

## TALLER 1 – PROGRAMACIÓN EN LA WEB

**Christian Ruiz Lagos**                      **2152204**

**Kevin Joel Dlaikan**                      **2160090**

1. Acceder a un sitio estático usando un navegador [https://es.wikipedia.org/wiki/P%C3%A1gina\\_web](https://es.wikipedia.org/wiki/P%C3%A1gina_web) <https://www.lashorasperdidas.com> (mirar headers e info interesante) Cual es la diferencia entre los códigos 200, 204, 304, 500

El navegador le solicita al servidor los archivos necesarios para mostrar la página web correctamente (HTML, CSS, Javascript, imágenes, etc.). Esto se realiza por medio del protocolo HTTP, realizando una petición con el método GET; cada petición tiene una respuesta, y estas respuestas son los códigos de estado.

- Los códigos 200 indican que todo fue exitoso.  
200 = Ok (la solicitud ha tenido éxito)  
204 = Accepted (la solicitud se ha recibido, pero aún no se ha actuado)
- Los códigos 300 indican redirecciones.  
304 =Not Modified (esta es usada para propósitos de "caché". Le indica al cliente que la respuesta no ha sido modificada. Entonces, el cliente puede continuar usando la misma versión almacenada en su caché.)
- Los códigos 500 indican errores en el servidor.  
500 = Internal Server Error (el servidor ha encontrado una situación que no sabe cómo manejarla.)

En [https://es.wikipedia.org/wiki/P%C3%A1gina\\_web](https://es.wikipedia.org/wiki/P%C3%A1gina_web) casi todas las peticiones fueron exitosas (códigos de estado 200), así como también hubo códigos de estado 304 que indican que algunos archivos solicitados ya estaban en caché.

Name	Status	Type	Initiator	Size	Time	W
P%C3%A1gina_web	304	document	Other	61 B	95 ms	
load.php?lang=es&modules=ext.cite....	200	stylesheet	<a href="#">Página web</a>	(memory ...)	0 ms	
load.php?lang=es&modules=ext.gad...	200	stylesheet	<a href="#">Página web</a>	(memory ...)	0 ms	
load.php?lang=es&modules=site.styl...	200	stylesheet	<a href="#">Página web</a>	(memory ...)	0 ms	
220px-Página_web_autoreferente.jpg	200	jpeg	<a href="#">Página web</a>	(memory ...)	0 ms	
220px-Alcal%C3%A1_de_Henares_%2...	200	webp	<a href="#">Página web</a>	(memory ...)	0 ms	
load.php?lang=es&modules=startup...	200	script	<a href="#">Página web</a>	(disk cach...	2 ms	
20px-Wikidata-logo.svg.png	200	webp	<a href="#">Página web</a>	(memory ...)	0 ms	
poweredby_mediawiki_88x31.png	200	png	<a href="#">Página web</a>	(memory ...)	0 ms	
wikimedia-button.png	200	png	<a href="#">Página web</a>	(memory ...)	0 ms	
wikipedia-wordmark-en.svg	200	svg+xml	<a href="#">Página web</a>	(memory ...)	0 ms	
magnify-clip-ltr.svg?8330e	200	svg+xml	<a href="#">Página web</a>	(memory ...)	0 ms	
bullet-icon.svg?d4515	200	svg+xml	<a href="#">Página web</a>	(memory ...)	0 ms	
external-link-ltr-icon.svg?b4b84	200	svg+xml	<a href="#">Página web</a>	(memory ...)	0 ms	
bullet-icon.svg?d4515	200	svg+xml	<a href="#">Página web</a>	(memory ...)	0 ms	
external-link-ltr-icon.svg?b4b84	200	svg+xml	<a href="#">Página web</a>	(memory ...)	0 ms	
external-link-ltr-icon.svg?b4b84	200	svg+xml	<a href="#">Página web</a>	(memory ...)	0 ms	
external-link-ltr-icon.svg?b4b84	200	svg+xml	<a href="#">Página web</a>	(memory ...)	0 ms	
user-avatar.svg?b7f58	200	svg+xml	<a href="#">Página web</a>	(memory ...)	0 ms	
search.svg?4d50a	200	svg+xml	<a href="#">Página web</a>	(memory ...)	0 ms	

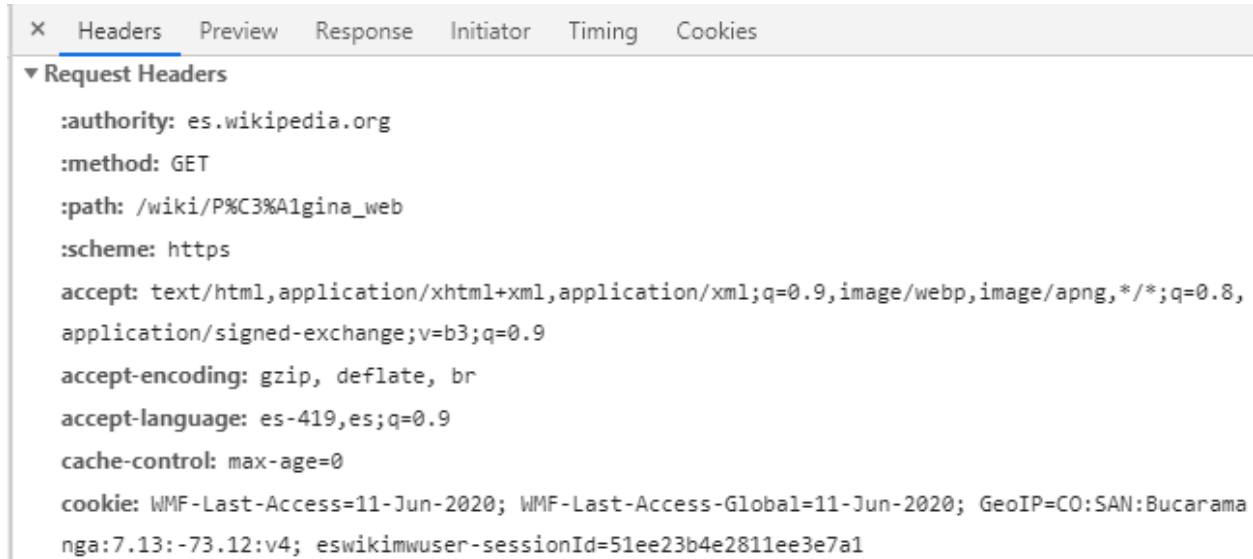
44 requests | 2.9 kB transferred | 846 kB resources | Finish: 43.31 s | **DOMContentLoaded: 178 ms** | **Load: 233 ms**

