

## TRABAJO PRÁCTICO N° 2 – Clases y Objetos

PROGRAMACIÓN II - 2025 – 2do cuatrimestre

TECNICATURA UNIVERSITARIA EN DESARROLLO WEB

### Objetivos

- Aplicar y reforzar los conceptos fundamentales relacionados a la Programación Orientada a Objetos.
- Interpretar y traducir correctamente diagramas de clases en código Python.

### Condiciones de Entrega

- Deberá realizarse una entrega grupal, con grupos de no más de 3 (tres) alumnos.
- Los ejercicios a entregar son los correspondientes a la sección “B”. La entrega deberá consistir de una carpeta comprimida que contenga un archivo con extensión .py por cada clase creada (cancion.py y circulo.py) y un tercer archivo main.py donde se realice el resto de los ejercicios que no implican crear o alterar la definición de una clase. Se deberá indicar con claridad la sección del código que se corresponde con cada uno de los ejercicios del documento. La carpeta debe ser subida en la sección correspondiente del Campus.
- Se deberá respetar la fecha de entrega, la misma será informada en el campus.
- El Trabajo Práctico será calificado como **Aprobado** o **Desaprobado**.
- Las soluciones del grupo deberán ser de autoría propia. De encontrarse soluciones idénticas entre diferentes grupos, dichos trabajos prácticos

serán calificados como **Desaprobado**, lo cual será comunicado en la devolución.

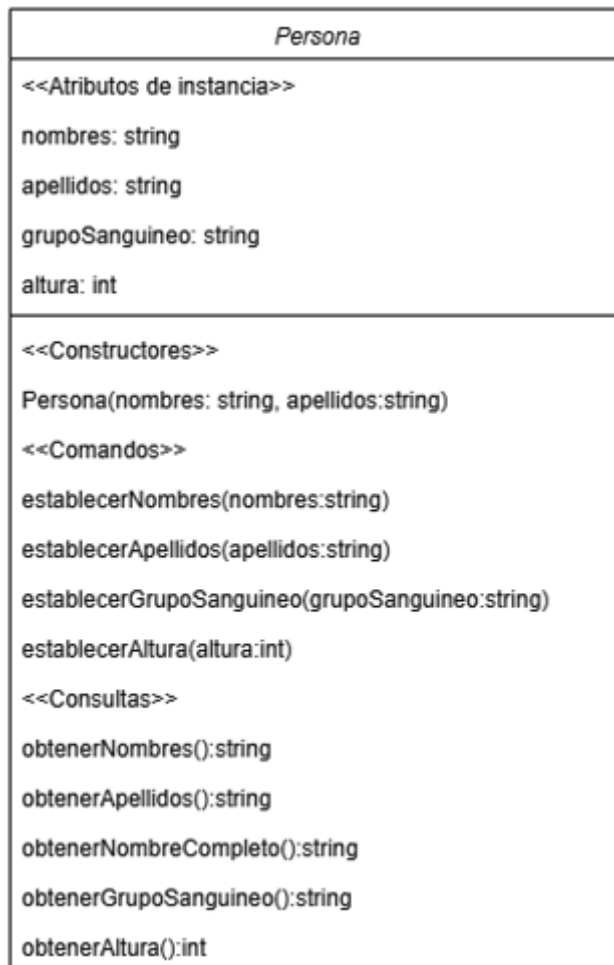
- Las entregas individuales serán calificadas como **Desaprobado**.

## EJERCICIOS

*Nota: asumir que todos los parámetros recibidos en los métodos correspondientes a los ejercicios del trabajo son del tipo de datos correcto, con excepción de los casos donde se especifique la necesidad de agregar validaciones.*

**Sección A** (para ver en clase):

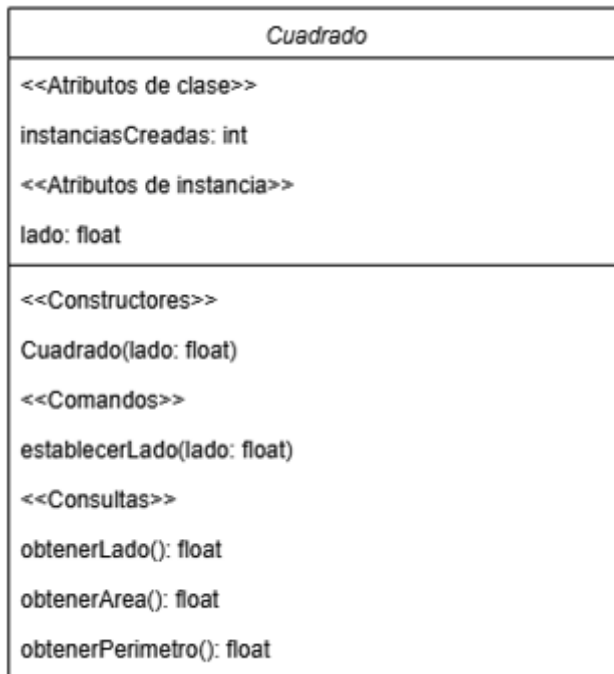
- 1) En el archivo persona.py, generar la clase correspondiente al siguiente diagrama:



