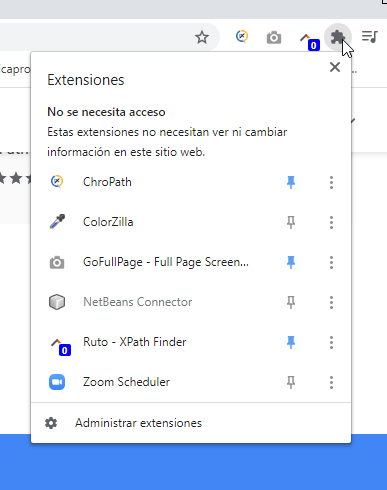
**CLASE 2**

-Chequear las configuraciones básicas vistas en clase 1.

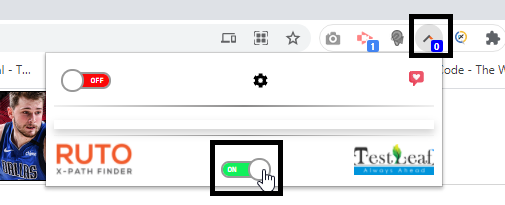
-Para ayudar en identificar los elementos HTML se pueden descargar las siguientes extensiones de Google Chrome: RutoXPath y Chropath.



-**RutoXPath** (Chrome extensión) => Ayuda a identificar selectores de HTML como id, name, XPath. Descargar y ALT+R para activar.

a. Ir a la pagina web deseada, ejemplo <https://www.infobae.com/>

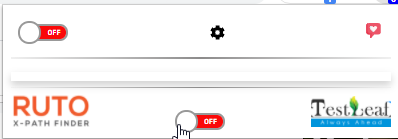
b. Activar en la extensión de RutoXPATH:



c. Seleccionar el elemento deseado (lo vemos remarcado en rojo) :



d. Finalmente apagamos el xpath Finder:

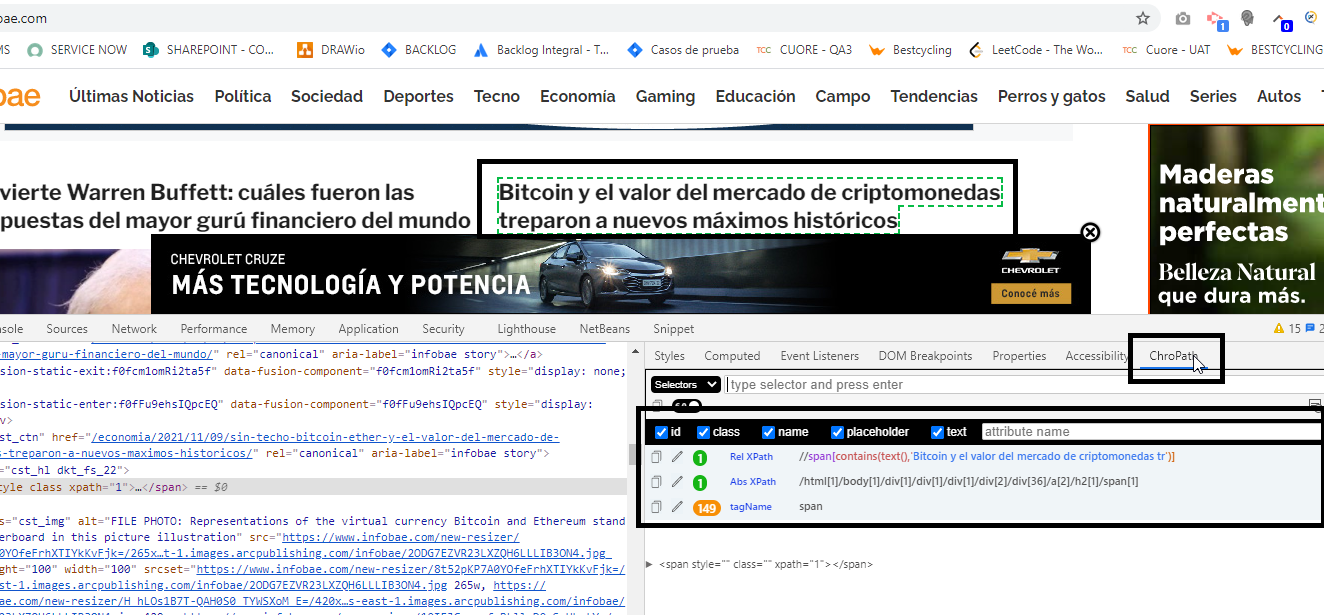


Y vemos como se genera automáticamente el siguiente archivo con **FindBy** y diferentes locators:



-**Chropath**: Ayuda a generar y validar selectores como xpath con iframe, svg, también permite generar testcases a nivel browser.

