

OBJETIVOS

Interiorizar el patrón de automatización Screenplay de manera teórica y práctica

Comprender la responsabilidad de cada capa definida en la arquitectura

Aprender a usar el actor, las tareas, interacciones y preguntas

Asimilar el desarrollo de software como un proceso evolutivo

INTRODUCCIÓN

TIEMPOS DE CAMBIO

- ¿Por qué estamos cambiando?
- ¿cuáles son los pilares de la programación orientada a objetos?
- ¿qué son los principios solid?
- ¿en qué consiste page object model?
- ¿en dónde falla page object model?
- ¿qué podemos hacer?

URL: <https://youtu.be/lzp--luYthc>

INTRODUCCIÓN A SCREENPLAY

- ¿En qué consiste screenplay?
- ¿cuál es la arquitectura?
- ¿cuál es la responsabilidad de cada capa?
- ¿qué es una tarea?
- ¿cuál es el stack de tecnologías a utilizar?

URL: <https://youtu.be/xwH97NiGue4>

SCREENPLAY EN LA PRÁCTICA

1. CREACIÓN DE UN PROYECTO

Tutorial que explica el procedimiento para crear un proyecto gradle en el ide eclipse

URL: <https://youtu.be/v7J8nYasUuw>

PILARES POO

La programación orientada a objetos es un paradigma de programación sustentado sobre 4 pilares fundamentales:

Abstracción
Herencia
Polimorfismo
Encapsulamiento

PRINCIPIOS SOLID

Son un conjunto de principios que facilitan la implementación de una correcta programación orientada a objetos

Single Responsibility
Open/Closed
Liskov Substitution
Interface Segregation
Dependency Inversion

2. CONFIGURACIÓN DEL BUILD.GRADLE

Uso del archivo build.gradle para la configuración del proyecto, plugins, propiedades y gestión de dependencias

URL: <https://youtu.be/0gUPfVi0Wro>

3. ARQUETIPO BASE

Creación de la estructura del proyecto acorde a la arquitectura definida

URL: <https://youtu.be/9Iis8shAnVg>

4. ¿QUÉ VAMOS A HACER?

Explicación del propósito del proyecto práctico para comenzar su desarrollo

URL: https://youtu.be/toj8_2BVJIO

5. FEATURES

- Creación de los archivos .feature en notación Gherkin haciendo uso de Cucumber
- Escritura de buenos escenarios

URL: <https://youtu.be/d2ed64OVq8E>

6. RUNNERS & STEP DEFINITIONS

- ¿Qué es un runner?
- ¿Qué son los step definitions?
- ¿Qué son los snippets?
- Uso de @RunWith() y @CucumberOptions()
- ¿Cómo evitar cometer errores al ingresar las rutas a los feature y a los step definitions?

URL: https://youtu.be/EHEnR8Cvd_k

ACTOR

Es el ente principal de Screenplay, quien ejecuta las tareas y orquesta la ejecución de los test

El actor:

- Tiene habilidades
- Puede realizar tareas
- Puede hacer preguntas

TAREAS

Son el corazón de Screenplay. Usualmente son descritas por verbos de negocio que envuelven una serie de interacciones con un objetivo definido.

Ejemplos:

- Autenticar
- Traducir
- Buscar
- Comprar
- Diligenciar

7. DRIVERS

- ¿Dónde descargar el driver?
- ¿Cómo instanciarlo?

URL: https://youtu.be/Fd5wXwiO_bA

8. EL ACTOR EN SCREENPLAY

- ¿Qué es el actor?
- ¿Por qué es un ente tan importante en Screenplay?
- ¿Cómo se crea un actor?

URL: <https://youtu.be/DfDeQFvIao0>

9. MI PRIMERA TAREA

- ¿Qué es una tarea?
- ¿Cómo se hace una tarea?
- ¿Dónde se colocan las interacciones?
- ¿Cómo debemos instanciar las tareas?
- ¿Cómo utilizar la interacción *Open*?

URL: <https://youtu.be/Clw-P1w6nl8>

10. MI SEGUNDA TAREA

- ¿Cómo mapear objetos?
- ¿Qué es un Target?
- ¿Cuándo usar `located()` y cuándo usar `locatedBy()`?
- ¿Cómo interactuar con la página?
- ¿Cómo utilizar la interacción *Click*?

URL: <https://youtu.be/PPZlJsZLW7c>

INTERACTIONS

Permiten interactuar con los elementos de las páginas web

Algunas interacciones ya definidas por Serenity BDD son:

- Open
- Click
- Enter
- Hit

QUESTIONS

Son clases utilizadas para realizar verificaciones o Assertions.

Hacen uso clases proveídas por SerenityBDD para acceder a información de la página como el texto de un label o el valor de la visibilidad de un campo.

Algunas de estas clases son:

- Text
- Value
- Visibility

SOFTWARE COMO PROCESO EVOLUTIVO

11. REFACTOR I

- ¿Cómo podemos mejorar las tareas creadas?
- ¿Por qué es necesario realizar mejoras?
- “Si algo puede cambiar, entonces es una variable”
- ¿Cómo enviar información a la tarea?

URL: https://youtu.be/5klZgV_j7Ac

12. TAREA 3 PARTE I

- Mapeo de elementos avanzado
- ¿Cómo construir un xPath manualmente?
- ¿Qué elementos necesitamos para llevar a cabo la tarea?

URL: https://youtu.be/uI_wu9nBaMo

13. TAREA 3 PARTE II

- ¿Cómo organizar una tarea?
- Métodos sobre cargados de las interacciones
- ¿Qué método utilizar?
- Prueba y error
- Lectura de errores en la consola
- Diferencias entre la instancia del navegador que utilizamos y la instancia que abre el driver

URL: <https://youtu.be/wy5p1Uk3pVg>

USER INTERFACE

En esta capa se realiza la abstracción de las páginas web. Cada elemento es capturado como un Target.

Un Target puede encontrarse haciendo uso de 2 métodos, dependiendo del localizador utilizado.

- **locatedBy():** Se utiliza para localizar elementos por XPath o selector CSS
- **located():** Se utiliza para localizar elementos con los demás localizadores, como id, name y class