



**Buholegal S. de R.L. de C.V.**

# **Manual de usuario**

**API para la búsqueda de  
antecedentes legales**

**v3.1**

# Indice

<b>Introducción</b>	<b>3</b>
<b>Endpoints</b>	<b>4</b>
Obtener token	7
Abreviaturas para cada estado	9
Realizar búsquedas judiciales	12
Realizar búsquedas judiciales utilizando solo el CURP	19
Consultar búsquedas realizadas	28
Consultar cantidad de búsquedas realizadas	30
Consultar búsquedas en la canasta (solo para clientes prepago)	34
Validar CURP	36
Consulta de Homonimia	40
Listas negras del SAT	43
Edictos Federales	46
<b>Ambiente de pruebas</b>	<b>48</b>
<b>Uso de la herramienta Postman</b>	<b>49</b>
Obtener token	49
Uso del token en los endpoints	49

# Introducción

El presente documento tiene como objetivo apoyar a los usuarios que utilicen el API de Buholegal para búsquedas judiciales. En él se describen cada uno de los endpoints que ofrece el API, incluyendo ejemplos de consultas y resultados para un mejor entendimiento.

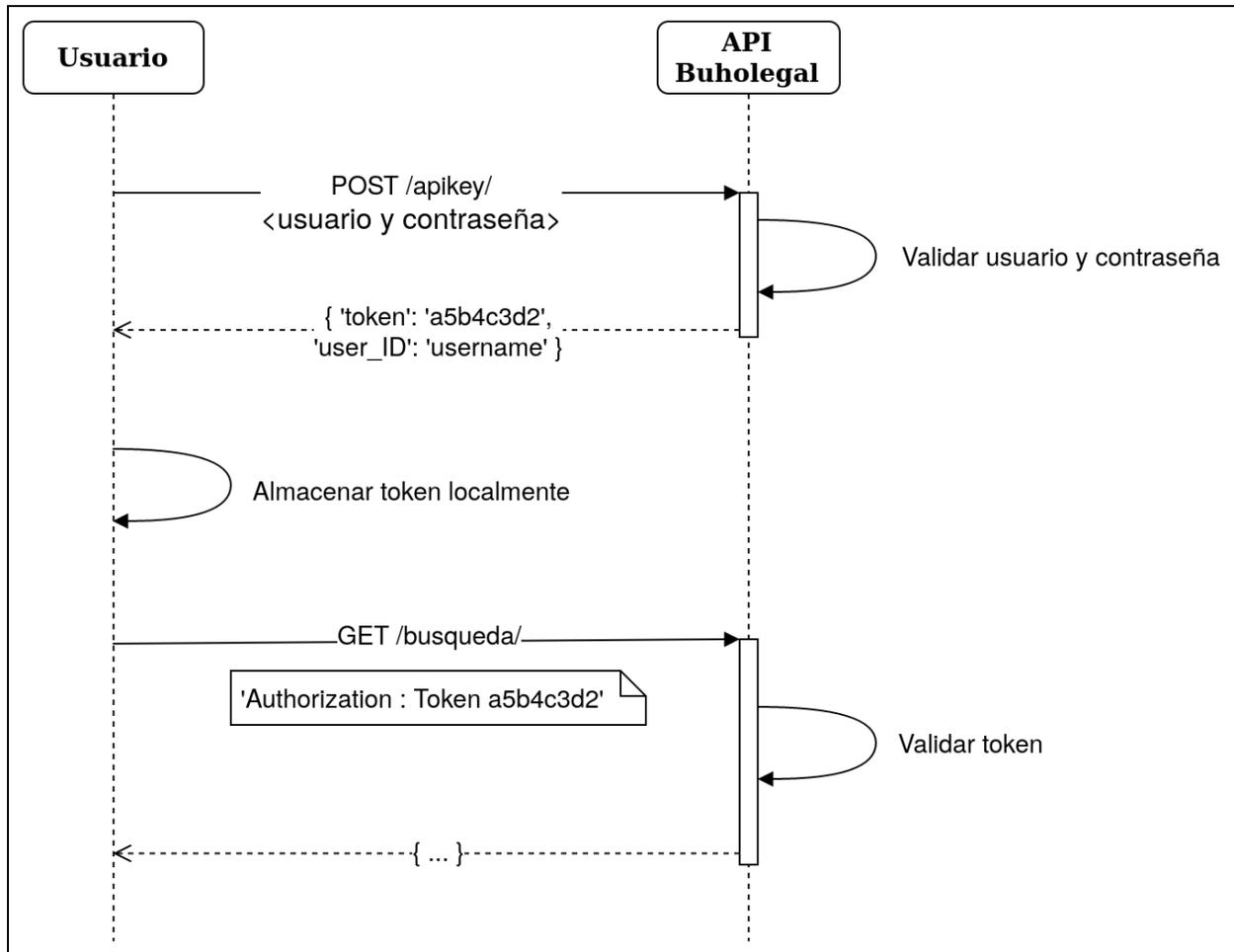
# Endpoints

En este apartado se describen cada uno de los endpoints que conforman el API de búsquedas de BuhoLegal. En total se cuenta con 5 endpoints los cuales se listan en la siguiente tabla:

<b>Endpoint</b>	<b>Método</b>	<b>Autenticación</b>
/apikey/	POST	Usuario y contraseña
/crear_usuario/	POST	Token (apikey)
/lista_usuarios/	GET	Token (apikey)
/inhabilitar_usuario/	PUT	Token (apikey)
/rehabilitar_usuario/	PUT	Token (apikey)
/estados/	GET	
/busqueda/	GET	Token (apikey)
/ultimas_busquedas/	GET	Token (apikey)
/cantidad_ultimas_busquedas/	GET	Token (apikey)
/busquedas_disponibles/	GET	Token (apikey)
/validar_curp/	GET	Token (apikey)
/cedula/	GET	

Para acceder a los endpoints “/busqueda/”, “/ultimas\_busquedas/” y “/busquedas\_disponibles/” es necesario consultar el API (endpoint “/apikey/”) para obtener un token que le permita

autenticarse. En la siguiente gráfica se muestra la secuencia de pasos a seguir para realizar una búsqueda (estos pasos son aplicables al resto de los endpoints).



**NOTA:** Una vez que se cuenta con un token almacenado localmente no es necesario volver a solicitarlo al API. Solo debe solicitar un nuevo token cuando se reciba un error de autenticación, lo cual indica que el token almacenado previamente ha caducado.



## Obtener token

El objetivo de esta funcionalidad es permitir a los usuarios obtener un token de manera segura y confidencial que les permita acceder a algunos de los endpoints del API de Buholegal.

