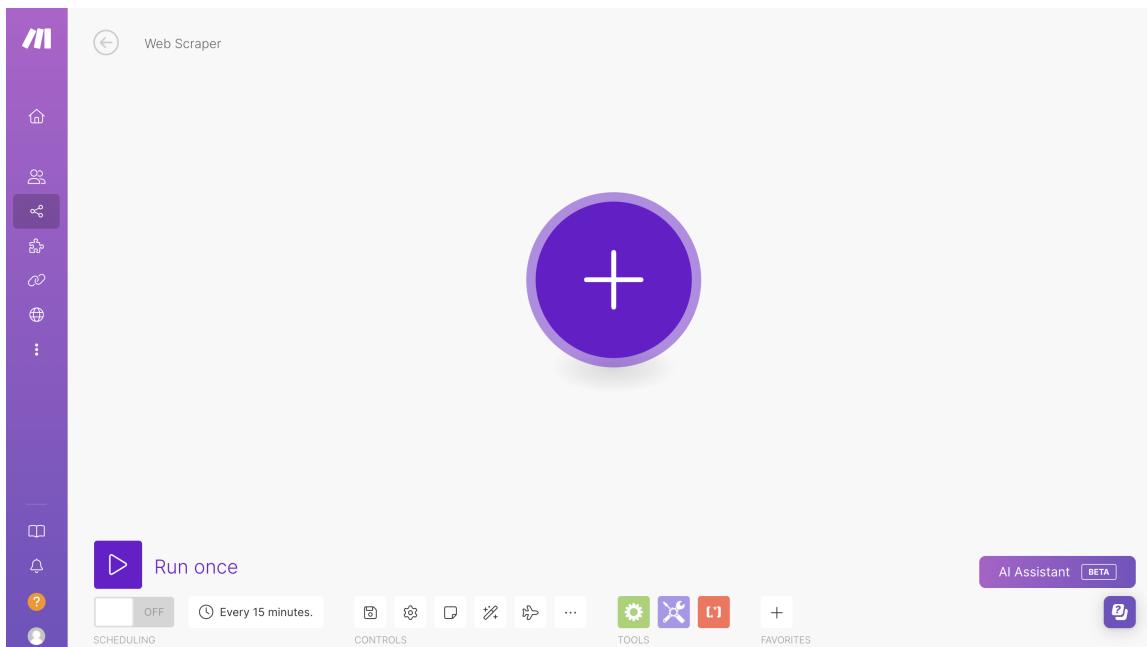


# Web Scraping [Make + OpenAI]

## Guía paso a paso

1. Accedemos a **Make** y creamos una nueva automatización, adicionalmente le podemos asignar un nombre.



2. En este caso vamos a querer scrapear, por ejemplo, libros de **Agapea**. Pongamos por ejemplo que queremos recopilar todos los libros clásicos de **Ciencia Ficción** que ofrece la plataforma.

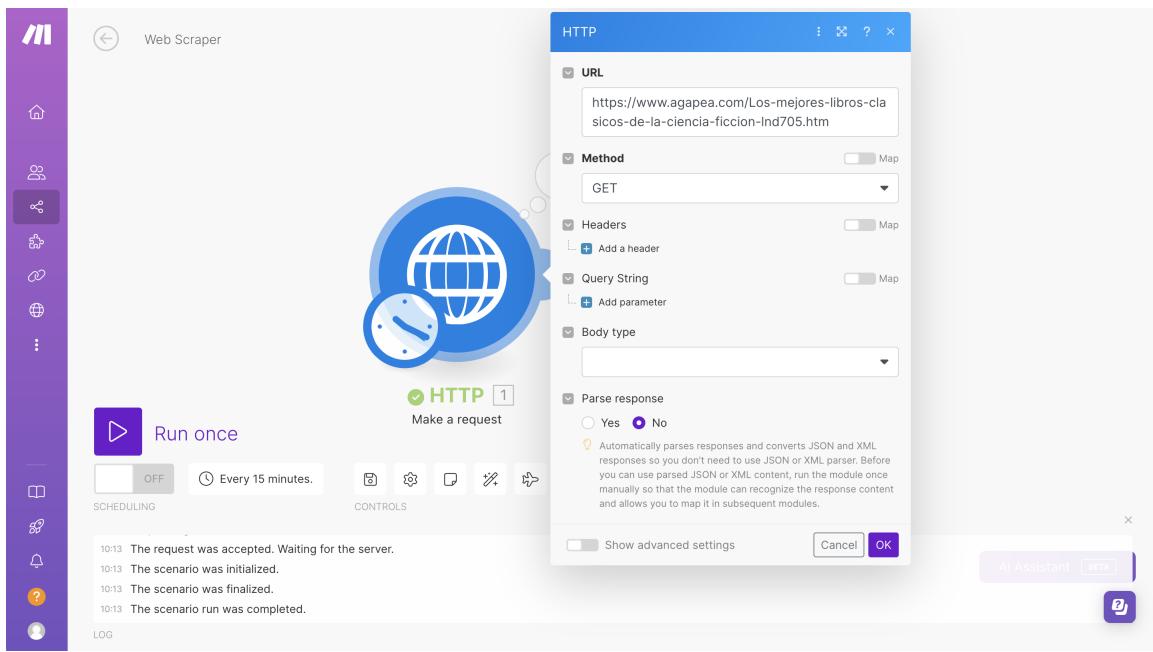
The screenshot shows the homepage of agapea.com. On the left, there's a sidebar with a navigation menu titled "CATEGORÍAS" containing links to various book categories. Below this is a contact section with a phone icon and the number "915 757 000". At the top right, there's a search bar, a login link, and a shopping cart icon showing "Total 0,00€". The main content area features a banner with the text "ELIGE TUS LIBROS ¡EL ENVIO ES GRATIS!" and a link to a recommended book list. Below the banner, a red button says "¿No sabes qué libro leer? Prueba nuestro recomendador de libros". A breadcrumb trail indicates the user is at "Libros > Los mejores libros clásicos de la ciencia ficción". The main title "Los mejores libros clásicos de la ciencia ficción" is displayed in bold. To the right, there's a grid of book covers for titles like "1984", "Fahrenheit 451", and "Distopías Maestras". Each book entry includes its price (e.g., 29,30€) and a "Comprar libro" button.

La idea es que podamos sacar de esta aplicación web tanto las imágenes, como los títulos, los precios y la información que necesitemos de forma automatizada.

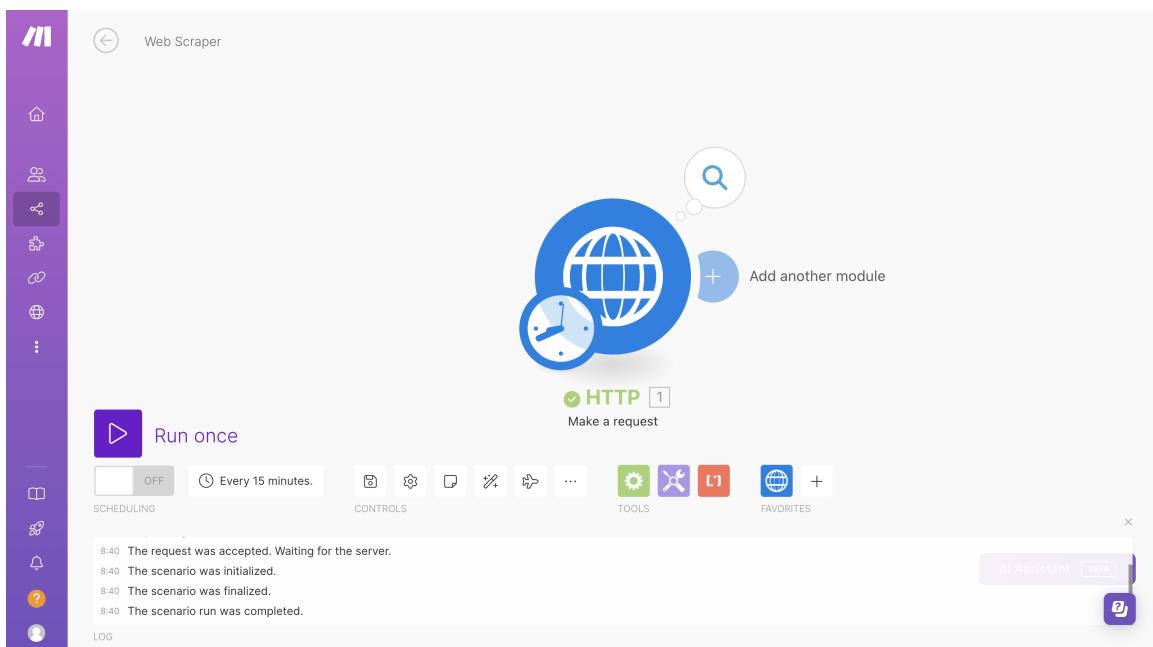
3. El primer módulo que vamos a crear es un módulo **HTTP** junto a la opción **Make a Request**, ya que queremos hacer una petición básica a una dirección URL.

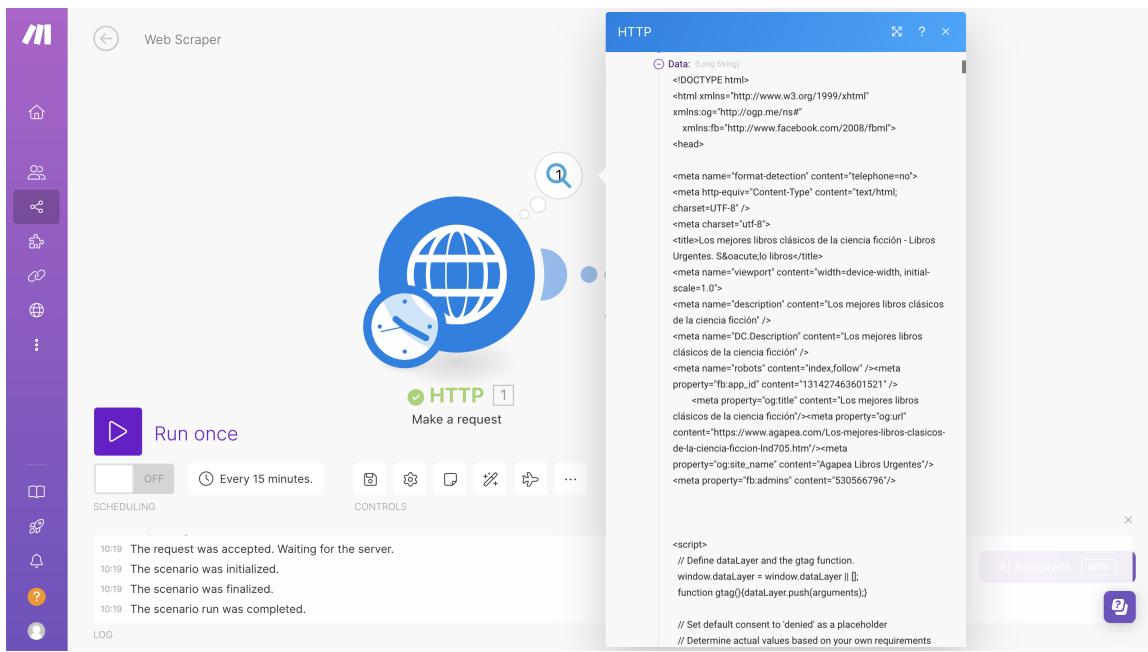
The screenshot shows the "Web Scraper" interface. On the left, there's a sidebar with icons for different modules. In the center, a modal window is open for creating a new module. The title is "HTTP" and the action selected is "Make a Basic Auth request". The modal also lists other actions like "Get a file", "Client Certificate Auth request", and "Make an API Key Auth request". Below the modal, the main interface shows a scheduled task named "HTTP" with the status "Run once". The task configuration includes a "Method" dropdown set to "GET", a "Headers" section with "Add a header", and a "Body type" dropdown. There's also a "Parse response" section with a "No" radio button selected. At the bottom right, there are "Cancel" and "OK" buttons, and an "AI Assistant" icon.

4. En las opciones de este módulo introduciremos en el campo **URL** la dirección de la que queremos extraer los datos:  
<https://www.agapea.com/Los-mejores-libros-clasicos-de-la-ciencia-ficcion-Ind705.htm>. El resto de opciones las dejaremos por defecto teniendo en consideración la opción **Parse response** en **No**, ya que transformaremos nosotros mediante otros métodos dicha información en algo tangible.



5. Vamos a probar a ver qué nos devuelve este modulo haciendo click en **Run once** y posteriormente en el icono de la lupa superior al módulo. Ahí podremos inspeccionar la salida de este módulo, y dentro del apartado **Data** podremos ver cómo ha extraído todo el código HTML de la aplicación web. Esta información será el punto de partida desde el cual extraeremos los datos que queramos.

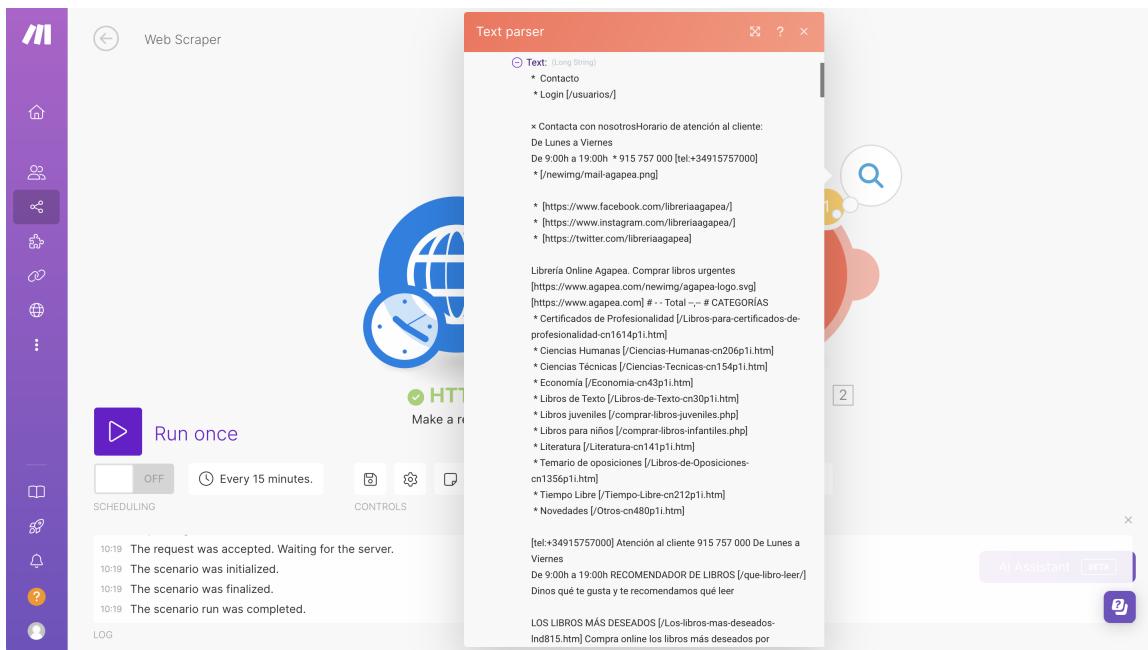




6. Nuestra tarea ahora es **transformar** este código HTML en algo tangible y manejable sin necesidad de intervenir con código, para ello vamos a utilizar un módulo que viene incluido en **Make** llamado **Text parser/HTML to Text**, el cual añadiremos a la derecha del primer módulo. En la opción de HTML que nos indica, introduciremos la salida **Data** que anteriormente hemos inspeccionado.



