

# **DIPLOMADO EN**

# **“QUALITY ASSURANCE Y SOFTWARE**

# **TESTING”**



## **MODULO II: WEBSERVICES TESTING MOBILE**

**Título: “Pruebas al API Restful Booker”**

**Presentado por: Silvia E Rodríguez Sanabria**

**Fecha: 04/01/2023**

## ÍNDICE GENERAL

1. INTRODUCCIÓN .....	1
1.2. SOBRE LA APLICACIÓN RESTFUL-BOOKER .....	1
2. ALCANCES Y DELIMITACIONES .....	1
4. TECNOLOGIAS UTILIZADAS .....	9
4.1. TESTLINK .....	9
4.2. XAMPP: .....	15
4.3. POSTMAN .....	16
4.4. SOAP UI: .....	19
5. CONCLUSIONES .....	22
6. RECOMENDACIONES.....	24

## ÍNDICE DE TABLAS

Tabla 1. Caso de Prueba1 .....	4
Tabla 2. Caso de Prueba 2.....	5
Tabla 3. Caso de Prueba 3 .....	6
Tabla 4. Caso de Prueba 4.....	7
Tabla 5. Caso de Prueba 5.....	9
Tabla 2. Herramientas.....	9

## ÍNDICE DE FIGURAS

Figura 1. Usuarios.....	10
Figura 2. Roles .....	11
Figura 3. Test plan .....	11
Figura 4. Roles asignados a proyecto .....	11
Figura 5. Casos de prueba testlink: Caso de prueba RB-1 .....	12
Figura 6. Caso de prueba testlink: Caso de prueba RB-2 .....	12
Figura 7. Caso de prueba testlink: Caso de prueba RB-3 .....	13
Figura 8. Caso de prueba testlink: Caso de prueba RB-4 .....	13
Figura 9. Caso de prueba testlink: Caso de prueba RB-5 .....	14
Figura 10. Descripción general de pruebas ejecutadas .....	14
Figura 11. Descripción general de pruebas fallidas .....	15
Figura 12. Resultado de las pruebas ejecutadas .....	15
Figura 13. Postman: CreateToken .....	17
Figura 14. Postman: GetBookId.....	17
Figura 15. Postman: GetBooks.....	18
Figura 16. Postman: GetBookByName.....	18
Figura 17. Postman: CreateBooking .....	19
Figura 18. SOAP UI: GetBooks .....	20
Figura 19. SOAP UI: GetBookByName .....	20
Figura 20. SOAP UI: CreateBook .....	21
Figura 21. SOAP UI: Resultados.....	21
Figura 22. POSTMAN: Resultados GetByName .....	23
Figura 23. SOAP UI: Resultados GetByName.....	23
Figura 24. POSTMAN: Resultados CreateBook .....	23
Figura 25. SOAP UI: ResultadosCreateBook .....	24

## 1. INTRODUCCIÓN

### 1.2. SOBRE LA APLICACIÓN RESTFUL-BOOKER

**RESTFUL-BOOKER** es una API que se usa para obtener información y realizar pruebas con distintas herramientas.

RESTFUL-BOOKER es una API Web (Create, Read, Update, Delete) que viene con funciones de autenticación(Token) y está cargada con errores para realizar la exploración correspondiente.

La API viene precargada con 10 registros para que se pueda realizar distintas peticiones y se restablece cada 10 minutos a su estado predeterminado. Mencionar que viene con documentación API detallada para realizar las pruebas API de inmediato.

## 2. ALCANCES Y DELIMITACIONES

Se realizará el control de calidad de los siguientes casos de pruebas a los servicios:

### **AUTH**

- CreateToken

### **BOOKING**

- GetBookingIds
- GetBookingByName
- CreateBooking
- UpdateBooking

### 3. DISEÑO DE CASOS DE PRUEBA

En la API RESTFUL BOOKER se tienen actualmente las opciones que se testearán son los siguientes:

url= <https://restful-booker.herokuapp.com/>

url= <https://restful-booker.herokuapp.com/booking/>

#### AUTH

- POST/ [{{url}}](#) /auth, este recurso crea un nuevo token de autenticación para insertar o actualizar un recurso (libro).

#### BOOKING

- GET/ [{{url}}](#), este recurso obtiene la información de todos los libros que se encuentran registrados en la API.
- GET/ [{{url}}](#) /[{id}](#), este recurso obtiene la información de la reserva del libro a través del [id](#) de identificación del libro.
- GET/ [{{url}}](#)?firstname=[{Name}](#)&lastname=[{LastName}](#), este recurso obtiene los [Id's](#) de todos los libros reservados basados en los criterios de búsqueda (nombre y apellido).
- POST/ [{{url}}](#), este recurso crea una nueva reserva en la API.

ID	01
TITULO	Verificar que el método POST/ <a href="#">{{url}}</a> /auth, cree un token de autenticación.
AMBIENTE	Windows 11 Pro v21H2, Php versión 5.5.30, Xampp v3.2.2, Chrome versión 109.0.5414.120, Postman versión 10.9.3, SoapUI versión 5.7.0
PRIORIDAD	Medio
DESCRIPCIÓN	Este test case verifica que el método <a href="#">POST/ {{url}} /auth</a> , cree un token de autenticación que luego será validada por el servidor para acceder al método PUT/ <a href="#">{{url}} /{id}</a> .

PRECONDICIONES	
<ul style="list-style-type: none"> <li>• Configurar el navegador Google Chrome como predeterminado.</li> <li>• Se debe instalar previamente Postman en el ordenador.</li> <li>• El usuario debe tener acceso a internet.</li> </ul>	
PASOS	RESULTADOS ESPERADOS
<ol style="list-style-type: none"> <li>1. Hacer clic en el link:  <a href="https://restful-booker.herokuapp.com/apidoc/#api-Auth-CreateToken">https://restful-booker.herokuapp.com/apidoc/#api-Auth-CreateToken</a> </li> <li>2. Hacer clic en Postman y crear una colección con el nombre de <b>"RestfulBooker"</b></li> <li>3. Hacer clic en la colección <b>RestfulBooker</b> y creamos una <b>petición (Request)</b>.</li> <li>4. Hacer clic en el icono "..." que está al lado de la petición <b>(Request)</b> creada, después hacer clic en la opción <b>Rename</b> y renombrar como <b>CreateToken</b>.</li> <li>5. Hacer click en el icono <b>Environment quick look</b></li> <li>6. Ingresar <b>"urlt"</b> en el campo: <b>variable</b>, después copiar <b>"https://restful-booker.herokuapp.com/"</b> en el campo: <b>initial value</b> y hacer clic en el icono guardar.</li> <li>7. Seleccionar la opción <b>POST</b> e ingresar <b>"{{urlt}}/auth"</b></li> <li>8. Hacer clic en Body y copiar lo siguiente:  <pre>{   "username": "admin",   "password": "password123" }</pre> </li> </ol>	<p>La página Restful se carga con todas sus opciones en el navegador predeterminado.</p> <p>Se inicia Postman y se crea la colección <b>"RestfulBooker"</b></p> <p>Se visualiza una ventana donde se harán las pruebas correspondientes.</p> <p>Se visualizará el cambio de nombre en la petición creada como <b>CreateToken</b>.</p> <p>Se visualiza una ventana donde se ingresará la url de la página <b>RestfulBooker</b>.</p> <p>Se guarda la configuración con éxito.</p> <p>Se ingresaron los datos y se muestra sin errores.</p> <p>Se realiza con éxito el copiado del username y password para realizar la petición.</p>


<p>9. Hacer clic en Authorization y seleccionar en la opción <b>Type: Bearer Token</b>.</p> <p>10. Hacer clic en la opción <b>Body</b> y clic en el botón <b>Send</b>.</p>	<p>Se realiza con éxito la autorización para crear el token.</p> <p>Se visualiza el token en la ventana de resultados mostrando el código 200 Ok, además del token generado.</p>
ARCHIVO:	 Create_Token.png

Tabla 1. Caso de Prueba 1

ID	02
TITULO	Verificar que el método GET/ {{url}} /{id}, obtiene toda la información de la reserva.
AMBIENTE	Windows 11 Pro v21H2, Php versión 5.5.30, Xampp versión3.2.2, Chrome versión 109.0.5414.120, Postman versión 10.9.3, SoapUI versión 5.7.0
PRIORIDAD	High
DESCRIPCIÓN	Este caso de prueba verifica que el método GET/ {{url}} /{Id} , devuelva la información de la reserva solicitada por el Id (identificador de la reserva) a través de la petición GET.
PRECONDIONES	
<ul style="list-style-type: none"><li>• Configurar el navegador Google Chrome como predeterminado.</li><li>• Se debe instalar previamente Postman en el ordenador.</li><li>• El usuario debe tener acceso a internet.</li></ul>	
PASOS	RESULTADOS ESPERADOS
<ol style="list-style-type: none"><li>1. Hacer clic en el link: https://restful-booker.herokuapp.com/apidoc/#api-Booking-GetBookings</li><li>2. Hacer clic en Postman</li><li>3. Hacer clic en la colección <b>ResfulBooker</b> y creamos una <b>petición (Request)</b>.</li><li>4. Hacer clic en el icono “...” que está al lado de la petición <b>(Request)</b> creada, después</li></ol>	<p>La página Restful se carga con todas sus opciones en el navegador predeterminado.</p> <p>Se inicia Postman.</p> <p>Se visualiza una ventana donde se harán las pruebas correspondientes.</p> <p>Se visualiza el cambio de nombre en la petición creada como <b>GetBookId</b>.</p>



