = new Estudiante("Juan Pérez");  
119  
120         universidad.agregarDepartamento(departamentoCS);  
121         departamentoCS.agregarCurso(curso);  
122         curso.agregarEstudiante(estudiante);  
123  
124         System.out.println("Universidad: " + universidad.getNombre());  
125         System.out.println("Departamento: " + departamentoCS.getNombre());  
126         System.out.println("Curso: " + curso.getNombre());  
127         System.out.println("Profesor: " + profesor.getNombre());  
128         System.out.println("Estudiante: " + estudiante.getNombre());  
129     }  
}
```

input

```
Universidad: Universidad XYZ  
Departamento: Ciencias de la Computación  
Curso: Programación 101  
Profesor: Dra. Martínez  
Estudiante: Juan Pérez
```

Activar Windows
Ve a Configuración para activar Windows.

2. Sistema de biblioteca



Evidencia:

// Clase Biblioteca

```
class Biblioteca {
    public String nombre;
    public Libro[] libros;

    public Biblioteca(String nombre) {
        this.nombre = nombre;
        this.libros = new Libro[0];
    }

    public void agregarLibro(Libro libro) {
        Libro[] nuevosLibros = new Libro[libros.length + 1];
        System.arraycopy(libros, 0, nuevosLibros, 0, libros.length);
        nuevosLibros[libros.length] = libro;
        libros = nuevosLibros;
    }

    public String getNombre() {
        return nombre;
    }

    public Libro[] getLibros() {
        return libros;
    }
}
```

// Clase Libro

```
class Libro {  
    public String titulo;  
    public Autor[] autores;  
    public Categoria categoria;  
  
    public Libro(String titulo, Autor[] autores, Categoria categoria) {  
        this.titulo = titulo;  
        this.autores = autores;  
        this.categoria = categoria;  
    }  
  
    public String getTitulo() {  
        return titulo;  
    }  
  
    public Autor[] getAutores() {  
        return autores;  
    }  
  
    public Categoria getCategoria() {  
        return categoria;  
    }  
}
```

// Clase Autor

```
class Autor {  
    public String nombre;  
  
    public Autor(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
}
```

// Clase Categoria

```
class Categoria {
```



```
public String nombre;

public Categoria(String nombre) {
    this.nombre = nombre;
}

public String getNombre() {
    return nombre;
}
}

// Clase Usuario
class Usuario {
    public String nombre;
    public Libro[] prestamos;

    public Usuario(String nombre) {
        this.nombre = nombre;
        this.prestamos = new Libro[0];
    }

    public void prestarLibro(Libro libro) {
        Libro[] nuevosPrestamos = new Libro[prestamos.length + 1];
        System.arraycopy(prestamos, 0, nuevosPrestamos, 0, prestamos.length);
        nuevosPrestamos[prestamos.length] = libro;
        prestamos = nuevosPrestamos;
    }

    public String getNombre() {
        return nombre;
    }

    public Libro[] getPrestamos() {
        return prestamos;
    }
}

// Clase principal para ejecutar
public class Main {
    public static void main(String[] args) {
```



```
Biblioteca biblioteca = new Biblioteca("Biblioteca Central");
Categoria categoria = new Categoria("Ciencia Ficción");
Autor autor1 = new Autor("Isaac Asimov");
Autor autor2 = new Autor("Arthur C. Clarke");
Libro libro = new Libro("Ficción Espacial", new Autor[]{autor1, autor2},
categoria);
Usuario usuario = new Usuario("Carlos López");
biblioteca.agregarLibro(libro);
usuario.prestarLibro(libro);

