

# Práctica 4: Creación de clases y objetos

## Objetivo

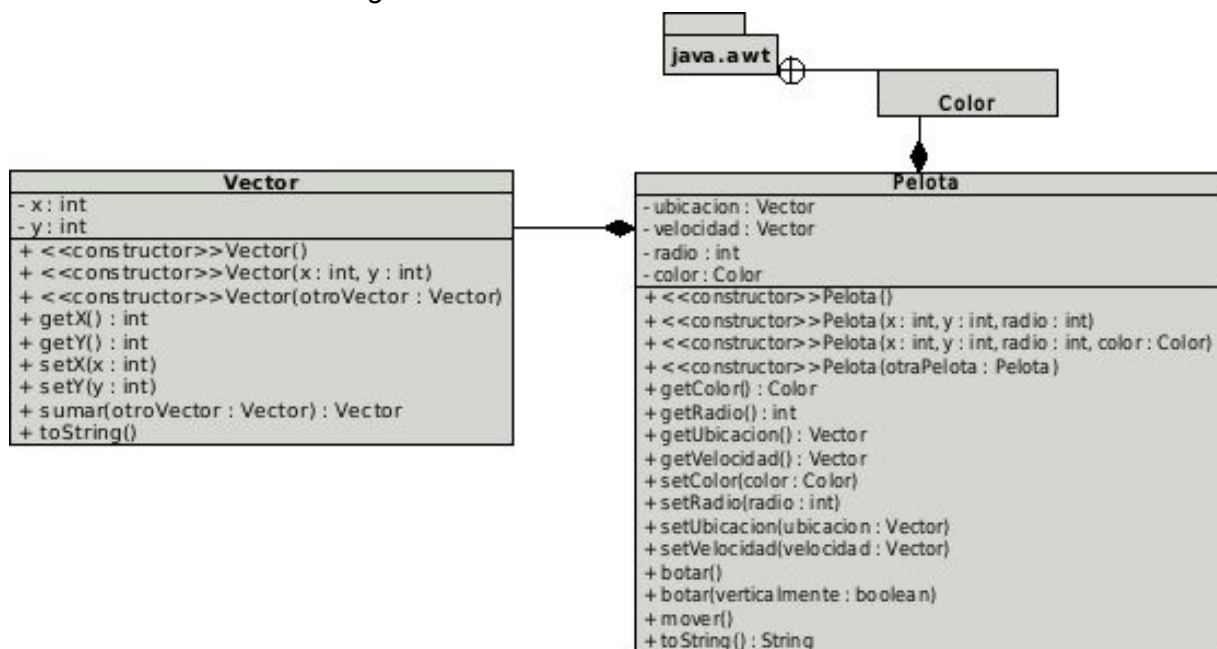
Crear objetos a partir de clases y manipular sus propiedades.

## Desarrollo

Descargar el proyecto de NetBeans JuegoPelota.

Una vez descargado puedes verificar que existen tres paquetes distintos. Los paquetes no son más que una estructura de directorios en la que puedes organizar tus proyectos (tal como la estructura de directorios que tienes para organizar tus materias).

Crearás o modificarás las siguientes clases:



Color es una clase que se encuentra en el paquete `java.awt`, no la debes crear.

**Sugerencia:** Los métodos `mover` y `botar` escríbelos mientras solucionas los ejercicios, de esta forma irás añadiendo funcionalidad poco a poco.

### Requerimientos pelota.

1. La clase **Pelota** deberá estar ubicada en el paquete `mx.unam.fciencias.core`.
2. Los valores por defecto (aquellos que no se especifiquen explícitamente) que deberá tener toda **Pelota** son:
  - a. `ubicación = (25, 25)`
  - b. `velocidad = (0, 0)`
  - c. `radio = 20`
  - d. `color = Color.BLACK` (`Color.BLACK` es una constante que se refiere a un objeto `Color` cuyo color es negro, para inicializar el atributo se hace de la forma: `color = Color.BLACK`)

3. Todos los constructores que tengan algún parámetro deberán respetar el parámetro introducido y no usar el valor por defecto mencionado en el punto 1.
4. Cada componente de la ubicación de la Pelota siempre deberá estar en el intervalo [0, 1024]. Ojo, Esto no significa que un Vector se restrinja a estos valores.
5. radio siempre debe estar en el intervalo [5, 100].
6. velocidad puede tener valores negativos.
7. botar() sólo imprimirá el mensaje “Botando desde [**x**, **y**]”, donde **x** es la x del vector ubicacion y **y** es la y del vector ubicacion. Debe ser exactamente ese mensaje, respetando espacios, mayúsculas y minúsculas. Lo único que varía en el mensaje es **x** y **y**.
8. botar(boolean), si verticalmente tiene el valor true quiere decir que se debe cambiar el sentido en el componente y del vector velocidad, de otra manera se tendrá que cambiar el sentido en el componente x del mismo vector.
9. botar(boolean) deberá llamar a botar() en cualquier caso.
10. mover() mueve la ubicación de la pelota de acuerdo al vector velocidad (se suman los vectores y el resultante es la nueva ubicación).

