

INSTITUTO POLITÉCNICO NACIONAL

Unidad Profesional Interdisciplinaria en  
Ingeniería y Tecnologías Avanzadas

Programación avanzada

Práctica 2: Clases y Objetos

Adalberto Cabrera Vazquez

Boleta: 2023640791

Ingeniería Mecatrónica, Grupo 2MV7

06 de Marzo del 2024

# 1 Objetivo

Determinar las clases que conforman un problema y crearlas en un lenguaje de programación.

## 2 Introducción

### 2.1 ¿Qué es un objeto en programación?

Los objetos en programación representan cosas del mundo real, así como conceptos abstractos con sus características y comportamientos específicos. Un objeto cuenta con su estructura interna que combina variables, funciones y estructuras de datos. Usando el nombre del objeto y la sintaxis según el lenguaje de programación, puedes visualizar los valores del objeto y llamar las funciones que tiene predefinidas. Los elementos de un objeto se dividen en dos categorías principales: propiedades y métodos.

Las propiedades, también conocidas como atributos, incluyen información sobre el objeto. Por ejemplo, si consideramos un objeto Coche, algunas de sus propiedades serán: el color, la marca, el modelo o el año de fabricación.

### 2.2 Clases

Una clase es un elemento de la programación orientada a objetos que actúa como una plantilla y va a definir las características y comportamientos de una entidad. La clase va a ser como un molde a partir del cual vamos a poder definir entidades. Una clase va a definir las características y los comportamientos de una entidad. Si yo defino, por ejemplo, la clase persona, sus características o atributos podrían ser género, es decir, si es hombre o mujer, edad y nombre. Los comportamientos o métodos que tendríamos en el caso de la clase persona podrían ser respirar, moverse, caminar, pensar, entre otras.

## 3 Desarrollo

### 3.1 Programa 1

Escriba un programa que pida al usuario el nombre, la edad, la estatura y el número telefónico de tres personas y lo guarde en una lista de tipo Persona. Para realizarlo, deberá definir la clase Persona con las propiedades Nombre, Edad, Estatura y Teléfono. Al finalizar su programa, este deberá mostrar en pantalla los datos de cada persona capturada. Para esto deberá crear el método to string(). La salida deberá ser como sigue:

```
Persona 1:  
¡Hola!, Mi nombre es José, tengo 20 años, nací en 2003,  
mido 1.70 metros y mi número de contacto es 55123456789 ¡Saludos!  
Persona 2:  
¡Hola!, Mi nombre es Karina, tengo 21 años, nací en 2002,  
mido 1.60 metros y mi número de contacto es 55123123123 ¡Saludos!  
Persona 3:  
¡Hola!, Mi nombre es Luis, tengo 22 años, nací en 2001,  
mido 1.80 metros y mi número de contacto es 55456456456 ¡Saludos!
```

Figure 1: Ejemplo de como debe de imprimirse

Para hacerlo lo primero que debemos hacer es importar las bibliotecas que podamos utilizar y después crear nuestra clase llamada Persona. Definimos el objeto así como sus atributos. Después definiremos como se va a imprimir cuando llamemos a imprimir.

```

import numpy as np
import os
class Persona:
    def __init__(self, nombre:str, edad:int, estatura: float, numero:int)->None:
        self.__nombre=nombre
        self.__edad=edad
        self.__estatura=estatura
        self.__numero=numero

    def __str__(self) ->str:
        return f'¡Hola!, Mi nombre es {self.__nombre}, tengo {self.__edad} años, nací en {2024-self.__edad}, mi

```

Figure 2: Creacion de la clase con sus atributps y métodos correspondientes

Después creamos nuestra lista de personas vacia, para que podamos meter los objetos(personas) en ella. Despues usamos de un bucle for para meter los objetos, con los atributos que nos den el usuario, en la lista. Solamente es para una lista de 3 personas.

```

lista_personas=[]
for i in range(3):
    print('')
    persona =Persona(str(input('Favor de ingresar tu nombre: ')),int(input('Edad:')),float(input('Estatura:')),i
    lista_personas.append(persona)
    i+=1
os.system('cls')

```

Figure 3: Ingresar la lista de personas

Finalmente utilizamos otro for para poder imprimir la lista de personas pero con el método que antes en la clase ya especificamos.

```

for i in range(3):
    print('Persona '+str(i+1)+':')
    print(lista_personas[i])
    print('')
    i+=1

```

Figure 4: Mostrar en la terminal la lista de personas

## 3.2 Ejecución Programa 1

```

Favor de ingresar tu nombre: Adalberto
Edad:19
Estatura:1.75
numero telefonico:9984037473

Favor de ingresar tu nombre: Emily
Edad:19
Estatura:1.61
numero telefonico:5564351824

Favor de ingresar tu nombre: Emilio
Edad:20
Estatura:1.70
numero telefonico:5568427894

```

Figure 5: Captura de datos de las personas

```

Persona 1:
¡Hola!, Mi nombre es Adalberto, tengo 19 años, nací en 2005, mido 1.75 metros y mi numero es 9984037473. ¡SALUDOS!

Persona 2:
¡Hola!, Mi nombre es Emily, tengo 19 años, nací en 2005, mido 1.61 metros y mi numero es 5564351824. ¡SALUDOS!

Persona 3:
¡Hola!, Mi nombre es Emilio, tengo 20 años, nací en 2004, mido 1.7 metros y mi numero es 5568427894. ¡SALUDOS!

