

Documentación de Rocketbot Studio



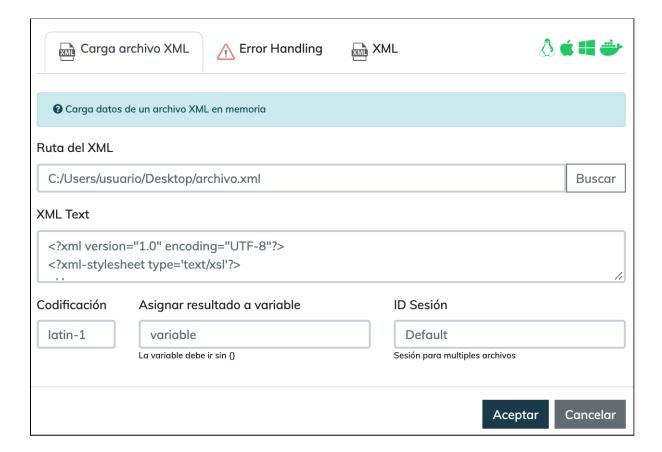
Con este módulo podrás realizar:

- Carga archivo XML
- Insertar Nodo
- Obtener Nodo
- Modificar Nodo
- Borrar Nodo
- Guarda archivo XML
- Termina sesión XML



1. Carga archivo XML Nos permite cargar un XML en memoria para trabajar con él mediante los demás comandos del módulo. Podemos cargar un archivo XML o el texto directo en **XML Text**, se debe indicar la codificación y además podemos retornar en una variable los datos del archivo en formato json. Tener en cuenta que esta variable no debe llevar llaves {}.

Si requerimos trabajar con varios XML, debemos identificarlos mediante un ID en el input **ID Sesión**.



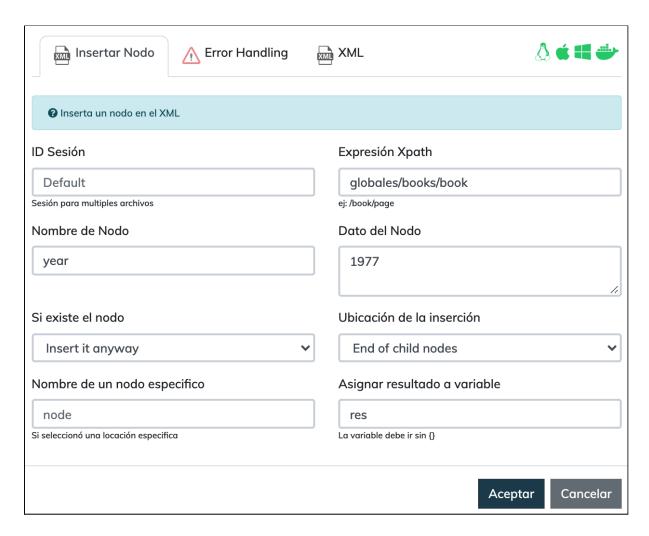
- **2. Insertar Nodo** Nos permite insertar un nodo en el XML que se encuentra en memoria, describiremos la información a llenar:
- ID Sesión: Si estamos trabajando con más de un archivo, debemos especificar su ID de sesión, indicado en el comando anterior (<u>Cargar</u> <u>archivo XML</u>)
- Expresión Xpath: Path del nodo a donde queremos insertar, ej: globales/books
- Nombre de Nodo: Nombre del nodo que queremos agregar.
- Dato del Nodo: Información que queremos agregar dentro del nodo.
- Si existe el nodo: Indicar la acción que queremos realizar si el nodo ya existe en el XML, las opciones son:
 - o Insert it anyway: Insertará el nodo si o si
 - Skip it: Omitir
 - Overwrite it: Sobreescribir
- **Ubicación de la inserción:** Podemos indicar en qué parte específica queremos agregar el nodo, las opciones son:
 - **Beginning of child nodes:** Al inicio de los nodos hijos.
 - End of child nodes: Al final de los nodos hijos.
 - **Before specific child node:** Antes del nodo hijo específico.
 - o After specific child node: Después del nodo hijo específico.
- Nombre de un nodo específico: Si en la opción anterior elegimos Before specific child node o After specific child node, acá debemos ingresar el nombre del nodo específico.
- Asignar resultado a variable: Nos retornará el XML modificado en formato ison.

