

**Instituto Tecnológico de Costa Rica**  
**Administración de Tecnologías de Información**

**Programación Orientada a Objetos**

**Grupo 02**

**Profesor**  
**Luis Javier Chavarría Sánchez**

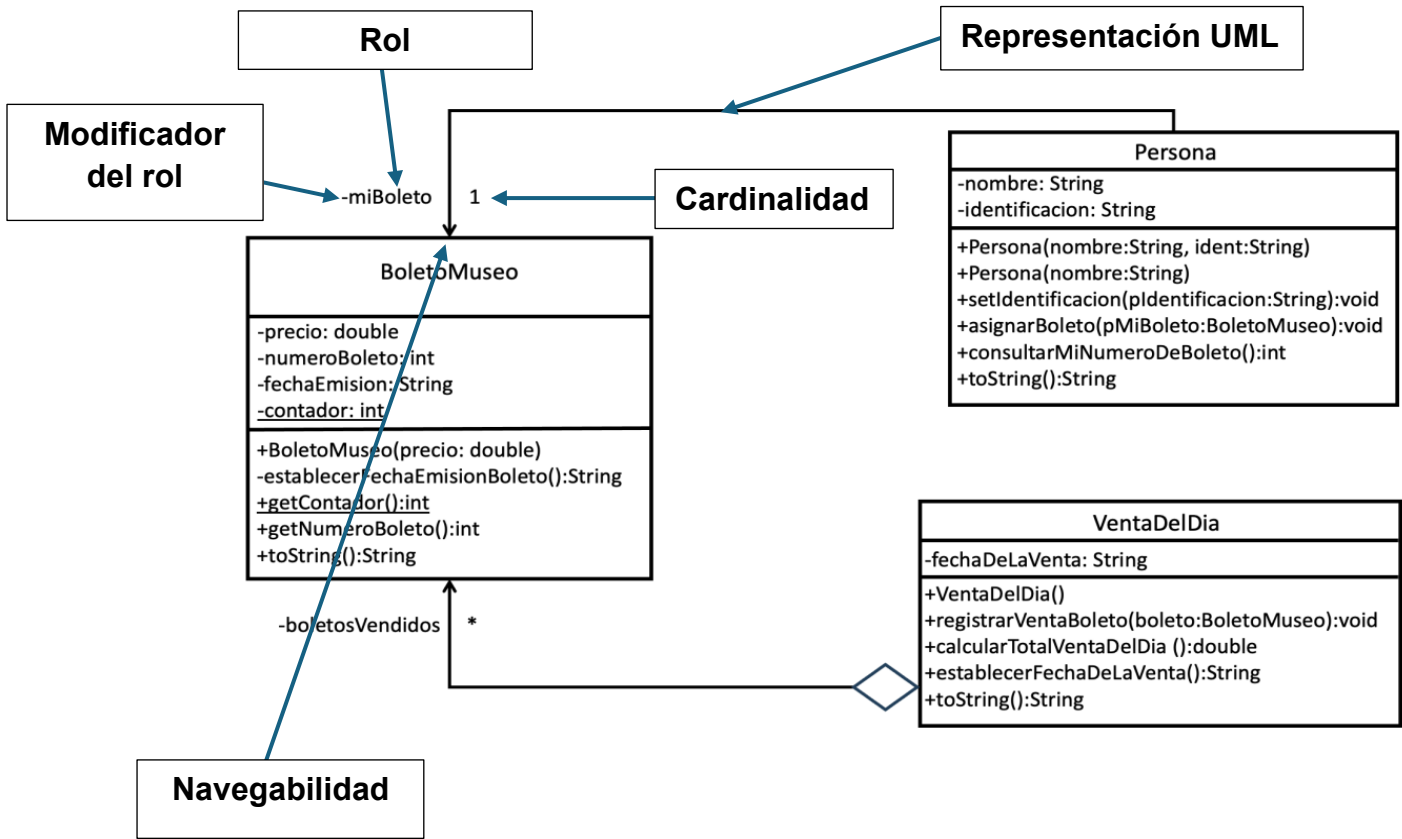
**Estudiante**  
**Esteban Brenes Montoya**

