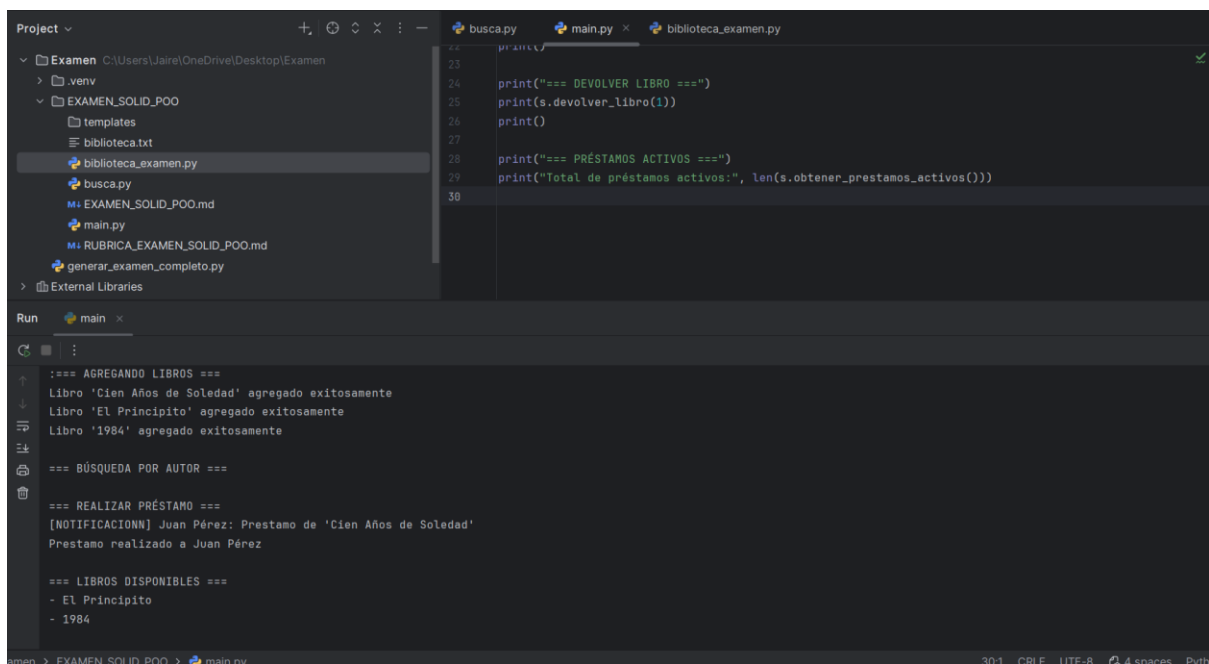


## Primer Ejercicio:

Resumen: Antes buscar un libro tenía toda la lógica en un solo método y había que modificarlo para cada búsqueda ahora creé un archivo busca.py con un buscador que puede buscar de 4 formas: por título, por autor, por ISBN y por disponible.