**Endpoint:** /apikey/

**Metodo:** POST

**Parametros de entrada:**

Parametro	Descripción
username	Nombre de usuario de la cuenta Buholegal (página).
subusername	Nombre de usuario de una cuenta especial creada para desarrolladores vinculados a un cliente. Ver xxxxxx
password	Contraseña de la cuenta Buholegal (página) o de una cuenta especial creada por Buholegal para desarrolladores vinculados a un cliente

**Salida:**

Si las credenciales son correctas se obtiene como resultado un objeto JSON que contiene el valor del token y el usuario que hizo la solicitud. A continuación se muestra un ejemplo de resultado:

```
{  
  "token": "a5b4c3d2",  
  "user_ID": "usuario"  
}
```

### Manejo de errores:

Los posibles errores que pueden ocurrir al momento de hacer uso del endpoint son los siguientes:

Código	Descripción
401	Error de autenticación. Puede ocurrir al introducir usuario y/o contraseña incorrectos. También si se intenta acceder al endpoint sin proporcionar las credenciales de autenticación.
405	Método no permitido. Ocurre cuando se intenta acceder al endpoint utilizando otro método que no sea "POST".
500	Error interno del servidor. En caso de ocurrir este error, por favor ponerse en contacto con el equipo de Buholegal.

### Ejemplo de uso (Python):

```
import requests, json

credenciales = { 'username' : 'usuario', 'password' : 'contraseña' }
response = requests.post('https://api.buholegal.com/apikey/', data=credenciales)

if response.status_code == 200:
    token = response.json()['token']

# Guardar token para su uso posterior
```

## Abreviaturas para cada estado

El objetivo de esta funcionalidad es permitir a los usuarios conocer las abreviaturas que se utilizan para referirse a cada estado a la hora de realizar una búsqueda.

**Endpoint:** /estados/

**Método:** GET

**Parámetros de entrada:**

Sin parámetros de entrada

**Salida:**

El resultado es un arreglo de objetos JSON, cada uno de estos objetos especifica el nombre del estado y la abreviatura utilizada para referirse a éste. A continuación se muestra un ejemplo:

```
{  
  "entidad": "Aguascalientes",  
  "abreviatura": "AS"  
}
```

**Manejo de errores:**

Los posibles errores que pueden ocurrir al momento de hacer uso del endpoint son los siguientes:

Código	Descripción
405	Método no permitido. Ocurre cuando se intenta acceder al endpoint utilizando otro método que no sea "GET".
500	Error interno del servidor. En caso de ocurrir este error, por favor ponerse en contacto con el equipo de Buholegal.

### Ejemplo de uso (Python):

```
import requests, json

respose = requests.post("https://api.buholegal.com/estados/")

estados_json = respose.json()

# Procesar los estados
```

En la siguiente tabla se listan todos los estados y sus abreviaturas:

<b>Estado</b>	<b>Abreviatura</b>
Aguascalientes	AS
Baja California	BC
Baja California Sur	BS
Campeche	CC
Coahuila	CL
Colima	CM
Chiapas	CS
Chihuahua	CH
Ciudad de México	DF
Durango	DG
Guanajuato	GT
Guerrero	GR
Hidalgo	HG
Jalisco	JC
Estado de México	MC
Michoacán	MN

<b>Estado</b>	<b>Abreviatura</b>
Morelos	MS
Nayarit	NT
Nuevo León	NL
Oaxaca	OC
Puebla	PL
Querétaro	QT
Quintana Roo	QR
San Luis Potosí	SP
Sinaloa	SL
Sonora	SR
Tabasco	TC
Tamaulipas	TS
Tlaxcala	TL
Veracruz	VZ
Yucatán	YN
Zacatecas	ZS

## Realizar búsquedas judiciales

Funcionalidad principal del API. Permite realizar búsquedas de expedientes y acuerdos publicados por los diferentes juzgados y tribunales de la República Mexicana.

**Endpoint:** /busqueda/

**Método:** GET

**Parámetros de entrada:**

Parámetro	Descripción
token	Token que identifica al usuario y le permite autenticarse.
persona	Tipo de persona que se desea buscar (“física”, “moral”). <b>Es requerido</b>
nombre	Nombre de la persona física o moral. Es requerido si se trata de una persona moral o en caso de no especificar el apellido materno de una persona física.
paterno	Apellido paterno. Solo aplica a personas físicas, en cuyo caso es requerido.
materno	Apellido materno. Solo aplica a personas físicas, en cuyo caso es requerido sólo si no se especifica el nombre de la persona.
estado	Indica el estado en el cual se desea realizar la búsqueda. Ver endpoint “/estados/”. En caso de no indicar ningún estado se realizará una búsqueda a nivel nacional.
detalle	Sirve para determinar si se mostrarán los acuerdos de cada

	expediente. Toma valores “True” o “False”(valor por defecto).
fecha_inicio	Fecha inicial del rango de fechas sobre el que se desea realizar la búsqueda. Se tomarán en cuenta sólo los expedientes que contengan acuerdos con fecha igual o superior a la especificada por el parámetro. El formato aceptado es “dd-mm-AAAA”. Este parámetro <b>no es requerido</b> .
fecha_fin	Fecha final del rango de fechas sobre el que se desea realizar la búsqueda. Se tomarán en cuenta sólo los expedientes que contengan acuerdos con fecha igual o inferior a la especificada por el parámetro. El formato aceptado es “dd-mm-AAAA”. Este parámetro <b>no es requerido</b> .
curp	Parámetro opcional. Sirve para validar el curp proporcionado ante el Registro Nacional de Población.
validar_homonimia	Solo debe indicarse “ <b>False</b> ” en caso que no se desee obtener un valor estimado de homonimia
validar_cedula	Solo debe indicarse “ <b>False</b> ” en caso que no se desee obtener el listado de las cédulas relacionadas al nombre de la búsqueda.
validar_edictos	Solo debe indicarse “ <b>False</b> ” en caso que no se desee obtener el listado de los edictos en los que aparece el nombre de la persona.
validar_sat	Solo debe indicarse “ <b>False</b> ” en caso que no se desee obtener los resultados de las listas negras del SAT para la persona.

**NOTA:** De todos los parametros antes listados solo el token se incluye en la cabecera (header) de la consulta.

**Salida:**

Si el token es correcto se obtiene como resultado un objeto JSON que contiene 3 campos: “query”, “numero\_resultados” y “resultados”.

