

**Nombre del proyecto:** Meli (Acortador de URL)

## Tutorial de Caso de prueba

**ID de caso de prueba:** [000001](#)

**Prueba diseñada por:** Jairo Alexi Mendoza

**Nombre del módulo:** Acortador

**Prueba ejecutada por:** Jairo Alexi Mendoza

**Título de la prueba:** Verificar que la App Meli, responde adecuadamente desde Postman y en aplicativo creado desde net core con Patron de Diseño MVC.

**Fecha de ejecución de la prueba:** 28/04/2021

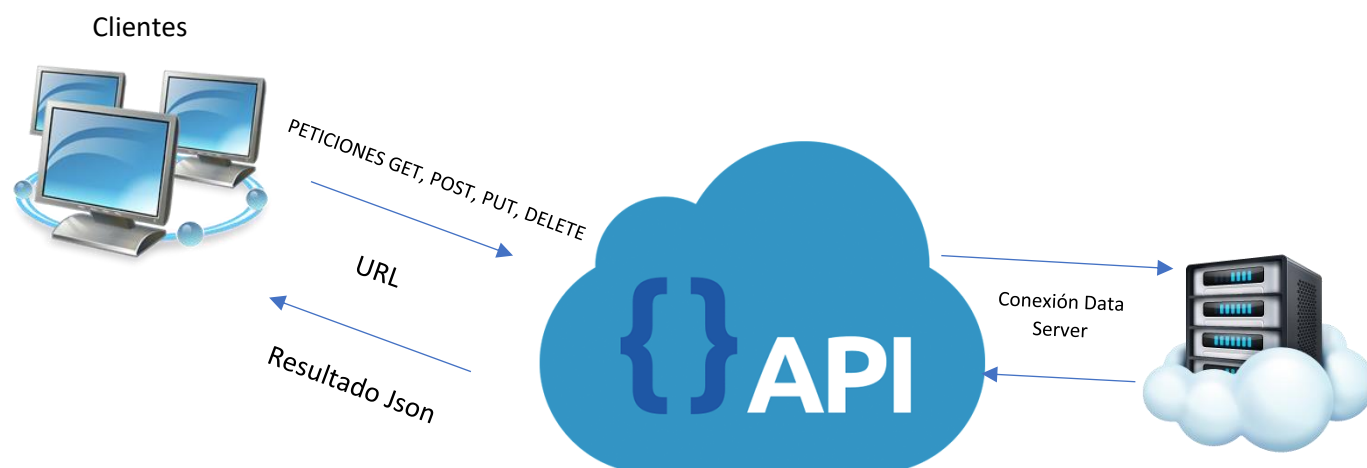
**Descripción:** Crear peticiones de creación, listado y eliminación de url acortadas, al ingresar una url acortada en el navegador, el aplicativo meli debe redireccionar a la url original.

**Condiciones previas:** Se debe tener disponible los siguientes elementos.

- Proyecto Meli desarrollado en Net Core
- ISS (Internet Information Server)
- Visual Studio 2019
- Sql Server
- Postman (Para pruebas de funcionamiento)
- Navegador Web

**Dependencias:**

Diagrama general de la Api:



Pasos:

## 1. CREACIÓN DE URL CORTAS DESDE LA APP DE PRUEBA

- a. Accedemos a la url <http://me.li/> y se nos mostrará un aplicativo web de prueba, en el cual elegimos “acortador”:



- b. Clic en “Nuevo”



- c. Se mostrará una ventana modal, en la cual se deberá ingresar los datos solicitados:

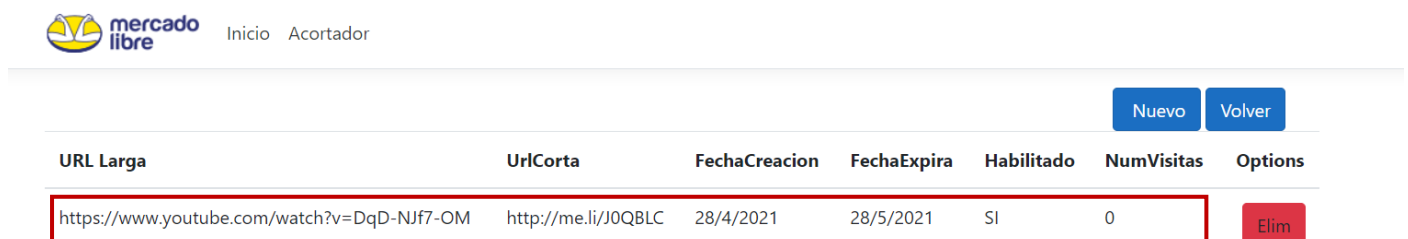
Acortar URL ×

URL Larga

Fecha Expiración

☒ Habilitado

- d. Hacer clic en “Acortar”, la app de prueba enviará la petición mediante Ajax a la Api, para realizar la inserción y el api devolverá una respuesta json que será interpretada por la app de prueba para mostrar el resultado del acortado de la URL:



Como se puede observar se crea la url corta, además de mostrar datos adicionales tales como número de visitas la cual se inicializa en cero y se irá incrementando cada vez que un usuario abre la url corta desde su navegador, esta información podría ser usada por el equipo de marketing para medir el impacto de visitas de una campaña por ejemplo, si se quisiera además se podrían agregar algunas columnas más de datos para medir el impacto por país o región inclusive, pero para efectos de este ejemplo, sólo se registra el número de visitas.