En esta salida del módulo Text parser podremos ver de una forma un poco más legible la información recopilada dentro del apartado **Text**.



7. La idea es que no nos quedemos con los datos de listado de libros sino que seamos capaces de automatizar el acceder al detalle, por ejemplo, el primer resultado que vemos es el libro **Distopías Maestras**, y si hacemos click en él veremos el detalle de artículo.

The screenshot shows a product page for 'Distopías maestras (pack que incluye: 1984, Fahrenheit 451 y Un mundo feliz)' by Aldous Huxley, George Orwell, and Ray Bradbury. The page features a banner at the top advertising free shipping. Below the banner, there's a red button for a book recommendation tool. The product details include the title, authors, price (29,30 €), delivery information (1 a 7 días), and a 'Comprar / Recoger' button. The sidebar on the left lists various book categories, and a contact phone number (915 757 000) is displayed.

Para ello vamos a investigar en qué sitio del texto que nos ha devuelto Make está el enlace del detalle. Vamos a probar con esta dirección en la que estamos: <https://www.agapea.com/Aldous-Huxley/Distopias-maestras-pack-que-incluye-1984-Fahrenheit-451-y-Un-mundo-feliz--9788466359542-i.htm>

Si buscamos con Chrome dentro del texto del segundo módulo, veremos que esta dirección aparece en una posición determinada (no incluyendo agapea.com):

The screenshot shows a Make workflow titled 'Web Scraper'. It includes a 'Text parser' step with the URL: 'ELIGE TUS LIBROS ¡EL ENVIO ES GRATIS!\* \* Envíos a partir de 18€ para España peninsular, Tenerife, Gran Canaria y Mallorca'. The workflow also includes a 'SCHEDULING' section with a 'Run once' button and a 'Every 15 minutes' option. The log on the left shows the execution history:

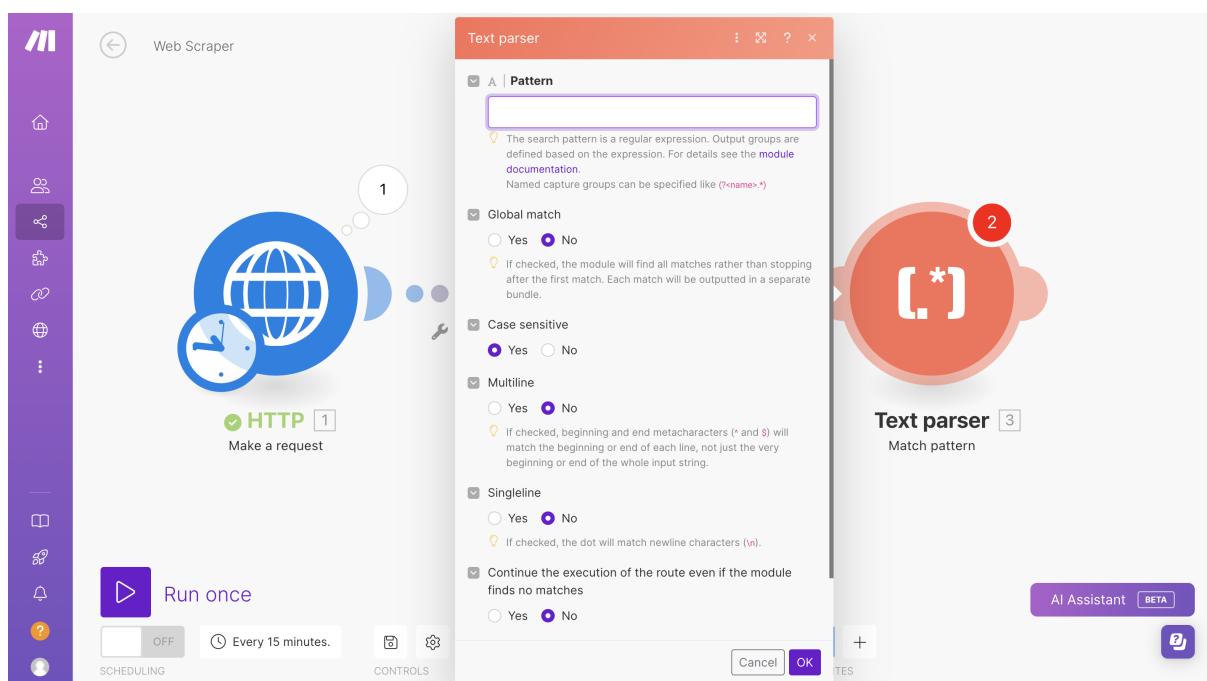
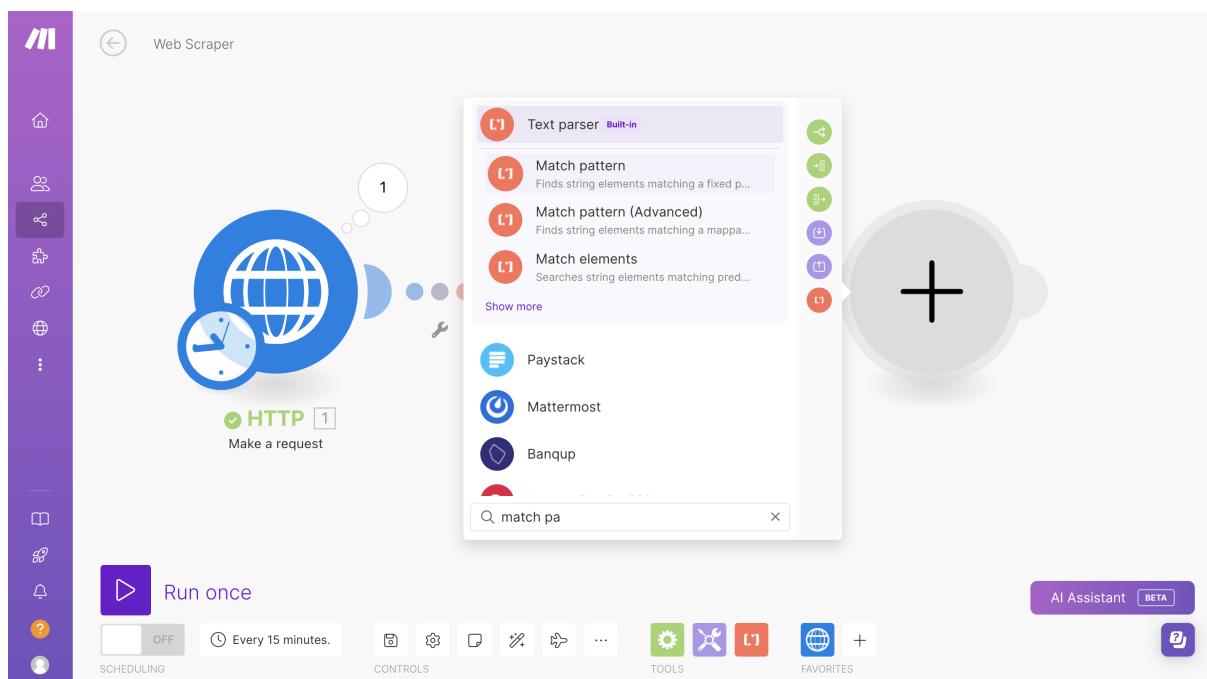
```

10:19 The request was accepted. Waiting for the server.
10:19 The scenario was initialized.
10:19 The scenario was finalized.
10:19 The scenario run was completed.

```

Si inspeccionamos otros libros vemos como el patrón se repite, por lo que podremos recuperar todas estas direcciones si sabemos donde buscar. Al final, lo que queremos conseguir es de alguna forma, recopilar todos estos enlaces, clicar en ellos y recopilar dicha información. Para ello tenemos que buscar un **patrón**, así que haremos uso de una herramienta integrada en Make para ello.

- Vamos a añadir a la derecha de nuestro HTML to Text otro **Text Parser** con la opción **Match pattern**. En la opción **Pattern** tendremos que introducir un patrón similar a la URL que queríamos de un libro.



Estos patrones se introducen mediante un lenguaje un poco extraño llamado **REGEX**, el cual desconocemos. Pero gracias a la IA podemos crear un patrón sin mucha complicación.

Para crear el patrón iremos a ChatGPT y utilizaremos un GPT llamado **Regex Helper**, que está entrenado exclusivamente en la generación de estos patrones.

The screenshot shows the GPTs interface with a dark theme. At the top right are buttons for 'Mis GPTs' and '+ Crear un GPT'. The main title 'GPTs' is centered above a search bar containing the query 'Regex Helper'. Below the search bar, there are two tabs: 'Todos los resultados' and 'Espacio de trabajo ThePower Productos'. A list of search results follows:

- Regex Helper**  
Expert in creating, explaining, and testing regular expressions.  
Por Austin Coleman ⚡ 1K+
- Regex Helper**  
Generates regex patterns from examples, explains how they work.  
Por evlo.app ⚡ 0
- Regex Helper**  
Creates precise regex patterns for Visual Studio searches.  
Por Johan Johansson ⚡ 2
- Regex Helper**  
Creates secure regex patterns with clear explanations and safe test URLs.  
Por thedevartisan.com ⚡ 4
- Regex Helper**  
I swiftly provide regex solutions, test, then explain minimally.  
Por Airfill Prepaid AB ⚡ 4
- Regex Helper**  
Friendly regex creator and tester  
Por Mr George Roberts ⚡ 1
- RegEx Helper**  
I simplify and adjust RegEx.  
Por Mr George Roberts ⚡ 3

Le vamos a pasar uno de los links y le vamos a enviar 4 o 5 ejemplos más con el texto recuperado del anterior **Text Parser**.



AN

Tú

Quiero encontrar textos como este haciendo uso de Regex: [/Aldous-Huxley/Distopias-maestras-pack-que-incluye-1984-Fahrenheit-451-y-Un-mundo-feliz--9788466359542-i.htm]

Aquí un fragmento de un texto completo:

Distopías Maestras (pack Que Incluye: 1984, Fahrenheit 451 Y Un Mundo Feliz) De Huxley, Aldous; Orwell, George; Bradbury, Ray [https://cdn.agapea.com/Distopias-maestras-pack-que-incluye-1984-Fahrenheit-451-y-Un-mundo-feliz--i0n21251476.jpg] [/Aldous-Huxley/Distopias-maestras-pack-que-incluye-1984-Fahrenheit-451-y-Un-mundo-feliz--9788466359542-i.htm] DISTOPÍAS MAESTRAS (PACK QUE INCLUYE: 1984, FAHRENHEIT 451 Y UN MUNDO FELIZ) [/ALDOUS-HUXLEY/DISTOPIAS-MAESTRAS-PACK-QUE-INCLUYE-1984-FAHRENHEIT-451-Y-UN-MUNDO-FELIZ--9788466359542-I.HTM]

\* Huxley, Aldous; Orwell, George; Bradbury, Ray

\* 480 páginas

\* Presentamos un pack con tres obras fundamentales del siglo XX: Un mundo feliz (1932), 1984 (1949) y Fahrenheit 451 (1953). Sin ninguna duda, pocas novelas de anticipación han sido tan influyentes como estas en la imaginación literaria poster...

30,85€ 29,30€

\* 1984 De George Orwell [https://cdn.agapea.com/1984-i0n21535026.jpg] [/George-Orwell/1984-9788418933011-i.htm] 1984 [/GEORGE-ORWELL/1984-9788418933011-i.HTM]

\* Orwell, George

\* 336 páginas

\* El periodista británico George Orwell fue ninguneado a partes iguales por sus correligionarios y por sus adversarios políticos. Alarmado y encantado por el cariz que tomaba el nuevo orden surgido tras la Segunda Guerra Mundial, y enfermo...

0

Mensaje Regex Helper...



Las conversaciones del espacio de trabajo ThePower Producto no se utilizan para entrenar nuestros modelos. ChatGPT puede cometer errores.



Y con esta información nos devolverá y nos justificará un patrón en código Regex con el cual podremos encontrar solamente las coincidencias que cumplan este patrón.

Regex Helper ▾

Copy code

```
\/[A-Za-z-]+/[A-Za-z0-9-]+\.\htm\]
```

Desglose del patrón:

- `\[` : Coincide con un corchete de apertura seguido de una barra. Se usa una barra invertida `\\` para escapar el corchete, ya que de otra manera sería considerado como un metacarácter.
- `[A-Za-z-]+` : Coincide con una o más letras (mayúsculas o minúsculas) o guiones. Esta parte está destinada a coincidir con el nombre inicial en la URL.
- `/` : Coincide con la barra que separa las partes de la URL.
- `[A-Za-z0-9-]+` : Coincide con una o más letras (mayúsculas o minúsculas), números o guiones. Esta parte está destinada a coincidir con el segmento medio de la URL, que puede incluir el título del libro y otros identificadores.
- `\.\htm` : Coincide con `.htm`. La barra invertida se usa para escapar el punto, ya que un punto sin escapar coincidiría con cualquier carácter.
- `\]` : Coincide con un corchete de cierre. Se usa una barra invertida para escapar el corchete.

