

UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



Práctica N°2

Docente: Mg. Aldo Hernan Zanabria Galvez
Alumno: Yoel Nhelio Canaza Chagua

Curso:
Programación Orientada a Objetos II

CICLO III – SEMESTRE 2023 – II
PUNO, PERÚ
2023

1. Objetivos

1. Creación de Clases y Objetos: Practicar la creación de clases (Libro y Biblioteca) y la instanciación de objetos a partir de esas clases.
2. Atributos y Métodos: Entender cómo definir atributos (como titulo, autor, anio_publicacion, y disponible) y métodos en una clase.
3. Encapsulación: Aplicar el concepto de encapsulación al usar atributos privados y proporcionar métodos públicos para interactuar con esos atributos.
4. Operaciones Básicas: Realizar operaciones básicas sobre los objetos, como agregar un libro a la biblioteca, buscar un libro por título, prestar un libro y devolver un libro.
5. Manejo de Listas: Trabajar con listas para mantener un conjunto de libros en la biblioteca.
6. Mensaje de Usuario: Incluir mensajes informativos para el usuario, como confirmaciones de acciones realizadas.

2. Desarrollo

```
1 class Libro:
2     def __init__(self, titulo, autor, anio_publicacion):
3         self.titulo = titulo
4         self.autor = autor
5         self.anio_publicacion = anio_publicacion
6         self.disponible = True
7
8     def __str__(self):
9         disponibilidad = "disponible" if self.disponible else "prestado"
10        return f"{self.titulo} - {self.autor} ({self.anio_publicacion}), {
11            disponibilidad}"
12
13 class Biblioteca:
14     def __init__(self):
15         self.libros = []
16
17     def agregar_libro(self, libro):
18         self.libros.append(libro)
19         print(f"Libro '{libro.titulo}' agregado a la biblioteca.")
20
21     def buscar_libro(self, titulo):
22         for libro in self.libros:
23             if libro.titulo == titulo:
24                 return libro
25         return None
26
27     def prestar_libro(self, titulo):
28         libro = self.buscar_libro(titulo)
29         if libro:
30             if libro.disponible:
31                 libro.disponible = False
32                 print(f"Libro '{libro.titulo}' prestado.")
33             else:
34                 print(f"El libro '{libro.titulo}' no est disponible en
este momento.")
35         else:
```

```

35         print(f"No se encontr el libro con t tulo '{titulo}' en la
biblioteca.")
36
37     def devolver_libro(self, titulo):
38         libro = self.buscar_libro(titulo)
39         if libro:
40             if not libro.disponible:
41                 libro.disponible = True
42                 print(f"Libro '{libro.titulo}' devuelto.")
43             else:
44                 print(f"El libro '{libro.titulo}' ya est disponible en la
biblioteca.")
45         else:
46             print(f"No se encontr el libro con t tulo '{titulo}' en la
biblioteca.")
47
48 # Crear algunos libros
49 libro1 = Libro("Python Crash Course", "Eric Matthes", 2015)
50 libro2 = Libro("Clean Code", "Robert C. Martin", 2008)
51 libro3 = Libro("The Art of Computer Programming", "Donald E. Knuth", 1968)
52
53 # Crear una biblioteca
54 biblioteca = Biblioteca()
55
56 # Agregar libros a la biblioteca
57 biblioteca.agregar_libro(libro1)
58 biblioteca.agregar_libro(libro2)
59 biblioteca.agregar_libro(libro3)
60
61 # Realizar operaciones de pr stamo y devoluci n
62 biblioteca.prestar_libro("Clean Code")
63 biblioteca.devolver_libro("Python Crash Course")
64 biblioteca.prestar_libro("The Art of Computer Programming")
65
66 # Mostrar informaci n de los libros en la biblioteca
67 print("\nEstado actual de la biblioteca:")
68 for libro in biblioteca.libros:
69     print(libro)

```

Listing 1: Código base sin modificaciones.

3. Problema

Crea una clase llamada Libro que tenga los siguientes atributos:

- titulo (cadena): el título del libro.
- autor (cadena): el autor del libro.
- anio-publicacion (entero): el año de publicación del libro.
- disponible (booleano): indica si el libro está disponible para ser prestado.
- Luego, crea una clase llamada Biblioteca que tenga una lista de libros y los siguientes métodos:
 - agregar_libro(libro): agrega un libro a la biblioteca.
 - buscar_libro(titulo): busca un libro por su título y devuelve la información del libro si está en la biblioteca.
 - prestar_libro(titulo): marca un libro como prestado si está disponible.
 - devolver_libro(titulo): marca un libro como disponible si estaba prestado.

4. Ejercicios Planteados

El código con las modificaciones guardado en:

<https://github.com/YoelCanaza/poo2-practica-2/blob/2289090a20512564e76f267bf34728590dbd6293/practica-2.py>

```
1 #pr ctica 2
2 #alumno: Canaza Chagua Yoel Nhelio
3
4 import csv
5
6 class ErrorLibroNoEncontrado(Exception):
7     pass
8
9 class ErrorLibroNoDisponible(Exception):
10     pass
11
12 class ErrorLibroYaDisponible(Exception):
13     pass
14
15 class Libro:
16     def __init__(self, titulo, autor, anio_publicacion):
17         self.titulo = titulo
18         self.autor = autor
19
20         self.disponible = True
21
22         if isinstance(anio_publicacion, int) and anio_publicacion > 0:
23             self.anio_publicacion = anio_publicacion
24         else:
25             print("El a o de publicaci n debe ser un n mero entero
positivo.")
26             self.anio_publicacion = int(input('vuelve a ingresar el a o de
publicaci n: '))
27
28     def __str__(self):
29         disponibilidad = "disponible" if self.disponible else "prestado"
30         return f"{self.titulo} - {self.autor} ({self.anio_publicacion}), {
disponibilidad}"
31
32     def actualizar_informacion(self, new_titulo, new_autor, new_year_pub):
33         self.titulo = new_titulo
34         self.autor = new_autor
35         self.anio_publicacion = new_year_pub
36
37 class Biblioteca:
38     def __init__(self):
39         self.libros = []
40
41     def __str__(self):
42         return f"{self.conteo_libros()}"
43
44     def agregar_libro(self, libro):
45         self.libros.append(libro)
46         print(f"Libro '{libro.titulo}' agregado a la biblioteca.")
47
48     def conteo_libros(self):
49         num_libros = len(self.libros)
50         return f"N mero de libros en la biblioteca: {num_libros}"
51
52     def buscar_libro(self, titulo):
53         # ejercicio: hacer que el m todo sea insensible a may sculas
54         # o min sculas al buscar por t tulo
```

```

55
56     for libro in self.libros:
57         if libro.titulo.lower() == titulo.lower():
58             return libro
59         raise ErrorLibroNoEncontrado(f'No se encontr ning n libro
llamado {titulo}')
60         # return None
61
62     def buscar_libro_autor(self, autor):
63         # ejercicio b squeda por autor
64         for libro in self.libros:
65             if libro.autor.lower() == autor.lower():
66                 return libro
67             raise ErrorLibroNoEncontrado(f'No se encontr ning n libro de
autor {autor}')
68
69
70     def actualizar_informacion_libros(self, titulo, new_titulo, new_autor,
new_year_pub):
71         try:
72             libro = self.buscar_libro(titulo)
73         except ErrorLibroNoEncontrado as e:
74             print(f'Error {e}')
75         else:
76             libro.actualizar_informacion(new_titulo, new_autor,
new_year_pub)
77
78         '''if libro:
79         else:
80             print(f"No se encontr el libro con t tulo '{titulo}' en la
biblioteca.")'''
81
82
83
84     def prestar_libro(self, titulo):
85
86         try:
87             libro = self.buscar_libro(titulo)
88         except ErrorLibroNoEncontrado:
89             libro = None
90
91         try:
92             libro_autor = self.buscar_libro_autor(titulo)
93         except ErrorLibroNoEncontrado:
94             libro_autor = None
95
96         try:
97             if libro:
98                 if libro.disponible:
99                     libro.disponible = False
100                     print(f"Libro '{libro.titulo}' prestado.")
101                 else:
102                     raise ErrorLibroNoDisponible(f'El libro {libro.titulo}
no est disponible para ser prestado')
103             elif libro_autor:
104                 if libro_autor.disponible:
105                     libro_autor.disponible = False
106                     print(f"Libro '{libro_autor.titulo}' prestado.")
107                 else:
108                     print(f"El libro de autor '{libro_autor.titulo}' no
est disponible en este momento.")
109                     raise ErrorLibroNoDisponible(f'El libro {libro_autor.
titulo} no est disponible para ser prestado')

```

```

110         else:
111             raise ErrorLibroNoEncontrado(f'No se encontr el libro con
nombre o autor {titulo} en la biblioteca')
112         except (ErrorLibroNoDisponible, ErrorLibroNoEncontrado) as e:
113             print(f'Error {e}')
114
115     def devolver_libro(self, titulo):
116
117         try:
118             libro = self.buscar_libro(titulo)
119         except ErrorLibroNoEncontrado:
120             libro = None
121
122         try:
123             libro_autor = self.buscar_libro(titulo)
124         except ErrorLibroNoEncontrado:
125             libro_autor = None
126
127         try:
128             if libro:
129                 if not libro.disponible:
130                     libro.disponible = True
131                     print(f"Libro '{libro.titulo}' devuelto.")
132                 else:
133                     raise ErrorLibroYaDisponible(f"El libro '{libro.titulo}
ya est disponible en la biblioteca")
134             elif libro_autor:
135                 if not libro_autor.disponible:
136                     libro_autor.disponible = True
137                     print(f"Libro '{libro_autor.titulo}' devuelto.")
138                 else:
139                     raise ErrorLibroYaDisponible(f"El libro '{libro_autor.
titulo} ya est disponible en la biblioteca")
140             else:
141                 raise ErrorLibroNoEncontrado(f"No se encontr el libro con
t tulo o autor '{titulo}' en la biblioteca.")
142         except (ErrorLibroYaDisponible, ErrorLibroNoEncontrado) as e:
143             print(f'Error {e}')
144
145     def ordenar_libros_titulo(self):
146         return sorted(self.libros, key=lambda libro: libro.titulo)
147
148     def ordenar_libros_publicacion(self):
149         return sorted(self.libros, key=lambda libro: libro.anio_publicacion
)
150
151
152     def guardar_biblioteca_csv(self, archivo):
153         try:
154             with open(archivo, 'w', newline='', encoding='utf-8') as f:
155                 writer = csv.writer(f)
156                 # Escribir el encabezado
157                 writer.writerow(["Titulo", "Autor", "Anio de Publicacion",
"Disponible"])
158                 # Escribir los datos de cada libro
159                 for libro in self.libros:
160                     writer.writerow([libro.titulo, libro.autor, libro.
anio_publicacion, libro.disponible])
161                 print("Informaci n de la biblioteca guardada en formato CSV
con xito .")
162         except Exception as e:
163             print(f"Error al guardar la biblioteca en formato CSV: {e}")
164

