



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2513 - Sección 1 — Tecnologías y Aplicaciones Web

Tarea 1

Actualización: 16 de agosto de 2024

Entrega

- **Fecha y hora:** Martes 20 de agosto del 2024, a las 23:59
- **Lugar:** Repositorio individual en la organización del curso en GitHub

Objetivos

- **Construir** programas usando herramientas para interactuar con aplicaciones web
- **Utilizar** la herramienta *DevTools* para explorar páginas web
- **Producir** documentación efectiva y clara que permita el entendimiento del proceso realizado

Descripción

En esta tarea, podrán adentrarse en el mundo de la automatización web y el análisis de páginas a través de herramientas avanzadas de programación, realizando *web scraping*. El objetivo principal es desarrollar habilidades prácticas en la construcción de programas que interactúan con aplicaciones web, utilizando para ello la biblioteca de [Selenium](#). Una parte crucial de esta tarea será el uso efectivo de las herramientas DevTools, que les permitirá inspeccionar y comprender la estructura de las páginas web.

Para poder desarrollar correctamente esta tarea, contarán con 2 librerías. A continuación se encuentran los *links* de instalación:

- [Selenium](#) (ver la sección *Python*)
- [webdriver-manager](#)

Los objetivos de esta tarea son:

- Desarrollar un controlador de navegador que permita el manejo correcto de interacciones automatizadas con páginas web mediante el uso de un webdriver.
- Aprender a configurar y utilizar el webdriver para interactuar de manera eficiente con elementos web como formularios, botones y enlaces, para realizar tareas específicas como la extracción de datos o la automatización de pruebas web.
- Familiarizarse con la estructura HTML de páginas web funcionales, y utilizar *DevTools* para inspeccionar páginas web.

Problema



Con la conclusión de los Juegos Olímpicos de París 2024, es crucial realizar un análisis detallado de los resultados obtenidos por los diferentes países y atletas. Para ello, se requiere extraer los datos más relevantes relacionados con las medallas obtenidas durante los juegos. Esta tarea implica la automatización de la recopilación de información desde la [página oficial de los Juegos Olímpicos](#), utilizando técnicas avanzadas de *scraping* web. El objetivo es construir un script que permita extraer y almacenar de manera eficiente los datos de medallas, facilitando así el análisis posterior.









Para lograr esto, deberás dirigirte a la [página de resultados](#) para extraer varios datos de la tabla de medallas y medallistas. En específico, deberás extraer los siguientes datos:

1. **Extraer los primero 10 países con más medallas de oro:** Desde la *tabla de medallas*. El csv debe contener los siguientes datos:

COUNTRY;GOLD;SILVER;BRONZE;TOTAL

2. **Extraer los primeros deportes en los que un país ha ganado medallas:** Esta función permite seleccionar los n deportes de la lista de un país específico de la *tabla de medallas* y extraer información detallada sobre los deportes en los que ese país ha obtenido medallas. El csv debe contener los siguientes datos:

SPORT;GOLD;SILVER;BRONZE;TOTAL

12	 Brazil	3	7	10	20	
	Artistic Gymnastics	1	2	1	4	
	Athletics	0	1	1	2	
	Beach Volleyball	1	0	0	1	
	Boxing	0	0	1	1	
	Canoe Sprint	0	1	0	1	
	Football	0	1	0	1	

Para el caso de Brasil, si piden extraer tres deportes, hay que extraer la información de *Artistic Gymnastics*, *Athletics* y *Beach Volley* independiente de la cantidad de medallas obtenidas en la disciplina.

3. **Extraer el primer atleta de un país específico que ha ganado una medalla en un deporte determinado:** Esta función permite seleccionar uno o más países y un deporte. A partir de estos, se debe buscar el primer atleta que haya ganado una medalla en ese deporte. Deberás utilizar los filtros que se encuentran integrados en la página, en específico el filtro por país y el filtro de deportes. La función interactúa con la *tabla de medallas* para extraer la información del atleta. El csv debe contener los siguientes datos:

NAME;CATEGORY;MEDAL;COUNTRY;SPORT

Donde NAME corresponde al nombre del atleta, CATEGORY corresponde a la categoría en la que participa, MEDAL es la letra inicial de la medalla obtenida, COUNTRY el país del atleta y SPORT el deporte.