### Requerimientos de Vector.

1. La clase Vector deberá estar ubicada en el paquete *mx.unam.fciencias.core*.
2. sumar(Vector) suma el vector que llama al método con el vector del parámetro. El vector resultante debe regresarse.
3. Al sumar vectores, ninguno de ellos deberá ser modificado, se deberá crear uno nuevo como resultado de la operación.

Para cada uno de los siguientes ejercicios deberás crear una clase con el nombre del ejercicio en el paquete *mx.unam.fciencias.principal*.

### Ejercicio1

Crear una pelota con cada constructor. En el caso de aquella creada por el constructor por copia mover la copia 10 unidades a la derecha y cambiar su color, esto para poderlos diferenciar.

### Ejercicio2

Escribe el siguiente código:

```
Pelota pelota = new Pelota(100, 50, 20);  
Pelota otraPelota = pelota;  
otraPelota.getUbicacion().setY(200);  
otraPelota.setColor(Color.MAGENTA);
```

Agrega ambas pelotas al panel. Al visualizar la escena ¿qué puedes observar? ¿Existe algún problema y de ser así cómo lo arreglarías?

### Ejercicio3

Crea una Pelota en la posición y con el color que desees. Establece una velocidad (0, 2). Haz las validaciones necesarias en el método verificarColision (clase VerificadorColisiones) de modo que la pelota bote en los bordes inferior y superior cuando estos son tocados.

#### Ejercicio4

Crea una Pelota en la posición y con el color que desees. Establece una velocidad (2, 0). Haz las validaciones necesarias en el método verificarColision (clase VerificadorColisiones) de modo que la pelota bote en los bordes izquierdo y derecho cuando estos son tocados.

#### Ejercicio5

Crea una Pelota en la posición y con el color que desees. Establece una velocidad (2, 3). Verifica si tienes que hacer alguna validación extra de modo que la pelota siempre se encuentre botando en los bordes y se mantenga dentro del área.

#### Ejercicio6

Crea al menos tres pelotas distintas con ubicaciones, radios, colores y velocidades diferentes. Observalas en acción ¿ves que esto divertido?

#### ***Estableciendo el ambiente.***

Para agregar una pelota se usa el método agregar(Pelota) de PanelPelota. En cada uno de tus programas harás lo siguiente:

```
VentanaPelota ventana = new VentanaPelota();  
PanelPelota panel = ventana.getPanelPelota();
```

Esas sentencias iniciarán el ambiente en el que vivirán las Pelotas. panel es la referencia al panel donde se pintan las pelotas, de modo que para agregar una pelota puedes escribir:

```
panel.agregar(pelota);
```

asumiendo que la variable pelota tenga una referencia a un objeto de tipo Pelota.

#### *Consideraciones*

Habrás **penalizaciones** si no se siguen siguientes aspectos:

- La práctica deberá ser entregada individualmente.
- Deberán mantener una correcta indentación.
- **No** se deben incluir archivos .class en la entrega (Da clic derecho en el proyecto y después en clean).
- El nombre de su clase (public class ...) debe iniciar con **mayúscula**
- Se deberán poner comentarios (//, /\* \*/), estos deberán incluir los nombres de los autores de cada clase, cuál es el objetivo de la clase que están programando, una breve explicación (tan breve como se posible, puede ser uen unas cuantas palabras) de las partes complicadas, etc.
- Se deberán seguir las indicaciones mostradas abajo respecto a la entrega.

## *Entregables*

- Se deberá crear un comprimido con una carpeta nombrada P4\_ + ambos apellidos del autor de la práctica en notación camello y **sin acentos**. Ejemplo:
  - P3\_ContrerasBarajas. (El comprimido deberá tener el mismo nombre)
- Dentro de esta carpeta incluirán:
  - Reporte en formato PDF donde se explique brevemente cómo se resolvieron los problemas incluyendo una captura de la ejecución del programa así como las respuestas de las preguntas.
  - El proyecto de NetBeans resultante.
- Subir el comprimido a la plataforma classroom y entregarlo.
- El asunto para dudas deberá ser **DUDA** a cualquiera de los 3 correos
  - Diego [nemesio@ciencias.unam.mx](mailto:nemesio@ciencias.unam.mx)
  - Armando [armando.orozco@ciencias.unam.mx](mailto:armando.orozco@ciencias.unam.mx)
  - Rogelio [rogelio.mendez.rivera@gmail.com](mailto:rogelio.mendez.rivera@gmail.com)

## *Fecha de entrega*

Martes 28 de abril de 2020