El campo “query” es un objeto JSON que contiene los parámetros y valores especificados en la búsqueda. El campo “numero\_resultados” es un valor entero con la cantidad de resultados obtenidos. Por último, el campo “resultados” es un arreglo de objetos JSON donde cada uno de estos objetos representa a un estado; conteniendo, además del nombre del estado, un arreglo de expedientes. A continuación se muestran algunos ejemplos de resultados:

Ejemplo de búsqueda estatal sin detalle.

```
{
  "query":{
    "nombre":"juan",
    "paterno":"rodriguez",
    "materno":"jimenez",
    "persona":"fisica",
    "estado":"AS",
    "fecha_inicio":"2020-01-01",
    "fecha_fin":null,
    "detalle":false
  },
  "numero_resultados":1,
  "resultados":[
    {
      "entidad":"Aguascalientes",
      "expedientes":[
        {
          "expediente":"00000/2896",
          "actor":"JUAN ROMAN RODRIGUEZ JIMENEZ",
          "demandado":"",
          "entidad":"Aguascalientes",
          "juzgado":"Aguascalientes - Juzgado de Oralidad",
          "tipo":"CARPETA",
          "fuero":"Común",
          "fecha":"08-01-2896",
          "acuerdos":[]
        }
      ]
    }
  ]
}
```

Donde los campos “**validacion\_curp**” y “**cedula\_profesional**” tienen el siguiente formato:

```
"validacion_curp": {
  "curp": "ASDF801332GYBQL00",
  "paterno": "RODRIGUEZ",
  "materno": "JIMENEZ",
  "nombre": "JUAN",
  "sexo": "HOMBRE",
  "fecha_nacimiento": "32/13/1980",
  "entidad_nacimiento": "OAXACA"
}
```

```
"cedula_profesional": [
  {
    "cedula": 55555,
    "nombre": "JUAN RODRÍGUEZ JIMÉNEZ",
    "carrera": "LICENCIATURA COMO MÉDICO CIRUJANO",
    "anho": 1973
  },
  {
    "cedula": 88789,
    "nombre": "JUAN RODRÍGUEZ JIMÉNEZ",
    "carrera": "PROFESOR EN EDUCACIÓN PRIMARIA",
    "anho": 1945
  }
]
```

Ejemplo de búsqueda estatal con detalle.

```
{
  "query":{
    "nombre":"juan",
    "paterno":"rodriguez",
    "materno":"jimenez",
    "persona":"fisica",
    "estado":"AS",
    "fecha_inicio":"2020-01-01",
    "fecha_fin":null,
    "detalle":false
  },
  "numero_resultados":1,
  "cedula_profesional": [ {...}, {...} ],
  "validacion_curp": {...},
  "resultados":[
    {
      "entidad":"Aguascalientes",
      "expedientes":[
        {
          "expediente":"00000/2896",
          "actor":"JUAN ROMAN RODRIGUEZ JIMENEZ",
          "demandado":"",
          "entidad":"Aguascalientes",
          "juzgado":"Aguascalientes - Juzgado de Oralidad",
          "tipo":"CARPETA",
          "fuero":"Común",
          "fecha":"08-01-2896",
          "acuerdos":[
            {"fecha": "08-01-2896", "acuerdo": "texto del acuerdo 1"},
            {"fecha": "08-02-2896", "acuerdo": "texto del acuerdo 2"},
          ]
        }
      ]
    }
  ]
}
```

Ejemplo de búsqueda nacional.

```
{
  "query":{
    ...,
    "estado":"nacional"
  },
  "numero_resultados":10,
  "cedula_profesional": [ {...}, {...} ],
  "validacion_curp": {...},
  "resultados":[
    {
      "entidad":"Aguascalientes",
      "expedientes":[
        {
          "expediente":"00000/2896",
          "actor":"JUAN ROMAN RODRIGUEZ JIMENEZ",
          "demandado":"",
          "entidad":"Aguascalientes",
          "juzgado":"Aguascalientes - Juzgado de Oralidad",
          "tipo":"CARPETA",
          "fuero":"Común",
          "fecha":"08-01-2896",
          "acuerdos":[]
        }
      ]
    },
    {
      "entidad":"Puebla",
      "expedientes":[ ... ]
    },
    {
      "entidad":"Oaxaca",
      "expedientes":[ ... ]
    }
  ]
}
```

### Manejo de errores:

Los posibles errores que pueden ocurrir al momento de hacer uso del endpoint son los siguientes:

Código	Descripción
400	Formato incorrecto de los parámetros. Ocurre si se introduce un valor incorrecto en alguno de los parámetros proporcionados o si se omite algún parámetro obligatorio.
401	Error de autenticación. Puede ocurrir al introducir un token incorrecto. También si se intenta acceder al endpoint sin proporcionar las credenciales de autenticación.
405	Método no permitido. Ocurre cuando se intenta acceder al endpoint utilizando otro método que no sea "GET".
500	Error interno del servidor. En caso de ocurrir este error, por favor ponerse en contacto con el equipo de Buholegal.

### Ejemplo de uso (Python):

```
import requests, json

# Se asume que se tiene guardado en la variable token el apikey del usuario

#Realizando la búsqueda.
headers = { 'Authorization' : 'Token ' + token }
PARAMS = {}
PARAMS['nombre'] = 'juan'
PARAMS['paterno'] = 'mengano'
PARAMS['materno'] = 'sultano'
PARAMS['persona'] = 'fisica'
PARAMS['estado'] = 'AS' # AS se refiere a Aguascalientes.
PARAMS['detalle'] = True
PARAMS['fecha_inicio'] = '01-01-2019'
PARAMS['fecha_fin'] = '01-06-2020'
PARAMS[curp] = 'ASDF801332GYBQL00'

response = requests.get('https://api.buholegal.com/busqueda/', headers=headers, params =
PARAMS)

resultado = response.json()

# Procesar el resultado
```

## Realizar búsquedas judiciales utilizando solo el CURP

Funcionalidad principal del API. Permite realizar búsquedas de expedientes y acuerdos publicados por los diferentes juzgados y tribunales de la República Mexicana.

**Endpoint:** /busqueda/

**Método:** GET

**Parámetros de entrada:**

Parámetro	Descripción
token	Token que identifica al usuario y le permite autenticarse.
curp	Parámetro obligatorio. Sirve para validar el curp proporcionado ante el Registro Nacional de Población. Con la información obtenida se realizará la búsqueda judicial.
estado	Indica el estado en el cual se desea realizar la búsqueda. Ver endpoint “/estados/”. En caso de no indicar ningún estado se realizará una búsqueda a nivel nacional.
detalle	Sirve para determinar si se mostrarán los acuerdos de cada expediente. Toma valores “True” o “False”(valor por defecto).
fecha_inicio	Fecha inicial del rango de fechas sobre el que se desea realizar la búsqueda. Se tomarán en cuenta sólo los expedientes que contengan acuerdos con fecha igual o superior a la especificada por el parámetro. El formato aceptado es “dd-mm-AAAA”. Este parámetro <b>no es requerido</b> .
fecha_fin	Fecha final del rango de fechas sobre el que se desea realizar la

	búsqueda. Se tomarán en cuenta sólo los expedientes que contengan acuerdos con fecha igual o inferior a la especificada por el parámetro. El formato aceptado es “dd-mm-AAAA”. Este parámetro <b>no es requerido</b> .
--	--

**NOTA:** De todos los parámetros antes listados solo el token se incluye en la cabecera (header) de la consulta.

**Salida:**

Si el token es correcto se obtiene como resultado un objeto JSON que contiene 3 campos: “query”, “numero\_resultados” y “resultados”.

