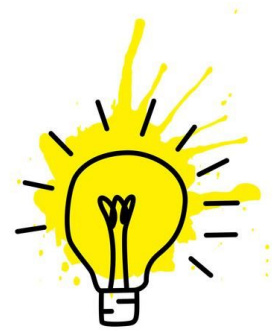
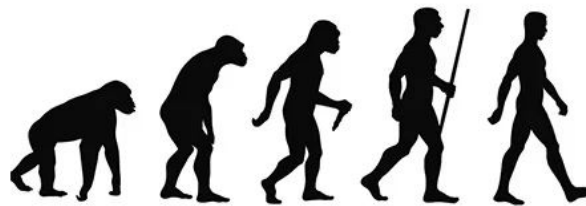


Refactoring



Qué es el code refactor?



Refactoring

Improving the Design of Existing Code

Factories



Qué es una fábrica?



Cómo se implementa una fábrica?

Se crearán muchas instancias de una clase

```
@Factory
public Object[] facebookFactoryTest(){
    return new Object[]{
        new FacebookTest(),
        new FacebookTest(),
        new FacebookTest(),
    };
}
```



Cómo se implementa una fábrica?

Se crearán muchas instancias de una clase

Esto permite que se puedan realizar ejecuciones con diferentes parámetros

```
@Factory
public Object[] factoryMethod(){
    return new Object[] {
        new DocusignFactoryTest( param: 3),
        new DocusignFactoryTest( param: 6),
        new DocusignFactoryTest( param: 9)
    };
};
```



Cómo se implementa una fábrica?

La clase de test, tendrá un constructor que reciba ese parámetro, y lo setee a una variable de clase

```
private WebDriver driver;  
private int parametro = 0;  
  
public DocusignFactoryTest(int param){  
    this.parametro = param;  
}
```



Waits

Clase 13



Qué pasa cuando nuestros tests se ejecutan más rápido que el sitio?



NoSuchElementException

```
org.openqa.selenium.NoSuchElementException: no such element: Unable to locate element  
(Session info: chrome=86.0.4240.183)
```

Tipos de esperas

1. Thread.Sleeps
2. Implicit waits
3. Explicit waits

Thread.sleep(x)



Thread.sleep

- Espera provista por Java
- Causa que la ejecución del test se suspenda por un periodo de tiempo determinado en mili-segundos.
- Thread.sleep(X) puede lanzar una excepción



Thread.sleep(milisegundos);

.... **throws** InterruptedException

Thread.sleep(**5000**);



Test con Thread.sleep(5000)

```
@Test
public void threadTest() throws InterruptedException {
    WebDriver driver = getDriver(url: "salesforce");
    Thread.sleep(millis: 5000);
    driver.findElement(By.partialLinkText("Forgot")).click();
}
```



Problema de Thread.sleep

Supongamos que tenemos un sleep de 5 segundos...

1. Qué pasa si el elemento aparece al segundo 1?
2. Qué pasa si tenemos una espera de 5 segundos por cada test y nuestra clase tiene 50 tests?



Problema de Thread.sleep

Supongamos que tenemos un sleep de 5 segundos...

1. Qué pasa si el elemento aparece al segundo 1?
→ Se demora la ejecución 4 segundos más
2. Qué pasa si tenemos una espera de 5 segundos por cada test y nuestra clase tiene 50 tests?



Problema de Thread.sleep

Supongamos que tenemos un sleep de 5 segundos...

1. Qué pasa si el elemento aparece al segundo 1?
→ Se demora la ejecución 4 segundos más
2. Qué pasa si tenemos una espera de 5 segundos por cada test y nuestra clase tiene 50 tests?
→ El test demorará al menos unos 250 segundos en ejecutarse todos los test



Waits implícitos



Waits implícitos

El wait implícito le dice al driver que espere una cierta cantidad de tiempo antes de tirar la excepción de “No Such Element Exception”

```
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS) ;
```

```
driver.manage().timeouts().implicitlyWait(1000, TimeUnit.MILLISECONDS) ;
```

Waits implícitos

La espera aplica para todos los elementos que se traten de localizar

Usualmente se coloca en el setup



Waits explícitos



Explicit Wait

Se indica al driver que espere a que se cumplan ciertas condiciones (**Expected Conditions**), y si las condiciones no aparecen por un periodo de tiempo, se lanza la excepción de “**ElementNotVisibleException**”

Una vez que el elemento aparece, el flujo del test continua normalmente.

El wait explícito es un wait “inteligente”, ya que espera a que un elemento sea cargado dinámicamente.



Explicit Wait

Se debe crear una instancia del objeto WebDriverWait

```
WebDriverWait wait = new WebDriverWait(driver, 10);
```

```
wait.until(ExpectedConditions.elementToBeClickable(By.id("username"));
```

```
wait.until(ExpectedConditions.elementToBeSelected(By.id("gender"));
```



Métodos de ExpectedConditions

- * `elementToBeClickable()`
- * `elementToBeSelected()`
- * `textToBePresentInElement()`
- * *`textToBePresentInElementLocated()`*

Métodos de ExpectedConditions

- * *`titleIs()`*
- * *`titleContains()`*
- * *`visibilityOf()`*
- * *`visibilityOfElementLocated()`*

Ventajas de Explicit waits

La espera es hasta que aparezca el elemento en el sitio!!

Una vez que el elemento aparece, el test sigue su flujo normal.

Si el elemento ya se encuentra presente, entonces la espera es 0 segundos



Implicit vs. Explicit waits



Thread.sleep(X)	<p>Detiene la ejecución de todo el test</p> <p>Si el driver localiza el elemento, la espera continua</p>
Implicit wait	<p>Brinda al test, un margen de espera fijo para localizar a los elementos</p> <p>Cuando encuentra el elemento, continúa el flujo</p>
Explicit wait	<p>El test espera a que un elemento sea localizado en el sitio por el driver, durante un determinado tiempo</p> <p>Cuando encuentra el elemento, continúa el flujo</p>