Esta expresión regular es bastante simple y puede necesitar ajustes para manejar casos más complejos o variaciones en el patrón de las URLs que quieras encontrar. Por ejemplo, si el caso de las letras (mayúsculas/minúsculas) es relevante o si hay variaciones en la extensión del archivo (como `.HTM`), la expresión puede necesitar ser más específica.

¿Te gustaría probar esta expresión regular  algunos textos de ejemplo para ver cómo funciona?

Mensaje Regex Helper...

Las conversaciones del espacio de trabajo ThePower Producto no se utilizan para entrenar nuestros modelos. ChatGPT puede cometer errores.

Si introducimos dicho pattern en su input correspondiente y activamos la opción **Global match** (de esta manera no se detendrá en la primera coincidencia si no que mostrará todas las coincidencias), veremos como el resultado refleja cada uno de los links recuperados tal y como queríamos en un montón de **Bundles** independientes.

Text parser

Pattern

```
\/[A-Za-z-]+/[A-Za-z0-9-]+\.\htm\]
```

The search pattern is a regular expression. Output groups are defined based on the expression. For details see the [module documentation](#). Named capture groups can be specified like `(?<name>...)`

Global match

Yes  No

If checked, the module will find all matches rather than stopping after the first match. Each match will be outputted in a separate bundle.

Case sensitive

Yes  No

Multiline

Yes  No

If checked, beginning and end metacharacters (^ and \$) will match the beginning or end of each line, not just the very beginning or end of the whole input string.

Singleline

Yes  No

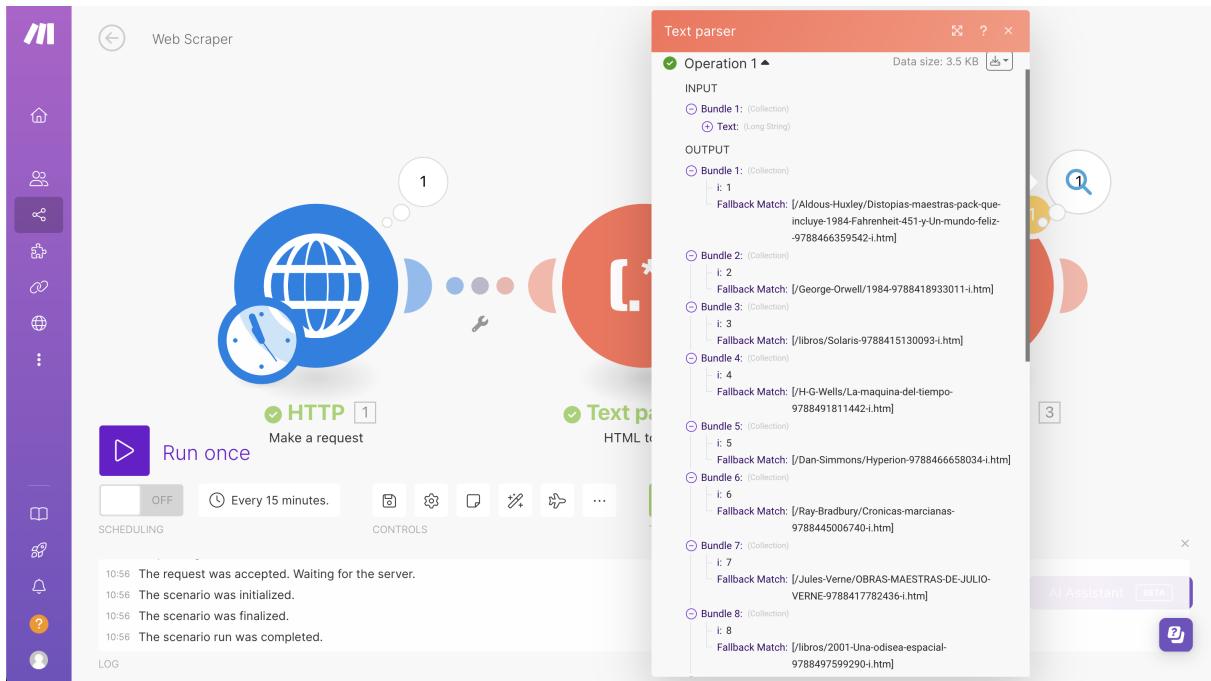
If checked, the dot will match newline characters (\n).

Continue the execution of the route even if the module finds no matches

Yes  No

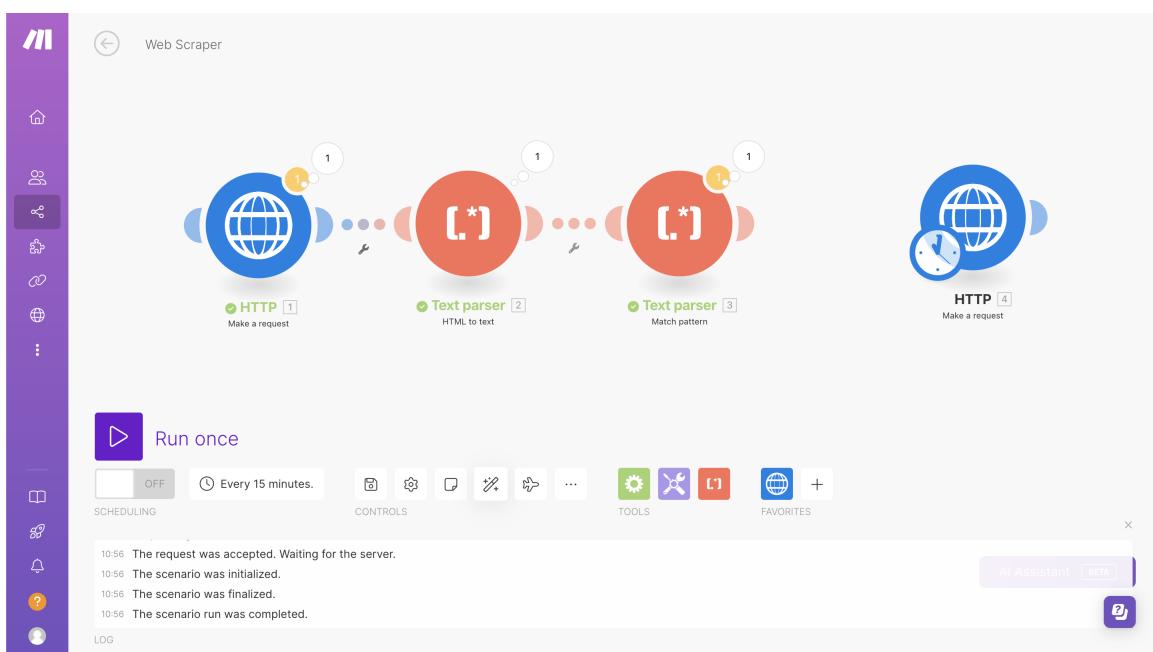
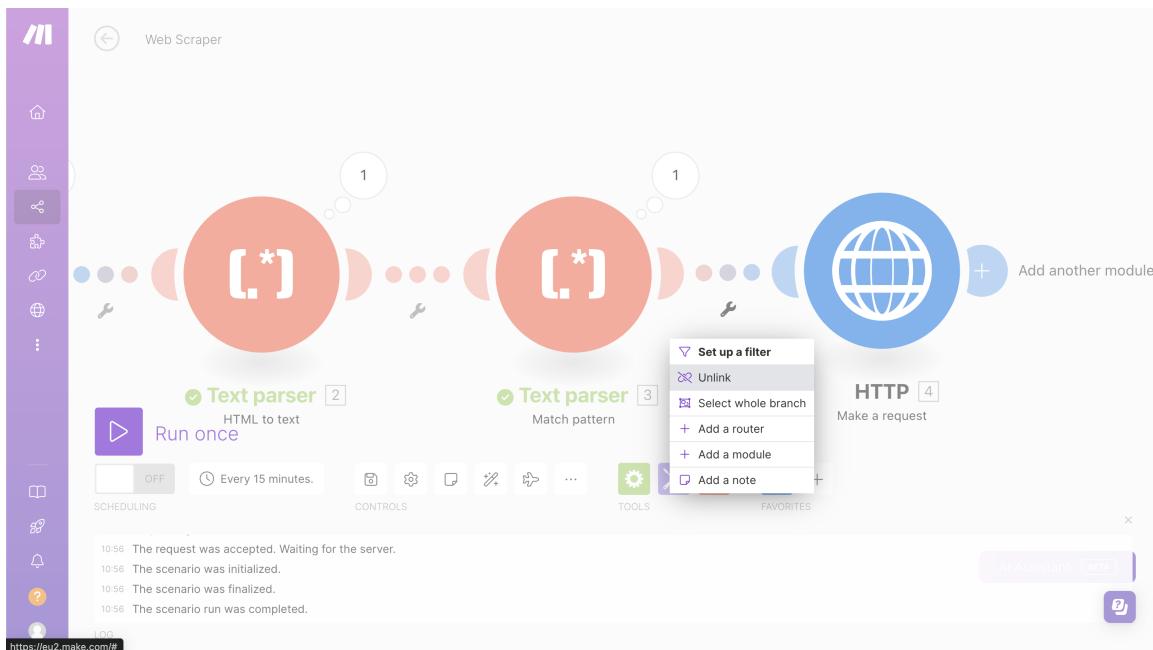
Show advanced settings

Cancel OK

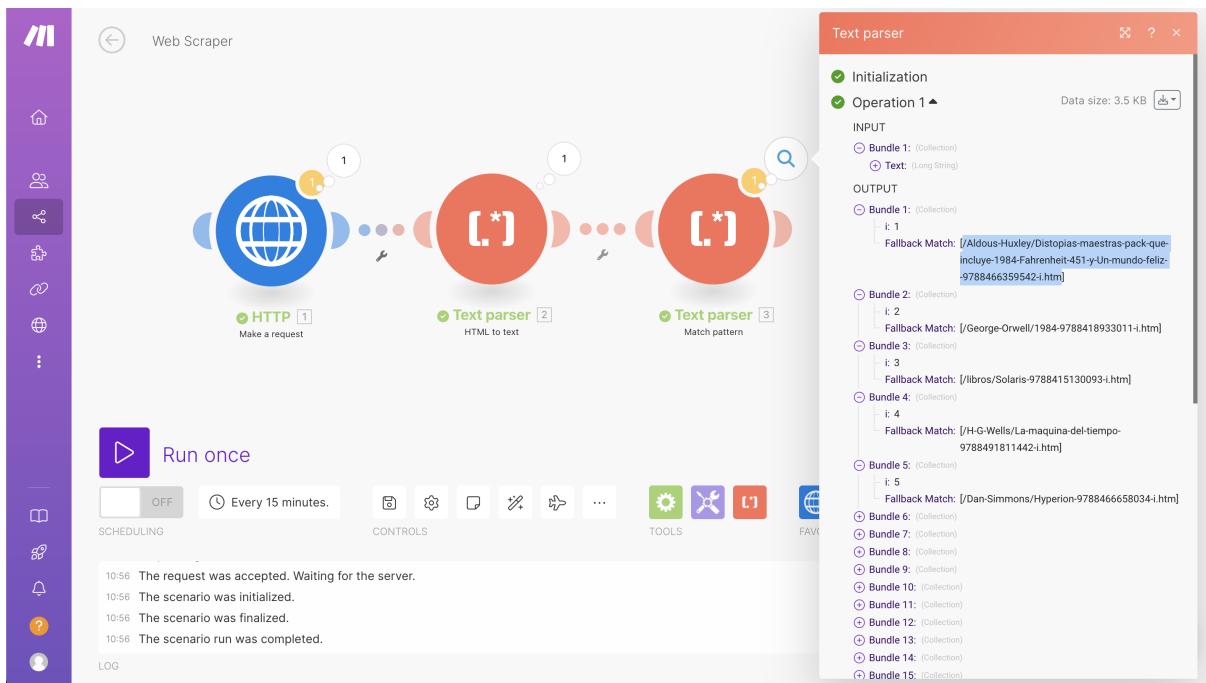


9. En este momento tenemos parte del link (ya que falta justo delante <https://www.agapea.com/>), y queremos clicar en cada uno de los recuperados, así que justo a la derecha de nuestro último parser, añadiremos otro módulo **HTTP** con **Make Request** de nuevo, pero en esta ocasión en vez de hacer una petición a una URL global, la haremos en cada una de las que le indiquemos recuperadas del parser.

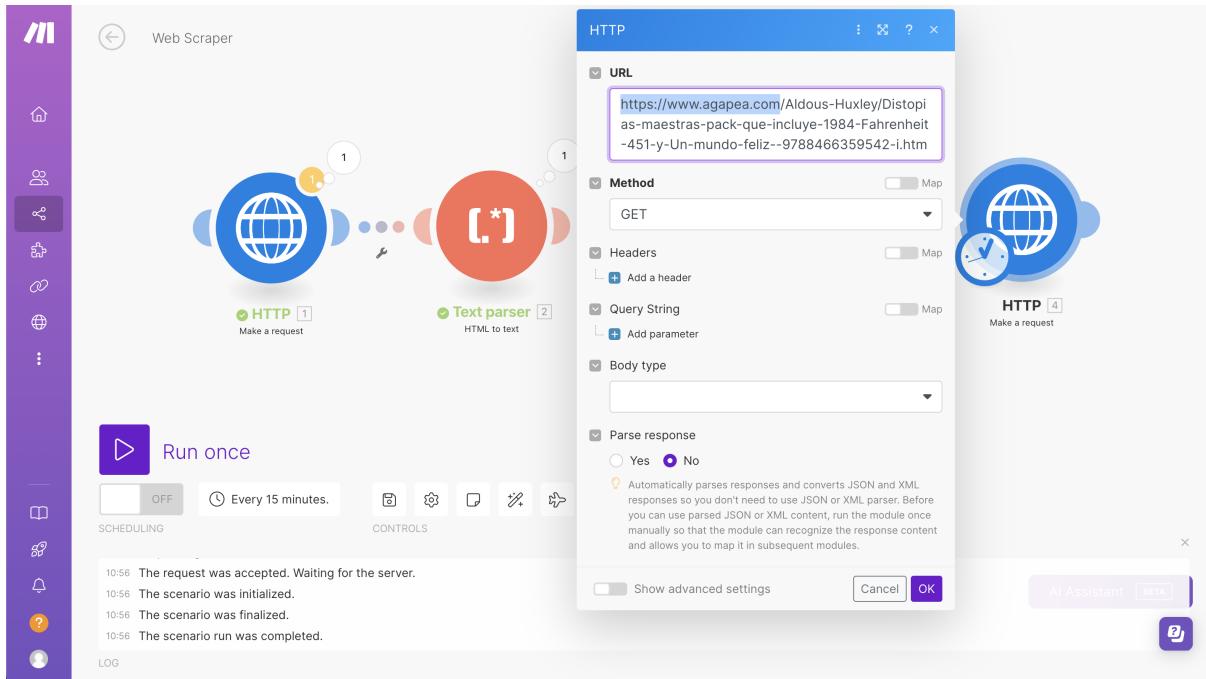
Pero primero, vamos a probar con uno de ellos de manera aislada, así que vamos a separar este módulo nuevo y le arrastraremos el icono del reloj del primer módulo para empezar la ejecución ahí. De esta manera comprobaremos si funciona en uno de los libros para que funcione en todos automáticamente.