<p>hacer clic en la opción <b>Rename</b> y renombrar como <b>GetBookId</b>.</p> <p>5. Hacer click en el icono <b>Environment quick look</b></p> <p>6. Ingresar “url” en el campo: <b>variable</b>, después copiar “https://restful-booker.herokuapp.com/booking” en el campo: <b>initial value</b> y hacer clic en el icono guardar.</p> <p>7. Seleccionar la opción <b>GET</b> e ingresar “{{url}}/{{id}}”, en la opción {id} ingresar el Id de la reserva a buscar.</p> <p>8. Hacer clic en el botón <b>Send</b>.</p>	<p>Se visualiza una ventana donde se ingresará la url de la página <b>RestfulBooker</b>.</p> <p>Se guarda la configuración con éxito.</p> <p>Se ingresaron la <b>url</b> y el <b>Id</b> (identificador) de la reserva solicitada con éxito.</p> <p>Se visualiza los resultados de la reserva buscada a través de su Id (identificador), además se muestra el código 200 Ok.</p>
<b>ARCHIVO:</b>	 <b>Test_Case_Get_BookById.png</b>

Tabla 2. Caso de Prueba 2

<b>ID</b>	03
<b>TITULO</b>	Verificar que el método GET/ <b>{{url}}</b> , devuelva la inf. de las reservas registradas en la API.
<b>AMBIENTE</b>	Windows 11 Pro v21H2, Php versión 5.5.30, Xampp v3.2.2, Chrome versión 109.0.5414.120, Postman versión 10.9.3, SoapUI versión 5.7.0
<b>PRIORIDAD</b>	Alto
<b>DESCRIPCIÓN</b>	Este caso de prueba verifica que el método GET/ <b>{{url}}</b> , devuelva la información de las reservas registradas en la API.
<b>PRECONDICIONES</b>	
<ul style="list-style-type: none"> <li>• Configurar el navegador Google Chrome como predeterminado.</li> <li>• Se debe instalar previamente Postman en el ordenador.</li> <li>• El usuario debe tener acceso a internet.</li> </ul>	
<b>PASOS</b>	<b>RESULTADOS ESPERADOS</b>



<ol style="list-style-type: none"> <li>1. Hacer clic en el link "https://restful-booker.herokuapp.com/booking"</li> <li>2. Hacer clic en Postman.</li> <li>3. Hacer clic en la colección <b>ResfulBooker</b> y creamos una <b>petición (Request)</b>.</li> <li>4. Hacer clic en el icono "..." que esta al lado de la petición <b>(Request)</b> creada, después hacer clic en la opción <b>Rename</b> y renombrar como <b>GetBooks</b>.</li> <li>5. Seleccionar la opción <b>GET</b> e ingresar "<b>{{url}}</b>" y hacer clic en el botón <b>Send</b>.</li> </ol>	<p>La página Restful se carga con todas sus opciones en el navegador predeterminado.</p> <p>Se inicia Postman.</p> <p>Se visualiza una ventana donde se harán las pruebas correspondientes.</p> <p>Se visualiza el cambio de nombre en la petición creada como <b>GetBooks</b>.</p> <p>Se visualiza los resultados de todas las reservas que se encuentran registrados en la API, además se muestra el código 200 Ok.</p>
<b>ARCHIVO:</b>	 <b>Test_Case_GetBooks.png</b>

**Tabla 3. Caso de Prueba 3**

<b>ID</b>	<b>04</b>
<b>TITULO</b>	Verificar que el método GET/{{url}}?firstname={Name}&lastname={LastName}, obtiene las Id's de reservas.
<b>AMBIENTE</b>	Windows 11 Pro v21H2, Php versión 5.5.30, Xampp v3.2.2, Chrome versión 109.0.5414.120, Postman versión 10.9.3, SoapUI versión 5.7.0
<b>PRIORIDAD</b>	High
<b>DESCRIPCIÓN</b>	Este caso de prueba verifica que el método GET/{{url}}?firstname={Name}&lastname={LastName}, devuelva la información de todas las reservas realizadas de acuerdo al criterio de búsqueda.
<b>PRECONDIONES</b>	
<ul style="list-style-type: none"> <li>• Configurar el navegador Google Chrome como predeterminado.</li> <li>• Se debe instalar previamente Postman en el ordenador.</li> <li>• El usuario debe tener acceso a internet.</li> </ul>	



PASOS	RESULTADOS ESPERADOS
<ol style="list-style-type: none"> <li>1. Hacer clic en el link “https://restful-booker.herokuapp.com/booking”</li> <li>2. Hacer clic en Postman.</li> <li>3. Hacer clic en la colección <b>ResfulBooker</b> y creamos una <b>petición (Request)</b>.</li> <li>4. Hacer clic en el icono “...” que está al lado de la petición (<b>Request</b>) creada, después hacer clic en la opción <b>Rename</b> y renombrar como <b>GetBookByName</b>.</li> <li>5. Seleccionar la opción <b>GET</b> e ingresar <code>GET/{{url}}?firstname=eric</code> &amp;lastname=jones y hacer clic en el botón <b>Send</b>.</li> <li>6. Ingresar nuevamente <code>GET/{{url}}?firstname=Eric</code> &amp;lastname=Jones y hacer clic en el botón <b>Send</b>.</li> </ol>	<p>La página Restful se carga con todas sus opciones en el navegador predeterminado.</p> <p>Se inicia Postman.</p> <p>Se visualiza una ventana donde se harán las pruebas.</p> <p>Se visualiza el cambio de nombre en la petición creada como <b>GetBookByName</b>.</p> <p>Se visualiza los resultados de todas las reservas registradas por Eric Jones.</p> <p>Se visualiza los resultados de todas las reservas registradas por Eric Jones.</p>
ARCHIVO:	<div data-bbox="440 1220 496 1262"></div> <div data-bbox="496 1247 833 1281">Test_Case_SearchByName.png</div> <div data-bbox="440 1310 496 1352"></div> <div data-bbox="496 1337 937 1371">bug_case_sensitive_SearchByName.png</div>

Tabla 4. Caso de Prueba 4

ID	05		
TITULO	Verificar que el método POST/ {{url}}, registre una nueva reserva en la API.		
AMBIENTE	Windows 11 Pro v21H2, Php versión 5.5.30, Xampp v3.2.2, Chrome versión 109.0.5414.120, Postman versión 10.9.3, SoapUI versión 5.7.0		
PRIORIDAD	Alto		
DESCRIPCIÓN	Este caso de prueba verifica que el método POST/ {{url}}, registre una nueva reserva en la API.		
PRECONDIONES			
<ul style="list-style-type: none"><li>• Configurar el navegador Google Chrome como predeterminado.</li><li>• Se debe instalar previamente Postman en el ordenador.</li><li>• El usuario debe tener acceso a internet.</li></ul>			
PASOS		RESULTADOS ESPERADOS	
<ol style="list-style-type: none"><li>1. Hacer clic en el link “https://restful-booker.herokuapp.com/booking”</li><li>2. Hacer clic en Postman.</li><li>3. Hacer clic en la colección <b>ResfulBooker</b> y creamos una <b>petición (Request)</b>.</li><li>4. Hacer clic en el icono “...” que está al lado de la petición (<b>Request</b>) creada, después hacer clic en la opción <b>Rename</b> y renombrar como <b>CreateBooking</b>.</li><li>5. Hacer clic en <b>Body</b> y luego clic en la opción <b>raw</b> e ingresar: <pre>{   "bookingid": 1,   "booking": {     "firstname": "Jim",     "lastname": "Perez",     "totalprice": " "   } }</pre></li></ol>		<p>La página Restful se carga con todas sus opciones en el navegador predeterminado.</p> <p>Se inicia Postman.</p> <p>Se visualiza una ventana donde se harán las pruebas correspondientes.</p> <p>Se visualizará el cambio de nombre en la petición creada como <b>CreateBooking</b>.</p> <p>Se visualiza la nueva reserva registrada en la API,</p>	


<pre> "depositpaid": true, "bookingdates": {   "checkin": "2022-01-01",   "checkout": "2022-05-01" }, "additionalneeds": "Breakfast" } } </pre> <p>, después seleccionar la opción <b>POST</b> e ingresar <b>"{{url}}"</b> y hacer clic en el botón <b>Send</b>.</p>	
<b>ARCHIVO:</b>	 Bug_created_reservations_with_empty_fields.png

Tabla 5. Caso de Prueba 5

## 4. TECNOLOGIAS UTILIZADAS

Las herramientas que se utilizaron para el desarrollo de las pruebas son:

HERRAMIENTAS	
Testlink versión 1.9	
Xampp versión 3.2.2	Php versión 5.5.30
Postman Versión 10.9.3	UI Versión 10.9.3-ui-230203-1409
Chrome Versión 109.0.5414.120	
SoapUI versión 5.7.0	

Tabla 6. Herramientas

**4.1. TESTLINK:** es un sistema de gestión de pruebas basado en la web que facilita el aseguramiento de la calidad de software. Es desarrollado y mantenido por Teamtest.