4. **Extraer los primeros países a partir del criterio de cantidad de medallas:** Deberás utilizar el filtro que viene integrado en la página y seleccionar que se ordene por *Total medals*, luego desde la *tabla de medallas*, extraer la información de los primeros n países. El csv debe contener los siguientes datos:

COUNTRY;GOLDS;SILVERS;BRONZES;TOTAL

5. **Extraer los primeros países a partir del criterio de orden alfabético:** Deberás utilizar el filtro que viene integrado en la página y seleccionar que se ordene por *Alphabetical*, luego desde la *tabla de medallas*, extraer la información de los primeros n países. El csv debe contener los siguientes datos:

COUNTRY;GOLDS;SILVERS;BRONZES;TOTAL

6. **Extraer los principales medallistas de un país en un deporte específico:** Desde la *tabla de medallistas*, deberás utilizar los filtros que se encuentran integrados en la página, en específico el filtro por país y el filtro de deportes. Luego deberás extraer los primeros n elementos de la tabla. El csv debe contener los siguientes datos:

NAME;GOLD;SILVER;BRONZE;TOTAL

7. **Extraer los principales medallistas de un género específico:** Desde la *tabla de medallistas*, deberás utilizar los filtros que se encuentran integrados en la página, en específico el filtro por género. Luego deberás extraer los primeros n elementos de la tabla. El csv debe contener los siguientes datos:

NAME;GOLD;SILVER;BRONZE;TOTAL

Código base

Para facilitar la corrección de esta tarea, se proporcionan los siguientes archivos base:

1. `driver.py`: **[NO MODIFICAR]** Esta clase está destinada a ser utilizada como controlador del driver del navegador. Según la documentación del paquete [webdriver-manager](#), la biblioteca soporta varios navegadores; sin embargo, se recomienda utilizar [ChromeDriver](#) por su compatibilidad y estabilidad.
2. `main.py`: Este script contiene un menú interactivo base para probar sus métodos del archivo `scraper.py`. Pueden agregar más funciones para las pruebas que necesiten realizar.
3. `test.py`: **[NO MODIFICAR]** Este script contiene los test básicos para poder comprobar que su código esta realizando correctamente lo solicitado en el enunciado.
4. `scraper.py`: El propósito de esta clase es realizar la tarea de "scrapear" la información solicitada, utilizando como apoyo la clase `Driver`. Se deben completar los siguientes métodos:

```
class Scraper:

    def __init__(self, chrome: Driver):
        self.chrome = chrome

    def extract_top_10_countries(self) -> list:
        # Completar punto 1

    def extract_top_n_sports_from(self, country: str, n:int) -> list:
        # Completar punto 2

    def extract_first_athlete_from(self, countries: list, sport: str) -> list:
        # Completar punto 3

    def extract_by_total_medals(self, quantity: int) -> list:
        # Completar punto 4

    def extract_by_alphabetical_order(self, quantity: int) -> list:
        # Completar punto 5

    def extract_top_medallists(self, country: str, sport: str, quantity: int) -> list:
        # Completar punto 6

    def extract_top_medallists_gender(self, gender: str, quantity: int) -> list:
        # Completar punto 7
```

[1] Es importante considerar que no es obligatorio emplear todos los métodos disponibles en tu Scraper de la clase `Driver`.

[2] Los test entregados son referenciales, por lo que deben considerar que su código debe ser capaz de poder realizar diferentes búsquedas según las distintas variaciones de los parámetros de los métodos.