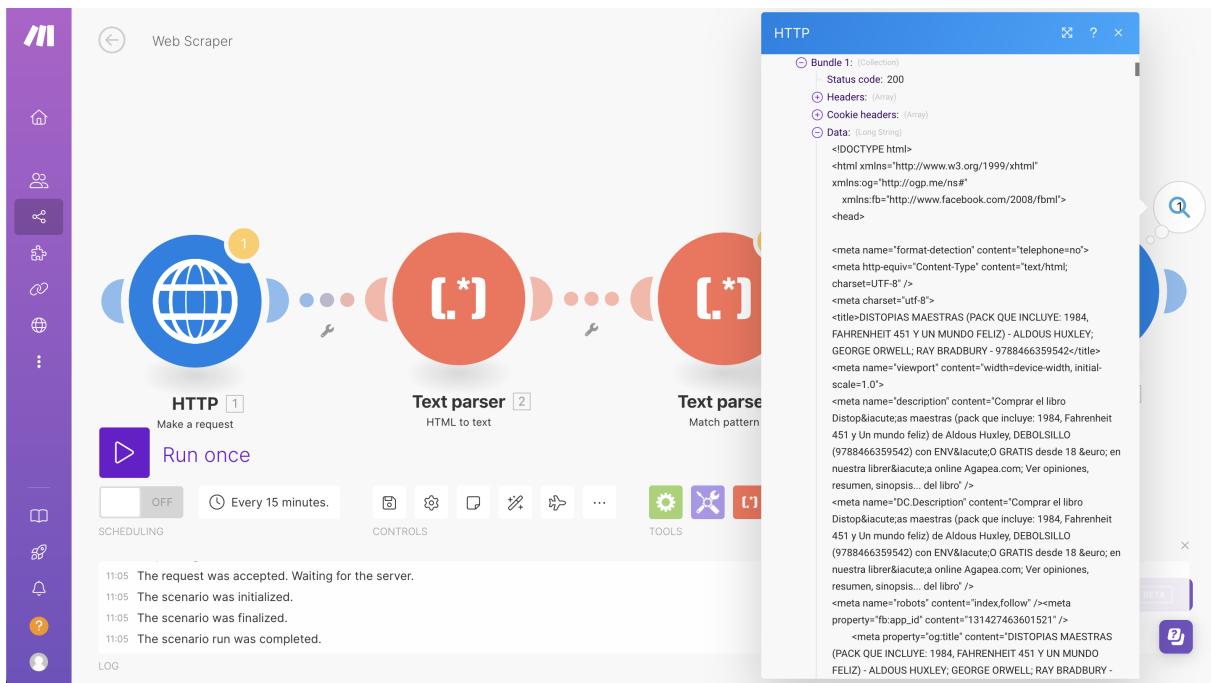
Para ello vamos a recuperar uno de los resultados, en este caso **/Aldous-Huxley/Distopias-maestras-pack-que-incluye-1984-Fahrenheit-451-y-Un-mundo-feliz--9788466359542-i.htm**



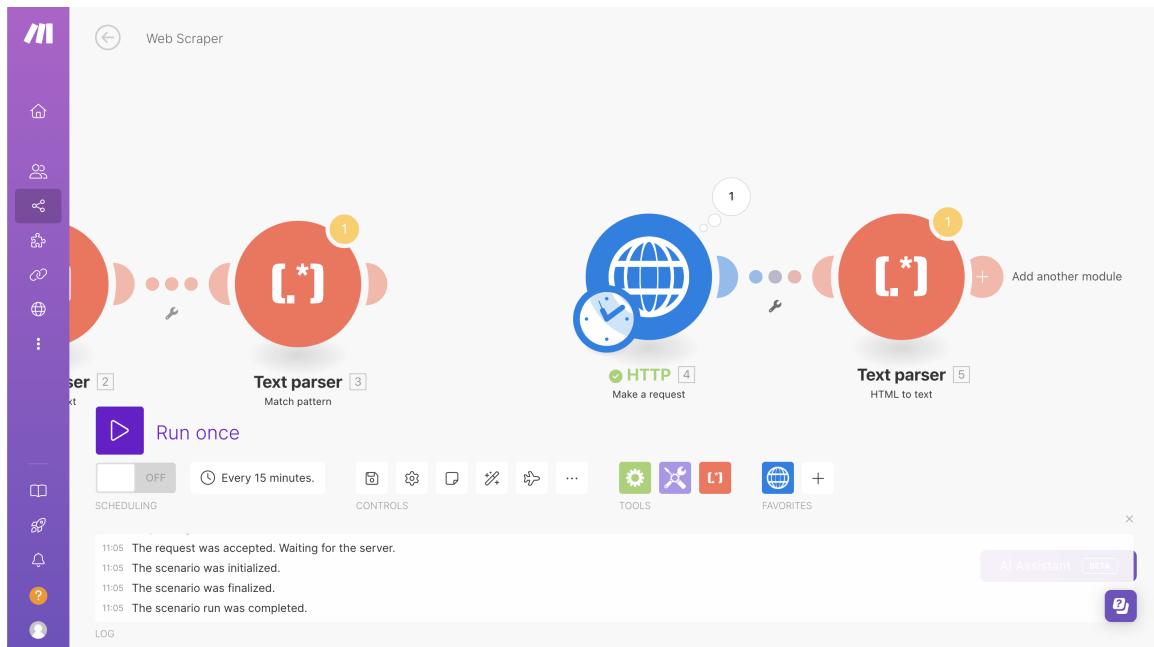
Y lo vamos a añadir a este módulo temporalmente independiente teniendo en cuenta que tenemos que añadirle al principio <https://www.agapea.com> (sin barra al final porque la barra ya la tiene el link recuperado).

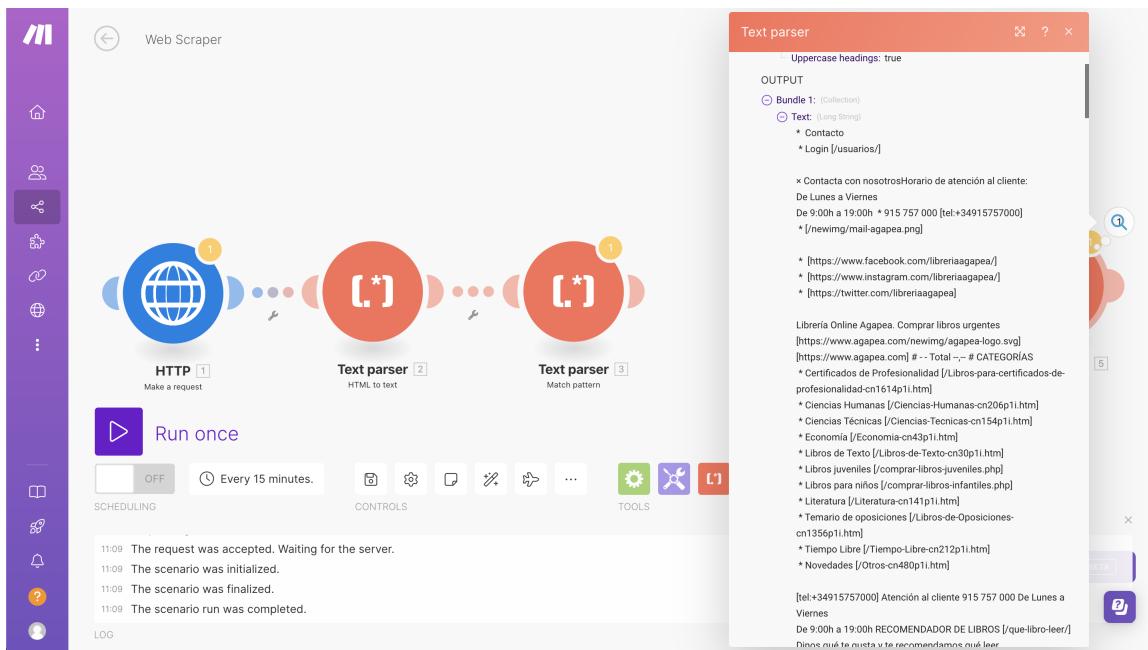
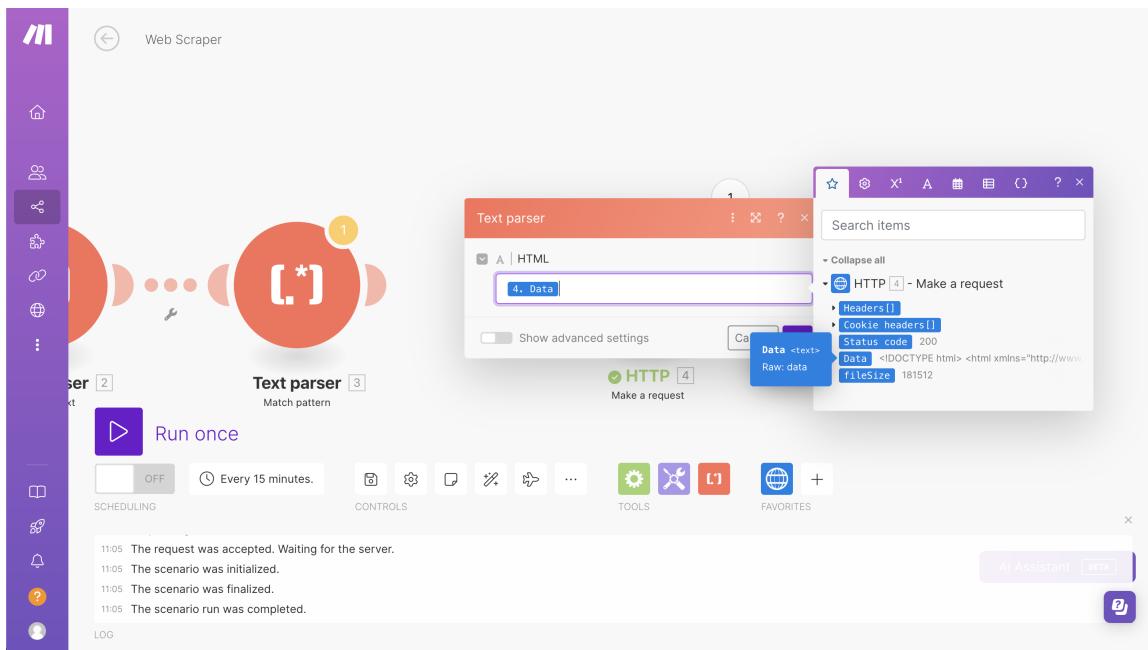


Lo probamos y al mirar el resultado veremos que nos vuelve a devolver en **Data** el HTML sin parsear.

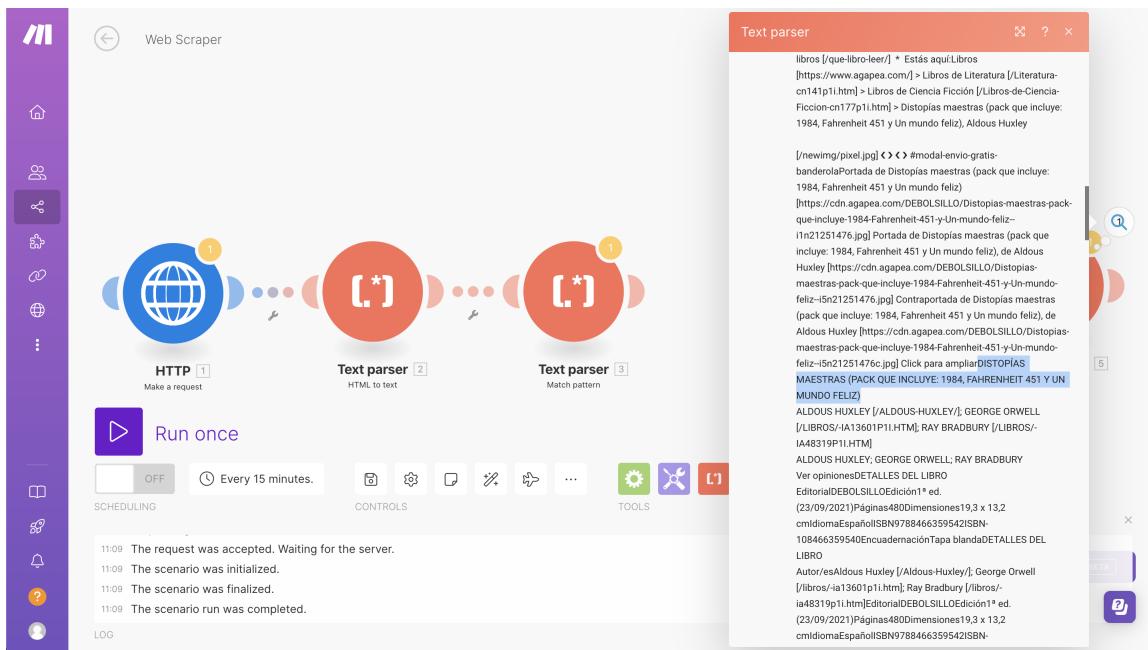


10. En este punto, a este módulo tenemos que volver a añadirle un **Text parser** con **HTML to text** para convertirlo a un formato más amigable pasándole el resultado de esta petición HTTP:

