

Evaluación Sumativa 2

ÁREA ACADÉMICA		Informática y Telecomunicaciones			
ASIGNATURA	Redes Avanzadas I			CÓDIGO	OMRA05
SEDE	Santiago Centro	DOCENTE	Horacio Vega Fuentes		
Unidad Aprendizaje	de	N°1	Criterios a Evaluar	2.1.1 2.1.2 2.1.3 2.1.4 2.1.5	
DURACIÓN	120 minutos		FECHA	11-05-2023	

Nombre:

Nombre:

INSTRUCCIONES GENERALES:

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

Indicaciones generales:

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

Requerimientos

A. 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.

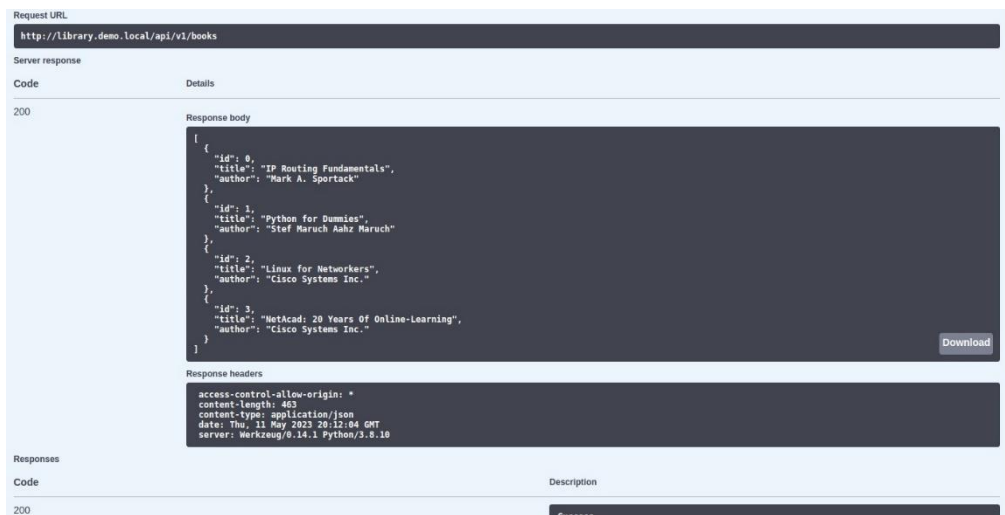
Como evidencia de la actividad realice una captura de pantalla de cada uno de los siguientes pasos.

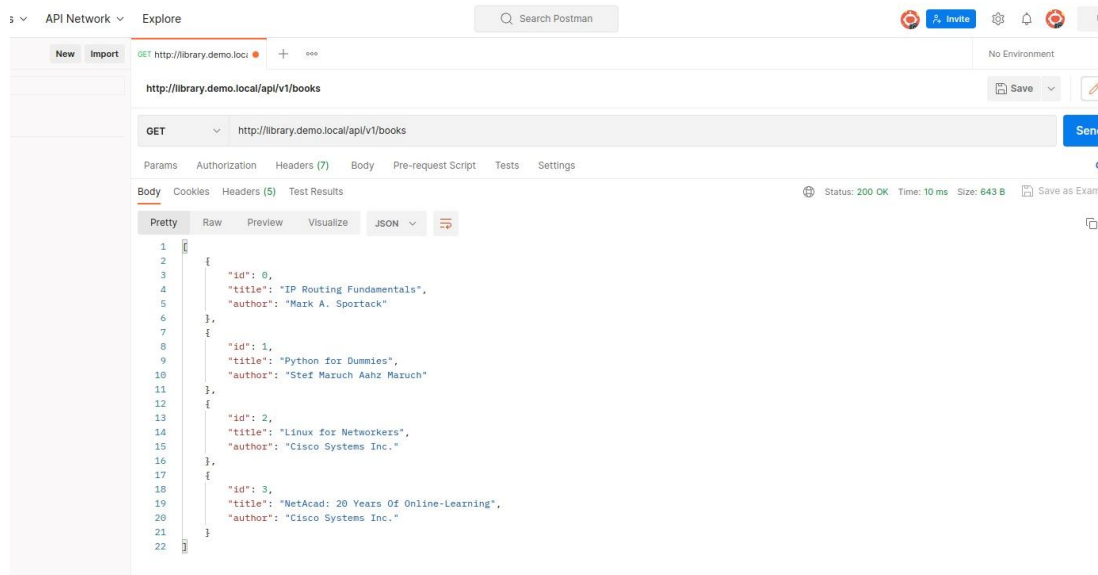
Paso 1: Abra Postman.

