



**Administración de Tecnologías de la Información**

**Programación Orientada a Objetos**

**Estudiante**

Caleb Segura Rodríguez - 2024105617 - GR 1

**AA5 Varios**

**Semestre 2 2025**

## Resultado de ejecutar el programa principal

Detalle del primer objeto Persona: Persona

Nombre: Nicolás Maduro

Identificacion: 666-6

Boleto asignado: #1

Detalle del segundo objeto Persona: Persona

Nombre: Donald Trump

Identificacion: 333-3

Boleto asignado: #2

Detalle del tercer objeto Persona: Persona

Nombre: Claudia Sheinbaum

Identificacion: 777-7

Boleto asignado: #3

Contador global de boletos creados: 3

Detalle de la Venta Del Día: VentaDelDia

Fecha: 2025-09-24

Cantidad de boletos: 3

Detalle:

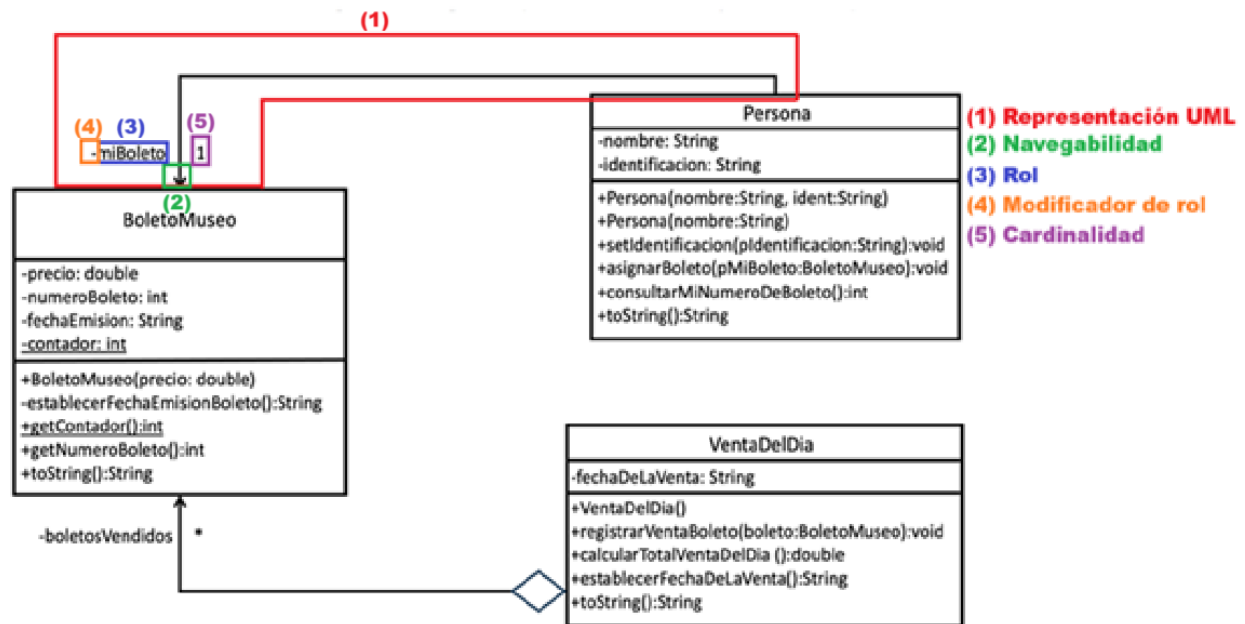
- Boleto #1 | 4500.0

- Boleto #2 | 6000.0

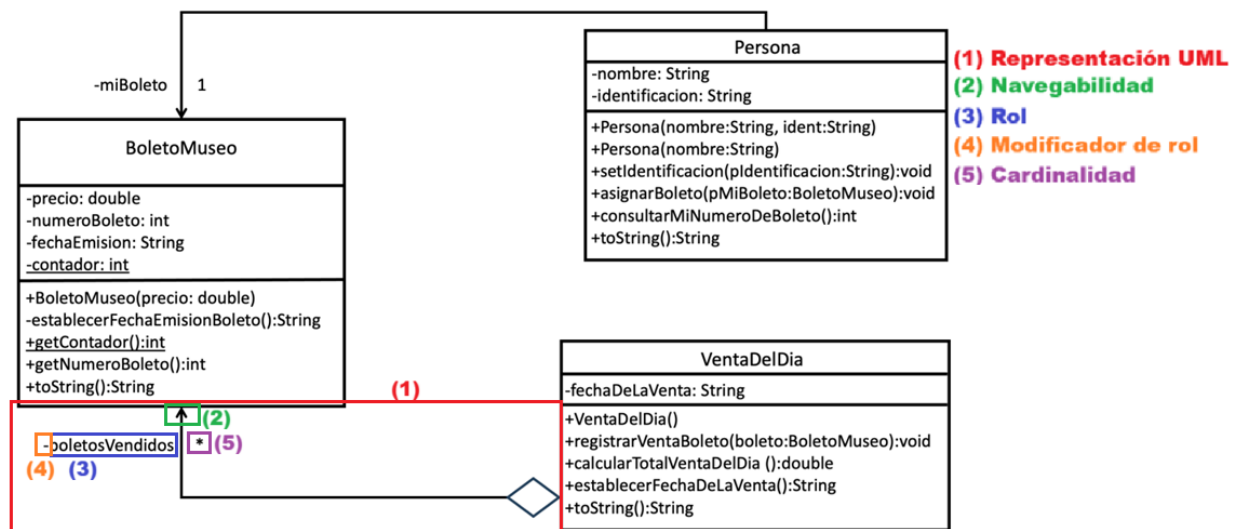
- Boleto #3 | 580.0

Total: 11080.0

## Identificación de Elementos (Asociación)



## Identificación de Elementos (Agregación)



## Respuestas a preguntas teóricas (a-e)

- a. Si la clase A está vinculada con la clase B mediante una relación de asociación. ¿La estructura de la clase B se ve impactada? Explique con detalle

R./ No, B no se ve afectada. Su estructura permanece igual, ya que no necesita almacenar ninguna referencia de los objetos de A.