El campo “query” es un objeto JSON que contiene los parámetros y valores especificados en la búsqueda. El campo “numero\_resultados” es un valor entero con la cantidad de resultados obtenidos. Por último, el campo “resultados” es un arreglo de objetos JSON donde cada uno de estos objetos representa a un estado; conteniendo, además del nombre del estado, un arreglo de expedientes. A continuación se muestran algunos ejemplos de resultados:

Ejemplo de resultados de una búsqueda estatal sin detalle.

```
{
  "query": {
    "nombre": "JUAN",
    "paterno": "PEREZ",
    "materno": "PEREZ",
    "persona": "fisica",
    "curp": "ASDF801332GYBHQ00",
    "estado": "nacional",
    "fecha_inicio": null,
    "fecha_fin": null,
    "detalle": false
  },
  "numero_resultados": 1,
  "resultados": [
    {
      "entidad": "Veracruz",
      "expedientes": [
        {
          "expediente": "0000/2099",
          "actor": "KARELI GOMEZ ACOSTA/KARELI GOMEZ AGOS",
          "demandado": "JUAN PEREZ PEREZ",
          "entidad": "Veracruz",
          "juzgado": "Juzgado Primera Instancia",
          "tipo": "EXPEDIENTE",
          "fuero": "Común",
          "fecha": "14-11-2099",
          "acuerdos": []
        }
      ]
    }
  ],
  "homonimia": "1",
  "validacion_curp": {...},
  "cedula_profesional": [{...}, {...}],
  "edictos": {...},
  "situacion_SAT": {...}
}
```

Donde los campos “**validacion\_curp**”, “**cedula\_profesional**” y “**situacion\_SAT**” tienen el siguiente formato:

```
"validacion_curp": {
  "curp": "ASDF801332GYBQL00",
  "paterno": "RODRIGUEZ",
  "materno": "JIMENEZ",
  "nombre": "JUAN",
  "sexo": "HOMBRE",
  "fecha_nacimiento": "32/13/1980",
  "entidad_nacimiento": "OAXACA"
}
```

```
"cedula_profesional": [
  {
    "cedula": 55555,
    "nombre": "JUAN RODRÍGUEZ JIMÉNEZ",
    "carrera": "LICENCIATURA COMO MÉDICO CIRUJANO",
    "anho": 1973
  },
  {
    "cedula": 88789,
    "nombre": "JUAN RODRÍGUEZ JIMÉNEZ",
    "carrera": "PROFESOR EN EDUCACIÓN PRIMARIA",
    "anho": 1945
  }
]
```

```
"situacion_SAT": {  
  "criterio": "JUAN PEREZ PEREZ",  
  "total": 0,  
  "efos": {  
    "cantidad": 0,  
    "resultados": []  
  },  
  "incumplidos": {  
    "cantidad": 0,  
    "resultados": []  
  }  
}
```

Ejemplo de búsqueda estatal con detalle.

```
{
  "query":{
    "nombre":"juan",
    "paterno":"rodriguez",
    "materno":"jimenez",
    "persona":"fisica",
    "estado":"AS",
    "fecha_inicio":"2020-01-01",
    "fecha_fin":null,
    "detalle":false
  },
  "numero_resultados":1,
  "cedula_profesional": [ {...}, {...} ],
  "validacion_curp": {...},
  "resultados":[
    {
      "entidad":"Aguascalientes",
      "expedientes":[
        {
          "expediente":"00000/2896",
          "actor":"JUAN ROMAN RODRIGUEZ JIMENEZ",
          "demandado":"",
          "entidad":"Aguascalientes",
          "juzgado":"Aguascalientes - Juzgado de Oralidad",
          "tipo":"CARPETA",
          "fuero":"Común",
          "fecha":"08-01-2896",
          "acuerdos":[
            {"fecha": "08-01-2896", "acuerdo": "texto del acuerdo 1"},
            {"fecha": "08-02-2896", "acuerdo": "texto del acuerdo 2"},
          ]
        }
      ]
    }
  ]
}
```

Ejemplo de búsqueda nacional.

```
{
  "query":{
    ...,
    "estado":"nacional"
  },
  "numero_resultados":10,
  "cedula_profesional": [ {...}, {...} ],
  "validacion_curp": {...},
  "resultados":[
    {
      "entidad":"Aguascalientes",
      "expedientes":[
        {
          "expediente":"00000/2896",
          "actor":"JUAN ROMAN RODRIGUEZ JIMENEZ",
          "demandado":"",
          "entidad":"Aguascalientes",
          "juzgado":"Aguascalientes - Juzgado de Oralidad",
          "tipo":"CARPETA",
          "fuero":"Común",
          "fecha":"08-01-2896",
          "acuerdos":[]
        }
      ]
    },
    {
      "entidad":"Puebla",
      "expedientes":[ ... ]
    },
    {
      "entidad":"Oaxaca",
      "expedientes":[ ... ]
    }
  ]
}
```

### Manejo de errores:

Los posibles errores que pueden ocurrir al momento de hacer uso del endpoint son los siguientes:

Código	Descripción
400	Formato incorrecto de los parámetros. Ocurre si se introduce un valor incorrecto en alguno de los parámetros proporcionados o si se omite algún parámetro obligatorio.
401	Error de autenticación. Puede ocurrir al introducir un token incorrecto. También si se intenta acceder al endpoint sin proporcionar las credenciales de autenticación.
405	Método no permitido. Ocurre cuando se intenta acceder al endpoint utilizando otro método que no sea "GET".
500	Error interno del servidor. En caso de ocurrir este error, por favor ponerse en contacto con el equipo de Buholegal.

### Ejemplo de uso (Python):

```
import requests, json

# Se asume que se tiene guardado en la variable token el apikey del usuario

#Realizando la búsqueda.
headers = { 'Authorization' : 'Token ' + token }
PARAMS = {}
PARAMS['nombre'] = 'juan'
PARAMS['paterno'] = 'mengano'
PARAMS['materno'] = 'sultano'
PARAMS['persona'] = 'fisica'
PARAMS['estado'] = 'AS' # AS se refiere a Aguascalientes.
PARAMS['detalle'] = True
PARAMS['fecha_inicio'] = '01-01-2019'
PARAMS['fecha_fin'] = '01-06-2020'
PARAMS[curp] = 'ASDF801332GYBQL00'

response = requests.get('https://api.buholegal.com/busqueda/', headers=headers, params =
PARAMS)

resultado = response.json()

# Procesar el resultado
```



## Consultar búsquedas realizadas

Esta funcionalidad le sirve al usuario para llevar un control de las búsquedas realizadas.