```

```

165     def cargar_biblioteca_csv(self, archivo):
166         try:
167             with open(archivo, 'r', encoding='utf-8') as f:
168                 reader = csv.reader(f)
169                 # Ignorar el encabezado
170                 next(reader)
171                 # Leer los datos de cada libro
172                 for row in reader:
173                     titulo, autor, anio_publicacion, disponible = row
174                     anio_publicacion = int(anio_publicacion)
175                     disponible = disponible.lower() == 'true'
176                     libro = Libro(titulo, autor, anio_publicacion)
177                     libro.disponible = disponible
178                     self.libros.append(libro)
179                 print("Informaci n de la biblioteca cargada desde formato CSV
con xito .")
180             except FileNotFoundError:
181                 print("No se encontr el archivo. La biblioteca se mantendr
vac a.")
182             except Exception as e:
183                 print(f"Error al cargar la biblioteca desde formato CSV: {e}")
184
185 # Crear algunos libros
186 libro1 = Libro("Python Crash Course", "Eric Matthes", 2015)
187 libro2 = Libro("Clean Code", "Robert C. Martin", 2008)
188 libro3 = Libro("The Art of Computer Programming", "Donald E. Knuth", 1968)
189
190 # Crear una biblioteca
191 biblioteca = Biblioteca()
192
193 # Agregar libros a la biblioteca
194 biblioteca.agregar_libro(libro1)
195 biblioteca.agregar_libro(libro2)
196 biblioteca.agregar_libro(libro3)
197 # Ejercicio: agregar dos libros m s a la biblioteca:
198 libro4 = Libro("Introduction to Algorithms", "Thomas Cormen", 1989)
199 libro5 = Libro("Competitive Programming Book", "Halim", 2018)
200 biblioteca.agregar_libro(libro4)
201 biblioteca.agregar_libro(libro5)
202 biblioteca.prestar_libro("introduction to algorithms")
203 biblioteca.devolver_libro("competitive programming Book")
204
205
206 # Realizar operaciones de pr stamo y devoluci n
207 biblioteca.prestar_libro("Clean Code")
208 biblioteca.devolver_libro("Python Crash Course")
209 biblioteca.prestar_libro("The Art of Computer Programming")
210 biblioteca.devolver_libro("eric matthes")
211
212 # actualizando informacion
213 biblioteca.actualizar_informacion_libros("Competitive Programming Book", "
CompProgBook", "Hal", 8102)
214
215 # Mostrar informaci n de los libros en la biblioteca
216 print("\nEstado actual de la biblioteca:")
217 for libro in biblioteca.libros:
218     print(libro)
219
220 print(biblioteca)
221
222
223
224 print('\n-----')

```

```

225 print('BIBLIOTECA')
226 print('1. Buscar libro por autor')
227 print('2. Buscar libro por t tulo')
228 print('3. Cambiar la informaci n de un libro')
229 print('4. Prestar libro')
230 print('5. Devolver libro')
231
232 opcion = int(input('\n Escribe un n mero del 1 al 5 para seleccionar tu
    opci n: '))
233
234 if opcion == 1:
235     nombre_autor = input('Ingrese el nombre del autor: ')
236     try:
237         libro_encontrado = biblioteca.buscar_libro_autor(nombre_autor)
238     except ErrorLibroNoEncontrado as e:
239         print(f'Error: {e}')
240         libro_encontrado = None
241     else:
242         print('El libro con ese autor es: ', biblioteca.buscar_libro_autor(
            nombre_autor))
243
244 elif opcion == 2:
245     nombre_libro = input('Ingrese el nombre del libro: ')
246     try:
247         libro_encontrado = biblioteca.buscar_libro(nombre_libro)
248     except ErrorLibroNoEncontrado as e:
249         print(f'Error: {e}')
250         libro_encontrado = None
251     else:
252         print('El libro con ese nombre es: ', biblioteca.buscar_libro(
            nombre_libro))
253 elif opcion == 3:
254     nombre_libro = input('Ingrese el nombre del libro: ')
255     print(f"El libro que has escogido es {biblioteca.buscar_libro(
        nombre_libro)}")
256     new_titulo = input('Ingrese el nuevo nombre de libro: ')
257     new_autor = input('Ingrese el nuevo nombre de autor: ')
258     new_year_pub = input('Ingrese el nuevo a nio de publicaci n: ')
259     biblioteca.actualizar_informacion_libros(nombre_libro, new_titulo,
        new_autor, new_year_pub)
260     print(f"Los nuevos datos del libro son {biblioteca.buscar_libro(
        new_titulo)}")
261 elif opcion == 4:
262     for libro in biblioteca.libros:
263         print(libro)
264
265     nombre_titulo_autor = input('\n Ingrese el nombre o el autor del libro
        que quiere prestar: ')
266     biblioteca.prestar_libro(nombre_titulo_autor)
267 elif opcion == 5:
268     for libro in biblioteca.libros:
269         print(libro)
270
271     nombre_titulo_autor = input('\n Ingrese el nombre o el autor del libro
        que quiere devolver: ')
272     biblioteca.devolver_libro(nombre_titulo_autor)
273
274
275 biblioteca_ordenada_titulo = biblioteca.ordenar_libros_titulo()
276 print("\nLIBROS ORDENADOS POR T TULO:")
277 for libro in biblioteca_ordenada_titulo:
278     print(libro)
279

```



```

280 biblioteca_ordenada_publicacion = biblioteca.ordenar_libros_publicacion()
281 print("\nLIBROS ORDENADOS POR FECHA DE PUBLICACION:")
282 for libro in biblioteca_ordenada_publicacion:
283     print(libro)
284
285 print('\n')
286 # Guardamos la biblioteca en un archivo CSV
287 biblioteca.guardar_biblioteca_csv("biblioteca.csv")
288
289
290
291 # Cargamos la informaci3n desde el archivo CSV
292 biblioteca_importada = Biblioteca()
293 biblioteca_importada.cargar_biblioteca_csv("biblioteca.csv")
294
295 print('-' * 8)
296 print('MOSTRANDO LOS LIBROS DE LA BIBLIOTECA IMPORTADA:')
297 for libro in biblioteca_importada.libros:
298     print(libro)

