

DOCUMENTACIÓN DEL SISTEMA DE CONTROL BIBLIOTECARIO Y REPORTES

Requisitos

- Python 3.7+
 - Flask
 - pandas
 - openpyxl
 - Waitress
 - Base de datos compatible con SQL (ej. SQL Server, SQLite con adaptaciones)
-

1. `/buscar` – Buscar libros

- **Métodos:** GET, POST
 - **Función:** Permite buscar libros en la base de datos por título, autor o código de barras.
 - **Detalles:**
 - Se hace una consulta SQL dinámica con filtros condicionales.
 - Se utiliza COLLATE Latin1_General_CI_AI para hacer búsquedas insensibles a mayúsculas y acentos.
 - Muestra resultados en `resultados_busqueda.html`.
-

2. `/borrar_libro` – Eliminar un libro

- **Métodos:** GET, POST
 - **Función:** Elimina un libro usando su código de barras.
 - **Detalles:**
 - Si el libro existe, se elimina; si no, se muestra un mensaje de error.
 - Usa `eliminar_libro.html` para mostrar el resultado.
-

3. `/buscar_libro/<codigo_barras>` – Obtener datos de un libro

- **Método:** GET
- **Función:** Retorna los datos de un libro en formato JSON a partir del código de barras.
- **Salida:**

- JSON con los campos del libro o {'libro': None} si no se encuentra.
-

4. /modificar_libro/<codigo_barras> – Actualizar datos de un libro

- **Método:** PUT
 - **Función:** Modifica los datos de un libro, validando que el nuevo código de barras (si se cambia) no exista previamente.
 - **Entradas JSON:** campos como titulo, autor, anio, etc.
 - **Salida:** JSON con mensaje de éxito o error.
-

5. /reporte – Generar reporte de libros prestados

- **Métodos:** GET, POST
 - **Función:** Exporta un Excel con libros prestados entre dos fechas.
 - **Detalles:**
 - Incluye filtros, estilos en cabeceras, y contador de libros.
 - Guarda el archivo en memoria y lo envía como descarga (.xlsx).
 - Usa la tabla prestamos y une con Libros.
-

6. /reporte2 – Reporte de uso de computadoras

- **Métodos:** GET, POST
 - **Función:** Genera un Excel con registros de uso de computadoras por fecha.
 - **Detalles:**
 - Exporta con formato, filtros y contador de registros (12).
 - Usa la tabla computadoras.
-

7. /descargar-archivo – Descargar manual principal

- **Método:** GET
 - **Función:** Descarga un archivo PDF ubicado en la carpeta /archivos.
-

8. /descargar-instalador – Descargar manual para importar

- **Método:** GET
 - **Función:** Descarga el segundo manual en PDF desde la carpeta /archivos.
-

9. /importar_alumnos – Importar archivo de alumnos

- **Métodos:** GET, POST
 - **Función:** Permite subir y procesar un archivo .xlsx o .csv con datos de alumnos.
 - **Validaciones:**
 - Matrícula numérica.
 - Carrera válida (LCC, LF, etc.).
 - Género M o F.
 - **Importa usando MERGE:** actualiza si ya existe o inserta si es nuevo.
-

10. /descargar_plantilla – Descargar plantilla de importación de alumnos

- **Método:** GET
 - **Función:** Genera y descarga una plantilla .xlsx para importar alumnos.
 - **Campos incluidos:** Matricula, Nombre, Carrera, Genero, Semestre.
-



warm_up() – Precalienta dependencias

- **Propósito:** Valida conexión a la base de datos y carga de librerías (pandas, openpyxl) antes de arrancar el servidor.
-

Servidor (`waitress`)

```
python
CopiarEditar
if __name__ == '__main__':
    warm_up()
    from waitress import serve
    serve(app, host='127.0.0.1', port=5000)
```

- Inicia el servidor en `localhost:5000` usando Waitress.
 - `warm_up()` se ejecuta antes para evitar errores de dependencias no cargadas.
-

Notas adicionales

- **Uso de Pandas:** facilita transformación y validación de datos.
- **openpyxl:** para crear reportes con formato visual.
- **MERGE SQL:** evita duplicados y mantiene actualizados los datos de alumnos.
- **Seguridad:** falta autenticación y control de acceso. Recomendado para versiones futuras.