Evaluación Sumativa 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ÁREA ACADÉMICA** | | **Informática y Telecomunicaciones** | | | | | |
| **ASIGNATURA** | Redes Avanzadas I - 2024 | | | | | **CÓDIGO** | OMRA05 |
| **SEDE** | Santiago Centro - Valparaíso | | | **DOCENTE** | Horacio Vega Fuentes | | |
| **Unidad de Aprendizaje** | | | N°1 | **Criterios a Evaluar** | | 2.1.1  2.1.2  2.1.3  2.1.4  2.1.5 | |
| **DURACIÓN** | 90 minutos | | | **FECHA** | | 03-05-2023 | |

**Nombre: Daniel Quezada**

**Nombre: Yamil Leiva**

|  |
| --- |
| **INSTRUCCIONES GENERALES:**   1. **La nota 4.0 se obtiene logrando un 60% del puntaje total. Ponderación de un 30%** |

**Indicaciones generales:**

1. Formar equipos de dos Alumnos.
2. Debe trabajar en el entorno de la VM DEVASC.
3. La VM **DEVASC\_base 2024** cuenta con Visual Studio y Python, la cual ya está actualizada.
4. Cuenta de GitHub activa y Token de acceso activado.
5. Los resultados deben ser compartidos por un enlace de su cuenta GitHub.
6. Para cada uno de los ítems, debe pegar la pantalla de lo realizado. Las imágenes deben se claras y debe mostrar lo realizado en la configuración. Para ello recorte solo lo necesario, para que se visualice de manera clara.

**Requerimientos**

1. **Biblioteca Escolar. Evaluación Grupal Presencial en Clases. (40 Puntos)**

Usar la máquina virtual DEVASC, conectarse a la Biblioteca Escolar. **Utilice Postman** para realizar llamadas API al simulador de API, para lo cual utilice la Guía de laboratorio.

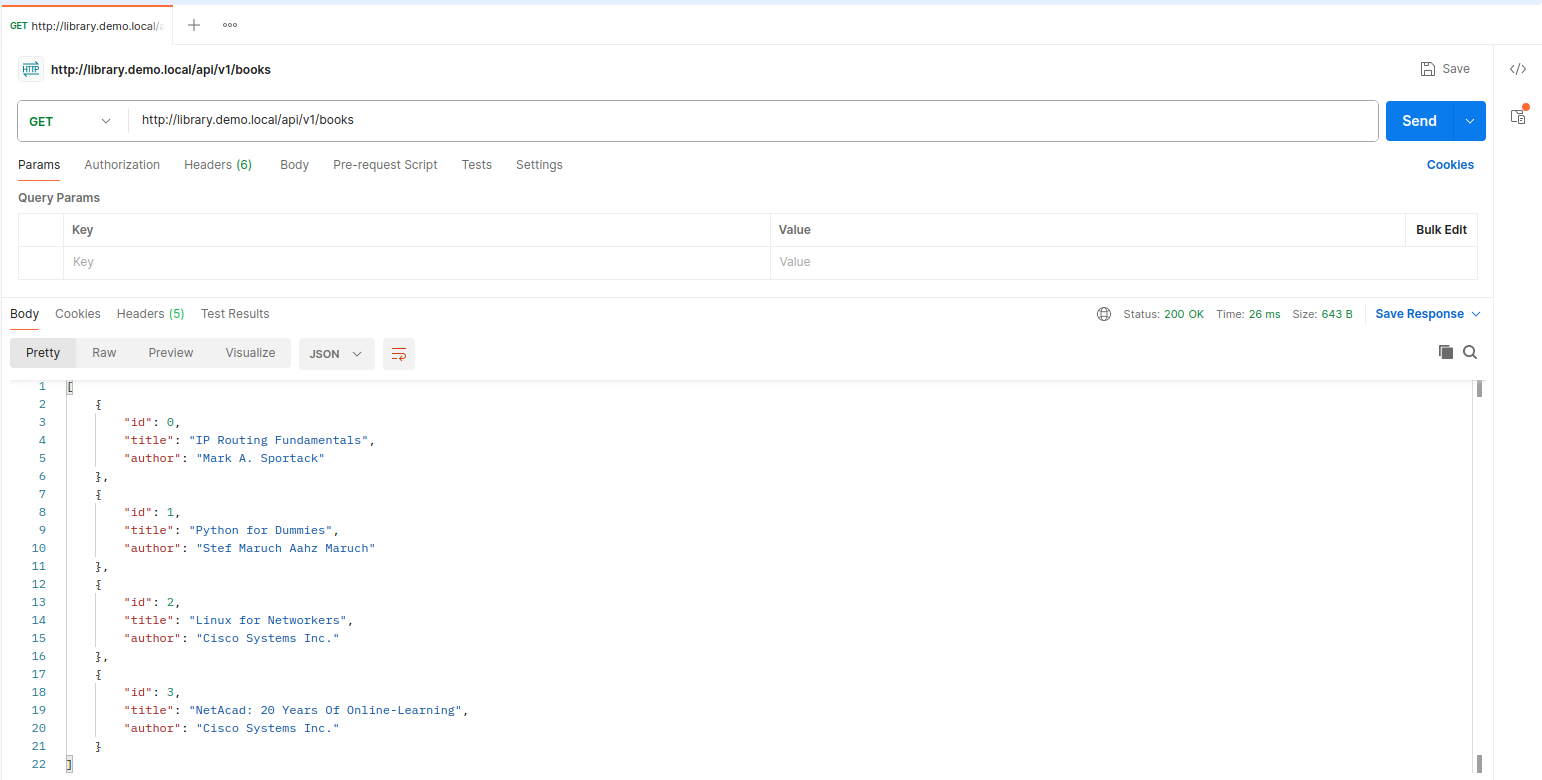
**Como evidencia de la actividad realice una captura de pantalla de cada uno de los siguientes pasos.**

### Paso 1: Abra Postman.

Abrir la Biblioteca de su VM en el explorador Chromium. Luego Haga doble clic en el icono de Postman en el escritorio, iniciaría sesión en Postman.

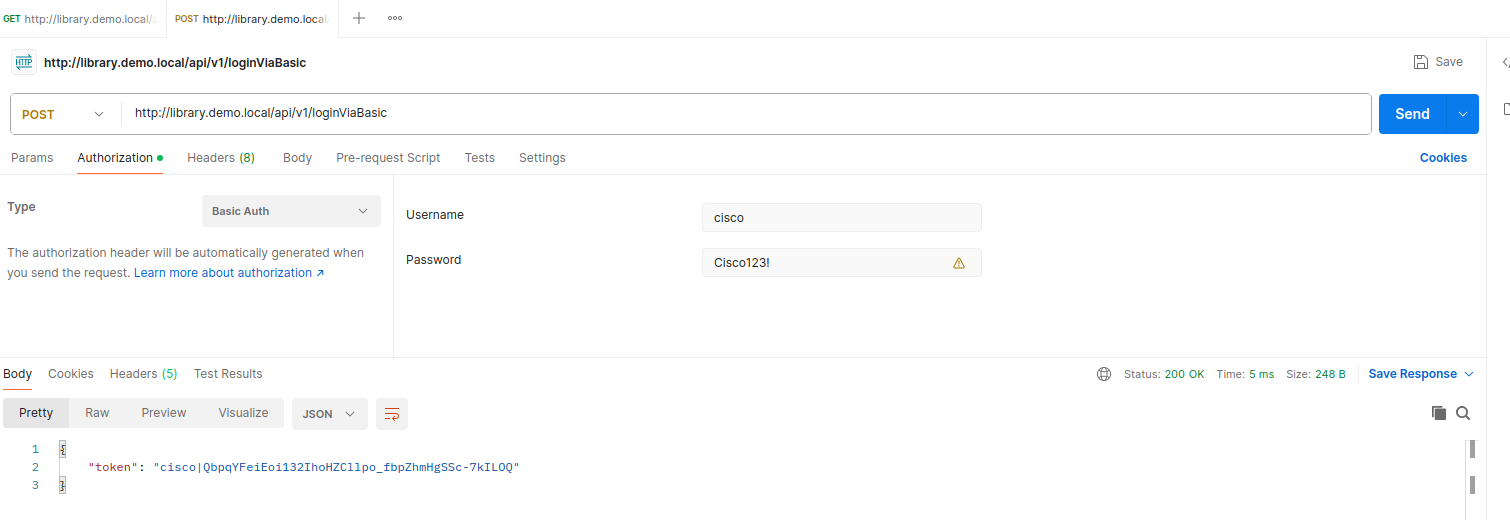
**Paso 2: Enumere los libros utilizando la API GET /books. Realizar esta actividad con Postman.**

