

---

## Reto 3 Nodel

Algunas veces la data requerida a extraer no está disponible en alguna api o link oficial de descarga, por eso en algunas ocasiones acudimos al **Scraping** de páginas web para lograr este reto. Una manera de realizar esto es usando el api de **SELENIUM** para python, la cual nos permite simular una navegación web, interactuar con la página y bajar la data requerida.

Esta página [link](#) está orientada a ambientar a las personas con el uso del mouse en sus computadoras. El desafío en este reto es mediante el uso de python y SELENIUM, crear un script que permita avanzar en los diferentes niveles de la página mencionada interactuando con los elementos y siguiendo las órdenes que se muestran. La página tiene 41 niveles. (si se logra completar los 41 niveles se obtendrán 2 puntos adicionales en la evaluación).

Tip. Cada página tiene una orden o petición específica que se debe hacer, se recomienda crear un diccionario o sistema de reglas dependiendo de la página en donde se encuentra para setear la orden a ejecutar. Ejemplo: Pág. 22, clickear todos los links y continuar, Pág. 29 doble clic en todas las imágenes y continuar.

**Autor:** Johao Villarroel

**Fecha de Entrega:** 05 diciembre de 2022

---

# Índice de contenido

Acrónimos .....	2
Caso de estudio .....	2
Requerimientos .....	2
Procedimientos .....	2

## Acrónimos

## Caso de estudio

Se desea validar los conocimientos con una serie de retos, en el presente reto se validara el uso del api Selenium para web scraping con el objetivo de obtener datos y poder automatizar un proceso, en este caso será de resolver niveles de una pag web.

## Requerimientos

- Python 3.9
- Librería selenium

## Procedimientos

Se realizará un driver el cual será Choome este se bajara automáticamente con el código con el objetivo de automatizar el proceso el cual se podrá ver en pantalla, en este creamos opciones para que inicie maximizado y las extensiones que se tengan instaladas no se usen para evitar conflictos.

```
# Opciones de navegación
options = Options()
options.add_argument("start-maximized")
options.add_argument('--disable-extensions')

driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()), options=options)
```

Iniciaremos el navegador con la página web con la que se trabajara

```
# Inicializamos el navegador
driver.get('http://www.pbclibrary.org/raton/mouserice.htm?')
```

Luego de acabar con los retos debería darnos una pantalla como la siguiente para verificar que se a cumplido con todos los niveles.

**¡Felicidades!** Has llegado al final de los ejercicios del ratón. Eso no era tan difícil, ¿cierto?