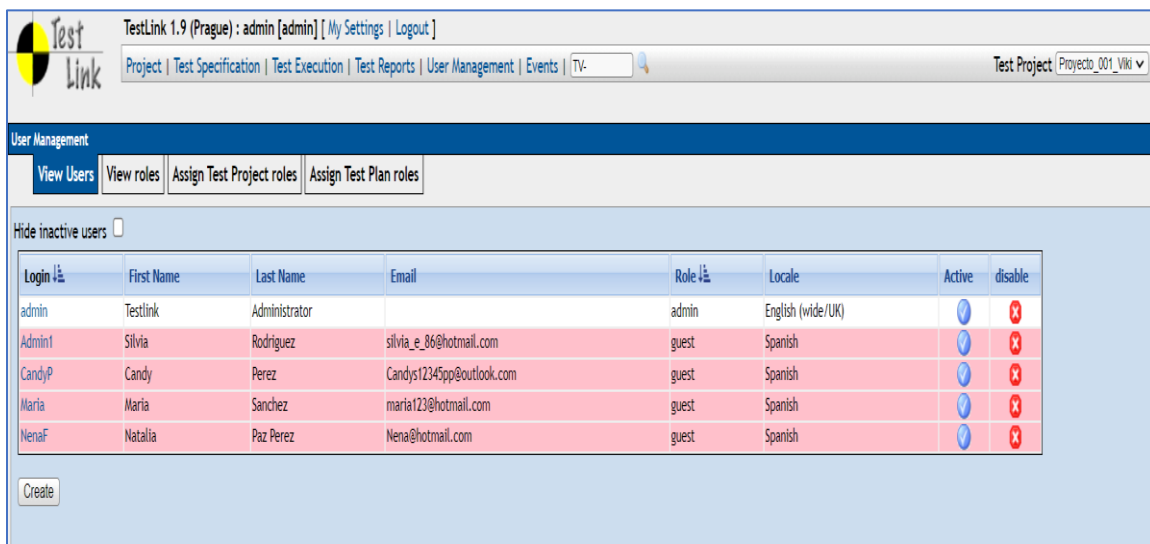
✓ **Importancia:** son las siguientes:

- Sincroniza la especificación de requisitos y la especificación de prueba.
- El usuario crea proyectos de prueba y documenta casos de prueba utilizando esta herramienta.
- Puede crear una cuenta para varios usuarios y asignar diferentes roles de usuario.

- El administrador puede gestionar la tarea de asignación de casos de pruebas.
- Es compatible con la ejecución automática y manual de casos de prueba.
- Los evaluadores pueden generar un plan de prueba y un informe de prueba en una fracción del tiempo con esta herramienta.
- Admite informes de prueba en varios formatos como Excel, MS Word y HTML. Aparte de estos, también admite la integración con muchos sistemas de seguimiento de defectos populares como JIRA, MANTIS, BUGZILLA, TRAC, etc.
- Los usuarios pueden acceder a su funcionalidad al mismo tiempo con sus credenciales y roles asignados.

Se realizaron las siguientes actividades sobre la herramienta:

### ❖ Crear usuarios



The screenshot shows the TestLink 1.9 (Prague) interface. The top navigation bar includes links for Project, Test Specification, Test Execution, Test Reports, User Management, and Events. The 'User Management' section is active, with sub-tabs for View Users, View roles, Assign Test Project roles, and Assign Test Plan roles. A table lists the following users:

Login	First Name	Last Name	Email	Role	Locale	Active	disable
admin	Testlink	Administrator		admin	English (wide/UK)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Admin1	Silvia	Rodriguez	silvia_e_86@hotmail.com	guest	Spanish	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CandyP	Candy	Perez	Candys12345pp@outlook.com	guest	Spanish	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Maria	Maria	Sanchez	maria123@hotmail.com	guest	Spanish	<input checked="" type="checkbox"/>	<input type="checkbox"/>
NenaF	Natalia	Paz Perez	Nena@hotmail.com	guest	Spanish	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Below the table is a 'Create' button.

Figura 1. Usuarios

## ❖ Asignar Roles

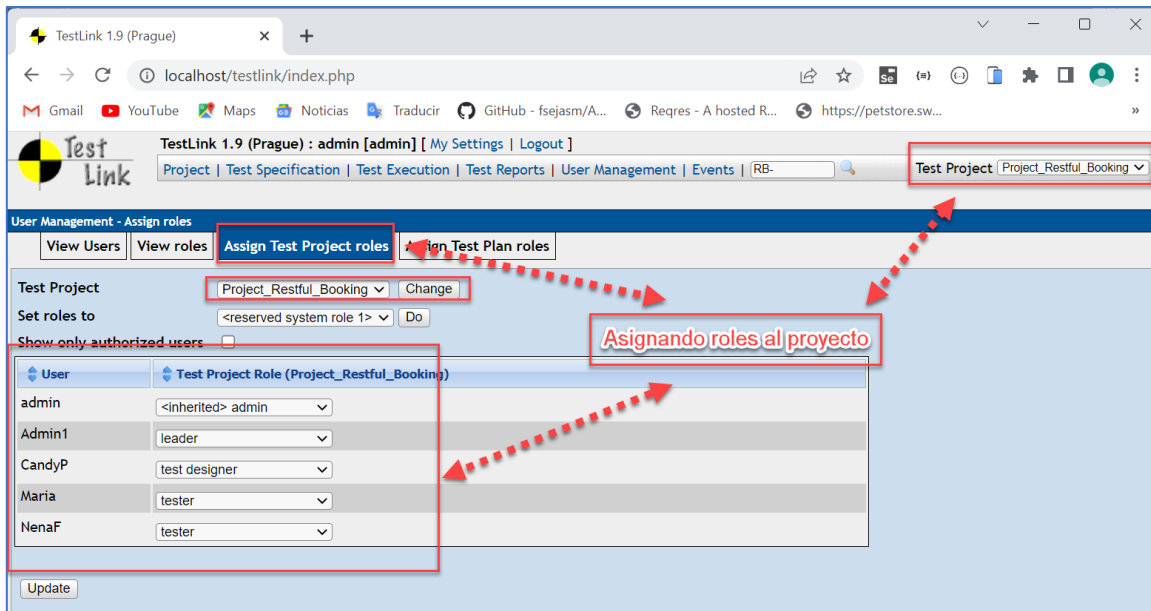


Figura 2. Roles

## ❖ Crear Test Plan

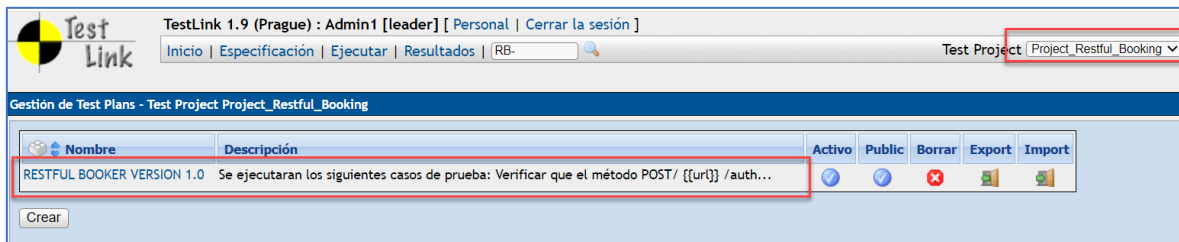


Figura 3. Test plan

## ❖ Asignar Test Plan

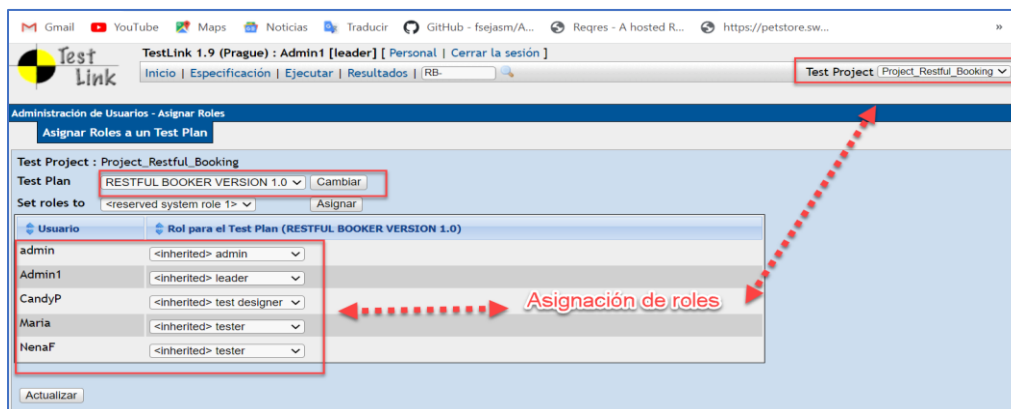


Figura 4. Roles asignados a proyecto

## ❖ Crear casos de Prueba

TestLink 1.9 (Prague) : Admin! [leader] [ Personal | Cerrar la sesión ]

Inicio | Especificación | Ejecutar | Resultados | RB-

Test Project: Project\_Restful\_Booking

Navegador - Especificación de Test

Settings

Actualizar el Árbol de Navegación después de cada operación

Filters

ID del Test Case: RB-

Título del Test Case:

Test Suite:

Execution type: [cualquiera]

Aplicar Filtro Reset Filters

Project\_Restful\_Booking (5)

Test\_Cases\_Restful\_Booking(5)

RB-1 Verificar que el método POST (/url) /auth, crea un token de autenticación

RB-2 Verificar que el método GET (/url) /id, devuelve toda la información de la reserva.

RB-3 Verificar que el método GET (/url) /id, devuelve la inf. de las reservas registradas en la API

RB-4 Verificar que el método GET (/url) /firstname=(Name)lastname=(LastName), obtiene las IDs de la reserva

RB-5 Verificar que el método POST (/url) /register, registre una nueva reserva en la API

Resumen

Environment: Windows 11 Pro v21H2, Php versión 5.5.30, Xampp v3.2.2, Chrome versión 109.0.5414.120, Postman versión 10.9.3

Priority: Alto

Description: Este test case verifica que el método POST (/url) /auth, crea un token de autenticación que luego será validado por el servidor para acceder al método PUT (/url) /id.

Preconditions

- Configurar el navegador Google Chrome como predeterminado.
- Se debe instalar previamente Postman en el ordenador.
- El usuario debe tener acceso a internet.