**Endpoint:** /ultimas\_búsquedas/

**Metodo:** GET

**Parametros de entrada:**

Parametro	Descripción
token	Token que identifica al usuario y le permite autenticarse.
fecha_inicio	Fecha inicial del rango de fechas sobre el que se desea realizar la búsqueda. Se tomarán en cuenta sólo las búsquedas realizadas en una fecha igual o superior a la especificada por el parámetro. El formato aceptado es “dd-mm-AAAA”. Este parámetro <b>no es requerido</b> .
fecha_fin	Fecha final del rango de fechas sobre el que se desea realizar la búsqueda. Se tomarán en cuenta sólo las búsquedas realizadas en una fecha igual o inferior a la especificada por el parámetro. El formato aceptado es “dd-mm-AAAA”. Este parámetro <b>no es requerido</b> .

**NOTA:** De todos los parametros antes listados solo el token se incluye en la cabecera (header) de la consulta.

**NOTA:** En caso de no especificarse los parámetros “fecha\_inicio” y “fecha\_fin” se retornará el historial completo de búsquedas del usuario.

**NOTA:** El día especificado en el parámetro “fecha\_fin” no se incluye en la consulta realizada. **Ejemplo** si se quisiera consultar las búsquedas realizadas en el mes de marzo se debe especificar el rango del “01-03-2020” al “01-04-2020”.

**Salida:**

Si el token es correcto se obtiene como resultado un objeto JSON con los campos “total\_realizadas” y “busquedas”.

El campo “total\_realizadas” es un número entero con el total de búsquedas realizadas en el rango de fecha especificado en la búsqueda; mientras que el campo “busquedas” es un arreglo de objetos JSON donde cada uno especifica el criterio de búsqueda utilizado, el estado sobre el cual se realizó la búsqueda, la fecha es que fue realizada la búsqueda y el usuario que la realizó. A continuación se muestra un ejemplo de resultado:

```
{
  "total_realizadas": 1,
  "busquedas": [
    {
      "criterio": "juan rodriguez jimenez",
      "estado": "Aguascalientes",
      "fecha": "2020-05-01T16:26:23.563Z",
      "usuario": "usuario"
    }
  ]
}
```

### Manejo de errores:

Los posibles errores que pueden ocurrir al momento de hacer uso del endpoint son los siguientes:

Código	Descripción
400	Formato incorrecto de los parámetros. Ocurre si se el formato para los parámetros de fecha_inicio y/o fecha_fin son incorrectos (no cumplen con el formato dd-mm-AAAA).
401	Error de autenticación. Puede ocurrir al introducir un token incorrecto. También si se intenta acceder al endpoint sin proporcionar las credenciales de autenticación.
405	Método no permitido. Ocurre cuando se intenta acceder al endpoint utilizando otro método que no sea "GET".
500	Error interno del servidor. En caso de ocurrir este error, por favor ponerse en contacto con el equipo de Buholegal.

### Ejemplo de uso (Python):

```
import requests, json

# Se asume que se tiene guardado en la variable token el apikey del usuario

#Realizando la búsqueda.
headers = { 'Authorization' : 'Token ' + token }
PARAMS = {}
PARAMS['fecha_inicio'] = '01-01-2019'
PARAMS['fecha_fin'] = '01-06-2020'

response = requests.get('https://api.buholegal.com/ultimas_busquedas/', headers=headers,
params = PARAMS)

resultado = response.json()

# Procesar el resultado
```

## Consultar cantidad de búsquedas realizadas

Esta funcionalidad le sirve al usuario para llevar un control del total de búsquedas realizadas.

**Endpoint:** /cantidad\_ultimas\_búsquedas/

**Metodo:** GET

**Parametros de entrada:**

Parametro	Descripción
token	Token que identifica al usuario y le permite autenticarse.
fecha_inicio	Fecha inicial del rango de fechas sobre el que se desea realizar la búsqueda. Se tomarán en cuenta sólo las búsquedas realizadas en una fecha igual o superior a la especificada por el parámetro. El formato aceptado es “dd-mm-AAAA”. Este parámetro <b>no es requerido</b> .
fecha_fin	Fecha final del rango de fechas sobre el que se desea realizar la búsqueda. Se tomarán en cuenta sólo las búsquedas realizadas en una fecha igual o inferior a la especificada por el parámetro. El formato aceptado es “dd-mm-AAAA”. Este parámetro <b>no es requerido</b> .

**NOTA:** De todos los parametros antes listados solo el token se incluye en la cabecera (header) de la consulta.

**NOTA:** En caso de no especificarse los parámetros “fecha\_inicio” y “fecha\_fin” se retornará el historial completo de búsquedas del usuario.

**NOTA:** El día especificado en el parámetro “fecha\_fin” no se incluye en la consulta realizada. **Ejemplo** si se quisiera consultar las búsquedas realizadas en el mes de marzo se debe especificar el rango del “01-03-2020” al “01-04-2020”.

### Salida:

Si el token es correcto se obtiene como resultado un objeto JSON con el campo “total\_realizadas”.

El campo “total\_realizadas” es un número entero con el total de búsquedas realizadas en el rango de fecha especificado en la búsqueda. A continuación se muestra un ejemplo de resultado:

```
{  
  "total_realizadas": 1  
}
```

### Manejo de errores:

Los posibles errores que pueden ocurrir al momento de hacer uso del endpoint son los siguientes:

Código	Descripción
400	Formato incorrecto de los parámetros. Ocurre si se el formato para los parámetros de fecha_inicio y/o fecha_fin son incorrectos (no cumplen con el formato dd-mm-AAAA).
401	Error de autenticación. Puede ocurrir al introducir un token incorrecto. También si se intenta acceder al endpoint sin proporcionar las credenciales de autenticación.
405	Método no permitido. Ocurre cuando se intenta acceder al endpoint utilizando otro método que no sea “GET”.
500	Error interno del servidor. En caso de ocurrir este error, por favor ponerse en contacto con el equipo de Buholegal.

## Ejemplo de uso (Python):

```
import requests, json

# Se asume que se tiene guardado en la variable token el apikey del usuario

#Realizando la búsqueda.
headers = { 'Authorization' : 'Token ' + token }
PARAMS = {}
PARAMS['fecha_inicio'] = '01-01-2019'
PARAMS['fecha_fin'] = '01-06-2020'

response = requests.get('https://api.buholegal.com/ultimas_búsquedas/', headers=headers,
params = PARAMS)

resultado = response.json()

# Procesar el resultado
```

## Consultar búsquedas en la canasta (solo para clientes prepago)

Le permite conocer al usuario la cantidad de búsquedas nacionales y estatales con las que cuenta en su canasta.

