

## Ejercicios | Informatorio | Mentoría

Comisión 3

Practicamos Herencia y Polimorfismo

### 1 – Herencia en la Creación de Clases {

En este ejercicio, vas a practicar el concepto de **herencia** en Python. Crearás una clase base llamada *Persona* y luego, crearás una clase hija llamada *Estudiante* que herede de *Persona*. El objetivo es reutilizar el código de la clase base y extenderlo con funcionalidades adicionales en la clase hija

#### Requerimientos:

##### 1- Clase Persona:

- \* Debe tener dos atributos: nombre y apellido.
- \* Debe tener un método nombre\_completo() que retorne el nombre completo de la persona, en formato “nombre apellido”

##### 2- Clase Estudiante:

- \* Debe heredar de la clase Persona
- \* Debe agregar un atributo adicional carrera que almacene la carrera que está estudiando el estudiante.
- \* Debe tener un método obtener\_carrera() que retorne un mensaje indicando la carrera que está estudiando el estudiante. El formato del mensaje debe ser: “Está estudiando la carrera de {carrera}”.

##### 3- Ejemplo de uso:

- \* Crea una instancia de la clase Persona y llama al método nombre\_completo() para mostrar su nombre completo.
- \* Crea una instancia de la clase Estudiante y llama a los métodos nombre\_completo() y obtener\_carrera() para mostrar tanto el nombre completo del estudiante como la carrera que está estudiando.

}

### 2- Polimorfismo – Vehículo {

En este ejercicio, vamos a practicar el concepto de **polimorfismo** en Python. Crearás una clase base llamada *Vehiculo*, y luego crearás varias clases derivadas que implementarán el método mover() de manera diferente, dependiendo del tipo de vehículo.

#### Requerimientos:

##### 1- Clase Vehiculo:

- \* Debe tener un método mover(), pero este método debe estar vacío (puede estar definido con pass). Las clases hijas implementarán este método de manera específica.

## 2- Clases derivadas de Vehiculo

\* Clase Coche:

Debe sobrescribir el método mover() para que imprima: “El coche está conduciendo por la carretera”

\* Clase Bicicleta:

Debe sobrescribir el método mover() para que imprima: “La bicicleta está pedaleando”

\* Clase Avion:

Debe sobrescribir el método mover() para que imprima: “El avión está volando en el cielo”

## 3- Polimorfismo:

\* Crea una lista de objetos que incluya instancias de las clases Coche, Bicicleta y Avion.

\* Utiliza un bucle for para recorrer la lista y llamar al método mover() en cada vehículo. El resultado debe ser diferente dependiendo del tipo de vehículo, lo que demuestra el uso del polimorfismo.

}