```

Listing 2: C3digo con todas las modificaciones pedidas en los ejercicios

```

yoel@DESKTOP-GUKFQE7:~/unaSistemas/ciclo3/poo2/unidad-2/practica-2$ python3 practica-2.py
Libro 'Python Crash Course' agregado a la biblioteca.
Libro 'Clean Code' agregado a la biblioteca.
Libro 'The Art of Computer Programming' agregado a la biblioteca.
Libro 'Introduction to Algorithms' agregado a la biblioteca.
Libro 'Competitive Programming Book' agregado a la biblioteca.
Libro 'Introduction to Algorithms' prestado.
Error El libro 'Competitive Programming Book ya est3a disponible en la biblioteca
Libro 'Clean Code' prestado.
Error El libro 'Python Crash Course ya est3a disponible en la biblioteca
Libro 'The Art of Computer Programming' prestado.
Error No se encontr3 el libro con t3tulo o autor 'eric matthes' en la biblioteca.

Estado actual de la biblioteca:
Python Crash Course - Eric Matthes (2015), disponible
Clean Code - Robert C. Martin (2008), prestado
The Art of Computer Programming - Donald E. Knuth (1968), prestado
Introduction to Algorithms - Thomas Cormen (1989), prestado
CompProgBook - Hal (8102), disponible
N3mero de libros en la biblioteca: 5

-----
BIBLIOTECA
1. Buscar libro por autor
2. Buscar libro por t3tulo
3. Cambiar la informaci3n de un libro
4. Prestar libro
5. Devolver libro

Escribe un n3mero del 1 al 5 para seleccionar tu opci3n: 2
Ingrese el nombre del libro: clean code
El libro con ese nombre es: Clean Code - Robert C. Martin (2008), prestado

LIBROS ORDENADOS POR T3TULO:
Clean Code - Robert C. Martin (2008), prestado
CompProgBook - Hal (8102), disponible
Introduction to Algorithms - Thomas Cormen (1989), prestado
Python Crash Course - Eric Matthes (2015), disponible
The Art of Computer Programming - Donald E. Knuth (1968), prestado

```

Figura 1:

```

LIBROS ORDENADOS POR FECHA DE PUBLICACIÓN:
The Art of Computer Programming - Donald E. Knuth (1968), prestado
Introduction to Algorithms - Thomas Cormen (1989), prestado
Clean Code - Robert C. Martin (2008), prestado
Python Crash Course - Eric Matthes (2015), disponible
CompProgBook - Hal (8102), disponible

Información de la biblioteca guardada en formato CSV con éxito.
Información de la biblioteca cargada desde formato CSV con éxito.
-----
MOSTRANDO LOS LIBROS DE LA BIBLIOTECA IMPORTADA:
Python Crash Course - Eric Matthes (2015), disponible
Clean Code - Robert C. Martin (2008), prestado
The Art of Computer Programming - Donald E. Knuth (1968), prestado
Introduction to Algorithms - Thomas Cormen (1989), prestado
CompProgBook - Hal (8102), disponible
yoel@DESKTOP-GUKFQE7:~/unaSistemas/ciclo3/poo2/unidad-2/practica-2$

```

Figura 2:

- 4.1. Agregar más libros: Modifica el código para agregar al menos dos libros más a la biblioteca. Realiza operaciones de préstamo y devolución con estos nuevos libros.

```

63 # Ejercicio: agregar dos libros más a la biblioteca:
64 libro4 = Libro("Introduction to Algorithms", "Thomas Cormen", 1989)
65 libro5 = Libro("Competitive Programming Book", "Halim", 2018)
66 biblioteca.agregar_libro(libro4)
67 biblioteca.agregar_libro(libro5)
68 biblioteca.prestar_libro("Introduction to Algorithms")
69 biblioteca.devolver_libro("Competitive Programming Book")
70
71

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

```

Libro 'The Art of Computer Programming' agregado a la biblioteca.
Libro 'Introduction to Algorithms' agregado a la biblioteca.
Libro 'Competitive Programming Book' agregado a la biblioteca.
Libro 'Introduction to Algorithms' prestado.
El libro 'Competitive Programming Book' ya está disponible en la biblioteca.
Libro 'Clean Code' prestado.
El libro 'Python Crash Course' ya está disponible en la biblioteca.
Libro 'The Art of Computer Programming' prestado.

Estado actual de la biblioteca:
Python Crash Course - Eric Matthes (2015), disponible
Clean Code - Robert C. Martin (2008), prestado
The Art of Computer Programming - Donald E. Knuth (1968), prestado
Introduction to Algorithms - Thomas Cormen (1989), prestado
Competitive Programming Book - Halim (2018), disponible
yoel@DESKTOP-GUKFQE7:~/unaSistemas/ciclo3/poo2/unidad-2/practica-2$

```

Figura 3: Caption

- 4.2. Mejorar la búsqueda: Modifica el método `buscar_libro` para que sea insensible a mayúsculas/minúsculas al buscar por título. Implementa una búsqueda que permita encontrar libros por autor.

```
def buscar_libro(self, titulo):
    # ejercicio: hacer que el método sea insensible a mayúsculas
    # o minúsculas al buscar por título
    titulo_lower = titulo.lower()
    for libro in self.libros:
        if libro.titulo.lower() == titulo_lower:
            return libro
    return None
```

Figura 4: Método modificado para que sea insensible a mayúsculas/minúsculas

```
69 biblioteca.agregar_libro(libro4)
70 biblioteca.agregar_libro(libro5)
71 biblioteca.prestar_libro("introduction to algorithms")
72 biblioteca.devolver_libro("competitive programming Book")
73
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

Libro 'The Art of Computer Programming' agregado a la biblioteca.
Libro 'Introduction to Algorithms' agregado a la biblioteca.
Libro 'Competitive Programming Book' agregado a la biblioteca.
Libro 'Introduction to Algorithms' prestado.
El libro 'Competitive Programming Book' ya está disponible en la biblioteca.
Libro 'Clean Code' prestado.
El libro 'Python Crash Course' ya está disponible en la biblioteca.
Libro 'The Art of Computer Programming' prestado.