En la ventana principal junto a la pestaña **Launchpad**, haga clic en el icono más “**+**” para crear una solicitud sin título. Realice los pasos para obtener el la Capture la pantalla de Postman, en donde se vean los libros enumerados, el código 200 ok en verde**. (5 Puntos)**



**Paso 3 Obtenga un token usando la API POST /LoginViaBasic. Realizar esta actividad con Postman.**

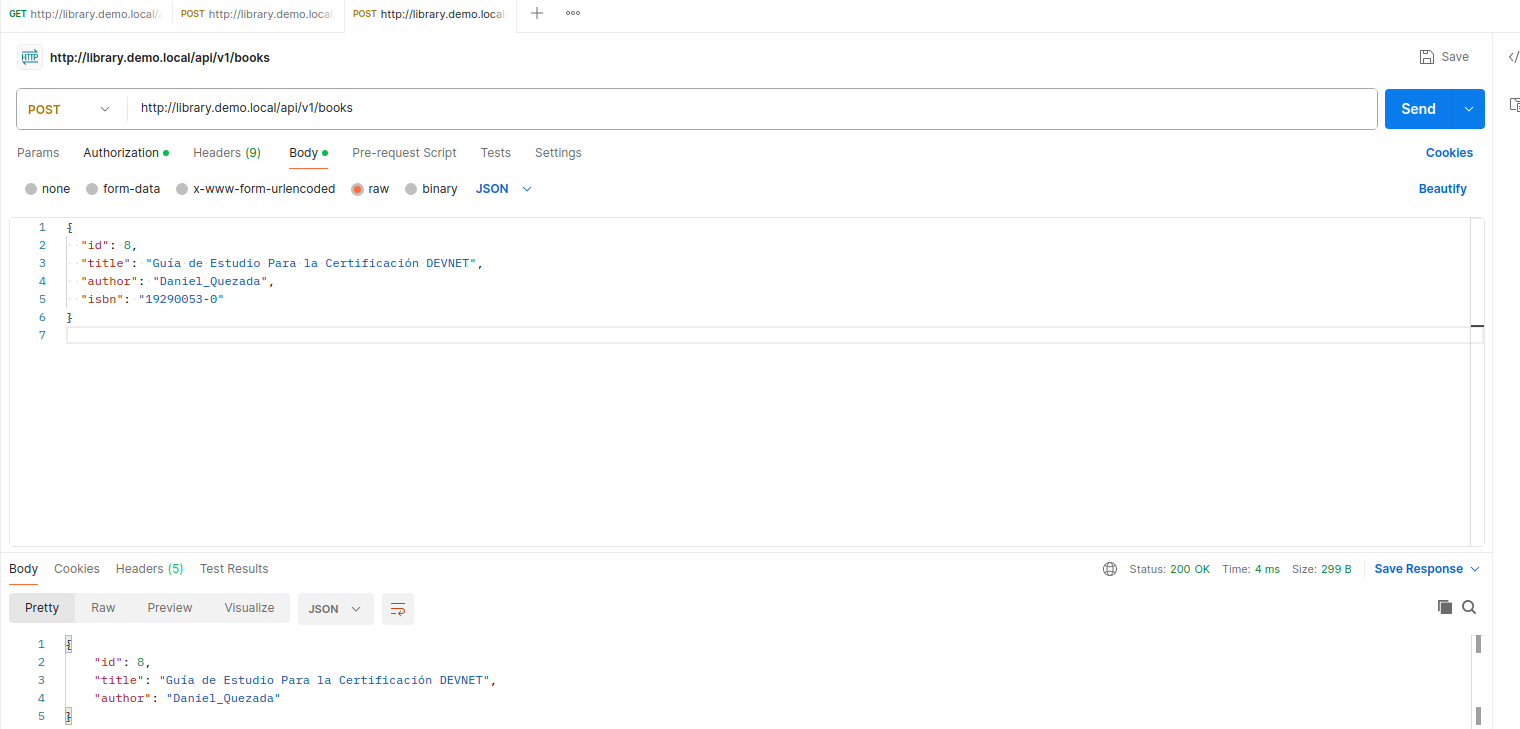
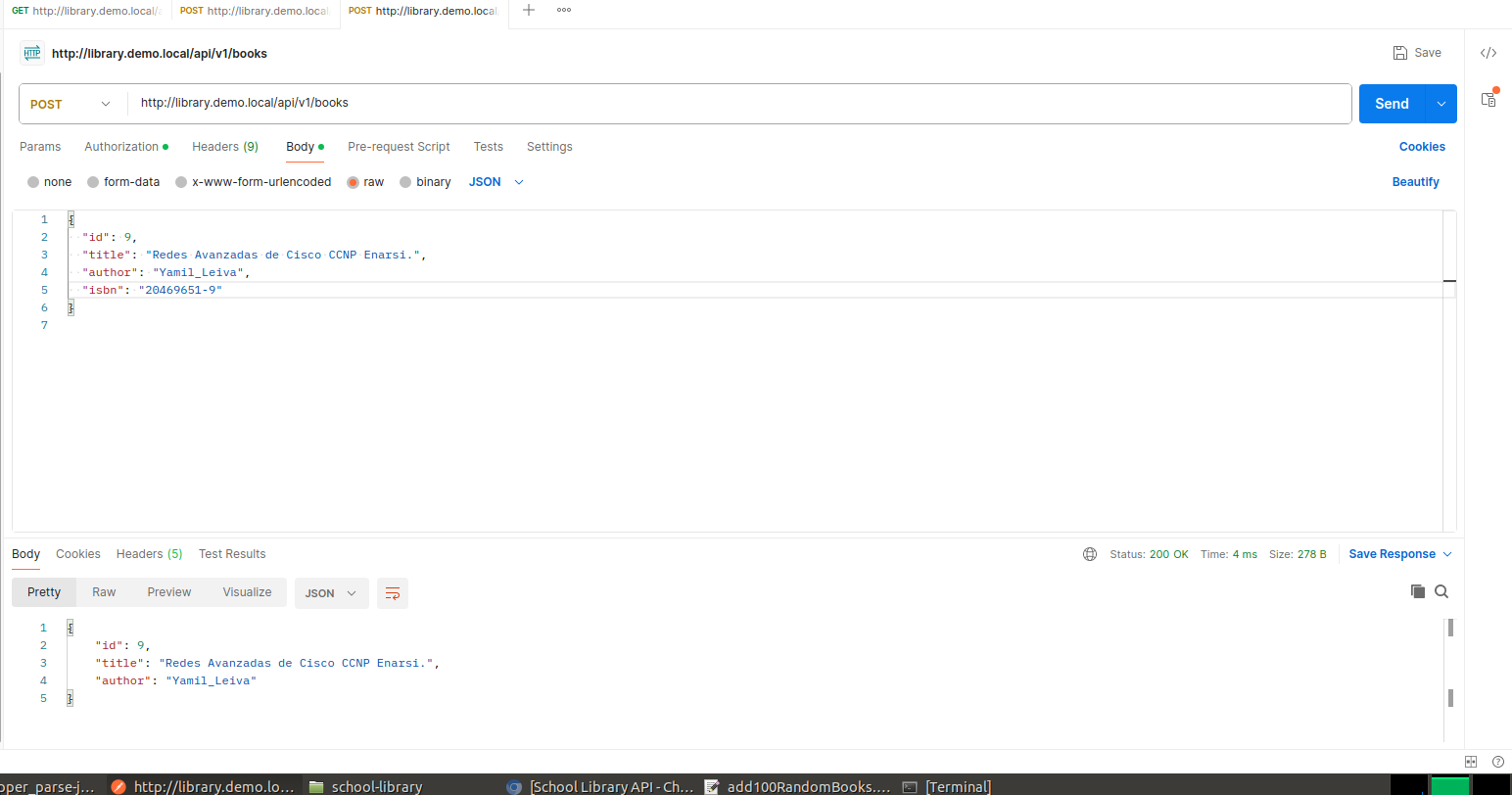
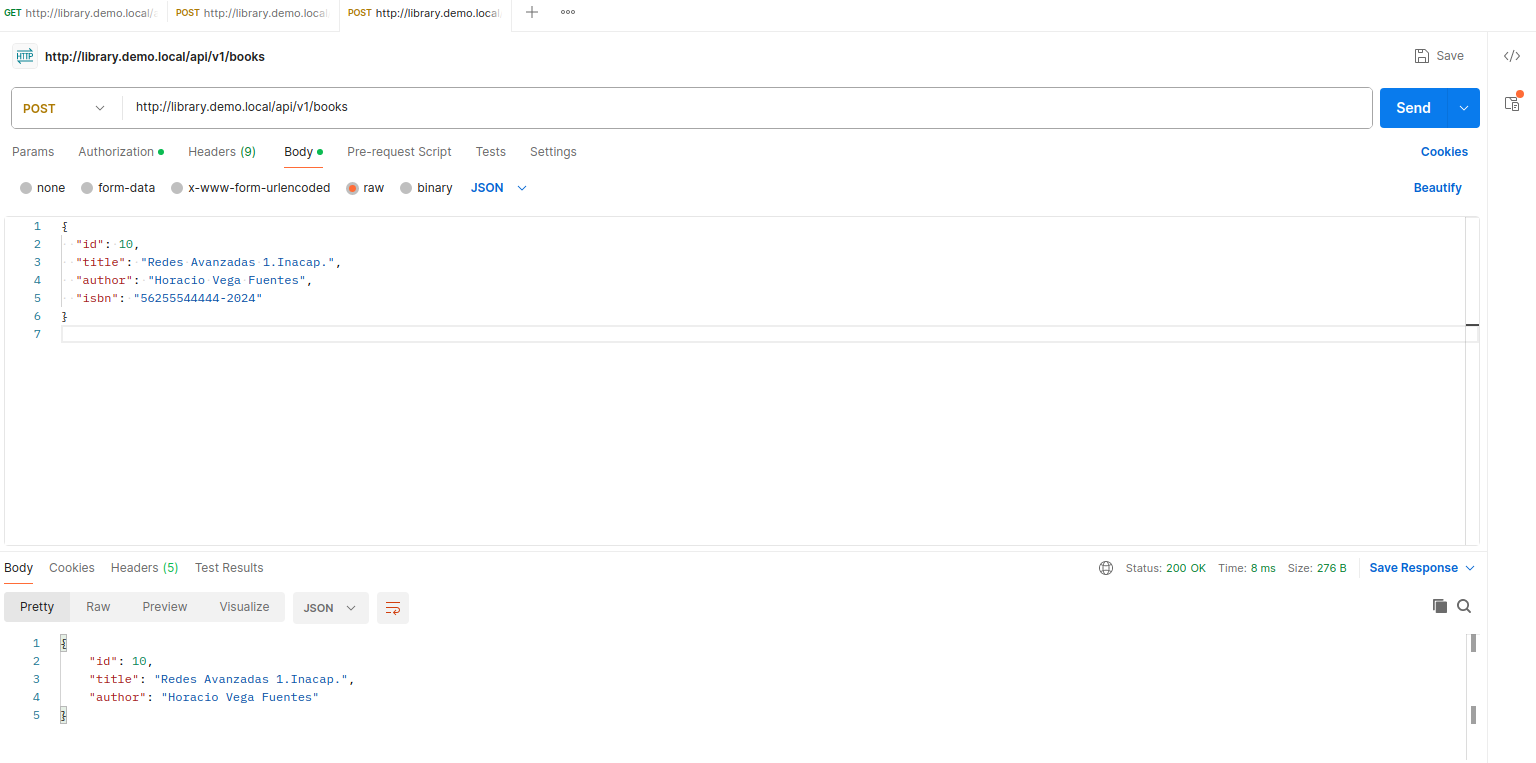
En la ventana principal de Postman, haga clic en el icono más "**+**" para crear una nueva **solicitud sin título**. Debe realizar los pasos para la autorización para obtener un nuevo Token. Realice los pasos para obtener la Captura de pantalla de Postman con su resultado. **(5 Puntos)**