System.out.println("Biblioteca: " + biblioteca.getNombre());
System.out.println("Usuario: " + usuario.getNombre());
System.out.println("Libros prestados por el usuario: ");
for (Libro l : usuario.getPrestamos()) {
    System.out.println("Título: " + l.getTitulo());
}
}
```

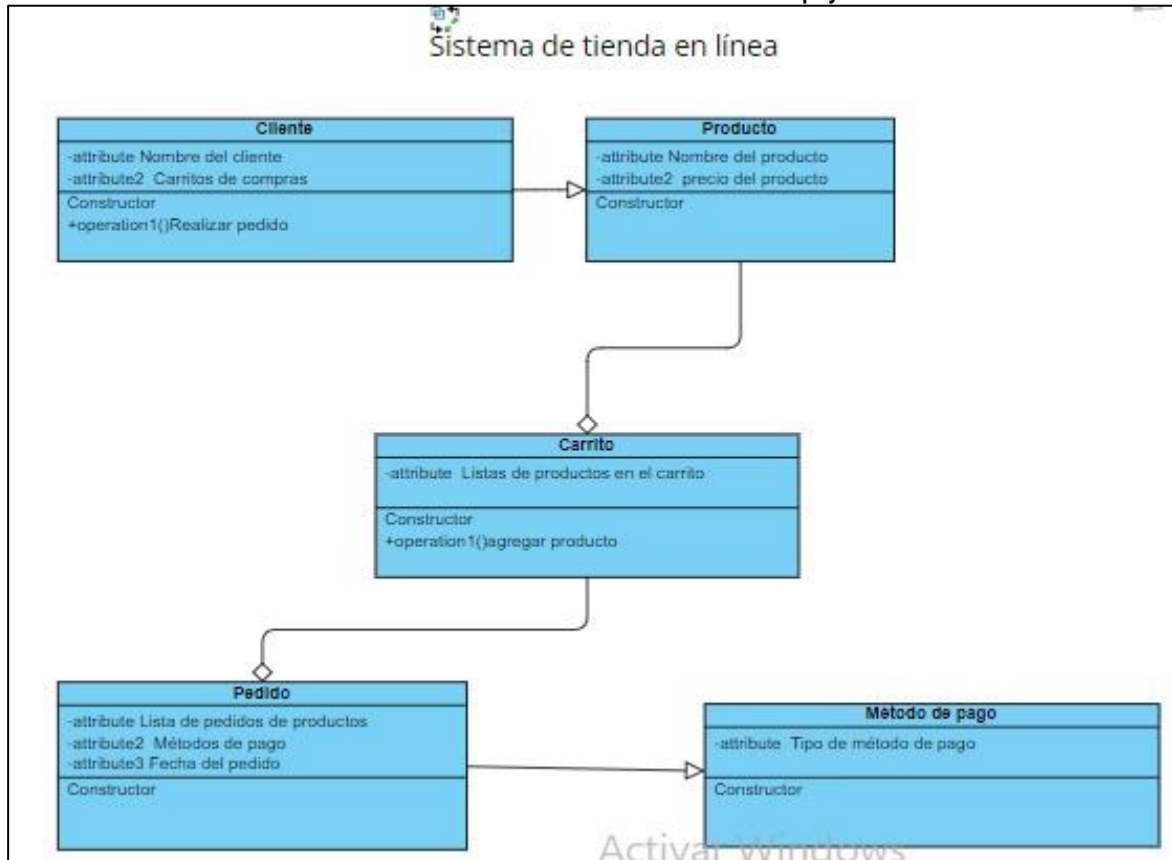
```
104 // Clase principal para ejecutar
105 public class Main {
106     public static void main(String[] args) {
107         Biblioteca biblioteca = new Biblioteca("Biblioteca Central");
108         Categoria categoria = new Categoria("Ciencia Ficción");
109         Autor autor1 = new Autor("Isaac Asimov");
110         Autor autor2 = new Autor("Arthur C. Clarke");
111         Libro libro = new Libro("Ficción Espacial", new Autor[]{autor1, autor2}, categoria);
112         Usuario usuario = new Usuario("Carlos López");
113         biblioteca.agregarLibro(libro);
114         usuario.prestarLibro(libro);
115
116         System.out.println("Biblioteca: " + biblioteca.getNombre());
117         System.out.println("Usuario: " + usuario.getNombre());
118         System.out.println("Libros prestados por el usuario: ");
119         for (Libro l : usuario.getPrestamos()) {
120             System.out.println("Título: " + l.getTitulo());
121         }
122     }
123 }
124
```

```
input
Usuario: Carlos López
Libros prestados por el usuario:
Título: Ficción Espacial

...Program finished with exit code 0
Press ENTER to exit console.
```

Activar Windows
Ve a Configuración para activar Windows.

3. Sistema en línea de una tienda



Evidencia:

```
// Clase Cliente
class Cliente {
    public String nombre;
    public Carrito carrito;

    public Cliente(String nombre) {
        this.nombre = nombre;
        this.carrito = new Carrito();
    }

    public Pedido realizarPedido(MetodoDePago metodoDePago) {
        return new Pedido(carrito.getProductos(), metodoDePago);
    }

    public Carrito getCarrito() {
        return carrito;
    }
}
```



```
// Clase Producto
class Producto {
    public String nombre;
    public double precio;

    public Producto(String nombre, double precio) {
        this.nombre = nombre;
        this.precio = precio;
    }

    public String getNombre() {
        return nombre;
    }

    public double getPrecio() {
        return precio;
    }
}

// Clase Carrito
class Carrito {
    public Producto[] productos;

    public Carrito() {
        this.productos = new Producto[0];
    }

    public void agregarProducto(Producto producto) {
        Producto[] nuevosProductos = new Producto[productos.length + 1];
        System.arraycopy(productos, 0, nuevosProductos, 0,
productos.length);
        nuevosProductos[productos.length] = producto;
        productos = nuevosProductos;
    }

    public Producto[] getProductos() {
        return productos;
    }
}
```



```
// Clase Pedido
class Pedido {
    public Producto[] productos;
    public MetodoDePago metodoDePago;
    public java.util.Date fecha;

    public Pedido(Producto[] productos, MetodoDePago metodoDePago) {
        this.productos = productos;
        this.metodoDePago = metodoDePago;
        this.fecha = new java.util.Date();
    }

    public Producto[] getProductos() {
        return productos;
    }

    public MetodoDePago getMetodoDePago() {
        return metodoDePago;
    }

    public java.util.Date getFecha() {
        return fecha;
    }
}

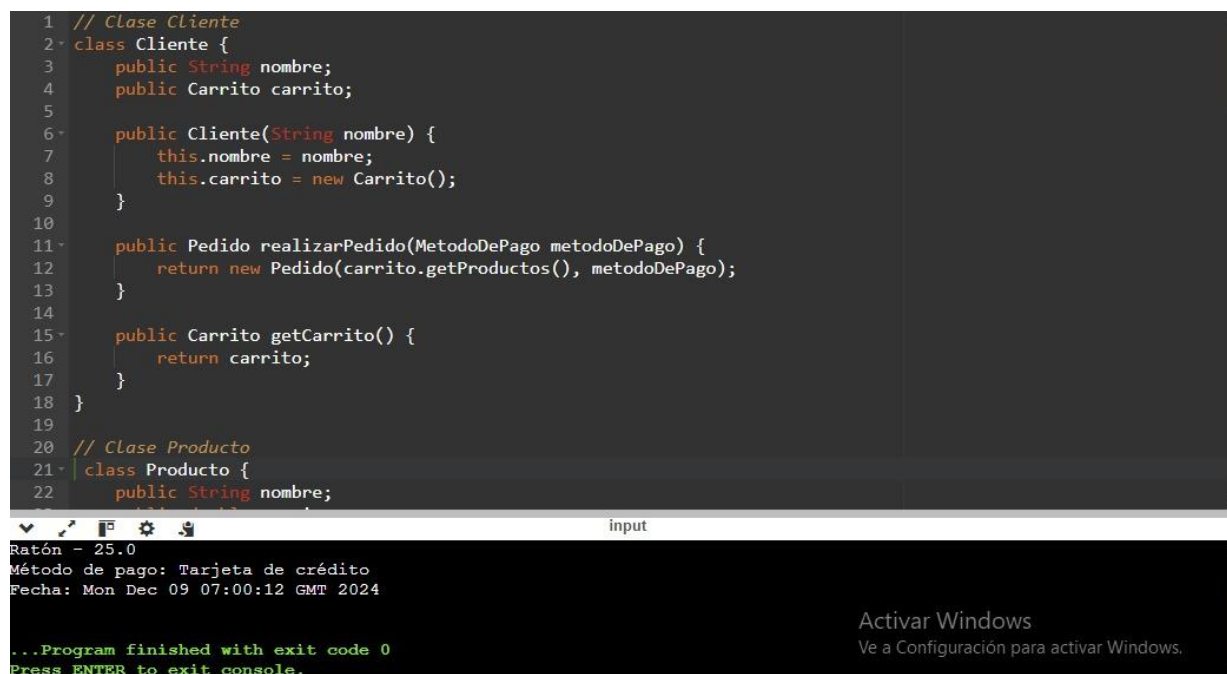
// Clase MetodoDePago
class MetodoDePago {
    public String tipo;

