

INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS Y UML
BIBIOTECA

HANNA KATHERINE ABRIL GÓNGORA
KAROL ASLEY ORJUELA MAPE

UNIVERSIDAD MANUELA BELTRÁN

PROGRAMACIÓN ORIENTADA A OBJETOS

DOCENTE

DIANA MARCELA TOQUICA RODRÍGUEZ

BOGOTÁ DC VIERNES 23 DE FEBRERO

1. PREGUNTAS ORIENTADORAS

1.1 En este apartado se realiza el análisis de los datos obtenidos, estos pueden ser de forma cualitativa o cuantitativa según la naturaleza de la práctica.

- En el código, se adoptó una estructura de código organizada en clases y métodos. Esto nos permite modularizar la funcionalidad, facilitando la comprensión, el mantenimiento y la reutilización del código.
- En el código, se desarrollaron funciones específicas para inscribir usuarios, consultar disponibilidad de libros y gestionar la biblioteca de manera eficiente. Esto nos permite ofrecer un servicio más ágil y satisfactorio para nuestros usuarios.
- En el código se utilizaron bucles para iterar sobre colecciones de datos y automatizar procesos. Esto mejora la eficiencia y la legibilidad del código.

1.2 ¿Cuáles fueron los aprendizajes obtenidos al realizar esta guía?, liste como mínimo 3 aprendizajes y relaciónelos con su futuro que hacer profesional.

- Durante este trabajo, ampliamos nuestras habilidades en Python y profundizamos en los principios de la programación orientada a objetos. Esto nos permite abordar proyectos más complejos y escribir código más estructurado y mantenible.
- Durante nuestro aprendizaje, adquirimos habilidades en la creación y uso de diagramas UML, aplicándolos con éxito en proyectos de desarrollo en Python. Esto nos permite diseñar y comunicar de manera más efectiva la arquitectura de nuestros programas.
- Saber diseñar las estructuras básicas para sistemas simples como en este caso el de la biblioteca, donde se implementaron acciones básicas en un sistema de ese tipo.

2. PREGUNTAS

2.1 ¿Cómo escribir programas que tengan un alto nivel de flexibilidad?

Para escribir programas con un alto nivel de flexibilidad en el análisis orientado a objetos:

- **Abstracción y Modularidad:**

La abstracción representa las características esenciales de un objeto sin preocuparse por los detalles internos. Diseña clases y métodos que sean lo más abstractos posible. La modularidad divide tu código en módulos o clases independientes. Cada clase debe tener una responsabilidad específica. Esto facilita la reutilización y el mantenimiento.

- **Encapsulamiento:**

Oculta los detalles internos de una clase y expone solo lo necesario. Utiliza modificadores de acceso (como public, private y protected) para controlar la visibilidad de los miembros de la clase.

El encapsulamiento protege el estado interno de los objetos y permite cambios internos sin afectar el código externo.

- **Herencia:**

Utiliza la herencia para crear jerarquías de clases. Una clase puede heredar propiedades y métodos de otra.

La herencia permite reutilizar código y definir relaciones entre clases (por ejemplo, una clase “Empleado” hereda de una clase “Persona”).

- **Polimorfismo:**

El polimorfismo permite que un objeto se comporte de diferentes maneras según el contexto. Utiliza interfaces, clases abstractas y métodos sobrescritos para lograr polimorfismo. Por ejemplo, una interfaz “Figura”

puede tener métodos como “calcularÁrea()” que se implementan de manera diferente en las clases “Círculo” y “Rectángulo”.

- Interfaces y Composición:

Las interfaces definen contratos que las clases deben cumplir. Utiliza interfaces para establecer comportamientos comunes. La composición implica combinar objetos más pequeños para formar objetos más complejos. Por ejemplo, una clase “Coche” puede contener objetos de las clases “Motor”, “Rueda” y “Transmisión”.

- Manejo de Excepciones:

Implementa manejo de excepciones para lidiar con errores de manera elegante. Esto evita que el programa se bloquee y permite una recuperación adecuada. Utiliza bloques try, catch y finally para controlar situaciones excepcionales.

2.2 ¿Qué tipos de objetos deben existir en el programa?

En un programa de una biblioteca simple, varios tipos de objetos son esenciales para su funcionamiento. Aquí están algunos de ellos:

- Libros:

Representan las publicaciones impresas que contienen información sobre temas específicos. Cada libro tiene atributos como su código, título, autor y disponibilidad. Pueden ser prestados a los socios de la biblioteca.

- Socios:

Los socios son las personas que se dan de alta en la biblioteca. Cada socio está caracterizado por un número de membresía, nombre y email. Pueden

tomar prestados libros de la biblioteca.

- Préstamos:

Los préstamos representan la acción de prestar un libro a un socio.