```

Figure 6: Datos mostrados en la terminal

### 3.3 Programa 2

Escriba un programa el cual permita crear una lista de alumnos. El programa deberá almacenar: nombre, apellido paterno, apellido materno, número de boleta, fecha de nacimiento, carrera, grupo y correo electrónico.

El programa deberá solicitar los datos de un profesor(nombre completo y número de empleado)y posteriormente permitir el registro de n estudiantes.

Al finalizar la captura de los n alumnos, se deberá mostrar en la pantalla la lista completa con el siguiente formato:

```

Profesor: José Luis
No. Empleado: 123456
=====
Lista de Alumnos Inscritos
=====
1. Cruz Mora, José Luis - 123456789 - 01/01/2000 -
Ing. Mecatrónica - 2MV13 - jlcruz@ipn.mx
...
[Demás elementos de la lista]
...
=====
Total de alumnos inscritos: [n]

```

Figure 7: Formato de los datos

Primero creamos nuestra clase de Alumnos en la que le pondremos todos los atributos que nos pidieron y también agregaremos un método para poder mandarlo a imprimir de cierta manera.

```
#Crear la clase alumnos con todos sus atributos
class Alumno:
    def __init__(self, nombre:str, apellidop:str, apellidom:str, noboleta:int, nacimiento:str, carrera:str, grupo:str, correo:str):
        self.__nombre = nombre
        self.__apellidop = apellidop
        self.__apellidom = apellidom
        self.__noboleta = noboleta
        self.__nacimiento = nacimiento
        self.__carrera = carrera
        self.__grupo = grupo
        self.__correo= correo
    #Definir que va a hacer cuando se mande a imprimir
    def __str__(self) ->str:
        return f'{i+1}. {self.__apellidop} {self.__apellidom}, {self.__nombre} --- {self.__noboleta} --- {self.__nacimiento} --- {self.__carrera} --- {self.__grupo} --- {self.__correo}'
```

Figure 8: Creación de la clase Alumnos

Despues creamos la clase de Profesor para crear al objeto de profesor con los atributos antes mencionados e igualmente le definiremos un método para imprimirlo de cierta manera.

```
#Crear la clase Profesor
class Profesor:
    def __init__(self, nombrep:str, empleado:str) -> None:
        self.__nombrep= nombrep
        self.__empleado= empleado
    #Definir que va a hacer cuando se mande a imprimir
    def __str__(self) ->str:
        return f'Profesor: {self.__nombrep}\nNo. Empleado: {self.__empleado}'
```

Figure 9: Creación de la clase Profesor

Ahora, tenemos que crear al objeto llamado profesor agregandole los atributos que el usuario nos brinde. Tambien le pediremos el numero de alumnos que va a registrar y lo almacenaremos en la variable noalumnos.

```
#Crear el objeto llamado profesor y posteriormente agregas el numero de alumnos que va a registrar
profesor =Profesor(str(input('Favor de ingresar tu nombre: ')),int(input('No. Empleado:')))
noalumnos=int(input('Favor de ingresar el numero de alumnos que va a registrar: '))
```

Figure 10: Creacion del objeto 'profesor'

Lo que sigue es crear nuestra lista de alumnos vacia para poder ingresar los objetos 'alumnos' creados por el usuario. Para ingresar los datos usé un for en donde el limite es el numero de alumnos ingresados anteriormente y los va agregando a la lista.

```

os.system('cls')
#Creacion de la lista de alumnos
lista_alumnos=[]
#Bucle para registrar a los alumnos y agregarlos en la lista creando objetos llamados alumnos con la clase de Al
for i in range(noalumnos):
    print('Alumno '+str(i+1))
    alumno=Alumno(str(input('Nombre del alumno: ')), str(input('Apellido paterno del alumno: ')), str(input('Ape
    lista_alumnos.append(alumno)
    print('\n')
    i+=1

```

Figure 11: Creación de la lista de alumnos

Por último, imprimimos el objeto del profesor y posteriormente la lista de alumnos conforme al formato antes establecido, haciendo uso de print's ydelfor para imprimir los objetos de la lista de alumnos.

```

os.system('cls')
#Imprime el objeto de profesor
print(profesor)
print('*'*100)
print('Lista de alumnos inscritos')
print('*'*100)
#Imprime la lista de objetos(alumnos)
for i in range(noalumnos):
    print(lista_alumnos[i])
    i+=1
print('*'*100)
print('Total de alumnos incritos: '+str(noalumnos))