Estado actual de la biblioteca:
Python Crash Course - Eric Matthes (2015), disponible
Clean Code - Robert C. Martin (2008), prestado
The Art of Computer Programming - Donald E. Knuth (1968), prestado
Introduction to Algorithms - Thomas Cormen (1989), prestado
Competitive Programming Book - Halim (2018), disponible

yoel@DESKTOP-GUKFQE7:~/unaSistemas/ciclo3/poo2/unidad-2/practica-2\$

Figura 5: Gracias a la función modificada, se puede poner como argumentos los títulos de los libros en minúsculas

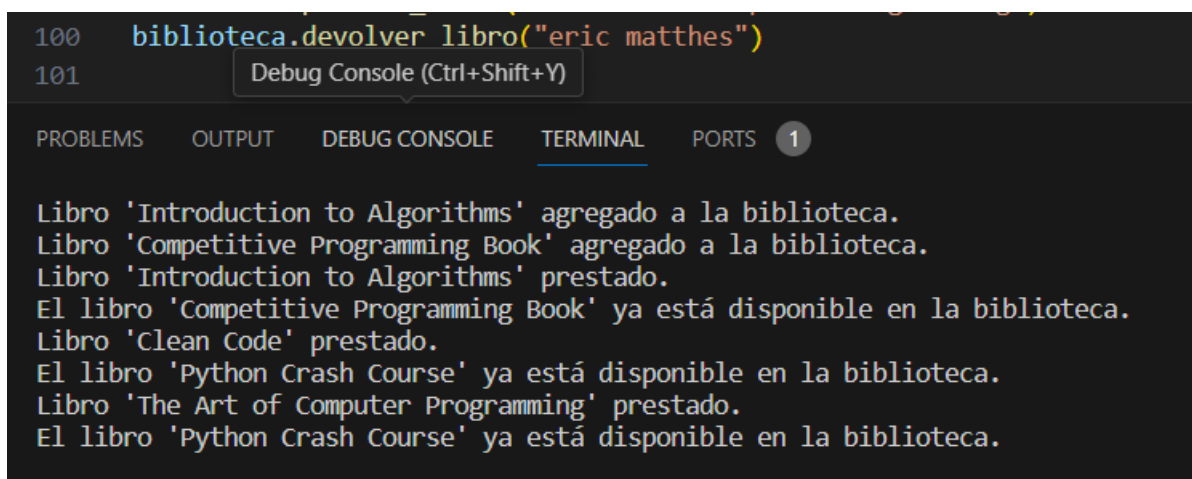
4.2.1. búsqueda por autor

```
def buscar_libro_autor(self, autor):
    # ejercicio búsqueda por autor
    for libro in self.libros:
        if libro.autor.lower() == autor.lower():
            return libro
    return None

def prestar_libro(self, titulo):
    libro = self.buscar_libro(titulo)
    libro_autor = self.buscar_libro_autor(titulo)
    if libro:
        (function) disponible: Any
        if libro.disponible:
            libro.disponible = False
            print(f"Libro '{libro.titulo}' prestado.")
        else:
            print(f"El libro '{libro.titulo}' no está disponible en este momento.")
    elif libro_autor:
        if libro_autor.disponible:
            libro_autor.disponible = False
            print(f"Libro '{libro.titulo}' prestado.")
        else:
            print(f"El libro '{libro_autor.titulo}' no está disponible en este momento.")
    else:
        print(f"No se encontró el libro con título o autor '{titulo}' en la biblioteca.")

def devolver_libro(self, titulo):
    libro = self.buscar_libro(titulo)
    libro_autor = self.buscar_libro_autor(titulo)
    if libro:
        if not libro.disponible:
            libro.disponible = True
            print(f"Libro '{libro.titulo}' devuelto.")
        else:
            print(f"El libro '{libro.titulo}' ya está disponible en la biblioteca.")
    elif libro_autor:
        if not libro_autor.disponible:
            libro_autor.disponible = True
            print(f"Libro '{libro_autor.titulo}' devuelto.")
        else:
            print(f"El libro '{libro_autor.titulo}' ya está disponible en la biblioteca.")
    else:
        print(f"No se encontró el libro con título o autor '{titulo}' en la biblioteca.")
```

Figura 6: También se modificaron las funciones prestar libro y devolver libro para que se pueda poner como argumento autor o nombre del libro



```
100 biblioteca.devolver_libro("eric matthes")
101
```

Debug Console (Ctrl+Shift+Y)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

Libro 'Introduction to Algorithms' agregado a la biblioteca.
Libro 'Competitive Programming Book' agregado a la biblioteca.
Libro 'Introduction to Algorithms' prestado.
El libro 'Competitive Programming Book' ya está disponible en la biblioteca.
Libro 'Clean Code' prestado.
El libro 'Python Crash Course' ya está disponible en la biblioteca.
Libro 'The Art of Computer Programming' prestado.
El libro 'Python Crash Course' ya está disponible en la biblioteca.