The screenshot shows a Python IDE with a project named 'Examen' located at 'C:\Users\Jaire\OneDrive\Desktop\Examen'. The project structure includes a '.venv' folder, an 'EXAMEN\_SOLID\_POO' folder containing 'templates', 'biblioteca.txt', 'busca.py', 'EXAMEN\_SOLID\_POO.md', 'main.py', and 'RUBRICA\_EXAMEN\_SOLID\_POO.md', and a 'generar\_examen\_completo.py' file. The 'Run' panel shows the output of the 'main.py' script, which includes sections for adding books, searching by author, making a loan, and listing available books.

```
Project
├── Examen
│   ├── .venv
│   ├── EXAMEN_SOLID_POO
│   │   ├── templates
│   │   ├── biblioteca.txt
│   │   ├── busca.py
│   │   ├── EXAMEN_SOLID_POO.md
│   │   ├── main.py
│   │   └── RUBRICA_EXAMEN_SOLID_POO.md
│   └── generar_examen_completo.py
└── External Libraries

Run
main
:=== AGREGANDO LIBROS ===
Libro 'Cien Años de Soledad' agregado exitosamente
Libro 'El Principito' agregado exitosamente
Libro '1984' agregado exitosamente

=== BÚSQUEDA POR AUTOR ===

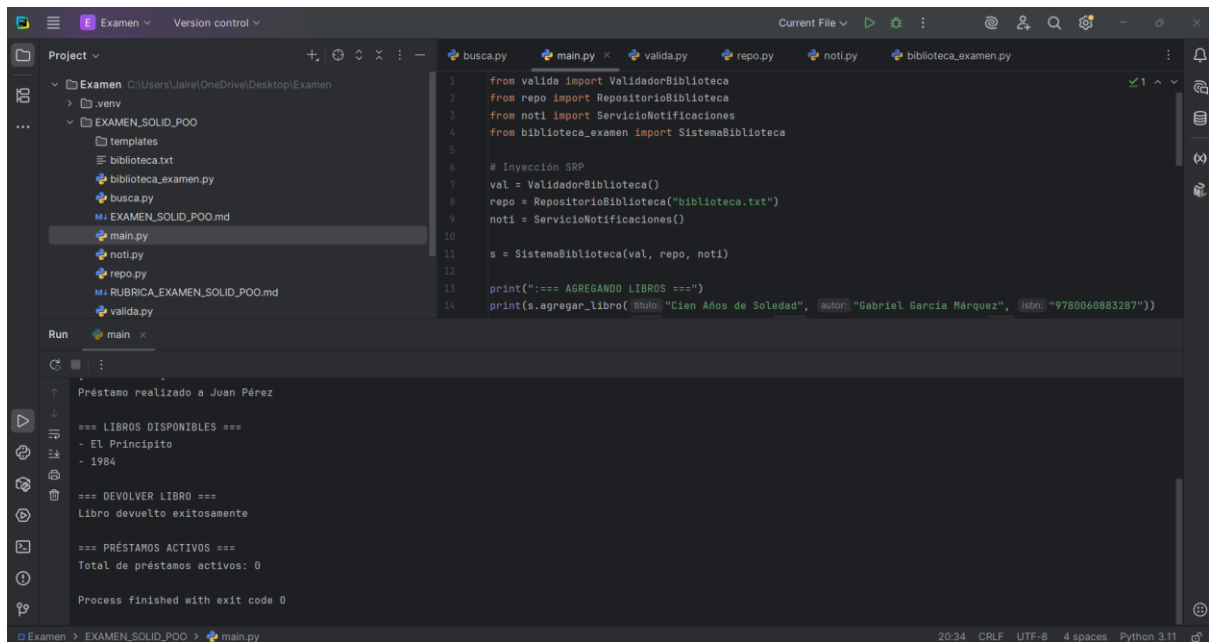
=== REALIZAR PRÉSTAMO ===
[NOTIFICACIONN] Juan Pérez: Prestamo de 'Cien Años de Soledad'
Prestamo realizado a Juan Pérez

=== LIBROS DISPONIBLES ===
- El Principito
- 1984
```

## Segundo ejercicio:

El SistemaBiblioteca se aventaba el jale de validar, guardar datos y mandar las notificaciones como uno solo que hacía todo ahora le separé las responsabilidades en 3 archivos valida.py que valida libros y usuarios, repo.py que guarda y carga datos en biblioteca.txt, y noti.py que envía notificaciones. El SistemaBiblioteca solo los coordina.