*Nota: la altura está medida en centímetros*

- 2) En el archivo main.py, instanciar la clase “Persona” 4 veces indicando en cada caso apellidos y los nombres.
- 3) En el archivo main.py, definir valores para los atributos “grupoSanguineo” y “altura” en 2 de las instancias creadas en el punto 2.

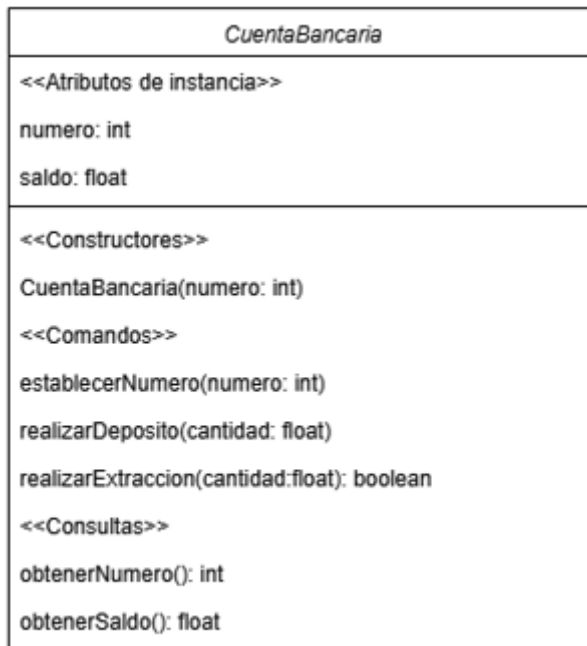
- 4) En el archivo main.py, imprimir los nombres completos de cada instancia creada de la clase “Persona” utilizando el método creado para ese fin.
- 5) En el archivo cuadrado.py, generar la clase correspondiente al siguiente diagrama:



*Nota: el lado está medido en centímetros*

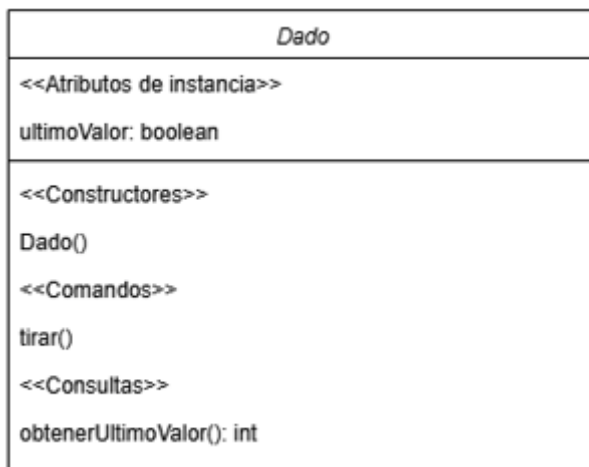
- 6) Asegurar que el constructor incremente en una unidad el valor de la variable de clase “instanciasCreadas” cada vez que se utiliza el mismo.
- 7) En el archivo main.py, instanciar la clase “Cuadrado” 3 veces.
- 8) En el archivo main.py, imprimir el valor del perímetro para cada instancia creada de la clase “Cuadrado”.
- 9) En el archivo main.py, imprimir la cantidad de instancias creadas para la clase “Cuadrado”.
- 10) En el archivo main.py, crear 2 instancias más que tengan valores idénticos para el lado, e imprimir el resultado de compararlas utilizando el operador ==.
- 11) En el archivo main.py, imprimir el resultado de comparar los valores del lado de cada instancia creada en el punto anterior.

- 12) En el archivo `cuenta_bancaria.py`, generar la clase correspondiente al siguiente diagrama:



- 13) Agregar una validación al método “realizarExtraccion” que cause que el mismo retorne False y cancele la operación en caso de que la cantidad a extraer sea mayor al saldo disponible. Caso contrario, la función debe retornar True.
- 14) En el archivo `main.py`, instanciar la clase “CuentaBancaria” 2 veces.
- 15) En el archivo `main.py`, utilizar el método “realizarDeposito” en una de las instancias creadas en el punto anterior.
- 16) En el archivo `main.py`, utilizar el método “realizarExtraccion” en la instancia utilizada en el punto anterior con una cantidad válida y después imprimir el saldo de la cuenta.