Cada archivo solicitado a la página posee un conjunto de headers (permiten al cliente y al servidor enviar información adicional junto a una petición o respuesta):

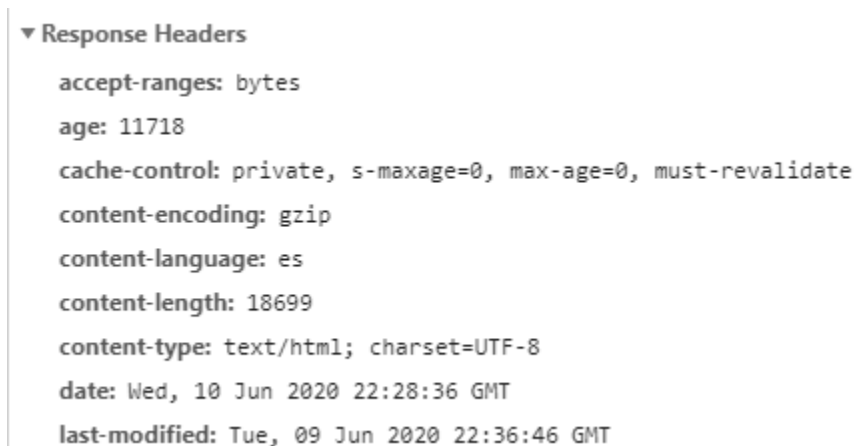
Los **headers generales** que se aplican tanto a las peticiones como a las respuestas, pero sin relación con los datos que finalmente se transmiten en el cuerpo.

A screenshot of the Chrome DevTools Network tab. The 'Headers' tab is selected. Under the 'General' section, the following information is displayed: Request URL: https://es.wikipedia.org/wiki/P%C3%A1gina\_web, Request Method: GET, Status Code: 304 (with a green circle icon), Remote Address: 208.80.154.224:443, and Referrer Policy: no-referrer-when-downgrade.

Los **Request headers** que contienen más información sobre el contenido que va a obtenerse o sobre el cliente



Los **Response headers** que contienen más información sobre el contenido, como su origen o el servidor (nombre, versión, etc.).



En <https://www.lashorasperdidas.com> la mayor parte de los códigos de estado fueron 200, sin embargo, hubo más variedad. Hay estados "failed" que los interpreto como que los recursos solicitados han sido eliminados, además de códigos 302 (el recurso solicitado ha cambiado temporalmente) y 304 (el recurso se encuentra en caché). Finalmente obtuve códigos 403 (**Forbidden**) que indica que el cliente no posee los permisos necesarios para cierto contenido, por lo que el servidor está rechazando otorgar una respuesta apropiada.

Name	Status	Type	Initiator	Size	Time	Wat
jot	302	text/html	?doing_wp_cron=...	100 B	102 ms	
sodar2.js	200	script	pubads_impl_2020...	(disk cach...	1 ms	
runner.html	200	document	sodar2.js:25	(disk cach...	1 ms	
sodar?sv=200&tid=gpt&tv=2020060...	200	xhr	pubads_impl_2020...	5.7 kB	106 ms	
abg_lite_fy2019.js	200	script	zrt_lookup.html?fs...	7.5 kB	52 ms	
jot.html	304	document	syndication.twitter...	427 B	67 ms	
window_focus_fy2019.js	200	script	zrt_lookup.html?fs...	(disk cach...	35 ms	
osd_listener.js?cache=r20110914	200	script	zrt_lookup.html?fs...	(disk cach...	46 ms	
qs_click_protection_fy2019.js	200	script	zrt_lookup.html?fs...	(disk cach...	45 ms	
l?ebcid=ALh7CaTbkZtnTwNAIVDRHb...	204	text/html	zrt_lookup.html?fs...	0 B	108 ms	
one_click_handler_one_afma_fy2019.js	200	script	zrt_lookup.html?fs...	(disk cach...	43 ms	
2830664535836983611?sqp=4sqPyQ...	200	jpeg	zrt_lookup.html?fs...	14.5 kB	55 ms	
sodar2.js	200	script	pubads_impl_2020...	(memory ...	0 ms	
runner.html	200	document	sodar2.js:25	(disk cach...	2 ms	
re?v=r20150723	200	document	zrt_lookup.html?fs...	210 B	89 ms	
rAuT1ABTnpq02XvzkEYnei6rTZKfo9qz...	200	script	runner.html:20	(disk cach...	2 ms	
rAuT1ABTnpq02XvzkEYnei6rTZKfo9qz...	200	script	zrt_lookup.html?fs...	(disk cach...	1 ms	
sodar?sv=200&tid=gpt&tv=2020060...	200	xhr	pubads_impl_2020...	5.6 kB	96 ms	
foroicono.gif	500	text/html	Other	0 B	30 ms	
activeview?xai=AKAOisuSld9z61K5-2...	200	aif	osd_listener.is?cac...	65 B	89 ms	
257 requests   511 kB transferred   6.6 MB resources   Finish: 1.0 min   DOMContentLoaded: 799 ms   Load: 2.33 s						

appnexus?anxs_uid=81275420977155...	(failed)		async_usersync.ht...	0 B	1.9 min	
usync.php?px=guSgCI60&puid=8127...	200	text/plain	async_usersync.ht...	192 B	320 ms	
sync.gif?dm=ib.adnxs.com&fck=8127...	403	text/html	async_usersync.ht...	0 B	187 ms	
usermatch.gif?adnxs_uid=812754209...	204	text/plain	async_usersync.ht...	337 B	311 ms	
user?vendor_id=1&uid=81275420977...	200	json	async_usersync.ht...	95 B	289 ms	
pixelSync?nid=84&gdpr=0&gdpr_co...	302		async_usersync.ht...	748 B	379 ms	
dispatch?ptr=10901	200	gif	async_usersync.ht...	86 B	1.05 s	
ping_match.gif?ei=AN&rurl=https://i...	302		async_usersync.ht...	742 B	392 ms	
bct?pid=7cbd1ad4-ddd6-44bd-986a-...	200	gif	async_usersync.ht...	342 B	320 ms	
?id=16975&user_id=8127542097715...	200	json	async_usersync.ht...	70 B	415 ms	
sync.gif?dm=ib.adnxs.com&fck=8127...	403	text/html	async_usersync.ht...	0 B	132 ms	
appnexus/	302		async_usersync.ht...	313 B	2.0 min	
get?stype=sync&sExtCookieId=8127...	302	text/plain	async_usersync.ht...	539 B	380 ms	