**Paso 4: Añadir tres libros mediante la API POST /books. Realizar esta actividad con Postman.**

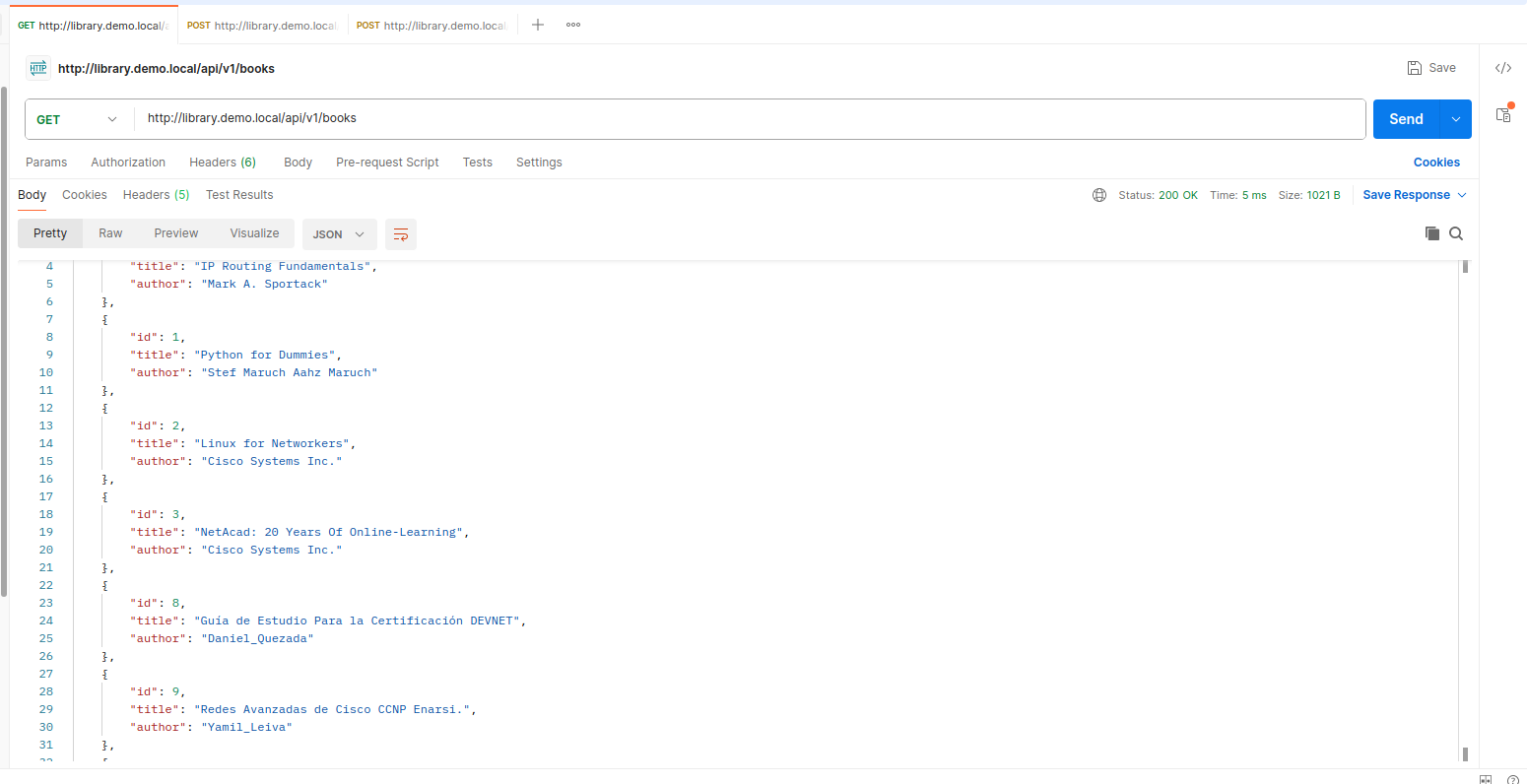
Realice los pasos para obtener la Captura de pantalla de Postman con su resultado, en donde debe agregar tres libros a su biblioteca. **(15 Puntos)**

