**Alumno:** Olivarez Alexis Emanuel  **DNI**: 44.542.230 **Laboratorio**: 2

**Profesores:** Medina – Airaldi **TP**: 3 **Fecha de Entrega**: 10/09/23

En este trabajo practico trabajamos nuevos conceptos teóricos/prácticos útiles para la programación orientada a objetos. En la mayoría de los ejercicios se dio conocimientos y colaboración entre objetos para poder llevar acabo las acciones que establecían las actividades.

**Conceptos aprendidos y utilizados en el Trabajo Practico N° 3:**

**Colaboración entre objetos:** En la mayoría de los ejercicios se dio, un objeto necesita la colaboración de otros objetos. Por ejemplo, las clases **Rectángulo** y Circulo que para calcular distancias entre objetos tenían que utilizar a la clase **Punto** y su método “distanciaA”.

**JavaDoc:** Tipo de documentación externa que sirve para documentar código y hacerlo más legible y entendible para los demás programadores. Comienza con los caracteres /\*\* y termina con \*/ Ejemplo:

/\*\*

*\* Constructor de la clase CuentaDeAhorro.*

*\**

*\* @param pNroCuenta El número de cuenta de la persona titular de la*

*\*                   cuenta de ahorro.*

*\* @param pSaldo     Saldo al iniciar la cuenta de ahorro.*

*\* @param pTitular   Persona titular de la cuenta de ahorro.*

\*/

**Formas de conocimientos entre objetos:** Es fundamental tener presente que para que un objeto le envié mensajes a otro, debe conocerlo.

* **Parámetros:** Pueden ser de cualquier tipo de dato (Objeto, String, int, double …). En cuanto al conocimiento, se utiliza para nombrar objetos que el receptor necesita para llevar a cabo una instrucción o una serie de instrucciones. Este conocimiento se establece dentro del cuerpo del método y se mantiene durante el tiempo que el método se encuentra activo.
* **Variables temporales:** Se usan en un método y cuando termina de ejecutarse, se libera el espacio en memoria de estas.
* **Pseudo-variables:** Es un mecanismo que permite al objeto poder nombrarse y enviarse mensajes a si mismo.
* **Variables de instancias:** Se relacionan con una única instancia de clase, cada vez que se instancia un nuevo objeto el sistema proporciona una copia de todas las variables que están vinculadas a dicha clase, haciéndolas propias de esa instancia (objeto). Pueden ser de distintos tipos. Acompañan al objeto hasta que “se muere”.

**Desarrollo para reutilización:** Reutilizamos código de trabajos anteriores, con la importación de una clase ya creada (como la clase Laboratorio del Trabajo Practico N° 2).