Figura 7: Como se puede ver en la imagen, como argumento de la función devolver libro se puso el nombre del autor y en minúsculas. Gracias a las modificaciones, la operación se pudo realizar sin problemas

- 4.3. **Conteo de libros:** Agrega un método a la clase Biblioteca que cuente y muestre la cantidad total de libros en la biblioteca. Modifica la salida para incluir esta información.

```
15 class Biblioteca:
16     def __init__(self):
17         self.libros = []
18
19     def __str__(self):
20         return f"{self.conteo_libros()}"
21
22     def agregar_libro(self, libro):
23         self.libros.append(libro)
24         print(f"Libro '{libro.titulo}' agregado a la biblioteca.")
25
26     def conteo_libros(self):
27         num_libros = len(self.libros)
28         return f"Número de libros en la biblioteca: {num_libros}"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

Libro 'Competitive Programming Book' agregado a la biblioteca.
Libro 'Introduction to Algorithms' prestado.
El libro 'Competitive Programming Book' ya está disponible en la biblioteca.
Libro 'Clean Code' prestado.
El libro 'Python Crash Course' ya está disponible en la biblioteca.
Libro 'The Art of Computer Programming' prestado.
El libro 'Python Crash Course' ya está disponible en la biblioteca.

Estado actual de la biblioteca:
Python Crash Course - Eric Matthes (2015), disponible
Clean Code - Robert C. Martin (2008), prestado
The Art of Computer Programming - Donald E. Knuth (1968), prestado
Introduction to Algorithms - Thomas Cormen (1989), prestado
Competitive Programming Book - Halim (2018), disponible
Número de libros en la biblioteca: 5

yoel@DESKTOP-GUKFQE7:~/unaSistemas/ciclo3/poo2/unidad-2/practica-2\$

Figura 8: Método para contar y mostrar los libros de la biblioteca

- 4.4. **Actualizar información:** Agrega un método a la clase Libro que permita actualizar la información de un libro (título, autor, año de publicación). Utiliza este método para actualizar la información de al menos un libro en la biblioteca.

```
def actualizar_informacion(self, new_titulo, new_autor, new_year_pub):
    self.titulo = new_titulo
    self.autor = new_autor
    self.anio_publicacion = new_year_pub
```

Figura 9: Método en la claes libro para actualizar información

```
def actualizar_informacion_libros(self, titulo, new_titulo, new_autor, new_year_pub):
    libro = self.buscar_libro(titulo)
    if libro:
        libro.actualizar_informacion(new_titulo, new_autor, new_year_pub)
    else:
        print(f"No se encontró el libro con título '{titulo}' en la biblioteca.")
```

Figura 10: Método miembro de Biblioteca para buscar un libro y actualizar su información

```
123 # actualizando informacion
124 biblioteca.actualizar_informacion_libros("Competitive Programming Book", "CompProgBook", "Hal", 8102)
125
126 # Mostrar información de los libros en la biblioteca
127 print("\nEstado actual de la biblioteca:")
128 for libro in biblioteca.libros:
129     print(libro)
130
131 print(biblioteca)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

Libro 'Competitive Programming Book' agregado a la biblioteca.
 Libro 'Introduction to Algorithms' prestado.
 El libro 'Competitive Programming Book' ya está disponible en la biblioteca.
 Libro 'Clean Code' prestado.
 El libro 'Python Crash Course' ya está disponible en la biblioteca.
 Libro 'The Art of Computer Programming' prestado.
 El libro 'Python Crash Course' ya está disponible en la biblioteca.

Estado actual de la biblioteca:
 Python Crash Course - Eric Matthes (2015), disponible
 Clean Code - Robert C. Martin (2008), prestado
 The Art of Computer Programming - Donald E. Knuth (1968), prestado
 Introduction to Algorithms - Thomas Cormen (1989), prestado
 CompProgBook - Hal (8102), disponible
 Número de libros en la biblioteca: 5

yoel@DESKTOP-GUKFQE7:~/unaSistemas/ciclo3/poo2/unidad-2/practica-2\$

Figura 11: Utilización del método para actualizar la información de un libro

4.5. Validaciones adicionales: Mejora el código para incluir validaciones adicionales, como asegurarse de que el año de publicación sea un número positivo.

```
class Libro:
    def __init__(self, titulo, autor, anio_publicacion):
        self.titulo = titulo
        self.autor = autor

        self.disponible = True

        if isinstance(anio_publicacion, int) and anio_publicacion > 0:
            self.anio_publicacion = anio_publicacion
        else:
            print("El año de publicación debe ser un número entero positivo.")
            self.anio_publicacion = int(input('vuelve a ingresar el año de publicación: '))
```

Figura 12: Se valida que el año de publicación sea un entero positivo

- 4.6. Interfaz de usuario mejorada: Crea una interfaz de usuario simple en la consola para que el usuario pueda interactuar con la biblioteca (por ejemplo, seleccionar opciones desde un menú).

```
print('\n-----')
print('BIBLIOTECA')
print('1. Buscar libro por autor')
print('2. Buscar libro por título')
print('3. Cambiar la información de un libro')
print('4. Prestar libro')
print('5. Devolver libro')

opcion = int(input('\n Escribe un número del 1 al 5 para seleccionar tu opción: '))

if opcion == 1:
    nombre_autor = input('Ingrese el nombre del autor: ')
    print('El libro con ese autor es: ', biblioteca.buscar_libro_autor(nombre_autor))
elif opcion == 2:
    nombre_libro = input('Ingrese el nombre del libro: ')
    print('El libro con ese nombre es: ', biblioteca.buscar_libro(nombre_libro))
elif opcion == 3:
    nombre_libro = input('Ingrese el nombre del libro: ')
    print(f"El libro que has escogido es {biblioteca.buscar_libro(nombre_libro)}")
    new_titulo = input('Ingrese el nuevo nombre de libro: ')
    new_autor = input('Ingrese el nuevo nombre de autor: ')
    new_year_pub = input('Ingrese el nuevo año de publicación: ')
    biblioteca.actualizar_informacion_libros(nombre_libro, new_titulo, new_autor, new_year_pub)
    print(f"Los nuevos datos del libro son {biblioteca.buscar_libro(new_titulo)}")
elif opcion == 4:
    for libro in biblioteca.libros:
        print(libro)

    nombre_titulo_autor = input('\n Ingrese el nombre o el autor del libro que quiere prestar: ')
    biblioteca.prestar_libro(nombre_titulo_autor)
elif opcion == 5:
    for libro in biblioteca.libros:
        print(libro)

    nombre_titulo_autor = input('\n Ingrese el nombre o el autor del libro que quiere devolver: ')
    biblioteca.devolver_libro(nombre_titulo_autor)
```