Ejemplo:

Cargaremos el siguiente XML:



Agregaremos un nuevo nodo dentro de <book> llamado <year>, el cual tendrá como valor 1977, si el nodo ya existe elegiremos **Insertarlo de todas formas** (Insert it anyway) y como posición será **Al final de los nodos hijos** (End of child nodes), retornaremos la respuesta a la variable **{res}**



Al ejecutar, en la variable {res} obtendremos lo siguiente:



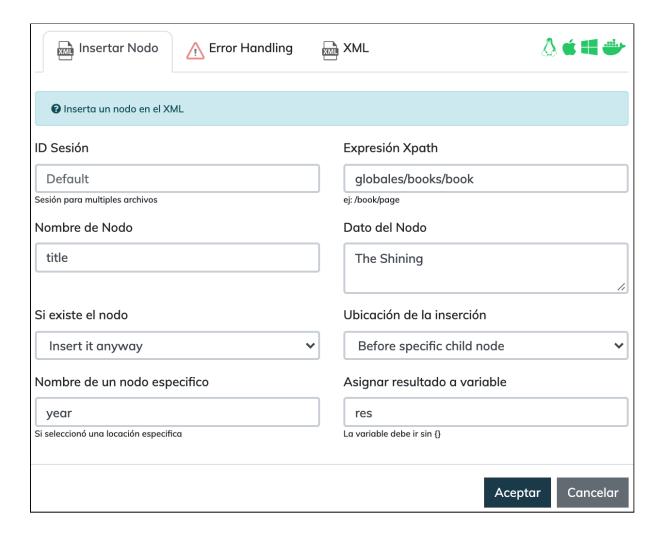


Para poder ver cómo queda en un archivo, debemos utilizar el comando Guarda archivo XML, al ejecutarlo, quedará de esta forma:

```
<pre
```

Ahora agregaremos otro nodo dentro de <book> llamado <title> pero esta vez queremos que quede antes del nodo <year> recientemente agregado. Si el nodo ya existe elegiremos Insertarlo de todas formas (Insert it anyway) y como posición será Antes del nodo hijo específico (Before specific child node), en este caso en el input Nombre de un nodo específico debemos indicar el nombre del nodo del cual se insertará el nuevo dato arriba de éste, que será en este caso <year>, y retornaremos la respuesta a la variable {res}





Al ejecutar, en la variable {res} obtendremos lo siguiente:

Al guardarlo en un archivo lo veremos de esta forma:

*Se agregó el nuevo nodo <title> antes del nodo <year>

Si queremos agregar nodos a un hijo específico, podemos indicar su posición mediante números, por ejemplo:

Agregaremos el tag <year> pero en el segundo hijo <book> (author: Edgar Allan Poe), en este caso la Expresión Xpath sería: globales/books/book[2] donde [2] es la posición, la cual corresponde al segundo nodo <book>.

