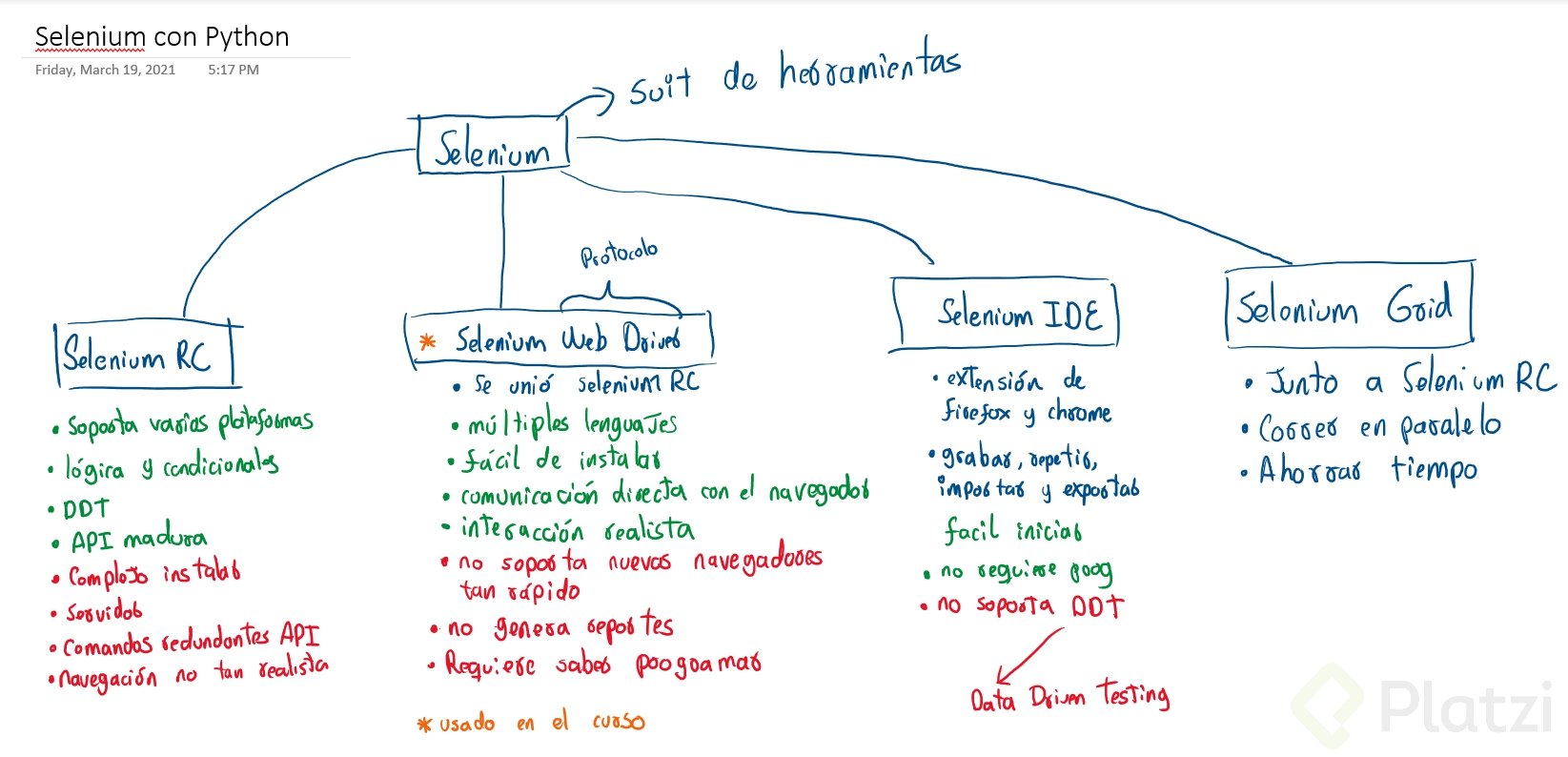
**Resumen**

¿Qué es Selenium? Es una **SUIT de herramientas** para la automatización de navegadores Web. El objetivo de Selenium NO fue para el **Testing** ni para el **Web Scraping** (aunque se puede usar para eso), por lo tanto, no es el más optimo para estas actividades. **Protocolo: WebDriver,** herramienta que se conecta a un API. **Selenium WebDriver** es la herramienta que utilizaremos en el curso. -Selenium NO es un Software, ES una SUIT de Softwares. \***DDT: Data Drive Testing:** Ingresar datos para que realice varias pruebas (sin intervención humana).



<https://selenium-python.readthedocs.io/locating-elements.html>

<https://techbeamers.com/websites-to-practice-selenium-webdriver-online/>

https://techbeamers.com/important-selenium-webdriver-commands/

https://selenium-python.readthedocs.io/waits.html

**Clases**

Como viste en clases anteriores, un sitio web se construye por código HTML en forma de árbol, conteniendo distintos elementos con los que podemos interactuar según estén presentes o no en nuestra interfaz gráfica.

Selenium WebDriver nos brinda la posibilidad de poder referirnos a estos elementos y ejecutar métodos específicos para realizar las mismas acciones que un humano haría sobre los mismos, gracias a las clases WebDriver y WebElement.

