

[Solución]

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

```
public class dataProviderTests {  
  
    @DataProvider(name = "personas")  
    public Object[][] crearPersonas() {  
        return new Object[][] {  
            { "Juan", new Integer( value: 36) },  
            { "Maria", new Integer( value: 37)},  
        };  
    }  
  
    @Test(dataProvider = "personas")  
    public void nombresTests(String nombre, Integer edad) {  
  
        System.out.println(nombre + " " + edad);  
    }  
}
```

Ejercicio

Crear una clase llamada **DataGenerator**

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



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

@Test(dataProvider = "países", dataProviderClass = DataGenerator.class)
public void testOfDataProviders(String capital, String país) {
    System.out.println("La capital de " + país + " es " + capital);
}
```

- a) Crear un método llamado `mostrarDatosPersonas` que reciben el nombre, apellido y edad de las personas.

```
import org.testng.annotations.DataProvider;

public class DataProvidersClass {
    @DataProvider(name = "personas")
    public static Object[][] dataProviderMethod2() {
        return new Object[][]{{"Ana", "Rodriguez", "22"},
                               {"Juan", "Perez", "19"}};
    }
}

@Test(dataProvider = "personas", dataProviderClass = DataProvidersClass.class)
public void mostrarDatosPersonas(String nombre, String apellido, String edad)
{
    System.out.println("Mi nombre es" + nombre + " " + apellido + " y tengo " + edad);
}
```

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.

```
public class PeopleSimpleTest {

    @DataProvider(name = "personas")
    public Object[][] crearPersonas() {
        return new Object[][] {
            { "Juan", 36 },
            { "Maria", 37 }
        };
    }

    @Test(dataProvider = "personas")
    public void nombresTests(String nombre, Integer edad) {
        System.out.println(nombre + " tiene " + edad);
    }
}
```

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.

```
import org.testng.annotations.DataProvider;

public class DataProvidersPeople {
    @DataProvider(name = "people")
    public static Object[][] peopleDataProvider() {
        return new Object[][]{{"Ana", "Rodriguez", "22"},
            {"Juan", "Perez", "19"}};
    }
}
```

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

```
@Test(dataProvider = "people", dataProviderClass = DataProvidersPeople.class)
public void mostrarDatosPersonas(String nombre, String apellido, String edad)
{
    System.out.println("Mi nombre es" + nombre + " " + apellido + " y tengo " + edad);
}
```



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

Ingresa 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

```
@Test
public void testingErrorMessages() {
    driver.get("https://login.salesforce.com/?locale=eu");
    WebElement TryToFreeButton = driver.findElement(By.id("signup_link"));
    TryToFreeButton.click();

    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    driver.findElement(By.name("Start my free trial")).click();

    List<WebElement> errorMessagesList = driver.findElements(By.className("error-msg"));

    Boolean titleError = false;
    Boolean phoneError = false;
    Boolean emailError = false;
    Boolean companyError = false;
    for (WebElement webElement : errorMessagesList) {
        System.out.println(webElement.getText());
        if (webElement.getText().equals("Enter your title")){
            titleError = true;
        } else if (webElement.getText().equals("Enter a valid phone number")) {
            phoneError = true;
        } else if (webElement.getText().equals("Enter a valid email address")) {
            emailError = true;
        } else if (webElement.getText().equals("Enter your company name")) {
            companyError = true;
        }
    }

    Assert.assertTrue(titleError);
    Assert.assertTrue(phoneError);
    Assert.assertTrue(emailError);
    Assert.assertTrue(companyError);
}
```

Ejercicio :

Completar el método testingErrorMessages: 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.



```
String email = DataFactory.getEmail();
emailError = false;

driver.findElement(By.name("UserEmail")).sendKeys(email);
driver.findElement(By.name("Start my free trial")).click();
errorMessagesList = driver.findElements(By.className("error-msg"));
for (WebElement webElement : errorMessagesList) {
    if (webElement.getText().equals("Enter a valid email address")) {
        emailError = true;
    }
}
Assert.assertFalse(emailError);
```

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

```
@Test
public void completeSalesforceFormTest() {
    driver.get("https://login.salesforce.com/?locale=eu");
    WebElement TryToFreeButton = driver.findElement(By.id("signup_link"));
    TryToFreeButton.click();
    Select selectBox = new Select(driver.findElement(By.name("CompanyEmployees")));
    selectBox.selectByValue("50");
    driver.findElement(By.name("UserFirstName")).sendKeys("name");
    driver.findElement(By.name("UserLastName")).sendKeys("name");
    driver.findElement(By.name("UserTitle")).sendKeys("name");
    driver.findElement(By.name("UserEmail")).sendKeys("name@test.com");
    driver.findElement(By.name("UserPhone")).sendKeys("123123");
    driver.findElement(By.name("CompanyName")).sendKeys("ORT");
    driver.findElement(By.name("Start my free trial")).click();

    WebDriverWait wait=new WebDriverWait(driver, timeOutInSeconds: 20);

    WebElement errorMessage;
    errorMessage = wait.until(ExpectedConditions.visibilityOf( driver.findElement(By.id("SubscriptionAgreement-akLN-errMsg"))));
    System.out.println(errorMessage.getText());
}
```




Ejercicio :

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

- getFirstName
- getLastName
- getEmail
- getPhone
- getAddress

```
public class DataFactory {  
  
    public static Faker faker = new Faker();  
  
    public static String getFirstName() {  
        String firstName = faker.name().firstName();  
        return firstName;  
    }  
  
    public static String getLastName() {  
        String lastName = faker.name().lastName();  
        return lastName;  
    }  
  
    public static String getEmail() {  
        String emailId = faker.internet().emailAddress();  
        return emailId;  
    }  
  
    public static String getPhone() {  
        Random random = new Random();  
        return String.valueOf(random.nextInt( bound: 1000000) +1000000000);  
    }  
}
```

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



```
String firstName = DataFactory.getFirstName();
String lastName = DataFactory.getLastName();
String phone = DataFactory.getPhone();
String email = DataFactory.getEmail();
@Test
public void fillFormWithFakersTest() {
    driver.get("https://login.salesforce.com/?locale=eu");
    WebElement TryToFreeButton = driver.findElement(By.id("signup_link"));
    TryToFreeButton.click();

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

    Select selectBox = new Select(driver.findElement(By.name("CompanyEmployees")));
    selectBox.selectByValue("50");
    driver.findElement(By.name("UserFirstName")).sendKeys(firstName);
    driver.findElement(By.name("UserLastName")).sendKeys(lastName);
    driver.findElement(By.name("UserEmail")).sendKeys(email);
    driver.findElement(By.name("UserPhone")).sendKeys(phone);
    driver.findElement(By.name("CompanyName")).sendKeys("...charSequences: "ORT");
    driver.findElement(By.name("Start my free trial")).click();

    WebDriverWait wait=new WebDriverWait(driver, 20);

    WebElement errorMessage;
    errorMessage = wait.until(ExpectedConditions.visibilityOf(driver.findElement(By.id("SubscriptionAgreement-akLN-errMsg"))));
    System.out.println(errorMessage.getText());
}
```