**Los libros que debe agregar son:**

1. **id 8**
2. **título: Guía de Estudio Para la Certificación DEVNET**
3. **autor: Su-Nombre\_Su-Apellido del Alumno1**
4. **isbn: Su RUT (Sin puntos y con Guion)**
5. **id 9**
6. **título: Redes Avanzadas de Cisco CCNP Enarsi.**
7. **autor: Su-Nombre\_Su-Apellido del Alumno2**
8. **isbn: Su RUT (Sin puntos y con Guion)**
9. **id 10**
10. **título: Redes Avanzadas 1.Inacap.**
11. **autor: Horacio Vega Fuentes**
12. **isbn: 56255544444-2024**
13. 
14. 
15. 

### Paso 5: Verifique los libros adicionales con la API Get /books. Realizar esta actividad con Postman. (5 Puntos)

* + - 1. Vuelva a la primera pestaña GET. Como puede ver, Postman hace que sea fácil cambiar entre diferentes llamadas de API. Haga clic en **Send** (Enviar).
      2. Para verificar que la solicitud de API fue correcta, ahora usted verá una respuesta que incluye el código de **estado 200 OK** en verde.
      3. Haga clic en **Cuerpo** para ver la respuesta. Observe que el predeterminado es **Pretty** y **JSON**. **(5 Puntos)**

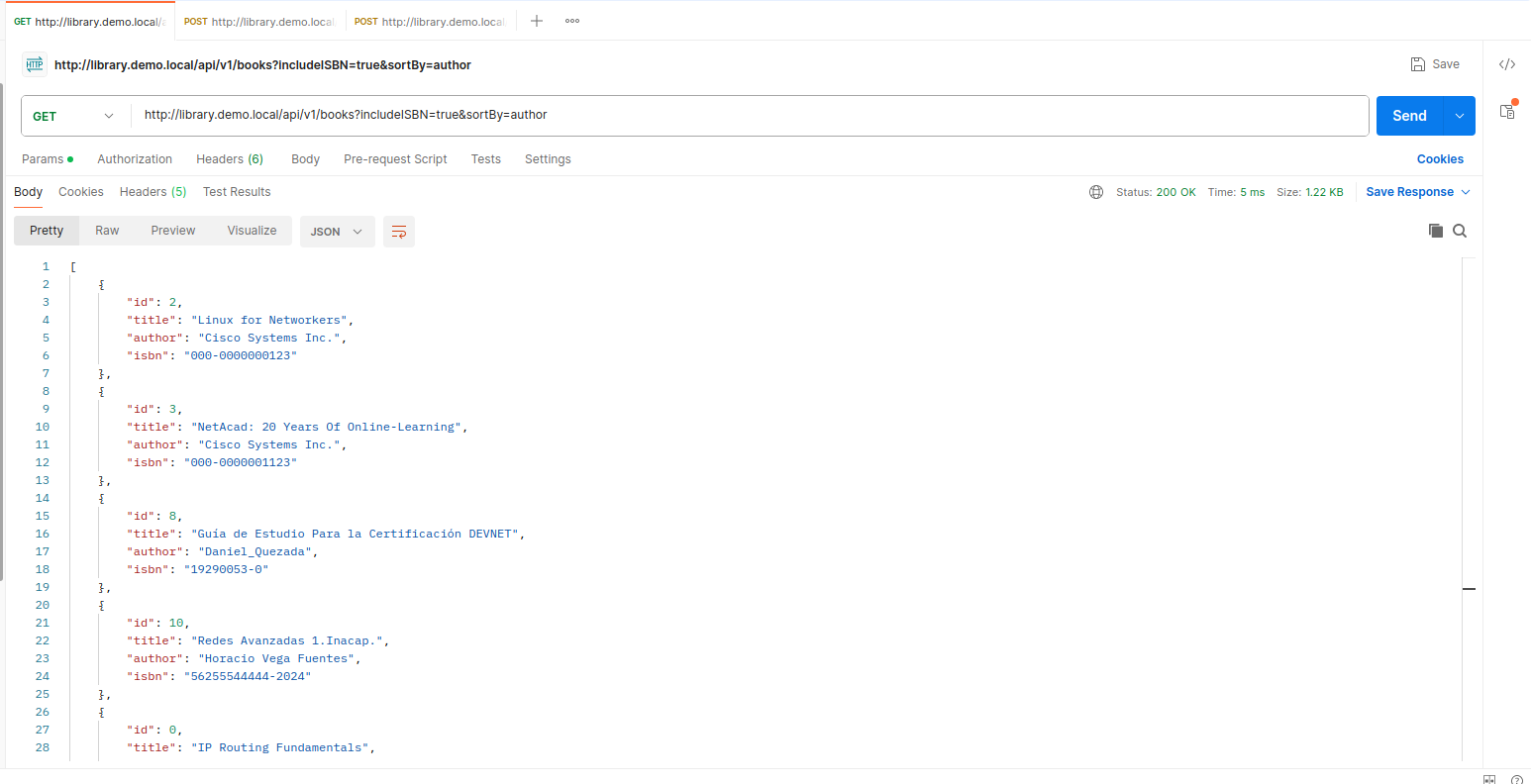
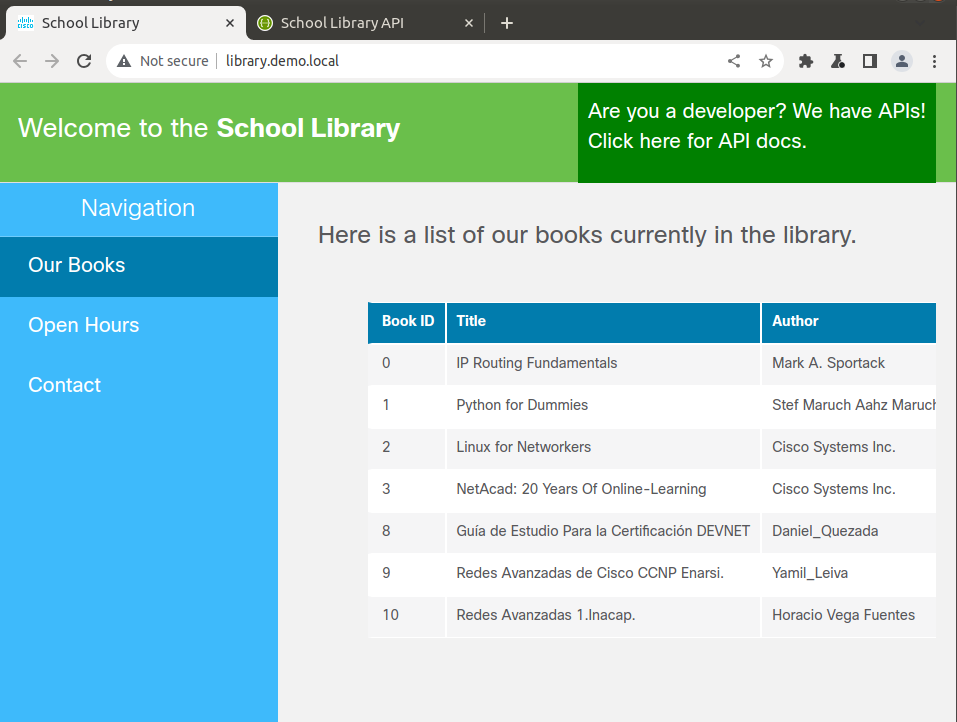


### Paso 6: Utilice parámetros adicionales con el API Get /books con Postman.

* + - 1. **En Postman y vaya a la primera pestaña API GET** que tiene el siguiente enlace ***http://library.demo.local/api/v1/books***. Ahora incluirá algunos de los parámetros del sitio web de Librería Escolar de API.
      2. Haga clic en **parámetros**. Verá en los cuadros de entrada de **parámetros de consulta** para **KEY** y **VALUE**. Ingrese la siguiente información:
* En **KEY**, escriba **includeISBN** y en **Valor** ingrese **Cierto**

Observe que una marca de verificación se incluirá automáticamente a la izquierda del valor y se agregará una nueva fila.