1. Vamos a una pagina determinada, ejemplo <https://www.infobae.com/>.
2. Hacemos **inspeccionar** sobre el elemento deseado:



Vemos que se creo una solapa **CroPath** con diferentes locators por Xpath relativo y absoluto.

**CLASE 2**

-Los siguientes paquetes son necesarios para operar con Selenium y JUNIT.

import org.junit.\*;

import org.openqa.selenium.\*;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

Impoer org.openqa.selenium.support.ui.Select

*import java.util.concurrent.TimeUnit; > Para poder operar con las esperas*

**Manual del Alumno:**

-**Teoria Hasta pagina 5, configuramos un ejemplo en eclipse:**

**IMPLICIT WAITS**

**Codigo Eclipse CLASE\_02\_EJEMPLO\_1 (Package a\_ImplicitWaits):**

En el siguiente ejemplo declaramos un implicit wait de 10 segundos. Si en ese lapso de tiempo algun elemento no es localizado entonces se arrojara una excepción “No such element Exception”.

Sintaxis:

driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);

1. **Run as -> Junit Test**

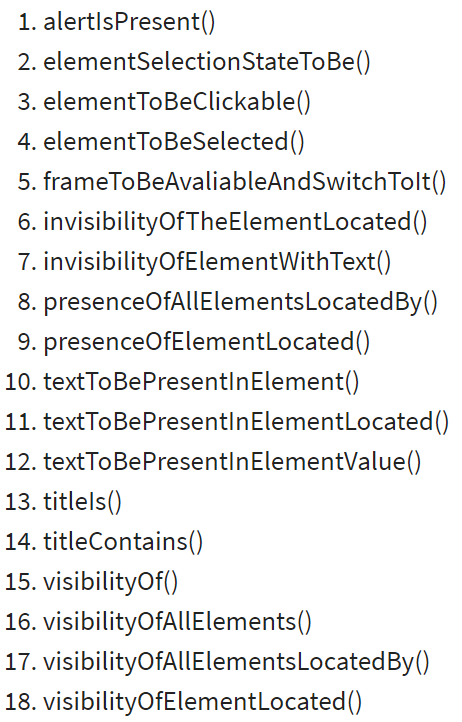
**EXPLICIT WAITS**

*Objetivo: Ir viendo las diferentes formas de encontrar un elemento e interactuar con los mismos desde Selenium, teniendo como base el manual del alumno.*

**1. Codigo Eclipse CLASE\_02\_EJEMPLO\_1 (Package b\_ExplicitWaits).**

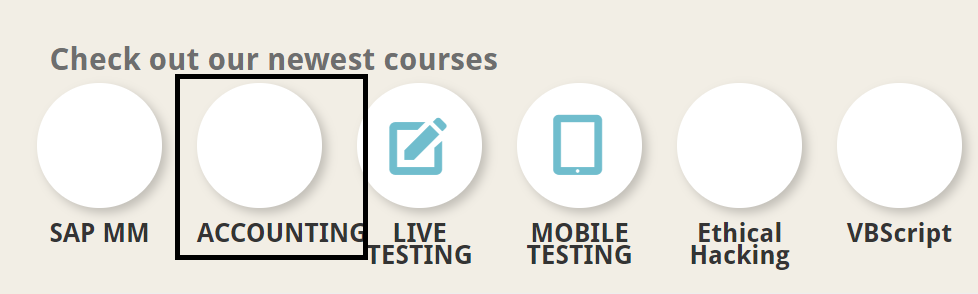
a. Vemos un primer ejemplo donde no aparece el elemento buscado. Entonces se ve como espera 10 segundos (espera explicita) y arroja la excepción.