    public MetodoDePago(String tipo) {
        this.tipo = tipo;
    }

    public String getTipo() {
        return tipo;
    }
}

// Ejemplo de uso
class Main {
```

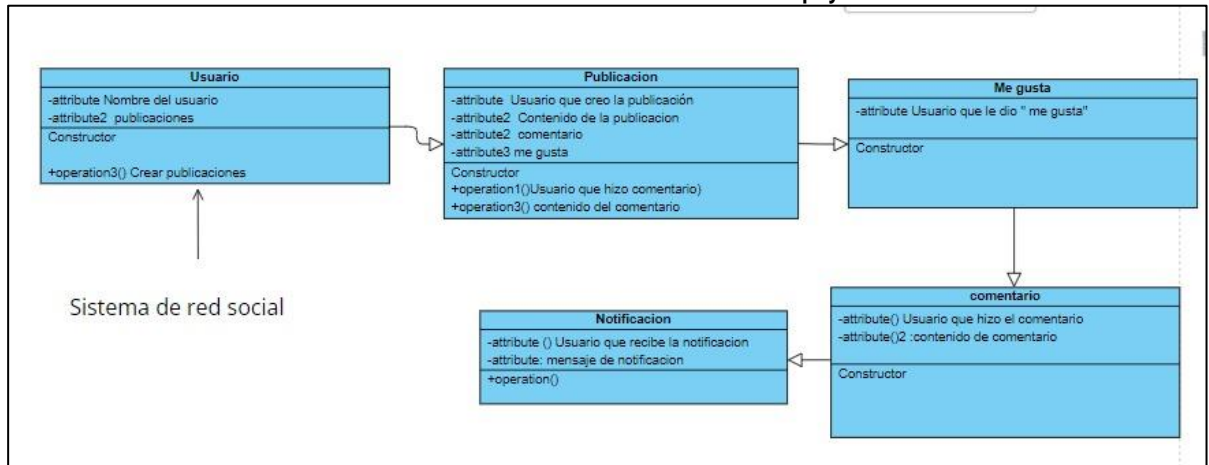
```
public static void main(String[] args) {  
    Cliente cliente = new Cliente("Lucía Gómez");  
    Producto producto1 = new Producto("Laptop", 1500);  
    Producto producto2 = new Producto("Ratón", 25);  
    MetodoDePago metodoPago = new MetodoDePago("Tarjeta de  
crédito");  
    cliente.getCarrito().agregarProducto(producto1);  
    cliente.getCarrito().agregarProducto(producto2);  
    Pedido pedido = cliente.realizarPedido(metodoPago);  
    System.out.println("Productos:");  
    for (Producto producto : pedido.getProductos()) {  
        System.out.println(producto.getNombre() + " - " +  
producto.getPrecio());  
    }  
    System.out.println("Método de pago: " +  
pedido.getMetodoDePago().getTipo());  
    System.out.println("Fecha: " + pedido.getFecha());  
}
```



```
1 // Clase Cliente  
2 class Cliente {  
3     public String nombre;  
4     public Carrito carrito;  
5  
6     public Cliente(String nombre) {  
7         this.nombre = nombre;  
8         this.carrito = new Carrito();  
9     }  
10  
11     public Pedido realizarPedido(MetodoDePago metodoDePago) {  
12         return new Pedido(carrito.getProductos(), metodoDePago);  
13     }  
14  
15     public Carrito getCarrito() {  
16         return carrito;  
17     }  
18 }  
19  
20 // Clase Producto  
21 class Producto {  
22     public String nombre;  
23     public double precio;  
24     public String descripcion;  
25 }
```

Ratón - 25.0
Método de pago: Tarjeta de crédito
Fecha: Mon Dec 09 07:00:12 GMT 2024
...Program finished with exit code 0
Press ENTER to exit console.

4. Sistema de red social



Evidencia:

// Clase Usuario

```
class Usuario {
    public String nombre;
    public Publicacion[] publicaciones;

    public Usuario(String nombre) {
        this.nombre = nombre;
        this.publicaciones = new Publicacion[0];
    }

    public Publicacion crearPublicacion(String contenido) {
        Publicacion publicacion = new Publicacion(this, contenido);
        Publicacion[] nuevasPublicaciones = new Publicacion[publicaciones.length +
1];
        System.arraycopy(publicaciones, 0, nuevasPublicaciones, 0,
publicaciones.length);
        nuevasPublicaciones[publicaciones.length] = publicacion;
        publicaciones = nuevasPublicaciones;
        return publicacion;
    }

    public String getNombre() {
        return nombre;
    }

    public Publicacion[] getPublicaciones() {
```



```
        return publicaciones;
    }
}

// Clase Publicación
class Publicacion {
    public Usuario usuario;
    public String contenido;
    public Comentario[] comentarios;
    public MeGusta[] meGustas;