Figura 13:

```
-----
BIBLIOTECA
1. Buscar libro por autor
2. Buscar libro por título
3. Cambiar la información de un libro
4. Prestar libro
5. Devolver libro

Escribe un número del 1 al 5 para seleccionar tu opción: 3
Ingrese el nombre del libro: clean code
El libro que has escogido es Clean Code - Robert C. Martin (2008), prestado
Ingrese el nuevo nombre de libro: cleancode
Ingrese el nuevo nombre de autor: robert
Ingrese el nuevo año de publicación: 3000
Los nuevos datos del libro son cleancode - robert (3000), prestado
yoel@DESKTOP-GUKFQE7:~/unaSistemas/ciclo3/poo2/unidad-2/practica-2$
```

Figura 14:

- 4.7. Manejo de excepciones: Implementa manejo de excepciones para situaciones como buscar un libro que no está en la biblioteca o devolver un libro que ya está disponible.

```
3
4 class ErrorLibroNoEncontrado(Exception):
5     pass
6
7 class ErrorLibroNoDisponible(Exception):
8     pass
9
10 class ErrorLibroYaDisponible(Exception):
11     pass
12
```

Figura 15: Creamos nuevas clases de excepciones que heredan de 'Exception'

```
def buscar_libro(self, titulo):
    # ejercicio: hacer que el método sea insensible a mayúsculas
    # o minúsculas al buscar por título

    for libro in self.libros:
        if libro.titulo.lower() == titulo.lower():
            return libro
    raise ErrorLibroNoEncontrado(f'No se encontró ningún libro llamado {titulo}')
    # return None

def buscar_libro_autor(self, autor):
    # ejercicio búsqueda por autor
    for libro in self.libros:
        if libro.autor.lower() == autor.lower():
            return libro
    raise ErrorLibroNoEncontrado(f'No se encontró ningún libro de autor {autor}')
```

Figura 16: Las funciones de búsqueda por título y búsqueda por autor se modifican para lanzar excepciones en caso de que no se encuentre el libro en la biblioteca

```
def actualizar_informacion_libros(self, titulo, new_titulo, new_autor, new_year_pub):
    try:
        libro = self.buscar_libro(titulo)
    except ErrorLibroNoEncontrado as e:
        print(f'Error {e}')
    else:
        libro.actualizar_informacion(new_titulo, new_autor, new_year_pub)

    '''if libro:
    else:
        print(f"No se encontró el libro con título '{titulo}' en la biblioteca.")'''
```

Figura 17: La función para actualizar información de los libros también se modifica para lanzar y atrapar la excepción ErrorLibroNoEncontrado.


```

def prestar_libro(self, titulo):
    try:
        libro = self.buscar_libro(titulo)
    except ErrorLibroNoEncontrado:
        libro = None

    try:
        libro_autor = self.buscar_libro_autor(titulo)
    except ErrorLibroNoEncontrado:
        libro_autor = None

    try:
        if libro:
            if libro.disponible:
                libro.disponible = False
                print(f"Libro '{libro.titulo}' prestado.")
            else:
                raise ErrorLibroNoDisponible(f'El libro {libro.titulo} no está disponible para ser prestado')
        elif libro_autor:
            if libro_autor.disponible:
                libro_autor.disponible = False
                print(f"Libro '{libro_autor.titulo}' prestado.")
            else:
                print(f"El libro de autor '{libro_autor.titulo}' no está disponible en este momento.")
                raise ErrorLibroNoDisponible(f'El libro {libro_autor.titulo} no está disponible para ser prestado')
        else:
            raise ErrorLibroNoEncontrado(f'No se encontró el libro con nombre o autor {titulo} en la biblioteca')
    except (ErrorLibroNoDisponible, ErrorLibroNoEncontrado) as e:
        print(f'Error {e}')

```

Figura 18: Manejo de excepciones en caso de que se intente tomar prestado un libro que no se encuentra disponible

```

def devolver_libro(self, titulo):
    try:
        libro = self.buscar_libro(titulo)
    except ErrorLibroNoEncontrado:
        libro = None

    try:
        libro_autor = self.buscar_libro_autor(titulo)
    except ErrorLibroNoEncontrado:
        libro_autor = None

    try:
        if libro:
            if not libro.disponible:
                libro.disponible = True
                print(f"Libro '{libro.titulo}' devuelto.")
            else:
                raise ErrorLibroYaDisponible(f"El libro '{libro.titulo}' ya está disponible en la biblioteca")
        elif libro_autor:
            if not libro_autor.disponible:
                libro_autor.disponible = True
                print(f"Libro '{libro_autor.titulo}' devuelto.")
            else:
                raise ErrorLibroYaDisponible(f"El libro '{libro_autor.titulo}' ya está disponible en la biblioteca")
        else:
            raise ErrorLibroNoEncontrado(f"No se encontró el libro con título o autor '{titulo}' en la biblioteca.")
    except (ErrorLibroYaDisponible, ErrorLibroNoEncontrado) as e:
        print(f'Error {e}')

```

Figura 19: Manejo de excepciones en caso de que se intente devolver un libro que ya está disponible

- 4.8. Ordenar libros: Agrega un método a la clase Biblioteca que permita ordenar los libros por título o por año de publicación. Muestra la lista ordenada de libros en la biblioteca.

```
def ordenar_libros_titulo(self):  
    return sorted(self.libros, key=lambda libro: libro.titulo)  
  
def ordenar_libros_publicacion(self):  
    return sorted(self.libros, key=lambda libro: libro.anio_publicacion)
```

