

SISTEMA DE GENERACIÓN AUTOMÁTICA DE SCRIPTS DE EJECUCIÓN PARA PRUEBAS UNITARIAS EN APLICACIONES WEB

Sandra Marcela Giraldo Colorado¹, Jorge Eliécer Giraldo Plaza²

¹ Ingeniera Informática. Automatizadora en Tata Consultancy Services. Medellín, Colombia

² PhD(c) en ingeniería de sistemas e informática, Universidad Nacional de Colombia. Medellín, Colombia

Email autor de correspondencia: jegiraldp@gmail.com

RESUMEN

Se presenta la propuesta de un sistema para la transformación de scripts de pruebas de manera automática a scripts de ejecución, para pruebas unitarias en aplicaciones web. Dicha generación reduce la dependencia con las herramientas de automatización, optimizando la labor de los analistas de pruebas en la configuración de los scripts. El sistema se compone de una estrategia innovadora para la construcción de los scripts y un sistema de software basado en una arquitectura orientada a servicios web.

Palabras clave: Script de prueba, Script de ejecución, Automatización de pruebas, Servicio web.

Recibido: 18 de septiembre de 2013. Aceptado: 26 de noviembre de 2013.

Received: September 18th, 2013. Accepted: November 26th, 2013.

AUTOMATIC GENERATION SYSTEM OF EXECUTION SCRIPTS FOR UNIT TESTS IN WEB APPLICATIONS

ABSTRACT

A system for automated transformation of test scripts into execution scripts for unit testing in web applications is suggested. This system reduces dependence to automation tools, optimizing the test analysts functions in the configuration of the test scripts. The system is composed by an innovative strategy for building scripts and a software system based on web services architecture.

Keywords: *Test Script, Execution Script, Automation Testing, Web Service.*

1. INTRODUCCIÓN

Las pruebas automatizadas tienen como propósito aligerar el proceso de ejecución de pruebas, sin embargo traen consigo limitantes, como la alta dependencia que se tiene con las herramientas de automatización y la complejidad que conlleva su configuración a la hora de interactuar con ellas. Con este trabajo se plantea una investigación para determinar la solución a estas limitantes y brindar un aporte válido en la optimización del proceso de automatización de pruebas, mediante un sistema de generación automática de scripts de ejecución, el cual se implementa mediante una estrategia de generación y un sistema de software orientado a servicios web.

1.1. Objetivo general

Diseñar un sistema de generación automática de scripts de ejecución de pruebas unitarias basado en una arquitectura orientada a servicios.

1.2. Objetivos específicos

- Definir un esquema de metadatos para el diseño de casos prueba con el fin de utilizarlo como insumo principal en el desarrollo del sistema.
- Proponer la estrategia de generación de scripts de ejecución a partir de un formato de casos prueba.
- Implementar la estrategia propuesta mediante un módulo web basado en una arquitectura basada en servicios para la obtención de un prototipo de validación funcional.
- Seleccionar un caso de estudio con sus respectivos requisitos funcionales para la validación del sistema propuesto.

2. FUNDAMENTO TEÓRICO

En esta sección se exponen los conceptos relacionados con la automatización de pruebas de software, así como las herramientas utilizadas en dicho proceso.

2.1. Proceso de pruebas de software

De acuerdo al estándar ISTQB, (International Software Testing Qualifications Board) las pruebas de software se definen como: “El proceso que consiste en todas las actividades del ciclo de vida, tanto estáticas como dinámicas, relacionadas con la planificación, preparación y evaluación de productos de software y productos relacionados

con el trabajo para determinar que cumplen los requisitos especificados, para demostrar que son aptos para el propósito y para detectar defectos.”

Reciben el nombre de pruebas funcionales, aquellos procesos de evaluación del software enfocados en la validación del cumplimiento de los requisitos funcionales de un sistema informático, certifican que el producto de software ejecute de forma correcta las actividades para las cuales fue desarrollado.

Éste tipo de pruebas son aquellas que se conocen con el nombre de pruebas de caja negra, puesto que para el analista encargado de ejecutarlas es desconocido el proceso de operación al interior del sistema y solo centra su atención en el análisis de insumos y resultados. Existen varios tipos de pruebas funcionales de software, entre las cuales se encuentran las siguientes: Pruebas de humo, pruebas de regresión, pruebas de aceptación y pruebas unitarias o modulares.

2.2. Derivación de casos de prueba

Existen diferentes técnicas de derivación de casos de prueba para probar una unidad de software. Estas se pueden dividir en dos, de caja negra (funcional) o de caja blanca (estructural). Las técnicas de caja negra, para derivar los casos de prueba, se basan únicamente en la funcionalidad que debe proveer la unidad. Por otro lado, las técnicas de caja blanca consideran la implementación para derivar los datos de prueba.

2.3. Pruebas de software automatizadas

Las pruebas automatizadas son el uso de software para el control de: la ejecución de las pruebas, la comparación de los resultados reales con los resultados previstos, la creación de las condiciones previas de la prueba, otros controles de las pruebas y funciones de reporte de resultados. De igual forma la automatización de pruebas es implementada mediante herramientas de software que se encargan de realizar validaciones que no requieran intervención manual y que puedan ser ejecutadas sin vigilancia o monitoreo constante por parte del recurso humano. En pocas palabras desarrollar una prueba automatizada es automatizar el proceso de pruebas manual ya existente.

Las pruebas automatizadas pueden ser aplicadas a pruebas de humo, pruebas de regresión, pruebas de migración, pruebas integrales, entre otras. De igual forma pueden ser utilizadas en creación y actualización de set de datos. Es importante tener en cuenta que para la implementación de la automatización es necesario contar con un proceso de pruebas establecido y formalizado al igual que con aplicativos estables con requerimientos estables.

El proceso de automatización de pruebas es ideal para la realización de operaciones altamente repetitivas y para el trabajo con aplicativos que requieren mantenimiento continuo. Debido a que es un proceso que demanda una inversión de tiempo y personal capacitado, se hace necesario que los proyectos tengan ROI (Return of Investment) para que el proceso se vea balanceado conforme al número de corridas que se realicen.

2.4. Proceso de automatización

Para llevar a cabo un plan de pruebas automatizadas se requiere una plantilla de casos prueba, o si bien un script de prueba que será aquel que deberá transformarse para convertirlo en un script de ejecución. Se entiende por script de ejecución al programa o código en sintaxis formal que automatiza la ejecución de una prueba, contiene inmerso el desarrollo de un requerimiento específico del proyecto que por medio de instrucciones y comandos, puede ejecutar más de un caso prueba de acuerdo a la estructuración del mismo.

3. REVISIÓN LITERARIA

En esta sección se exponen los conceptos investigados tomados como fundamentos para el desarrollo del sistema propuesto.

3.1. Pruebas Unitarias Web

Gutiérrez et. al 2007, en [1] resaltan las fases del modelo de automatización de pruebas dividiéndolas en 3 niveles separados, así: automatización del proceso de generación de casos de prueba a partir de los requisitos, automatización de la ejecución de los casos de prueba y comprobación de resultados. Gutiérrez et. al 2008, en [2], trabajan la temática de pruebas unitarias y pruebas de aceptación, resalta artefactos resultantes de dicha etapa y el levantamiento de requisitos, sin embargo no

involucra el tema de automatización ni de ejecución o transformación de scripts de prueba. Involucra JUnit como herramienta de trabajo para la implementación de las pruebas unitarias y una serie de escenarios que describen técnicas de programación orientada a aspectos.

Díaz et. al 2009, en [3] plantean las pruebas de software como proceso transversal al ciclo de vida del desarrollo de software, se presentan aspectos importantes de los distintos tipos de pruebas, y un análisis comparativo de herramientas de código abierto agrupadas según el tipo de prueba que se puede realizar en una aplicación Web. Algunas desventajas que se encontraron en el estudio a este proyecto fue el hecho que enfoca su investigación en herramientas y frameworks de automatización de código libre más no en el proceso de creación de pruebas y scripts de ejecución como tal.

Yagüe y Garbajosa, 2009, en [4] desarrollan diferentes tipos de pruebas funcionales al igual que las técnicas llevadas a cabo con el fin de generar casos prueba insumo para las ejecuciones. La limitante encontrada en este trabajo es la falta de acoplamiento de estas pruebas funcionales con automatización.

Marín et al., 2012 en [5], reflexionan sobre la necesidad de disminuir el trabajo manual de los analistas de pruebas en aplicaciones web, ya que sus características tienden a: (1) Ser automodificables, (2) tener un comportamiento autónomo, (3) manejar información asincrónica de vital importancia, (4) intensivas en información y conocimiento, y (5) funcionalidad a gran escala. Por lo anterior se plantea un modelo basado en la búsqueda de recursos e información, donde sea posible construir dinámicamente casos de prueba, sin embargo, persiste la dependencia, en este caso a conocimientos específicos en el área de análisis de inferencias.

3.2. Automatización de pruebas Unitarias

Esmite et. al 2007, en [6], plantean que el objetivo de la investigación en el área de pruebas debe ser su automatización, para ello diseñan casos de estudio para diferentes navegadores y diferentes plataformas de desarrollo. Muestra valores reales de los resultados de un proyecto llevado a cabo y determina riesgos en el mismo relacionados con las limitantes que aporta la herramienta de automatización y las diversas soluciones que mitigaron los riesgos ya consolidados.

Rodríguez y Camarillo, 2012, en [7] realizan una investigación sobre las distintas herramientas CAST (Computer Aided Software Testing). Involucra el tema de automatización y pruebas funcionales sobre aplicativos web, presenta de igual forma herramientas que brindan soporte a la generación de casos prueba. Involucra IBM Rational Functional Tester herramienta reconocida en el sector de pruebas automatizadas. Una debilidad del trabajo es que no determina aspectos técnicos respectivos a su funcionamiento o implantación.

Villa y Giraldo, 2012, en [8] proponen una estrategia para la automatización de pruebas unitarias en códigos PHP, empleando la librería PHP Unit. Determinando buenas prácticas de programación, aspecto que se considera relevante a la hora de determinar los estándares para la creación de la plantilla insumo de casos prueba que será transformada en el script de ejecución. Abarca el proceso de diseño de los casos prueba y su ejecución, lo cual brinda un panorama amplio de la prueba permitiendo analizar los resultados obtenidos. La desventaja encontrada en este proyecto es que limita el lenguaje de programación a PHP.

3.3. Generación automática de scripts de pruebas

Quintana y Solari, 2006, en [9], presentan las técnicas que se reportan como más eficientes y eficaces para la generación automática de casos de prueba estructurales y las clasifica de acuerdo al método de implementación de la técnica. Este proceso se usa ampliamente en la industria para asegurar la calidad del software.

De la Cal et. al 2006, en [10] desarrollan el tema de generación automática de pruebas, evidenciando un punto de referencia para el proyecto de la creación de scripts de prueba manuales y archivos de entrada y salida, denotando el alto consumo de tiempo que demandan dichas actividades y presentando una herramienta para la generación automática de scripts de prueba que satisface la necesidad encontrada. La limitante principal es la falta de información con respecto a los tipos de aplicativos sobre los cuales trabaja dicha herramienta y la poca especificidad al respecto de este tema.

Vallespir et. al 2009, en [11], se basan directamente en la generación automática de casos prueba unitarios mediante métodos establecidos como las clases de equivalencia y los valores límite. Es

necesario tener en cuenta este tipo de procesos, ya que los casos prueba estandarizados son el insumo del script de prueba, por lo tanto debe estar desarrollado correctamente para que sea analizado por el robot.

Marback et al, 2012 en [12] presentan un sistema de verificación automática para actividades de regresión de pruebas, su implementación se realiza en sobre códigos PHP, sin embargo solo se centra en datos de tipo numérico y texto, restringiéndose a los campos de fecha y booleanos.

3.4. Análisis de la revisión literaria.

Se identifica una segmentación de temáticas en los trabajos, separando los aspectos de pruebas unitarias, los tipos de aplicativos en un área específica y los proyectos de automatización en un ámbito diferente sin incluir los filtros previamente mencionados. Aquellos proyectos que involucran el aspecto de automatización, utilizan herramientas de automatización para el desarrollo de las investigaciones, evidenciando la dependencia existente con estas y no plantean estrategias alternas para llevar a cabo las pruebas unitarias sin su utilización. La mayor parte de los trabajos de automatización enfocados a generación de scripts de ejecución no involucran los scripts de prueba.

4. METODOLOGÍA

En esta sección se presenta el esquema de metadatos o script de prueba diseñado para la generación automática de scripts de ejecución. Se compone de tres partes, así:

Bloque de Descripción: Segmento encargado de brindar información general y específica del caso de prueba, con el objetivo de identificarlo y categorizarlo. También se encuentra la información del proyecto asociado al caso prueba y detalles tales como tiempos de ejecución, iteraciones, y personal asignado para la ejecución de dicha prueba.

Bloque de Pasos: Segmento que contiene el listado de pasos numerados del caso de prueba, dichos pasos se deben encontrar diseñados de forma clara, concisa y explícita, con el fin de eliminar ambigüedades en la interpretación de los mismos a la hora de la ejecución. En este bloque se encuentran campos tales como la secuencia de pasos general, el objeto con el que se interactúa, la

acción ejecutada, y los parámetros enviados para la interacción en caso de ser requeridos.

