

Objetivo

Desarrollar un Generador de Escenarios de Prueba Funcionales y un Generador de Escenarios de Prueba Integrales a partir de Casos de Uso descritos en el RDL (*Requirements Definition Language*, Lenguaje de Definición de Requerimientos) de la empresa Softtek®.

Introducción

Anteriormente, probar era una actividad poco atractiva entre los programadores, sin embargo, debido a la creciente demanda de sistemas cada vez más complejos, dicha actividad ganó importancia, ya que con ésta se detectan errores en el código o se identifican defectos y comportamientos alejados del esperado. Para llevar a cabo las acciones anteriores, se recurre al uso de Escenarios de Prueba (especificaciones del proceso de ejecutar un producto de software para encontrar problemas en este [1]) y de Casos de Prueba (Escenarios de Prueba completados con información específica), pero, un componente de software puede generar cientos de estos escenarios, por lo que su identificación es costosa, complicada y tardada, así que los esfuerzos actuales se centran en la generación automática de pruebas, no obstante, no se sabe exactamente de dónde partir para realizar este proceso, ya que algunos orígenes suelen introducir errores o requieren de la intervención de expertos, por consiguiente, el trabajo de tesis resuelve esta problemática en la generación de pruebas funcionales gracias al uso de Casos de Uso descritos en el RDL de la empresa, ya que éste permite la representación estandarizada de las características deseables de un software [2]; en el caso de las pruebas integrales se usaron diagramas de actividades.

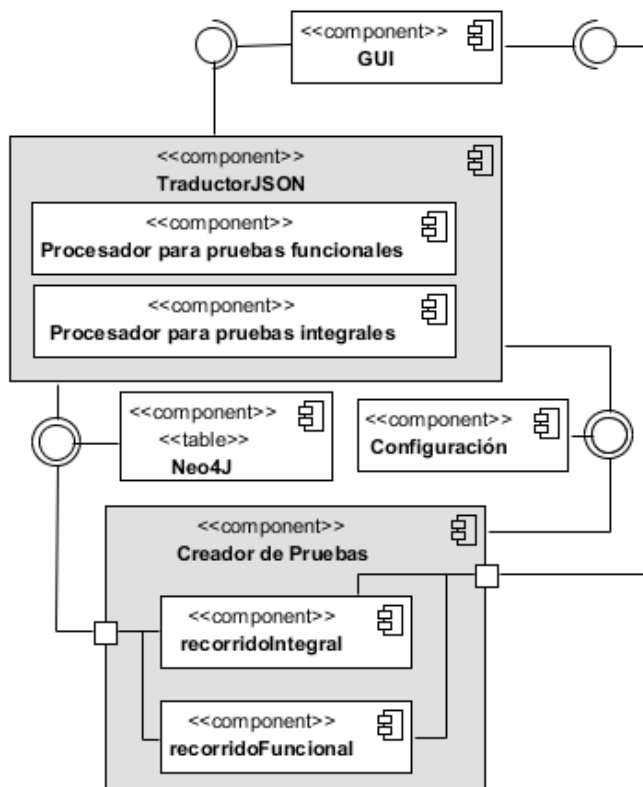


Figura 1. Arquitectura general del Generador de Escenarios de Pruebas Funcionales e Integrales

Arquitectura

En la Figura 1 se presenta la arquitectura general del Generador de Escenarios de Pruebas Funcionales e Integrales. Cuando se desean crear pruebas funcionales, el usuario debe ingresar a través del componente GUI uno o más archivos JSON equivalentes a Casos de Uso descritos con el RDL de la empresa, ya que en la circunstancia donde un Caso de Uso hace inclusiones a otros, es necesario agregar todos los archivos JSON correspondientes a dichas inclusiones mientras que, para generar pruebas integrales, solo es necesario ingresar el JSON relacionado al diagrama de actividades del sistema. Posteriormente, el TraductorJSON, se encarga de interpretar los archivos de entrada para generar grafos dirigidos y guardarlos en la base de datos orientada a grafos Neo4j, por lo que dependiendo del tipo de prueba que se esté generando, entra en acción el componente Procesador para pruebas funcionales o el Procesador para pruebas Integrales.