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.

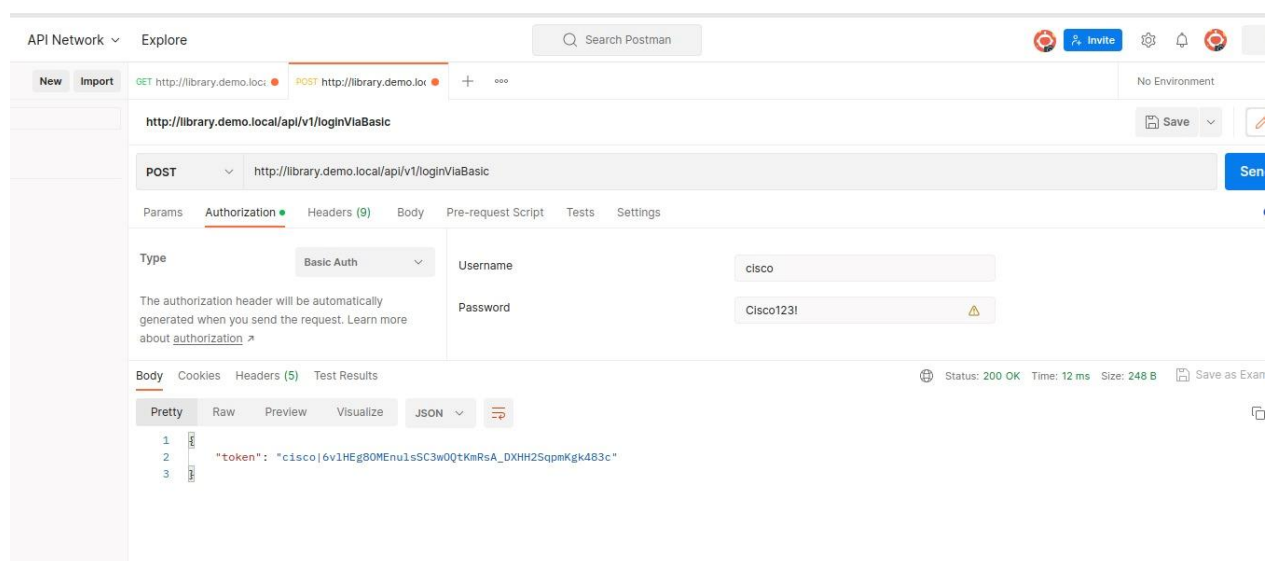
En la ventana principal junto a la pestaña **Launchpad**, haga clic en el icono más “+” para crear una solicitud sin título. Realice los pasos para obtener el la Capture la pantalla de Postman, en donde se vean los libros enumerados, el código 200 ok en verde. **(5 Puntos)**





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. **(10 Puntos)**

Los libros que debe agregar son:

id 5

título: Guía de Estudio Para la Certificación Ccna Security

autor: Su-Nombre_Su-Apellido

isbn: Su RUT (Sin puntos y Sin Guión)

id 6

título: Redes Cisco. CCNA Routing y Switching.

autor: Su-Nombre_Su-Apellido

isbn: Su RUT (Sin puntos y Sin Guión)

id 7

título: Redes Avanzadas. CCNA Devnet.