    public Publicacion(Usuario usuario, String contenido) {
        this.usuario = usuario;
        this.contenido = contenido;
        this.comentarios = new Comentario[0];
        this.meGustas = new MeGusta[0];
    }

    public void agregarComentario(Comentario comentario) {
        Comentario[] nuevosComentarios = new Comentario[comentarios.length +
1];
        System.arraycopy(comentarios, 0, nuevosComentarios, 0,
comentarios.length);
        nuevosComentarios[comentarios.length] = comentario;
        comentarios = nuevosComentarios;
    }

    public void agregarMeGusta(MeGusta meGusta) {
        MeGusta[] nuevosMeGustas = new MeGusta[meGustas.length + 1];
        System.arraycopy(meGustas, 0, nuevosMeGustas, 0, meGustas.length);
        nuevosMeGustas[meGustas.length] = meGusta;
        meGustas = nuevosMeGustas;
    }

    public Usuario getUsuario() {
        return usuario;
    }

    public String getContenido() {
        return contenido;
    }
}
```

```
}

public Comentario[] getComentarios() {
    return comentarios;
}

public MeGusta[] getMeGustas() {
    return meGustas;
}
}

// Clase Comentario
class Comentario {
    public Usuario usuario;
    public String contenido;

    public Comentario(Usuario usuario, String contenido) {
        this.usuario = usuario;
        this.contenido = contenido;
    }

    public Usuario getUsuario() {
        return usuario;
    }

    public String getContenido() {
        return contenido;
    }
}

// Clase MeGusta
class MeGusta {
    public Usuario usuario;

    public MeGusta(Usuario usuario) {
        this.usuario = usuario;
    }

    public Usuario getUsuario() {
        return usuario;
    }
}
```



```
}  
}
```

// Clase Notificación

```
class Notificacion {  
    public Usuario usuario;  
    public String mensaje;  
  
    public Notificacion(Usuario usuario, String mensaje) {  
        this.usuario = usuario;  
        this.mensaje = mensaje;  
    }  
  
    public Usuario getUsuario() {  
        return usuario;  
    }  
  
    public String getMensaje() {  
        return mensaje;  
    }  
}
```

// Ejemplo de uso

```
class Main {  
    public static void main(String[] args) {  
        Usuario usuario1 = new Usuario("Pedro");  
        Usuario usuario2 = new Usuario("María");  
        Publicacion publicacion = usuario1.crearPublicacion("¡Hola, mundo!");  
        Comentario comentario = new Comentario(usuario2, "¡Qué buena  
publicación!");  
        MeGusta meGusta = new MeGusta(usuario2);  
        publicacion.agregarComentario(comentario);  
        publicacion.agregarMeGusta(meGusta);  
        System.out.println("Publicación:");  
        System.out.println("Usuario: " + publicacion.getUsuario().getNombre());  
        System.out.println("Contenido: " + publicacion.getContenido());  
        System.out.println("Comentarios:");  
        for (Comentario c : publicacion.getComentarios()) {  
            System.out.println("Usuario: " + c.getUsuario().getNombre());  
            System.out.println("Contenido: " + c.getContenido());  
        }  
    }  
}
```



```
}  
System.out.println("Me Gusta:");  
for (MeGusta mg : publicacion.getMeGustas()) {  
    System.out.println("Usuario: " + mg.getUsuario().getNombre());  
}  
}  
}
```