Step actions	Resultados Esperados
1. Hacer clic en el link: <a href="https://restful-booker.herokuapp.com/apidoc/#api-Auth-Credentials">https://restful-booker.herokuapp.com/apidoc/#api-Auth-Credentials</a>	La página Restful se carga con todas sus opciones en el navegador predeterminado
2. Hacer clic en Postman y crear una colección con el nombre de "RestfulBooker"	Se inicia Postman y se crea la colección "RestfulBooker"
3. Hacer clic en la colección RestfulBooker y creamos una petición (Request).	Se visualiza una ventana donde se harán las pruebas correspondientes.
4. Hacer clic en el icono "..." que está al lado de la petición (Request) creada, después hacer clic en la opción Rename y renombrar como CreateToken.	Se visualiza el cambio de nombre en la petición creada como CreateToken.
5. Hacer clic en el icono Environment quick look.	Se visualiza una ventana donde se ingresará la url de la página RestfulBooker
6. Ingresar "url" en el campo variable, después copiar "https://restful-booker.herokuapp.com/" en el campo initial value y hacer clic en el icono guardar	Se guarda la configuración con éxito.
7. Seleccionar la opción POST e ingresar "[url]/auth"	Se ingresaron los datos y se muestra sin errores
8. Hacer clic en Body y copiar lo siguiente: <pre>{   "username": "admin",   "password": "password123" }</pre>	Se realiza con éxito el copiado del username y password para realizar la petición.
9. Hacer clic en Authorization y seleccionar en la opción Type: Bearer Token.	Se realiza con éxito la autorización para crear el token.
10. Hacer clic en la opción Body y clic en el botón Send.	Se visualiza el token en la ventana de resultados mostrando el código 200 OK, además del token generado

Figura 5. Casos de prueba testlink: Caso de prueba RB-1

TestLink 1.9 (Prague) : Admin! [leader] [ Personal | Cerrar la sesión ]

Inicio | Especificación | Ejecutar | Resultados | RB-

Test Project: Project\_Restful\_Booking

Navegador - Especificación de Test

Settings

Actualizar el Árbol de Navegación después de cada operación

Filters

ID del Test Case: RB-

Título del Test Case:

Test Suite:

Execution type: [cualquiera]

Aplicar Filtro Reset Filters

Project\_Restful\_Booking (5)

Test\_Cases\_Restful\_Booking(5)

RB-1 Verificar que el método POST (/url) /auth, crea un token de autenticación

RB-2 Verificar que el método GET (/url) /id, devuelve toda la información de la reserva.

RB-3 Verificar que el método GET (/url) /id, devuelve la inf. de las reservas registradas en la API

RB-4 Verificar que el método GET (/url) /firstname=(Name)lastname=(LastName), obtiene las IDs de la reserva

RB-5 Verificar que el método POST (/url) /register, registre una nueva reserva en la API

Resumen

Environment: Windows 11 Pro v21H2, Php versión 5.5.30, Xampp v3.2.2, Chrome versión 109.0.5414.120, Postman versión 10.9.3

Priority: High

Description: Este caso de prueba verifica que el método GET (/url) /id, devuelva la información de la reserva solicitada por el id (identificador de la reserva) a través de la petición GET.

Preconditions

- Se debe instalar previamente Postman en el ordenador.
- Configurar el navegador Google Chrome como predeterminado.
- El usuario debe tener acceso a internet.

Step actions	Resultados Esperados
1. Hacer clic en el link: <a href="https://restful-booker.herokuapp.com/apidoc/#api-Booking-GetBookings">https://restful-booker.herokuapp.com/apidoc/#api-Booking-GetBookings</a>	La página Restful se carga con todas sus opciones en el navegador predeterminado
2. Hacer clic en Postman	Se inicia Postman.
3. Hacer clic en la colección RestfulBooker y creamos una petición (Request).	Se visualiza una ventana donde se harán las pruebas correspondientes.
4. Hacer clic en el icono "..." que está al lado de la petición (Request) creada, después hacer clic en la opción Rename y renombrar como GetBookId.	Se visualiza el cambio de nombre en la petición creada como GetBookId.
5. Hacer clic en el icono Environment quick look	Se visualiza una ventana donde se ingresará la url de la página RestfulBooker
6. Ingresar "url" en el campo variable, después copiar "https://restful-booker.herokuapp.com/bookings" en el campo initial value y hacer clic en el icono guardar	Se guarda la configuración con éxito
7. Seleccionar la opción GET e ingresar "[url]/id", en la opción (id) ingresar el id de la reserva a buscar.	Se ingresaron la url y el id (identificador) de la reserva solicitada con éxito.
8. Hacer clic en el botón Send.	Se visualiza los resultados de la reserva buscada a través de su id (identificador), además se muestra el código 200 OK.