Cada préstamo está caracterizado por el código del libro prestado, el número de socio y la fecha del préstamo.

- Clases y Métodos:

Utiliza programación orientada a objetos (POO) para crear clases como Libro, Usuario y Préstamo. Cada clase debe tener métodos para registrar, consultar y gestionar los libros, socios y préstamos.

2.3 ¿Cuáles son las propiedades (atributos) y comportamientos (métodos) de los objetos a crear?

Un sistema de biblioteca simple requiere una estructura organizada para gestionar los procesos relacionados con la biblioteca. A continuación, te presento algunos atributos y métodos que podrían ser necesarios para un sistema de biblioteca básico:

2.3.1 Atributos:

- Libro:

- Título: El nombre del libro.
- Autor: El autor o autores del libro.
- Género: La categoría a la que pertenece el libro (por ejemplo, ficción, no ficción, ciencia ficción, etc.).
- ISBN: El número de identificación único del libro.
- Disponibilidad: Indica si el libro está disponible para préstamo.

- Prestatario:

- Nombre: El nombre completo del prestatario.
- Número de identificación: Un identificador único para el prestatario (por ejemplo, número de estudiante o cédula).

- Historial de préstamos: Registra los libros que el prestatario ha tomado prestados anteriormente.
- Préstamo:
 - Fecha de préstamo: La fecha en que se prestó el libro.
 - Fecha de devolución: La fecha en que se debe devolver el libro.
 - Estado del préstamo: Indica si el libro ha sido devuelto o está pendiente de devolución.

2.3.2 Métodos:

- Agregar libro: Permite registrar un nuevo libro en la biblioteca.
- Eliminar libro: Elimina un libro de la base de datos.
- Buscar libro por título o autor: Facilita la búsqueda de libros específicos.
- Registrar préstamo: Registra un préstamo de libro a un prestatario.
- Devolver libro: Actualiza el estado del préstamo cuando un libro se devuelve.

3. EXPLICACIÓN DEL CÓDIGO EN PYTHON

Este código es una simulación simple de un sistema de gestión de una biblioteca que permite a los usuarios inscribirse, buscar libros, solicitarlos y devolverlos, así como también permite al bibliotecario administrar los libros y usuarios de la biblioteca. Veamos cada parte del código:

- *Clase: Biblioteca*, define atributos estáticos para almacenar información sobre usuarios (nombre, documento, email, etc.) y libros (título, categoría, autor, etc.).

```

1  class Biblioteca:
2
3      nombre=[]
4      documento=[]
5      email=[]
6      tipo_de_membresia=[]
7      pasword=[]
8
9      titulo=[]
10     categoria=[]
11     autor=[]
12     isbn=[]
13     fecha_entrega=[]
14     fecha_devolucion=[]
15     cantidad=[]
16
17     cont_usuarios=0
18     cont_libros=0
19

```

- *inscribir()*: Agrega un nuevo usuario a la biblioteca, esta permite agregar al usuario mediante la petición de alguna información al usuario y el almacenaje de esta en sus respectivas listas.

```

21  @staticmethod
22  def inscribir(nombre, documento, email, tipo_de_membresia, pasword):
23      Biblioteca.nombre.append(nombre)
24      Biblioteca.documento.append(documento)
25      Biblioteca.email.append(email)
26      Biblioteca.tipo_de_membresia.append(tipo_de_membresia)
27      Biblioteca.pasword.append(pasword)
28      Biblioteca.cont_usuarios+=1
29      print("Usuario inscrito correctamente.")
30

```

- *consultar()*: Busca y muestra información de un usuario por su documento, la información mostrada es la que se almaceno respectivamente al momento del usuario inscribirse mediante el método *inscribir()* se imprime toda la información correspondiente excepto información importante y privada como la contraseña.

```

31  @staticmethod
32  def consultar(documento):
33      encontrado=False
34      for j in range(Biblioteca.cont_usuarios):
35          if Biblioteca.documento[j]==documento:
36              print("-----")
37              print(f"Nombre: {Biblioteca.nombre[j]}")
38              print(f"Documento: {Biblioteca.documento[j]}")
39              print(f"Email: {Biblioteca.email[j]}")
40              print(f"Tipo de membresia: {Biblioteca.tipo_de_membresia[j]}")
41              print("-----")
42              encontrado=True
43              break
44
45      if not encontrado:
46          print("No se encontró ningún usuario correspondiente.")