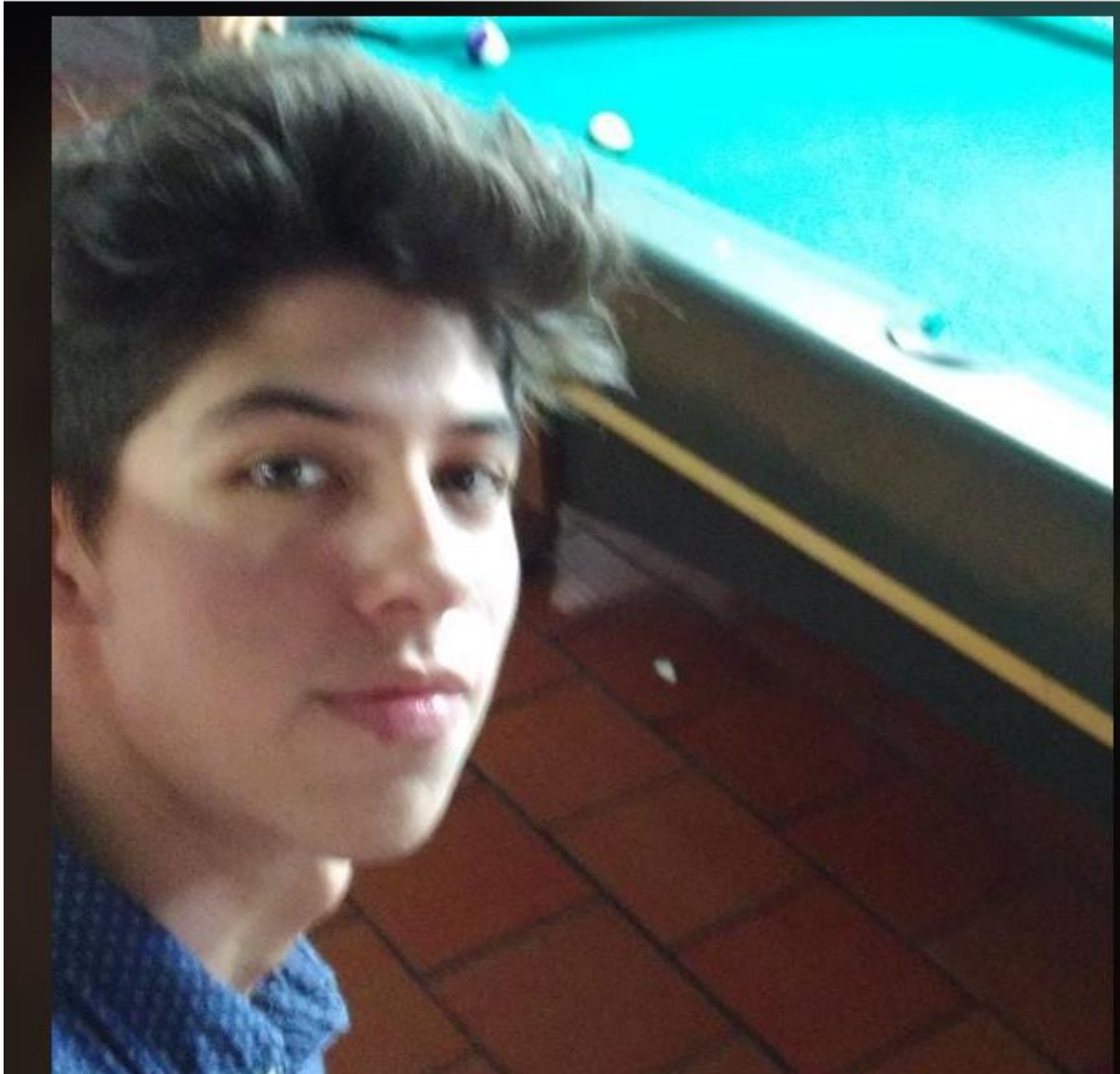
2. Entre a los sitios de Facebook analice la autenticación

Entre y mire la siguiente dirección [https://developers.facebook.com/docs/facebook-login/access-tokens/?locale=es\\_LA](https://developers.facebook.com/docs/facebook-login/access-tokens/?locale=es_LA)


Cambie parámetros mire el resultado

<https://www.facebook.com/photo?fbid=183366732784261&set=a.113297896457812>

facebook.com/photo?fbid=183366732784261&set=a.113297896457812



Tomé una foto de mi perfil de Facebook para analizar los parámetros.

Name	Status	Type	Initiator	Size	Time
 photo?fbid=183366732784261&set=...	200	document	Other	166 kB	769 ms

En el archivo correspondiente al HTML de la página, en la parte final se pueden observar los siguientes datos:

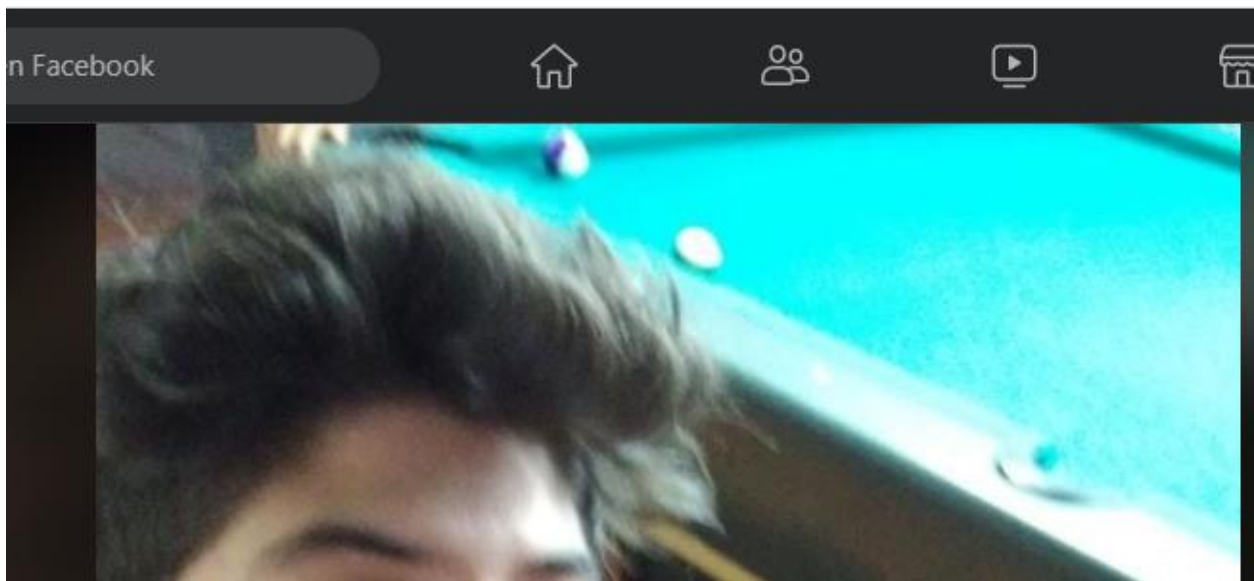
▼ Query String Parameters	view source	view URL encoded
fbid: 183366732784261		
set: a.113297896457812		

