

Casos y Escenarios de Pruebas API-BOOKINDER

Universidad De Medellín.

Facultad de Ingeniería.

Validación y Verificación de Software.

Sara Velásquez, Diego Alejandro Durango,

Santiago Franco, Alexander Restrepo.

2024.

Control de versiones

Fecha	Actividad	Descripción	Autor	Versión
13/03/2023	Ejecución Pruebas Caja Negra, Caja blanca	Ejecución detallada de pruebas	Equipo QA Bookinder	1.0

Contenido

Casos y Escenarios de Pruebas API-BOOKINDER	1
Control de versiones	2
Contenido	3
Pruebas Caja Negra	4
Controlador Abstracto	4
Autenticación	5
Libros	11
Pruebas Caja Blanca	19
Libros	19

Pruebas Caja Negra

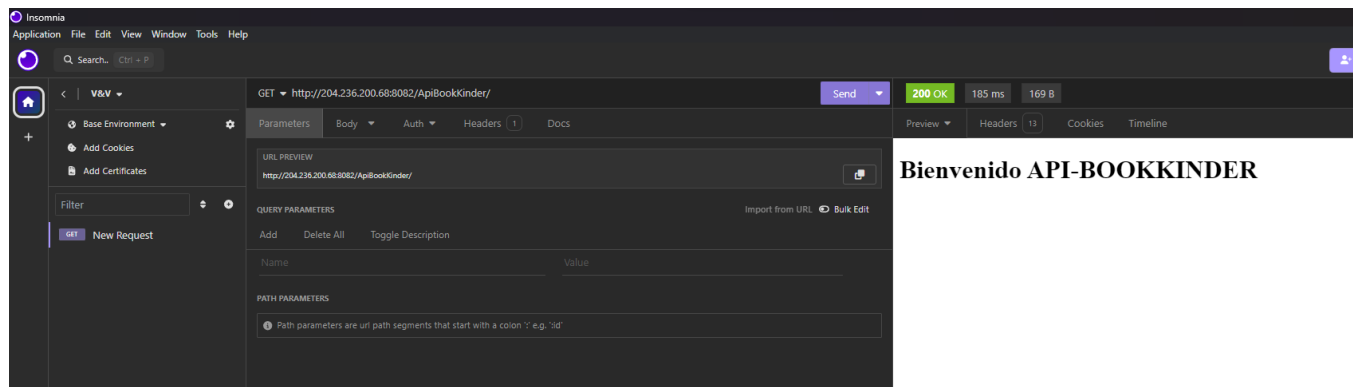
Para la ejecución de estas pruebas, se documenta la evidencia obtenida y los parámetros utilizados. Para obtener información detallada sobre cada prueba de caja negra, se recomienda revisar el archivo "Contexto Pruebas API-BOOKINDER.docx", específicamente en la sección de "Criterios de aceptación, criterios de entrada y salida"; estas pruebas fueron realizadas con el Software desplegado en la nube de AWS.

Controlador Abstracto

CA-001:

Escenario 01:

- **URL:** <http://204.236.200.68:8082/ApiBookKinder/>
- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:** NA.
- **Body:** NA.
- **Evidencia:**



Escenario 02:

- **URL:** <http://204.236.200.68:8082/ApiBookKinder/?param=Hola123/7788>
- **Path variables:** 7788
- **Query parameters:** ?param=Hola123.
- **Headers:**

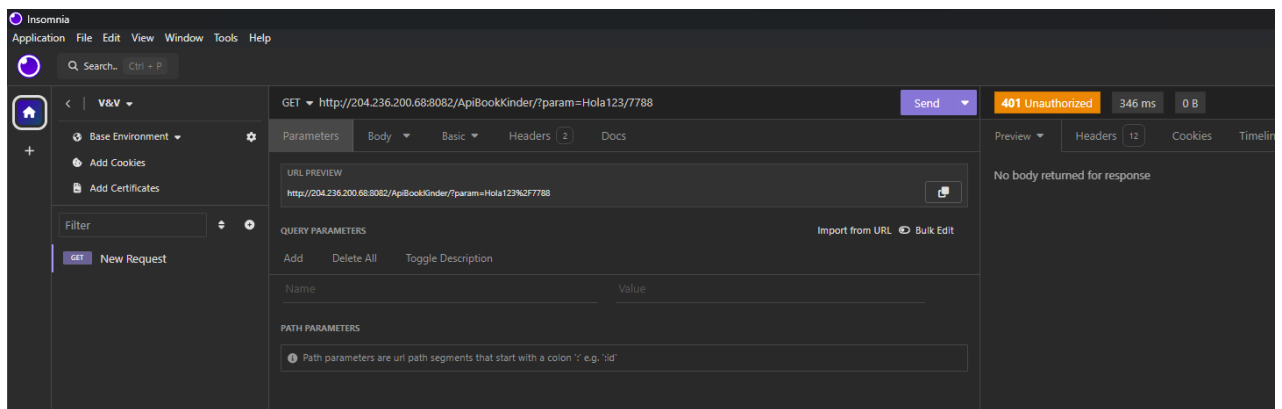
Add Delete All Toggle Description	
User-Agent	insomnia/8.6.1
token	12345

- **Auth:**

Parameters	Body ▾	Basic ▾	Headers 2	Docs
ENABLED	<input checked="" type="checkbox"/>			
USERNAME	santiago			
PASSWORD	franco			
USE ISO 859-1	<input type="checkbox"/>			

- **Body: NA.**

- **Evidencia:**



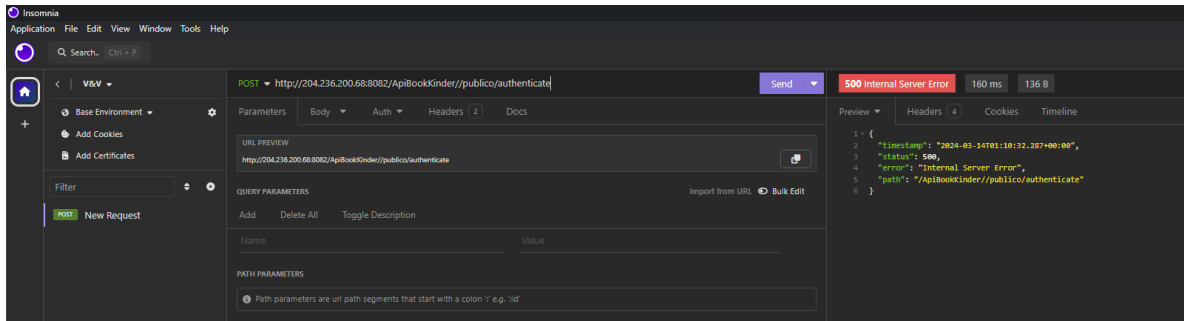
Autenticación

Auth-001:

Escenario 01:

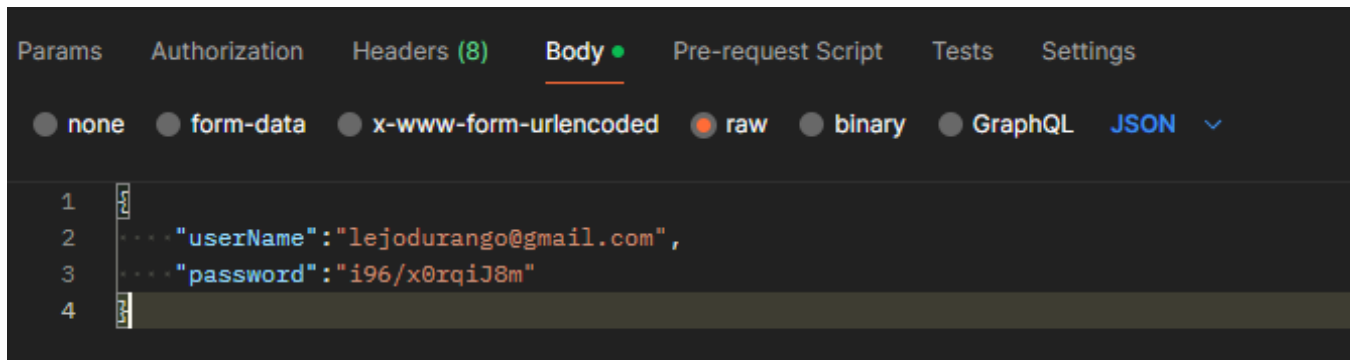
- **URL:** <http://204.236.200.68:8082/ApiBookKinder/publico/authenticate>

- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:** NA.
- **Body:** NA.
- **Evidencia:**



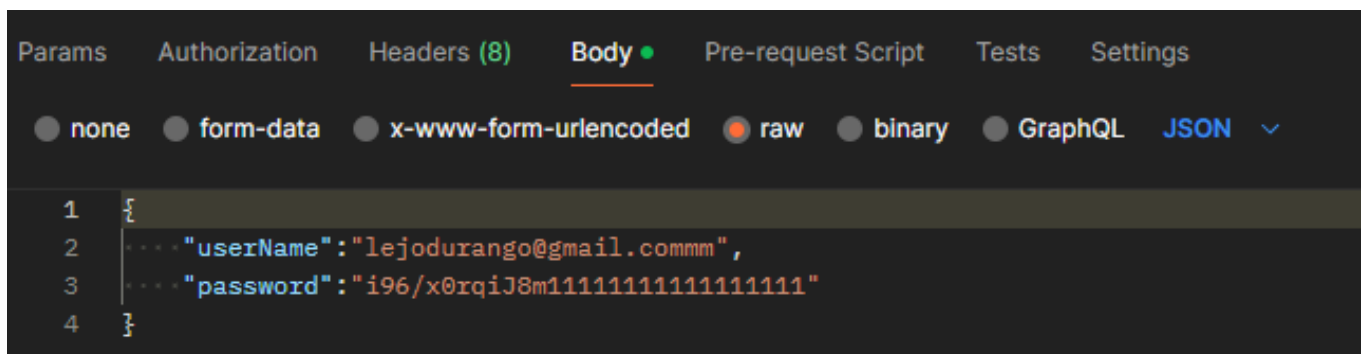
Escenario 02:

- **URL:** http://204.236.200.68:8082/ApiBookKinder/publico/authenticate
- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:** NA.
- **Body:**



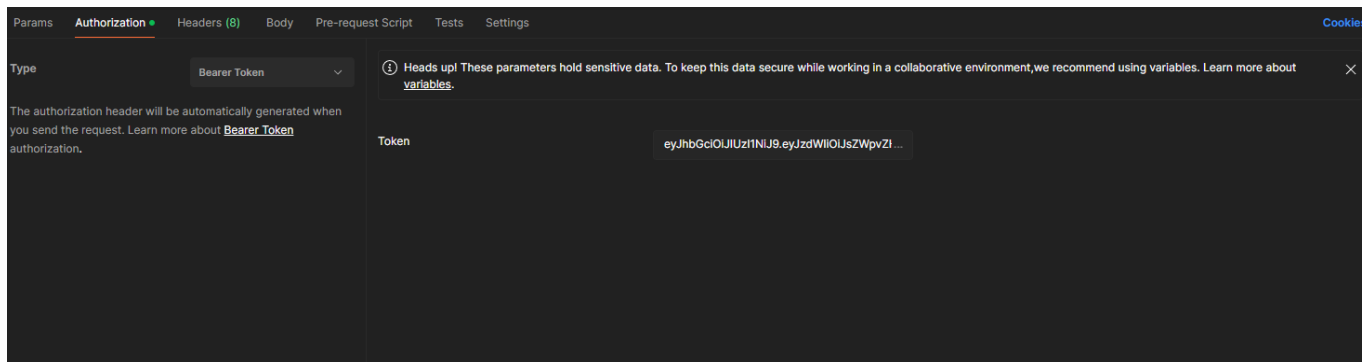
- **Evidencia:**

- **URL:** http://204.236.200.68:8082/ApiBookKinder/publico/authenticate
- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:** NA.
- **Body:** NA.

