

# Cómo obtener los identificadores web

Para poder controlar las páginas web, necesitamos acceder a los identificadores de los elementos para así especificar en cuál de ellos queremos realizar acciones. En este documento veremos el paso a paso para lograrlo.

Tomaremos como ejemplo la página [RPA Challenge](#)

## Inspeccionar la página

Primero debemos inspeccionar la web para visualizar el código fuente, para esto podemos dar click derecho en cualquier lugar y seleccionar la opción **Inspeccionar**, también podemos utilizar los atajos de teclado CTRL + SHIFT + C o F12 para Windows o Command + SHIFT + C para macOS.

Veremos una ventana como esta

 *Consola del navegador*

Clickeamos la flecha que se encuentra en la esquina superior izquierda y una vez que esté de color azul, significa que ya podemos marcar cualquier elemento de la web. Seleccionamos el elemento deseado, y automáticamente se resaltará en azul en el código, abarcando todo el contenido del elemento elegido.

 *Seleccionando input*

Los elementos pueden tener diferentes tipos de identificadores (id, class, name, etc.), para más detalles, revisa la siguiente entrada [Tipos de Identificadores Web](#).

## Elemento Dinámico

Podemos ver que el elemento marcado tiene **id**, **name** y **class**

```
<input _ngcontent-c2="" ng-reflect-name="labelCompanyName" id="jnPCa" name="jnPCa" class="ng-untouched ng-pristine ng-invalid">
```

Algo bastante común es que algunos de éstos sean dinámicos. Si recargamos la página y revisamos nuevamente, veremos que, tanto el id como el name cambiaron

```
<input _ngcontent-c2="" ng-reflect-name="labelCompanyName" id="UDONR" name="UDONR" class="ng-untouched ng-pristine ng-invalid">
```

Esto sucederá cada vez que la página se recargue o se vuelva a abrir, por lo que nuestro robot fallaría si indicamos uno de estos datos, el **Xpath** tampoco nos serviría ya que como el id es dinámico, también sería diferente

```
//*[@id="jnPCa"]
```

```
//*[@id="UDONR"]
```

El **fullXpath** tampoco serviría, ya que si nos fijamos, los elementos cambian de posición

 *Posición inicial del elemento*

 *Posición del elemento cambiada*

Una forma óptima de controlar esto, es crear un Xpath centrándose en algún atributo específico, más detalles acá: [Xpath para elementos dinámicos](#).


Si nos fijamos en el elemento, veremos que lo único que no cambia es el siguiente atributo

```
ng-reflect-name="labelCompanyName"
```

Podemos entonces crear un Xpath utilizando esa información, de esta forma

```
//input[@ng-reflect-name="labelCompanyName"]
```

Para saber si está correcto, podemos probarlo en el mismo navegador, donde inicialmente abrimos la consola, seguimos en **Elements** y presionamos CTRL + F si estamos en Windows o Command + F si estamos macOS, esto habilitará la casilla de búsqueda (1) y pegaremos el xpath previamente mencionado, al hacerlo nos marcará el elemento tanto en el código (2) como en la vista web (3)

 (1) *Input de búsqueda*


(2) *Elemento marcado en código*

(3) *Elemento marcado en la vista web*

Si está todo correcto, entonces ese identificador es el que debemos utilizar en los comandos de Rocketbot para indicar el elemento con el que queremos trabajar.

## Elemento No Dinámico

Ahora veremos el caso de un elemento donde su posición e identificadores no cambian, en este ejemplo tomaremos el botón para descargar el archivo Excel

 *Botón descarga*

El elemento es el siguiente, si recargamos y lo volvemos a revisar, seguirá teniendo la misma información

```
<a _ngcontent-c1="" class=" col s12 m12 l12 btn waves-effect waves-light  
uiColorPrimary center" target="_blank"  
href="./assets/downloadFiles/challenge.xlsx"> Download Excel <i _ngcontent-  
c1="" class="material-icons right">cloud_download</i></a>
```

Obtenemos el fullXpath y lo utilizamos en el comando web de Rocketbot donde necesitamos indicar el elemento

```
/html/body/app-root/div[2]/app-rpa1/div/div[1]/div[6]/a
```