Donde fbid y set son los parámetros también presentes en la URL (después del signo de interrogación“?” y separados por un “&”):

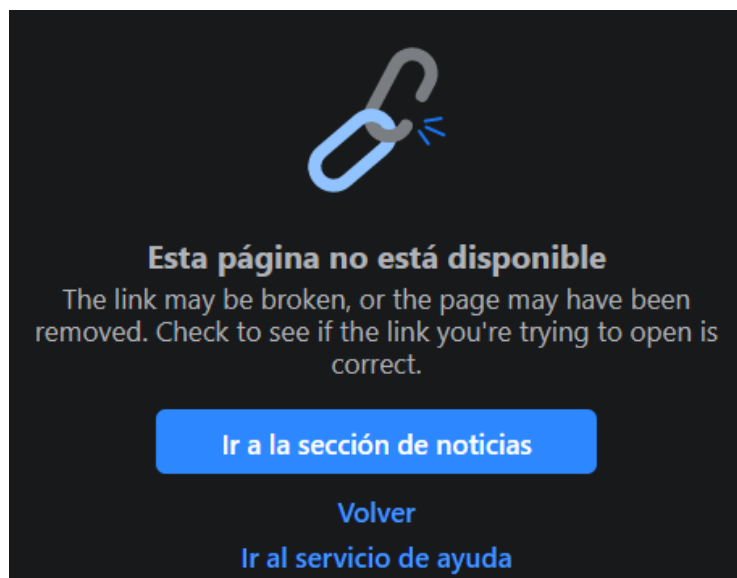
facebook.com/photo?fbid=183366732784261&set=a.113297896457812

Si a la URL le quito el parámetro “set”, la página sigue mostrándose con normalidad.

facebook.com/photo?fbid=183366732784261



Sin embargo, si le quito el parámetro “fbid”, la página ya no se mostrará.





Probando con otra foto (también de mi galería de fotos de perfil), vemos que el parámetro "set" no cambia, lo cual puede hacer referencia a que las fotos que se están visualizando pertenecen a un mismo conjunto (fotos de perfil). Por otra parte, el parámetro "fbid" sí es diferente para cada foto, puesto que es un identificador único, y es realmente el único parámetro fundamental para poder mostrar la foto.

facebook.com/photo?fbid=153019825818952&set=a.113297896457812



Ahora, agregando a esta misma URL parámetros:

<https://www.facebook.com/photo?fbid=153019825818952&set=a.113297896457812&estoes=unaprueba&nombre1=Christian%20Ruiz&nombre2=Kevin%20Dlaikan>

Donde se le agregó a la URL lo siguiente:

[2&estoes=unaprueba&nombre1=Christian%20Ruiz&nombre2=Kevin%20Dlaikan](#)

E inspeccionando el archivo HTML, en la sección de parámetros, se obtuvo:

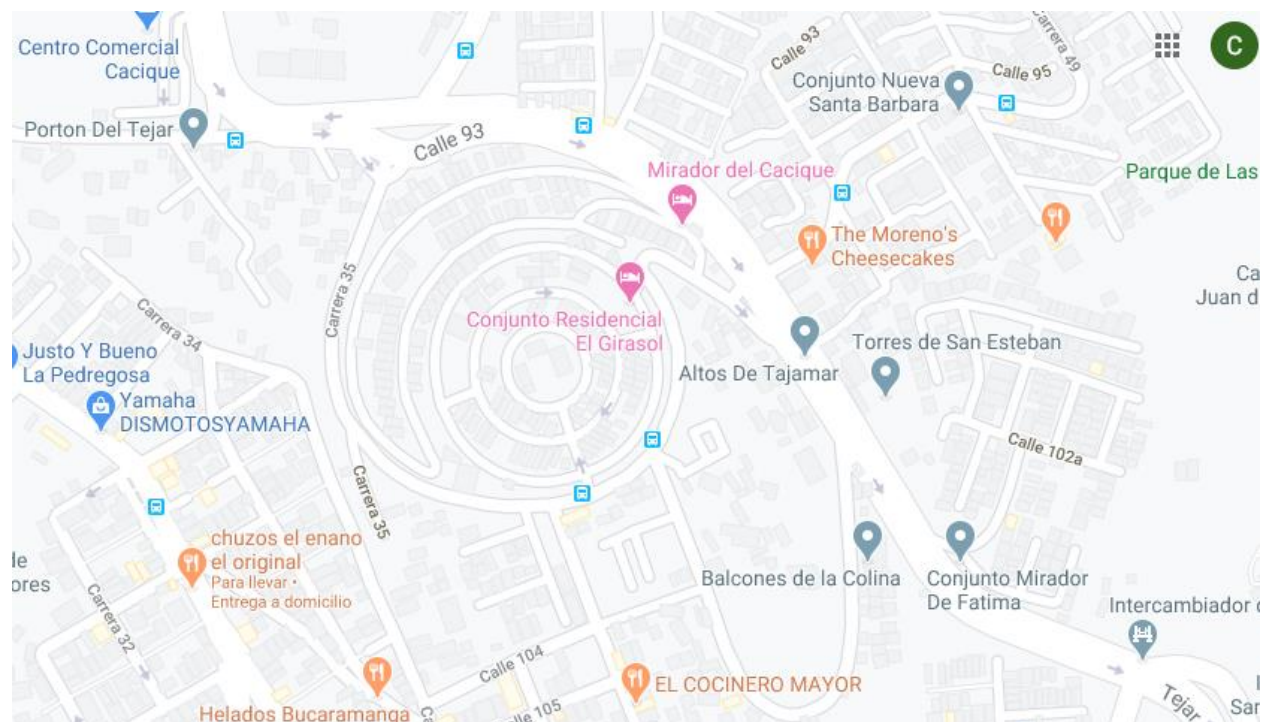
▼ Query String Parameters   view source   view URL encoded

**fbid:** 153019825818952  
**set:** a.113297896457812  
**esto:** unaprueba  
**nombre1:** Christian Ruiz  
**nombre2:** Kevin Dlaikan

3. Entre a google maps mire los parametros de la url cambielos y analice resultados cuales son los parámetros que pudo cambiar y cuales aparecen con las funcionalidades.

