

# INTRODUCCIÓN A LA PROGRAMACIÓN

## Laboratorio 05

ING. MARTIN POLIOTTO

Docente a cargo del módulo

Agosto 2020



DIPLOMATURA EN

**NUEVAS  
TECNOLOGÍAS**

SECRETARÍA DE  
EXTENSIÓN  
UNIVERSITARIA

**SEU**

**UTN**  
Facultad Regional Córdoba

Ministerio de  
PROMOCIÓN DEL EMPLEO  
Y DE LA ECONOMÍA FAMILIAR

Ministerio de  
CIENCIA Y  
TECNOLOGÍA

**CBA**  
ENTRE TODOS

Oficina de Montevideo  
Organización  
de las Naciones Unidas  
para la Educación,  
la Ciencia y la Cultura

### Practica 01:

Crear una clase Libro que contenga los siguientes atributos:

- ISBN
  - Título
  - Autor
  - Número de páginas
- 
- Crear sus respectivos métodos get y set correspondientes para cada atributo. Crear el método toString() para mostrar la información relativa al libro con el siguiente formato:  
    “El libro con ISBN creado por el autor tiene páginas”
  - En una clase Ejecutable (que contenga main()), crear 2 objetos Libro (los valores que se quieran) y mostrarlos por pantalla. Por último, indicar cuál de los 2 tiene más páginas.

### Practica 02:

Se nos ha solicitado desarrollar una clase que permita modelar un punto en el plano. Cada punto se representa mediante un par ordenado de coordenadas cartesianas y es posible conocer:

- cuál es su distancia al eje de coordenadas
- el cuadrante en el que se encuentra ubicado. Para esto último considerar que si el punto tiene una de sus coordenadas con valor cero, el punto se considera sobre los ejes.
- Adicionalmente se necesita mostrar un punto como una cadena con formato “(x;y)”.
- Se necesita realizar una clase Ejecutable para crear un punto, y mostrar: su estado, el cuadrante donde se ubica y la distancia al origen.

*Practica 03:*

Desarrollar una clase java llamada Persona que siga las siguientes condiciones:

- Sus atributos son: nombre, edad, DNI, sexo (H hombre, M mujer), peso y altura. No queremos que se accedan directamente a ellos. Por defecto, todos los atributos menos el DNI serán valores según su tipo (0 números, cadena vacía para String, etc.). Sexo será hombre por defecto, usa una constante para ello.
- Se implantaran varios constructores:
  - ✓ Un constructor por defecto.
  - ✓ Un constructor con el nombre, edad y sexo, el resto por defecto.
  - ✓ Un constructor con todos los atributos como parámetro.
- Los métodos que se implementaran son:
  - ✓ calcularIMC(): calculara si la persona está en su peso ideal (peso en kg/(altura<sup>2</sup> en m)), si esta fórmula devuelve un valor menor que 20, la función devuelve un -1, si devuelve un número entre 20 y 25 (incluidos), significa que esta por debajo de su peso ideal la función devuelve un 0 y si devuelve un valor mayor que 25 significa que tiene sobrepeso, la función devuelve un 1. Te recomiendo que uses constantes para devolver estos valores.
  - ✓ esMayorDeEdad(): indica si es mayor de edad, devuelve un booleano.
  - ✓ comprobarSexo(char sexo): comprueba que el sexo introducido es correcto. Si no es correcto, sera H. No sera visible al exterior.
  - ✓ toString(): devuelve toda la información del objeto.
  - ✓ generaDNI(): genera un número aleatorio de 8 cifras, genera a partir de este su número su letra

correspondiente. Este método sera invocado cuando se construya el objeto. Puedes dividir el método para que te sea más fácil. No será visible al exterior.

- ✓ Métodos set de cada parámetro, excepto de DNI.