## 2. CREACIÓN DE URL CORTAS DESDE CUALQUIER APP (EJEMPLO USANDO POSTMAN)

- a. Abrimos la aplicación Postman y configuramos de la siguiente manera:

me.li / http://me.li/Acortador/Create

POST http://me.li/Acortador/Create ...

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

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

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> UrlLarga	https://www.google.com/	
<input checked="" type="checkbox"/> habilitado	True	
<input checked="" type="checkbox"/> fechaCreacion	'2021-02-01'	
<input checked="" type="checkbox"/> fechaExpira	'2021-02-02'	
<input type="checkbox"/> numVisitas	0	
<input type="checkbox"/> ID	0	
<input type="checkbox"/> urlCorta		
Key	Value	Description

El campo “UrlLarga” corresponde a la url que se va a acortar, los demás valores son opcionales.

- b. Al hacer clic en “enviar”, el api devolverá la respuesta del proceso en formato JSON:

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 6 ms Size: 401 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "auto_ID": 3,
3   "success": true,
4   "uri": {
5     "id": 3,
6     "urlLarga": "https://www.google.com/",
7     "urlCorta": "http://me.li/LRD3IH",
8     "fechaCreacion": "2021-04-28T22:09:43.45",
9     "fechaExpira": "2021-05-28T22:09:43.45",
10    "habilitado": true,
11    "numVisitas": 0
12  }
13 }
```

- c. En caso de producirse alguna excepción devolverá “success” : false, y una mensaje con la descripción del evento:

Pretty

Raw

Preview

Visualize

JSON



```

1  {
2    "success": false
3    "message": "No se recibió la Url a acortar"
4  }

```

### 3. Verificación de la Url acortada

- a. Al ingresar la url corta generada por la Api meli, automáticamente será redirigido a la url en mención y se incrementara en 1 el contador de visitas:

URL Larga	UrlCorta	FechaCreacion	FechaExpira	Habilitado	NumVisitas	Options
https://www.google.com/	http://me.li/LRD3IH	28-04-2021	28-05-2021	SI	0	Elim
https://www.google.com/	http://me.li/J125LX	28-04-2021	28-05-2021	SI	0	Elim
https://www.google.com/	http://me.li/IHNI6Z	28-04-2021	28-05-2021	SI	0	Elim

Se visualiza :

Gmail Imágenes



Y el contador de visitas aumenta en 1 cada vez que se visita el link:



Inicio Acortador

						Nuevo	Volver
URL Larga	UrlCorta	FechaCreacion	FechaExpira	Habilitado	NumVisitas	Options	
https://www.google.com/	http://me.li/LRD3IH	28-04-2021	28-05-2021	SI	1	Elim	
https://www.google.com/	http://me.li/J125LX	28-04-2021	28-05-2021	SI	0	Elim	
https://www.google.com/	http://me.li/IHNI6Z	28-04-2021	28-05-2021	SI	0	Elim	

### 4. Eliminación o destrucción de Url

El proceso es el mismo y se usa el controler <http://me.li/Acortador/Eliminar>, mediante el método post y una variable llamada "ID" tal como se describe a continuación:

me.li / http://me.li/Acortador/Eliminar

Save

...

✎

🗨

POSThttp://me.li/Acortador/Eliminar

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

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

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	ID	1			
	Key	Value	Description		

El resultado del api, al igual que los casos anteriores es en formato json:

BodyCookiesHeaders (5)Test Results

🌐 Status: 200 OK Time: 8 ms Size: 197 B Save Response

PrettyRawPreviewVisualizeJSON

≡

1

2

3

4

⌵

⌵

⌵

⌵

"success": true,

"id": 2

🗨

En caso de que el id no exista, se devolverá el mensaje correspondiente:

BodyCookiesHeaders (5)Test Results

🌐 Status: 200 OK Time: 6 ms Size: 232 B Save Response

PrettyRawPreviewVisualizeJSON

≡

1

2

3

4

⌵

⌵

⌵

⌵

"success": false,

"message": "No se encontró el ID: 2"

🗨

También se podrían agregar los eventos para eliminar por ejemplo una url su correspondiente corta y no por id, además de eliminación por url larga pero para este ejemplo, se programó para eliminar sólo por id.