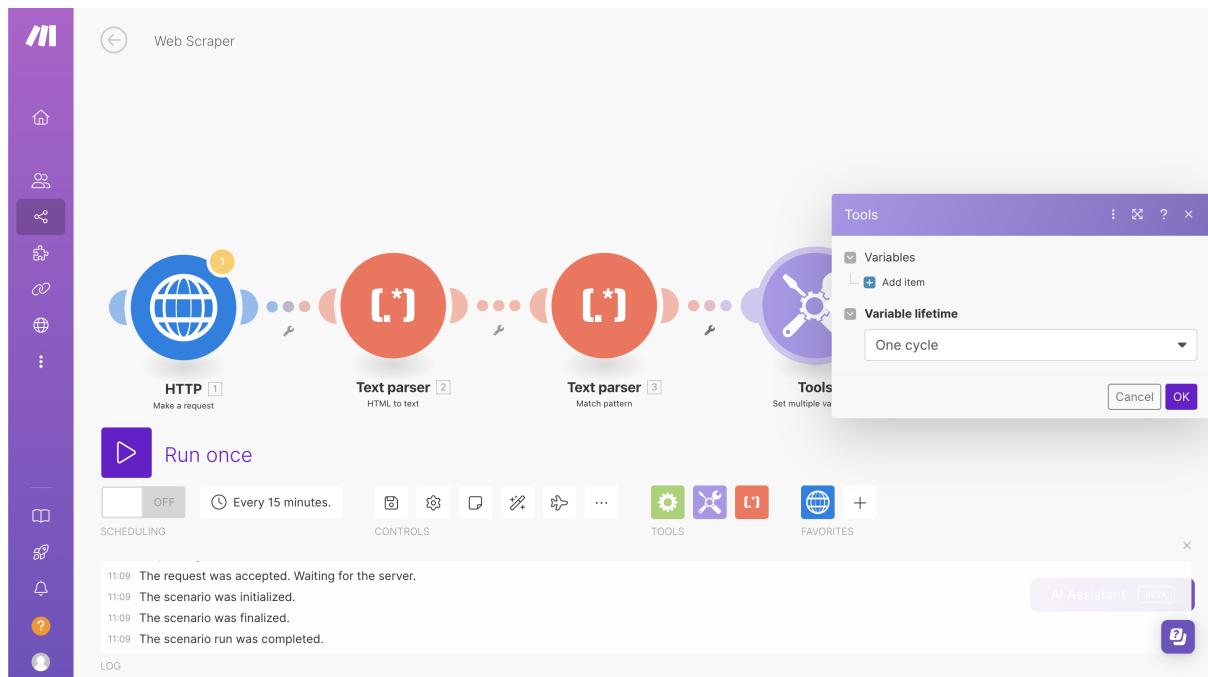




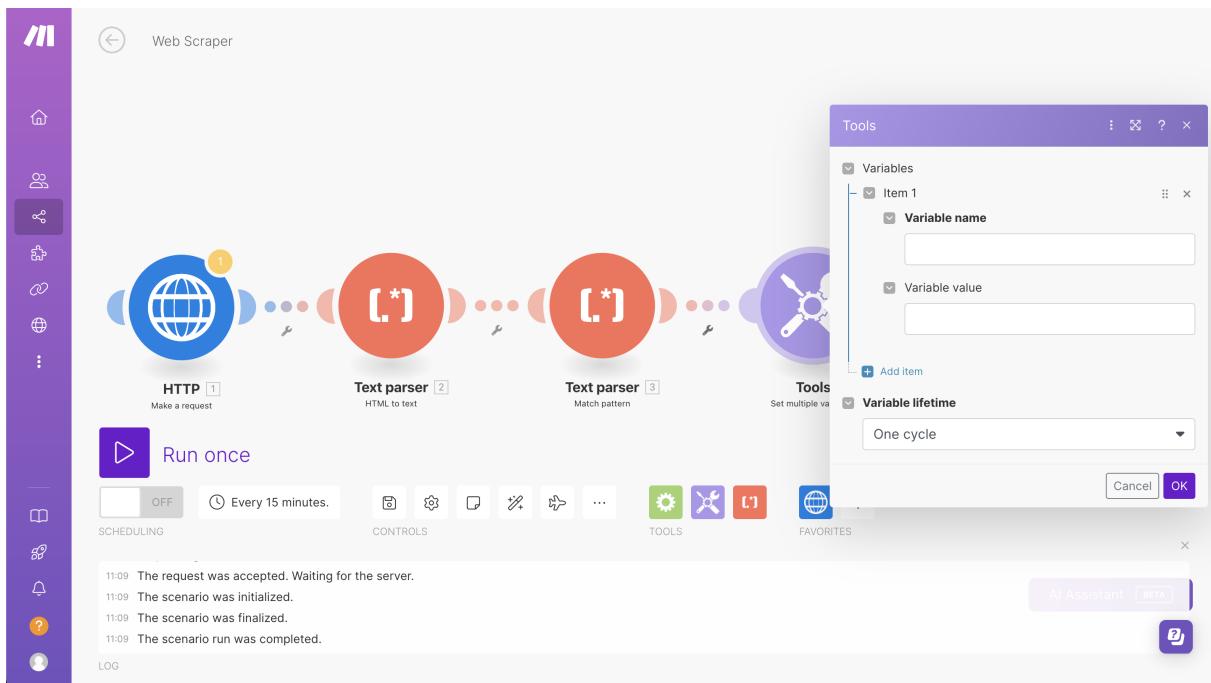
11. Ahora tenemos que investigar qué necesitamos de estos datos. Si buscamos dentro de este resultado de texto, podremos encontrar los datos que necesitamos:



La extracción de estos datos la tendremos que hacer mediante una función que almacene dichas informaciones en diferentes variables. Esto lo haremos a través de **Tools** y **Set multiple variables**.

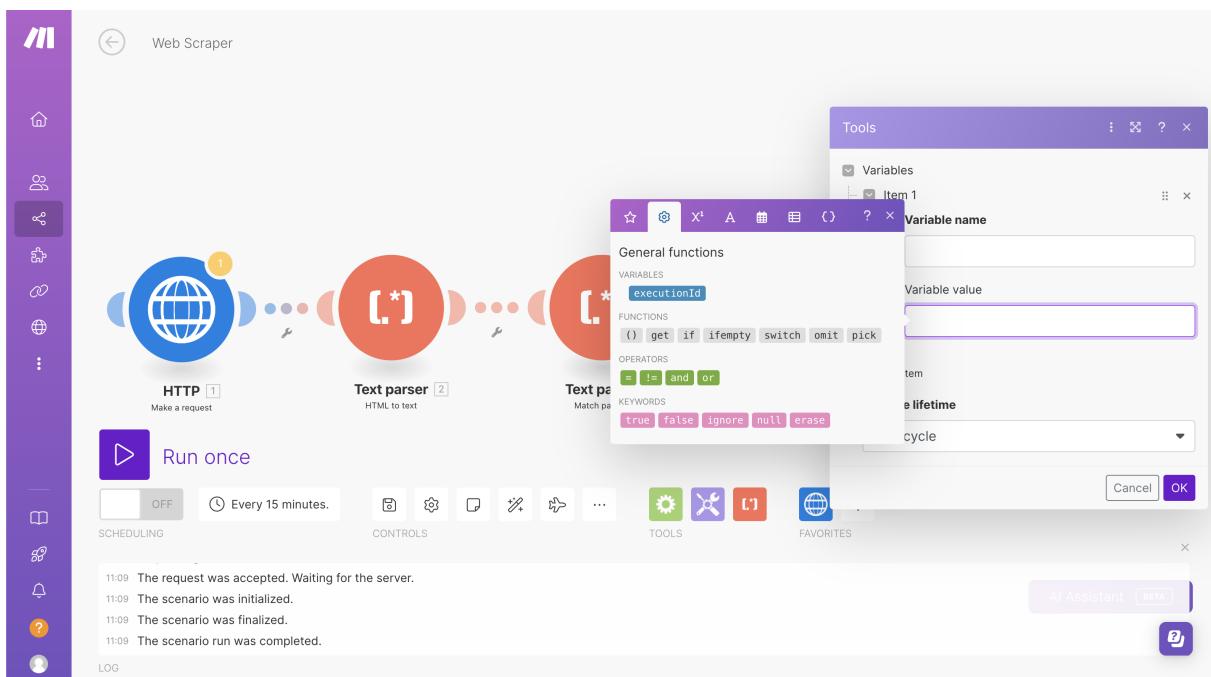


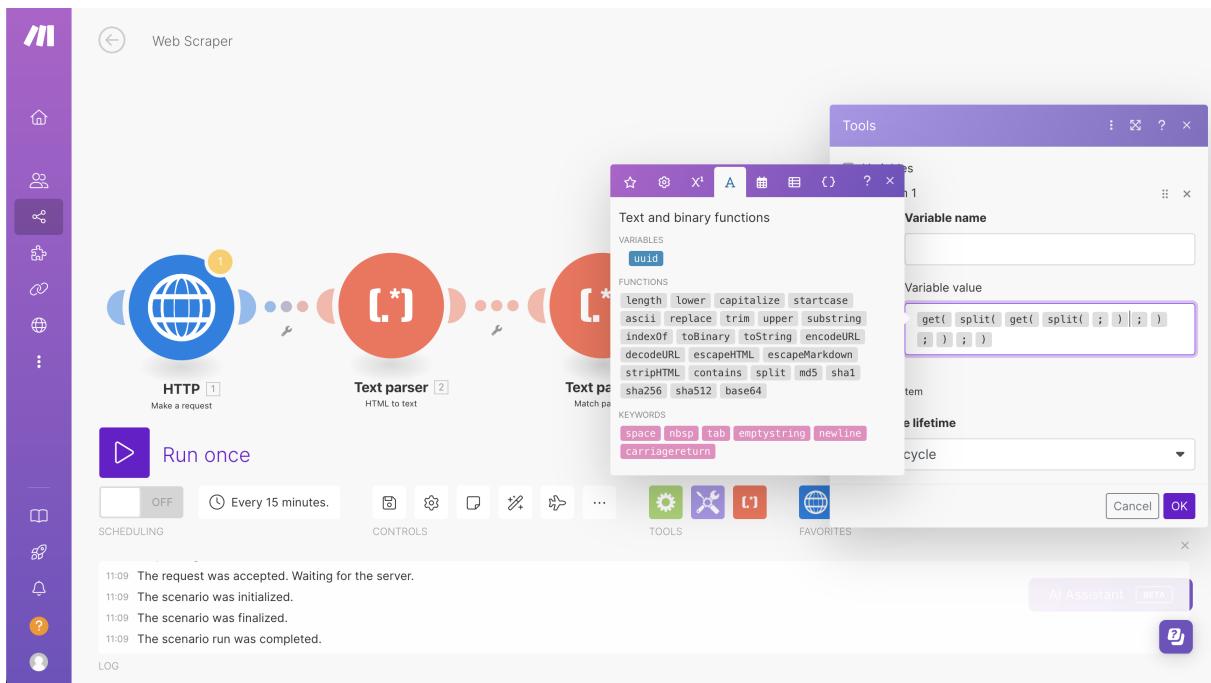
Añadimos un nuevo item.



Y vamos a definir una fórmula para trabajar en “horquillas”, es decir, vamos a intentar atrapar lo que hay después y antes de un elemento, la información estará en medio. Para ello utilizaremos **get** para obtener y **split** para trocear. Esto parece complicado en un principio pero es una tarea muy repetitiva.

En **Variable value** escogeremos la opción **General functions** para obtener el **get** y en **Text and binary functions** para obtener el **split**.