Al ingresar a Google maps me centra en mi ubicación:

<https://www.google.com/maps/@7.0952962,-73.1060177,17z>



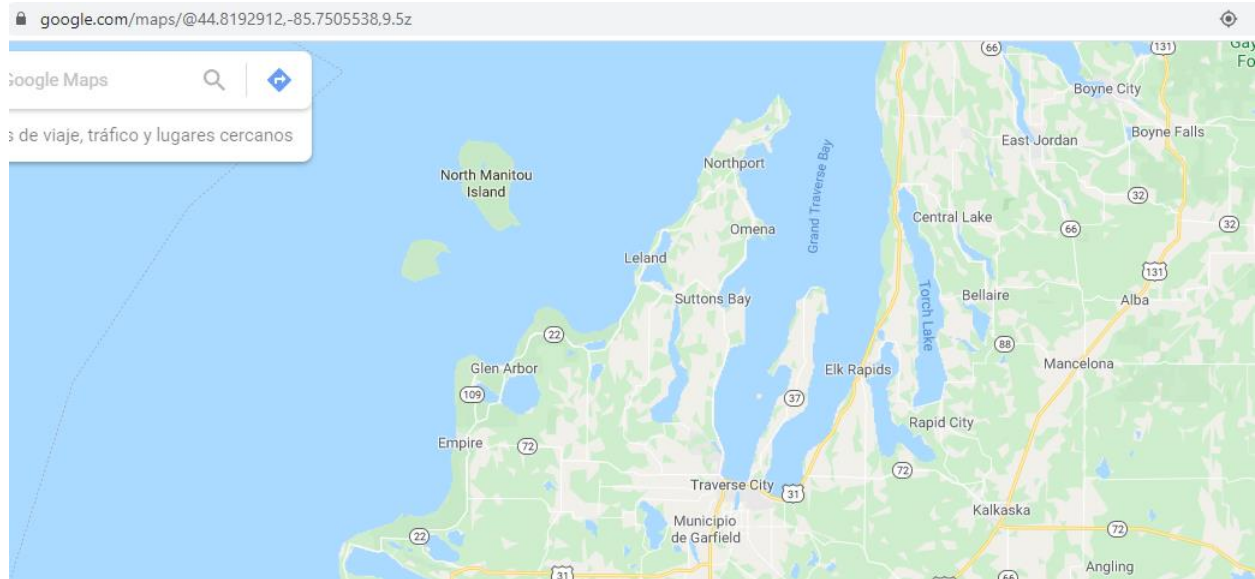
Donde los parámetros modificables son respectivamente:

[@7.0952962,-73.1060177,17z](https://www.google.com/maps/@7.0952962,-73.1060177,17z)

Que parecen ser coordenadas de longitud y latitud. Modificando levemente las cifras menos significativas de los parámetros, no resultan en grandes cambios (sigo estando en la misma locación).

Por otra parte, un cambio en los primeros dígitos de los parámetros resulta en una locación totalmente distinta (otras partes del mundo).





4. Entre a la siguiente dirección: <https://docs.postman-echo.com/?version=latest>

Use <https://postman-echo.com>

Realice:

GET GET Request

POST POST Raw Text

POST POST Form Data

PUT PUT Request

PATCH PATCH Request

DEL DELETE Request

Analice cada comando y que función tiene dentro de una página web.

- **GET** Request

El método HTTP GET es usado para recuperar información de un servidor. La información es identificada por una URI (Uniform Resource Identifier) única.

Una petición GET puede pasar parámetros al servidor usando "Query String Parameters". Por ejemplo:

<https://postman-echo.com/get?estudiante1=christian&estudiante2=kevin>

GET First Api No Environment

First Api Comments 0

GET https://postman-echo.com/get?estudiante1=christian&estudiante2=kevin Send

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

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> estudiante1	christian	
<input checked="" type="checkbox"/> estudiante2	kevin	
Key	Value	Description

Los Query Params en este caso son:

estudiante1 = christian

estudiante2 = kevin

De la misma manera, la API nos responde con un formato JSON con la siguiente información:

Body Cookies (1) Headers (7) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON ≡

```

1 {
2   "args": {
3     "estudiante1": "christian",
4     "estudiante2": "kevin"
5   },
6   "headers": {
7     "x-forwarded-proto": "https",
8     "x-forwarded-port": "443",
9     "host": "postman-echo.com",
10    "x-amzn-trace-id": "Root=1-5ee24abe-5d33add56e7ac7fa3ef6c319",
11    "user-agent": "PostmanRuntime/7.25.0",
12    "accept": "*/*",
13    "postman-token": "6c45dd2c-1a9d-45fd-a9bf-885540858cc9",
14    "accept-encoding": "gzip, deflate, br",
15    "cookie": "sails.sid=s%3AST_tIk3QF-WmO53VyjapVUceK1R7ai-3.zD5AWK3xnKPAX1xdUIovy%2BBn%2BDk"
16  },
17   "url": "https://postman-echo.com/get?estudiante1=christian&estudiante2=kevin"
18 }

```

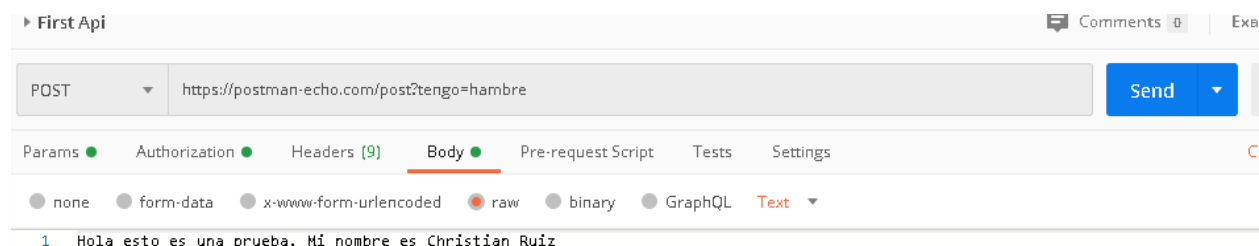
Vemos como la información que pasamos por parámetros está presente en el formato JSON que nos devuelve la API ("args").