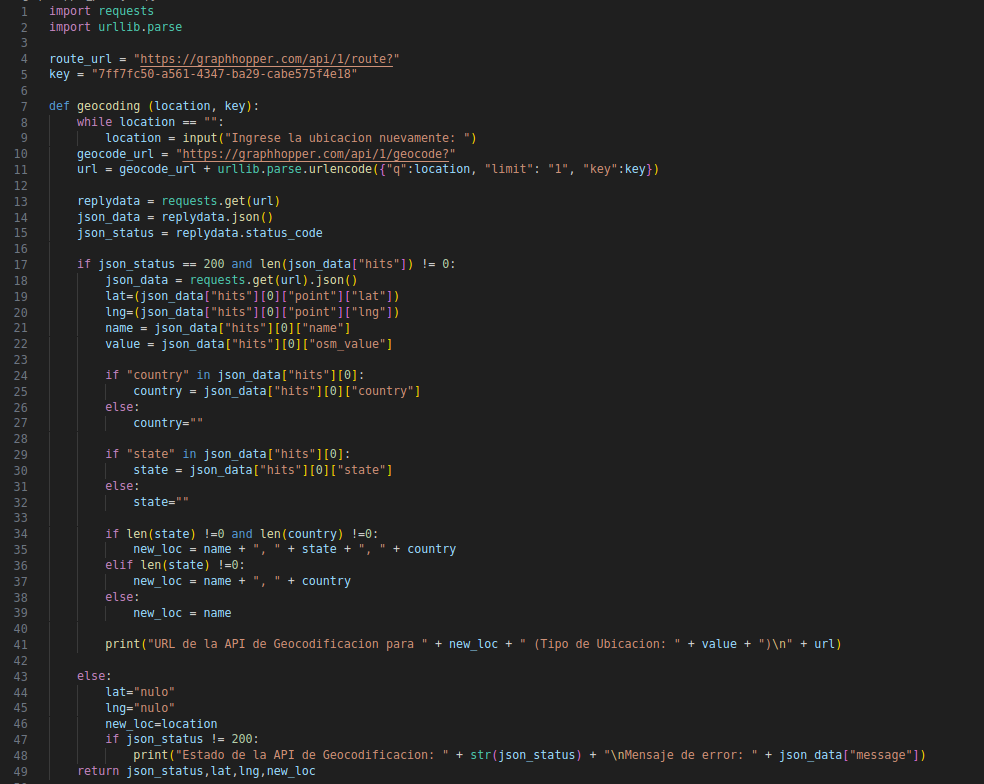
* En **KEY**, escriba **sortBy** y en **Valor** escriba **autor**

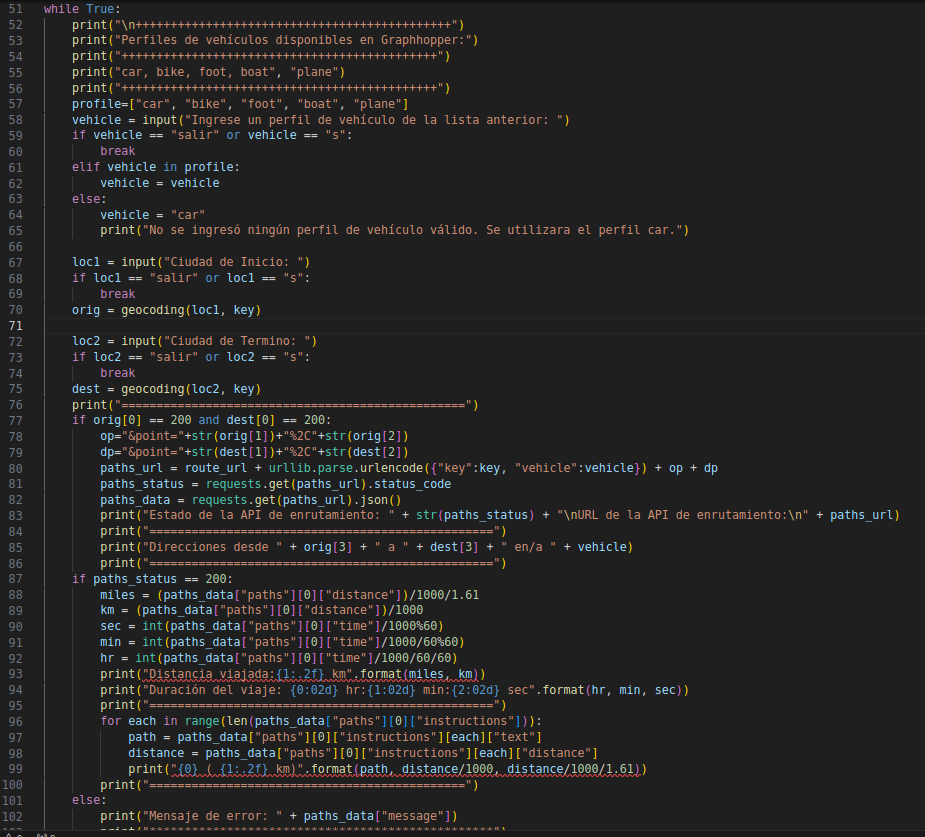
1. Haga clic en **Ejecutar**.
   * + 1. En el **cuerpo de la respuesta** verá la lista de libros ahora ordenados por autor e incluyendo los ISBN.
       2. **Realice una captura de pantalla con su resultado desde Postman.** además debe **obtener una captura de pantalla de la biblioteca escolar** donde muestre que los tres libros que aparecen agregados y ordenados. **(5 Puntos)**
2. Subir el documento en Word creado en un repositorio público en GitHub con el nombre de “**Evaluación2-ParteA-Su\_Nombre**” y en la descripción su Nombre y Apellido. Debe incluir un commit con el nombre de “**Consumo de API en Postman”. (5 Puntos)**
3. 
4. 
5. **Además comparte el archivo en Word en al ambiente de aprendizaje en el link disponible.**
6. **Luego Copie el link generado en GitHub y pegue en este documento. Para que así el docente pueda Evaluarlo.**
7. **https://github.com/YLeivaS/Evaluacion2-ParteA-Yamil\_Leiva**
8. **Consumo de API Pública. Evaluación Grupal Presencial en Clases. (60 Puntos)**

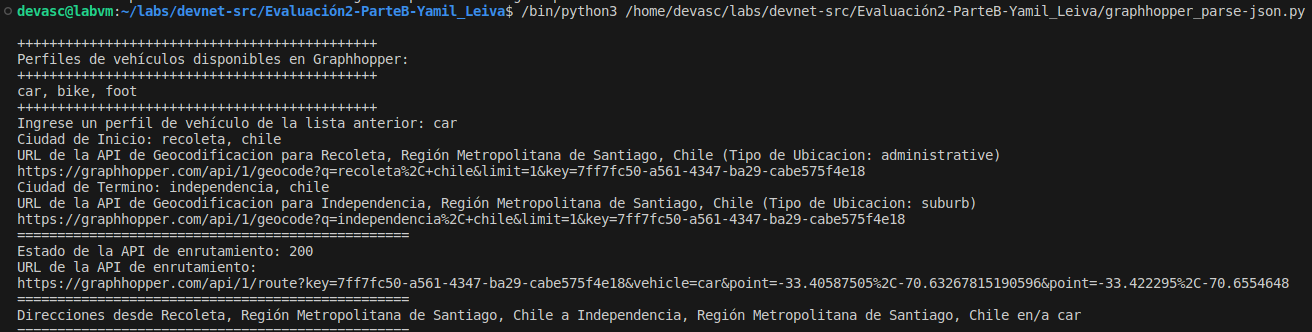
**Utilizando el sitio de Graphhopper Directions y el token generado en laboratorio respectivo, deberá crear un código en Visual Studio Code de la máquina virtual DEVASC, donde el programa realice lo siguiente:**