- b. Si la clase P está vinculada con la clase Q mediante una relación de agregación. ¿La estructura de la clase P se ve impactada? Explique con detalle

R./Sí, la estructura de P se ve afectada porque, necesita la referencia de los objetos de Q. Por lo que debe tener atributos y métodos para gestionar estos objetos, y para ello modificar su estructura.

- c. Si la clase A está vinculada con la clase B mediante una relación de asociación y la clase B está vinculada con la clase A mediante una relación de asociación. ¿La estructura de ambas clases se ve impactada? Explique con detalle.

R./ Sí, ya que ambas clases depende de los objetos que tiene la otra clase. Por lo que deben de tener atributos y métodos para gestionar los objetos que están en la otra clase y para ello modificar su estructura.

- d. ¿Un objeto de tipo Z podría enviar mensajes a otro objeto de tipo W, aun cuando no exista un vínculo (de asociación o agregación) entre la clase Z y la clase W? Explique con detalle.

R./ No, ya que dos objetos de distintas clases de pueden enviar mensajes solo cuando estas clases estas relacionadas por un vínculo.

- e. En un diagrama de clase con detalles de implementación, suponga que existe una relación de asociación entre la clase P y la clase Q. Suponga también que esa relación tiene los cinco elementos respectivos en el diagrama. Es decir, la relación presenta todo el detalle de implementación posible. ¿Eso es suficiente para establecer de forma completa el vínculo de asociación entre P y Q? Explique con detalle.

R./ No, el diagrama representa un diseño del vínculo entre las clases. Haría falta definir el comportamiento de los métodos que gestionan la relación, y por ello quedaría incompleto el vínculo en la implementación.

## Reflexión de Aprendizaje

Entre mis principales aprendizajes, en primer lugar, aprendí como se implementa una relación entre objetos y su gestión por medio de métodos en java, más que todo en la codificación, ya que la mayoría de la lógica fue vista en clase.

Por ejemplo, en la clase persona, depende de los atributos de los objetos de la clase BoletoMuseo, en este caso era el boleto de cada persona instanciada en la clase y fue gestionada por dos métodos: uno para establecer el boleto y otro para consultar su número:

```
public class Persona
{
    /**
     * Nombre de la persona.
     */
    private String nombre;
    /**
     * Identificacion de la persona.
     */
    private String identificacion;
    /**
     * El boleto que posee la persona
     */
    private BoletoMuseo miBoleto;

    /**
     *Asigna un boleto de la clase BoletoMuseo a una persona.
     *@param pMiBoleto, boleto de la clase BoletoMuseo a asignar.
     */
    public void asignarBoleto(BoletoMuseo pMiBoleto) {
        miBoleto = pMiBoleto;
    }

    /**
     *Devuelve el numero de boleto que tiene la persona
     *@return numero de boleto que tiene la persona
     */
    public int consultarMiNumeroDeBoleto() {
        return miBoleto.getNumeroBoleto();
    }
}
```

También, en la otra clase se manejó una relación de clases, y fue entre la clase VentaDelDia y la clase BoletoMuseo, en este caso VentaDelDia tenía un atributo tipo ArrayList, en el cual se registran todos los objetos instanciados en la clase BoletoMuseo. Y fue manejada por dos métodos: uno para añadir los objetos al ArrayList y otro para calcular el precio total de todos los objetos del ArrayList.

```
public class VentaDelDia
{
    /**
     * fecha del dia de venta
     */
    private String fechaDeLaVenta;
    /**
     * lista de los boletos vendidos
     */
    private List<BoletoMuseo> boletosVendidos;

    /**
     * Registra un boleto en la lista de boletosVendidos
     * @param boleto de la clase BoletoMuseo
     */
    public void registrarVentaBoleto(BoletoMuseo boleto) {
        boletosVendidos.add(boleto);
    }

    /**
     * Calcula la suma del precio de todos los boletos vendidos
     * @return precio total de los boletos vendidos
     */
    public double calcularTotalVentaDelDia() {
        double total = 0.0;
        for (BoletoMuseo b : boletosVendidos) {
            total += b.getPrecio();
        }
        return total;
    }
}
```

Por otro lado, aprendí a como realizar una documentación mediante Javadoc. Fue bastante interesante como una documentación simple sobre clases, atributos y métodos, puede pasar a una documentación más técnica, detallada y representada en un archivo HTML:

Clases	
Class	Description
BoletoMuseo	Clase BoletoMuseo (capa de lógica de negocios) que representa al boleto de entrada a un museo, la cual contiene informacion como el precio, numero de boleto fecha de emision y un contador estatico de la cantidad de boletos inicializados
Persona	Clase Persona(capa de lógica de negocios) representa a la persona que posee un boleto de la clase BoletoMuseo, contiene informacion como el nombre de la persona, idetificacion y su boleto de la clase BoletoMuseo
PrincipalMuseo	
VentaDelDia	Clase VentaDelDia(capa de lógica de negocios) representa el registro de los boletos vendidos en el dia, contiene la fecha del dia de venta y la lista de boletos vendidos en este dia.

Y por último reforcé conceptos teóricos sobre la lógica de las relaciones entre clases, mediante preguntas y la identificación de elementos en los diagramas UML.