autor: Su-Nombre_Su-Apellido

isbn: Su RUT (Sin puntos y Sin Guión)

GET http://library.demo.local POST http://library.demo.local POST http://library.demo.local + ... No Environment

http://library.demo.local/api/v1/books Save

POST http://library.demo.local/api/v1/books

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

2  ... "id": 5,
3  ... "title": "Guía de Estudio Para la Certificación Ccna Security",
4  ... "author": "Benjamin Lopez",
5  ... "isbn": "197554754"
6

```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 9 ms Size: 237 B Save as Example

Pretty Raw Preview Visualize JSON

http://library.demo.local/api/v1/books Save

POST http://library.demo.local/api/v1/books Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1  ... "id": 6,
2  ... "title": "Redes Cisco. CCNA Routing y Switching.",
3  ... "author": "David Carrasco",
4  ... "isbn": "212991317"
5
6

```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 9 ms Size: 237 B Save as Example

POST http://library.demo.local/api/v1/books Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

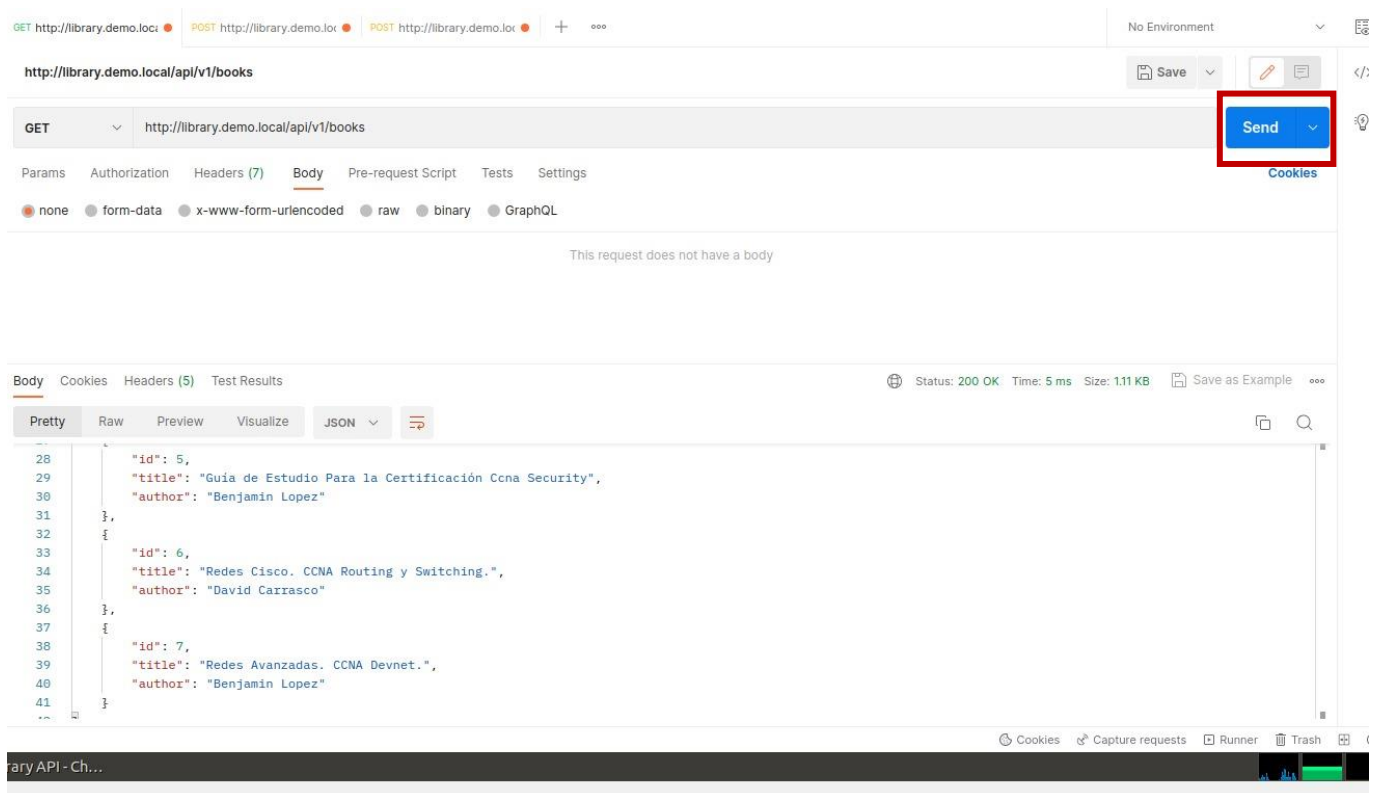
1  ... "id": 7,
2  ... "title": "Redes Avanzadas. CCNA Devnet.",
3  ... "author": "Benjamin Lopez",
4  ... "isbn": "197554754"
5
6

```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 9 ms Size: 237 B Save as Example

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