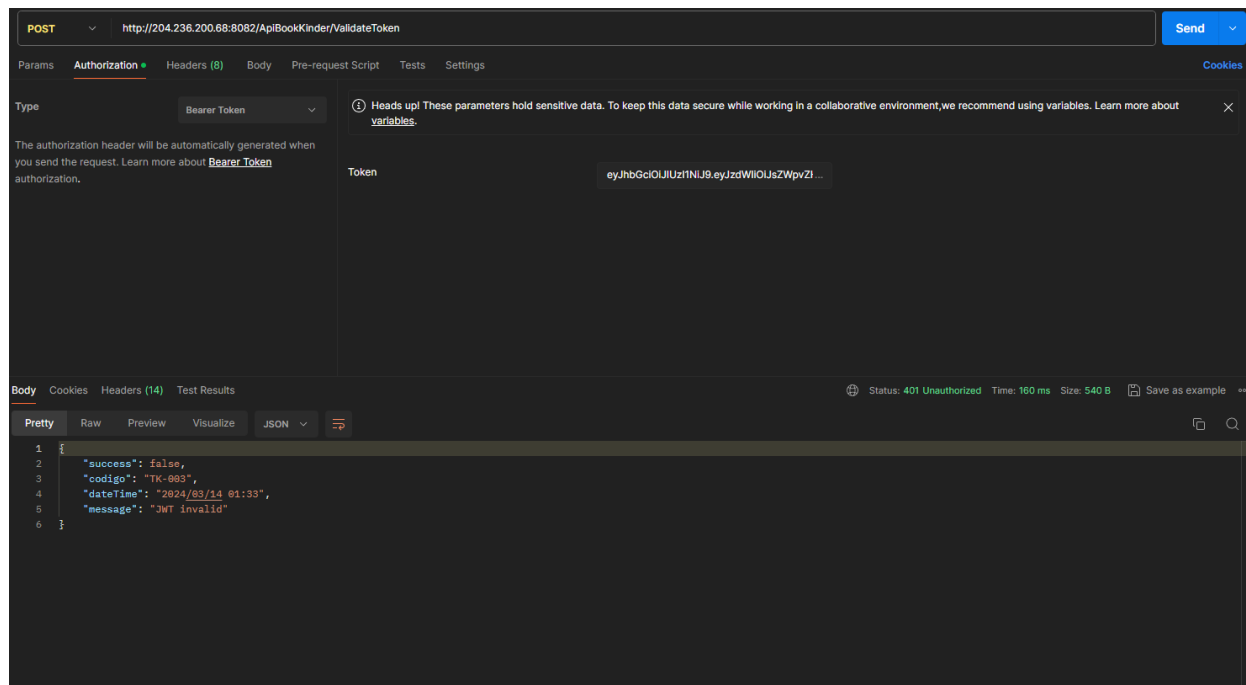


Escenario 01:

- **URL:** http://204.236.200.68:8082/ApiBookKinder/ValidateToken
- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:**

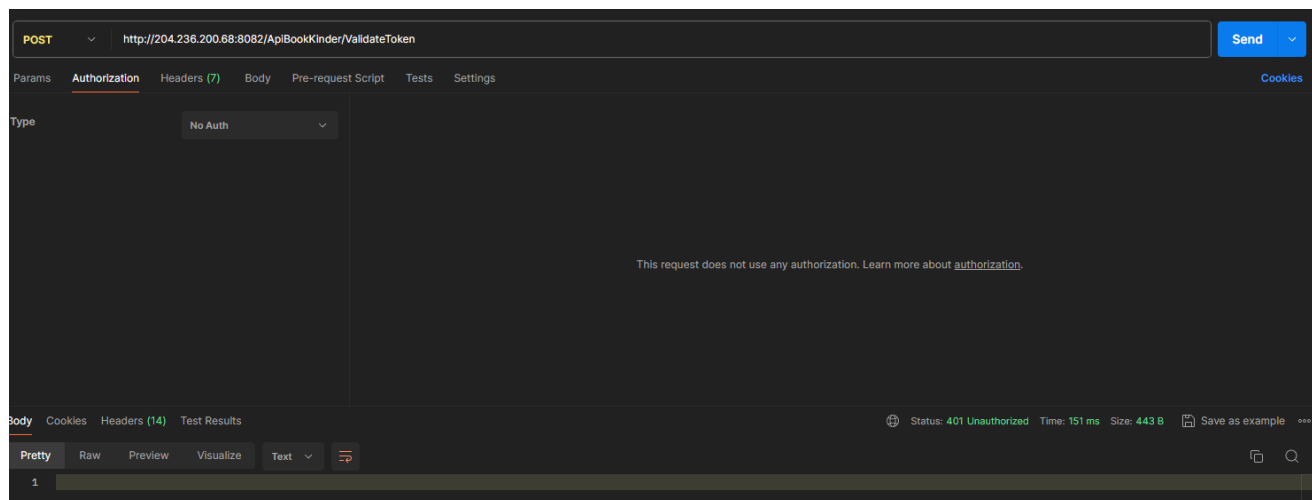


- **Body:** NA.
- **Evidencia:**



Escenario 03:

- **URL:** http://204.236.200.68:8082/ApiBookKinder/ValidateToken
- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:** NA.
- **Body:** NA.
- **Evidencia:**

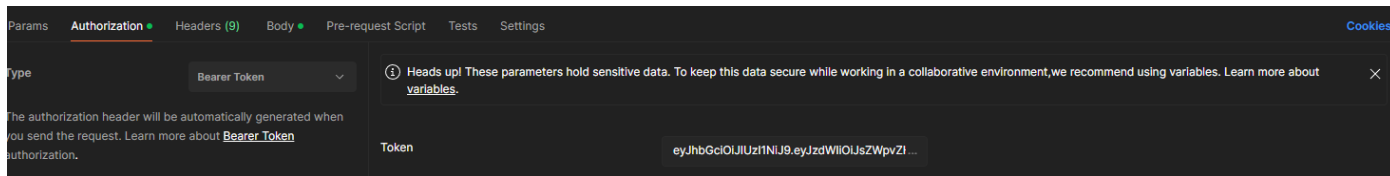


Libros

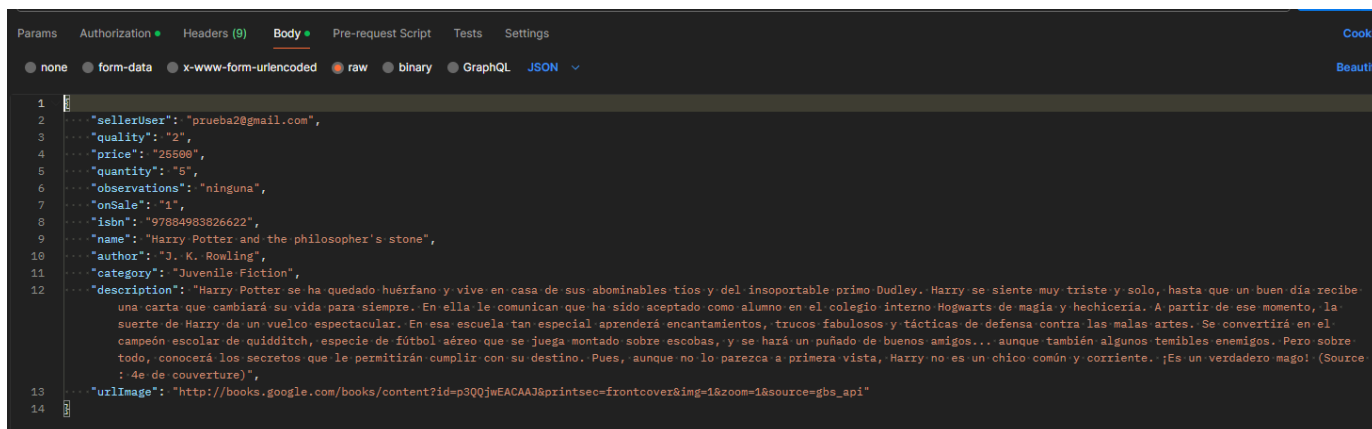
B-001 Registro de libros:

Escenario 01:

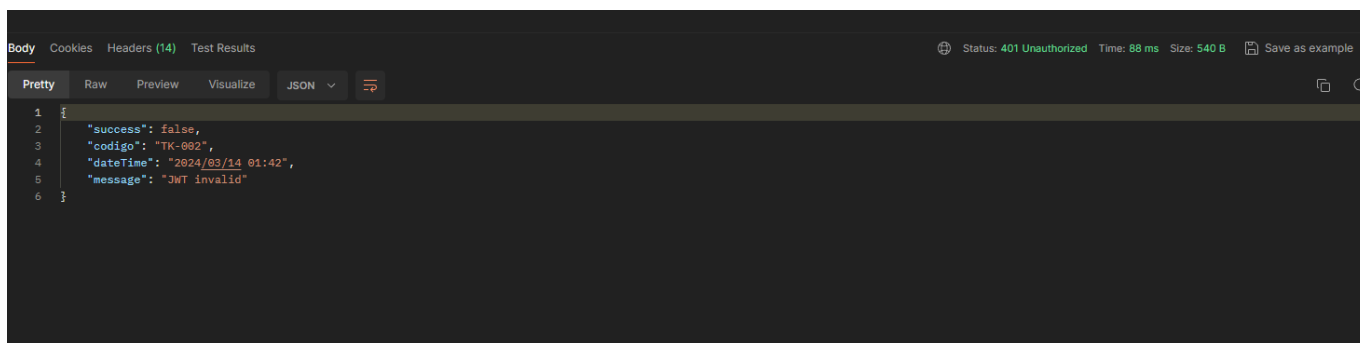
- **URL:** http://204.236.200.68:8082/ApiBookKinder/book/new
- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:**



- **Body:**

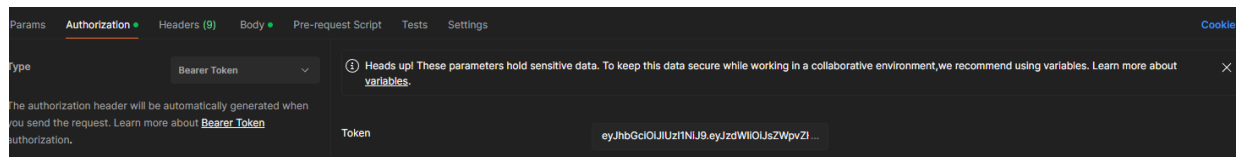


- **Evidencia:**



Escenario 02:

- **URL:** http://204.236.200.68:8082/ApiBookKinder/book/new
- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:**



- **Body:**



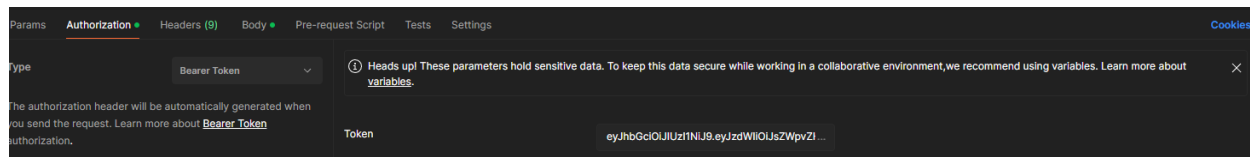
- **Evidencia:**



Nota: Se hizo con un libro ya existente.

Escenario 03:

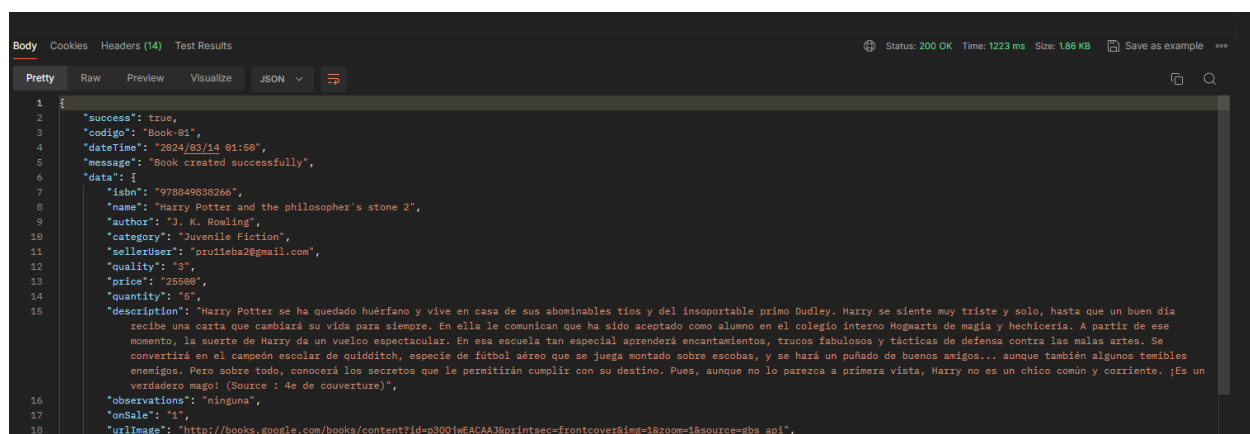
- **URL:** http://204.236.200.68:8082/ApiBookKinder/book/new
- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:**



- **Body:**



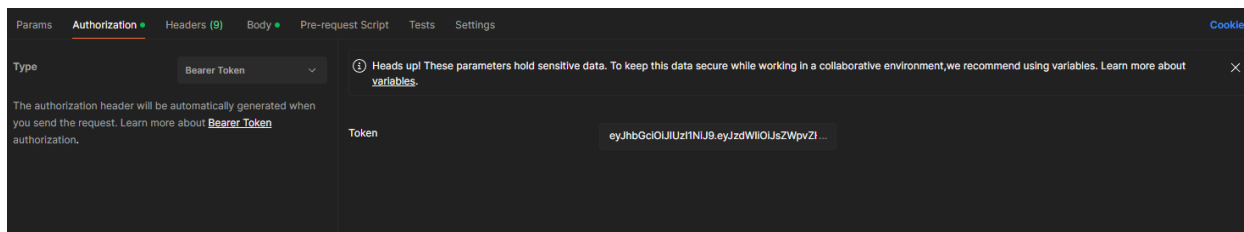
- **Evidencia:**



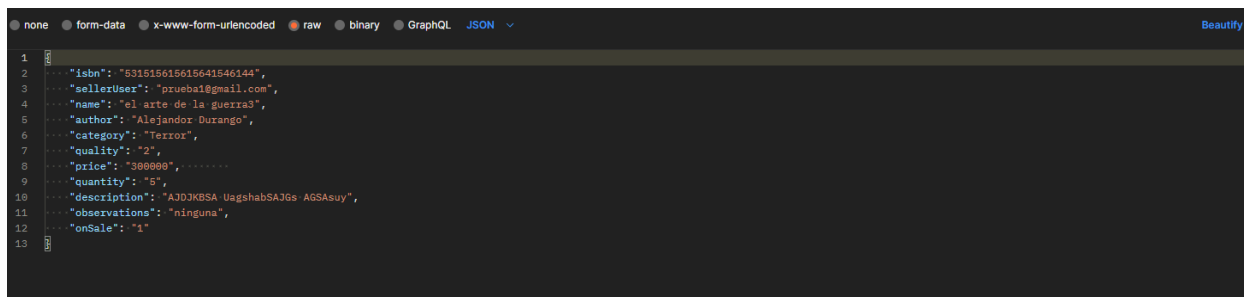
B-002 Actualizar libros:

Escenario 01:

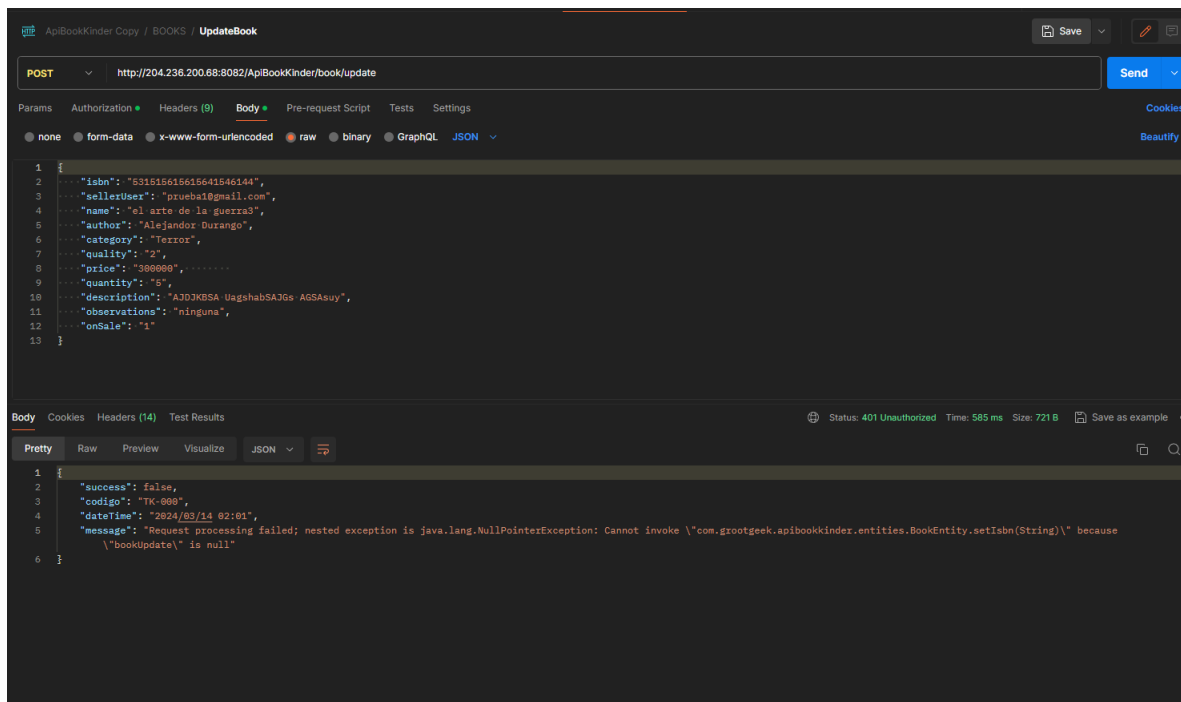
- **URL:** <http://204.236.200.68:8082/ApiBookKinder/book/update>
- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:**



- **Body:**




- **Evidencia:**



Escenario 02:

- **URL:** <http://204.236.200.68:8082/ApiBookKinder/book/update>
- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:**

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Type Bearer T...  Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#).

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.


Token


```
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJsZWpvc2hVyyYw5nb0BnbWVpY20iLCJleHAiOiJE3MTAONTI5NDQslmlhdCI6MTcxMDQyMjk0NCwiQXBwTmFtFtSI6IkFwaUJvb2tLaW5kZXIifQ.FIHZ31X6SI7qX-8yAYXdoVdNVdmd0vi7DYtekw3YSgU
```

- **Body:**

```
{
  "isbn": "531515615615641561",
  "sellerUser": "prueba1@gmail.com",
  "name": "el arte de la guerra3",
  "author": "Alejandor Durango",
  "category": "Terror",
  "quality": "2",
  "price": "300000",
  "quantity": "5",
  "description": "AJDJKBSA UagshabSAJGs AGSAsuy",
  "observations": "ninguna",
  "onSale": "1"
}
```

- **Evidencia:**

Body Cookies Headers (14) Test Results  Status: 200 OK Time: 1016 ms Size: 860 B

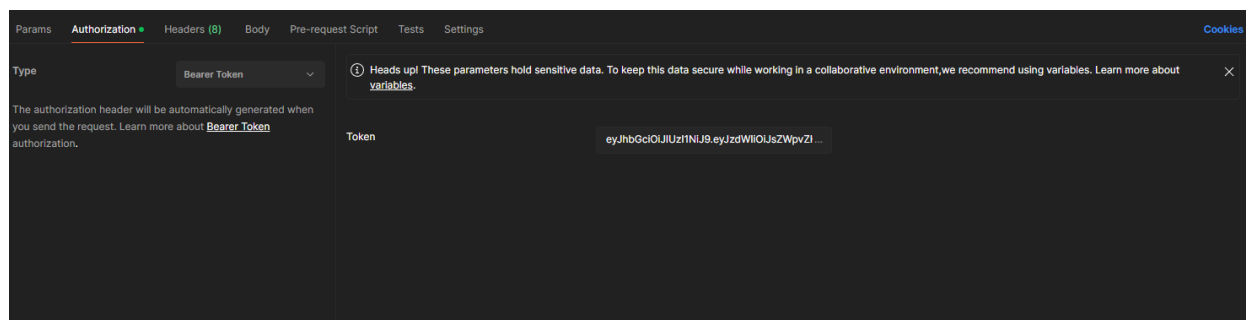
Pretty Raw Preview Visualize JSON 

```
1 {
2   "success": true,
3   "codigo": "Book-03",
4   "dateTime": "2024/03/14 13:31",
5   "message": "Book update successfully",
6   "data": {
7     "isbn": "531515615615641561",
8     "name": "el arte de la guerra3",
9     "author": "Alejandor Durango",
10    "category": "Terror",
11    "sellerUser": "prueba1@gmail.com",
12    "quality": "2",
13    "price": "300000",
14    "quantity": "5",
15    "description": "AJDJKBSA UagshabSAJGs AGSAsuy",
16    "observations": "ninguna",
17    "onSale": "1",
18    "id": "531515615615641561prueba1@gmail.com"
19  }
}
```

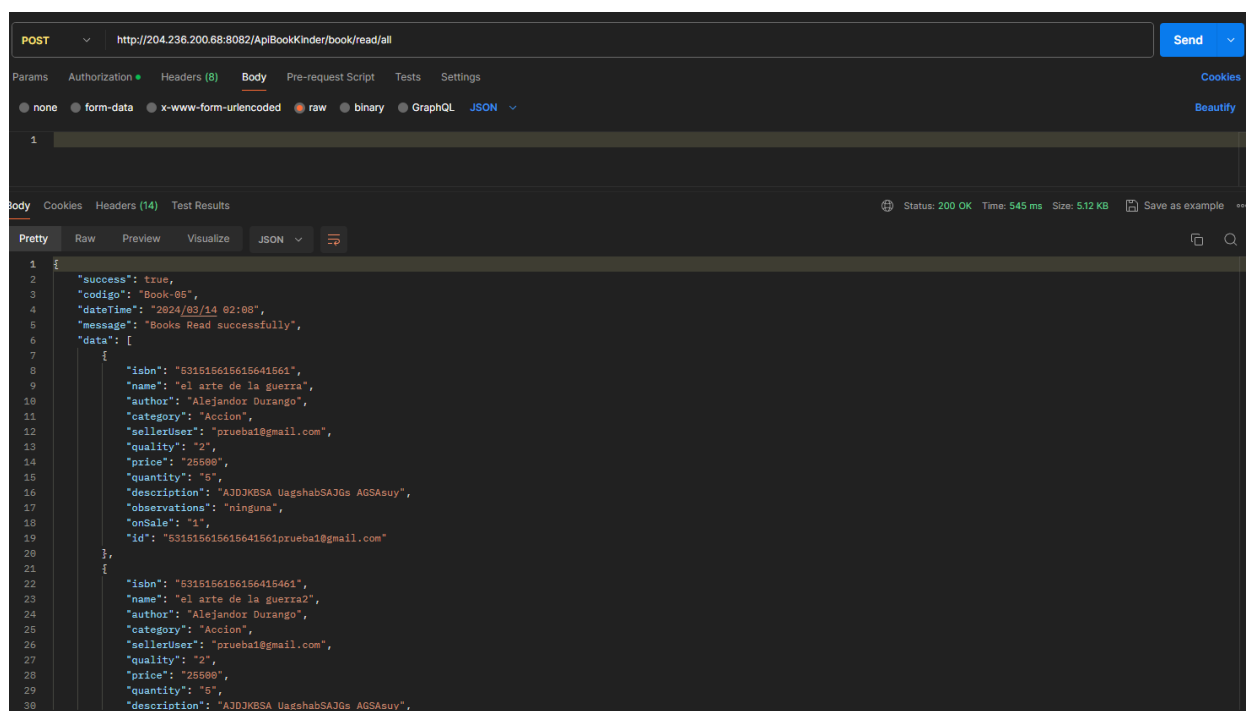
B-003 Buscar todos los Libros:

Escenario 01:

- **URL:** http://204.236.200.68:8082/ApiBookKinder/book/read/all
- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:**



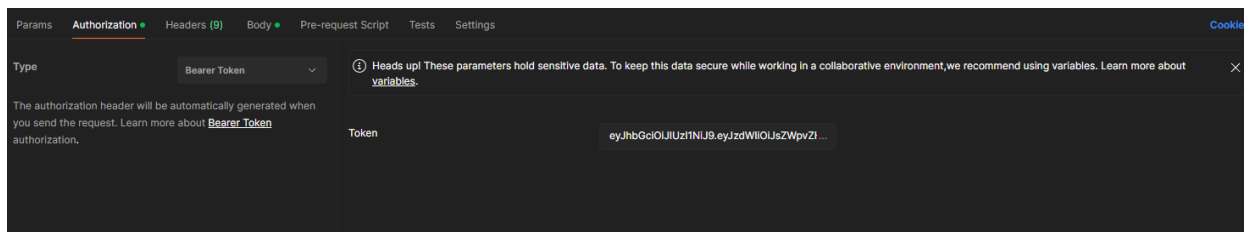
- **Body:** NA.
- **Evidencia:**



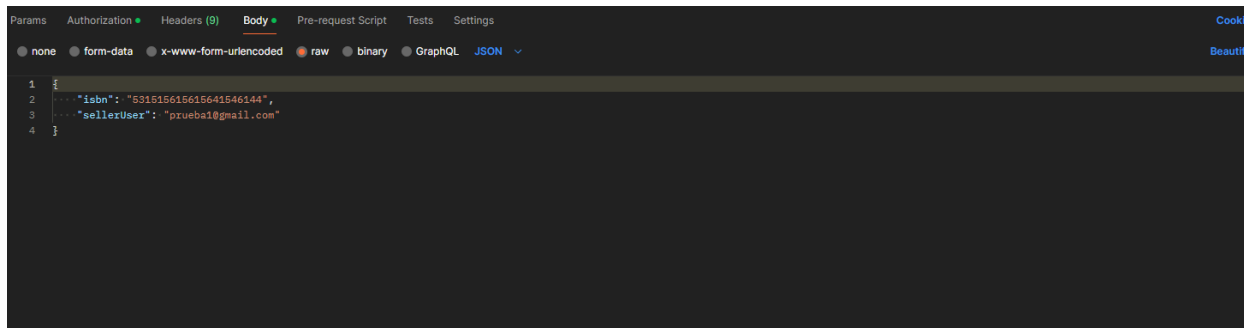
B-004 Eliminar libro:

Escenario 01:

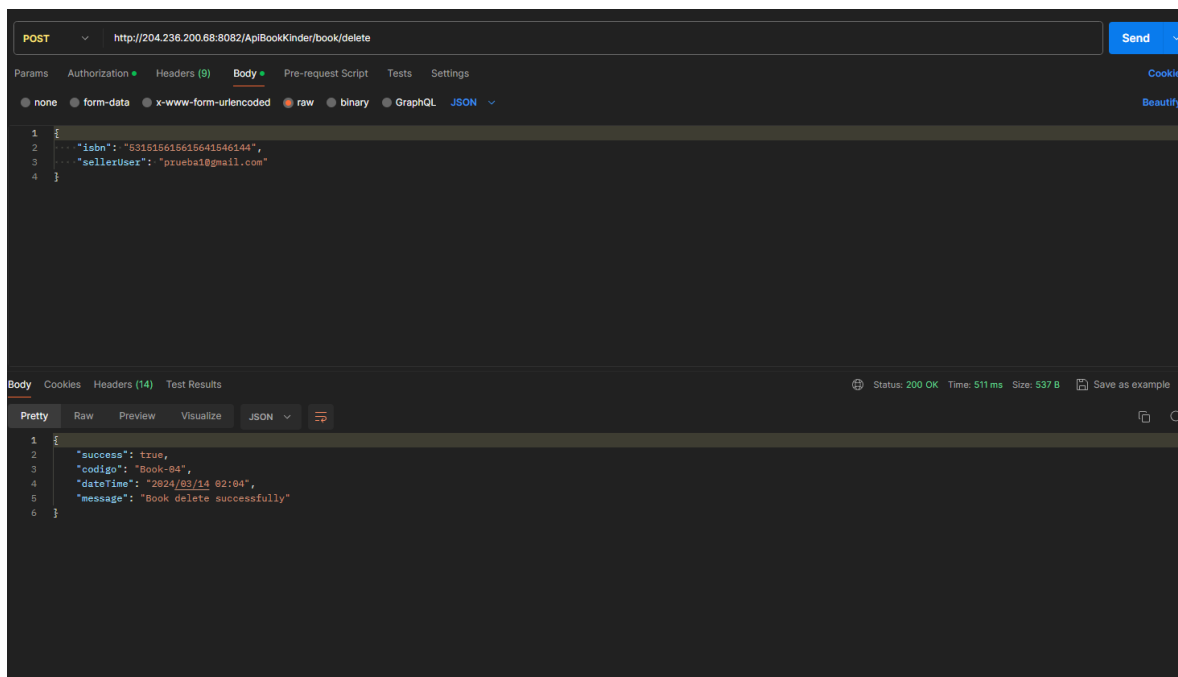
- **URL:** http://204.236.200.68:8082/ApiBookKinder/book/delete
- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:**



- **Body:**



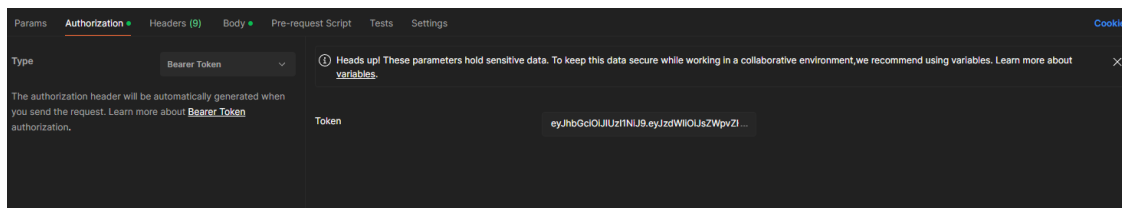
- **Evidencia:**



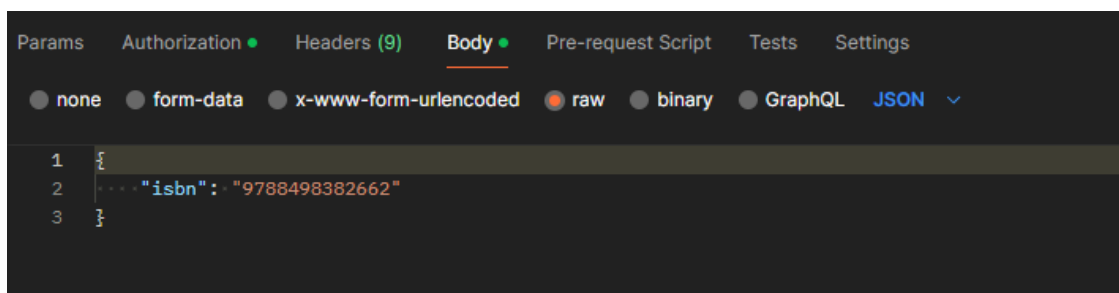
B-005 Buscar Información libro por ISBN:

Escenario 01:

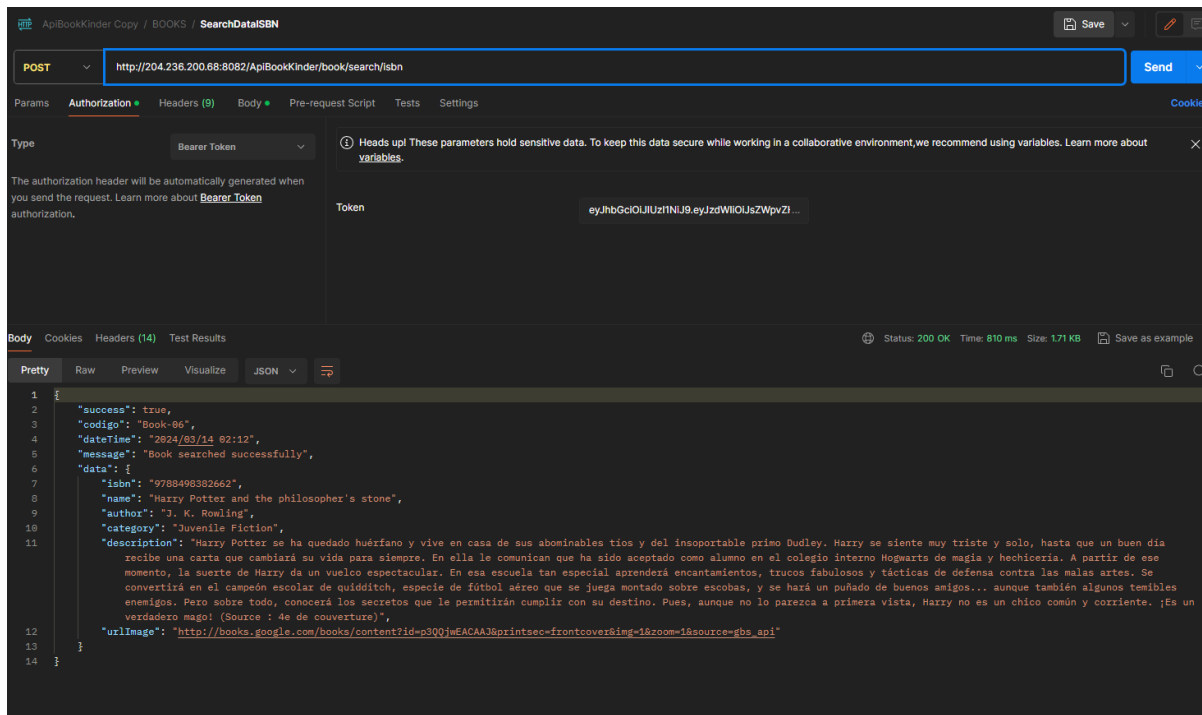
- **URL:** <http://204.236.200.68:8082/ApiBookKinder/book/search/isbn>
- **Path variables:** NA.
- **Query parameters:** NA.
- **Headers:** NA.
- **Auth:**



- **Body:**



- **Evidencia:**



Pruebas Caja Blanca

Para la ejecución de estas pruebas, se documenta la evidencia obtenida y los parámetros utilizados. Para obtener información detallada sobre cada prueba de caja negra, se recomienda revisar el archivo "Contexto Pruebas API-BOOKINDER.docx", específicamente en la sección de "Criterios de aceptación, criterios de entrada y salida".

Libros

Matriz Casos de Prueba

Número	Código	Descripción	Entradas	Pasos	Resultados Esperados	Estado	Prioridad	Responsable
1	B-001	Registro de libros Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual ingresar un libro para la venta en la plataforma.	Escenario1: Token invalid Escenario2: Datos de un libro existente token valido. Escenario3: Datos de un libro Nuevo (sellerUser, quality, price, quantity, observations, onSale, isbn, name, author, category, description, urlImage.), token valido.	1. Obtener token de Autenticación siguiendo los pasos de la prueba de autenticación (Auth-001). 2. Registrar url de la solicitud. 3. Agregar los valores de token obtenido en el paso 1 en la pestaña de Authorization. 4. Agregar el body con todos los datos para insertar los libros en el formato correcto JSON (FormatosJson/newbook). 5. Generar solicitud en Postman. 6. Revisar los datos obtenidos.	Escenario 1: Error por Token erróneo Escenario 2: Respuesta Ok, pero diciendo que el libro ya existe. Escenario 3: Respuesta correcta informando que el libro fue ingresado correctamente	Exitosa	Critica	Equipo QA
2	B-002	Actualizar Libro. Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual actualizar un libro	Escenario1: Datos de un libro inexistente token valido. Escenario2: Datos de un libro existente token valido.	1. Obtener token de Autenticación siguiendo los pasos de la prueba de autenticación (Auth-001). 2. Registrar url de la solicitud. 3. Agregar los valores de token obtenido en el paso 1 en la pestaña de Authorization. 4. Agregar el body con todos los datos para insertar los libros en el formato correcto JSON (FormatosJson/newbook). 5. Generar solicitud en Postman. 6. Revisar los datos obtenidos.	Escenario 1: Respuesta error, por no existir un libro en la BD para actualizar Escenario 2: Respuesta Ok, indicando el libro fue actualizado correctamente	Exitosa	Critica	Equipo QA

3	B-003	Buscar todos los Libros. Yo como vendedor y/o usuario quiero ingresar a la plataforma de Bookinder y de forma manual y buscar todos los libros disponibles.	Escenario 1: token valido, Endpoint Disponible	<ol style="list-style-type: none"> 1. Obtener token de Autenticación siguiendo los pasos de la prueba de autenticación (Auth-001). 2. Registrar url de la solicitud. 3. Agregar los valores de token obtenido en el paso 1 en la pestaña de Authorization. 4. Generar solicitud en Postman. 5. Revisar los datos obtenidos. 	Escenario 1: Respuesta OK, información de los libros Disponibles de la BD	Exitosa	Critica	Equipo QA
4	B-004							
5	B-005							

Casos de Pruebas

B-001 Registro de libros:

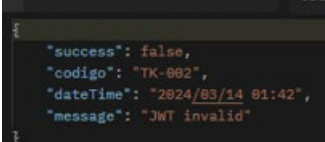
Nombre del proyecto	ApiBookKinder
Número	1
Código	B-001
Responsable	Equipo QA
Fecha	13/03/2024
Descripción	Registro de libros Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual ingresar un libro para la venta en la plataforma.
Actores	Tester QA, Desarrollador
Pre-Condiciones	Token Autenticación
Pasos Normales	<ol style="list-style-type: none"> 1. Obtener token de Autenticación siguiendo los pasos de la prueba de autenticación (Auth-001). 2. Registrar url de la solicitud. 3. Agregar los valores de token obtenido en el paso 1 en la pestaña de Authorization. 4. Agregar el body con todos los datos para insertar los libros en el formato correcto JSON (FormatosJson/newbook). 5. Generar solicitud en Postman. 6. Revisar los datos obtenidos.
Pasos alternativos	Insertarlo directamente en la BD
Excepciones	Libro Existente, Indisponibilidad de la BD, Datos Incorrectos
Resultados Esperados	<ul style="list-style-type: none"> ○ Escenario 1: Error por Token erróneo ○ Escenario 2: Respuesta Ok, pero diciendo que el libro ya existe. ○ Escenario 3: Respuesta correcta informando que el libro fue ingresado correctamente
Postcondiciones	N/A

Matriz Escenarios de Prueba:

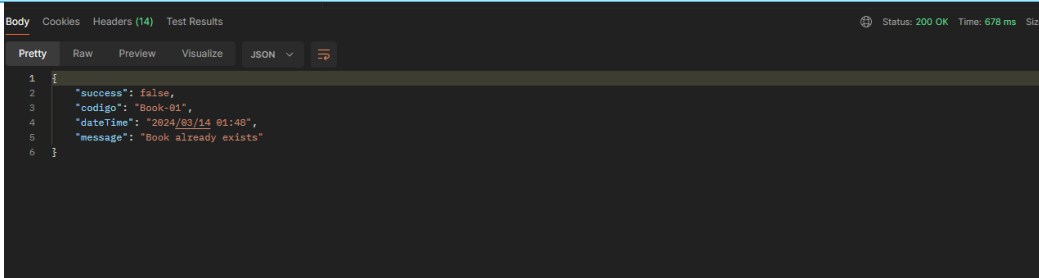
Nombre del Proyecto		ApiBookKinder	
Documentos de Referencia		HU Bookinder.xlsx, Contexto Pruebas API-BOOKINDER.docx (Criterios de aceptación, criterios de entrada y salida)	
Responsable		Equipo QA	
Fecha de Creación		13/03/2024	
Código Prueba		B-001	
N° Caso de Prueba		1	
Identificación Escenarios de Pruebas		Descripción Escenario de Prueba	Importancia
Escenario 01: Registro de Libros Con token Vencido		Se debe de realizar el proceso manual de ingresar un libro a la plataforma de Bookinder, ingresando los datos de manera errónea los datos del Token y visualizando las correspondientes validaciones de la plataforma que NO PERMITA registrar el libro con token erróneo.	Critica
Escenario 02: Registro de Libros con datos de un libro existente		Se debe de realizar el proceso manual de ingresar un libro a la plataforma de Bookinder, ingresando los datos de un libro ya existe y visualizando las correspondientes validaciones de la plataforma que NO PERMITA registrar el libro con datos de un libro existente.	Critica
Escenario 03: Registro de Libros con datos de un libro Nuevo		Se debe de realizar el proceso manual de ingresar un libro a la plataforma de Bookinder, ingresando los datos correspondientes que solicita el servicio bajo sus correspondientes parámetros y validaciones de manera correcta y como resultado visualizar el libro correctamente registrado y con los datos correctos.	Critica