```
1 // Clase Usuario  
2 class Usuario {  
3     public String nombre;  
4     public Publicacion[] publicaciones;  
5  
6     public Usuario(String nombre) {  
7         this.nombre = nombre;  
8         this.publicaciones = new Publicacion[0];  
9     }  
10  
11     public Publicacion crearPublicacion(String contenido) {  
12         Publicacion publicacion = new Publicacion(this, contenido);  
13         Publicacion[] nuevasPublicaciones = new Publicacion[publicaciones.length + 1];  
14         System.arraycopy(publicaciones, 0, nuevasPublicaciones, 0, publicaciones.length);  
15         nuevasPublicaciones[publicaciones.length] = publicacion;  
16         publicaciones = nuevasPublicaciones;  
17         return publicacion;  
18     }  
19  
20     public String getNombre() {  
21         return nombre;  
22     }  
23 }
```

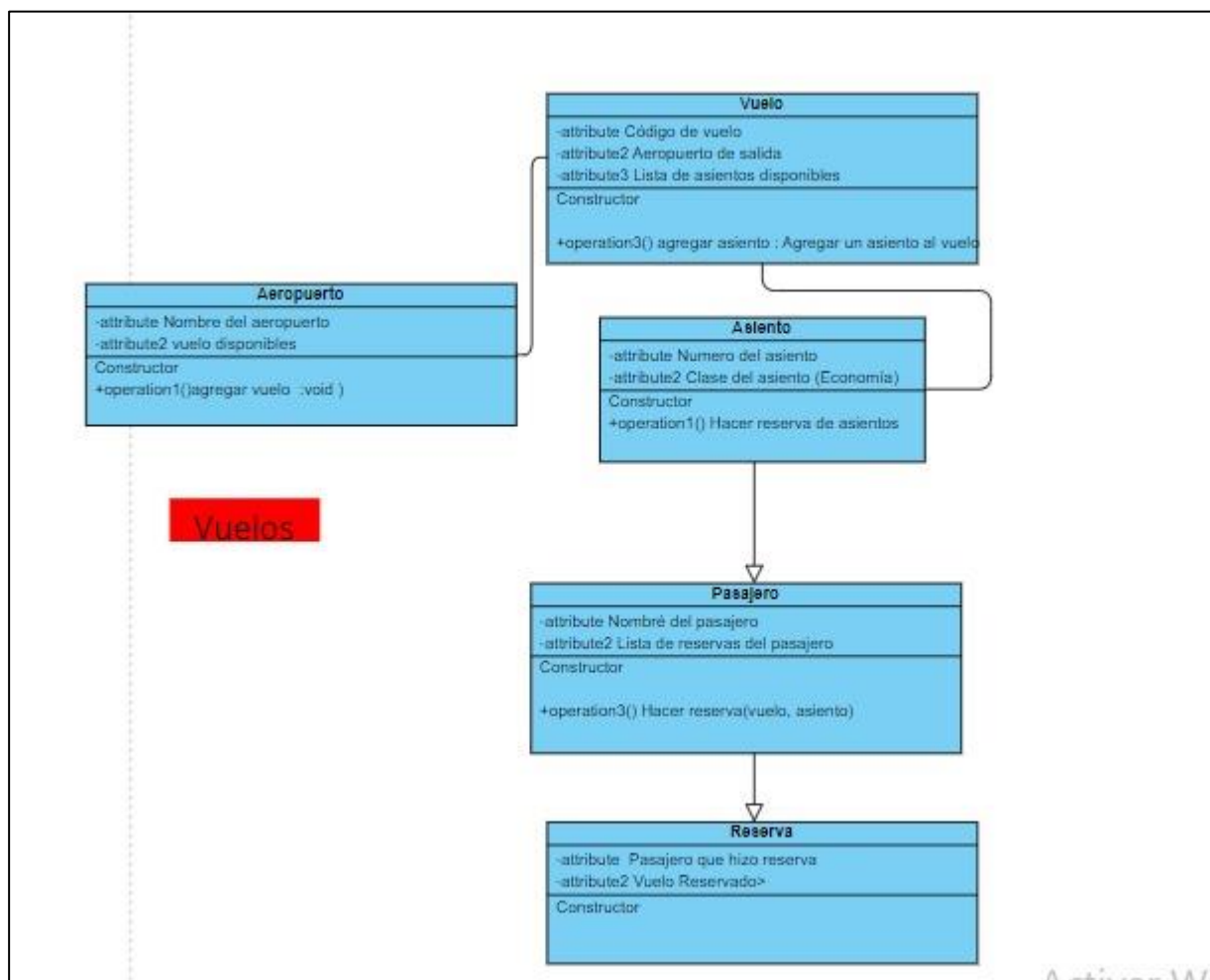
input

Contenido: ¡Qué buena publicación!
Me Gusta:
Usuario: María

...Program finished with exit code 0
Press ENTER to exit console.

Activar Windows
Ve a Configuración para activar Windows.

5. Sistema de reserva de vuelos



Evidencias:

```

// Clase Aeropuerto
class Aeropuerto {
    public String nombre;
    public Vuelo[] vuelos;

    public Aeropuerto(String nombre) {
        this.nombre = nombre;
        this.vuelos = new Vuelo[0];
    }
}
  
```