- 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).



- b. Para verificar que la solicitud de API fue correcta, ahora usted verá una respuesta que incluye el código de **estado** 200 OK en verde.

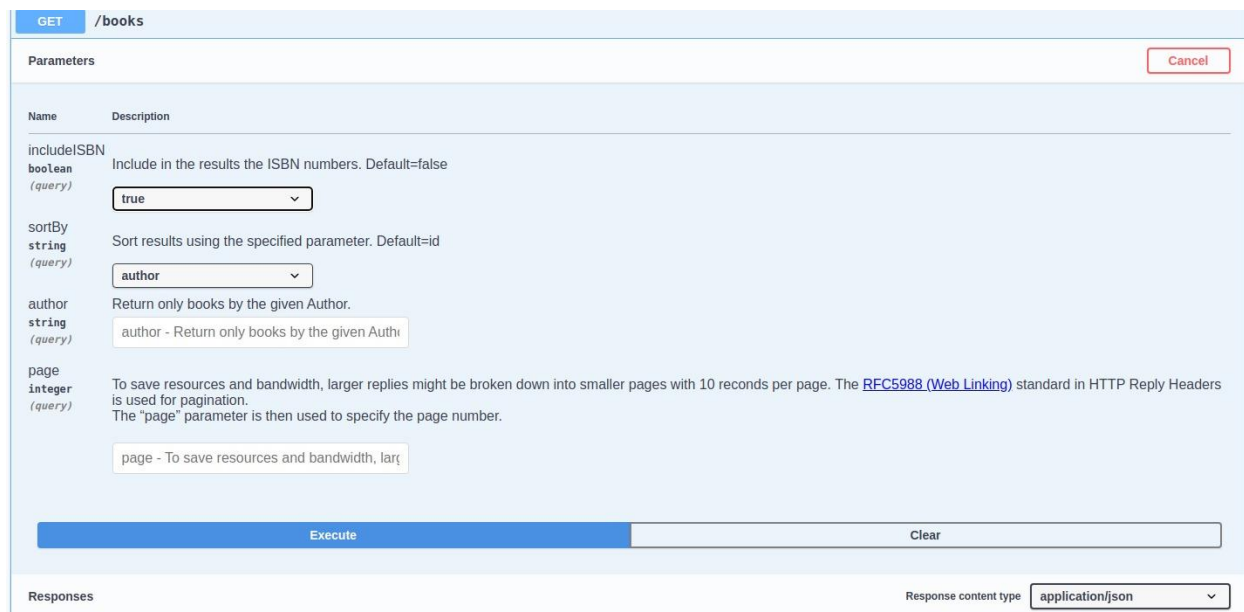


- c. Haga clic en **Cuerpo** para ver la respuesta. Observe que el predeterminado es **Pretty** y **JSON**.



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

- Vaya al sitio web de la **API de la biblioteca escolar**. Desplácese hacia arriba hasta **GET /books** API y expandirlo, si es necesario.
- Haga clic en **Probar**. Si ve un botón **Cancelar** en rojo, entonces no necesita seleccionar este botón.
- Bajo parámetros:
 - Haga clic en **IncluirSBN** y seleccione **verdadero**
 - Haga clic en **OrdenarPor** y seleccione **autor**
- Haga clic en **Ejecutar**.
- En el **cuerpo de la respuesta** verá la lista de libros ahora ordenados por autor e incluyendo los ISBN.



GET /books

Parameters Cancel

Name	Description
includeISBN boolean (query)	Include in the results the ISBN numbers. Default=false
sortBy string (query)	Sort results using the specified parameter. Default=id
author string (query)	Return only books by the given Author.
page integer (query)	To save resources and bandwidth, larger replies might be broken down into smaller pages with 10 records per page. The RFC5988 (Web Linking) standard in HTTP Reply Headers is used for pagination. The "page" parameter is then used to specify the page number.