Escenarios de Pruebas

Escenario 01:

Nombre del proyecto		ApiBookKinder
Documento de Referencia	HU Bookinder.xlsx, Contexto Pruebas API-BOOKINDER (Criterios de aceptación, criterios de entrada y salida)	
Código	B-001	
Escenario	1	
Responsable	Equipo QA	
Fecha de Creación	13/03/2024	
Descripción del Escenario	Se debe de realizar el proceso manual de ingresar un libro a la plataforma de Bookinder, ingresando los datos de manera errónea los datos del Token y visualizando las correspondientes validaciones de la plataforma que NO PERMITA registrar el libro con token erróneo.	
Requisitos o Historias de Usuario	HLR-01 - Registro de usuario	
Criterios de Aceptación	Código: B-001. (Contexto pruebas Bookinder.docx)	
Supuestos y Restricciones	No contar con los datos ni entorno requerido para las pruebas	
Riesgos	A. Fallos en el servidor, B. Ambiente incorrecto de prueba	
Recursos Asignados	A. Servidor de pruebas B. Tiempo y personas para la realización. C. Datos para las pruebas.	
Métricas	Tablas métricas B-001/Escenario 01	
Resultados Esperados	Error por Token erróneo	
Resultados Obtenidos	 <pre> { "success": false, "codigo": "TK-002", "dateTime": "2024/03/14 01:42", "message": "JWT invalid" } </pre>	

Escenario 02:

Nombre del proyecto	ApiBookKinder
Documento de Referencia	HU Bookinder.xlsx, Contexto Pruebas API-BOOKINDER (Criterios de aceptación, criterios de entrada y salida)
Código	B-001
Escenario	2
Responsable	Equipo QA
Fecha de Creación	13/03/2024
Descripción del Escenario	Se debe de realizar el proceso manual de ingresar un libro a la plataforma de Bookinder, ingresando los datos de un libro ya existe y visualizando las correspondientes validaciones de la plataforma que NO PERMITA registrar el libro con datos de un libro existente.
Requisitos o Historias de Usuario	HLR-01 - Registro de usuario
Criterios de Aceptación	Código: B-001. (Contexto pruebas Bookinder.docx)
Supuestos y Restricciones	No contar con los datos ni entorno requerido para las pruebas
Riesgos	A. Fallos en el servidor, B. Ambiente incorrecto de prueba
Recursos Asignados	A. Servidor de pruebas B. Tiempo y personas para la realización. C. Datos para las pruebas.
Métricas	Tablas métricas B-001/Escenario 02
Resultados Esperados	Respuesta Ok, pero diciendo que el libro ya existe.
Resultados Obtenidos	 <pre> 1 { 2 "success": false, 3 "codigo": "Book-01", 4 "dateTime": "2024/03/14 01:40", 5 "message": "Book already exists" 6 } </pre>

Escenario 03:

Nombre del proyecto		ApiBookKinder
Documento de Referencia	HU Bookinder.xlsx, Contexto Pruebas API-BOOKINDER (Criterios de aceptación, criterios de entrada y salida)	
Código	B-001	
Escenario	3	
Responsable	Equipo QA	
Fecha de Creación	13/03/2024	
Descripción del Escenario	Ve debe de realizar el proceso manual de ingresar un libro a la plataforma de Bookinder, ingresando los datos correspondientes que solicita el servicio bajo sus correspondientes parámetros y validaciones de manera correcta y como resultado visualizar el libro correctamente registrado y con los datos correctos.	
Requisitos o Historias de Usuario	HLR-01 - Registro de usuario	
Criterios de Aceptación	Código: B-001. (Contexto pruebas Bookinder.docx)	
Supuestos y Restricciones	No contar con los datos ni entorno requerido para las pruebas	
Riesgos	A. Fallos en el servidor, B. Ambiente incorrecto de prueba	
Recursos Asignados	A. Servidor de pruebas B. Tiempo y personas para la realización. C. Datos para las pruebas.	
Métricas	Tablas métricas B-001/Escenario 03	
Resultados Esperados	Respuesta correcta informando que el libro fue ingresado correctamente	
Resultados Obtenidos	<div><div>BodyCookiesHeaders (14)Test Results</div><div>Status: 200 OKTime: 1223 msSize: 1.86 KBSave as example</div><div>PrettyRawPreviewVisualizeJSON</div><div><pre>1 { 2 "success": true, 3 "codigo": "Book-01", 4 "dateTime": "2024/03/14 01:58", 5 "message": "Book created successfully", 6 "data": { 7 "isbn": "97804963266", 8 "name": "Harry Potter and the philosopher's stone 2", 9 "author": "J. K. Rowling", 10 "category": "Juvenile Fiction", 11 "sellerUser": "pruliba2@gmail.com", 12 "quality": "3", 13 "price": "25000", 14 "quantity": "5", 15 "description": "Harry Potter se ha quedado huérfano y vive en casa de sus abominables tíos y del insoportable primo Dudley. Harry se siente muy triste y solo, hasta que un buen día recibe una carta que cambiará su vida para siempre. En ella le comunican que ha sido aceptado como alumno en el colegio interno Hogwarts de magia y hechicería. A partir de ese momento, la suerte de Harry da un vuelco espectacular. En esa escuela tan especial aprenderá encantamientos, trucos fabulosos y técnicas de defensa contra las malas artes. Se convertirá en el campeón escolar de quidditch, especie de fútbol aéreo que se juega montado sobre escobas, y se hará un puñado de buenos amigos... aunque también algunos terribles enemigos. Pero sobre todo, conocerá los secretos que le permitirán cumplir con su destino. Pues, aunque no lo parezca a primera vista, Harry no es un chico común y corriente. ¡Es un verdadero mago! (Source : 4e de couverture)", 16 "observations": "ninguna", 17 "mobile": "1", 18 "urlImage": "http://books.google.com/books/content?id=p3QJjwACAAJ&printsec=frontcover&img=1&zoom=1&source=gbs_api", 19 } 20 }</pre></div></div>	

B-002 Actualizar libros:

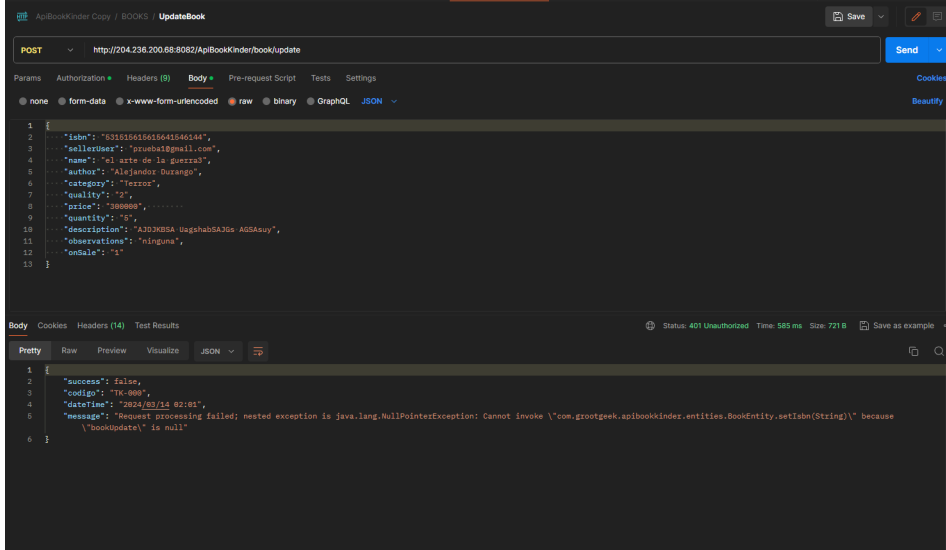
Nombre del proyecto	ApiBookKinder
Número	2
Código	B-002
Responsable	Equipo QA
Fecha	13/03/2024
Descripción	Actualizar Libro. Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual actualizar un libro
Actores	Tester QA, Desarrollador
Pre-Condiciones	Token Autenticación
Pasos Normales	<ol style="list-style-type: none"> 1. Obtener token de Autenticación siguiendo los pasos de la prueba de autenticación (Auth-001). 2. Registrar url de la solicitud. 3. Agregar los valores de token obtenido en el paso 1 en la pestaña de Authorization. 4. Agregar el body con todos los datos para insertar los libros en el formato correcto JSON (FormatosJson/newbook). 5. Generar solicitud en Postman. 6. Revisar los datos obtenidos.
Pasos alternativos	Actualizarlo directamente en la BD
Excepciones	Libro Inexistente, Indisponibilidad de la BD, Datos Incorrectos
Resultados Esperados	<ul style="list-style-type: none"> ○ Escenario 1: Respuesta error, por no existir un libro en la BD para actualizar ○ Escenario 2: Respuesta Ok, indicando el libro fue actualizado correctamente
Postcondiciones	N/A

Matriz Escenarios de Prueba:

Nombre del Proyecto			ApiBookKinder
Documentos de Referencia	HU Bookinder.xlsx, Contexto Pruebas API-BOOKINDER.docx (Criterios de aceptación, criterios de entrada y salida)		
Responsable	Equipo QA		
Fecha de Creación	13/03/2024		
Código Prueba	B-002		
N° Caso de Prueba	2		
Identificación Escenarios de Pruebas	Descripción Escenario de Prueba		Importancia
Escenario 01: Actualizar Libro con datos de un libro inexistente	Se debe de realizar el proceso manual de actualizar un libro a la plataforma de Bookinder, ingresando los datos de un libro inexistente y visualizando las correspondientes validaciones de la plataforma que NO PERMITA actualizar el libro ya que este no existe, dando su respectivo dato de error		Critica
Escenario 02: Registro de Libros con datos de un libro existente	Se debe de realizar el proceso manual de actualizar un libro a la plataforma de Bookinder, ingresando los datos de un libro que existe y visualizando las correspondientes validaciones de la plataforma que PERMITA actualizar el libro ya que este existe, dando su respectivo dato de confirmación.		Critica

Escenarios de Pruebas

Escenario 01:

Nombre del proyecto		ApiBookKinder	
Documento de Referencia	HU Bookinder.xlsx, Contexto Pruebas API-BOOKINDER (Criterios de aceptación, criterios de entrada y salida)		
Código	B-002		
Escenario	2		
Responsable	Equipo QA		
Fecha de Creación	13/03/2024		
Descripción del Escenario	Se debe de realizar el proceso manual de actualizar un libro a la plataforma de Bookinder, ingresando los datos de un libro inexistente y visualizando las correspondientes validaciones de la plataforma que NO PERMITA actualizar el libro ya que este no existe, dando su respectivo dato de error		
Requisitos o Historias de Usuario	HLR-01 - Registro de usuario		
Criterios de Aceptación	Código: B-002. (Contexto pruebas Bookinder.docx)		
Supuestos y Restricciones	No contar con los datos ni entorno requerido para las pruebas		
Riesgos	A. Fallos en el servidor, B. Ambiente incorrecto de prueba		
Recursos Asignados	A. Servidor de pruebas B. Tiempo y personas para la realización. C. Datos para las pruebas.		
Métricas	Tablas métricas B-002/Escenario 01		
Resultados Esperados	Respuesta error, por no existir un libro en la BD para actualizar		
Resultados Obtenidos			

Escenario 02:

Nombre del proyecto	ApiBookKinder
Documento de Referencia	HU Bookinder.xlsx, Contexto Pruebas API-BOOKINDER (Criterios de aceptación, criterios de entrada y salida)
Código	B-002
Escenario	2
Responsable	Equipo QA
Fecha de Creación	13/03/2024
Descripción del Escenario	Se debe de realizar el proceso manual de ingresar un libro a la plataforma de Bookinder, ingresando los datos de un libro ya existe y visualizando las correspondientes validaciones de la plataforma que NO PERMITA registrar el libro con datos de un libro existente.
Requisitos o Historias de Usuario	HLR-01 - Registro de usuario
Criterios de Aceptación	Código: B-002. (Contexto pruebas Bookinder.docx)
Supuestos y Restricciones	No contar con los datos ni entorno requerido para las pruebas
Riesgos	A. Fallos en el servidor, B. Ambiente incorrecto de prueba
Recursos Asignados	A. Servidor de pruebas B. Tiempo y personas para la realización. C. Datos para las pruebas.
Métricas	Tablas métricas B-002/Escenario 02
Resultados Esperados	Respuesta Ok, indicando el libro fue actualizado correctamente
Resultados Obtenidos	<div> <div>Body Cookies Headers (14) Test Results</div> <div>Status: 200 OK Time: 1016 ms Size: 860 B</div> <div> <div>Pretty Raw Preview Visualize JSON</div> <pre> 1 { 2 "success": true, 3 "codigo": "Book-03", 4 "dateTime": "2024/03/14 13:31", 5 "message": "Book update successfully", 6 "data": { 7 "isbn": "531515615615641561", 8 "name": "el arte de la guerra3", 9 "author": "Alejander Durango", 10 "category": "Terror", 11 "sellerUser": "prueba1@gmail.com", 12 "quality": "2", 13 "price": "300000", 14 "quantity": "5", 15 "description": "AJDJKBSA UagshabSAJGs AGSAsuy", 16 "observations": "ninguna", 17 "onSale": "1", 18 "id": "531515615615641561prueba1@gmail.com" 19 } </pre> </div> </div>