El componente Creador de Pruebas, está compuesto por los elementos recorridoIntegral y recorridoFuncional, ambos contienen algoritmos específicos que permiten la creación y priorización de escenarios de prueba a partir del recorrido del grafo almacenado en Neo4j.

Finalmente, debido a que el proceso de generación de grafos y de casos de prueba utilizan datos como los nombres de los cursos del Caso de Uso, o los tiempos para cada tipo de paso en diferentes momentos, dichos datos se encapsularon en el componente de Configuración.

Implementación de los Generadores de Escenarios de Prueba

Se construyeron dos API correspondientes a los Generadores de Escenarios de Prueba Funcionales e Integrales ya que este generador se va a incluir en las herramientas propias de la empresa. En la Figura 2, se muestra un fragmento de la vista principal de una aplicación creada exclusivamente para implementar las API antes mencionadas.

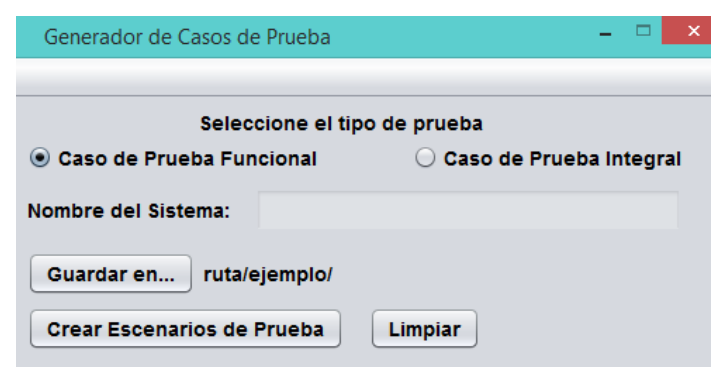


Figura 2. Interfaz gráfica del Generador de Casos de Prueba

Código o Nombre del Caso de Uso	ID Escenario de Prueba	ID Caso de Prueba	Categoría del Caso de Prueba	Pasos	Descripción de los Pasos de Prueba
CU0003: Registrar Prestamo	EP1	CP1	Happy Path	1	1.Indica que desea buscar al alumno o maestro
CU0003: Registrar Prestamo	EP1	CP1	Happy Path	2	2.Ver CU consultar usuario Biblioteca
CU0002: Consultar Usuario Biblioteca	EP1	CP1	Happy Path	3	1.Indica filtros de búsqueda(nombre propio y primer
CU0002: Consultar Usuario Biblioteca	EP1	CP1	Happy Path	4	2.Verifica que se haya indicado el

Figura 3. Fragmento del documento de Escenarios de Pruebas Funcionales

Conclusiones.

La utilización de un RDL como origen de casos de prueba dio los resultados esperados, ya que no se introducen errores en los escenarios de prueba generados; además, el uso de un grafo para la generación de dichas pruebas, permitió la creación de éstas de forma más óptima; el uso de Neo4j en el proyecto facilitó la creación y almacenamiento del dígrafo. Para el caso de los escenarios de prueba integrales, el manejo de diagramas de actividades garantizó la obtención de pruebas correspondientes al comportamiento del sistema.

Referencias

- [1] G. J. Myers, The Art of Software Testing, Second Edition, 2.a ed. New Jersey: Wiley, 2004.
- [2] M. del Carmen Gómez F., Notas del Curso: Análisis de Requerimientos, México, 2011.

Resultados

Las API correspondientes al Generador de Escenarios de Pruebas Funcionales e Integrales demostraron su utilidad y eficiencia en la empresa en varios casos de estudio, puesto que su implementación permitió la obtención de escenarios de pruebas en menos tiempo que su contraparte manual; dada la aceptación por parte de los usuarios y sus expectativas, se identificaron un gran número de funcionalidades a desarrollar en el futuro. En la Figura 3, se muestra un fragmento de uno de los documentos de Escenarios de Pruebas Funcionales.