Figura 6. Caso de prueba testlink: Caso de prueba RB-2

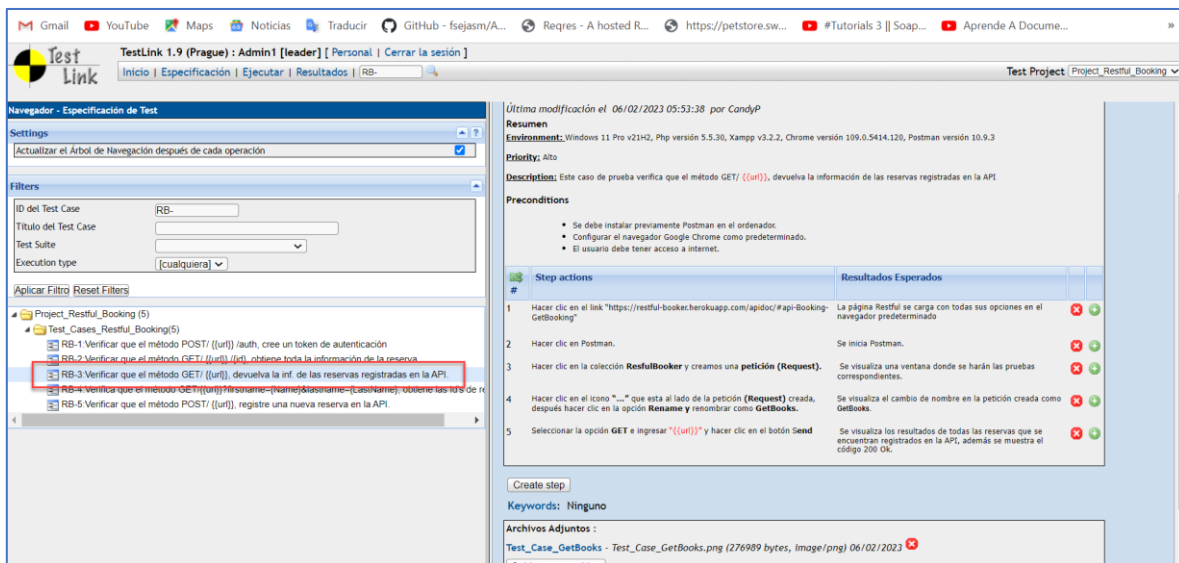


Figura 7. Caso de prueba testlink: Caso de prueba RB-3

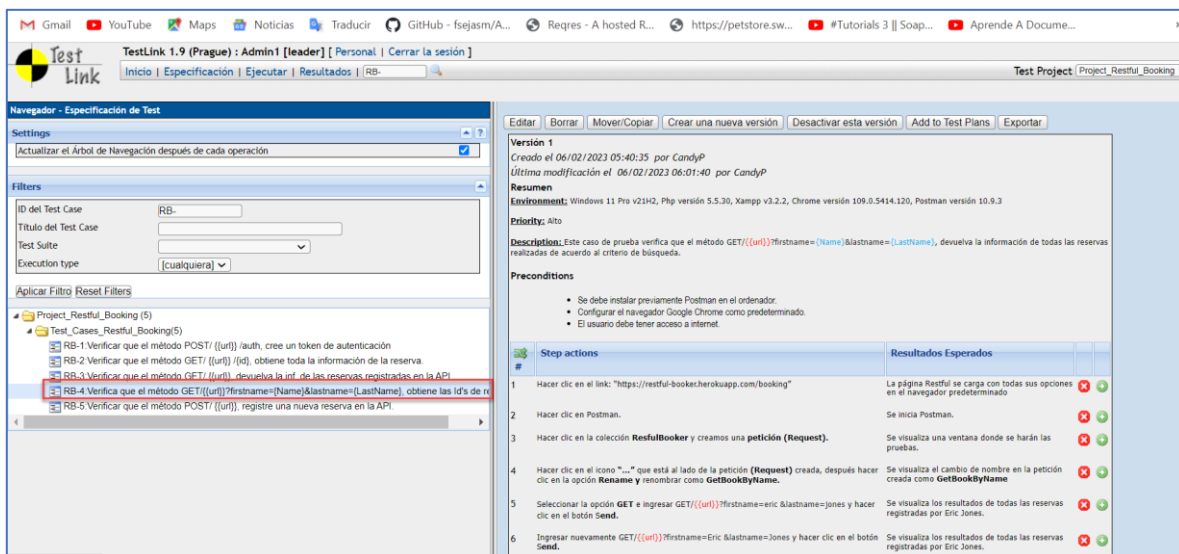


Figura 8. Caso de prueba testlink: Caso de prueba RB-4



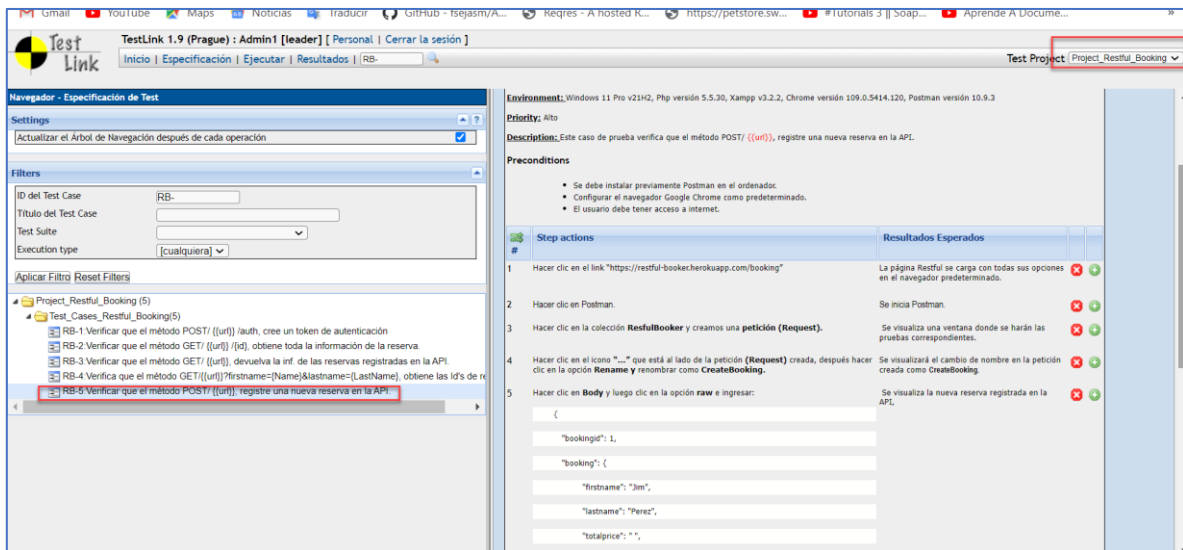


Figura 9. Caso de prueba testlink: Caso de prueba RB-5

## ❖ Descripción general de los casos de Prueba

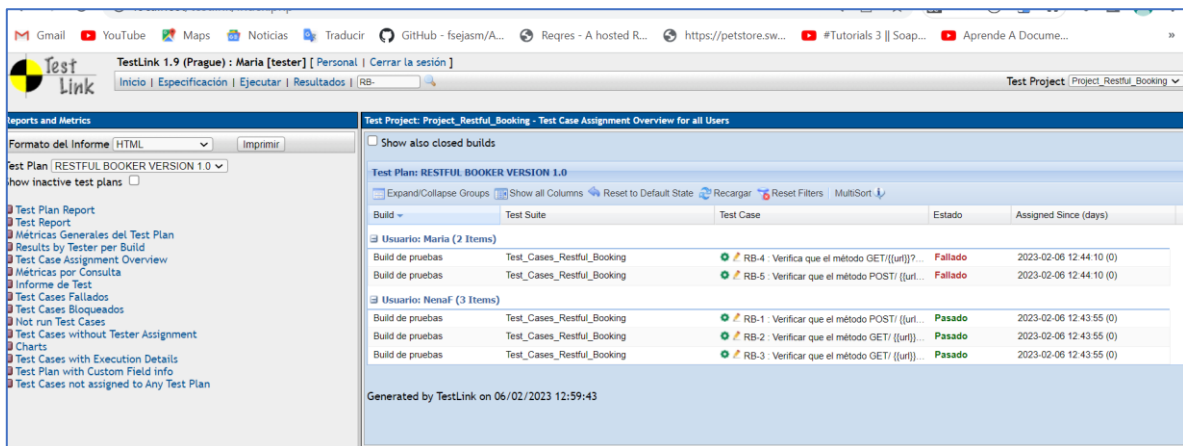


Figura 10. Descripción general de pruebas ejecutadas

## ❖ Descripción general de pruebas fallidas

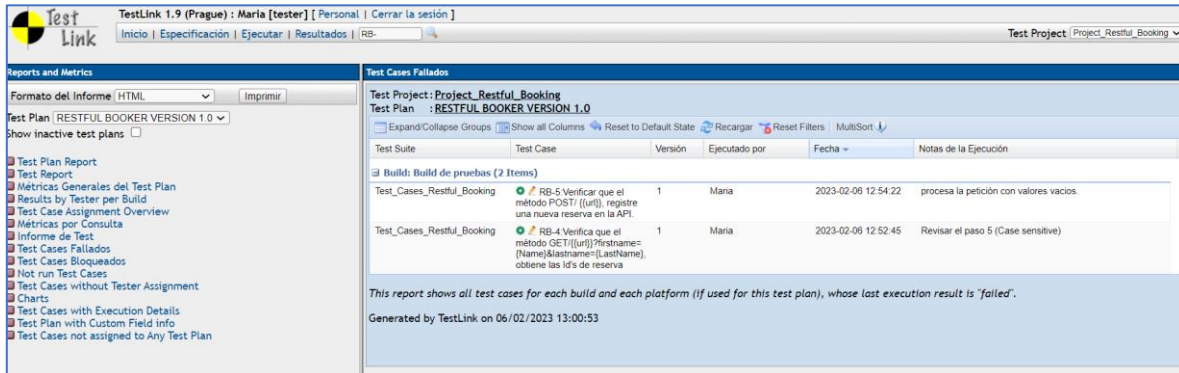


Figura 11. Descripción general de pruebas fallidas

## ❖ Gráficos

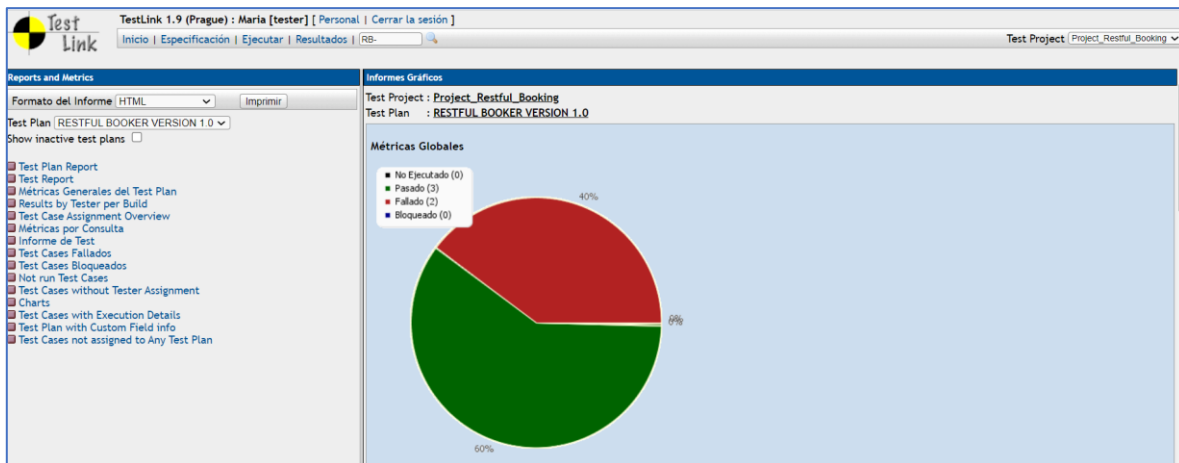


Figura 12. Resultado de las pruebas ejecutadas

**4.2. XAMPP:** es un paquete de software libre, que consiste principalmente en el sistema de gestión de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.

El nombre es en realidad un acrónimo:

- **X:** Hace referencia a los diferentes **sistemas operativos** en los que se puede instalar el programa, como Linux, Windows, Mac, Ubuntu, etc.
- **A:** Por el servidor web Apache que permitirá trabajar con las diferentes herramientas del paquete y que será la principal interfaz a usar.

- **M:** La letra M hace referencia a la incorporación del sistema de gestión de bases de datos conocido como **MySQL**; en algunas versiones de Xampp, sobre todo en las más recientes, incorpora **MariaDB**.
- **P:** Como todo servidor web de código abierto, Xampp utiliza un lenguaje de programación, siendo en este caso **PHP**; este lenguaje es bastante conocido y soporta diferentes sistemas de bases de datos.
- **P:** **Perl** es otro lenguaje de programación de Xampp, pero esta vez enfocado en la administración del sistema y programación de red.
- ✓ **Importancia:** son las siguientes:
  - Nos permite trabajar en un servidor local sin necesidad de tener conexión a internet.
  - Xampp ya **viene con todas las herramientas que necesitas** para trabajar en tus proyectos (panel de control, bases de datos, graphics draw, servidor apache, filezilla ftp serve).
  - Xampp elimina ese problema integrando Apache a un pack de herramientas que se instala fácilmente en cualquier sistema.
  - Xampp es gratis.

**4.3. POSTMAN:** es una plataforma de API para que los desarrolladores diseñen, construyan, prueben e iteren sus API

- ✓ **Importancia:** son las siguientes:
  - Postman tiene una **comunidad enorme de usuarios**.
  - Su **interfaz es muy intuitiva y sencilla**.
  - La plataforma **puede integrarse con otras herramientas**.
  - Ofrece la posibilidad de **añadir scripts** (JavaScript) para automatizar, configurar pruebas o agregar validaciones.
  - La herramienta permite **realizar trabajos en los que se puede colaborar** con el resto de miembros de un equipo.