**Endpoint:** /búsquedas\_disponibles/

**Metodo:** GET

**Parametros de entrada:**

Parametro	Descripción
token	Token que identifica al usuario y le permite autenticarse.

**NOTA:** El token se incluye en la cabecera (header) de la consulta.

**Salida:**

Si el token es correcto se obtiene como resultado un objeto JSON en el que se especifica la cantidad de búsquedas nacionales y estatales disponibles en la canasta del usuario. A continuación se muestra un ejemplo de resultado:

```
{
  "nacional_disponibles" : 10,
  "estatal_disponibles" : 10
}
```

### Manejo de errores:

Los posibles errores que pueden ocurrir al momento de hacer uso del endpoint son los siguientes:

Código	Descripción
401	Error de autenticación. Puede ocurrir al introducir un token incorrecto. También si se intenta acceder al endpoint sin proporcionar las credenciales de autenticación.
405	Método no permitido. Ocurre cuando se intenta acceder al endpoint utilizando otro método que no sea "GET".
500	Error interno del servidor. En caso de ocurrir este error, por favor ponerse en contacto con el equipo de Buholegal.

### Ejemplo de uso (Python):

```
import requests, json

# Se asume que se tiene guardado en la variable token el apikey del usuario

#Realizando la búsqueda.
headers = { 'Authorization' : 'Token ' + token }

respose = requests.get('https://api.buholegal.com/busquedas_disponibles/', headers=headers)

resultado = respose.json()

# Procesar el resultado
```

## Validar CURP

Le permite al usuario validar que el CURP ingresado se encuentre registrado en el Registro Nacional de Población. Además, con los datos obtenidos de dicha validación se calcula el RFC de la persona y se verifica que esté registrado ante el SAT, así como si aparece en las listas negras del SAT

**Endpoint:** /validar\_curp/

**Metodo:** GET

**Parametros de entrada:**

Parametro	Descripción
token	Token que identifica al usuario y le permite autenticarse.
curp	CURP que se desee consultar

**NOTA:** El token se incluye en la cabecera (header) de la consulta.

**Salida:**

Si el token es correcto se obtiene como resultado un objeto JSON que contiene los datos de la persona a la que pertenece el CURP, el RFC calculado y si el mismo es válido o no, situación ante el SAT( coincidencias en las listas negras) y un listado de las cédulas profesionales que coinciden con el nombre de la persona.. A continuación se muestra un ejemplo de resultado:

```

{
  "validacion_curp": {
    "nombre": "JUAN",
    "paterno": "MENGANO",
    "materno": "SULTANO",
    "fecha_nacimiento": "29/10/1971",
    "sexo": "HOMBRE",
    "entidad_nacimiento": "PUEBLA",
    "curp": "JUAN711029HPBRNS05"
  },
  "validacion_rfc": {
    "rfc": "JUAN711029CK8",
    "status": "valido"
  },
  "situacion_sat": {
    "criterio": "JUAN711029CK8",
    "total": 2,
    "efos": {
      "cantidad": 1,
      "resultados": [
        {
          "rfc": "JUAN711029CK8",
          "situacion": "Definitivo",
          "razon_social": "MENGANO SULTANO JUAN",
          "registros": [
            {
              "situacion": "Presunto",
              "fecha_dof": "2016-11-16T00:00:00",
              "fecha_oficio": "2016-11-01T00:00:00",
              "num_oficio": "500-05-2016-38655",
              "fecha_sat": "2016-11-01T00:00:00"
            }
          ]
        }
      ],
      "nota": ""
    }
  ]
},
  "incumplidos": {
    "cantidad": 1,
    "resultados": [

```

```

    {
      "rfc": "JUAN711029CK8",
      "registros": [
        {
          "primera_fecha": "2018-07-01T00:00:00",
          "fecha": "2018-07-01T00:00:00",
          "motivo": "",
          "monto": 6160,
          "entidad": "PUEBLA",
          "supueso": "CANCELADOS"
        }
      ],
      "razon_social": "JUAN SULTANO MENGANO",
      "tipo_persona": "FISICA"
    }
  ],
  "cedulas_profesionales": [
    {
      "cedula": 00000000,
      "nombre": "JUAN SULTANO MENGANO",
      "carrera": "LICENCIATURA EN CIENCIAS",
      "anho": 2030
    }
  ]
}

```

### Manejo de errores:

Los posibles errores que pueden ocurrir al momento de hacer uso del endpoint son los siguientes:

Código	Descripción
400	Formato incorrecto de los parámetros. Ocurre si el formato del CURP proporcionado no es correcto.
401	Error de autenticación. Puede ocurrir al introducir un token incorrecto. También si se intenta acceder al endpoint sin proporcionar las credenciales de autenticación.
405	Método no permitido. Ocurre cuando se intenta acceder al endpoint utilizando otro método que no sea "GET".
500	Error interno del servidor. En caso de ocurrir este error, por favor ponerse en contacto con el equipo de Buholegal.

### Ejemplo de uso (Python):

```
import requests, json

# Se asume que se tiene guardado en la variable token el apikey del usuario

#Realizando la búsqueda.
headers = { 'Authorization' : 'Token ' + token }

PARAMS = { curp : QWER123456HNETKU00 ' }
response = requests.get('https://api.buholegal.com/validar_curp/', headers=headers, params =
PARAMS)

resultado = response.json()

# Procesar el resultado
```

## Consulta de Homonimia

Le permite al usuario conocer la homonimia de un nombre dado (cantidad de personas con el mismo nombre) .

**Endpoint:** /homonimia/

**Metodo:** GET

**Parametros de entrada:**

Parametro	Descripción
nombre	Nombre(s) de la persona.
paterno	Apellido paterno de la persona.
materno	Apellido materno de la persona.
curp	El CURP de una persona. Este valor se verificará con el registro nacional de población y se realizará la consulta con los datos obtenidos.
estado	Entidad federativa donde se desea realizar la búsqueda.

**NOTA:** Si el CURP es proporcionado el resto de los campos no son necesarios. En caso de proporcionar el CURP es obligatorio el nombre y el apellido paterno. Si el estado se omite se realizará la consulta para toda la república.

**Salida:**

El resultado de la consulta es un JSON con los nombres y apellidos consultados, estado donde se realizó la consulta y el valor de la homonimia obtenido.

```
{
  "nombre": "juan",
  "paterno": "sultano",
  "materno": "mengano",
  "estado": "Estados Unidos Mexicanos",
  "homonimia": 100
}
```

**Manejo de errores:**

Los posibles errores que pueden ocurrir al momento de hacer uso del endpoint son los siguientes:

Código	Descripción
400	Formato incorrecto de los parámetros. Ocurre si todos los campos de la consulta están vacíos.
405	Método no permitido. Ocurre cuando se intenta acceder al endpoint utilizando otro método que no sea "GET".
500	Error interno del servidor. En caso de ocurrir este error, por favor ponerse en contacto con el equipo de Buholegal.