17) En el archivo dado.py, generar la clase correspondiente al siguiente diagrama:



*Nota: El dado puede generar valores del 1 al 6*

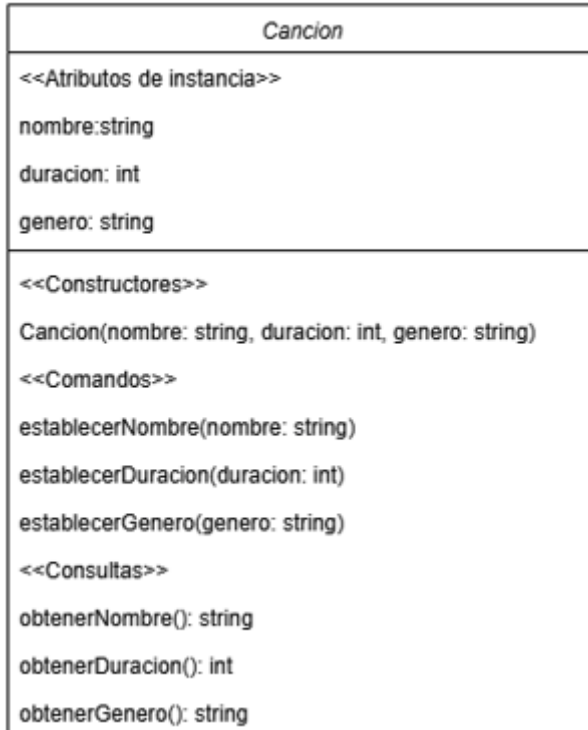
18) En el archivo main.py, instanciar la clase “Dado” 1 vez

19) En el archivo main.py, ejecutar el método “tirar” de la instancia creada en el punto anterior y después imprimir el valor obtenido utilizando el método correspondiente

20) En el archivo main.py, ejecutar nuevamente el método tirar de la instancia creada en el punto anterior y después imprimir el valor obtenido utilizando el método correspondiente

**Sección B** (para entregar):

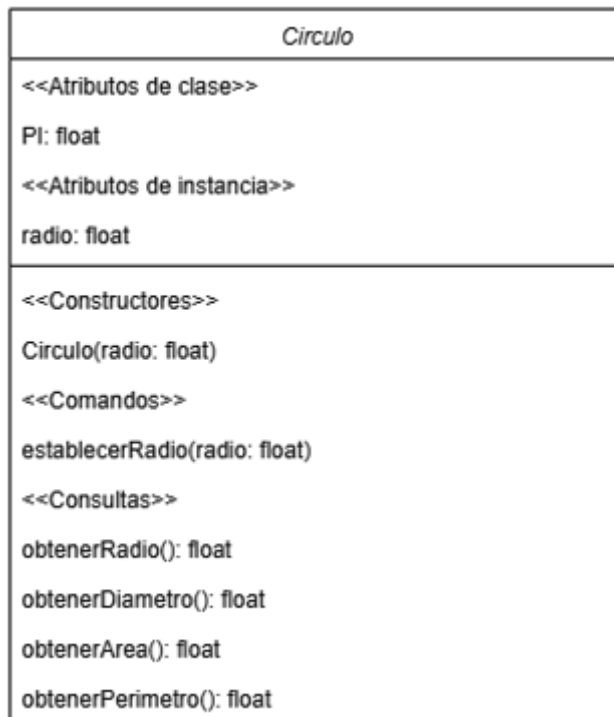
- 1) En el archivo `cancion.py`, generar la clase correspondiente al siguiente diagrama:



*Nota: la duración está medida en segundos*

- 2) En el archivo `main.py`, instanciar la clase “Cancion” 3 veces.
- 3) En el archivo `main.py`, imprimir el valor del atributo `genero` para cada instancia creada de la clase “Cancion”.
- 4) En el archivo `main.py`, modificar el valor del atributo `genero` de una de las instancias de “Cancion” e imprimir nuevamente su valor.

- 5) En el archivo `circulo.py`, generar la clase correspondiente al siguiente diagrama:



*Nota: el radio está medido en centímetros*

- 6) En el archivo `main.py`, instanciar la clase “Circulo” 3 veces.
- 7) En el archivo `main.py`, imprimir el valor del diámetro para cada instancia de “Circulo” creada.
- 8) En el archivo `main.py`, imprimir el valor del atributo PI para cada instancia de “Circulo” creada.
- 9) En el archivo `main.py`, crear 2 instancias más de “Circulo” que tengan valores idénticos para el radio, e imprimir el resultado de compararlas utilizando el operador `==`.
- 10) En el archivo `main.py`, imprimir el resultado de comparar los valores del perímetro de cada instancia creada en el punto anterior.