**Como evidencia de la actividad realice una captura de pantalla de cada uno de los siguientes pasos.**

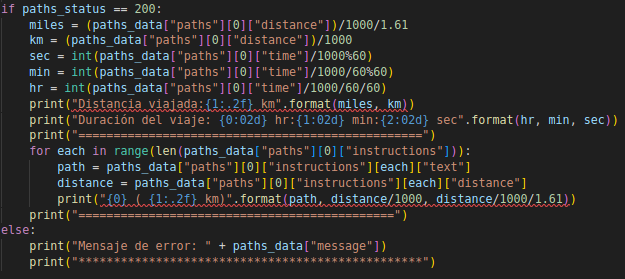
* Traducir al español los textos de todo el programa. No debe incluir las variables **(5 puntos)**





* Solicitar “Ciudad de Inicio” y “Ciudad de Termino” **(3 puntos)** 
* Mostrar la duración del viaje en horas, minutos y segundos **(4 puntos)**

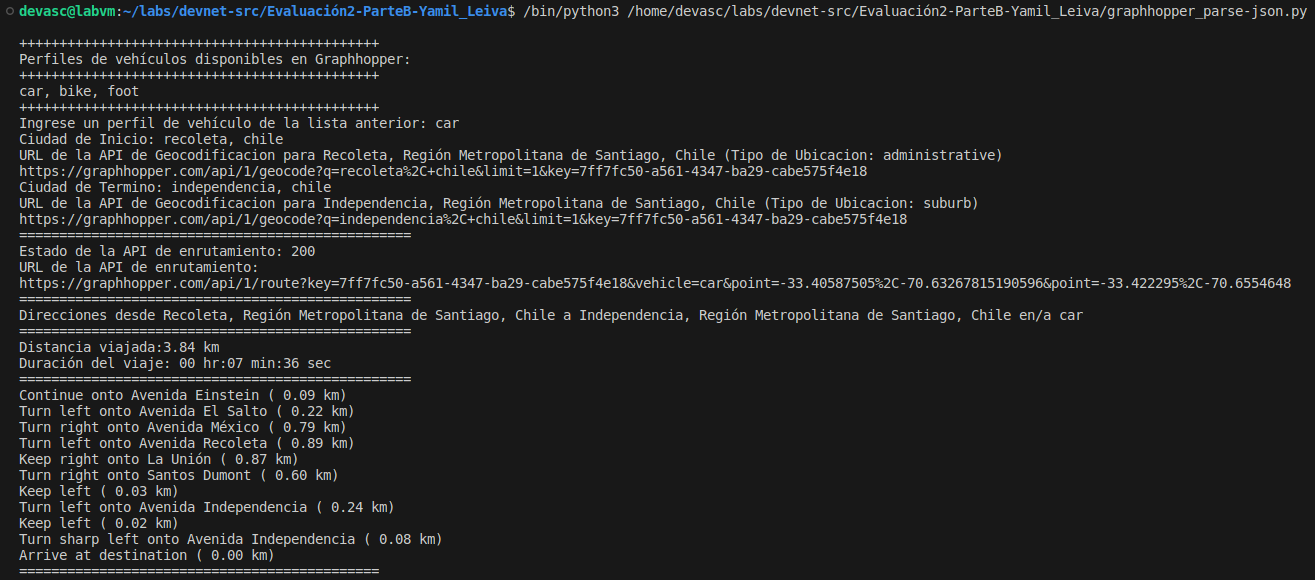


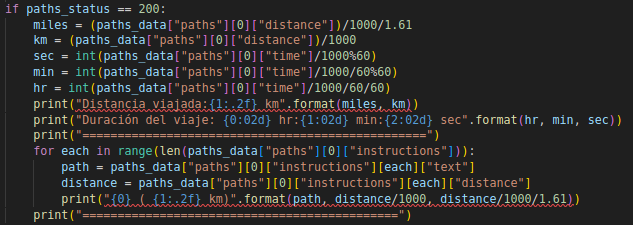


* Medir la distancia solo en kilómetros entre dos comunas distintas de la Región Metropolitana. **(5 puntos)**