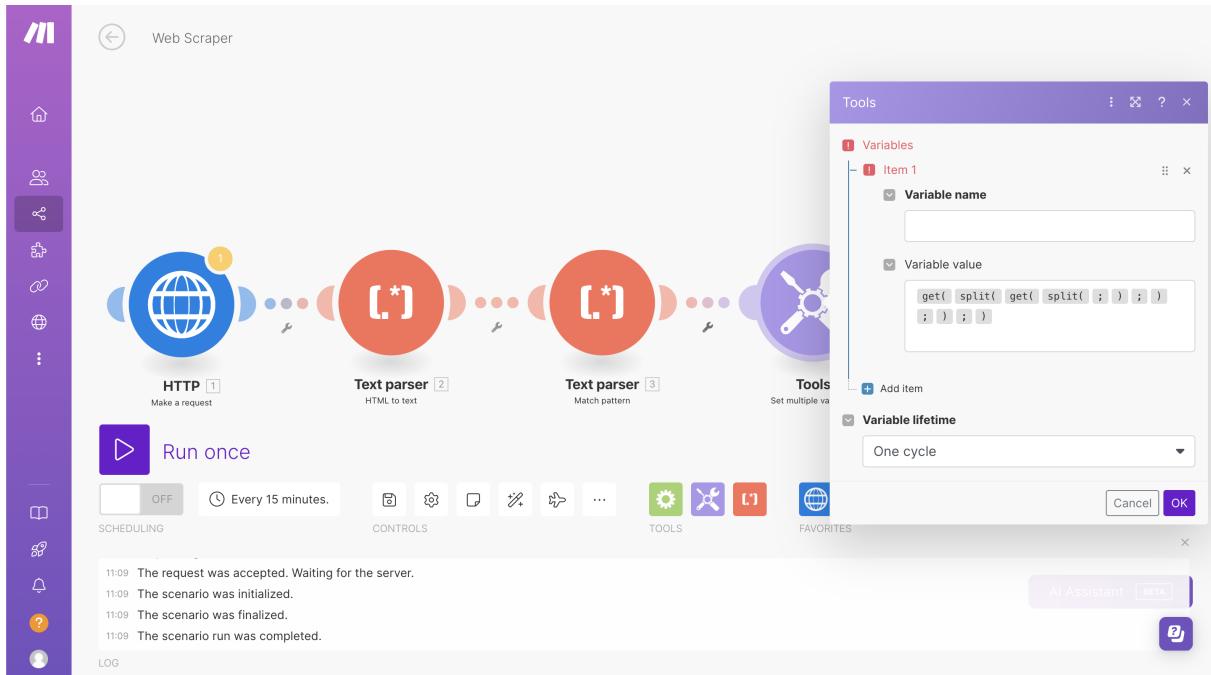




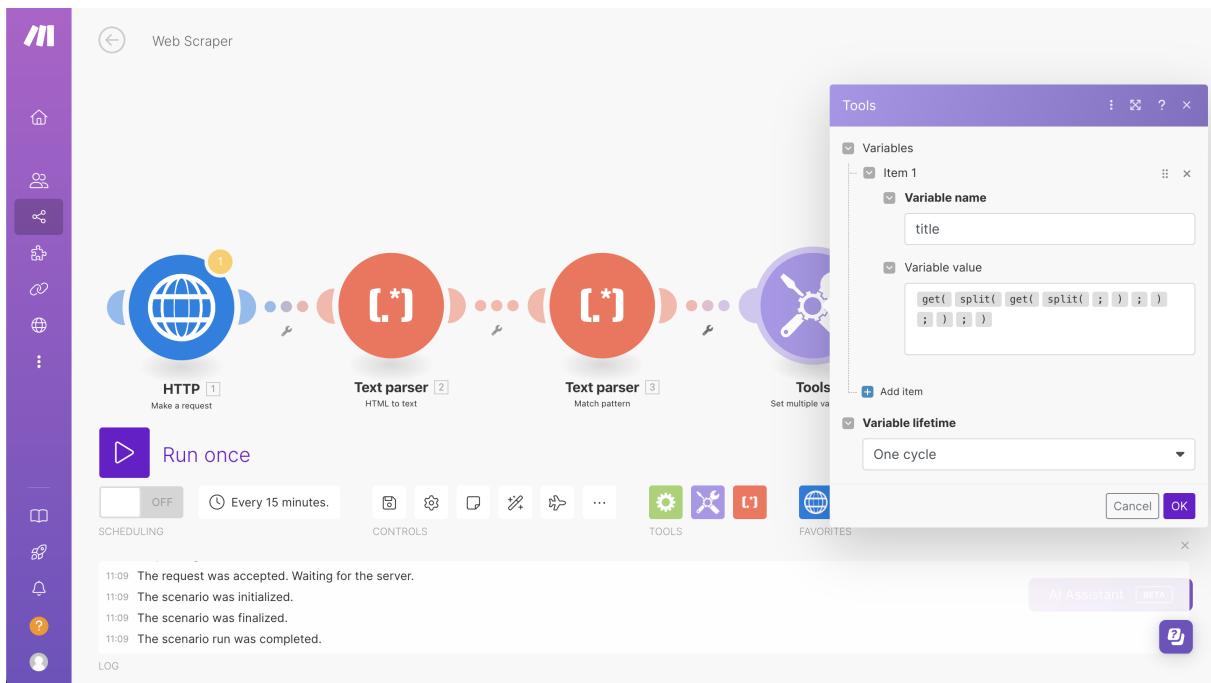
Y en este apartado escogeremos un get que contiene dentro un split que a su vez tiene un get que contiene un split. Debería quedar una fórmula así:

```
{{get(split(get(split(; ); ); ); ))}}
```

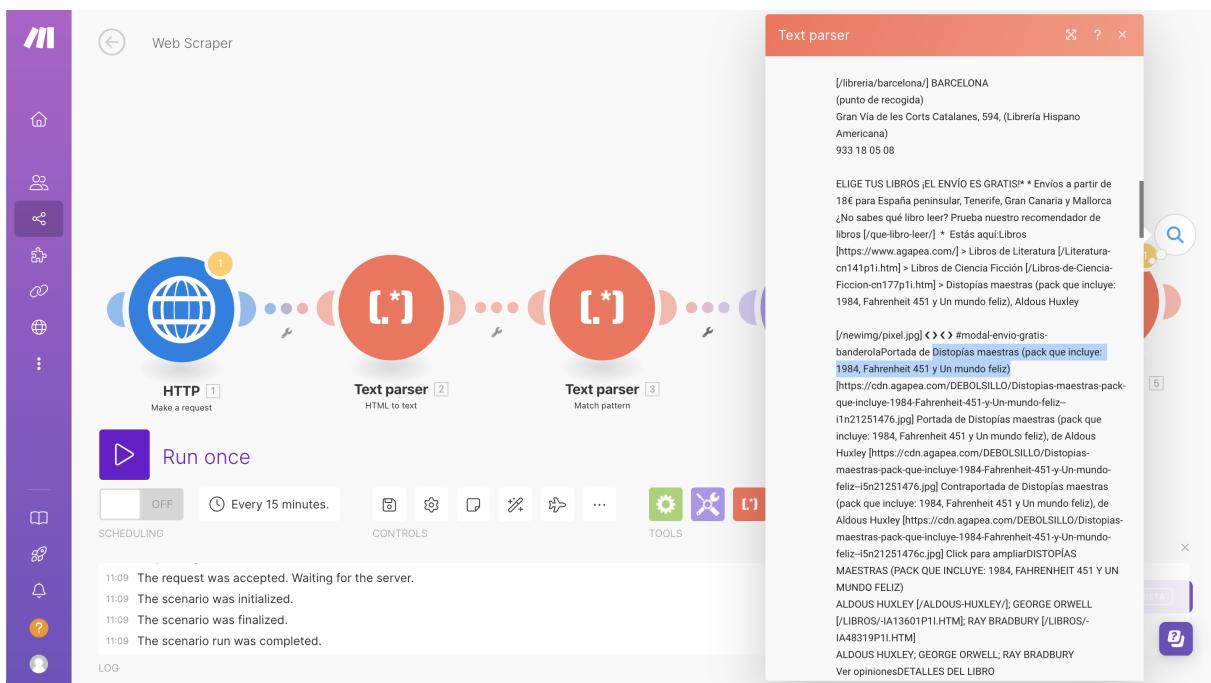
Podeis copiar y pegar este código y pulsar **Enter** para tener ese código incluido, ya que siempre será la misma fórmula.



En **Variable name** vamos a indicarle el nombre de nuestra primera caja, en la cual almacenaremos el título, por lo tanto la podemos llamar **title**.

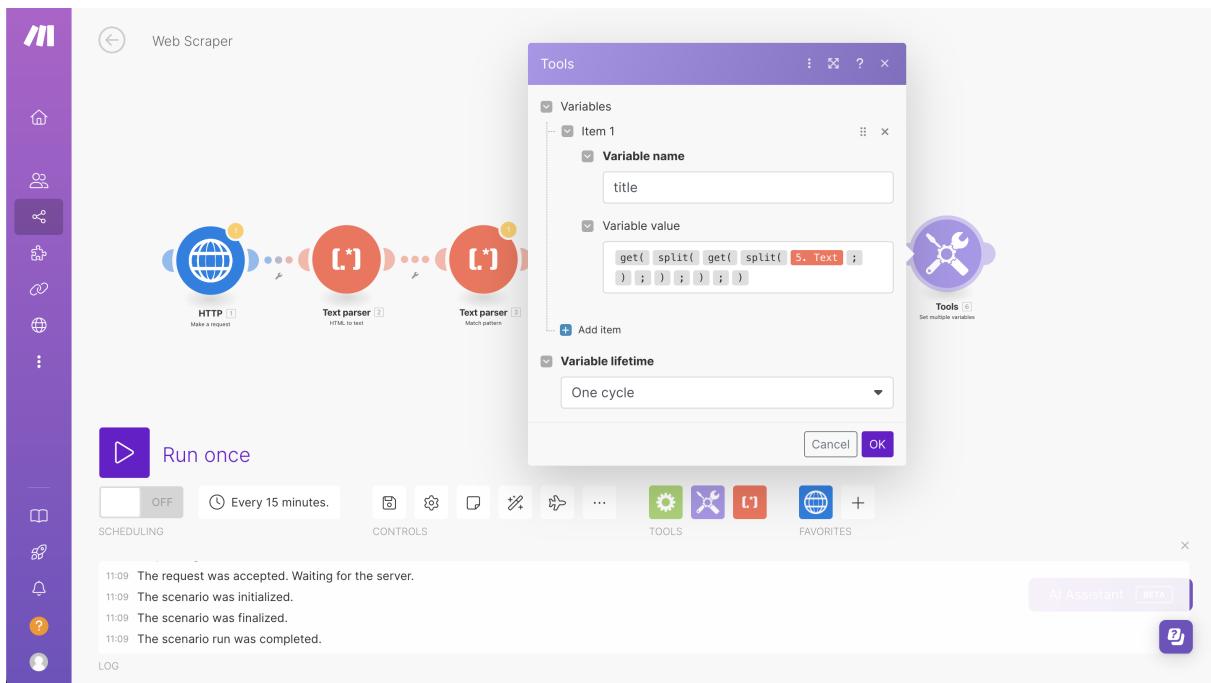


Y vamos a explorar en el texto del parser dónde se encuentra el título:

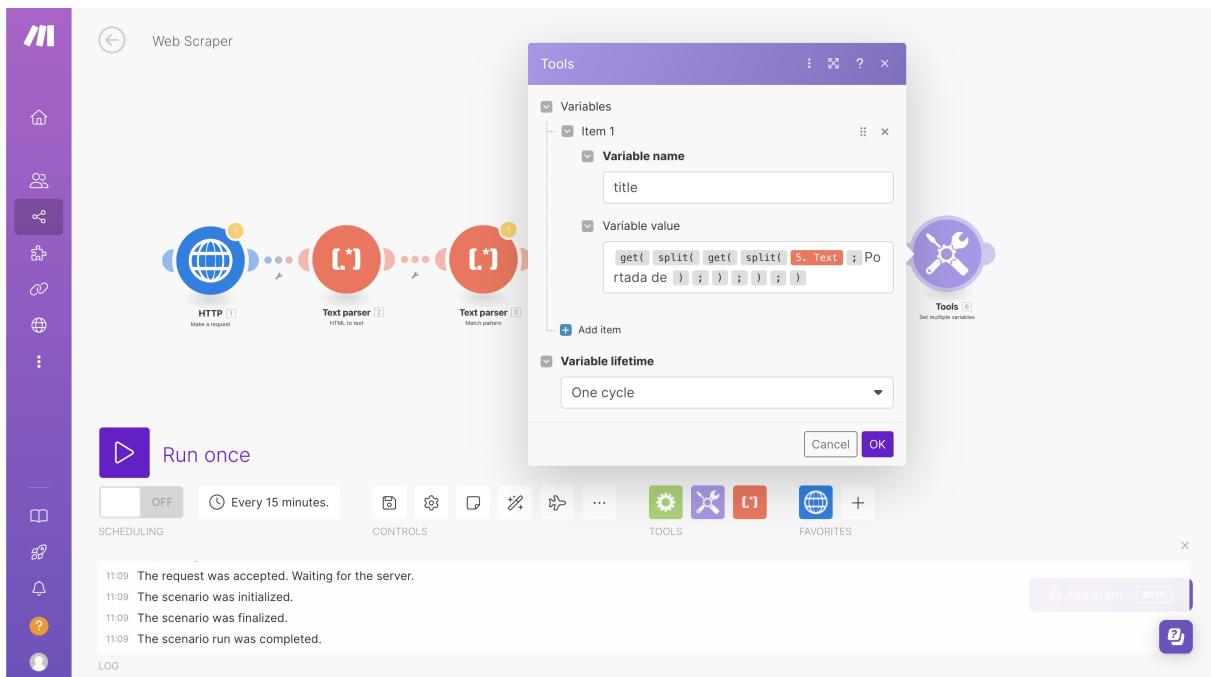


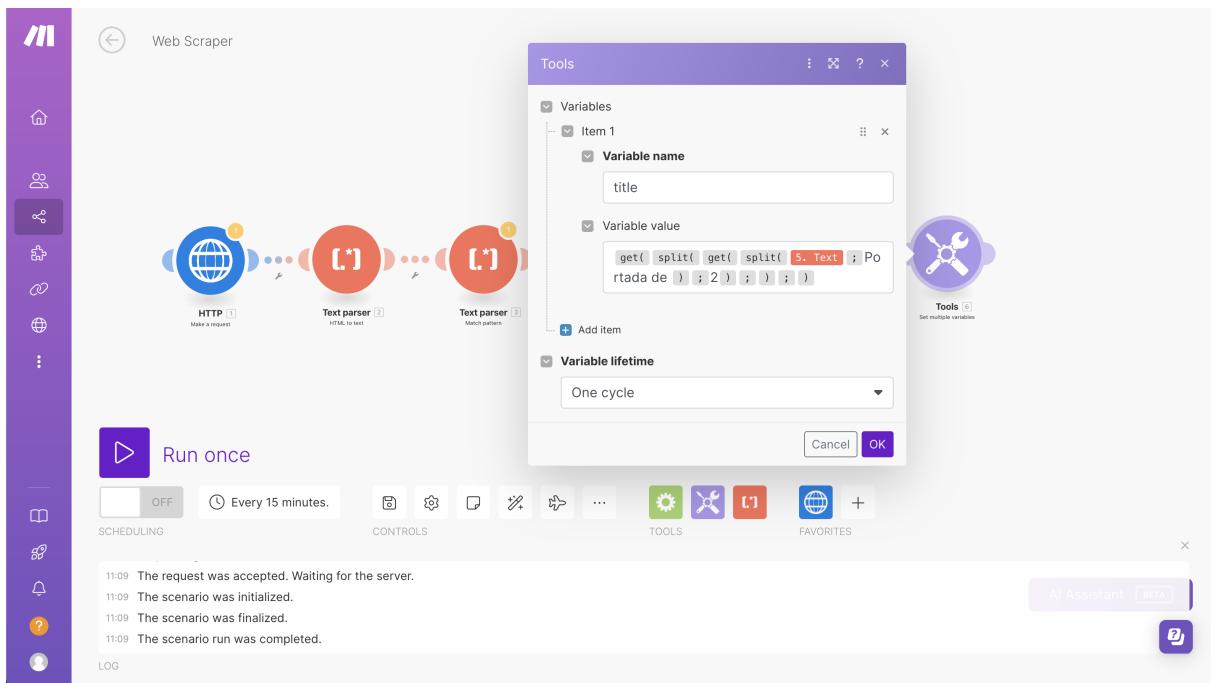
Si observamos bien, esta entre "Portada de " y un corchete de apertura "[", por lo que vamos a atrapar esta información y ver como la colocamos en nuestra formula.

La primera implementación será introducir este texto entero en la formula, y lo haremos después del segundo split, ya que vamos a "trocear un trozo" de este texto.