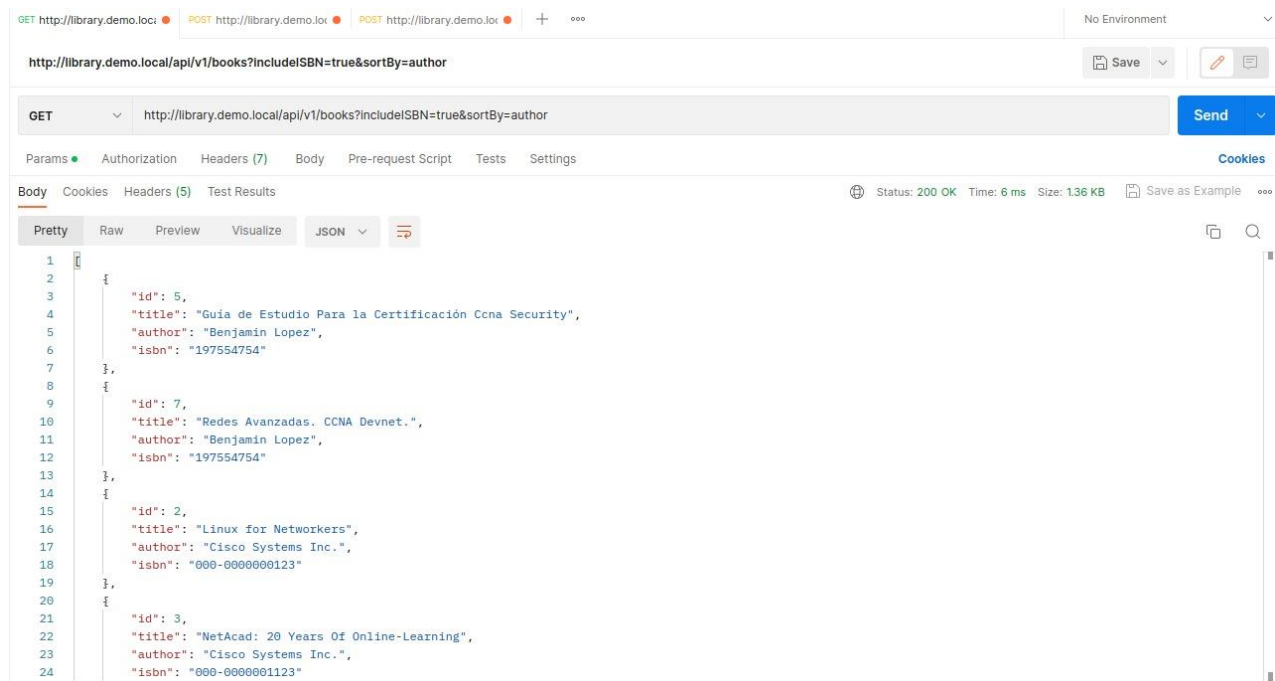
Execute Clear

Responses Response content type application/json

Code	Details
200	<p>Response body</p> <pre>[{ "id": 5, "title": "Guía de Estudio Para la Certificación Ccna Security", "author": "Benjamin Lopez", "isbn": "197554754" }, { "id": 7, "title": "Redes Avanzadas. CCNA Devnet.", "author": "Benjamin Lopez", "isbn": "197554754" }, { "id": 2, "title": "Linux for Networkers", "author": "Cisco Systems Inc.", "isbn": "000-0000001123" }, { "id": 3, "title": "NetAcad: 20 Years Of Online-Learning", "author": "Cisco Systems Inc.", "isbn": "000-0000001123" }, { "id": 6, "title": "Redes Cisco. CCNA Routing y Switching.", </pre> <p>Download</p>

Code	Details
200	<p>Response body</p> <pre> "author": "Cisco Systems Inc.", "isbn": "000-0000001123" }, { "id": 6, "title": "Redes Cisco. CCNA Routing y Switching.", "author": "David Carrasco", "isbn": "212991317" }, { "id": 0, "title": "IP Routing Fundamentals", "author": "Mark A. Sportack", "isbn": "978-1578700714" }, { "id": 4, "title": "IPv6 Fundamentals.", "author": "Rick Graziani", "isbn": "978 15871446778" }, { "id": 1, "title": "Python for Dummies", "author": "Stef Maruch Amh Maruch", "isbn": "978-0471778646" }] </pre> <p>Download</p>

- e. **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)**