Documentación del proceso

Este ítem tiene como propósito comprender el proceso detallado de cómo se llevó a cabo la obtención de datos. En primer lugar, se espera que expliquen cómo se familiarizaron con la composición de la página y

sus respectivas herramientas de desarrollo del navegador (*Dev Tools*). Deben describir los pasos realizados para navegar hasta secciones específicas de la página, así como la ubicación de información relevante dentro de los archivos y componentes.

Posteriormente, es necesario describir por qué esta exploración previa es crucial para poder automatizar la navegación y llevar a cabo la búsqueda de los datos requeridos de manera eficiente.

Para asegurar una correcta entrega de la documentación, se les hará entrega de una plantilla con la estructura que debe seguir su descripción y las preguntas que deben responder.

Entregables

Cada entrega deberá incluir los siguientes archivos:

- El archivo `scraper.py` con el código de la clase completada para realizar el proceso de *web scraping*
- `README.md` con el paso a paso de las acciones realizadas durante el proceso de *web scraping*, así como también de una descripción de que fue lo que extrajeron y cómo lo hicieron.

Rúbrica

A continuación, se describe el criterio de cada uno de los ítem de la rúbrica con la que se evaluará esta entrega. La nota se calcula al 50 % considerando un total de 16 puntos.

- **Clase `Scraper.py` [7 puntos]:** Que pueda inspeccionar el documento HTML de la página y que sea capaz de buscar y extraer la información solicitada, además de retornar la información extraída en el formato solicitado.
- **Test cases [7 puntos]:** Los códigos serán revisados por test cases, por lo que es importante que cumplan con la declaración de parámetros que recibe y retorna cada función.
- **Comprensión y documentación del proceso [5 puntos]:** Que se explique de manera clara y ordenada el paso a paso de la realización de la tarea. Comenzar con el comando que se debe ejecutar para dar inicio al *web scraping* y con la explicación del funcionamiento de tu código. Luego comentar el análisis y familiarización de la página web, seguir con como este se relaciona con el uso de Selenium y la utilidad de tener un controlador web. Finalmente, terminar con las sugerencias que te hubieran ayudado a realizar de manera más fácil el scraping.

Consideraciones importantes

- Recuerden seguir el formato de los csv de cada punto, en caso contrario, los test fallarán.
- Esta prohibido hacer uso de request mediante los params de URL, el incumplimiento de esto resultará en un 1 a su tarea, sin posibilidad de corrección.
- Se les entregará una carpeta con los archivos .csv base, para que se puedan guiar en el formato esperado de estos.
- El lenguaje por defecto de la página es inglés. Además no debe modificar el texto extraído de la página.
- En los casos de que se encuentren menos datos de los solicitados, deben ser capaz de manejarlos.

Dudas

Pueden dejar sus preguntas sobre instalación o enunciado en las [issues](#) del repositorio del curso. No se van a responder dudas por correo.

Integridad Académica

Cualquier situación de falta a la integridad académica detectada en el contexto del curso (por ejemplo, durante alguna evaluación) tendrá como sanción un 1,1 final en el curso. Esto sin perjuicio de sanciones posteriores que estén de acuerdo a la Política de Integridad Académica de la Escuela de Ingeniería y de la Universidad, que sean aplicables al caso. Rige para este curso tanto la política de integridad académica del Departamento de Ciencia de la Computación como el Código de Honor de la Escuela de Ingeniería.

Debido a la naturaleza de la disciplina en la que se enmarca el curso, está permitido el uso de código escrito por una tercera parte, pero solo bajo ciertas condiciones. Primero que todo, el uso de código ajeno siempre debe estar visible y correctamente referenciado, indicando la fuente de donde se obtuvo. Por otro lado, se permite el uso de código encontrado en internet u otra fuente de información similar, siempre y cuando su autor sea externo al curso, o en su defecto, sea parte del equipo docente del curso. Es decir, se puede hacer referencia a código ajeno al curso y código perteneciente al curso pero solo aquel escrito por el equipo docente, como material o ayudantías. Luego, compartir o usar código de una evaluación actual o pasada se considera una falta a la ética.