

Alexanyer Naranjo (@alexanyernas)

Disfruta del aprendizaje 

## Guía Práctica N°6

### Programación Orientada a Objetos (POO)

1. Realizar un programa que conste de una clase llamada **Alumno** que tenga como atributos el **nombre** y la **nota del alumno**. Definir los métodos para inicializar sus atributos (método constructor), imprimirlos y mostrar un mensaje con el resultado de la nota y si ha aprobado o no.
2. Realizar una clase que administre una **agenda**. Se debe almacenar para cada contacto el **nombre**, el **teléfono** y el **email**. Además, deberá mostrar un menú con las siguientes opciones:
  - Añadir contacto
  - Lista de contactos
  - Buscar contacto
  - Editar contacto
  - Cerrar agenda
3. Desarrollar un programa que conste de una **clase padre Cuenta** y **dos subclases PlazoFijo y CajaAhorro**. Definir los atributos **titular** y **cantidad** y un método para imprimir los datos en la clase Cuenta. La clase **CajaAhorro** tendrá un método para heredar los datos y uno para mostrar la información. La clase **PlazoFijo** tendrá dos atributos propios, **plazo** e **interés**. Tendrá un método para obtener el importe del interés ( $\text{cantidad} \times \text{interés} / 100$ ) y otro método para mostrar la información; datos del titular plazo, interés y total de interés. Crear al menos un objeto de cada subclase.

4. Defina una clase **Monedero** que permita gestionar la cantidad de dinero que una persona dispone en un momento dado. La clase deberá tener un constructor que permitirá crear un monedero con una cantidad de dinero inicial y deberá definir un método para meter dinero en el monedero, otro para sacarlo y finalmente, otro para consultar el disponible; solo podrá conocerse la cantidad de dinero del monedero a través de este último método. Por supuesto, no se podrá sacar más dinero del que haya en un momento dado en el monedero.
5. Defina una clase punto que tendrá dos atributos, de tipo Real  $x$  e  $y$ , que representarán las coordenadas del punto dentro del plano. Defina un método que tenga como argumento otro objeto de la clase punto y que calcule la distancia entre los dos puntos. Un punto puede estar ubicado en cualquier parte del sistema de coordenada. Puede definir otros métodos que considere necesarios.