Figura 20: Métodos para ordenar los libros por título o por año de publicación

```
238 biblioteca_ordenada_titulo = biblioteca.ordenar_libros_titulo()  
239 print("\nLIBROS ORDENADOS POR TÍTULO:")  
240 for libro in biblioteca_ordenada_titulo:  
241     print(libro)  
242  
243 biblioteca_ordenada_publicacion = biblioteca.ordenar_libros_publicacion()  
244 print("\nLIBROS ORDENADOS POR FECHA DE PUBLICACIÓN:")  
245 for libro in biblioteca_ordenada_publicacion:  
246     print(libro)  
247
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

LIBROS ORDENADOS POR TÍTULO:
Clean Code - Robert C. Martin (2008), prestado
CompProgBook - Hal (8102), disponible
Introduction to Algorithms - Thomas Cormen (1989), prestado
Python Crash Course - Eric Matthes (2015), disponible
The Art of Computer Programming - Donald E. Knuth (1968), prestado

LIBROS ORDENADOS POR FECHA DE PUBLICACIÓN:
The Art of Computer Programming - Donald E. Knuth (1968), prestado
Introduction to Algorithms - Thomas Cormen (1989), prestado
Clean Code - Robert C. Martin (2008), prestado
Python Crash Course - Eric Matthes (2015), disponible
CompProgBook - Hal (8102), disponible

yoel@DESKTOP-GUKFQE7:~/unaSistemas/ciclo3/poo2/unidad-2/practica-2\$

Figura 21: Mostrando las listas ordenadas por título y por año de publicación

- 4.9. Guardar y cargar datos: Agrega métodos a la clase Biblioteca para guardar y cargar la información de la biblioteca desde y hacia un archivo, respectivamente.

```
3  
4 import csv  
5
```

Figura 22: Importamos la librería CSV para poder guardar y cargar información en archivos .csv

```

def guardar_biblioteca_csv(self, archivo):
    try:
        with open(archivo, 'w', newline='', encoding='utf-8') as f:
            writer = csv.writer(f)
            # Escribir el encabezado
            writer.writerow(["Titulo", "Autor", "Anio de Publicacion", "Disponible"])
            # Escribir los datos de cada libro
            for libro in self.libros:
                writer.writerow([libro.titulo, libro.autor, libro.anio_publicacion, libro.disponible])
            print("Información de la biblioteca guardada en formato CSV con éxito.")
    except Exception as e:
        print(f"Error al guardar la biblioteca en formato CSV: {e}")

def cargar_biblioteca_csv(self, archivo):
    try:
        with open(archivo, 'r', encoding='utf-8') as f:
            reader = csv.reader(f)
            # Ignorar el encabezado
            next(reader)
            # Leer los datos de cada libro
            for row in reader:
                titulo, autor, anio_publicacion, disponible = row
                anio_publicacion = int(anio_publicacion)
                disponible = disponible.lower() == 'true'
                libro = Libro(titulo, autor, anio_publicacion)
                libro.disponible = disponible
                self.libros.append(libro)
            print("Información de la biblioteca cargada desde formato CSV con éxito.")
    except FileNotFoundError:
        print("No se encontró el archivo. La biblioteca se mantendrá vacía.")
    except Exception as e:
        print(f"Error al cargar la biblioteca desde formato CSV: {e}")

```

Figura 23: Métodos para guardar y cargar la información de la biblioteca desde y hacia un archivo .csv

```

285     print('\n')
286     # Guardamos la biblioteca en un archivo CSV
287     biblioteca.guardar_biblioteca_csv("biblioteca.csv")
288
289

```

Figura 24: Uso del método `guardar_biblioteca_csv` dentro del programa

```

x biblioteca.csv
1  Titulo,Autor,Anio de Publicacion,Disponible
2  Python Crash Course,Eric Matthes,2015,True
3  Clean Code,Robert C. Martin,2008,False
4  The Art of Computer Programming,Donald E. Knuth,1968,False
5  Introduction to Algorithms,Thomas Cormen,1989,False
6  CompProgBook,Hal,8102,True
7

```

Figura 25: Archivo .csv con la información de la biblioteca dentro

	A	B	C	D	E
1	Título	Autor	Año de Publicación	Disponible	
2	Python Crash Course	Eric Matthes	2015	VERDADERO	
3	Clean Code	Robert C. Martin	2008	FALSO	
4	The Art of Computer Programming	Donald E. Knuth	1968	FALSO	
5	Introduction to Algorithms	Thomas Cormen	1989	FALSO	
6	CompProgBook	Hal	8102	VERDADERO	
7					
8					
9					
10					
11					
12					
13					

Figura 26: Archivo .csv abierto con el programa 'Excel'

```

290
291 # Cargamos la información desde el archivo CSV
292 biblioteca_importada = Biblioteca()
293 biblioteca_importada.cargar_biblioteca_csv("biblioteca.csv")
294
295 print('-' * 8)
296 print('MOSTRANDO LOS LIBROS DE LA BIBLIOTECA IMPORTADA:')
297 for libro in biblioteca_importada.libros:
298     print(libro)

```

Figura 27: Creando una nueva biblioteca para asignarle los datos desde el .csv

```

Información de la biblioteca guardada en formato CSV con éxito.
Información de la biblioteca cargada desde formato CSV con éxito.
-----
MOSTRANDO LOS LIBROS DE LA BIBLIOTECA IMPORTADA:
Python Crash Course - Eric Matthes (2015), disponible
Clean Code - Robert C. Martin (2008), prestado
The Art of Computer Programming - Donald E. Knuth (1968), prestado
Introduction to Algorithms - Thomas Cormen (1989), prestado
CompProgBook - Hal (8102), disponible
yoel@DESKTOP-GUKFQE7:~/unaSistemas/ciclo3/poo2/unidad-2/practica-2$

```

Figura 28: