**Tarea 2 – Testing: Proyectos de automatización de pruebas**

* Proyectos de automatización relacionados con los diferentes niveles de testing:

1. **Pruebas unitarias automatizadas:** Estos proyectos se centran en la automatización de las pruebas a nivel de unidad, donde se prueba cada componente individual de un sistema de forma aislada. Algunas herramientas populares para automatizar pruebas unitarias son JUnit (para Java), NUnit (para .NET), PyTest (para Python) y Jasmine (para JavaScript).
2. **Pruebas de API automatizadas:** Estos proyectos se centran en la automatización de las pruebas de integración de API. Utilicen herramientas como Postman, SoapUI o RestAssured para enviar solicitudes HTTP a las API y verificar las respuestas. Puedes automatizar pruebas de integración como la autenticación, validación de parámetros y verificación de respuestas.
3. **Pruebas de interfaz de usuario automatizadas:** Estos proyectos se enfocan en la automatización de las pruebas de la interfaz de usuario de un sistema. Utilice herramientas como Selenium WebDriver, Cypress o Puppeteer para simular la interacción del usuario con la aplicación y verificar que la interfaz responde correctamente. Puede automatizar acciones como hacer clic en botones, ingresar datos en formularios y verificar la visualización correcta de elementos.
4. **Pruebas de aceptación automatizadas:** Estos proyectos se centran en la automatización de las pruebas de aceptación, que verifican si un sistema cumple con los requisitos del cliente o usuario final. Puede utilizar herramientas como Cucumber, SpecFlow o Behave, que permiten escribir pruebas en lenguaje natural y definir casos de prueba en formato de historias de usuario. Estas herramientas pueden ejecutar las pruebas y generar informes legibles para las partes interesadas.

Ejemplos:

* JUnit: es un popular marco de prueba de código abierto para Java. Se usa ampliamente para pruebas unitarias de aplicaciones Java para garantizar que las unidades individuales de código, como métodos o clases, funcionen correctamente. JUnit proporciona un conjunto de anotaciones y afirmaciones que ayudan a los desarrolladores a escribir casos de prueba y verificar el comportamiento esperado de su código.
* SoapUI: permite crear y ejecutar pruebas funcionales para servicios web SOAP y RESTful. Puede definir casos de prueba, especificar parámetros de entrada y validar el resultado esperado. SoapUI proporciona una interfaz fácil de usar para crear y organizar conjuntos de pruebas y casos de prueba.
* Selenium: permite automatizar los navegadores web, incluidos Chrome, Firefox, Safari, Internet Explorer y otros. Puede escribir secuencias de comandos en lenguajes de programación como Java, Python, C# o JavaScript para controlar el navegador, navegar por páginas web, interactuar con elementos y realizar diversas acciones.
* Cucumber: herramienta de prueba de código abierto que admite la metodología de desarrollo impulsado por el comportamiento (BDD). Proporciona un marco para escribir especificaciones ejecutables en un formato de lenguaje natural que puede ser fácilmente entendido por partes interesadas tanto técnicas como no técnicas. Cucumber permite la colaboración entre desarrolladores, evaluadores y partes interesadas del negocio para definir y automatizar escenarios de prueba.