## Ejemplo de uso (Python):

```
import requests, json

# Se asume que se tiene guardado en la variable token el apikey del usuario

#Realizando la búsqueda.
headers = { 'Authorization' : 'Token ' + token }

PARAMS = {'nombre' : 'Juan', paterno : 'Sultano', materno : 'Mengano'}
respose = requests.get('https://api.buholegal.com/homonimia/', params = PARAMS)

resultado = respose.json()

# Procesar el resultado
```

## Listas negras del SAT

Le permite al usuario conocer la homonimia de un nombre dado (cantidad de personas con el mismo nombre) .

**Endpoint:** /listas\_negras\_sat/

**Metodo:** GET

**Parametros de entrada:**

Parametro	Descripción
rfc	RFC de la persona (moral/fisica) que se desea buscar en las listas del SAT.
razon_social	Razón social de la persona (moral/fisica) que se desea buscar en las listas del SAT.

**NOTA:** Uno de los parámetros debe ser proporcionado.

**Salida:**

El resultado de la consulta es un objeto JSON con el criterio de búsqueda utilizado (RFC o razon social), el total de coincidencias y un listado con los resultados. A continuacion un ejemplo de salida:

```

{
  "criterio": "JUAN711029CK8",
  "total": 2,
  "efos": {
    "cantidad": 1,
    "resultados": [
      {
        "rfc": "JUAN711029CK8",
        "situacion": "Definitivo",
        "razon_social": "MENGANO SULTANO JUAN",
        "registros": [
          {
            "situacion": "Presunto",
            "fecha_dof": "2016-11-16T00:00:00",
            "fecha_oficio": "2016-11-01T00:00:00",
            "num_oficio": "500-05-2016-38655",
            "fecha_sat": "2016-11-01T00:00:00"
          }
        ],
        "nota": ""
      }
    ]
  },
  "incumplidos": {
    "cantidad": 1,
    "resultados": [
      {
        "rfc": "JUAN711029CK8",
        "registros": [
          {
            "primera_fecha": "2018-07-01T00:00:00",
            "fecha": "2018-07-01T00:00:00",
            "motivo": "",
            "monto": 6160,
            "entidad": "PUEBLA",
            "supueso": "CANCELADOS"
          }
        ],
        "razon_social": "JUAN SULTANO MENGANO",
        "tipo_persona": "FISICA"
      }
    ]
  }
}

```

```
}  
    ]  
}
```

### Manejo de errores:

Los posibles errores que pueden ocurrir al momento de hacer uso del endpoint son los siguientes:

Código	Descripción
400	Formato incorrecto de los parámetros. Ocurre si todos los campos de la consulta están vacíos.
405	Método no permitido. Ocurre cuando se intenta acceder al endpoint utilizando otro método que no sea "GET".
500	Error interno del servidor. En caso de ocurrir este error, por favor ponerse en contacto con el equipo de Buholegal.

### Ejemplo de uso (Python):

```
import requests, json  
  
# Se asume que se tiene guardado en la variable token el apikey del usuario  
  
#Realizando la búsqueda.  
headers = { 'Authorization' : 'Token ' + token }  
  
PARAMS = {rfc : 'JUAN711029CK8'}  
response = requests.get('https://api.buholegal.com/listas_negras_sat/', params = PARAMS)  
  
resultado = response.json()  
  
# Procesar el resultado
```

## Edictos Federales

Le permite al usuario conocer la homonimia de un nombre dado (cantidad de personas con el mismo nombre) .

**Endpoint:** /edictos/

**Metodo:** GET

**Parametros de entrada:**

Parametro	Descripción
query_text	Criterio de búsqueda a utilizar ( <b>obligatorio</b> ).
start_date	Fecha a partir de la cual se desean obtener resultados.
end_date	Fecha maxima hasta la cual se desean obtener resultados.

**Salida:**

El resultado de la consulta es un objeto JSON con el total de coincidencias y un listado con los resultados. A continuacion un ejemplo de salida:

```
{
  "total": 1,
  "edictos": [
    {
      "codNota": 00000,
      "fecha": "11-05-2022",
      "codDiario": 11111,
      "codSeccion": "UNICA",
      "pagina": 222,
      "cadenaContenido": "texto del edicto en formato HTML"
    }
  ]
}
```

```
]
}
```

### Manejo de errores:

Los posibles errores que pueden ocurrir al momento de hacer uso del endpoint son los siguientes:

Código	Descripción
400	Formato incorrecto de los parámetros. Ocurre si todos los campos de la consulta están vacíos o en un formato incorrecto.
405	Método no permitido. Ocurre cuando se intenta acceder al endpoint utilizando otro método que no sea "GET".
500	Error interno del servidor. En caso de ocurrir este error, por favor ponerse en contacto con el equipo de Buholegal.

### Ejemplo de uso (Python):

```
import requests, json

# Se asume que se tiene guardado en la variable token el apikey del usuario

#Realizando la búsqueda.
headers = { 'Authorization' : 'Token ' + token }

PARAMS = {query_text : 'JUAN SULTANO'}
response = requests.get('https://api.buholegal.com/edictos/', params = PARAMS)

resultado = response.json()

# Procesar el resultado
```

## Ambiente de pruebas

El API de Buholegal cuenta con un ambiente controlado para la realización de consultas de pruebas. Cada endpoint descrito en el presente documento tiene su par en el ambiente de pruebas, solo que le antecede el prefijo “/test”. En la siguiente tabla se tiene el listado de los endpoints del API y su par en el ambiente de prueba.

<b>API Endpoint</b>	<b>API Test Endpoint</b>
/apikey/	/test/apikey/
/estados/	/test/estados/
/busqueda/	/test/busqueda/
/ultimas_busquedas/	/test/ultimas_busquedas/
/cantidad_ultimas_busquedas/	/test/cantidad_ultimas_busquedas/
/busquedas_disponibles/	/test/busquedas_disponibles/

**NOTA:** Para utilizar los endpoints del ambiente de prueba es necesario obtener primeramente un token de prueba mediante el uso del endpoint “/test/apikey/”.