```

Figure 12: Impresión de la lista final

### 3.4 Ejecucion Programa 2

```

Favor de ingresar tu nombre: Cruz Mora Jose Luis
No. Empleado:123456
Favor de ingresar el numero de alumnos que va a registrar: 5

```

Figure 13: Ingreso de los datos del profesor por el usuario



<p>Alumno 1</p> <p>Nombre del alumno: Adalberto</p> <p>Apellido paterno del alumno: Cabrera</p> <p>Apellido materno del alumno: Vazquez</p> <p>No.boleta del alumno: 2023640791</p> <p>Fecha de nacimiento (DD/MM/AAAA): 26/05/2004</p> <p>Carrera: Ing. Mecatrónica</p> <p>Grupo: 1MV3</p> <p>Correo institucional: adal.cabrera.v@gmail.com</p>	<p>Alumno 3</p> <p>Nombre del alumno: Itzel Berenice</p> <p>Apellido paterno del alumno: Cabrera</p> <p>Apellido materno del alumno: Vazquez</p> <p>No.boleta del alumno: 2020548464</p> <p>Fecha de nacimiento (DD/MM/AAAA): 16/10/2001</p> <p>Carrera: Ing. Telemática</p> <p>Grupo: 1MV3</p> <p>Correo institucional: itzel.cabrera.v@gmail.com</p>
<p>Alumno 2</p> <p>Nombre del alumno: Emily Naomi</p> <p>Apellido paterno del alumno: Osorio</p> <p>Apellido materno del alumno: Esquivel</p> <p>No.boleta del alumno: 2023640712</p> <p>Fecha de nacimiento (DD/MM/AAAA): 05/10/2004</p> <p>Carrera: Ing. Mecatrónica</p> <p>Grupo: 1MV3</p> <p>Correo institucional: emily.osorio.e@gmail.com</p>	<p>Alumno 4</p> <p>Nombre del alumno: Abril Cassandra</p> <p>Apellido paterno del alumno: Cabrera</p> <p>Apellido materno del alumno: Vazquez</p> <p>No.boleta del alumno: 2024110338</p> <p>Fecha de nacimiento (DD/MM/AAAA): 30/12/2008</p> <p>Carrera: Ing. Arquitecto</p> <p>Grupo: 1MV3</p> <p>Correo institucional: abril.cabrera.v@gmail.com</p>
<p>Alumno 5</p> <p>Nombre del alumno: Emilio</p> <p>Apellido paterno del alumno: Jimenez</p> <p>Apellido materno del alumno: Rivera</p> <p>No.boleta del alumno: 2023948725</p> <p>Fecha de nacimiento (DD/MM/AAAA): 13/01/2001</p> <p>Carrera: Ing.Mecatrónica</p> <p>Grupo: 1MV3</p> <p>Correo institucional: emilio.jimenez.r@gmail.com</p>	

Figure 14: Ingreso de alumnos

```

Profesor: Cruz Mora Jose Luis
No. Empleado: 123456
*****
Lista de alumnos inscritos
*****
1. Cabrera Vazquez, Adalberto --- 2023640791 --- 26/05/2004 ---
Ing. Mecatrónica --- 1MV3 --- adal.cabrera.v@gmail.com

2. Osorio Esquivel, Emily Naomi --- 2023640712 --- 05/10/2004 ---
Ing. Mecatrónica --- 1MV3 --- emily.osorio.e@gmail.com

3. Cabrera Vazquez, Itzel Berenice --- 2020548464 --- 16/10/2001 ---
Ing. Telemática --- 1MV3 --- itzel.cabrera.v@gmail.com

4. Cabrera Vazquez, Abril Cassandra --- 2024110338 --- 30/12/2008 ---
Ing. Arquitecto --- 1MV3 --- abril.cabrera.v@gmail.com

5. Jimenez Rivera, Emilio --- 2023948725 --- 13/01/2001 ---
Ing. Mecatrónica --- 1MV3 --- emilio.jimenez.r@gmail.com

*****
Total de alumnos inscritos: 5

```

Figure 15: Pantalla final

## 4 Conclusión

En esta práctica pude determinar con éxito las clases necesarias para la realización del problema y poder así pasarlo a un código de programación.

## 5 Bibliografía

- León, N. (2020, November 10). Qué es una clase en POO - C esencial [Video]. LinkedIn. <https://es.linkedin.com/learning/c-sharp-esencial-3/que-es-una-clase-en-poo#:~:text=Una%20clase%20es%20un%20elemento,vamos%20a%20poder%20definir%20entidades.>
- Programación orientada a objetos. (n.d.). El Libro De Python. <https://ellibrodepython.com/programacion-orientada-a-objetos-python>
- Sandoval, E. (2023, September 21). ¿Qué es un objeto en programación? Ebac. <https://ebac.mx/blog/objeto-en-programacion>