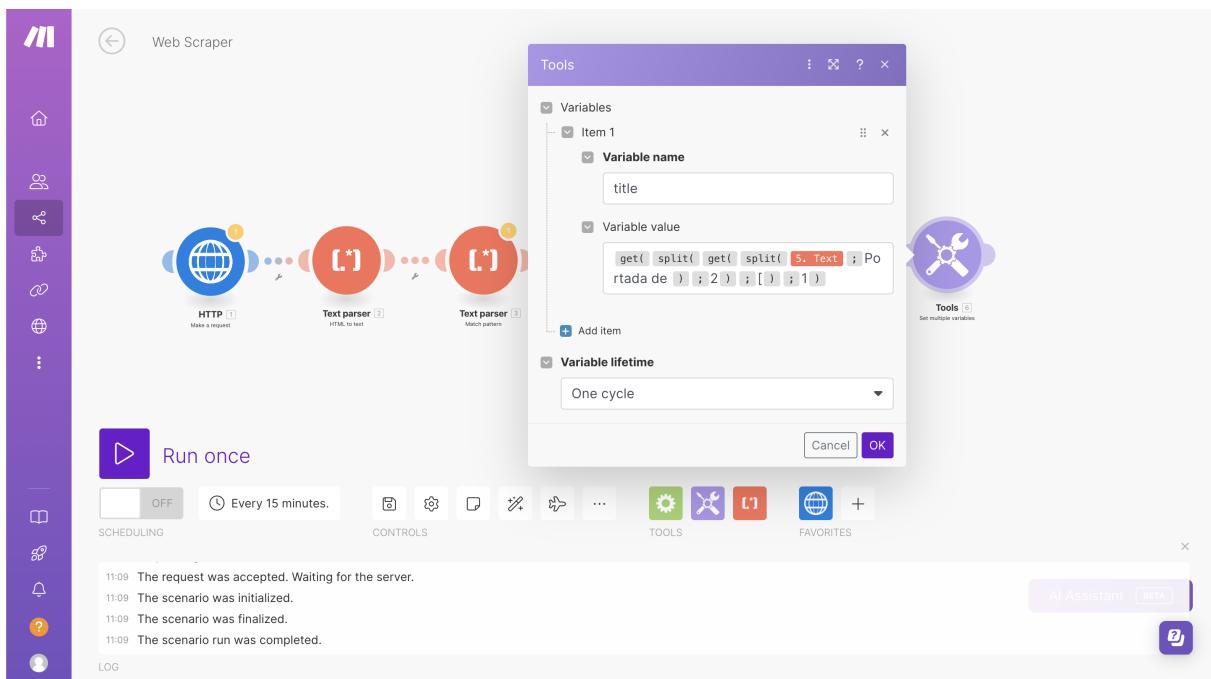


Lo segundo será indicar el texto de apertura de la horquilla con "Portada de " (no olvidéis el espacio para que no lo incluya en el recorte) y un 2 después del punto y coma para nos coja la segunda (es decir, excluimos esa frase y nos quedamos con la continuación).





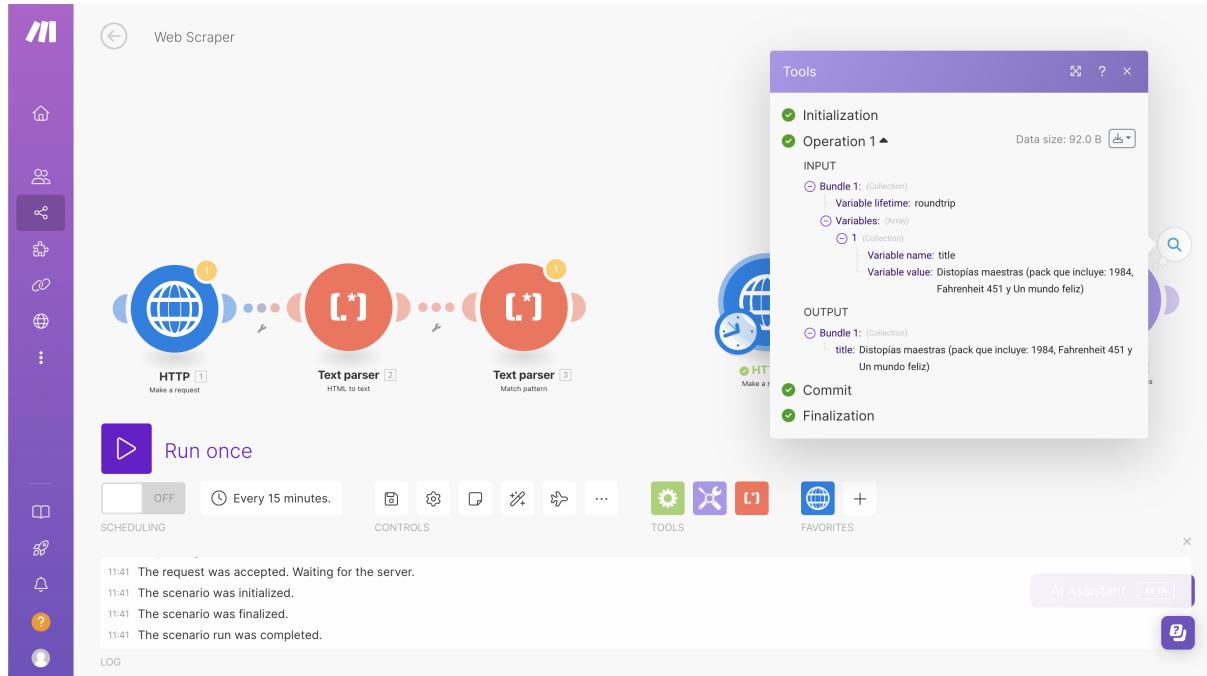
Y lo tercero, implementar el cierre con el "[", indicando con un 1 de igual manera que nos recupere la primera, al contrario que con la frase queremos quedarnos con lo que hay antes de este simbolo y no con el simbolo. Por eso lo hemos definido como una horquilla, porque queremos lo que hay después de "Portada de " y antes de "[".



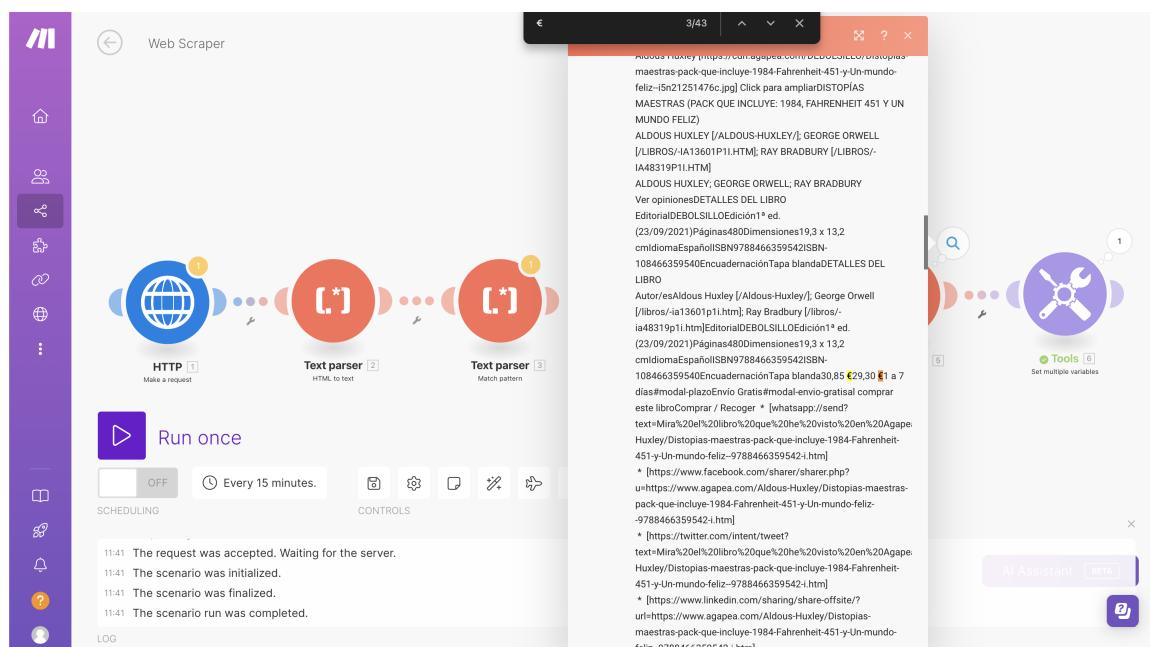
**Get** obtiene el contenido y **split** divide un texto. Cuando obtenemos el trozo de "Portada de ", tenemos tanto "Portada de " como el **texto que lo continua**, así

que de estos dos elementos queremos el 2. Y cuando obtenemos el trozo de **texto anterior** y “[“ queremos el 1. De esta manera combinando los dos gets con sus split hemos realizado la horquilla.

Si probamos nuestra automatización, veremos como ha sido capaz de recuperar el título que queríamos.



- Vamos a hacer otra variable que incluya el precio de este libro, así que vamos a buscar dónde se encuentra en el texto y seguiremos el mismo procedimiento.



Como podemos ver, el precio se encuentra entre “€” y “€” (cuidado con los espacios), ya que con obtener la cantidad nos valdría. Así que vamos a hacer exactamente lo mismo en una nueva variable llamada **price**. En este caso hemos elegido la 3 coincidencia, ya que en las páginas aparece un precio previo de los envíos, de esta forma nos saltamos esa coincidencia y vamos a la siguiente.

Es muy importante que detectéis patrones y que elijáis aplicaciones web que estén construidas de manera similar, ya que si no será una tarea imposible de automatizar.

Si probamos nuestro flujo ahora, veremos como nos devuelve también el precio del libro elegido:

The screenshot shows the Make Web Scraper interface. On the left, there's a sidebar with various icons. The main area displays a workflow: an 'HTTP' block (labeled '1') followed by two 'Text parser' blocks (labeled '2' and '3'). The first 'Text parser' is set to 'HTML to text' and the second to 'Match pattern'. To the right, a 'Tools' panel is open, showing the 'Initialization' and 'Operation 1' steps. Under 'INPUT', it shows a 'Bundle 1' collection with one item. This item has variables: 'Variable name: title' with value 'Distopias maestras (pack que incluye: 1984, Fahrenheit 451 y Un mundo feliz)' and 'Variable name: price' with value '29,30'. Under 'OUTPUT', it shows another 'Bundle 1' collection with one item, containing the same title and price. Below these are 'Commit' and 'Finalization' buttons. At the bottom, there's a log section with several messages indicating the process flow.

### 13. Por último vamos a recuperar el código ISBN, así que haremos exactamente lo mismo en una nueva variable.

Detectamos donde está el código.

This screenshot shows the updated Make Web Scraper interface. The workflow now includes an additional 'Text parser' block (labeled '4') after the second 'Text parser' block. This new block is also set to 'Match pattern'. The 'Tools' panel on the right shows the expanded output: it now includes the previous 'title' and 'price' variables, along with a new 'isbn' variable. The 'isbn' variable has the value '9788466359542'. The log at the bottom continues to show the execution status.

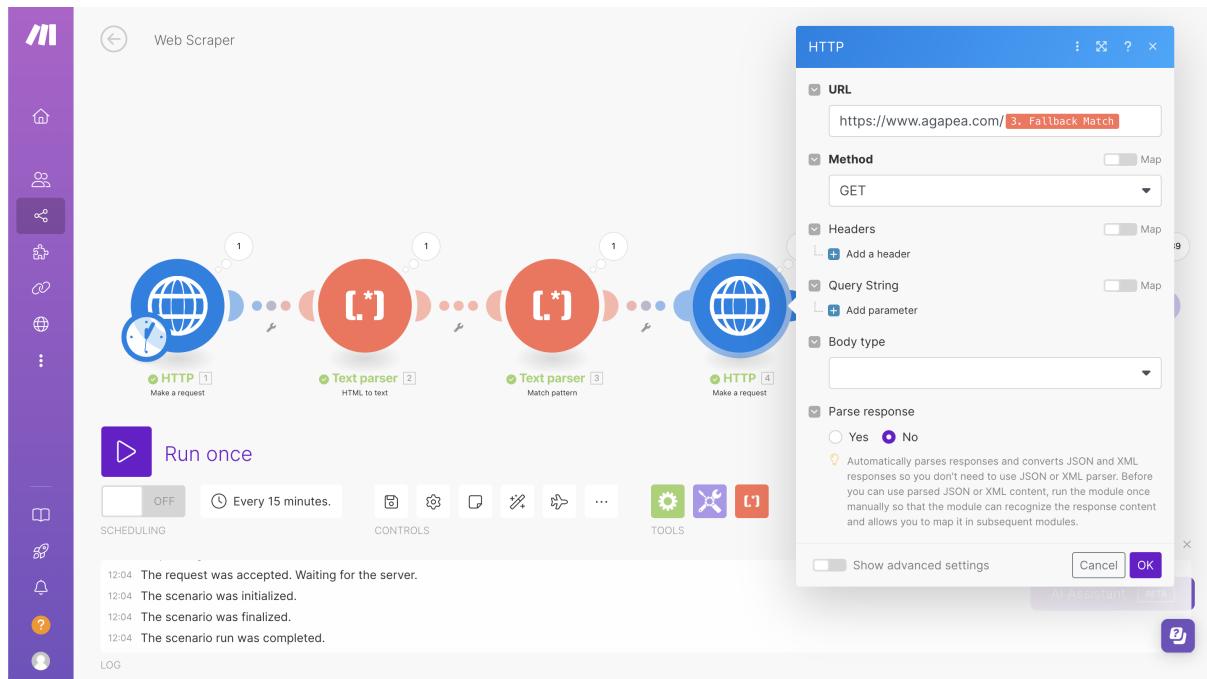
Definimos una nueva variable con la formula para detectar lo que hay entre "ISBN" y "ISBN".