```

- *consultar_libro()*: Busca y muestra información de un libro por título, categoría o autor, hay que tener en cuenta que tuvo que haberse inscrito un libro o libros con anterioridad para poder consultarlos, este método funciona de manera similar al de *consultar()* la diferencia es que en este método la búsqueda puede ser de diferentes maneras.

```

48     @staticmethod
49     def consultar_libro(titulo,categoria,autor):
50         encontrado=False
51         for i in range(Biblioteca.cont_libros):
52             if Biblioteca.titulo[i]==titulo or Biblioteca.categoria[i] == categoria or Biblioteca.autor[i]==autor:
53                 print("-----")
54                 print(f"Nombre: {Biblioteca.titulo[i]}")
55                 print(f"Categoria: {Biblioteca.categoria[i]}")
56                 print(f"Autor: {Biblioteca.autor[i]}")
57                 print(f"Codigo: {Biblioteca.isbn[i]}")
58                 print("-----")
59                 encontrado=True
60                 break
61
62         if not encontrado:
63             print("No se encontró ningún libro correspondiente.")
64

```

- *listar_lib()*: Muestra una lista de todos los libros en la biblioteca. Si no hay libros inscritos se arroja que no hay ningún libro correspondiente a ese parámetro de búsqueda. Este hace una búsqueda en donde se almaceno el libro al momento de agregarlo en este caso una lista.

```

65     @staticmethod
66     def listar_lib():
67         if Biblioteca.cont_libros == 0:
68             print("No hay libros en la biblioteca.")
69             return
70
71         for i in range(Biblioteca.cont_libros):
72             print("-----")
73             print(f"Nombre: {Biblioteca.titulo[i]}")
74             print(f"Categoria: {Biblioteca.categoria[i]}")
75             print(f"Codigo: {Biblioteca.isbn[i]}")
76             print(f"Autor: {Biblioteca.autor[i]}")
77             print(f"Cantidad: {Biblioteca.cantidad[i]}")
78             print("-----")
79

```

- *listar_usu()*: Muestra una lista de todos los usuarios inscritos, los cuales fueron almacenados al momento de inscribir en una lista.

```

80     @staticmethod
81     def listar_usu():
82         if Biblioteca.cont_usuarios == 0:
83             print("No hay usuarios inscritos.")
84             return
85
86         for j in range(Biblioteca.cont_usuarios):
87             print("-----")
88             print(f"Nombre: {Biblioteca.nombre[j]}")
89             print(f"Documento: {Biblioteca.documento[j]}")
90             print(f"Email: {Biblioteca.email[j]}")
91             print(f"Tipo de membresia: {Biblioteca.tipo_de_membresia[j]}")
92             print("-----")
93

```

- *agregar()*: Agrega un nuevo libro a la biblioteca, los cuales los usuarios podrán acceder mas adelante y los bibliotecarios podrán hacer consultas sobre estos.


```

94     @staticmethod
95     def agregar(titulo, isbn, autor, categoria, cantidad):
96         Biblioteca.titulo.append(titulo)
97         Biblioteca.isbn.append(isbn)
98         Biblioteca.autor.append(autor)
99         Biblioteca.categoria.append(categoria)
100        Biblioteca.cantidad.append(cantidad)
101        Biblioteca.cont_libros+=1
102        print("Libro agregado correctamente.")
103

```

- *eliminar()*: Remueve o almacena un libro de la biblioteca, se usa en el caso de que quieran descontinuar un libro o lo quieran almacenar.

```

104     @staticmethod
105     def eliminar(titulo):
106         if titulo in Biblioteca.titulo:
107             indice_a_eliminar = Biblioteca.titulo.index(titulo)
108             del Biblioteca.titulo[indice_a_eliminar]
109             del Biblioteca.isbn[indice_a_eliminar]
110             del Biblioteca.autor[indice_a_eliminar]
111             del Biblioteca.categoria[indice_a_eliminar]
112             del Biblioteca.cantidad[indice_a_eliminar]
113             Biblioteca.cont_libros -= 1
114             print(f"Libro '{titulo}' eliminado correctamente.")
115         else:
116             print("El libro no está en la lista")
117

```

- *eliminar_libro_pedido()*: Remueve un libro de la lista de libros pedidos por un usuario, este se usa al momento de que el usuario devuelva el libro, si este no lo devuelve este método no se usa, por lo que este método es dependiente del método *devolver_lib()*

```

119     @staticmethod
120     def eliminar_libro_pedido(titulo):
121         if titulo in Usuarios.libros_pedidos:
122             Usuarios.libros_pedidos.remove(titulo)
123             print(f"Libro '{titulo}' eliminado de la lista de libros pedidos")
124         else:
125             print(f"El libro '{titulo}' no está en la lista de libros pedidos")
126
127

```

- *Clase: Usuarios*, define un atributo estático para almacenar los libros pedidos por los usuarios, y tiene métodos que permiten realizar prestamos, devoluciones y revisar los libros que ese usuario pueda tener en su poder.

```

128     class Usuarios:
129
130         libros_pedidos=[]
131         cont_lib_usu=0
132