**Clase WebDriver**

Cuenta con una serie de propiedades y métodos para interactuar directamente con la ventana del navegador y sus elementos relacionados, como son pop-ups o alerts. Por ahora nos centraremos a las más utilizadas.

Propiedades de la clase WebDriver

Estas son las más comunes para acceder al navegador.

| Propiedad/Atributo | Descripción | Ejemplo |
| --- | --- | --- |
| current\_url | Obtiene la URL del sitio en la que se encuentra el navegador | driver.get\_url |
| current\_window\_handle | Obtiene la referencia que identifica a la ventana activa en ese momento | driver.current\_window\_handle |
| name | Obtiene el nombre del navegador subyacente para la instancia activa | [driver.name](http://driver.name/) |
| orientation | Obtiene la orientación actual del dispositivo móvil | driver.orientation |
| page\_source | Obtiene el código fuente de disponible del sitio web | driver.page\_source |
| title | Obtiene el valor de la etiqueta <title> del sitio web | driver.title |

**Clase WebElement**

Esta clase nos permite interactuar específicamente con elementos de los sitios web como textbox, text area, button, radio button, checkbox, etc.

Propiedades más comunes de la clase WebElement

| Propiedad/Atributo | Descripción | Ejemplo |
| --- | --- | --- |
| size | Obtiene el tamaño del elemento | login.size |
| tag\_name | Obtiene el nombre de la etiqueta HTML del elemento | login.tag\_name |
| text | Obtiene el texto del elemento | login.text |

Métodos más comunes de la clase WebElement

| Método/Atributo | Descripción | Ejemplo |
| --- | --- | --- |
| clear() | Limpia el contenido de un textarea | first\_name.clear() |
| click() | Hace clic en el elemento | send\_button.click() |
| get\_attribute(name) | Obtiene el valor del atributo de un elemento | submit\_button.get\_attribute(‘value’) last\_name.get\_attribute(max\_length) |
| is\_displayed() | Verifica si el elemento está a la vista al usuario | banner.is\_displayed() |
| is\_enabled() | Verifica si el elemento está habilitado | radio\_button.is\_enabled() |
| is\_selected() | Verifica si el elemento está seleccionado, para el caso de checkbox o radio button | checkbox.is\_selected() |
| send\_keys(value) | Simula escribir o presionar teclas en un elemento | email\_field.send\_keys(‘team@platzi.com’) |
| submit() | Envía un formulario o confirmación en un text area | search\_field.submit() |
| value\_of\_css\_property(property\_name) | Obtiene el valor de una propiedad CSS del elemento | header.value\_of\_css\_property(‘background-color’) |

**Condicionales esperadas**

| Expected Condition | Descripción | Ejemplo |
| --- | --- | --- |
| element\_to\_be\_clickable(locator) | Espera a que el elemento sea visible y habilitado para hacer clic en el mismo | WebDriverWait(self.driver,10).until(expected\_conditions.element\_to\_be\_clickable(([By.NAME](http://by.name/),“accept\_button”))) |
| element\_to\_be\_selected(element) | Espera a que un elemento sea seleccionado | subscription = self.driver.find\_element\_by\_name(“terms”). WebDriverWait(self.driver, 10).until(expected\_conditions.element\_to\_be\_selected(terms))) |
| invisibility\_of\_element\_located(locator) | Espera a que un elemento no sea visible o no se encuentre presente en el DOM | WebDriverWait(self.driver,10).until(expected\_conditions.invisibility\_of\_element\_located(([By.ID](http://by.id/),“loading\_banner”))) |
| presence\_of\_all\_elements\_located(locator) | Espera a que por lo menos uno de los elementos que se indican coincida con los presentes en el sitio | WebDriverWait(self.driver,10).until(expected\_conditions.presence\_of\_all\_elements\_located((By.CLASS\_NAME,“input-text”))) |
| presence\_of\_element\_located(locator) | Espera a que un elemento sea visible se encuentre presente en el DOM | WebDriverWait(self.driver,10).until(expected\_conditions.presence\_of\_element\_located(([By.ID](http://by.id/),“search-bar”))) |
| text\_to\_be\_present\_in\_element(locator,text\_) | Espera a que un elemento con el texto indicado se encuentre presente | WebDriverWait(self.driver,10).until(expected\_conditions.text\_to\_be\_present\_in\_element(([By.ID](http://by.id/),“seleorder”),“high”)) |
| title\_contains(title) | Espera a que la página contenga en el título exactamente como es indicado | WebDriverWait(self.driver, 10).until(expected\_conditions.title\_contains(“Welcome”)) |
| title\_is(title) | Espera a que la página tenga un título idéntico a como es indicado | WebDriverWait(self.driver, 10).until(expected\_conditions.title\_is(“Welcome to Platzi”)) |
| visibility\_of(element) | Espera a que el elemento indicado esté en el DOM, sea visible, su alto y ancho sean mayores a cero | first\_name = self.driver.find\_element\_by\_id(“firstname”) WebDriverWait(self.driver, 10).until(expected\_conditions.visibility\_of(first\_name)) |
| visibility\_of\_element\_located(locator) | Espera a que el elemento indicado por su selector esté en el DOM, sea visible y que su alto y ancho sean mayores a cero | WebDriverWait(self.driver,10).until(expected\_conditions.visibility\_of\_element\_located(([By.ID](http://by.id/),“firstname”))) |