- **POST** Raw Text

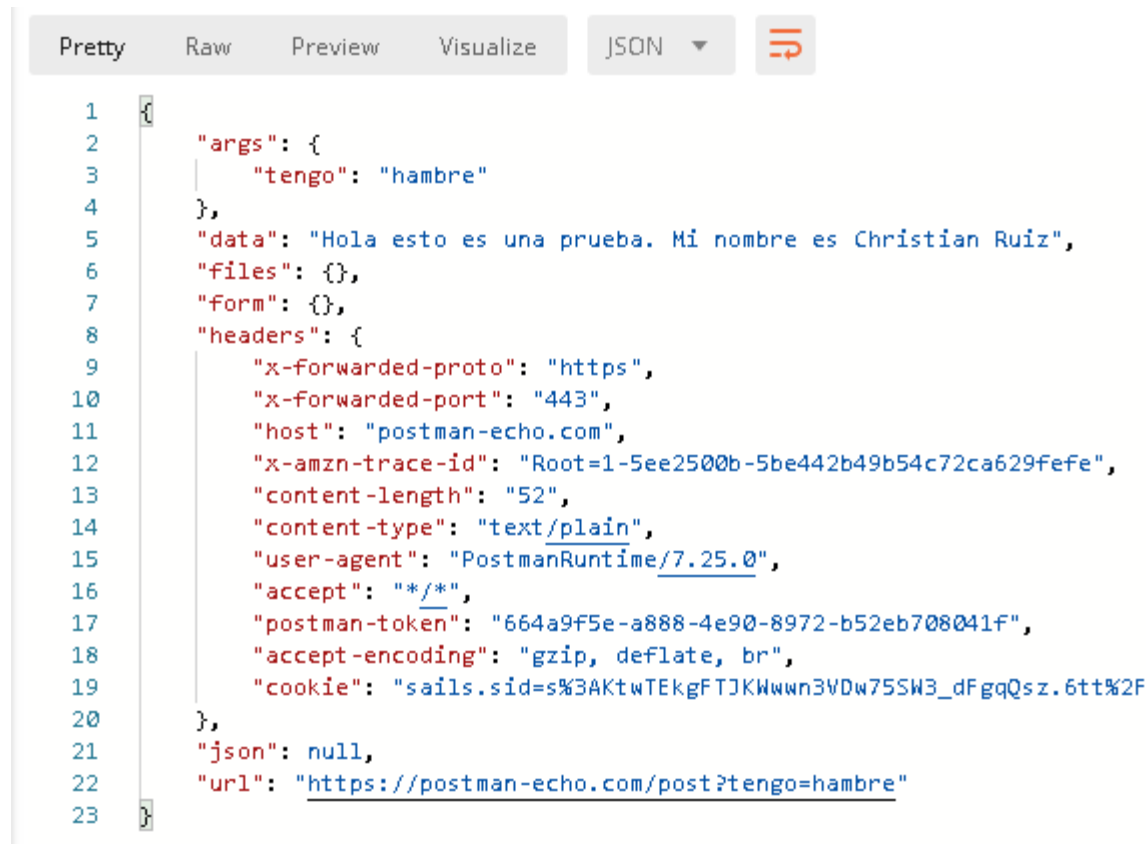
El método HTTP POST se usa para transferir información a un servidor (y obtener una respuesta). Qué información es retornada depende de la implementación del servidor.

Una petición POST puede pasar parámetros al servidor usando "Query String Parameters", así como en el cuerpo de la petición.

El cuerpo del request puede ir en diferentes formatos, para el ejemplo usé un texto crudo (Raw Text) y además añadí un Query String Param al request:



La respuesta la formateé en JSON y podemos ver los args con el parámetro que envié y en data la información suministrada en el body del request.



- **POST** Form Data

En este caso a la información del Body le cambié el formato a form-data y pasé la información de manera idéntica a como se pasan parámetros.

First Api Comments 0 Ext

POST ▼ <https://postman-echo.com/post?html=noesunlenguajedeprogramación> Send ▼

Params ● Authorization ● Headers [9] **Body ●** Pre-request Script Tests Settings ⌵

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

	KEY	VALUE	DESCRIPTION	**
<input checked="" type="checkbox"/>	I hate	living		
<input checked="" type="checkbox"/>	hola	profesor		

Obtuve como respuesta el siguiente JSON con la información que suministré:

Pretty Raw Preview Visualize JSON ▼ ≡

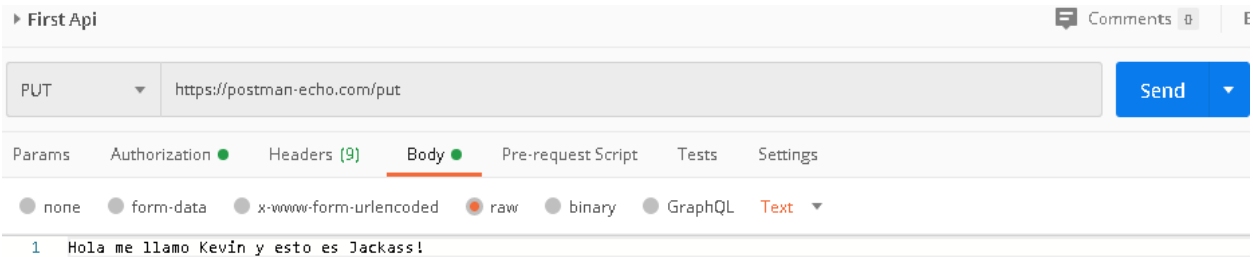
```

1  {
2    "args": {
3      "html": "noesunlenguajedeprogramación"
4    },
5    "data": {},
6    "files": {},
7    "form": {
8      "I hate": "living",
9      "hola": "profesor"
10   },
11   "headers": {
12     "x-forwarded-proto": "https",
13     "x-forwarded-port": "443",
14     "host": "postman-echo.com",
15     "x-amzn-trace-id": "Root=1-5ee251d3-c34aa6d65cc4f2e88aae0b...",
16     "content-length": "278",
17     "user-agent": "PostmanRuntime/7.25.0",
18     "accept": "*/*",
19     "postman-token": "2abb2681-9ad1-46f0-83b8-3241e59fdF27",
20     "accept-encoding": "gzip, deflate, br",
21     "content-type": "multipart/form-data; boundary=-----",
22     "cookie": "sails.sid=s%3A1CdJ46WNM5fZNNhsqyQ11IS64SW3cuES.1..."
23   },
24   "json": null,
25   "url": "https://postman-echo.com/post?html=noesunlenguajedeprog..."
26 }
```