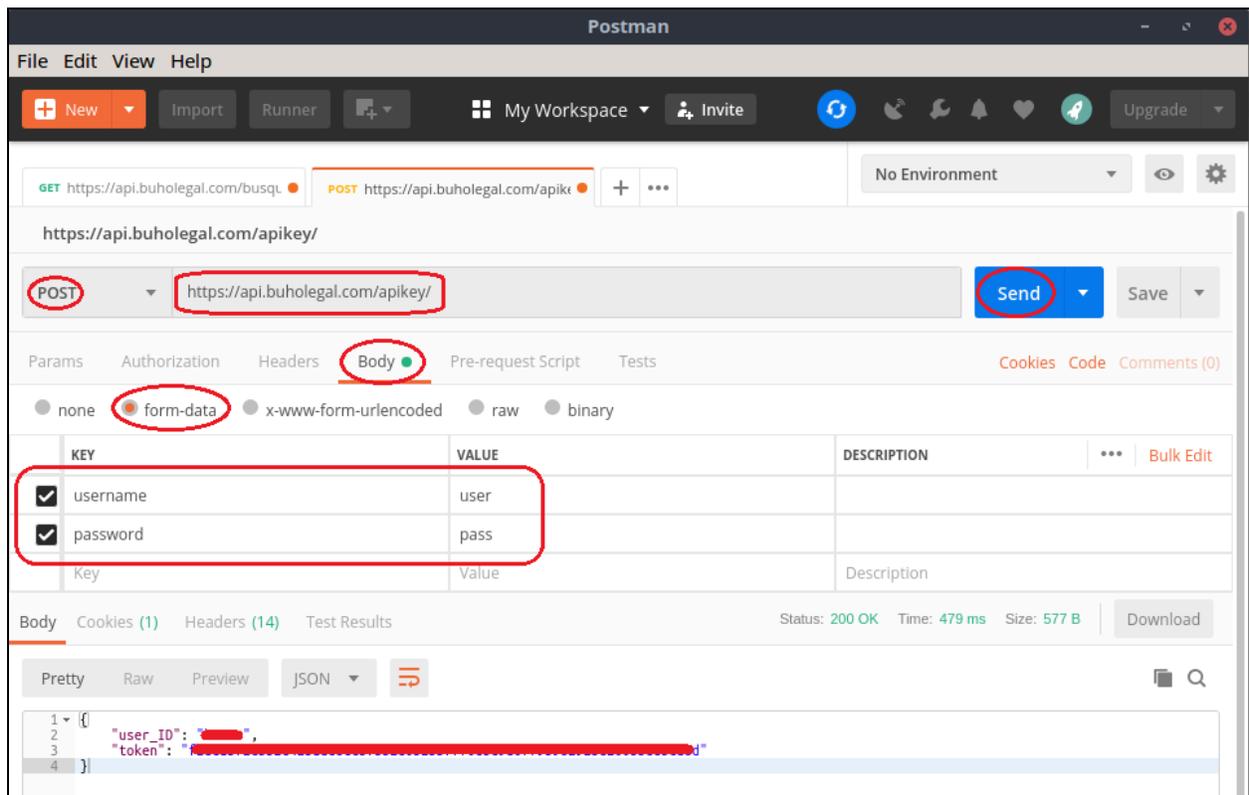
**NOTA:** Cada endpoint del ambiente de pruebas recibe los mismo parámetros y devuelven los resultados en el mismo formato que sus pares del API.

## Uso de la herramienta Postman

Para la realización de consultas al API, sin necesidad de utilizar un lenguaje de programación, puede utilizarse la herramienta Postman(<https://www.postman.com/api-documentation-tool/>), la cual es una herramienta para la realización de pruebas (testing) a APIs.

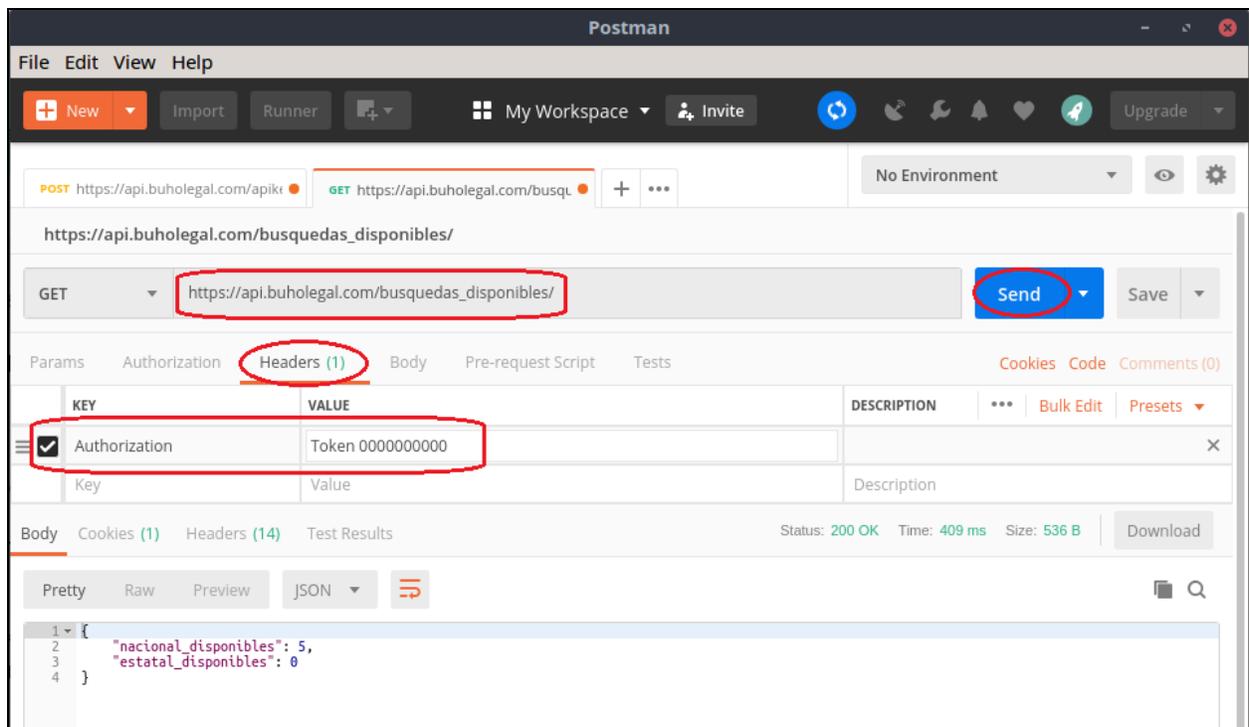
### Obtener token

Para obtener el token haciendo uso de Postman debe cambiar en la herramienta los campos que aparecen señalados en la siguiente imagen. Una vez tenga todos los valores colocados haga clic en el botón “Send”



### Uso del token en los endpoints

Si ya obtuvo el token ahora podrá hacer uso de este para acceder a otros endpoints. En la siguiente imagen se señalan los campos que debe modificar en Postman. Una vez tenga todos los valores colocados haga clic en el botón “Send”



Nótese que no se modificó el método a utilizar pues Postman por defecto utiliza “GET”. El valor del parámetro “Authorization” es “Token {token del usuario}”