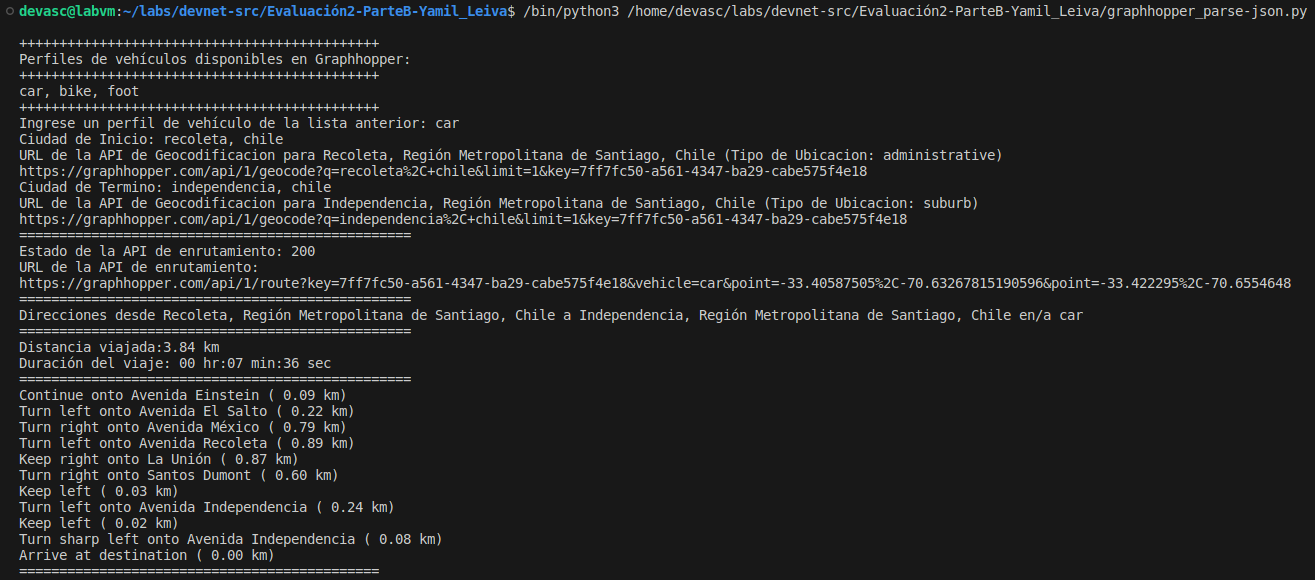


* Todos los valores deben utilizar dos decimales. **(5 puntos)**

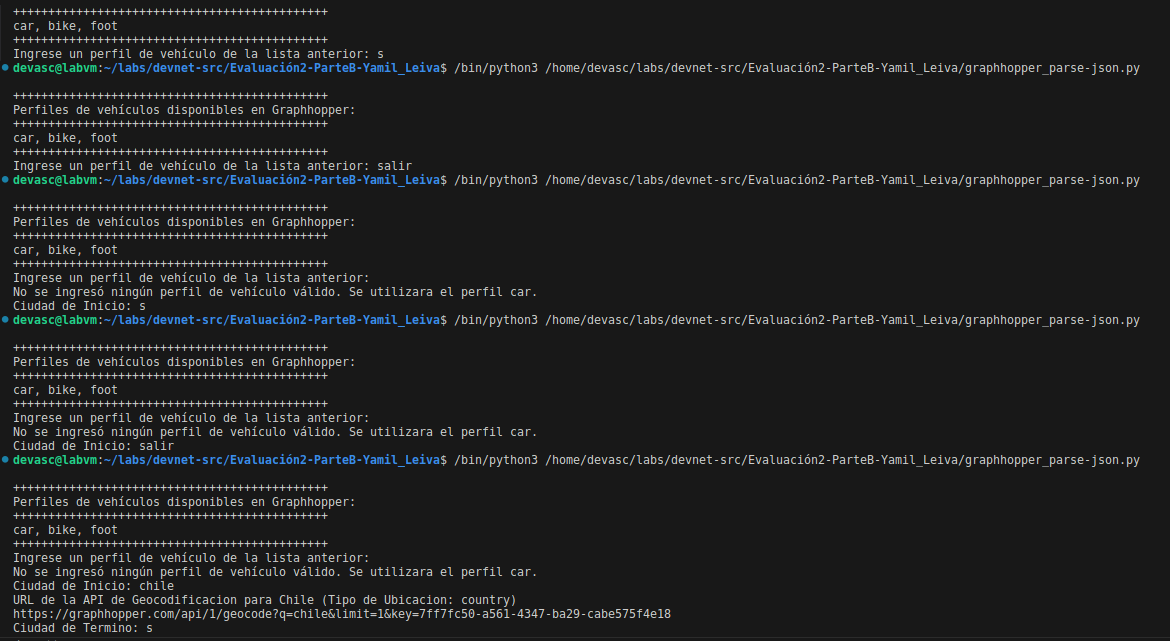


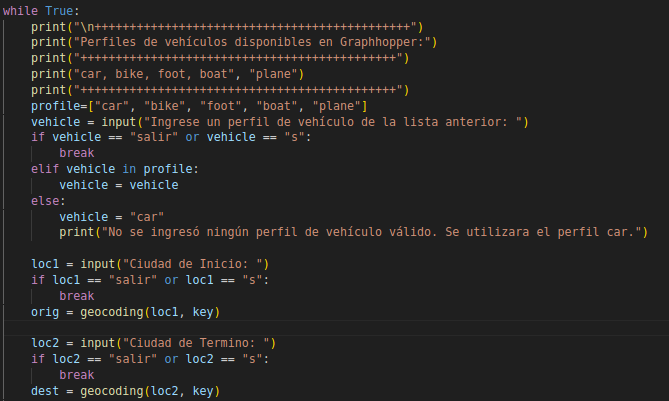


* Debe imprimir la narrativa del viaje. **(5 puntos)**

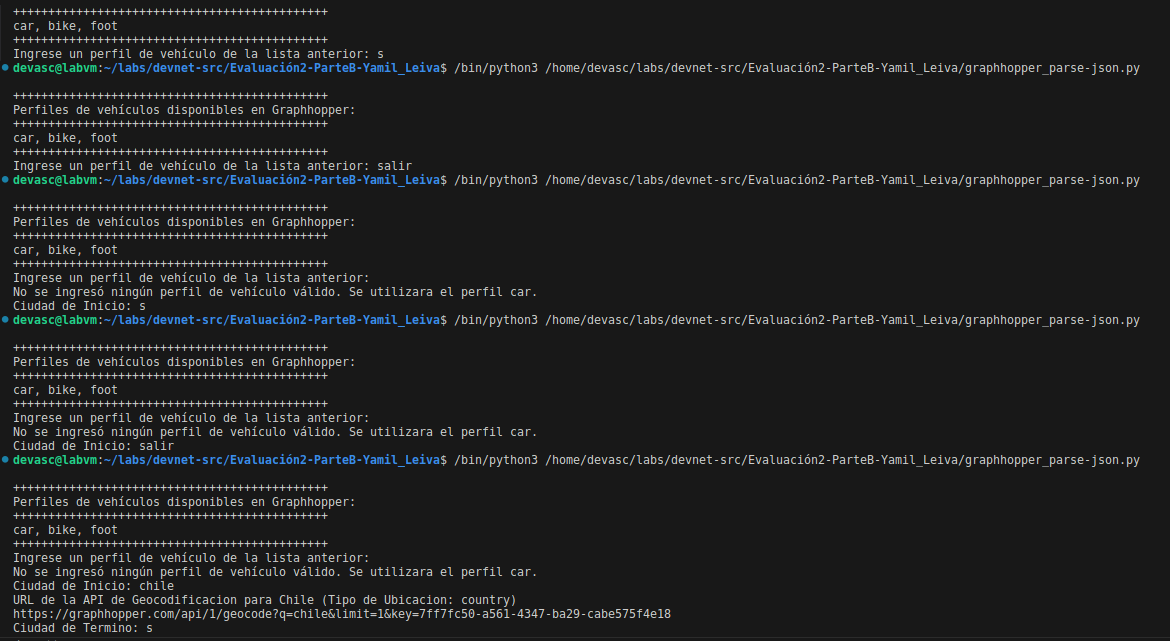


* Agregar una salida del programa con la letra “**s**” y con “**salir**” **(2 puntos)**

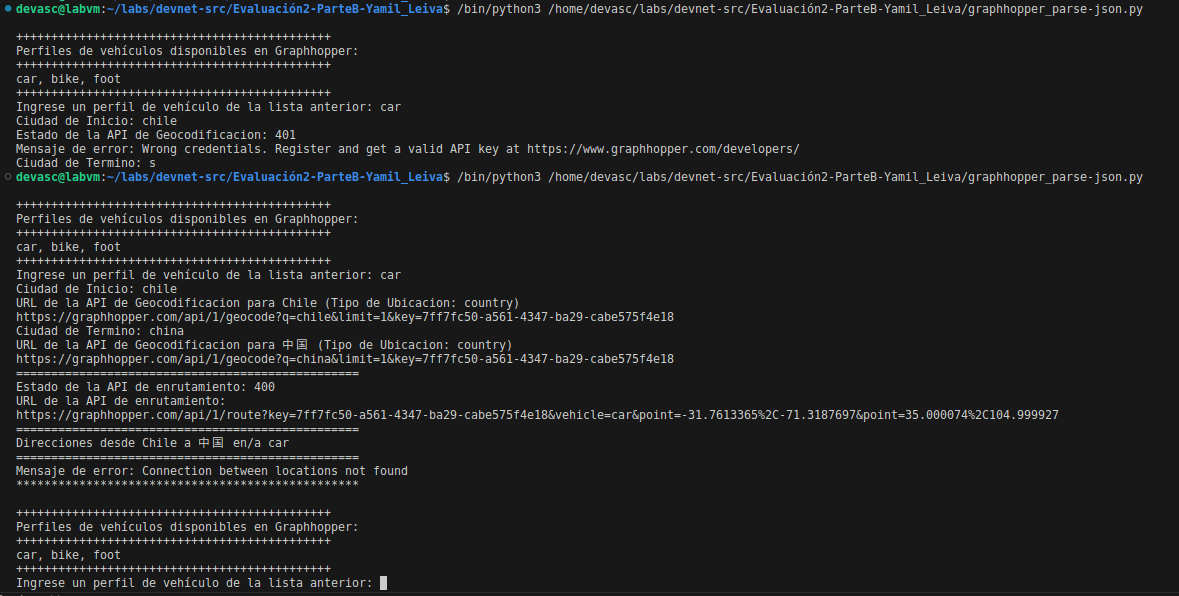




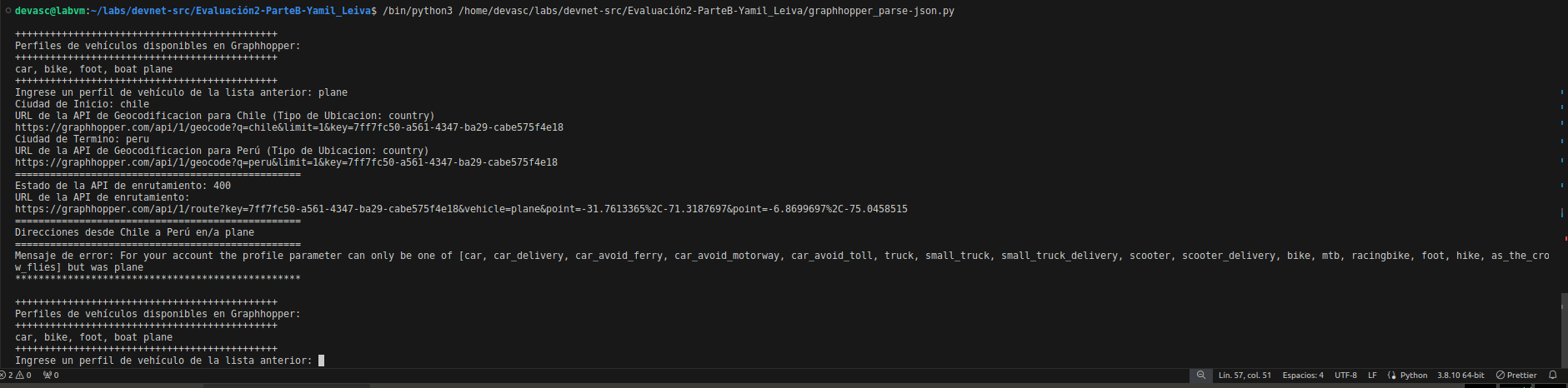
* Verificar si el funcionamiento es el solicitado. **(5 puntos)**