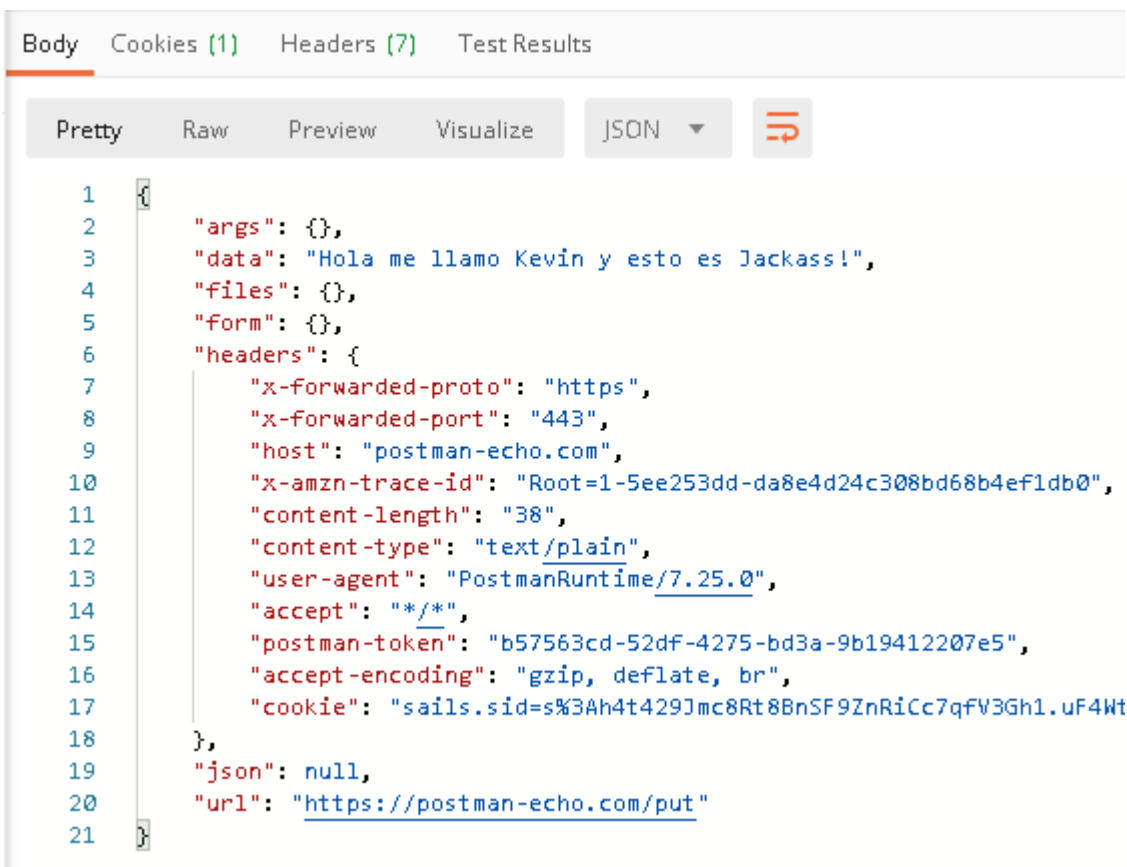
- **PUT Request**

El método HTTP PUT es similar al método POST. También está destinado a transferir información a un servidor (y obtener una respuesta). La información retornada depende de la implementación del servidor.

También puede pasar Query String Parameters, así como el cuerpo del request.



En esta ocasión no pasé Query Params y el cuerpo del request está en Raw Text. La respuesta obtenida fue la siguiente:

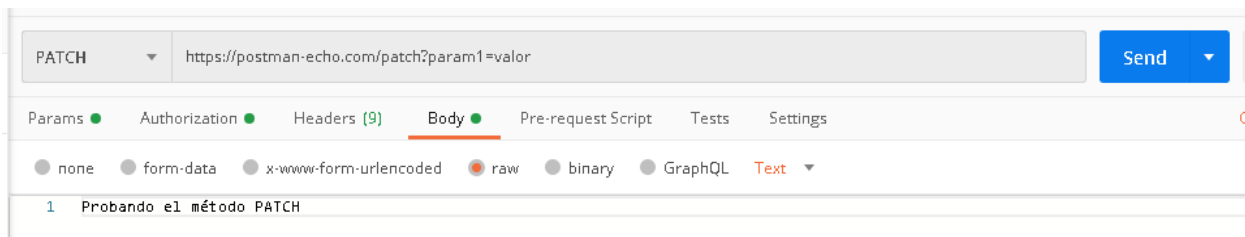


- **PATCH Request**

El método HTTP PATCH es usado para actualizar recursos de un servidor. El uso exacto de peticiones PATCH depende del servidor en cuestión. Hay implementaciones de servidores que manejan PATCH de forma diferente.

Técnicamente, PATCH soporta tanto Query String Parameters como Request Body.



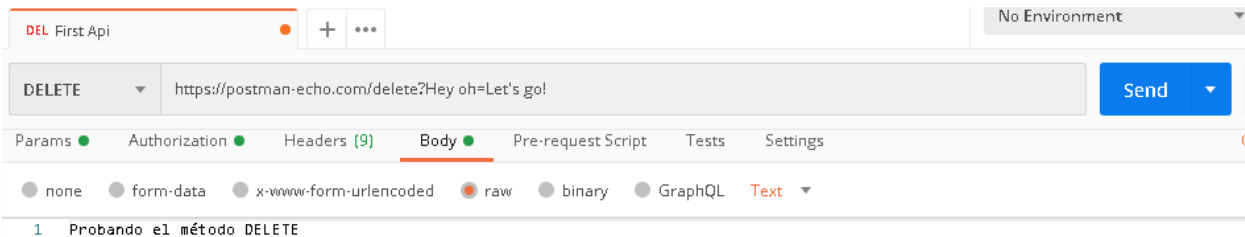


Obtengo un JSON con los headers, parámetros y el cuerpo de la petición:

```
1 {
2   "args": {
3     "param1": "valor"
4   },
5   "data": "Probando el método PATCH",
6   "files": {},
7   "form": {},
8   "headers": {
9     "x-forwarded-proto": "https",
10    "x-forwarded-port": "443",
11    "host": "postman-echo.com",
12    "x-amzn-trace-id": "Root=1-5ee2569c-10b1e4fc2aabdbe678e9a276",
13    "content-length": "25",
14    "content-type": "text/plain",
15    "user-agent": "PostmanRuntime/7.25.0",
16    "accept": "*/*",
17    "postman-token": "d621e4d4-6eb5-4fbd-b0b8-149516347c8b",
18    "accept-encoding": "gzip, deflate, br",
19    "cookie": "sails.sid=s%3Azj97I0m7m1vYN1n7TyeHPrDtHBR01YJE.DKoD68d4Zl"
20  },
21   "json": null,
22   "url": "https://postman-echo.com/patch?param1=valor"
23 }
```