```

- *pedir_lib()*: Permite a un usuario solicitar un libro, este revisa si el libro esta disponible en la lista de libros en la biblioteca y así poder prestarlo.

```

133     @staticmethod
134     def pedir_lib(titulo):
135         disponible = False
136         for i in range(Biblioteca.cont_libros):
137             if Biblioteca.titulo[i] == titulo:
138                 Usuarios.libros_pedidos.append(titulo)
139                 disponible = True
140                 Usuarios.cont_lib_usu+=1
141                 print("Solicitud de libro aceptada")
142                 break
143         if not disponible:
144             print("Libro solicitado no disponible")
145
146     @staticmethod
147     def devolver_lib(titulo):
148         if titulo in Usuarios.libros_pedidos:
149             Biblioteca.eliminar_libro_pedido(titulo)
150             print("Libro devuelto correctamente")
151         else:
152             print("El libro no está en la lista de libros pedidos")
153

```

- *devolver_lib()*: Permite a un usuario devolver un libro, se debe tener en cuenta que si el libro no esta en la lista de libros prestados a dicho usuario no se va a poder devolverlo, ya que este método hace la comparación directa con ese método.

```

146     @staticmethod
147     def devolver_lib(titulo):
148         if titulo in Usuarios.libros_pedidos:
149             Biblioteca.eliminar_libro_pedido(titulo)
150             print("Libro devuelto correctamente")
151         else:
152             print("El libro no está en la lista de libros pedidos")
153

```

- *listar_lib_usu()*: Lista los libros que ha pedido un usuario, la lista incluye toda la información de libro.

```

154     @staticmethod
155     def listar_lib_usu():
156         if Usuarios.cont_lib_usu == 0:
157             print("No hay libros en la biblioteca.")
158             return
159
160         for i in range(Biblioteca.cont_libros):
161             print("-----")
162             print(f"Nombre: {Biblioteca.titulo[i]}")
163             print(f"Categoría: {Biblioteca.categoria[i]}")
164             print(f"Código: {Biblioteca.isbn[i]}")
165             print(f"Autor: {Biblioteca.autor[i]}")
166             print("-----")
167

```

- Bucle principal: Un bucle *while* True que presenta un menú de opciones. Dependiendo de la opción elegida por el usuario, se llama a los métodos correspondientes de las clases Biblioteca o Usuarios.
Cada opción tiene un submenú, en especial Biblioteca y Usuarios, ya que estos tienes sus funciones por aparte.
En este bucle se llaman todas las clases y métodos definidos anterior mente para que el código funcione correctamente.

```

169 while True:
170     print("-----")
171     print("\n Menu:")
172     print("""
173     1. Bibliotecario
174     2. Usuario
175     3. Inscribirse
176     4. Salir del sistema
177     """)
178     print("-----")
179     opcion = int(input("Ingresa una opcion: "))
180
181
182     if opcion == 1:
183         print("-----")
184         print("\n Menu:")
185         print("""
186         1. Consultar usuario por documento
187         2. Consultar libro por titulo, categoria o autor
188         3. Listar todos los usuarios
189         4. Listar todos los libros
190         5. Agregar libros a la biblioteca/sistema
191         6. Almacenar (sacar o descontinuar) libros
192         7. Salir del sistema
193         """)
194         print("-----")
195         opc_Bl = int(input("Ingresa una opcion: "))
196
197         if opc_Bl == 1:
198             print("Consultar Usuario: ")
199             doc = input("Ingrese el documento: ")
200             Biblioteca.consultar(doc)

```

```

202         if opc_Bl == 2:
203             print("-----")
204             print("\n Escoja tipo de búsqueda:")
205             print("""
206             1. Consultar libro por título
207             2. Consultar libro por categoria
208             3. Consultar libro por autor
209             """)
210             print("-----")
211             op_lib = int(input("Ingresa una opcion: "))
212
213             if op_lib == 1:
214                 print("Consulta por título")
215                 titulo = input("Ingresa el título del libro:")
216                 Biblioteca.consultar_libro(titulo, "", "")
217             if op_lib == 2:
218                 print("Consulta por categoria")
219                 categoria = input("Ingresa la categoria del libro:")
220                 Biblioteca.consultar_libro("", categoria, "")
221             if op_lib == 3:
222                 print("Consulta por autor")
223                 autor = input("Ingresa el autor del libro:")
224                 Biblioteca.consultar_libro("", "", autor)
225
226
227             else:
228                 print("Opcion invalida")
229
230         if opc_Bl == 3:
231             print("Estos son los usuarios inscritos")
232             print("")
233             Biblioteca.listar_usu()
234
235         if opc_Bl == 4:
236             print("Estos son los libros en la biblioteca")
237             print("")
238             Biblioteca.listar_lib()