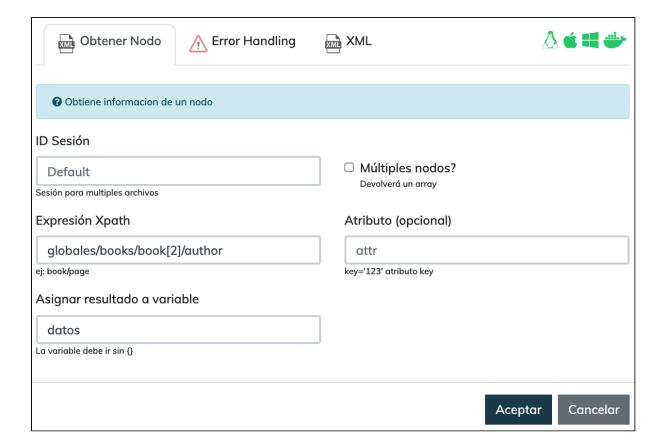


3. Obtener Nodo Nos permite obtener la información de un nodo.

Debemos indicar el ID de sesión si estamos trabajando con más de un XML y el path del nodo del cual queremos extraer la información, si necesitamos obtener más de uno, podemos marcar el check de **Múltiples nodos** para que nos devuelva una lista. También podemos indicar un atributo en el caso de ser necesario y agregar una variable donde retornaremos la información. Tener en cuenta que esta variable no debe llevar llaves {}.

Ejemplo:

En este caso solo estamos extrayendo la información del segundo nodo <book> específicamente el <author>.

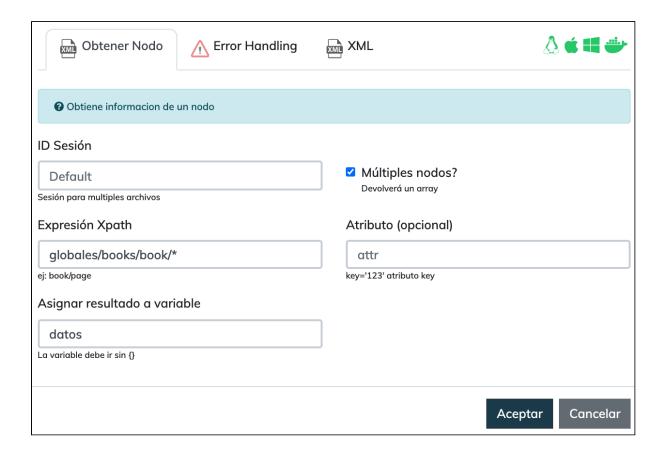


Obtendremos lo siguiente:

datos	Edgar Allan Poe
-------	-----------------

Si queremos obtener todos los nodos <book>, del siguiente XML

debemos indicar la ruta del nodo y agregar /* lo que significa que traiga toda la información dentro de <book>:

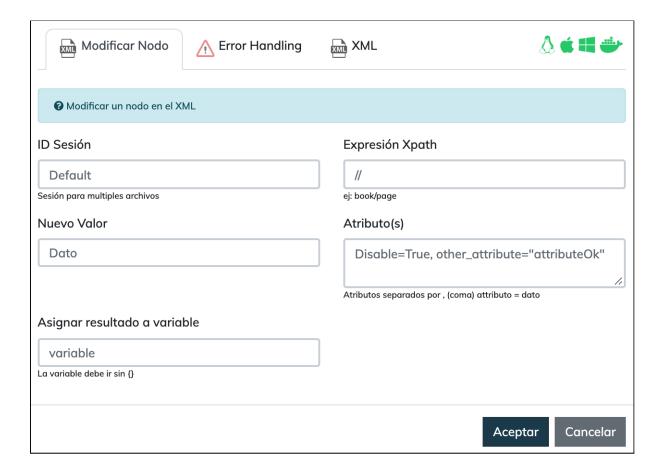


Obtendremos lo siguiente:

datos	['Stephen King', 'Edgar Allan Poe', '1977']
-------	---



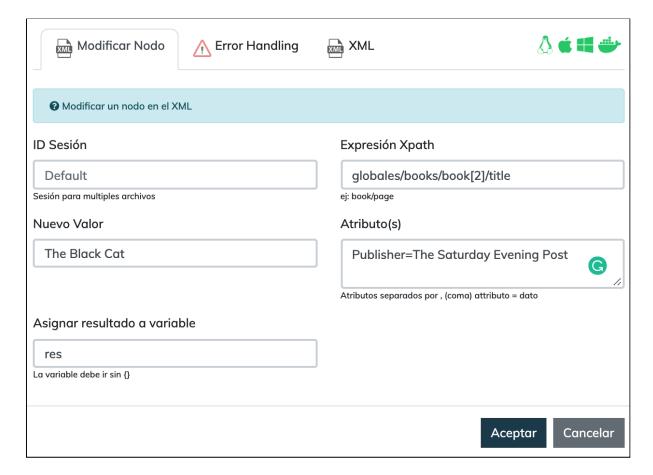
4. Modificar Nodo Nos permite modificar la información de un nodo agregando la expresión Xpath, el nuevo valor del nodo y además podemos indicar atributos de ser necesario y una variable donde retornar la información modificada. Tener en cuenta que esta variable no debe llevar llaves {}.



Ejemplo con atributos:

Tenemos el siguiente XML y modificaremos el nodo <title> del segundo nodo hijo <book> con el valor "The Black Cat" y además agregaremos un atributo llamado **publisher** con el valor "The Saturday Evening Post".

El comando quedaría de esta forma:





Al ejecutar obtendremos lo siguiente en nuestra variable:

```
res

{"configuration": {"globales": {"books": {"book": [{"author": "Stephen King", "title": "The Shining", "year": "1977"}, {"author": "Edgar Allan Poe", "title": {"@Publisher": "The Saturday Evening Post", "#text": "The Black Cat"}}]}}} 

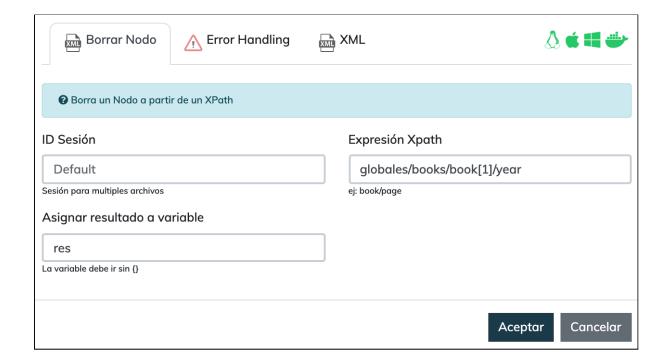
$\textstyre{\mathbb{contraer (33 caracteres)}}
$
```

y al guardar, obtendremos lo siguiente en el archivo:

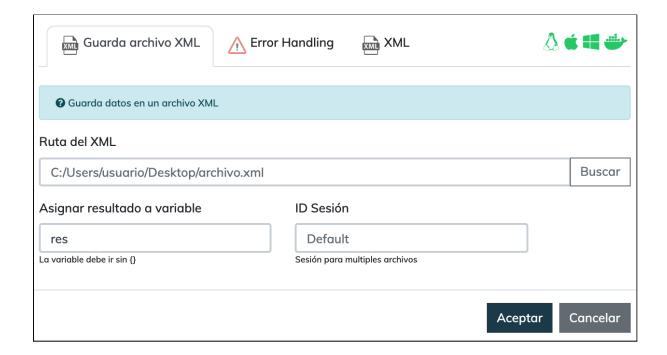
5. Borrar Nodo Nos permite eliminar un nodo indicando su ubicación mediante la Expresión Xpath, podemos asignar el valor modificado a una variable. Tener en cuenta que esta variable no debe llevar llaves {}.

Ejemplo:

En este caso estamos eliminando del primer nodo hijo <book> el nodo <year>



6. Guardar archivo XML Nos permite guardar todos los cambios realizados en memoria a un archivo .xml, debemos indicar la ruta donde lo queremos guardar, una sesión en caso de que estemos trabajando con más de un XML, además podemos asignar el valor modificado a una variable donde obtendremos True si el archivo se pudo guardar correctamente o False en caso contrario. Tener en cuenta que esta variable no debe llevar llaves {}.



7. Termina sesión XML Nos permite terminar la sesión iniciada mediante el comando Carga archivo XML, si no indicamos un nombre, cerrará la sesión por defecto.