Adicionalmente crear una clase Ejecutable que haga lo siguiente:

- Pide por teclado el nombre, la edad, sexo, peso y altura.
- Crea 3 objetos de la clase anterior, el primer objeto obtendrá las anteriores variables pedidas por teclado, el segundo objeto obtendrá todos los anteriores menos el peso y la altura y el último por defecto, para este último utiliza los métodos set para darle a los atributos un valor.
- Para cada objeto, deberá comprobar si está en su peso ideal, tiene sobrepeso o por debajo de su peso ideal con un mensaje.
- Indicar para cada objeto si es mayor de edad.
- Por último, mostrar la información de cada objeto.

*Practica 04:*

Desarrollar una clase llamada Password que siga las siguientes condiciones:

- Que tenga los atributos longitud y contraseña. Por defecto, la longitud sera de 8.
- Los constructores serán los siguiente:
  - ✓ Un constructor por defecto.
  - ✓ Un constructor con la longitud que nosotros le pasemos. Generará una contraseña aleatoria con esa longitud.
- Los métodos que implementa serán:
  - ✓ esFuerte(): devuelve un booleano si es fuerte o no, para que sea fuerte debe tener mas de 2 mayúsculas, mas de 1 minúscula y mas de 5 números.
  - ✓ generarPassword(): genera la contraseña del objeto con la longitud que tenga.
  - ✓ Método get para contraseña y longitud.
  - ✓ Método set para longitud.

Ahora, crea una clase ejecutable llamada Main:

- Crea un array de Passwords con el tamaño que tu le indiques por teclado, junto con la longitud de los passwords (todos tendrán la misma).
- Crea otro array de booleanos donde se almacene si el password del array de Password es o no

Al final, muestra la contraseña y si es o no fuerte (usa el bucle anterior).

Usa este simple formato:

- contraseña1 valor\_booleano1
- contraseña2 valor\_bololeano2

*Practica 05:*

Realizar un programa que permita modelar el funcionamiento básico de un automóvil, tal como se indica a continuación:

- **conducir:** esta opción debe recibir la cantidad de kilómetros a recorrer e informar si pueden ser recorridos y en tal caso descontar el combustible utilizado del tanque (se conoce que, por especificaciones técnicas del vehículo, con un litro recorre 11 km). Caso contrario informar que no hay combustible suficiente para el recorrido indicado.
- **cargar combustible:** esta opción debe reponer combustible en el tanque tantos litros como se reciban como argumento pero teniendo en cuenta que el tanque tiene una capacidad de 49 lts más una reserva de 5 lts., en el caso de que los litros informados superen la capacidad del tanque, devolver la cantidad de lts de combustible derramados.
- **chequear nivel de combustible:** informar si el tanque está al 25%, 50%, 75% o lleno y dar una alerta en el caso que el tanque se encuentre en reserva, indicando que hay que reponer combustible de inmediato. El mensaje debe devolver una cadena con 25% ... o el alerta.

Ayuda: para una resolución más sencilla, pensar en la existencia de dos clases: Automovil y Tanque.

*Practica 06:*

Vamos a desarrollar una baraja de cartas españolas orientado a objetos.

Una carta tiene un número entre 1 y 12 (el 8 y el 9 no los incluimos) y un palo (espadas, bastos, oros y copas)

La baraja estará compuesta por un conjunto de cartas, 40 exactamente.

Las operaciones que podrá realizar la baraja son:

- **barajar:** cambia de posición todas las cartas aleatoriamente
- **siguienteCarta:** devuelve la siguiente carta que está en la baraja, cuando no haya más o se haya llegado al final, se indica al usuario que no hay más cartas.
- **cartasDisponibles:** indica el número de cartas que aún puede repartir
- **darCartas:** dado un número de cartas que nos pidan, le devolveremos ese número de cartas (piensa que puedes devolver). En caso de que haya menos cartas que las pedidas, no devolveremos nada pero debemos indicárselo al usuario.
- **cartasMonton:** mostramos aquellas cartas que ya han salido, si no ha salido ninguna indicárselo al usuario
- **mostrarBaraja:** muestra todas las cartas hasta el final. Es decir, si se saca una carta y luego se llama al método, este no mostrara esa primera carta.