Se realizaron las siguientes actividades sobre la herramienta:

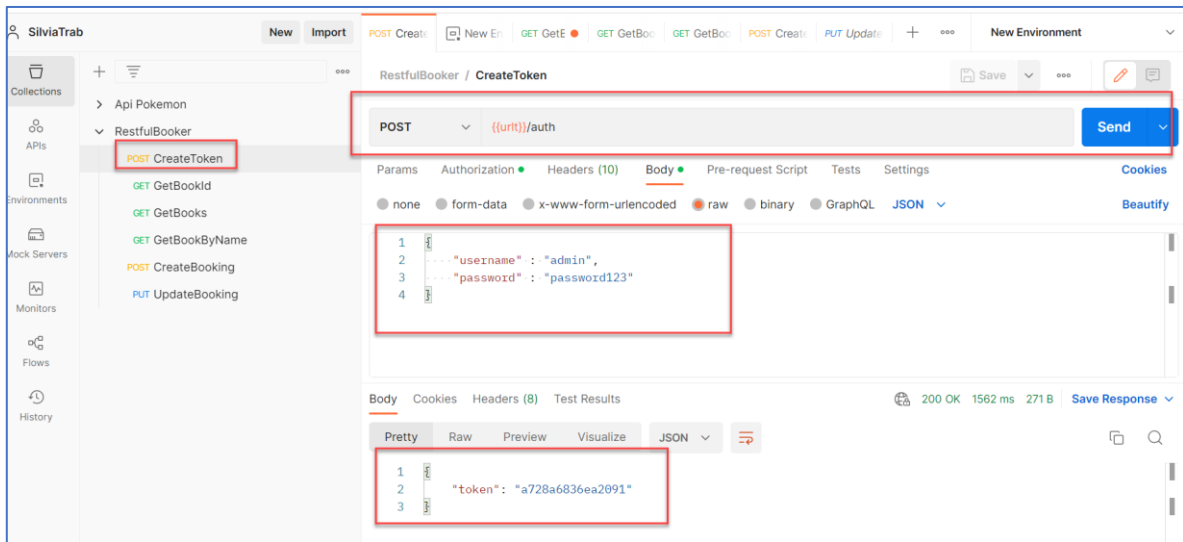


Figura 13. Postman: CreateToken

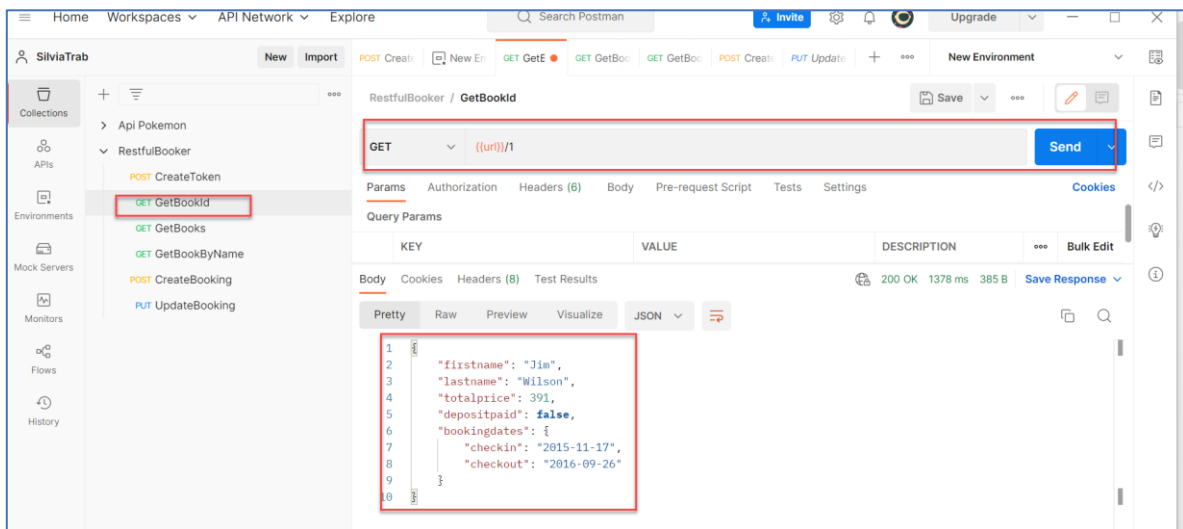


Figura 14. Postman: GetBookId

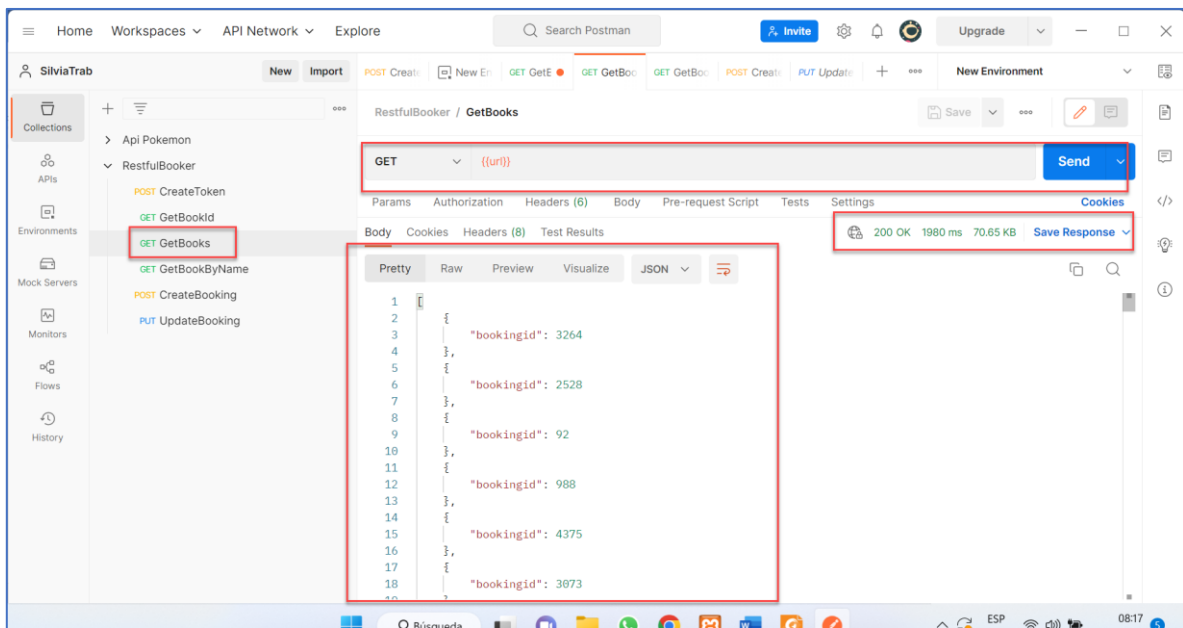


Figura 15. Postman: GetBooks

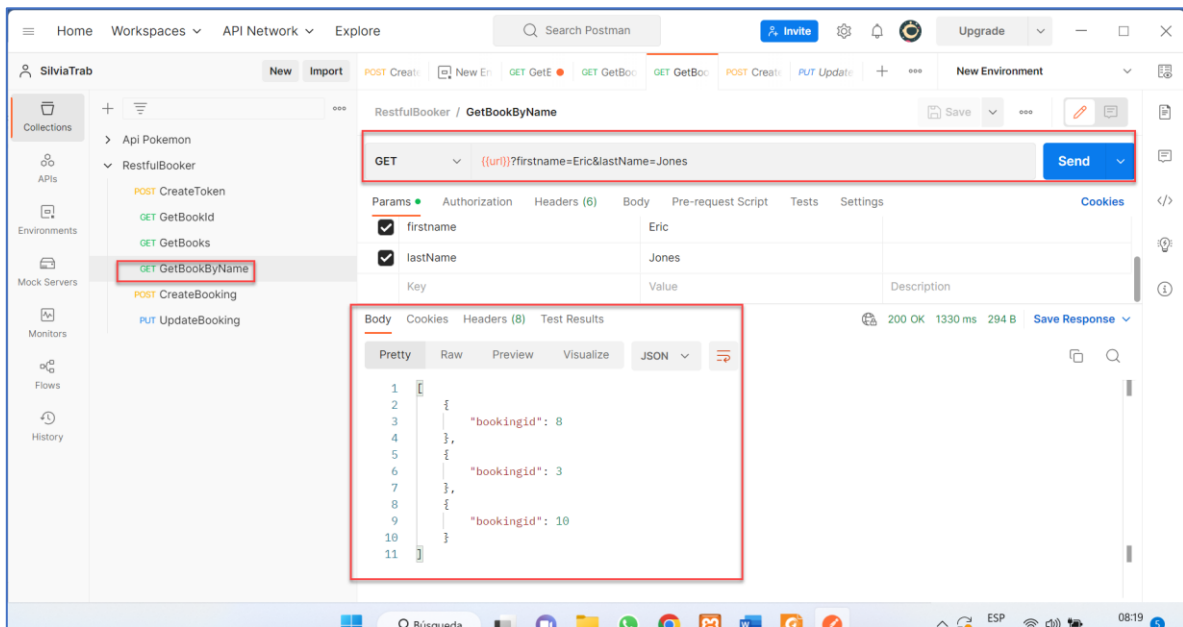
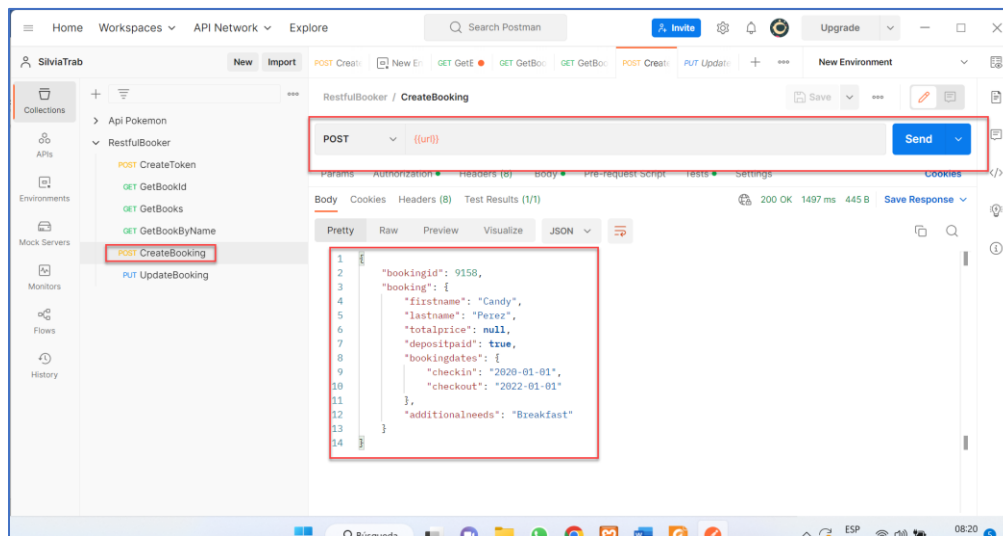


Figura 16. Postman: GetBookByName



**Figura 17. Postman: CreateBooking**

**4.4. SOAP UI:** es una herramienta, desarrollada en java, para la realización de pruebas a aplicaciones con arquitectura orientada a servicio y transferencia de estado representacional. Soporta múltiples protocolos como SOAP, REST, HTTP, JMS, AMF y JDBC.