- **DEL** Request

El método HTTP DELETE es usado para borrar recursos en un servidor. El uso exacto de este método depende de la implementación del servidor. En general, las peticiones DELETE soportan Query String Parameters y Body request.



```
1  {
2    "args": {
3      "Hey oh": "Let's go!"
4    },
5    "data": "Probando el método DELETE",
6    "files": {},
7    "form": {},
8    "headers": {
9      "x-forwarded-proto": "https",
10     "x-forwarded-port": "443",
11     "host": "postman-echo.com",
12     "x-amzn-trace-id": "Root=1-5ee257fd-aed416c4aa42db5a477d0120",
13     "content-length": "26",
14     "content-type": "text/plain",
15     "user-agent": "PostmanRuntime/7.25.0",
16     "accept": "*/*",
17     "postman-token": "cfab52fc-55cf-4091-a0c0-11395eb71284",
18     "accept-encoding": "gzip, deflate, br",
19     "cookie": "sails.sid=s%3Azj97I0m7m1vYN1n7TyeHPrDtHBR01YJE.DKoD"
20   },
21   "json": null,
22   "url": "https://postman-echo.com/delete?Hey%20oh=Let%27s%20go%21"
23 }
```

5. De postman-echo realice un get de repuesta json, analice la respuesta. Ahora compárelo contra el siguiente. **linkjsongis** Que diferencias en los parámetros encuentra y que diferencia en la respuesta.

<https://postman-echo.com/get?estudiante1=christian&estudiante2=kevin>

GET	https://postman-echo.com/get?estudiante1=christian&estudiante2=kevin	Send	Save
Params	Authorization	Headers (8)	Body
Pre-request Script	Tests	Settings	Cookies Code
Query Params			
KEY	VALUE	DESCRIPTION	...
<input checked="" type="checkbox"/> estudiante1	christian		
<input checked="" type="checkbox"/> estudiante2	kevin		
Key	Value	Description	

Respuesta en formato JSON:

```
1 {
2   "args": {
3     "estudiante1": "christian",
4     "estudiante2": "kevin"
5   },
6   "headers": {
7     "x-forwarded-proto": "https",
8     "x-forwarded-port": "443",
9     "host": "postman-echo.com",
10    "x-amzn-trace-id": "Root=1-5ee25fe2-c374678066482bce11ce43c",
11    "authorization": "Basic aG9sYTpxdWVhc2U=",
12    "user-agent": "PostmanRuntime/7.25.0",
13    "accept": "*/*",
14    "postman-token": "9a7bf9e7-8919-4c59-b850-33d133da01b1",
15    "accept-encoding": "gzip, deflate, br",
16    "cookie": "sails.sid=s%3A_gXoZj79NQrnPcl_NNBp6XCuk165dzzQ.6Hz9uok4UFqNdRZCjztF102ZoiVmUw60F5vrQDORlw0"
17  },
18  "url": "https://postman-echo.com/get?estudiante1=christian&estudiante2=kevin"
19 }
```

La respuesta me da información sobre los argumentos que le suministré y los headers.

Ahora bien, comparándolo con **linkjsongis**, puedo notar que el formato de respuesta no parece ser el mismo de JSON, es decir, no tiene una notación de objetos como en Javascript. Por otra parte, parece ser un formato XML.

Para verificar esto, cambié el formato de respuesta a XML:

```
1 {"args":{"estudiante1":"christian","estudiante2":"kevin"},"headers":{"x-Forwarded-proto":"https","x-Forwarded-port":"443","host":"postman-echo.com","x-amzn-trace-id":"Root=1-5ee25fe2-c374678066482bce11ce43c","authorization":"Basic aG9sYTpxdWVhc2U=","user-agent":"PostmanRuntime/7.25.0","accept":"*/*","postman-token":"9a7bf9e7-8919-4c59-b850-33d133da01b1","accept-encoding":"gzip, deflate, br","cookie":"sails.sid=s%3A_gXoZj79NQrnPcl_NNBp6XCuk165dzzQ.6Hz9uok4UFqNdRZCjztF102ZoiVmUw60F5vrQDORlw0"},"url":"https://postman-echo.com/get?estudiante1=christian&estudiante2=kevin"}
```

En formato XML sí parecen tener más similitudes entre sí.

6. Metodos de Autenticación realice en postman-echo GET Basic Auth Analice los parametros.

## GET Basic Auth

Para la autenticación POSTMAN acepta un usuario y una contraseña por defecto como credenciales de un login común y corriente. Si las credenciales son correctas devuelve un código de estado "200 okay", si no, devuelvo un "401 unauthorized".

Probando con un usuario y contraseña cualquiera:

The screenshot shows the Postman interface for a GET request to `https://postman-echo.com/basic-auth?esto es un parametro`. The **Authorization** tab is selected, showing **Basic Auth** type. The **Username** field contains `hola` and the **Password** field contains `quease`. The **Show Password** checkbox is checked. A warning message states: "Heads up! These parameters hold sensitive data. To keep this data secure while working environment, we recommend using variables. [Learn more about variables](#)". The status bar at the bottom indicates **Status: 401 Unauthorized** and **Time: 453 ms**. The **Body** tab is selected, showing the response `1 Unauthorized`.

Probando con las credenciales por defecto (Username = postman, Password = password):

The screenshot shows the Postman interface for a GET request to `https://postman-echo.com/basic-auth`. The **Authorization** tab is selected, showing **Basic Auth** type. The **Username** field contains `postman` and the **Password** field contains `password`. The **Show Password** checkbox is checked. A warning message states: "Heads up! These parameters hold sensitive data. To keep this data secure while working in a collabor environment, we recommend using variables. [Learn more about variables](#)". The status bar at the bottom indicates **Status: 200 OK**, **Time: 109 ms**, and **Size: 358 B**. The **Body** tab is selected, showing the response in JSON format: `1 {`, `2 "authenticated": true`, `3 }`.

Como introduje las credenciales por defecto, POSTMAN me devuelve un JSON con un único dato que dice que la autenticación es verdadera.

Al pasar un parámetro, podemos observar que la respuesta en caso de ser True, no nos muestra los parámetros ingresados, así como el Body request, que pese a que aceptar un formato Form-Data, la respuesta tampoco refleja esta información.

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

Query Params

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	estoesun	parametro	
	Key	Value	Description

Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 1

Pretty Raw Preview Visualize JSON ↕

```
1 {  
2   "authenticated": true  
3 }
```

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

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

	KEY	VALUE	DESCRIPTION
<input type="checkbox"/>	I hate	living	
<input checked="" type="checkbox"/>	hola	profesor	
	Key	Value	Description

Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 100 ms Size: 362 B

Pretty Raw Preview Visualize JSON ↕

```
1 {  
2   "authenticated": true  
3 }
```