**II Semestre**  
**Septiembre, 2025**

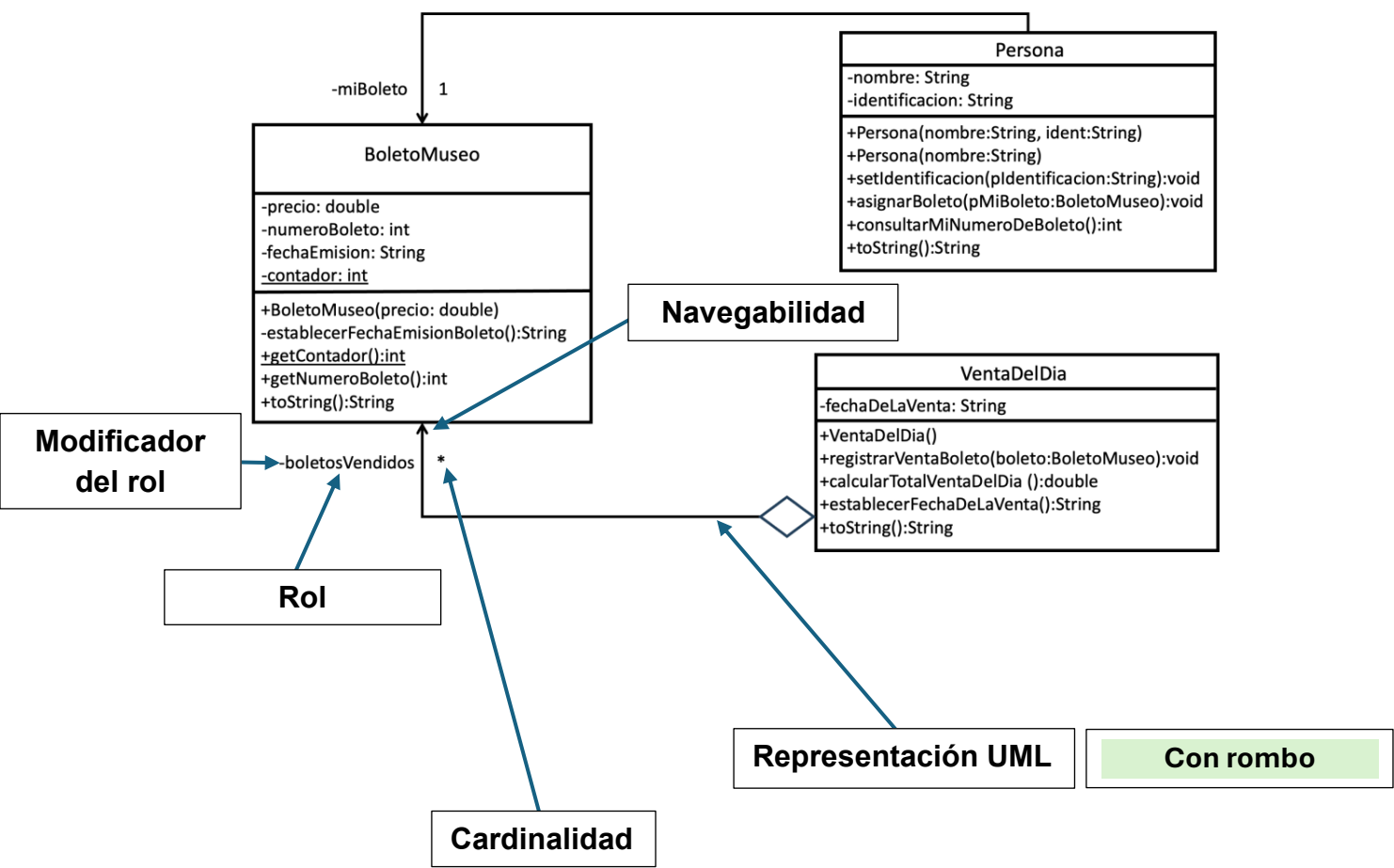
## E. Captura de pantalla de ejecutar el programa principal.

```
BlueJ: Ventana de Terminal - PrjMuseo
Opciones
Nombre: Claudia Sheinbaum
Identificacion: 777-7
Boleto asignado: #3
Contador global de boletos creados: 3
Detalle de la Venta Del Día:
VentaDelDia
Fecha: 2025-09-23
Cantidad de boletos: 3
Detalle:
- Boleto #1 | 4500.0
- Boleto #2 | 6000.0
- Boleto #3 | 5800.0
Total: 16300.0
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: 2025-09-23
Cantidad de boletos: 3
Detalle:
- Boleto #1 | 4500.0
- Boleto #2 | 6000.0
- Boleto #3 | 5800.0
Total: 16300.0
Can only enter input while your program is running
```