```

```

240         if opc_Bl == 5:
241             print("Agregar libro")
242             titulo = input("Ingrese el titulo completo del libro: ")
243             isbn = input("Ingrese codigo de 4 numeros: ")
244             autor = input("Ingrese nombre completo del autor o alias: ")
245             categoria = input("Ingrese la categoria o categorias del libro: ")
246             cantidad = int(input("Ingrese la cantidad de libros que ingresan de este titulo: "))
247             Biblioteca.agregar(titulo, isbn, autor, categoria, cantidad)
248
249         if opc_Bl == 6:
250             titulo = input("Pon el titulo del libro que desea eliminar:")
251             Biblioteca.eliminar(titulo)
252
253         if opc_Bl == 7:
254             print("Saliendo del sistema...")
255             break
256
257         else:
258             print("Opcion invalida. ")
259
260
261     if opcion == 2:
262         login=input("Ingresa el correo con el que estas inscrito: ")
263         for i in range(Biblioteca.cont_usuarios):
264             if Biblioteca.email[i]==login:
265                 password=input("Ingrese su contraseña: ")
266                 for j in range(Biblioteca.cont_usuarios):
267                     if Biblioteca.password[j]==password:
268                         print("-----")
269                         print("\n Menu:")
270                         print("""
271                         1. Pedir libro
272                         2. Devolver libro
273                         3. Listar libros que he pedido
274                         4. Salir del sistema
275                         """)
276                         print("-----")
277                         opc_Us = int(input("Ingresa una opcion: "))
278
279                     if opc_Us == 1:
280                         print("\n Solicitud de libro")
281                         titulo = input("Ingrese el titulo del libro que desea pedir: ")
282                         Usuarios.pedir_lib(titulo)
283                     if opc_Us == 2:
284                         titulo = input("Ingrese el titulo del libro que desea devolver: ")
285                         Usuarios.devolver_lib(titulo)
286                     if opc_Us == 3:
287                         print("Estos son los libros que has pedido hasta el momento: ")
288                         print("")
289                         Usuarios.listar_lib_usu()
290                     if opc_Us==4:
291                         print("Saliendo del sistema...")
292                         break
293                     else:
294                         print("Opcion invalida.")
295                 else:
296                     print("Contraseña incorrecta")
297             else:
298                 print("Usuario invalido")
299
300
301

```

```

302     if opcion == 3:
303         print("\nInscripción de Usuario:")
304         nombre = input("Nombre: ")
305         documento = input("Documento: ")
306         email = input("Email: ")
307         password=input("Ingresa la contraseña de tu usuario: ")
308         print("""Membresias:
309             1. Estandar: Maximo tiempo de prestamo de un libro son 7 dias
310             2. VIP: Maximo tiempo de prestamo de un libro son 15 dias""")
311         print("")
312         tipo_de_membresia = int(input("Escoja el tipo de membresia que desea: "))
313         if tipo_de_membresia==1:
314             tipo_de_membresia="Estandar"
315         if tipo_de_membresia==2:
316             tipo_de_membresia="VIP"
317         else:
318             print("Opcion no valida")
319             continue
320         Biblioteca.inscribir(nombre, documento, email, tipo_de_membresia,password)
321
322
323     if opcion == 4:
324         print("Saliendo del sistema...")
325         break
326
327
328     else:
329         print("Opcion invalida.")
330