```
1 {
2   {
3     "id": 5,
4     "title": "Guía de Estudio Para la Certificación Ccna Security",
5     "author": "Benjamin Lopez",
6     "isbn": "197554754"
7   },
8   {
9     "id": 7,
10    "title": "Redes Avanzadas. CCNA Devnet.",
11    "author": "Benjamin Lopez",
12    "isbn": "197554754"
13  },
14  {
15    "id": 2,
16    "title": "Linux for Networkers",
17    "author": "Cisco Systems Inc.",
18    "isbn": "000-0000000123"
19  },
20  {
21    "id": 3,
22    "title": "NetAcad: 20 Years Of Online-Learning",
23    "author": "Cisco Systems Inc.",
24    "isbn": "000-00000001123"
25  }
26 }
```

GET http://library.demo.local: POST http://library.demo.local: POST http://library.demo.local: + ... No Environment

http://library.demo.local/api/v1/books?includeISBN=true&sortBy=author Save

GET http://library.demo.local/api/v1/books?includeISBN=true&sortBy=author Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (5) Test Results Status: 200 OK Time: 6 ms Size: 1.36 KB Save as Example

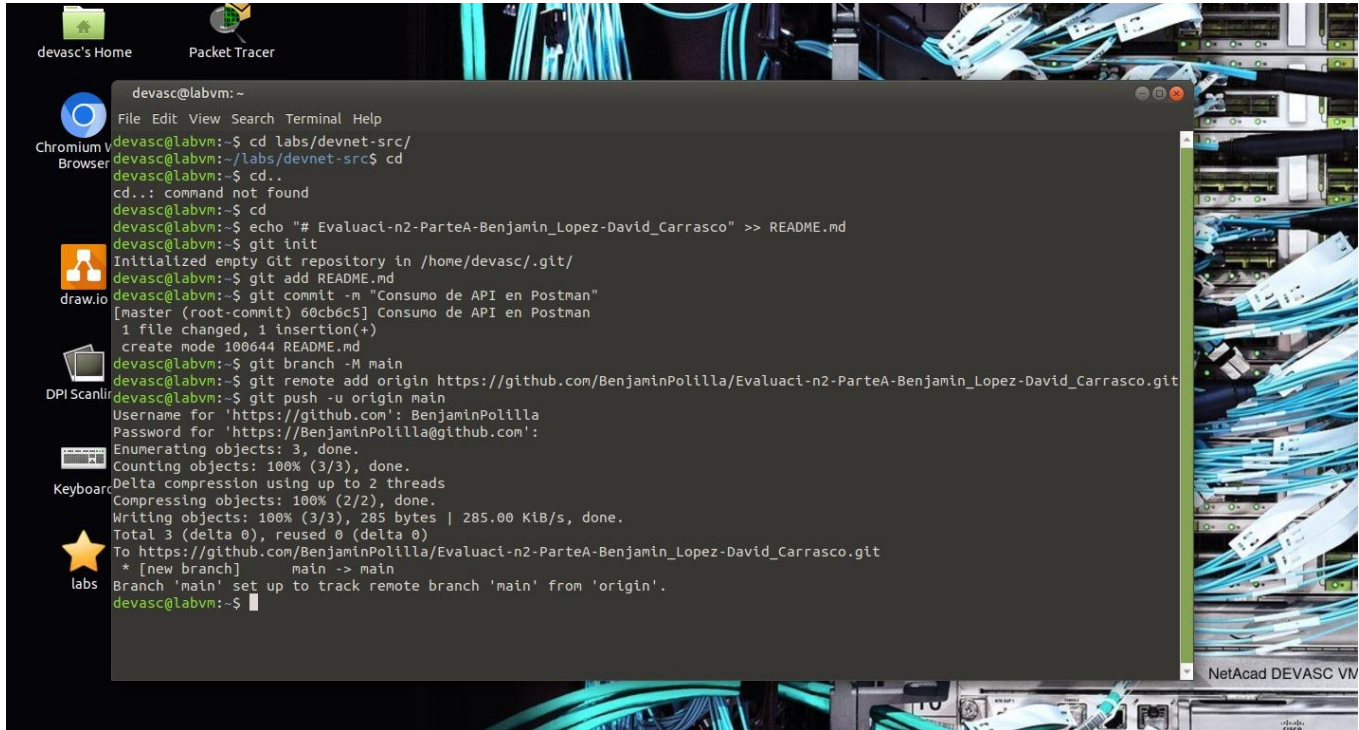
Pretty Raw Preview Visualize JSON