Y probamos:

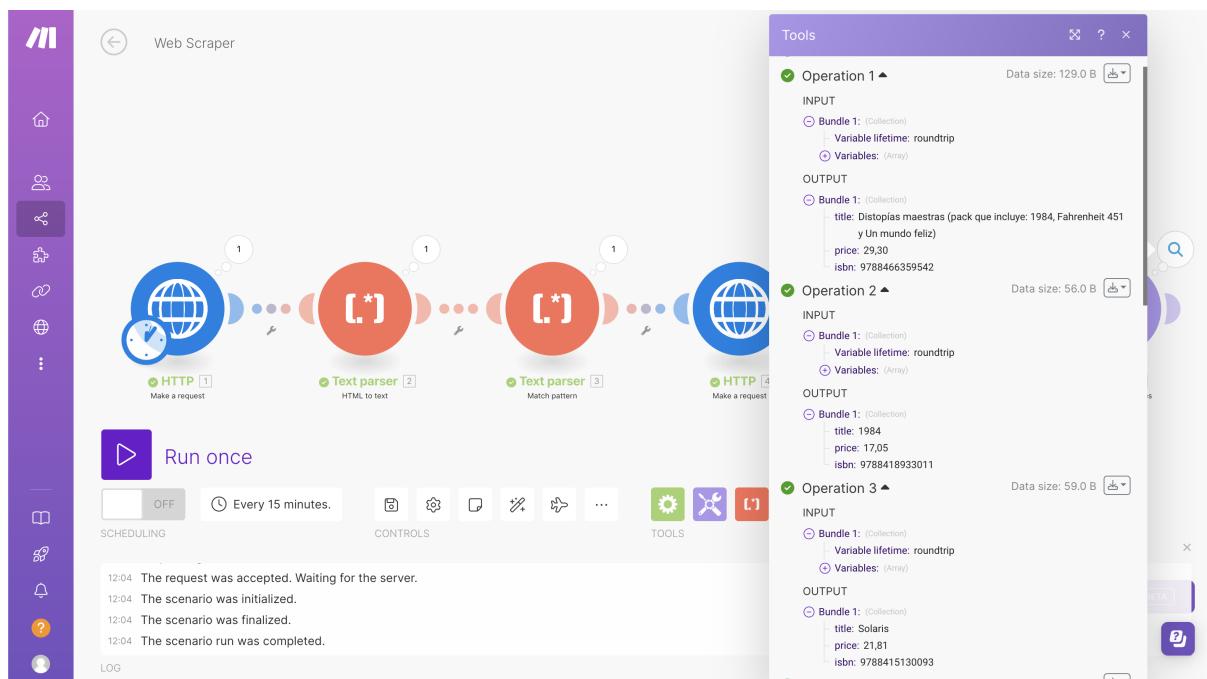
Lo tenemos, de esta forma esta recuperando automáticamente en base a formulas de detección los datos que queremos de un libro en concreto.

14. Vamos a enganchar nuestro segundo HTTP request y automatizar todos los resultados para que no nos haga solamente una petición, si no por todos los links de libros que encontramos.

Para ello, editaremos la la segunda petición **HTTP** y justo detrás de [agapea.com](https://www.agapea.com) pondremos el resultado de cada una de las coincidencias, que serán cada uno de los links.



Con todos los módulos enganchados, el resultado del último modulo ahora nos arrojará todos los resultados correctamente en diferentes operaciones.



15. Así que lo que quedaría sería almacenar los resultados en algo tangible, por lo que haremos una hoja de cálculo en nuestro Google Drive a través de **Google Sheets** e indicaremos en esa nueva hoja 3 columnas con los titulos de la información que queremos almacenar. En nuestro caso, titulo, precio e ISBN.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Título	Precio	ISBN												
2	2	Presto	9781234567890												
3															
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															

Añadimos un último modulo a la automatización con **Google Sheets** y la opción **Add a Row** para añadir una nueva fila.

Google Sheets

Connection: My Google connection (anton...)

Choose a Method: Select by path

Choose a Drive: My Drive

Spreadsheet ID: Click here to choose file

Unformatted: No

Show advanced settings

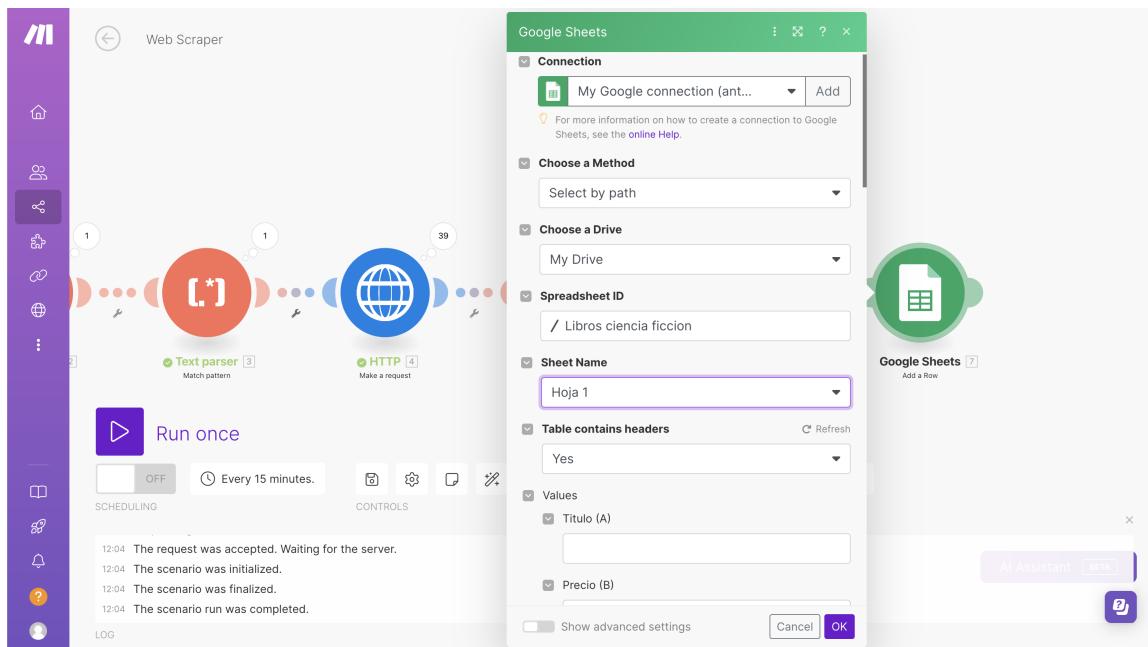
Cancel OK

AI Assistant BETA

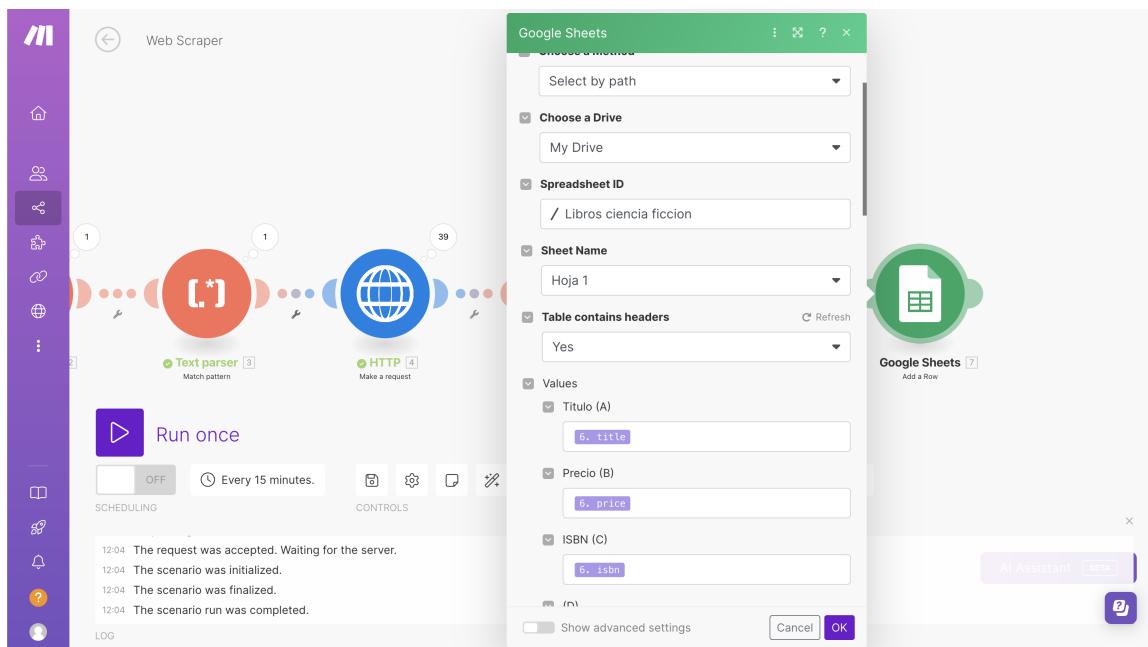
LOG

12:04 The request was accepted. Waiting for the server.  
12:04 The scenario was initialized.  
12:04 The scenario was finalized.  
12:04 The scenario run was completed.

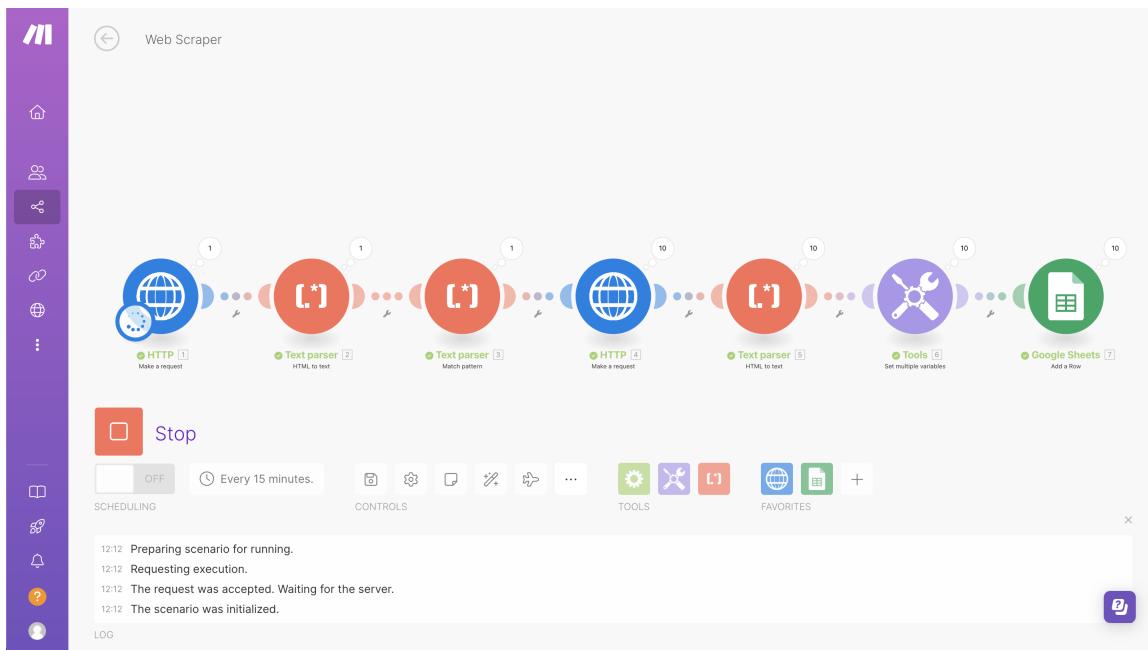
Seleccionamos nuestro documento a través de Google Drive y elegimos tanto el documento como la hoja correspondiente.



Y en los valores de las cabeceras que previamente hemos indicado en el documento, asignamos valores para almacenarlos en nuevas filas, que serán los resultados de nuestras variables en cada una de las operaciones:



Si todo ha ido bien, probaremos a lanzar nuestra automatización:



Y cuando termine de recabar la información en cuestión de segundos tendremos el siguiente resultado en nuestra hoja de cálculo:

The screenshot shows a Google Sheets document titled "Libros ciencia ficción". The spreadsheet has three columns: "Título" (Title), "Precio" (Price), and "ISBN". The data consists of 33 rows, each representing a book entry. The first few rows are:

	Título	Precio	ISBN
1	Título		
2	Distopias maestras (pack que incluye: 1984, Fahrenheit 451 y Un mundo feliz)	29,30 €	9788466359542
3	1984	17,05 €	9788418933011
4	Solaris	21,81 €	9788415130093
5	La máquina del tiempo	11,35 €	9788491811442
6	Hyperion	22,70 €	9788466658034
7	Crónicas marcianas	18,00 €	9788445006740
8	OBRA MAESTRAS DE JULIO VERNE	16,10 €	9788417782436
9	2001. Una odisea espacial	9,40 €	9788497599290
10	Clásicos bilingües. La isla misteriosa (español/inglés)	4,70 €	978846731972
11	UNA ODISEA ESPACIAL. LA SAGA COMPLETA	14,21 €	9788466343220
12	¿Sueñan los androides con ovejas eléctricas?	14,21 €	978845007723
13	De la Tierra a la Luna	11,35 €	9788408215103
14	La guerra interminable	12,00 €	9788490709269
15	Dune (Nueva edición) (Las crónicas de Dune 1)	11,35 €	9788466353779
16	La mano izquierda de la oscuridad	18,00 €	9788445009314
17	Un mundo feliz	10,40 €	9788497594257
18	Viaje al centro de la Tierra	12,30 €	9788467050660
19	Soy leyenda	16,00 €	9788445006764
20	El juego de Ender (Saga de Ender 1)	16,10 €	9788420434193
21	La guerra de los mundos	10,40 €	9788408210061
22	Trilogía de la Fundación	14,21 €	9788499083209
23	Fahrenheit 451	9,40 €	9788490321478
24	Ubik	17,05 €	9788445008232
25	Yo, robot	10,40 €	9788435918364
26	Fahrenheit 451	17,05 €	9788445006771
27	GUERRA DE LOS MUNDOS LA	9,40 €	9788415215684
28	Dune (Nueva edición) (Las crónicas de Dune 1)	13,25 €	9788466350954
29	El problema de los tres cuerpos: Primer volumen trilogía	19,86 €	9788466859734
30	La mano izquierda de la oscuridad	14,21 €	9788445009649
31	Crónicas marcianas	11,35 €	9788445008782
32	Guía del autoestopista galáctico	12,25 €	9788433961037
33	Va m...nt	12,25 €	9788433961230

De esta manera, podremos automatizar cualquier recuperación de información de forma automática en cuestión de segundos. Ten en cuenta que cada aplicación web es un mundo, por lo que tendremos que investigar de qué manera podemos automatizar cada caso de la misma manera que hemos hecho con este ejemplo. Empieza por una aplicación web sencilla y poco a poco podrás ir accediendo a cualquier caso que necesites.