```

4. EXPLICACION DEL CODIGO UML

```

@startuml Biblioteca

class Biblioteca {
    - nombre: List[str]
    - documento: List[str]
    - email: List[str]
    - tipo_de_membresia: List[str]
    - password: List[str]
    - titulo: List[str]
    - categoria: List[str]
    - autor: List[str]
    - isbn: List[str]
    - fecha_entrega: List[datetime]
    - fecha_devolucion: List[datetime]
    - cantidad: List[int]
    - cont_usuarios: int
    - cont_libros: int
    + inscribir(nombre: str, documento: str, email: str, tipo_de_membresia: str, password: str): None
    + consultar(documento: str): None
    + consultar_libro(titulo: str, categoria: str, autor: str): None
    + listar_lib(): None
    + listar_usu(): None
    + agregar(titulo: str, isbn: str, autor: str, categoria: str, cantidad: int): None
    + eliminar(titulo: str): None
    + eliminar_libro_pedido(titulo: str): None
}

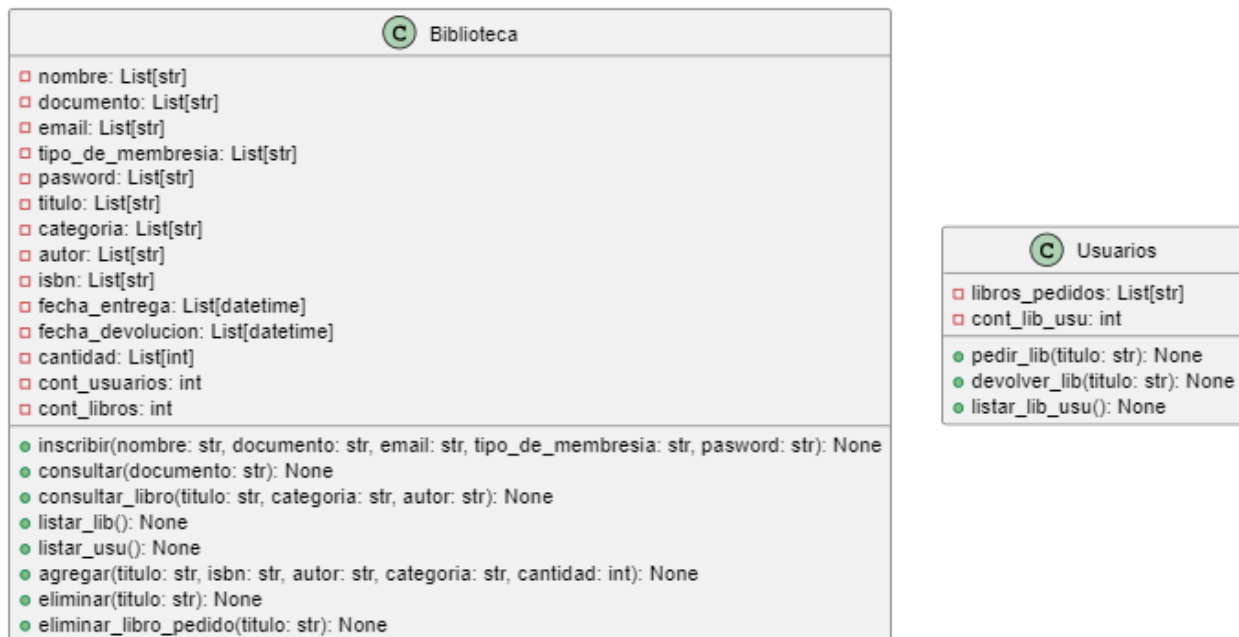
class Usuarios {
    - libros_pedidos: List[str]
    - cont_lib_usu: int
    + pedir_lib(titulo: str): None
    + devolver_lib(titulo: str): None
    + listar_lib_usu(): None
}

@enduml

```

- *@startuml* Biblioteca inicia un diagrama UML llamado “Biblioteca”.
- *class Biblioteca* , comienza la definición de una clase llamada “Biblioteca”.
- Las siguientes líneas definen los atributos de la clase “Biblioteca”, que incluyen listas de strings y datetimes, así como enteros.
- *__init__(...)* es el constructor de la clase que inicializa los atributos con los valores proporcionados cuando se crea un objeto de esta clase.
- Las siguientes líneas son métodos (funciones) que pertenecen a la clase “Biblioteca”. Estos métodos no retornan nada (None).
- *class Usuarios*, comienza la definición de otra clase llamada “Usuarios”.
- Nuevamente, las siguientes líneas definen los atributos y métodos para esta nueva clase.
- *@enduml* indica el final del diagrama UML.

5. DIAGRAMA DEL CODIGO EN UML



6. EXPLICACION Y USO

Es un proyecto en donde se implementa un pequeño sistema de biblioteca, con usuarios, libros y la biblioteca en sí.

En este sistema cada uno tiene accesos a cosas diferentes, por ejemplo, la biblioteca tiene accesos a funciones como inscribir usuarios, inscribir libros, revisar la lista tanto de usuarios inscritos como de libros agregados, hacer una búsqueda en específico de los dos, siendo la de usuario mediante su número de identificación y la de los libros mediante más parámetros como lo es el título, autor o categoría, la biblioteca también tiene la habilidad de almacenar o retirar libros. En cuanto al usuario después de inscrito este va a necesitar una contraseña para poder acceder al sistema de usuarios, después de ingresado este podrá realizar procesos como pedir libros, devolver libros, revisar libros que estén en su poder.

