



Práctico 14:

Data Providers y Fakers

Data Providers

Ejercicio

Crear una clase llamada **testngSalesforce** que importe `import org.testng.annotations.DataProvider;`

Crear un método llamado **crearPersonas** que retorne una matriz de tipo `object` con dos nombres y dos edades

Ejercicios

Crear una clase llamada **DataGenerator** que importe `import org.testng.annotations.DataProvider;`

Crear un método llamado **DataProviderEjemplo** que muestre en pantalla las capitales que se encuentran en la clase **DataGenerator** con las capitales y su respectivo país

Crear un método llamado **mostrarDatosPersonas** que reciba el nombre, apellido y edad de las personas.

Ejercicio:

Crear una clase llamada **PeopleSimpleTest** que contenga un `@DataProvider` llamado "personas" que retorne un arreglo bidireccional de objetos con datos el nombre y la edad de 2 personas.

Crear un método llamado **nombreTests** que reciba el nombre y la edad de las personas y las imprima en pantalla.

Ejercicio:

Crear una clase llamada **DataProviderPeople** y agregar un método con notación `@DataProvider` llamada **people**, que retorne una array bidireccional de objetos y retorne el nombre, apellido y edad de dos personas.

Crear una clase llamada **PeopleTest** que reciba las personas de la clase **DataProviderPeople** y las imprima en pantalla.



Fakers

Ejercicio :

Investigar la información que esta en <https://github.com/DiUS/java-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();
```

Ejercicio :

Agregar al pom.xml la dependencia a java-fakers

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

Ejercicio :

- Crear un método de test llamado testingErrorMessage
- Ingresar a la página <https://login.salesforce.com/?locale=eu>
- Hacer click en botón Try To Free
- Dar click en el botón de **"Start my free trial"**
- Mostrar todos los mensajes de error en pantalla
- Validar que se muestran los mensajes:
 - Enter your first name
 - Enter your last name
 - Enter your title
 - Enter a valid email address
 - Enter a valid phone number
 - Enter your company name
 - Select the number of employees
 - Select company language

Ejercicio :

Consultas a seleniumcurso@gmail.com

Práctico 14



Completar el método `testingErrorMessage`: utilizando `Fakers`, completar el campo de la persona. Luego, hacer click en el botón de “Start free trial”. Validar que el mensaje de error para el email no esté presente.

Ejercicio :

Crear un método de test llamado **`completeSalesforceFormTest`**

Ingresar a la página <https://login.salesforce.com/?locale=eu>.

Hacer click en botón Try To Free

Completar la información del registro sin ser el título

First Name: John

Last Name: Smith

Title: Tester

Email: test@test.com

Phone: 1234123412

Dar click en el botón de “**Start my free trial**”

Validar que la información de error mostrada al finalizar, sea la correcta

Validar que la información ingresada sea la que se muestra en pantalla

Ejercicio :

Crear una clase llamada `DataFactory`. Luego crear los siguientes métodos:

- `getFirstName`
- `getLastName`
- `Email`
- `getPhone`
- `getAddress`

Ejercicio :

Crear un método de test llamado **`fillFormWithFakersTest`**

Ingresar a la página <https://login.salesforce.com/?locale=eu>.

Hacer click en botón “Try To Free”

Completar toda la información utilizando el `dataFaker`. Para ello, debe realizar los siguientes pasos:

Crear una clase para la `DataFaker`

Crear una variable estática de tipo `Faker` dentro de `DataFactory`

Crear los métodos `getFirstName`, `getLastName`, `getEmail`, `getPhone`. Estos métodos deben retornar un string con un valor random dado por la clase `Faker`.

El teléfono debe ser un número randómico de 10 dígitos

Mostrar en pantalla los datos enviados al sitio



Validar que la información mostrada al finalizar, sea la correcta

Validar que la información ingresada sea la que se muestra en pantalla