```
public void agregarVuelo(Vuelo vuelo) {
    Vuelo[] nuevosVuelos = new Vuelo[vuelos.length + 1];
    System.arraycopy(vuelos, 0, nuevosVuelos, 0, vuelos.length);
    nuevosVuelos[vuelos.length] = vuelo;
    vuelos = nuevosVuelos;
}

public String getNombre() {
    return nombre;
}

public Vuelo[] getVuelos() {
    return vuelos;
}
}

// Clase Vuelo
class Vuelo {
    public String codigo;
    public Asiento[] asientos;
    public Aeropuerto aeropuerto; // Agregar la variable de instancia aeropuerto

    public Vuelo(String codigo, Aeropuerto aeropuerto) {
        this.codigo = codigo;
        this.aeropuerto = aeropuerto;
        this.asientos = new Asiento[0];
    }

    public void agregarAsiento(Asiento asiento) {
        Asiento[] nuevosAsientos = new Asiento[asientos.length + 1];
        System.arraycopy(asientos, 0, nuevosAsientos, 0, asientos.length);
        nuevosAsientos[asientos.length] = asiento;
        asientos = nuevosAsientos;
    }

    public String getCodigo() {
        return codigo;
    }
}
```



```
public Aeropuerto getAeropuerto() {  
    return aeropuerto;  
}  
  
public Asiento[] getAsientos() {  
    return asientos;  
}  
}  
  
// Clase Asiento  
class Asiento {  
    public int numero;  
    public String clase;  
  
    public Asiento(int numero, String clase) {  
        this.numero = numero;  
        this.clase = clase;  
    }  
  
    public int getNumero() {  
        return numero;  
    }  
  
    public String getClase() {  
        return clase;  
    }  
}  
  
// Clase Pasajero  
class Pasajero {  
    public String nombre;  
    public Reserva[] reservas;  
  
    public Pasajero(String nombre) {  
        this.nombre = nombre;  
        this.reservas = new Reserva[0];  
    }  
  
    public Reserva hacerReserva(Vuelo vuelo, Asiento asiento) {  
        Reserva reserva = new Reserva(this, vuelo, asiento);
```



```
        Reserva[] nuevasReservas = new Reserva[reservas.length + 1];
        System.arraycopy(reservas, 0, nuevasReservas, 0, reservas.length);
        nuevasReservas[reservas.length] = reserva;
        reservas = nuevasReservas;
        return reserva;
    }

    public String getNombre() {
        return nombre;
    }

    public Reserva[] getReservas() {
        return reservas;
    }
}

// Clase Reserva
class Reserva {
    public Pasajero pasajero;
    public Vuelo vuelo;
    public Asiento asiento;

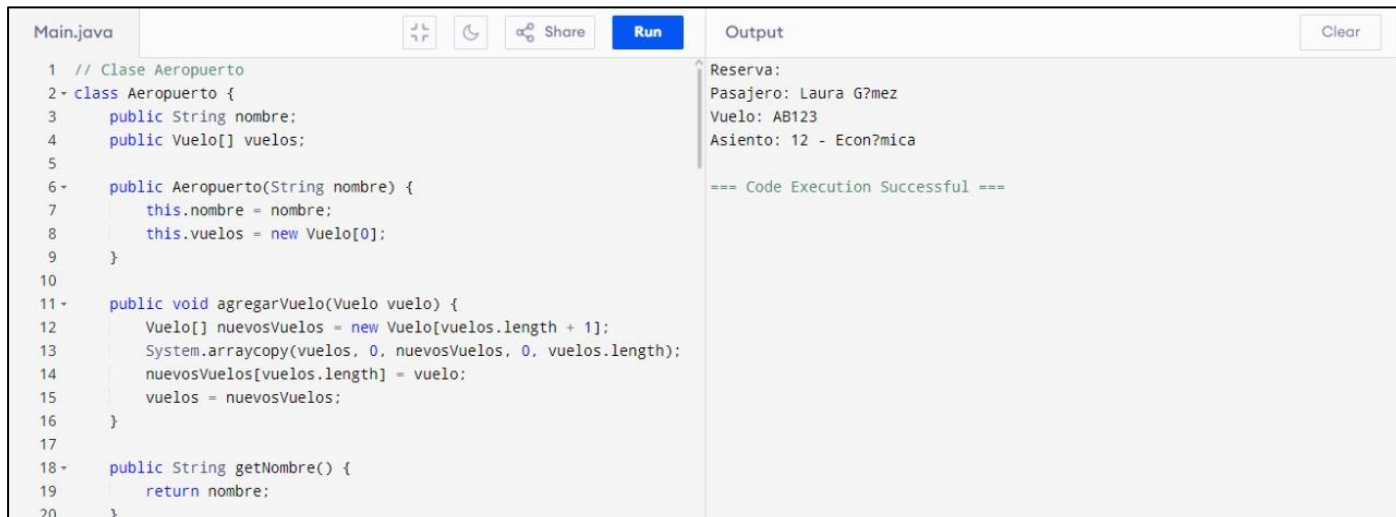
    public Reserva(Pasajero pasajero, Vuelo vuelo, Asiento asiento) {
        this.pasajero = pasajero;
        this.vuelo = vuelo;
        this.asiento = asiento;
    }