7. DEMOSTRACION DE PRUEBAS Y FUNCIONAMIENTO DEL CODIGO

7.1. Prueba 1.

```
Menu:
  1. Bibliotecario
  2. Usuario
  3. Inscribirse
  4. Salir del sistema

-----
Ingresa una opcion: 1
-----

Menu:
  1. Consultar usuario por documento
  2. Consultar libro por titulo, categoria o autor
  3. Listar todos los usuarios
  4. Listar todos los libros
  5. Agregar libros a la biblioteca/sistema
  6. Almacenar (sacar o descontinuar) libros
  7. Salir del sistema

-----
Ingresa una opcion: 5
Agregar libro
Ingresa el titulo completo del libro: Los miserables
Ingresa codigo de 4 numeros: 1862
Ingresa nombre completo del autor o alias: Victor Hugo
Ingresa la categoria o categorias del libro: Novela
Ingresa la cantidad de libros que ingresan de este titulo: 10
Libro agregado correctamente.
Opcion invalida.
Opcion invalida.
```


En esta prueba se demuestra la inscripción de un libro, en donde se accede al menú de bibliotecario, y después encuentra el menú de las acciones que este puede realizar.

7.2.Prueba 2.

```
Menu:

1. Bibliotecario
2. Usuario
3. Inscribirse
4. Salir del sistema

-----
Ingresa una opcion: 3

Inscripción de Usuario:
Nombre: Hanna Katherine Abril G
Documento: 1006258532
Email: abrilhanna@gmail.com
Ingresa la contraseña de tu usuario: 12345
Membresias:
    1. Estandar: Maximo tiempo de prestamo de un libro son 7 dias
    2. VIP: Maximo tiempo de prestamo de un libro son 15 dias

Escoja el tipo de membresia que desea: 2
Usuario inscrito correctamente.
Opcion invalida.
```

En esta prueba se muestra la inscripción de un usuario nuevo, directamente desde el menú principal.

7.3.Prueba 3.

```
Menu:

1. Bibliotecario
2. Usuario
3. Inscribirse
4. Salir del sistema

-----
Ingresa una opcion: 1
-----

Menu:

1. Consultar usuario por documento
2. Consultar libro por titulo, categoria o autor
3. Listar todos los usuarios
4. Listar todos los libros
5. Agregar libros a la biblioteca/sistema
6. Almacenar (sacar o descontinuar) libros
7. Salir del sistema

-----
Ingresa una opcion: 3
Estos son los usuarios inscritos

-----
Nombre: Hanna Katherine Abril G
Documento: 1006258532
Email: abrilhanna@gmail.com
Tipo de membresia: VIP
```

Se lista los usuarios inscritos, en este caso como solo se inscribió uno anteriormente, solo muestra ese usuario y toda su información, excepto información importante como la contraseña.

7.4.Prueba 4.

```
Menu:

1. Bibliotecario
2. Usuario
3. Inscribirse
4. Salir del sistema

-----
Ingresa una opcion: 1
-----

Menu:

1. Consultar usuario por documento
2. Consultar libro por titulo, categoria o autor
3. Listar todos los usuarios
4. Listar todos los libros
5. Agregar libros a la biblioteca/sistema
6. Almacenar (sacar o descontinuar) libros
7. Salir del sistema

-----
Ingresa una opcion: 4
Estos son los libros en la biblioteca

-----
Nombre: Los miserables
Categoria: Novela
Codigo: 1862
Autor: Victor Hugo
Cantidad: 10
-----
-----
Nombre: La ciudad de la alegría
Categoria: Ficción
Codigo: 2012
Autor: Dominique Lapierre
Cantidad: 15
```

En esta prueba se listan todos los libros agregados, en este caso podemos apreciar que hay dos libros que se agregaron anteriormente, para motivos de esta prueba.

7.5.Prueba 5.

```
Menu:

1. Bibliotecario
2. Usuario
3. Inscribirse
4. Salir del sistema

-----

Ingresa una opcion: 1
-----

Menu:

1. Consultar usuario por documento
2. Consultar libro por titulo, categoria o autor
3. Listar todos los usuarios
4. Listar todos los libros
5. Agregar libros a la biblioteca/sistema
6. Almacenar (sacar o descontinuar) libros
7. Salir del sistema

-----

Ingresa una opcion: 1
Consultar Usuario:
Ingresa el documento: 1006258532
-----

Nombre: Hanna Katherine Abril G
Documento: 1006258532
Email: abrilhanna@gmail.com
Tipo de membresia: VIP
```

En esta prueba se consulta el usuario mediante el numero de su documento, el programa hace una búsqueda en las listas de la información del usuario y retorna la coincidencia de este.

7.6.Prueba 6.

```
Menu:

1. Bibliotecario
2. Usuario
3. Inscribirse
4. Salir del sistema

-----
Ingresa una opcion: 1
-----

Menu:

1. Consultar usuario por documento
2. Consultar libro por titulo, categoria o autor
3. Listar todos los usuarios
4. Listar todos los libros
5. Agregar libros a la biblioteca/sistema
6. Almacenar (sacar o descontinuar) libros
7. Salir del sistema

-----
Ingresa una opcion: 2
-----

Escoja tipo de búsqueda:

1. Consultar libro por titulo
2. Consultar libro por categoria
3. Consultar libro por autor

-----
Ingresa una opcion: 1
Consulta por titulo
Ingresa el titulo del libro:La ciudad de la alegría
-----
Nombre: La ciudad de la alegría
Categoria: Ficción
Autor: Dominique Lapierre
Codigo: 2012
```

Se consulta un libro en este caso por el título, la búsqueda también se puede hacer con categoría o

autor.