## Resumen:



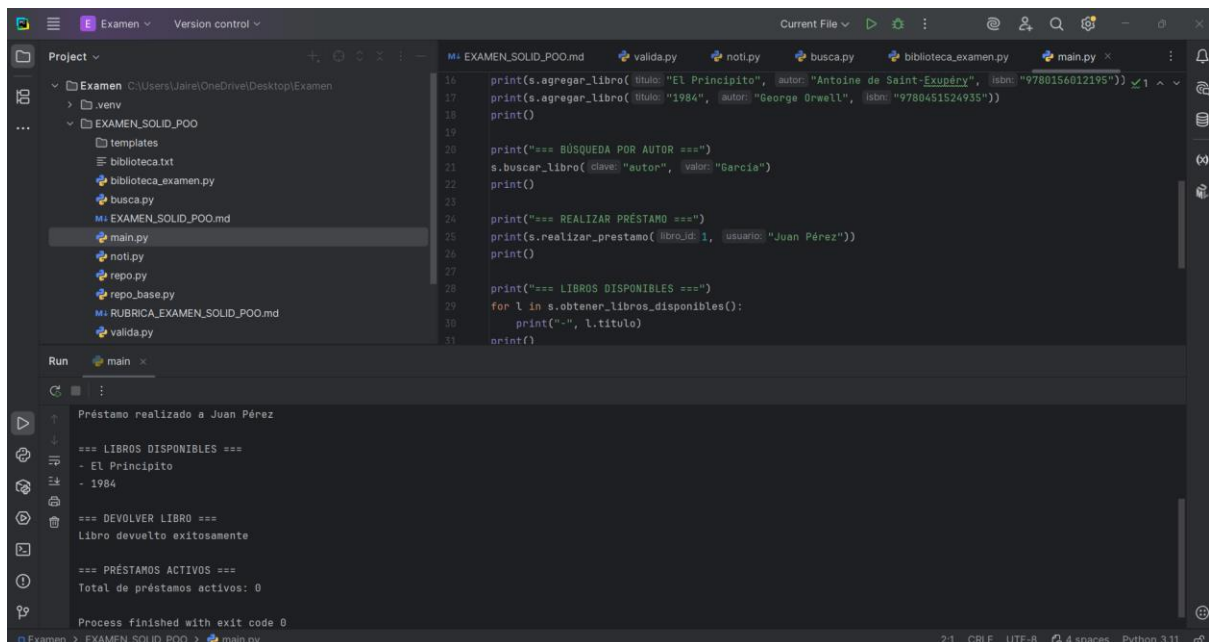
```
1 from valida import ValidadorBiblioteca
2 from repo import RepositorioBiblioteca
3 from noti import ServicioNotificaciones
4 from biblioteca_examen import SistemaBiblioteca
5
6 # Inyección SRP
7 val = ValidadorBiblioteca()
8 repo = RepositorioBiblioteca("biblioteca.txt")
9 noti = ServicioNotificaciones()
10
11 s = SistemaBiblioteca(val, repo, noti)
12
13 print("=== AGREGANDO LIBROS ===")
14 print(s.agregar_libro(titulo="Cien Años de Soledad", autor="Gabriel García Márquez", isbn="9780060883287"))
```

Run console output:

```
Préstamo realizado a Juan Pérez
=== LIBROS DISPONIBLES ===
- El Principito
- 1984
=== DEVOLVER LIBRO ===
Libro devuelto exitosamente
=== PRÉSTAMOS ACTIVOS ===
Total de préstamos activos: 0
Process finished with exit code 0
```

### Tercer ejercicio:

**Resumen:** el tuyo profe estaba amarrado a una implementación de repositorio concreta porque dependía de cómo se guardaba en lugar de una abstracción creé una interfaz para el repo ya existente que define el guardar y cargar. Se lo implementé a la ya creada. Modifiqué el de SistemaBiblioteca para que ya no dependa de un repositorio fijo y que reciba uno en cualquier momento ahora SistemaBiblioteca no necesita saber cómo se guarda la información. puede trabajar con un repositorio de archivo, de memoria o de base de datos sin cambios internos.



```
16 print(s.agregar_libro(titulo="El Principito", autor="Antoine de Saint-Exupéry", isbn="9780156012195"))
17 print(s.agregar_libro(titulo="1984", autor="George Orwell", isbn="9780451524935"))
18 print()
19
20 print("=== BÚSQUEDA POR AUTOR ===")
21 s.buscar_libro(clave="autor", valor="García")
22 print()
23
24 print("=== REALIZAR PRÉSTAMO ===")
25 print(s.realizarprestamo(libroId=1, usuario="Juan Pérez"))
26 print()
27
28 print("=== LIBROS DISPONIBLES ===")
29 for l in s.obtener_libros_disponibles():
30     print("-", l.titulo)
31 print()
```

Run console output:

```
Préstamo realizado a Juan Pérez
=== LIBROS DISPONIBLES ===
- El Principito
- 1984
=== DEVOLVER LIBRO ===
Libro devuelto exitosamente
=== PRÉSTAMOS ACTIVOS ===
Total de préstamos activos: 0
Process finished with exit code 0
```

**Resumen de como funciona el proyecto:**

En este proyecto la parte principal está en biblioteca\_examen.py, ahí está el sistema que hace todo lo importante: agregar libros, prestar, devolver y mostrar los mensajes principales. Las búsquedas se hacen en busca.py, donde puse las formas de buscar (por título, autor, ISBN y si está disponible). Lo de guardar y cargar datos está en repo.py y repo\_base.py, que trabajan con el archivo biblioteca.txt para no perder los libros ni los préstamos. En valida.py se revisa que los datos que metas estén bien, como que el libro tenga título, autor o que el usuario no esté vacío. En noti.py se mandan las notificaciones cuando alguien pide un libro, y sale el aviso en pantalla. El archivo biblioteca.txt es como la base de datos simple donde se guarda todo. Y en main.py es donde se junta todo para probar, ahí se crean los objetos y salen los prints como Libro agregado, Préstamo realizado o Libro devuelto.