

Ejercicio backend

Se realizó un código el cual permite ingresar a la página de la empresa Consultoría Global, y mediante una automatización con Java y Selenium (utilizando WebDriver), accionar de tal manera en dicha página.

Para empezar, se realizaron todos los *imports* necesarios para que el programa funcione, y previamente se instaló Selenium, Java y se descargó el driver de Google Chrome. Luego de tener el entorno correctamente configurado, se desarrolló el código necesario para completar el ejercicio. Dentro de la clase "Main", se encuentran tres métodos; el método **main**, que se refiere al *entry point* de la aplicación, y es donde se encuentra todo el desarrollo del código. Por otro lado, se encuentra el método **Esperar**, el cual permite la interrupción del programa durante los segundos indicados. Esto se realizó con el fin de que la app, no se ejecute de manera tan veloz, y de tiempo a mirar el paso a paso de su funcionamiento. Y por último, el método **DetectarElemento** logra (como dice su nombre) detectar un elemento de la página web y permite interactuar con el.

Si bien el código no es tan extenso, se utilizó un manejo de excepciones en gran parte del programa, con el fin de que cualquier posible error en tiempo de ejecución, no detenga abruptamente la app y esta pueda finalizar correctamente.

Por último, y en relación al manejo de excepciones, se hace uso de las clases **BufferedWriter** y **FileWriter** para así poder manejar un archivo de texto (.txt) y poder tener registro de los posibles errores que pudo llegar a experimentar el programa.

Facilidades y dificultades

Tuve muchas dificultades para realizar este programa, ya que no conocía ningún término de los cuales se plantean en el ejercicio. Al principio estaba bastante perdido y no sabía de qué manera podía arrancar el programa, así que fui investigando parte por parte cada concepto. Primero instalé Java, luego instalé Selenium e investigué qué era exactamente (es decir, una librería). Instalar Selenium y configurar todo el entorno de trabajo fue de lo que más me costó, ya que, en lo que voy de la tecnicatura que estoy estudiando, no tuve que instalar

muchas librerías externas (solamente *pygame* para usar en Python). Toda la configuración del proyecto (instalar Java, agregarlo al *path* en variables de entorno, relacionarlo con Selenium, etc) fue un trabajo que me llevó varias horas, pero que finalmente, luego de muchas pruebas, errores, de instalar y desinstalar varios archivos, pude lograr. Sin embargo, cabe destacar que no pude implementar **Spring boot**; si bien estuve investigando y realizando pruebas con diversos proyectos, no pude implementarlo en esta app. Por lo que investigué, tendría que modificar un archivo llamado **pom.xml** el cual no contenía mi proyecto por no ser de tipo **Maven**.

Por otro lado, pensar la lógica que el código tenía que realizar no me resultó muy complicado. Si bien nunca había programado en Java, actualmente en la facultad estoy viendo C#, lenguaje con el cual tiene muchas similitudes. Poseer conocimientos de POO me permitió entender muy fácilmente la creación de objetos (por ejemplo, de tipo **WebDriver**) y el uso de sus propiedades o métodos. Además, pude generar un código legible, el cual contiene dos métodos estáticos que me permitieron modularizar el código de manera óptima. Por último, ya que estos programas son muy propensos a generar algún error inesperado, decidí incluir un manejo adecuado de excepciones (*try catch* y *try catch finally*).

Referencias

- <https://chat.openai.com/>
- <https://chromedriver.chromium.org/>
- <https://www.selenium.dev/downloads/>
- <https://www.youtube.com/watch?v=W86KTBSiX2o&t=3070s>

Ejercicio front-end

Este código ejecuta una app que permite abrir una página web y muestra un sencillo organizador de tareas. Permite agregar tareas (siempre y cuando ese campo no esté vacío), marcarlas como completadas y eliminarlas si ya se completaron. También se utilizó una base de datos para guardar las tareas. Aunque el ejercicio no

lo pedía, se agregó, ya que permite que la app sea más práctica y funcional para el usuario.

Se hicieron todos los *imports* correspondientes y también se usó la librería **MaterialUI** para poder crear elementos en la interfaz de la web. Se declararon variables y funciones para poder realizar un correcto C.R.U.D en la app y además, en cada función, se actualiza la base de datos (IndexedDB) en base a la acción ejecutada.

Las funciones principales son 4; **manejarTarea** se encarga de cambiar el contenido del textbox donde se escribe la tarea, **manejarAgregadoDeTareas** tiene como función agregar tareas a la lista de tareas, **manejarCambioEnTareas** detecta si una tarea está o no completa y por último, **eliminarTareasCompletas**, como lo indica su nombre, se encarga de eliminar las tareas que ya se completaron.

Facilidades y dificultades

Como mencioné anteriormente, toda la configuración del entorno me fue un tanto difícil. A pesar de eso, después de instalar varios paquetes, pude instalar React y MaterialUI de manera exitosa. Además, me fue muy complicado entender cómo funciona React y su sintaxis, ya que si bien tengo algunos conocimientos de Javascript, tengo mayores conocimientos en los lenguajes orientados a objetos. Tuve que realizar una gran investigación sobre muchos nuevos conceptos que fueron apareciendo a medida que desarrollé la app, tales como el **await**, el **async**, el manejo de una base de datos, etc. Fueron muchos temas nuevos que investigué y comprendí para implementar en esta app.

Sinceramente, no tuve muchas facilidades para realizar este programa, me costó mucho hacerlo y entenderlo (por eso el código está lleno de comentarios). Es la primera vez que me encuentro con un lenguaje un tanto complejo, ya que para entenderlo, tuve que investigar un poco de Html, Css y Javascript. A su vez, me pareció interesante, ya que cada vez que conozco algo nuevo, ya sea un lenguaje, un concepto, un paradigma distinto de programación, me empeño en entenderlo y en ponerlo en práctica. El día de mañana, no sé con qué nuevos desafíos me voy a encontrar, así que es mejor estar listo para enfrentarlos.

Referencias

- <https://chat.openai.com/>
- <https://nodejs.org/en>
- <https://www.youtube.com/watch?v=pFyAu4R684s&t=278s>