B-003 Buscar todos los Libros:

Nombre del proyecto	ApiBookKinder
Número	3
Código	B-003
Responsable	Equipo QA
Fecha	13/03/2024
Descripción	<p>Buscar todos los Libros.</p> <p>Yo como vendedor y/o usuario quiero ingresar a la plataforma de Bookinder y de forma manual y buscar todos los libros disponibles.</p>
Actores	Tester QA, Desarrollador
Pre-Condiciones	Token Autenticación
Pasos Normales	<ol style="list-style-type: none"> 1. Obtener token de Autenticación siguiendo los pasos de la prueba de autenticación (Auth-001). 2. Registrar URL de la solicitud. 3. Agregar los valores de token obtenido en el paso 1 en la pestaña de Autorización. 4. Generar solicitud en Postman. 5. Revisar los datos obtenidos.
Pasos alternativos	Buscarlos directamente en la BD
Excepciones	Libros Inexistentes, Indisponibilidad de la BD, consulta Incorrectos
Resultados Esperados	<ul style="list-style-type: none"> ○ Escenario 1: Respuesta OK, información de los libros Disponibles de la BD
Postcondiciones	N/A

Matriz Escenarios de Prueba:

Nombre del Proyecto				
ApiBookKinder				
Documentos de Referencia	HU Bookinder.xlsx, Detallado pruebas Bookinder.docx (Criterios de aceptación, criterios de entrada y salida)			
Responsable	Santiago Franco			
Fecha de Creación	13/03/2024			
Código Prueba	B-001			
Identificación Escenarios de Pruebas	Documento de Referencia	Descripción Escenario de Prueba	Importancia	No de Caso de Prueba
Registro de Libros Con token Vencido	Casos y Escenarios de Pruebas.docx ()	Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual ingresar un libro para la venta en la plataforma.	Crítica	1

Escenarios de Pruebas

Escenario 01:

Nombre del proyecto	
ApiBookKinder	
Documento de Referencia	HU Bookinder.xlsx, Detallado pruebas Bookinder.docx (Criterios de aceptación, criterios de entrada y salida)
Código	B-001
Escenario	1
Responsable	Santiago Franco
Fecha de Creación	13/03/2024
Descripción del Escenario	Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual ingresar un libro para la venta en la plataforma.
Requisitos o Historias de Usuario	HLR-01 - Registro de usuario
Criterios de Aceptación	Código: B-001. (Detallado pruebas Bookinder.docx)
Supuestos y Restricciones	No contar con los datos ni entorno requerido para las pruebas
Riesgos	A. Fallos en el servidor, B. Ambiente incorrecto de prueba
Recursos Asignados	A. Servidor de pruebas B. Tiempo y personas para la realización. C. Datos para las pruebas.
Métricas	Tablas métricas

Resultados Esperados		<pre> { "success": false, "codigo": "TK-002", "dateTime": "2024/03/14 01:42", "message": "JWT invalid" } </pre>	
Resultados Obtenidos		<pre> { "success": false, "codigo": "TK-002", "dateTime": "2024/03/14 01:42", "message": "JWT invalid" } </pre>	

B-004 Eliminar libro:

Nombre del proyecto		ApiBookKinder
Número		1
Código		B-001
Responsable		Santiago Franco
Fecha		13/03/2024
Descripción		Registro de libros
Actores		Tester QA, Desarrollador
Pre-Condiciones		Token Autenticación
Pasos Normales	<ol style="list-style-type: none"> 1. Obtener token de Autenticación siguiendo los pasos de la prueba de autenticación (Auth-001). 2. Registrar url de la solicitud. 3. Agregar los valores de token obtenido en el paso 1 en la pestaña de Authorization. 4. Agregar el body con todos los datos para insertar los libros en el formato correcto JSON (FormatosJson/newbook). 5. Generar solicitud en Postman. 6. Revisar los datos obtenidos. 	
Pasos alternativos	Insertarlo directamente en la BD	
Excepciones	Libro Existente, Indisponibilidad de la BD, Datos Incorrectos	
Resultados Esperados	<ol style="list-style-type: none"> 1. Validar resultados esperado de la prueba (Auth-001). 2. Resultado prueba Actual: <pre> { "success": false, "codigo": "Book-01", "dateTime": "2024/03/14 02:44", "message": "Book already exists"} </pre> 	
Postcondiciones	N/A	

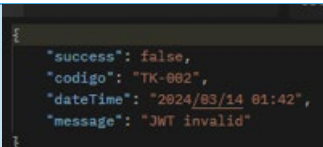
Matriz Escenarios de Prueba:

Nombre del Proyecto	ApiBookKinder
---------------------	---------------

Documentos de Referencia	HU Bookinder.xlsx, Detallado pruebas Bookinder.docx (Criterios de aceptación, criterios de entrada y salida)			
Responsable	Santiago Franco			
Fecha de Creación	13/03/2024			
Código Prueba	B-001			
Identificación Escenarios de Pruebas	Documento de Referencia	Descripción Escenario de Prueba	Importancia	No de Caso de Prueba
Registro de Libros Con token Vencido	Casos y Escenarios de Pruebas.docx ()	Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual ingresar un libro para la venta en la plataforma.	Crítica	1

Escenarios de Pruebas

Escenario 01:

Nombre del proyecto	ApiBookKinder			
Documento de Referencia	HU Bookinder.xlsx, Detallado pruebas Bookinder.docx (Criterios de aceptación, criterios de entrada y salida)			
Código	B-001			
Escenario	1			
Responsable	Santiago Franco			
Fecha de Creación	13/03/2024			
Descripción del Escenario	Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual ingresar un libro para la venta en la plataforma.			
Requisitos o Historias de Usuario	HLR-01 - Registro de usuario			
Criterios de Aceptación	Código: B-001. (Detallado pruebas Bookinder.docx)			
Supuestos y Restricciones	No contar con los datos ni entorno requerido para las pruebas			
Riesgos	A. Fallos en el servidor, B. Ambiente incorrecto de prueba			
Recursos Asignados	A. Servidor de pruebas B. Tiempo y personas para la realización. C. Datos para las pruebas.			
Métricas	Tablas métricas			
Resultados Esperados	 <pre> { "success": false, "codigo": "TK-002", "dateTime": "2024/03/14 01:42", "message": "JWT invalid" } </pre>			

Resultados Obtenidos

```
"success": false,  
"codigo": "TK-002",  
"dateTime": "2024/03/14 01:42",  
"message": "JWT invalid"
```

B-005 Buscar Información libro por ISBN:

Nombre del proyecto		ApiBookKinder		
Número		1		
Código		B-001		
Responsable		Santiago Franco		
Fecha		13/03/2024		
Descripción		Registro de libros		
Actores		Tester QA, Desarrollador		
Pre-Condiciones		Token Autenticación		
Pasos Normales		1. Obtener token de Autenticación siguiendo los pasos de la prueba de autenticación (Auth-001). 2. Registrar url de la solicitud. 3. Agregar los valores de token obtenido en el paso 1 en la pestaña de Authorization. 4. Agregar el body con todos los datos para insertar los libros en el formato correcto JSON (FormatosJson/newbook). 5. Generar solicitud en Postman. 6. Revisar los datos obtenidos.		
Pasos alternativos		Insertarlo directamente en la BD		
Excepciones		Libro Existente, Indisponibilidad de la BD, Datos Incorrectos		
Resultados Esperados		1. Validar resultados esperado de la prueba (Auth-001). 2. Resultado prueba Actual: { "success": false, "codigo": "Book-01", "dateTime": "2024/03/14 02:44", "message": "Book already exists"}		
Postcondiciones		N/A		

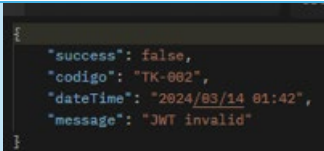
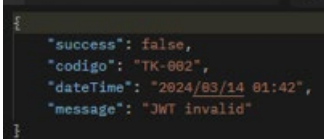
Matriz Escenarios de Prueba:

Nombre del Proyecto		ApiBookKinder		
Documentos de Referencia		HU Bookinder.xlsx, Detallado pruebas Bookinder.docx (Criterios de aceptación, criterios de entrada y salida)		
Responsable		Santiago Franco		
Fecha de Creación		13/03/2024		
Código Prueba		B-001		
Identificación Escenarios de Pruebas	Documento de Referencia	Descripción Escenario de Prueba	Importancia	No de Caso de Prueba

Registro de Libros Con token Vencido	Casos y Escenarios de Pruebas.docx ()	Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual ingresar un libro para la venta en la plataforma.	Critica	1
---	--	--	---------	---

Escenarios de Pruebas

Escenario 01:

Nombre del proyecto		ApiBookKinder	
Documento de Referencia	HU Bookinder.xlsx, Detallado pruebas Bookinder.docx (Criterios de aceptación, criterios de entrada y salida)		
Código	B-001		
Escenario	1		
Responsable	Santiago Franco		
Fecha de Creación	13/03/2024		
Descripción del Escenario	Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual ingresar un libro para la venta en la plataforma.		
Requisitos o Historias de Usuario	HLR-01 - Registro de usuario		
Criterios de Aceptación	Código: B-001. (Detallado pruebas Bookinder.docx)		
Supuestos y Restricciones	No contar con los datos ni entorno requerido para las pruebas		
Riegos	A. Fallos en el servidor, B. Ambiente incorrecto de prueba		
Recursos Asignados	A. Servidor de pruebas B. Tiempo y personas para la realización. C. Datos para las pruebas.		
Métricas	Tablas métricas		
Resultados Esperados			
Resultados Obtenidos			

Análisis de Algoritmo

B-001 Registro de libros:

Análisis Inicial

```
final BookEntity validateBook = booksService.findAllByisbnandseller(newBookReq.getIsbn(), newBookReq.getSellerUser());
if (validateBook != null) {
    responseNew.setSuccess(false);
    responseNew.setCodigo("Book-01");
    responseNew.setMessage("Book already exists");
    logbook = responseNew.getCodigo() + " ISBN: " + newBookReq.getIsbn() + " Seller user" + newBookReq.getSellerUser() + " " + responseNew.getMessage();
    utilLogs.logApi(1, logbook, BookController.class.getSimpleName(), request.getRemoteAddr());
} else {
    newBookReq.setID(newBookReq.getIsbn() + newBookReq.getSellerUser());
    final BookEntity createBook = booksService.saveNewBook(newBookReq);
    if (createBook != null) {
        responseNew.setSuccess(true);
        responseNew.setCodigo("Book-01");
        responseNew.setMessage("Book created successfully");
        responseNew.setData(newBookReq);
        logbook = responseNew.getMessage() + ": " + responseNew.getData() + " ";
    } else {
        responseNew.setSuccess(false);
        responseNew.setCodigo("ErrorBook-01");
        responseNew.setMessage("Error creating Book");
        logbook = responseNew.getCodigo() + " ISBN: " + newBookReq.getIsbn() + " sellerBook: " + newBookReq.getSellerUser();
        utilLogs.logApiError(logbook);
    }
    utilLogs.logApi(1, logbook, BookController.class.getSimpleName(), request.getRemoteAddr());
}
return responseNew;
```

Recomendaciones

- R-01: se debe puede solucionar esto optimizando a través de un método constructor para respuestas tipi success o error

Correcciones:

```
if (validateBook != null) {
    responseNewBook.responseError( success: false, code: "Book-01", message: "Book already exists");
    logNewBook = responseNewBook.getCode() + isbnBook + newBookReq.getIsbn() + sellerBook + newBookReq.getSellerUser() + " " + responseNewBook.getMessage();
    utilLogs.logApi( responseAPI: 1, logNewBook, BookController.class.getSimpleName(), request.getRemoteAddr());
} else {
    newBookReq.setId(newBookReq.getIsbn() + newBookReq.getSellerUser());
    final BookEntity createBook = booksService.saveNewBook(newBookReq);
    if (createBook != null) {
        responseNewBook.responseSuccess( success: true, code: "Book-01", message: "Book created successfully", createBook);
        logNewBook = responseNewBook.getMessage() + ": " + responseNewBook.getData() + " ";
    } else {
        responseNewBook.responseError( success: false, code: "ErrorBook-01", message: "Error creating Book");
        logNewBook = responseNewBook.getCode() + isbnBook + newBookReq.getIsbn() + sellerBook + newBookReq.getSellerUser();
        utilLogs.logApiError(logNewBook);
    }
    utilLogs.logApi( responseAPI: 1, logNewBook, BookController.class.getSimpleName(), request.getRemoteAddr());
}
return responseNewBook;
```

B-002 Actualizar libros:

Análisis Inicial:

```

if (updateBook != null) {
    responseUpdate.setSuccess(true);
    responseUpdate.setCodigo("Book-03");
    responseUpdate.setMessage("Book update successfully");
    responseUpdate.setData(updateBook);
    logUpdate = responseUpdate.getMessage() + ": " + responseUpdate.getData() + " ";
} else {
    responseUpdate.setSuccess(false);
    responseUpdate.setCodigo("ErrorBook-03");
    responseUpdate.setMessage("Error updating Book");
    logUpdate = responseUpdate.getCodigo() + " ISBN: " + updateBookReq.getIsbn() + " seller
    utilLogs.logApiError(logUpdate);
}
utilLoes.logApi(1, logUpdate, BookController.class.getSimpleName(), request.getRemoteAddr())

```

```

@Override
public BookEntity updateBookApp(BookEntity updateBookApp) {
    BookEntity bookUpdate = findallByisbnandseller(updateBookApp.getIsbn(), updateBookApp.getSellerUser());
    if (Objects.nonNull(updateBookApp.getID()) && !"".equalsIgnoreCase(updateBookApp.getID())) {
        bookUpdate.setID(updateBookApp.getID());
    }
    if (Objects.nonNull(updateBookApp.getIsbn()) && !"".equalsIgnoreCase(updateBookApp.getIsbn())) {
        bookUpdate.setIsbn(updateBookApp.getIsbn());
    }
    if (Objects.nonNull(updateBookApp.getName()) && !"".equalsIgnoreCase(updateBookApp.getName())) {
        bookUpdate.setName(updateBookApp.getName());
    }
    if (Objects.nonNull(updateBookApp.getAuthor()) && !"".equalsIgnoreCase(updateBookApp.getAuthor())) {
        bookUpdate.setAuthor(updateBookApp.getAuthor());
    }
    if (Objects.nonNull(updateBookApp.getCategory()) && !"".equalsIgnoreCase(updateBookApp.getCategory())) {
        bookUpdate.setCategory(updateBookApp.getCategory());
    }
    if (Objects.nonNull(updateBookApp.getSellerUser()) && !"".equalsIgnoreCase(updateBookApp.getSellerUser())) {
        bookUpdate.setSellerUser(updateBookApp.getSellerUser());
    }
    if (Objects.nonNull(updateBookApp.getQuality()) && !"".equalsIgnoreCase(updateBookApp.getQuality())) {
        bookUpdate.setQuality(updateBookApp.getQuality());
    }
    if (Objects.nonNull(updateBookApp.getPrice()) && !"".equalsIgnoreCase(updateBookApp.getPrice())) {
        bookUpdate.setPrice(updateBookApp.getPrice());
    }
}

```

Recomendaciones

- R-01: se debe puede solucionar esto optimizando a través de un método constructor para respuestas tipi sucss o error
- R-02: Alto Grado de Complejidad, se debe refactorizar para eliminar, la complejidad de condicionales, se recomienda un método que realice una única validación

Correcciones:

```
if (updateBook != null) {
    responseUpdateBook.responseSuccess( success: true, code: "Book-03", message: "Book update successfully", updateBook);
    logUpdateBook = responseUpdateBook.getMessage() + ": " + responseUpdateBook.getData() + " ";
} else {
    responseUpdateBook.responseError( success: false, code: "ErrorBook-03", message: "Error updating Book");
    logUpdateBook = responseUpdateBook.getCode() + isbnBook + updateBookReq.getIsbn() + sellerBook + updateBookReq.getSellerUser();
    utilLogs.logApiError(logUpdateBook);
}
```

```
public BookEntity updateBookApp(BookEntity updateBookApp) {
    BookEntity bookUpdate = findallByisbnandseller(updateBookApp.getIsbn(), updateBookApp.getSellerUser());
    updateIfNotNullOrEmpty(bookUpdate::setID, updateBookApp.getID());
    updateIfNotNullOrEmpty(bookUpdate::setIsbn, updateBookApp.getIsbn());
    updateIfNotNullOrEmpty(bookUpdate::setName, updateBookApp.getName());
    updateIfNotNullOrEmpty(bookUpdate::setAuthor, updateBookApp.getAuthor());
    updateIfNotNullOrEmpty(bookUpdate::setCategory, updateBookApp.getCategory());
    updateIfNotNullOrEmpty(bookUpdate::setSellerUser, updateBookApp.getSellerUser());
    updateIfNotNullOrEmpty(bookUpdate::setQuality, updateBookApp.getQuality());
    updateIfNotNullOrEmpty(bookUpdate::setPrice, updateBookApp.getPrice());
    updateIfNotNullOrEmpty(bookUpdate::setQuantity, updateBookApp.getQuantity());
    updateIfNotNullOrEmpty(bookUpdate::setDescription, updateBookApp.getDescription());
    updateIfNotNullOrEmpty(bookUpdate::setObservations, updateBookApp.getObservations());
    updateIfNotNullOrEmpty(bookUpdate::setOnSale, updateBookApp.getOnSale());
    updateIfNotNullOrEmpty(bookUpdate::setUrlImage, updateBookApp.getUrlImage());
    BookEntity updater = bookRepository.save(bookUpdate);
    if (updater.getIsbn().equals(bookUpdate.getIsbn()) && updater.getSellerUser().equals(bookUpdate.getSellerUser())) {
        return updater;
    } else {
        utilLogs.logApiError("Error Update Book " + updater.getIsbn() + " seller user: " + updater.getSellerUser());
        return null;
    }
}
```


B-003 Buscar todos los Libros:

Análisis Inicial

```
final BookEntity validateBook = booksService.findallByisbnandseller(newBookReq.getIsbn(), newBookReq.getSellerUser());
if (validateBook != null) {
    responseNew.setSuccess(false);
    responseNew.setCodigo("Book-01");
    responseNew.setMessage("Book already exists");
    logbook = responseNew.getCodigo() + " ISBN: " + newBookReq.getIsbn() + " Seller user" + newBookReq.getSellerUser() + " " + responseNew.getMessage();
    utilLogs.logApi(1, logbook, BookController.class.getSimpleName(), request.getRemoteAddr());
} else {
    newBookReq.setID(newBookReq.getIsbn() + newBookReq.getSellerUser());
    final BookEntity createBook = booksService.saveNewBook(newBookReq);
    if (createBook != null) {
        responseNew.setSuccess(true);
        responseNew.setCodigo("Book-01");
        responseNew.setMessage("Book created successfully");
        responseNew.setData(newBookReq);
        logbook = responseNew.getMessage() + ": " + responseNew.getData() + " ";
    } else {
        responseNew.setSuccess(false);
        responseNew.setCodigo("ErrorBook-01");
        responseNew.setMessage("Error creating Book");
        logbook = responseNew.getCodigo() + " ISBN: " + newBookReq.getIsbn() + " sellerBook: " + newBookReq.getSellerUser();
        utilLogs.logApiError(logbook);
    }
    utilLogs.logApi(1, logbook, BookController.class.getSimpleName(), request.getRemoteAddr());
}
return responseNew;
```

Recomendaciones

- R-01: se debe puede solucionar esto optimizando a través de un método constructor para respuestas tipi success o error

Correcciones:

B-004 Eliminar libro:

Análisis Inicial

```
final BookEntity validateBook = booksService.findallByisbnandseller(newBookReq.getIsbn(), newBookReq.getSellerUser());
if (validateBook != null) {
    responseNew.setSuccess(false);
    responseNew.setCodigo("Book-01");
    responseNew.setMessage("Book already exists");
    logbook = responseNew.getCodigo() + " ISBN: " + newBookReq.getIsbn() + " Seller user" + newBookReq.getSellerUser() + " " + responseNew.getMessage();
    utilLogs.logApi(1, logbook, BookController.class.getSimpleName(), request.getRemoteAddr());
} else {
    newBookReq.setID(newBookReq.getIsbn() + newBookReq.getSellerUser());
    final BookEntity createBook = booksService.saveNewBook(newBookReq);
    if (createBook != null) {
        responseNew.setSuccess(true);
        responseNew.setCodigo("Book-01");
        responseNew.setMessage("Book created successfully");
        responseNew.setData(newBookReq);
        logbook = responseNew.getMessage() + ": " + responseNew.getData() + " ";
    } else {
        responseNew.setSuccess(false);
        responseNew.setCodigo("ErrorBook-01");
        responseNew.setMessage("Error creating Book");
        logbook = responseNew.getCodigo() + " ISBN: " + newBookReq.getIsbn() + " sellerBook: " + newBookReq.getSellerUser();
        utilLogs.logApiError(logbook);
    }
    utilLogs.logApi(1, logbook, BookController.class.getSimpleName(), request.getRemoteAddr());
}
return responseNew;
```


Recomendaciones

- R-01: se debe puede solucionar esto optimizando a través de un método constructor para respuestas tipi sucess o error

Correcciones:

B-005 Buscar Información libro por ISBN:

Análisis Inicial

```
final BookEntity validateBook = booksService.findAllByisbnandseller(newBookReq.getIsbn(), newBookReq.getSellerUser());
if (validateBook != null) {
    responseNew.setSuccess(false);
    responseNew.setCodigo("Book-01");
    responseNew.setMessage("Book already exists");
    logbook = responseNew.getCodigo() + " ISBN: " + newBookReq.getIsbn() + " Seller user" + newBookReq.getSellerUser() + " " + responseNew.getMessage();
    utilLogs.logApi(1, logbook, BookController.class.getSimpleName(), request.getRemoteAddr());
} else {
    newBookReq.setID(newBookReq.getIsbn() + newBookReq.getSellerUser());
    final BookEntity createBook = booksService.saveNewBook(newBookReq);
    if (createBook != null) {
        responseNew.setSuccess(true);
        responseNew.setCodigo("Book-01");
        responseNew.setMessage("Book created successfully");
        responseNew.setData(newBookReq);
        logbook = responseNew.getMessage() + ": " + responseNew.getData() + " ";
    } else {
        responseNew.setSuccess(false);
        responseNew.setCodigo("ErrorBook-01");
        responseNew.setMessage("Error creating Book");
        logbook = responseNew.getCodigo() + " ISBN: " + newBookReq.getIsbn() + " sellerBook: " + newBookReq.getSellerUser();
        utilLogs.logApiError(logbook);
    }
    utilLogs.logApi(1, logbook, BookController.class.getSimpleName(), request.getRemoteAddr());
}
return responseNew;
```

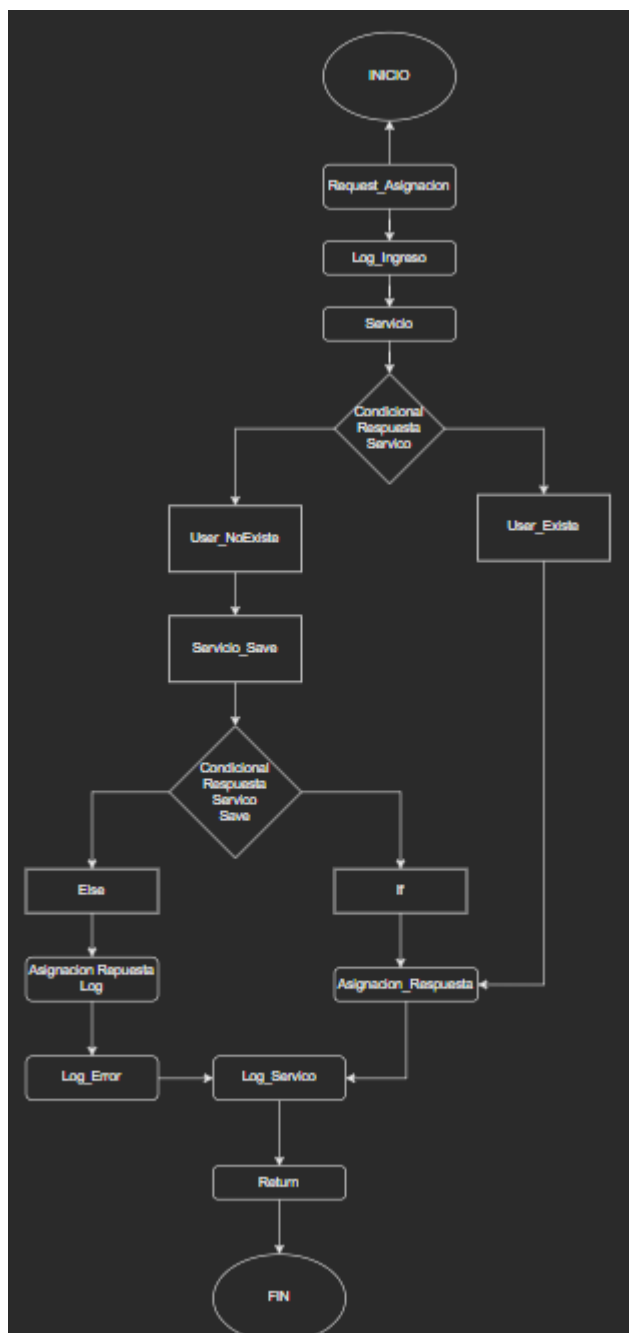
Recomendaciones

- R-01: se debe puede solucionar esto optimizando a través de un método constructor para respuestas tipi sucss o error

Correcciones:*Diagramas*

B-001 Registro de libros:

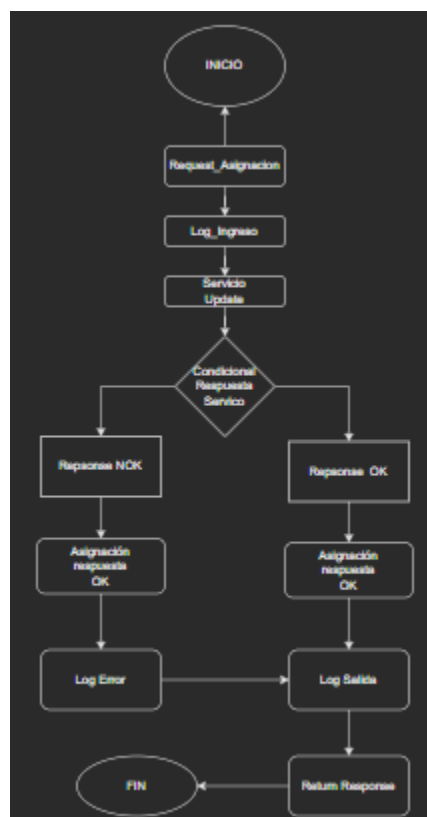
Flujo



Grafos



B-002 Actualizar libros:
Flujo

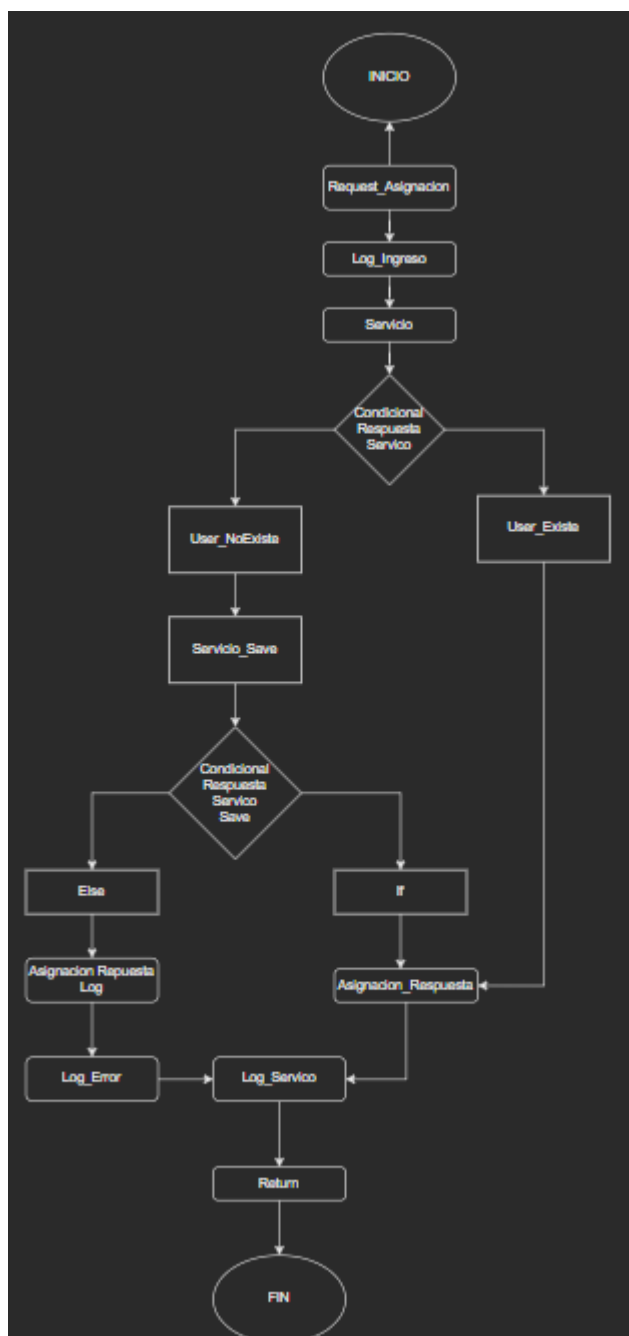


Grafos



B-003 Buscar todos los Libros:

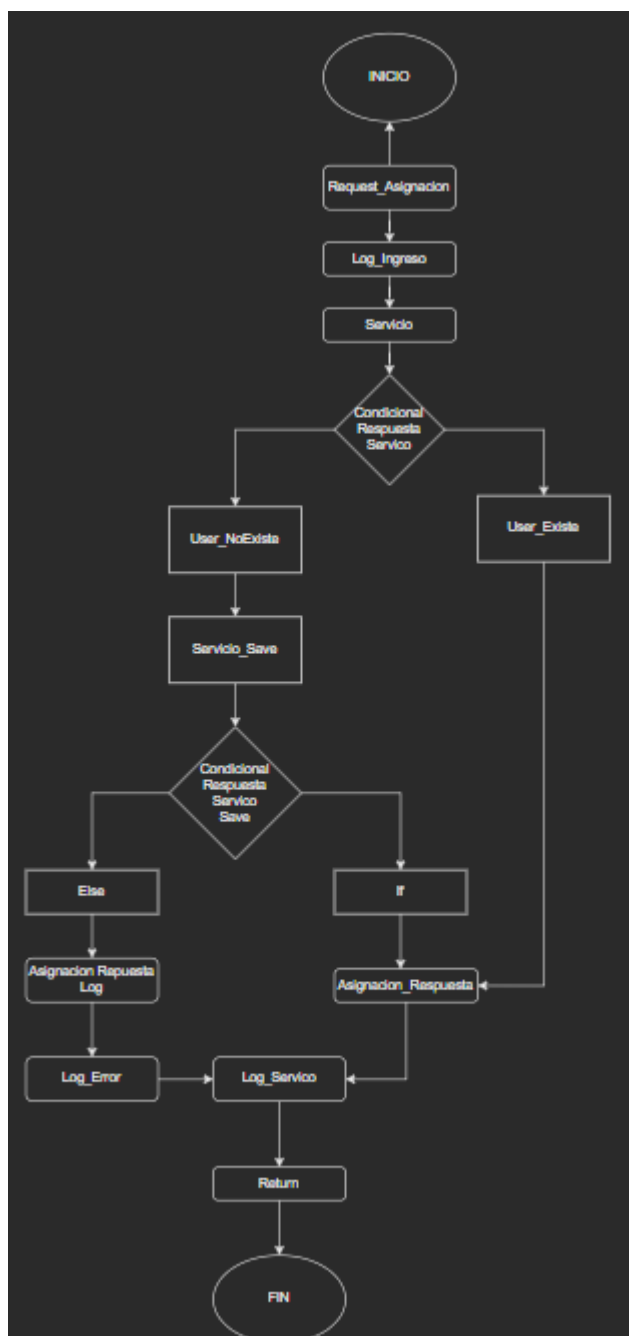
Flujo



Grafos



B-004 Eliminar libro:
Flujo

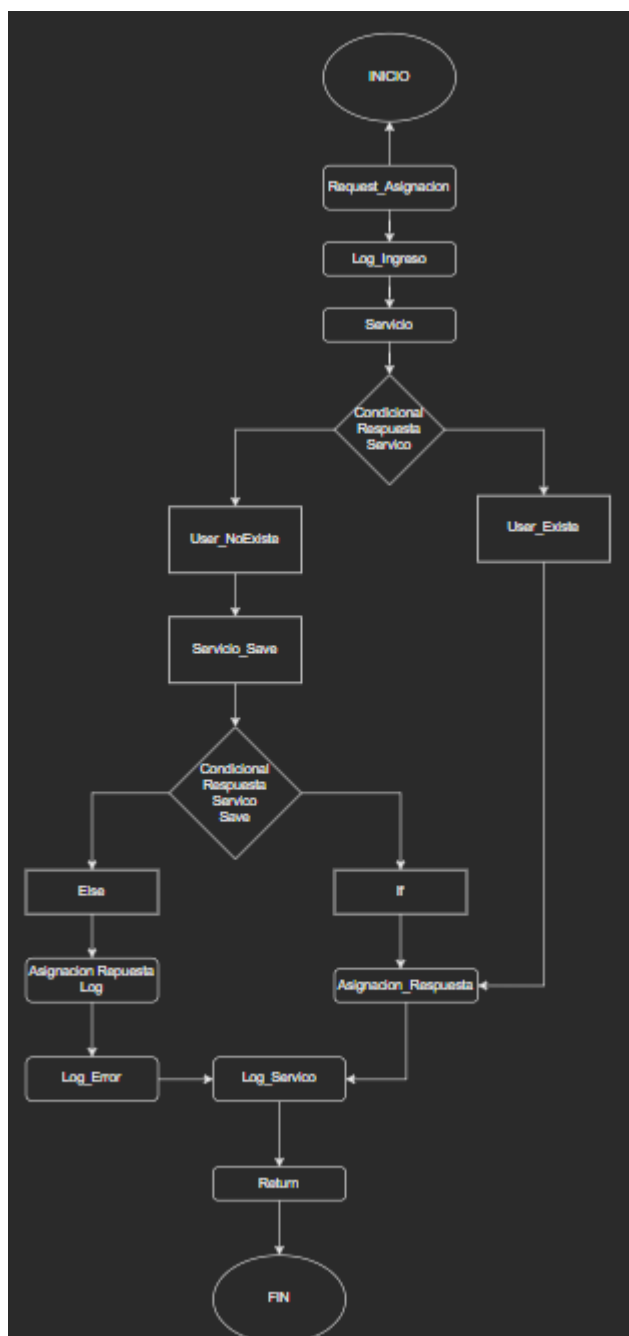


Grafos



B-005 Buscar Información libro por ISBN:

Flujo



Grafos



Métricas

B-001 Registro de libros:

Escenario 01:

METRICAS				
Código			B-001	
Escenario			1	
Cuantitativas				
Tiempo Latencia	Cantidad Errores	Numero Intentos	Porcentaje Éxito	Grado Complejidad
88ms	0	1	100	2
Cualitativas				
Nivel Satisfacción	Facilidad Mantenimiento	Nivel Errores	Complejidad Prueba	Prueba Exitosa
Excelente	Alto	Ninguno	Fácil	Si

Escenario 02:

METRICAS				
Código			B-001	
Escenario			2	
Cuantitativas				
Tiempo Latencia	Cantidad Errores	Numero Intentos	Porcentaje Éxito	Grado Complejidad
678 ms	0	1	100	2
Cualitativas				
Nivel Satisfacción	Facilidad Mantenimiento	Nivel Errores	Complejidad Prueba	Prueba Exitosa
Excelente	Alto	Ninguno	Medio	Si

Escenario 03:

METRICAS				
Código			B-001	
Escenario			3	
Cuantitativas				
Tiempo Latencia	Cantidad Errores	Numero Intentos	Porcentaje Éxito	Grado Complejidad
1223ms	0	3	100	2
Cualitativas				
Nivel Satisfacción	Facilidad Mantenimiento	Nivel Errores	Complejidad Prueba	Prueba Exitosa
Excelente	Alto	Ninguno	Alto	Si

B-002 Actualizar libros:

Escenario 01:

METRICAS				
Código			B-002	
Escenario			1	
Cuantitativas				
Tiempo Latencia	Cantidad Errores	Numero Intentos	Porcentaje Éxito	Grado Complejidad
585 ms	0	1	100	1
Cualitativas				
Nivel Satisfacción	Facilidad Mantenimiento	Nivel Errores	Complejidad Prueba	Prueba Exitosa
Excelente	Medio	Ninguno	fácil	Si

Escenario 02:

METRICAS				
Código			B-002	
Escenario			2	
Cuantitativas				
Tiempo Latencia	Cantidad Errores	Numero Intentos	Porcentaje Éxito	Grado Complejidad
1016 ms	1	2	50	1
Cualitativas				
Nivel Satisfacción	Facilidad Mantenimiento	Nivel Errores	Complejidad Prueba	Prueba Exitosa
Bueno	Alto	Ninguno	Medio	Si

B-003 Buscar todos los Libros:

Escenario 01:

METRICAS				
Código			B-001	
Escenario			1	
Cuantitativas				
Tiempo Latencia	Cantidad Errores	Numero Intentos	Porcentaje Éxito	Grado Complejidad
88ms	0	1	100	2
Cualitativas				
Nivel Satisfacción	Facilidad Mantenimiento	Nivel Errores	Complejidad Prueba	Prueba Exitosa
Excelente	Medio	Ninguno	Facil	Si

B-004 Eliminar libro:

Escenario 01:

METRICAS				
Código			B-001	
Escenario			1	
Cuantitativas				
Tiempo Latencia	Cantidad Errores	Numero Intentos	Porcentaje Éxito	Grado Complejidad
88ms	0	1	100	2
Cualitativas				
Nivel Satisfacción	Facilidad Mantenimiento	Nivel Errores	Complejidad Prueba	Prueba Exitosa
Excelente	Medio	Ninguno	Facil	Si

B-005 Buscar Información libro por ISBN:

Escenario 01:

METRICAS				
Código			B-001	
Escenario			1	
Cuantitativas				
Tiempo Latencia	Cantidad Errores	Numero Intentos	Porcentaje Éxito	Grado Complejidad
88ms	0	1	100	2
Cualitativas				
Nivel Satisfacción	Facilidad Mantenimiento	Nivel Errores	Complejidad Prueba	Prueba Exitosa
Excelente	Medio	Ninguno	fácil	Si