    public Pasajero getPasajero() {
        return pasajero;
    }

    public Vuelo getVuelo() {
        return vuelo;
    }

    public Asiento getAsiento() {
        return asiento;
    }
}
```

```
// Clase principal para ejecutar
public class Main {
    public static void main(String[] args) {
        Aeropuerto aeropuerto = new Aeropuerto("Aeropuerto Internacional");
        Vuelo vuelo = new Vuelo("AB123", aeropuerto);
        Asiento asiento = new Asiento(12, "Económica");
        Pasajero pasajero = new Pasajero("Laura Gómez");
        aeropuerto.agregarVuelo(vuelo);
        vuelo.agregarAsiento(asiento);
        Reserva reserva = pasajero.hacerReserva(vuelo, asiento);
        System.out.println("Reserva:");
        System.out.println("Pasajero: " + reserva.getPasajero().getNombre());
        System.out.println("Vuelo: " + reserva.getVuelo().getCodigo());
        System.out.println("Asiento: " + reserva.getAsiento().getNumero() + " - " +
reserva.getAsiento().getClase());
    }
}
```



```
Main.java
1 // Clase Aeropuerto
2 class Aeropuerto {
3     public String nombre;
4     public Vuelo[] vuelos;
5
6     public Aeropuerto(String nombre) {
7         this.nombre = nombre;
8         this.vuelos = new Vuelo[0];
9     }
10
11     public void agregarVuelo(Vuelo vuelo) {
12         Vuelo[] nuevosVuelos = new Vuelo[vuelos.length + 1];
13         System.arraycopy(vuelos, 0, nuevosVuelos, 0, vuelos.length);
14         nuevosVuelos[vuelos.length] = vuelo;
15         vuelos = nuevosVuelos;
16     }
17
18     public String getNombre() {
19         return nombre;
20     }
}
```

Output

```
Reserva:
Pasajero: Laura Gómez
Vuelo: AB123
Asiento: 12 - Económica

=== Code Execution Successful ===
```

Relaciones



Relaciones

- Sistema de gestión de estudiantes
 - Universidad: Lugar donde está ubicado y nombre del establecimiento
 - Departamento: Son cursos asociados de la universidad que administran las carreras.
 - Curso: Aula donde se recibirá a las clases de los maestros
 - Profesor: Autoridad que dará conocimientos en las aulas o clases
 - Estudiante: Alumno que recibirá clases en una aula de un profesor
- Sistema de biblioteca
 - Biblioteca: Lugar silencioso donde se encuentran espacios de silencio e información
 - Libro: Artículo físico que se encuentra en una biblioteca que es para leer.
 - Autor: Es la persona que escribió el libro que te otorga una fuente de información.
 - Categoría: Tipos diferentes de literatura como ciencia ficción entre otros
 - Usuario: Persona que llega a la biblioteca y adquiere un libro que más le guste.
- Sistema de tienda en línea.
 - Cliente: Persona que llega a comprar o adquirir algo.
 - Producto: Objeto que tiene que llamar la atención al cliente
 - Carrito: Esto es un lugar donde pones tus objetos que quieres llevarte a tu hogar
 - Pedido: La acción que el cliente hace para tener antes un objeto o sea reserva.
 - Método de pago: La acción que hace un cliente para pagar si de manera física o digital.
- Sistema de red social
 - Usuario: Persona que entra a crear una cuenta para página web.
 - Publicación: Usuario que publica una foto etc.
 - Me gusta: Usuario que otorga un like a una cuenta que no es de él.
 - Comentario: Usuario que da su opinión a una publicación
 - Notificación: Recibimiento que da la red social al teléfono.

Tema: Día: Mes: Año: ESTILO:

- Sistema vuelos en una aerolínea.

- Aeropuerto: lugar donde se van las personas a pedir un viaje a otros países
- Vuelo: Aviones que salen de los aeropuertos
- Asiento: Puestos que tiene el avión para transportar personas
- Pasajero: Persona que lleva sus cosas en el avión y tiene un asiento para viajar
- Reserva: Pasajero que compra ante de tiempo su boleto de viaje en el aeropuerto.

- Informe con evidencias 5.