```

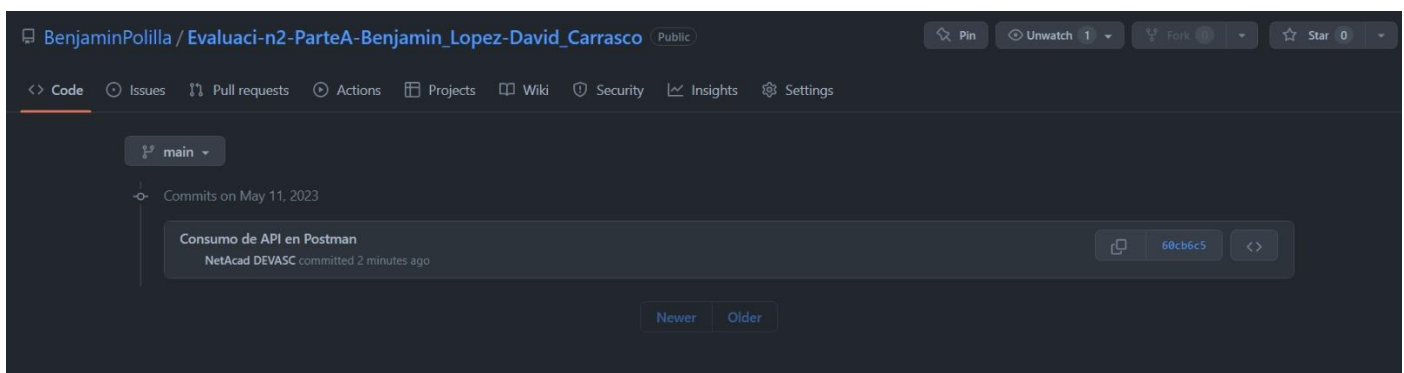
25  },
26  {
27    "id": 6,
28    "title": "Redes Cisco. CCNA Routing y Switching.",
29    "author": "David Carrasco",
30    "isbn": "212991317"
31  },
32  {
33    "id": 8,
34    "title": "IP Routing Fundamentals",
35    "author": "Mark A. Sportack",
36    "isbn": "978-1578700714"
37  },
38  {
39    "id": 4,
40    "title": "IPv6 Fundamentals.",
41    "author": "Rick Graziani",
42    "isbn": "978 15871446778"
43  },
44  {
45    "id": 1,
46    "title": "Python for Dummies",
47    "author": "Stef Maruch Aahz Maruch",
48    "isbn": "978-0471778646"

```

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”**. (10 Puntos)