7.7.Prueba 7.

```
Menu:

1. Bibliotecario
2. Usuario
3. Inscribirse
4. Salir del sistema

-----
Ingresa una opcion: 2
Ingresa el correo con el que estas inscrito: abrilhanna@gmail.com
Ingresa su contraseña: 12345
-----

Menu:

1. Pedir libro
2. Devolver libro
3. Listar libros que he pedido
4. Salir del sistema

-----
Ingresa una opcion: 1

Solicitud de libro
Ingresa el titulo del libro que desea pedir: Los miserables
Solicitud de libro aceptada
```

En esta prueba se realiza la comprobación en donde un usuario pide un libro de la biblioteca, para eso debe ingresar con sus credenciales las cuales puso en el momento de la inscripción y después solicita el libro si este esta disponible le acepta el préstamo, si no esta disponible le dice q el titulo no esta disponible.

7.8.Prueba 8.

```
Menu:

1. Bibliotecario
2. Usuario
3. Inscribirse
4. Salir del sistema

-----
Ingresa una opcion: 2
Ingresa el correo con el que estas inscrito: abrilhanna@gmail.com
Ingresa su contraseña: 12345
-----

Menu:

1. Pedir libro
2. Devolver libro
3. Listar libros que he pedido
4. Salir del sistema

-----
Ingresa una opcion: 3
Estos son los libros que has pedido hasta el momento:

-----
Nombre: Los miserables
Categoria: Novela
Codigo: 1862
Autor: Victor Hugo
-----
-----
Nombre: La ciudad de la alegría
Categoria: Ficción
Codigo: 2012
Autor: Dominique Lapierre
```

Se lista los libros que el usuario a pedido hasta el momento.

7.9.Prueba 9.

```
Menu:

1. Bibliotecario
2. Usuario
3. Inscribirse
4. Salir del sistema

-----

Ingresa una opcion: 2
Ingresa el correo con el que estas inscrito: abrilhanna@gmail.com
Ingresa su contraseña: 12345
-----

Menu:

1. Pedir libro
2. Devolver libro
3. Listar libros que he pedido
4. Salir del sistema

-----

Ingresa una opcion: 2
Ingresa el titulo del libro que desea devolver: Los miserables
Libro 'Los miserables' eliminado de la lista de libros pedidos
Libro devuelto correctamente
```

El usuario devuelve un libro, el sistema revisa en la lista de prestados que tiene el usuario lo remueve de ahí y lo devuelve a la lista de biblioteca.

7.10. Prueba 10.

```
Menu:

1. Bibliotecario
2. Usuario
3. Inscribirse
4. Salir del sistema

-----

Ingresa una opcion: 4
Saliendo del sistema...
```

Se comprueba que se puede salir del sistema.

8. DECISIÓN DE DISEÑO O CONSIDERACION IMPORTANTE

La principal decisión o consideración importante al hacer el diseño, fue tratar de saber que atributo pertenecía a que clase, y que métodos era pertenecientes a cada clase. Después de considerar eso fue bastante fácil el resto, también se le agregaron mas funciones que nosotros consideramos son necesarios de un sistema básico de biblioteca.

9. PILARES DE POO IMPLEMENTADOS

9.1. Clases: En el código, la clase Biblioteca actúa como un molde que contiene propiedades y métodos para construir objetos relacionados con usuarios y libros. Las listas como nombre, documento, email, etc., representan las características de los usuarios, mientras que las listas como título, categoría, autor, etc., representan las características de los libros. Las clases son la base para crear múltiples instancias u objetos.

9.2. Polimorfismo: Aunque no se implementa explícitamente aquí, el polimorfismo permite definir métodos con el mismo nombre, pero con diferentes comportamientos en clases derivadas. Por ejemplo, podríamos tener un método consultar_libro que se comporte de manera diferente según el tipo de libro o la categoría.

En resumen, este código muestra algunos aspectos de la POO, como la creación de una clase, la definición de propiedades y métodos, y la búsqueda de usuarios y libros. Para una implementación más completa, podríamos explorar más a fondo los pilares de la POO y aplicarlos de manera más extensa en el diseño de la biblioteca virtual.

10. CONCLUSIONES

En conclusión, creemos que este trabajo fue bastante útil para nosotros comprender mejor la programación orientada a objetos, y a perfeccionar mejor nuestros conocimientos sobre este e implementarlos en un caso aplicado a la vida real.