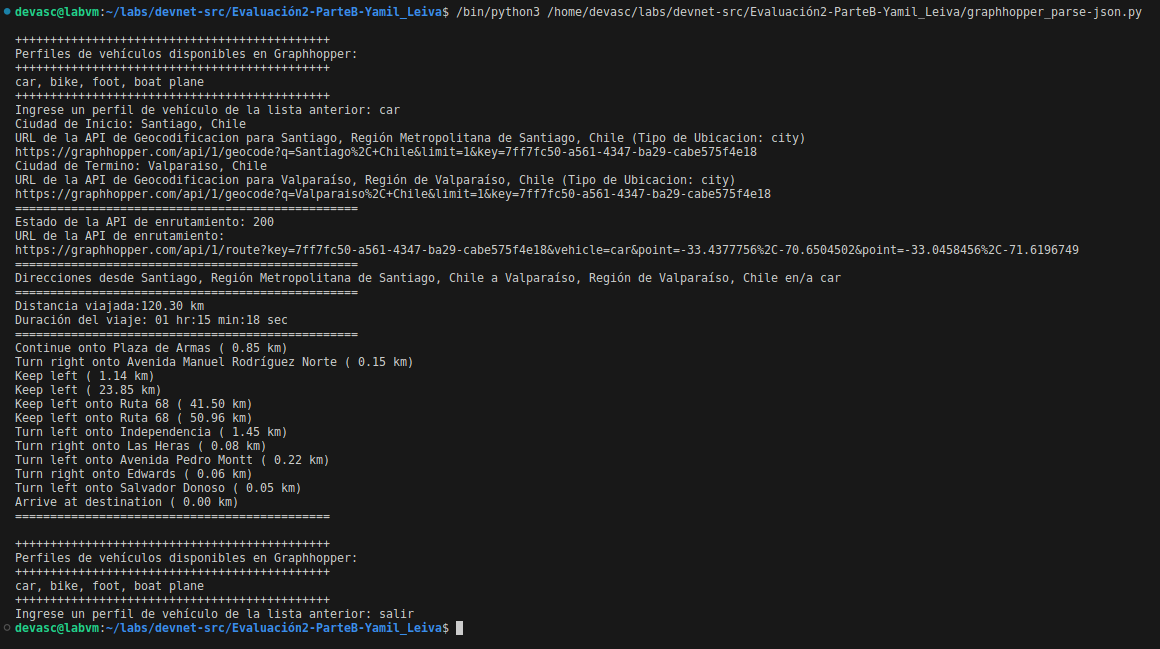
* Informar errores de la API de routing. **(5 puntos)**



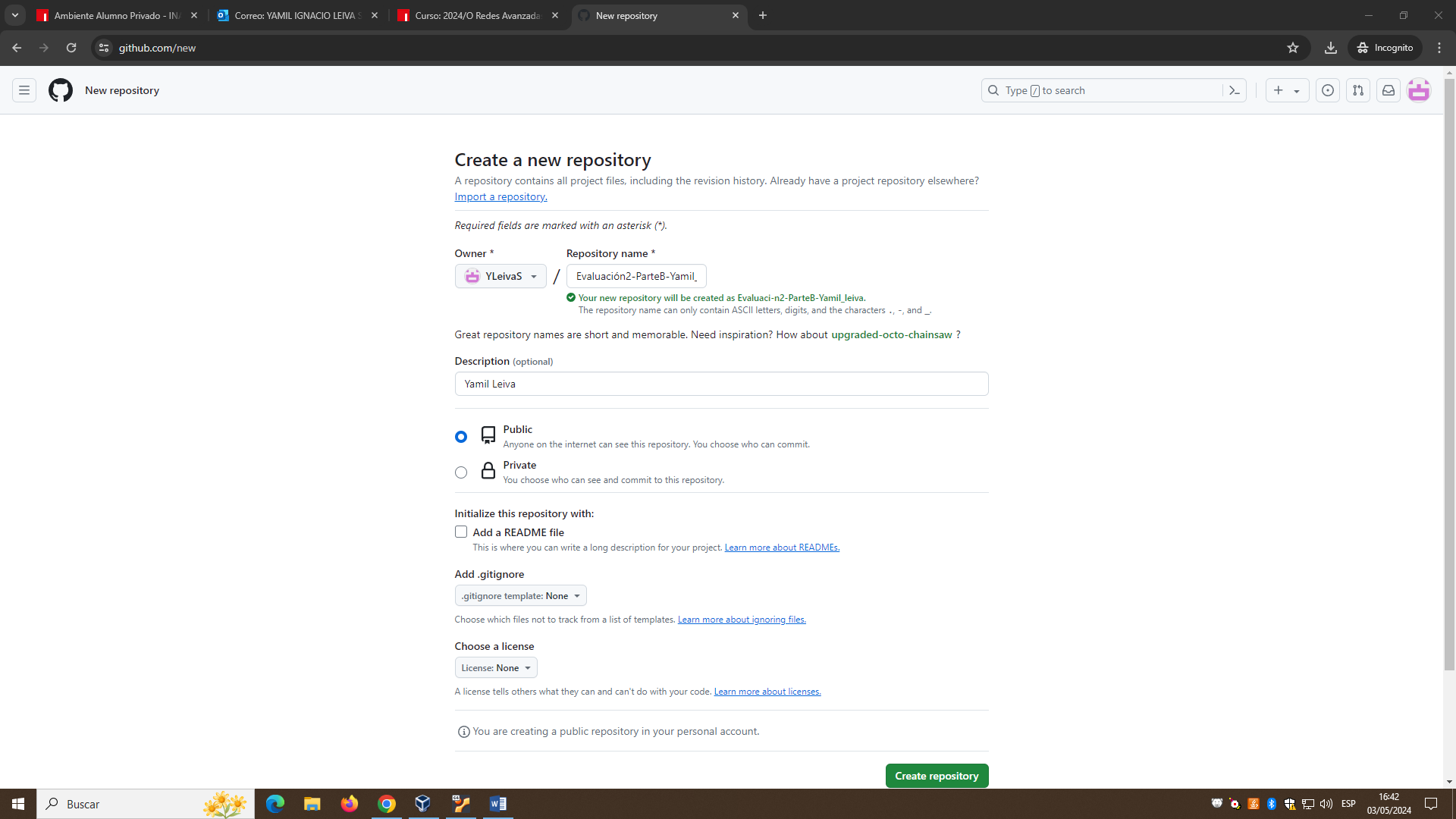
* Agregue más modos de transporte. **(5 puntos)**



* Pruebe la funcionalidad completa de la aplicación. **(5 puntos)**



* Subir el **script de Python** creado de **Graphhopper Directions** en un repositorio público en GitHub con el nombre de “**Evaluación2-ParteB-Su\_Nombre**” y en la descripción su Nombre y Apellido. Debe incluir un commit con el nombre de “**Consumo de API Pública de *Su\_Nombre*”. (10 puntos)**



1. **Además comparte el archivo en Word en al ambiente de aprendizaje en el link disponible.**
2. **Luego Copie el link generado en GitHub y pegue en este documento. Para que así el docente pueda Evaluarlo.**

**https://github.com/YLeivaS/Evaluacion2-ParteB-Yamil\_Leiva**