H. Relación de Asociación:



I. Relación de Agregación:



## **J. Respuestas a preguntas teóricas**

**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?**

**R/** No, la clase B mantiene su estructura original. En una asociación, solo la clase que contiene la referencia (A) se modifica. Por ejemplo, en nuestro proyecto, Persona tiene un atributo miBoleto, pero BoletoMuseo no sabe que existe Persona. La asociación es unidireccional y solo afecta a una de las clases.

**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?**

**R/** Sí, la clase P debe modificarse para contener la referencia a Q. En la agregación, P actúa como el "todo" y necesita un atributo para almacenar las "partes". En nuestro código, VentaDelDia tiene el atributo boletosVendidos para poder agregar los boletos. Sin este atributo, no existiría la relación.

**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?**

**R/** Sí, ambas clases se ven impactadas. Cada una debe contener un atributo que reference a la otra clase. Esto crea una asociación bidireccional donde A conoce a B y B conoce a A, aumentando el acoplamiento entre ambas clases.

**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?**

**R/** No, es imposible. Para que Z pueda enviar mensajes a W, debe tener una referencia a W. Esta referencia se establece mediante los vínculos estructurales. Sin un vínculo, Z no puede acceder a los métodos de W.

**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. ¿Eso es suficiente para establecer de forma completa el vínculo de asociación entre P y Q?**

**R/** Sí, es suficiente. Los cinco elementos (representación UML, navegabilidad, rol, modificador y cardinalidad) definen completamente cómo implementar la asociación en código. Indican la dirección, el nombre del atributo, su visibilidad y la multiplicidad, dejando claro exactamente qué código escribir.

## **L. Reflexión sobre lo aprendido en esta actividad:**

Esta actividad me permitió comprender profundamente las relaciones entre clases en el paradigma orientado a objetos. La diferencia práctica entre asociación y agregación ahora es clara: mientras la asociación conecta clases independientes, la agregación representa una relación todo-parte donde el todo contiene las partes pero estas pueden existir independientemente.

La implementación de Javadoc fue especialmente valiosa, ya que aprendí a documentar código de manera profesional, generando documentación automática similar a la API de Java. Esto mejora la mantenibilidad del software y facilita el trabajo en equipo.

El ejercicio de identificar y etiquetar los elementos UML reforzó mi comprensión de la notación, especialmente la importancia de la navegabilidad y cardinalidad en el diseño. La corrección del profesor sobre la pregunta J.b. me hizo reflexionar sobre la importancia de leer cuidadosamente los enunciados.

Finalmente, la integración de GitHub en el flujo de trabajo académico me prepara para prácticas profesionales reales donde el control de versiones es esencial. Esta actividad consolidó mis conocimientos de POO y me dio herramientas prácticas para diseñar sistemas más complejos.

**Bibliografía:**

<https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>

<https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>