Las ventajas son las siguientes:

- Es una aplicación muy completa, con muchas funcionalidades.
- Tiene mejor integración que Postman para trabajar con el protocolo SOAP.
- Es un proyecto mas duro y tiene mas tiempo en el mercado.
- Es una aplicación orientada mas a testing y no solo consumir API, documentarla y publicarla.
- Permite estructurar las pruebas en test suites, test cases y test steps.
- La ejecución de pruebas se puede integrar con herramientas tales como: Maven, motores de CI/CD, etc.
- Permite agregar scripts en lenguaje Groovy. Esto permite agregar validaciones, configurar y/o automatizar pruebas.

A continuación, se detallan en las siguientes figuras los resultados que generaron la Test Suite que se ejecutó en SOAP:

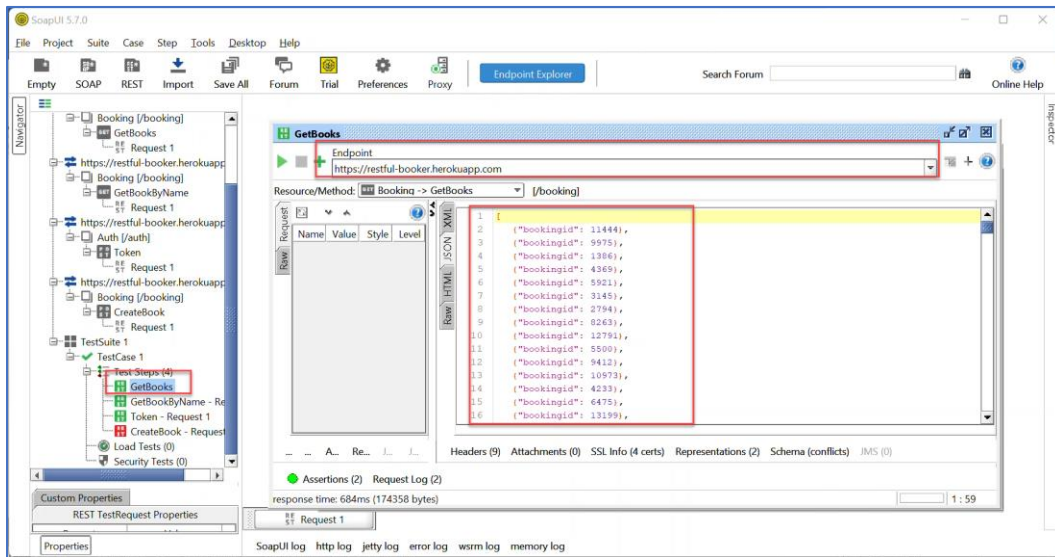


Figura 18. SOAP UI: GetBooks

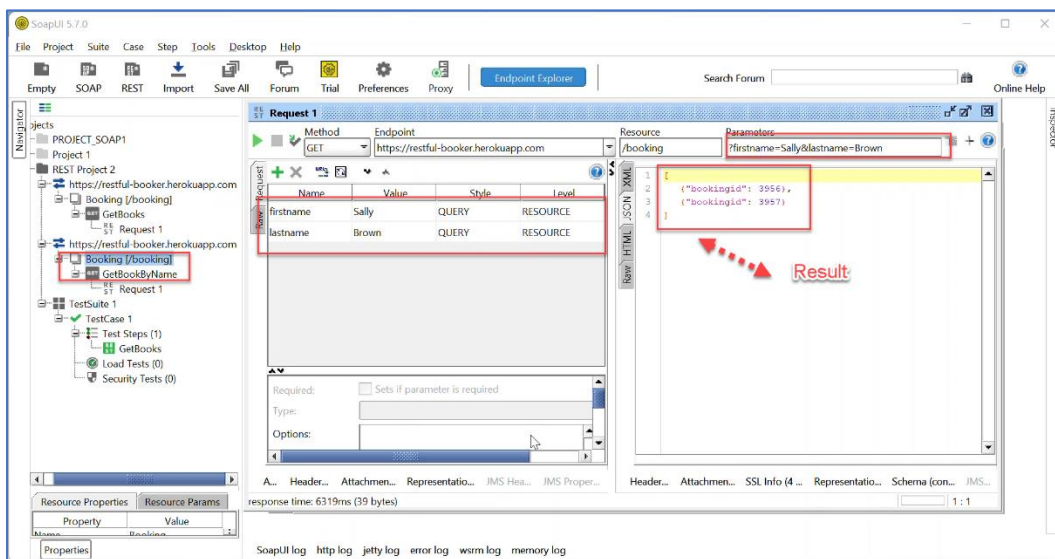


Figura 19. SOAP UI: GetBookByName

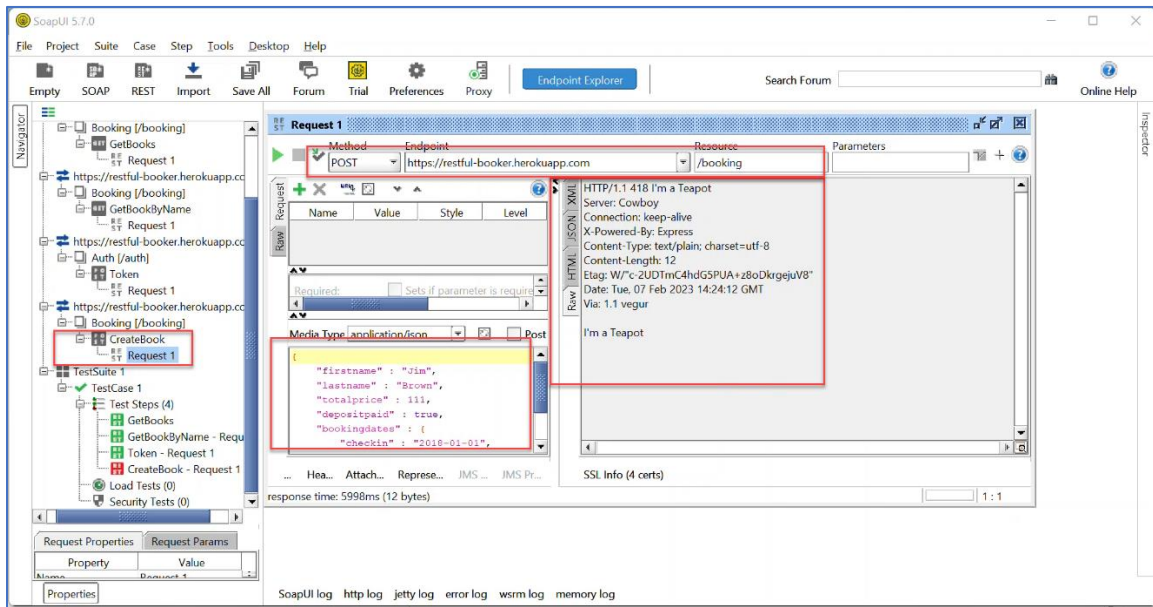


Figura 20. SOAP UI: CreateBook

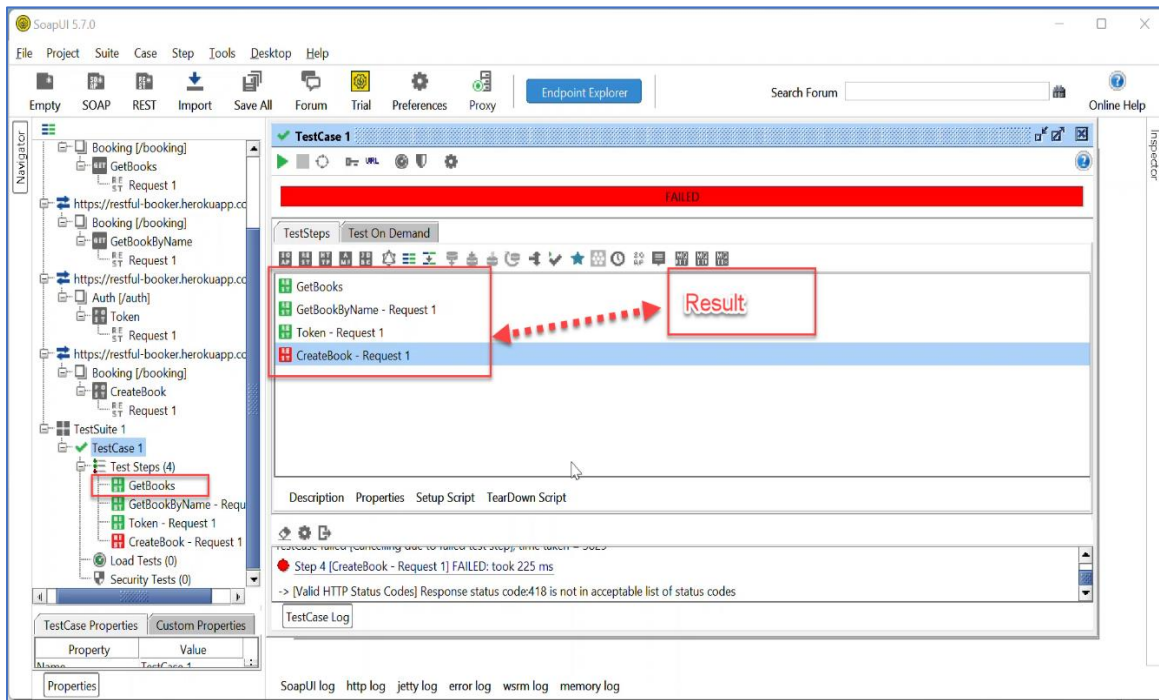


Figura 21. SOAP UI: Resultados



## 5. CONCLUSIONES

Concluido la fase de desarrollo y termino del proyecto se concluye lo siguiente:

- La fase de desarrollo de casos de prueba fue concluyente, y permitió realizar el control y seguimiento adecuado de los escenarios que se estaban abordando en la API.
- Respecto a las herramientas usadas tanto para la gestión de pruebas y para realizar las mismas, fueron amigables, intuitivas y ágiles en el proceso de desarrollo, se recomienda seguir con las herramientas para realizar el respectivo seguimiento para asegurar la calidad de sus servicios.
- Cabe mencionar que las herramientas usadas no requieren de exigentes requisitos de hardware para utilizarlos, sin embargo, para Testlink, si es un poco mas complicado porque requiere una versión de Xampp en específico para instalar y ejecutar la herramienta, pero es conveniente porque es muy útil.

Respecto a la API se evidencio lo siguiente:

- La API, es interesante, pero esta cargado con muchos errores, de validación en los campos.
- Es sensible a mayúsculas y minúsculas en sus búsquedas.
- Permite la creación de reservas, aunque los campos estén vacíos.
- No valida las fechas.
- Es excelente para hacer pruebas, es amigable la página y detalla los pasos a seguir en cada función que ofrece.

Respecto a las herramientas utilizadas tanto como Postman y Soap UI durante el desarrollo de pruebas a la API, mismas que se ejecutaron paralelamente arrojaron resultados distintos los cuales se aprecian en las Figuras 22 y Figuras 23.

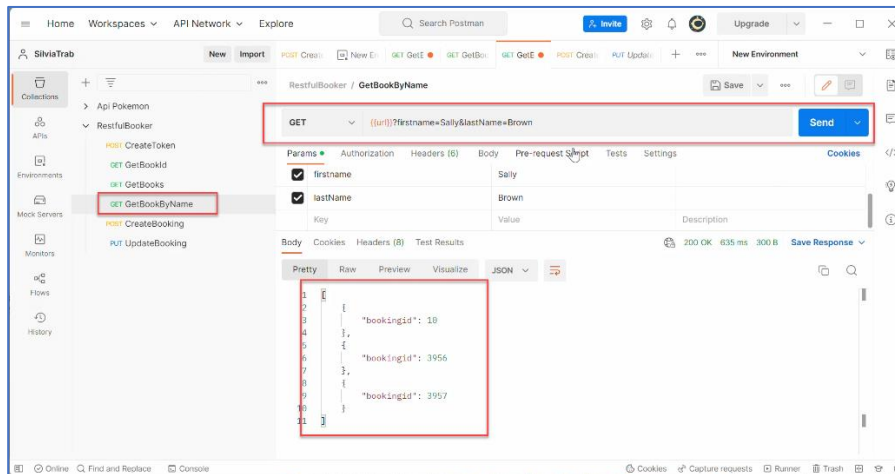


Figura 22. POSTMAN: Resultados GetByName

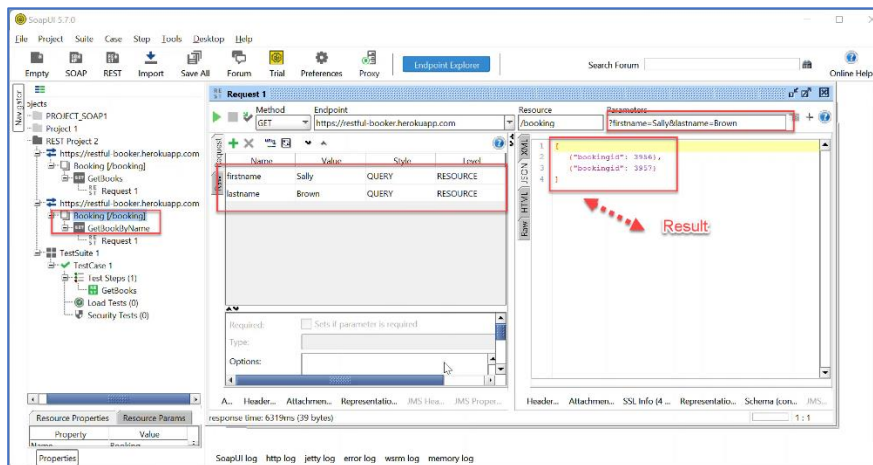


Figura 23. SOAP UI: Resultados GetByName

También arrojaron resultados distintos en la Figura 24, Figura 25

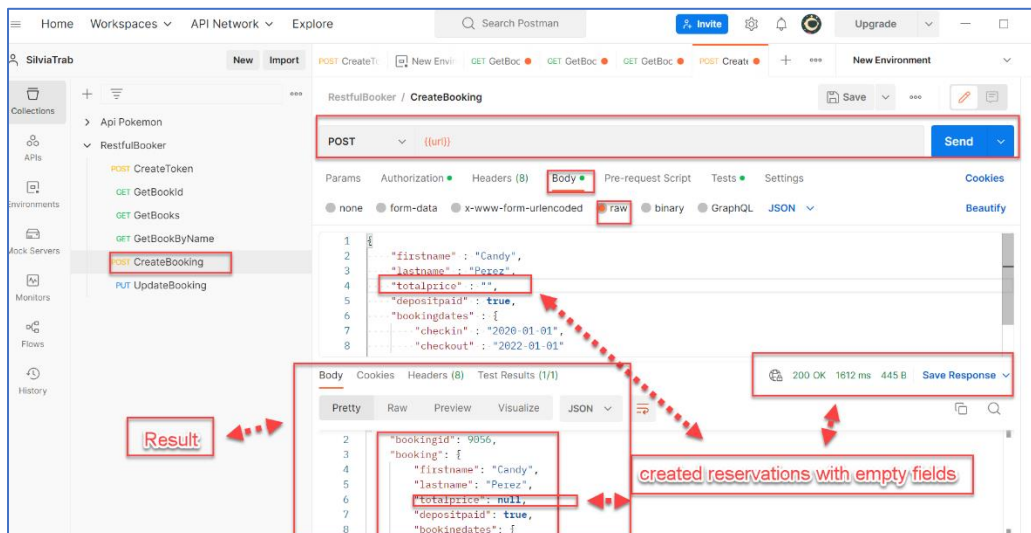


Figura 24. POSTMAN: Resultados CreateBook

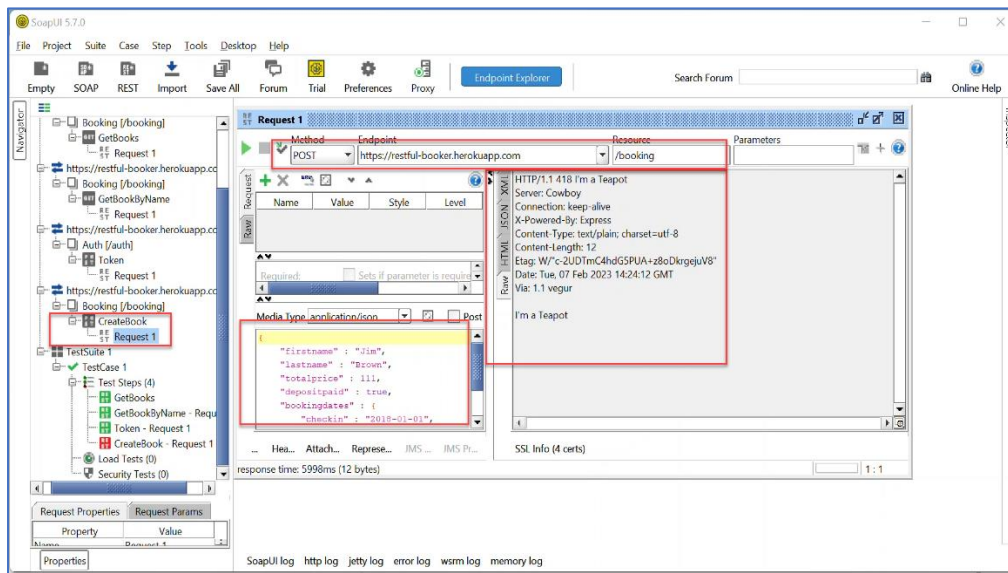


Figura 25. SOAP UI: ResultadosCreateBook

## 6. RECOMENDACIONES

En las próximas versiones se recomienda:

- Aumentar el conjunto de casos de prueba, con el objetivo de cubrir mas escenarios y de esta forma asegurar sus servicios de manera aceptable.
- Para la gestión de pruebas es recomendable usar Testlink, ya que es fácil y sencilla de usar
- Para realizar las pruebas se recomienda usar Postman ya que es una herramienta amigable, intuitiva, sencilla, se integra con otras herramientas y es muy usada por la comunidad. Sin embargo SOAP UI, también es otra herramienta potente para testing, pero al tener muchas mas funciones implementadas, es un poco mas complicado en su uso, sin mencionar las muchas ventanas que se solapan una tras otra eventualmente, en ese entendido, solo queda decir “para gustos colores”.

En conclusión, dependerá de el equipo si usan ambas o optan por alguna de las ya mencionadas.