



UNIVERSIDAD MAYOR DE SAN SIMÓN
FACULTAD DE CIENCIAS Y TECNOLOGÍA
CARRERA DE INGENIERÍA DE SISTEMAS



Automatización de las pruebas de interfaz de usuario en Salesforce Módulo de Productos y Listas de Precios

Modalidad: Diplomado de Doble Titulación

Elaborado por: Carlos Eduardo Caballero Burgoa

Cochabamba - Bolivia

Índice general

1. Marco Teorico	1
1.1. Automatización de pruebas	1
1.1.1. Criterios a seguir	2
1.1.2. Beneficios	2
1.1.3. Riesgos	2
1.2. Pirámide de la automatización de pruebas	3
1.2.1. La capa de interfaz	3
1.3. End-to-End Testing	4
1.3.1. Buenas prácticas para pruebas E2E	4
1.4. Frameworks de automatización	5
Bibliografía	6

1

Marco Teorico

1.1. Automatización de pruebas

La automatización de pruebas es una técnica usada en aplicaciones para implementar todo el ciclo de vida del software en menor tiempo, y proveyendo a este proceso de eficiencia y efectividad en su etapa de evaluación.

La automatización de las pruebas es mas útil en el lanzamiento de nuevas versiones de software, para evaluar que todos los errores anteriormente corregidos no sean introducidos nuevamente, mientras que será difícil y costoso hacer las pruebas manualmente, ejecutar las pruebas automatizadas será mas efectivo en términos de costo, uso de los recursos, y aprovechamiento del tiempo.

Se conoce que en los siguientes escenarios es útil automatizar:

- Los requerimientos no cambian frecuentemente.
- Se accede al aplicativo con múltiples y variados usuarios, roles y privilegios.
- El software es estable respecto a sus pruebas manuales.
- Se cuenta con el tiempo necesario para automatización en el proyecto.
- El proyecto sigue o debe seguir estándares estrictos.
- La complejidad del proyecto es elevada.
- El proyecto requiere constantes revisiones en algunas de sus características.

1.1.1. Criterios a seguir

Existen muchas herramientas útiles para escribir rutinas de automatización, pero los pasos identificables en el proceso pueden simplificarse en los siguientes:

- Identificar áreas dentro del software para automatizar.
- Elegir la herramienta adecuada para la automatización.
- Escribir las rutinas de prueba.
- Desarrollar el conjunto de casos de prueba.
- Ejecutar los casos de prueba.
- Generar los reportes de resultados.
- Encontrar los posibles errores o aspectos negativos.

1.1.2. Beneficios

Los beneficios de implementar un proceso de automatización de pruebas son diversos, entre los cuales destacan:

- Incremento de la productividad.
- Ahorro de dinero, en el costo del proyecto.
- Aumento de la calidad del software.
- Reducción del tiempo de evaluación del software.
- Soporte para múltiples aplicativos del software.
- Aumento de la cobertura de las pruebas.
- Reducción del trabajo repetitivo.
- Mejora de la consistencia del producto.

1.1.3. Riesgos

También existen riesgos implicados en la automatización de las pruebas, tales escenarios deben ser considerados antes de proceder con la automatización, entre estos están:

- El costo de arranque de la automatización puede llegar a ser muy alto.

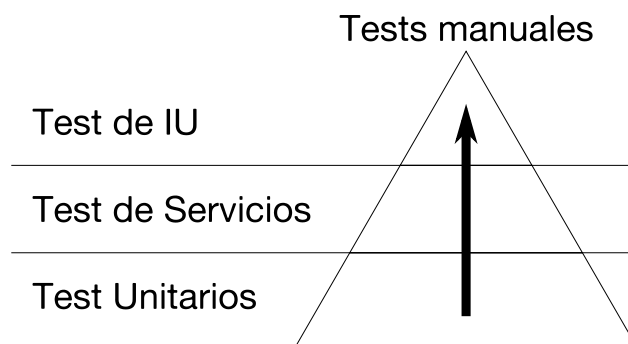


Figura 1.1: Piramide de testing.

- Debe tenerse en cuenta que la automatización de las pruebas jamás podrá cubrir el 100 % de cobertura del software.
- No es conveniente automatizar una interfaz de usuario no establecida.
- Si la aplicación de usuario cambia constantemente, el costo de mantenimiento de las pruebas automatizadas será muy alto.
- Bajo ciertos paradigmas de implementación los testers podrían requerir tener un buen conocimiento de programación.

1.2. Pirámide de la automatización de pruebas

La pirámide de automatización de pruebas, es un concepto que fue introducido por *Cohn* en su artículo «*Succeeding with Agile*», y describe como equilibrar la automatización, comenzando con las pruebas unitarias en el nivel más bajo, y pasando a las pruebas de servicios; finalmente las pruebas de interfaz de usuario se encuentran en la parte superior, como puede apreciarse en la figura 1.1.

Las pruebas unitarias son rápidas y fiables, las pruebas en la capa de servicios permiten evaluar la lógica del negocio donde la interfaz de usuario no está involucrada, cuanto más alto sea el nivel en la pirámide, más lentas y frágiles resultan las pruebas.

Finalmente aunque se debe realizar alguna automatización de las pruebas de interfaz de usuario, estas pruebas son más lentas, más difíciles de mantener y se llegan a fallar más fácilmente.

1.2.1. La capa de interfaz

Cuando el foco de la evaluación es la interfaz de usuario, se requiere que la mayoría del código y la lógica de negocio estén completamente evaluados. El enfoque ahora se centra en simplemente asegurarse de que la propia interfaz de usuario esté funcionando correctamente. Las pruebas de interfaz de usuario son muy frágiles, estas pruebas necesitarán mantenerse en cualquier momento que cambie la interfaz de usuario, y como hay muchos factores que entran

en juego cuando se ejecuta una prueba que emula clics en una pantalla, estas pruebas pueden dar como resultado falsos negativos. Estas fallas en las pruebas no pueden ser ignoradas, pero tampoco debe gastarse mas tiempo en solucionar los problemas y mantener las pruebas de UI que en encontrar defectos de código reales.

Con un solido diseño de pruebas, las pruebas sobre la interfaz de usuario complementan muy bien el conjunto de pruebas de automatización.

1.3. End-to-End Testing

Las pruebas End-to-End es una metodología que se utiliza para probar si el flujo de una aplicación se esta realizando según lo diseñado de principio a fin. El propósito de llevar a cabo pruebas de extremo a extremo es identificar las dependencias del sistema y garantizar que la información correcta se transmita entre varios componentes del sistema. Toda la aplicación se prueba en un escenario del mundo real, como la comunicación con la base de datos, la red, el hardware, y otros.

La prueba generalmente se realiza en la capa UI y se usa para validar que el proceso de negocio de extremo a extremo funciona en todos los sistemas como una prueba "horizontal". A diferencia de las pruebas verticales que miden el rendimiento hacia arriba y hacia abajo de las capas de la pila de tecnología, las pruebas horizontales de extremo a extremo se utilizan para garantizar que los procesos de negocio de misión crítica funcionen. Las pruebas se ejecutan en la capa UI y miden si la prueba continúa o no al siguiente paso del proceso.

1.3.1. Buenas prácticas para pruebas E2E

Una prueba típica de E2E puede ser compleja, con múltiples pasos que requieren mucho tiempo para realizarlos manualmente. Esta complejidad también puede hacer que las pruebas E2E sean difíciles de automatizar y lentas de ejecutar, para ayudar a administrar los costos de las pruebas automatizadas de E2E a la vez que mantienen los beneficios, se recomienda seguir las siguientes practicas:

- Mantener una perspectiva de usuario final.
- Limitar las pruebas de excepción.
- Aplicar análisis de riesgo.
- Ejecutar las pruebas en el orden correcto.
- Manejar apropiadamente el entorno de prueba.
- Separar la lógica de prueba de los elementos de interfaz de usuario.
- Manejar correctamente la espera de los elementos de interfaz de usuario.
- Escoger los dispositivos adecuados.
- Optimizar el proceso de configuración y desmontaje de la prueba.

1.4. Frameworks de automatización

La automatización generalmente se interpreta como el manejo automático de procesos a través de algoritmos inteligentes que involucran poca o ninguna intervención humana. En el caso del software, significa realizar varias pruebas en aplicaciones de software utilizando herramientas de automatización que son versiones con licencia o de código abierto. En términos técnicos, el framework de automatización de pruebas es un conjunto personalizado de componentes interactivos que facilitan la ejecución de las pruebas con rutinas y la generación de reportes completos de los resultados de las pruebas.

Dependiendo de como se desee abordar la creación de un framework y los requisitos de automatización del proyecto, se cuentan con diferentes clasificaciones:

Lineal: Son aquellos que registran los pasos de prueba y luego reproducir la rutina automáticamente para realizar la prueba.

Basado en módulos: Estos dividen la aplicación bajo pruebas (AUT) en varios módulos lógicos y poco acoplados. Para cada modulo se crea una rutina de prueba separada e independiente.

De arquitectura de biblioteca: Estos requieren determinar los pasos comunes de las pruebas, agruparlos en funciones en una biblioteca de funciones y llamar estas funciones en las rutinas de prueba cuando sea necesario.

Basado en datos: Se enfoca en separar la lógica de las rutinas de prueba y los datos utilizados. Con esto se puede hacer que las rutinas de prueba funcionen fácilmente para diferentes conjuntos de datos.

Orientado por palabras clave: Se enfoca en separar la parte técnica o de codificación del caso de prueba y de los pasos de necesarios de la prueba, para facilitar a una persona no técnica a entender bien la automatización.

Híbrido: Son la combinación de dos o mas marcos mencionados anteriormente, que intenta aprovechar los puntos fuertes y los beneficios de otros marcos para el entorno de prueba.

Bibliografía

- [1] Cohn, Mike.
The Forgotten Layer of the Test Automation Pyramid
Extraído el 12 de Marzo del 2018, de
<https://www.mountaingoatsoftware.com/blog/the-forgotten-layer-of-the-test-automation-pyramid>
- [2] Francino, Yvette.
The test automation basics every software developer should know
Extraído el 12 de Marzo del 2018, de
<https://techbeacon.com/app-dev-testing/test-automation-basics-every-software-developer-should-know>
- [3] King, Jackie.
10 Best Practices in Test Automation #9: Plan E2E Testing
Extraído el 13 de Marzo del 2018, de
<https://www.ranorex.com/blog/10-best-practices-test-automation-9-plan-e2e-testing>
- [4] Nguyen, Hien D.
Test automation frameworks: Everything you need to know
Extraído el 13 de Marzo del 2018, de
<https://www.logigear.com/blog/test-automation/test-automation-frameworks-everything-you-need-to-know>
- [5] Sharma, Laskshay.
What does End-to-End test mean?
Extraído el 13 de Marzo del 2018, de
<https://www.toolsqa.com/software-testing/what-does-end-to-end-test-mean>
- [6] STC, Admin.
What is Automation Testing?
Extraído el 12 de Marzo del 2018, de
<https://www.softwaretestingclass.com/what-is-automation-testing>
- [7] Techlabs, Maruti.
Everything you need to know about test automation frameworks
Extraído el 13 de Marzo del 2018, de
<https://www.marutitech.com/test-automation-frameworks>