Las Expected Conditions que pueden ser utilizadas son:



Ver el código ExplicitWait dentro del proyecto previo y vemos como no encuentra el elemento /html/body/div[1]/section/div[2]/div/div[1]/div/div[1]/div/div/div/div[2…

1. Clickeamos el Icono de Accounting:

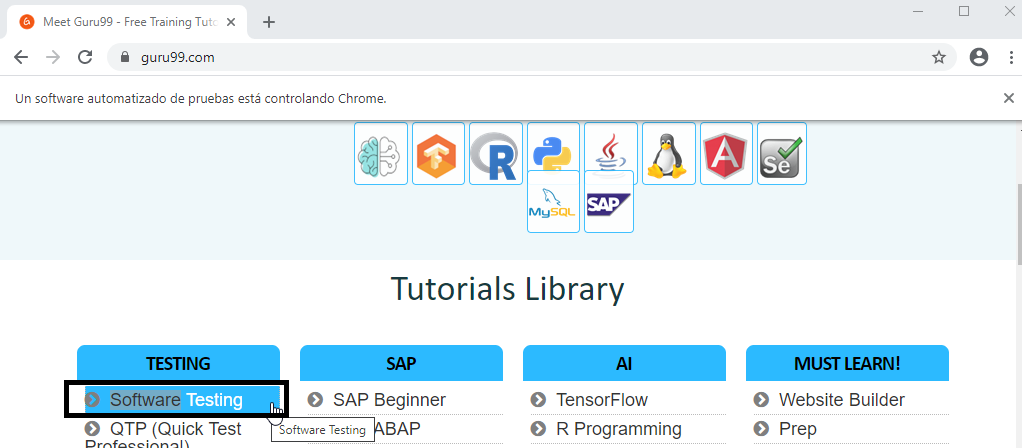


**2.** Buscamos un elemento por **name**. Ver pagina 6 del manual.

**3.** Pagina 7 a 14, Teoria VS Practica.

**-Localizando XPATH:** Relativo, pagina 7 (empieza con 2 barras).

4. XPATH Absoluto, pagina 8 (empieza con 1 barra).

****

**5.** Codigo: **Clase\_02\_EJEMPLO\_1\_c\_or\_and**

**6. Codigo: Clase\_02\_EJEMPLO\_1\_d\_order\_Lambda\_listas**

**Codigo:**

****

**Orden en los Test de Junit:** Por default Junit utiliza el MethodSorters.DEFAULT lo que significa que efectua un orden impredecible. Para establecer un orden podemos agregar la siguiente annotation para la clase

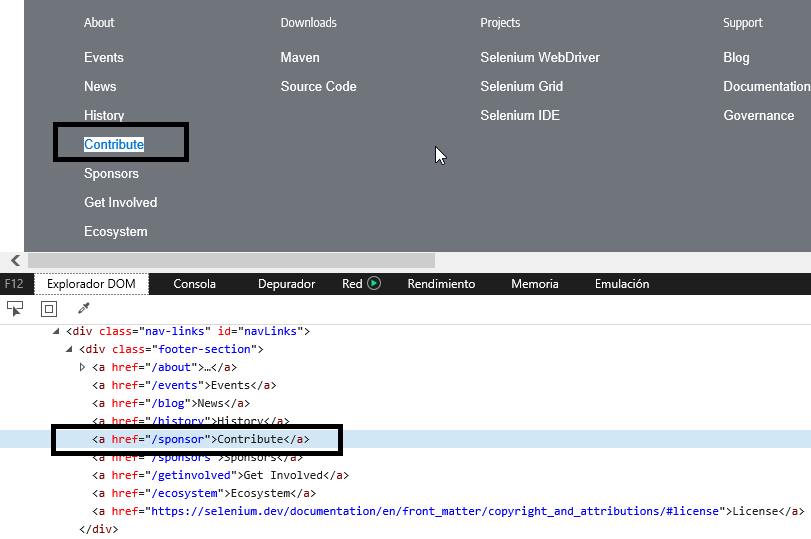
@FixMethodOrder(MethodSorters.NAME\_ASCENDING)

en cuyo caso se ordenara en forma alfabetica por el nombre de los métodos.

1. Pagina 13, 14. Busqueda por texto – text(). Buscamos el siguiente texto “Learn More”:



1. **-Pagina 15.** cssSelector(). Buscamos Contribute u otro selector.



1. **-Paginas 16 y 17.** Codificamos ejemplos en código buscarTexto(). Buscamos la posición del header en el html.

**-Pagina 18.** A modo informativo se muestra las extensiones útiles para adherir a navegadores.

1. **-Pagina 19** Listas Desplegables o drop-down box.
2. **-Pagina 20** Listas de Selección Multiple

**LABORATORIO:**





Desafio: Completar el ejercicio de modo tal que solucione lo siguiente,



Luego para los que lo hayan resuelto tendrán la posibilidad de mostrar la pantalla.

**BORRAR CACHE**

****