Curso de Selenium

Clase 14

Cronograma - Módulo 2

Clase 13:

- Refactoring, Factories y implicit waits y explicit waits

Clase 14:

- Data Providers y generación de datos dinámicos

Clase 15:

- Iframes y Page Object Pattern

Clase 16:

- Page Object Pattern y Herencia

@DataProviders

@DataProviders

Los data providers proveen de datos a los tests, de forma tal, de que un mismo test puede ser ejecutado múltiples veces, recibiendo diferentes valores.

La forma de declararlo es a través de la notación @DataProvider

```
@DataProvider(name="paises")
public Object[][] datosDePaises(){
    return new Object[][] {
            {"Santiago", "Chile"},
            {"Buenos Aires", "Argentina"},
            {"Montevideo", "Uruguay"}
    };
}
```

```java
public class DataProviderEjemplo1 {

    @DataProvider (name="personas")
    public Object[][] datosDePersonas(){
        return new Object[][] {
            {"Juan", 25},
            {"Maria", 32}

        };
    }

    @Test (dataProvider = "personas")
    public void mostrarInformacionTest(String unNombre, int unaEdad){
        System.out.println(unNombre + " tiene una edad de " + unaEdad);
    }

}
```

Clase DataProvider

Se puede crear una clase que contenga todos los datos providers

Por ejemplo, una clase llamada DataProviderGenerator donde se establezcan diferentes data providers a ser utilizados.

```java
@DataProvider(name="paises")
public Object[][] datosDePaises(){
    return new Object[][] {
            {"Santiago", "Chile"},
            {"Buenos Aires", "Argentina"},
            {"Montevideo", "Uruguay"}
    };
}
```

```java
@Test(dataProvider = "paises", dataProviderClass = DataProviderGenerator.class)
public void mostrarPaises(String unaCapital, String unPais){
    System.out.println("La capital de " + unPais + " es " + unaCapital);
}
```

@DataProviders

Los data providers no crean una nueva instancia por cada conjunto de datos, sino que ejecutan en una única instancia


seleniumacademy

Fakers

Motivación

Generar datos de forma aleatoria.

Cada vez que ejecuto los tests, quiero tener diferentes datos para enviar

Libreria Javafakers

```
<!-- https://mvnrepository.com/artifact/com.github.javafaker/javafaker -->
<dependency>
    <groupId>com.github.javafaker</groupId>
    <artifactId>javafaker</artifactId>
    <version>1.0.2</version>
</dependency>
```

Utilizando Fakers

```
Faker faker = new Faker();

String name = faker.name().fullName(); // Miss Samanta Schmidt
String firstName = faker.name().firstName(); // Emory
String lastName = faker.name().lastName(); // Barton

String streetAddress = faker.address().streetAddress(); // 60018 Sawayn Brooks Suite 449
```

http://dius.github.io/java-faker/apidocs/index.html

https://github.com/DiUS/java-faker

Crear una clase específica para generar Datos

```
public static Faker faker = new Faker();

public static String getFirstName() {
    //Generating the first name
    String firstName = faker.name().firstName();
    return firstName;
}
```

Crear una clase específica para generar Datos

```java
public static String getLastName() {
    //Generating last name
    String lastName = faker.name().lastName();
    return lastName;
}
```

Crear una clase específica para generar Datos

```
public static String getPhone() {
    //Generating password
    Random random = new Random();

    return String.valueOf(random.nextInt( bound: 1000000) +1000000000);
}
```

Crear una clase específica para generar Datos

```
public static String getEmail() {
    //Generating email Id
    String emailId = faker.internet().emailAddress();
    return emailId;
}
```

Crear una clase específica para generar Datos

```
public static String getEmail() {
    //Generating email Id
    String emailId = faker.internet().emailAddress();
    return emailId;
}
```

Groups

```
<suite name="Test suites" >
    <test name="All the Tests">
        <groups>
            <run>
                <include name = "successTests" />
                <include name = "failTests" />
            </run>
        </groups>
        <classes>
            <class name = "Curso_1_Básico.clase14_
        </classes>
    </test>
</suite>
```

Groups

```
@Test(groups = { "successTests" })
public void successTest1(){}

@Test(groups = { "successTests" })
public void successTest2(){}

@Test(groups = { "successTests" })
public void successTest3(){}

@Test(groups = { "failTests" })
public void failTest1(){}

@Test(groups = { "failTests" })
public void failTest2(){ }
```