```
devasc@labvm:~  
File Edit View Search Terminal Help  
devasc@labvm:~$ cd labs/devnet-src/  
devasc@labvm:~/labs/devnet-src$ cd  
devasc@labvm:~$ cd..  
cd.: command not found  
devasc@labvm:~$ cd  
devasc@labvm:~$ echo "# Evaluaci-n2-ParteA-Benjamin_Lopez-David_Carrasco" >> README.md  
devasc@labvm:~$ git init  
Initialized empty Git repository in /home/devasc/.git/  
devasc@labvm:~$ git add README.md  
devasc@labvm:~$ git commit -m "Consumo de API en Postman"  
[master (root-commit) 60cb6c5] Consumo de API en Postman  
1 file changed, 1 insertion(+)  
create mode 100644 README.md  
devasc@labvm:~$ git branch -M main  
devasc@labvm:~$ git remote add origin https://github.com/BenjaminPolilla/Evaluaci-n2-ParteA-Benjamin_Lopez-David_Carrasco.git  
devasc@labvm:~$ git push -u origin main  
Username for 'https://github.com': BenjaminPolilla  
Password for 'https://BenjaminPolilla@github.com':  
Enumerating objects: 3, done.  
Counting objects: 100% (3/3), done.  
Delta compression using up to 2 threads  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (3/3), 285 bytes | 285.00 KiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0)  
To https://github.com/BenjaminPolilla/Evaluaci-n2-ParteA-Benjamin_Lopez-David_Carrasco.git  
* [new branch] main -> main  
Branch 'main' set up to track remote branch 'main' from 'origin'.  
devasc@labvm:~$
```



Además comparte el archivo en Word en el ambiente de aprendizaje en el link disponible.

Luego Copie el link generado en GitHub y pegue en este documento. Para que así el docente pueda Evaluarlo.

https://github.com/BenjaminPolilla/Evaluaci-n2-ParteA-Benjamin_Lopez-David_Carrasco

B. Consumo de API Pública. Evaluación Individual. (60 Puntos)

Utilizando el sitio de MapQuest 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.

La entrega de este trabajo es hasta el domingo 14 de mayo hasta las 23:59 hrs. Subir el documento en Word en el link del Ambiente de Aprendizaje.

- Traducir al español los textos de todo el programa. **(10 puntos)**
- Cambiar los nombres de las variables a español y que sean distintos a los entregados por el laboratorio. **(5 puntos)**
- Solicitar “Ciudad de Origen” y “Ciudad de Destino” **(5 puntos)**

- Mostrar la duración del viaje en horas, minutos y segundos **(5 puntos)**
 - Medir la distancia en kilómetros entre dos ciudades distintas de la Quinta región. **(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” **(5 puntos)**
 - **Verificar si el funcionamiento es el solicitado. (5 puntos)**
-
- Subir el **script de Python** creado de **MapQuest** 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)**

Además comparte el archivo en Word en al ambiente de aprendizaje en el link disponible.

Luego Copie el link generado en GitHub y pegue en este documento. Para que así el docente pueda Evaluarlo.