Bloque de Resultados: Segmento que contiene los resultados de las ejecuciones de los casos prueba, en dicho bloque se encuentra campos tales como,

resultado esperado y resultado obtenido, que permiten la comparación en tiempo real del funcionamiento ideal del aplicativo y el funcionamiento con el que se cuenta actualmente.

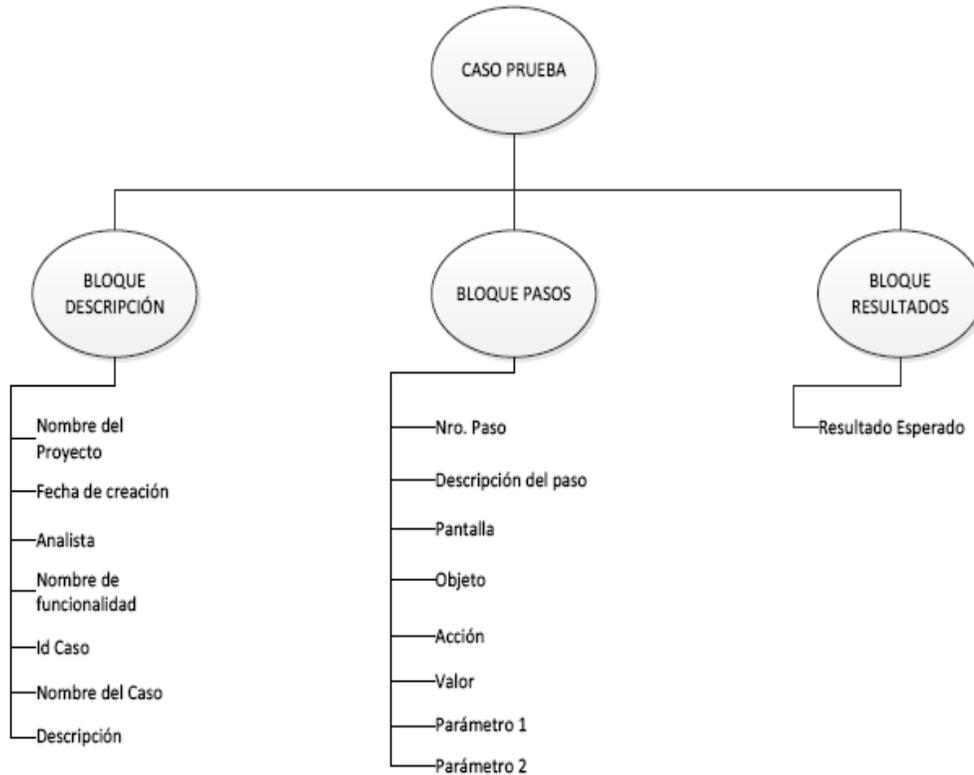


Figura 1. Estructura de caso de prueba definido.

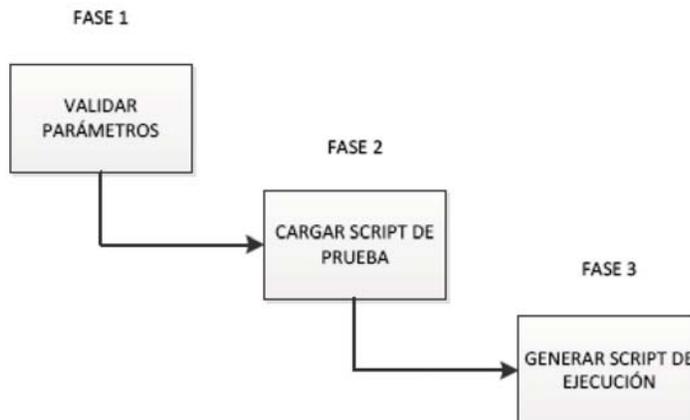


Figura 2. Estrategia propuesta

4.1. Estrategia propuesta para la generación automática de scripts

La estrategia propuesta se fomenta en un sistema de generación de código basado en palabras nativas reservadas las cuales denotarán una acción sobre una aplicación a probar, con el fin de obtener un comportamiento específico del mismo.

El sistema consta de tres fases generales principales: La fase de validación de parámetros, la fase de carga del script de prueba y la fase de generación de script de ejecución. A continuación se muestran dichas fases en la Figura 2.

Fase 1: Validar parámetros. El funcionamiento del sistema inicia con el despliegue de la interfaz, permitiendo al usuario manipular los campos asignados tales como la Ruta de Script donde se ingresará o se seleccionará la ruta del archivo el cual posee en palabras naturales, acciones o funciones a desarrollar (Script de prueba).

De igual forma se debe especificar la ruta que almacenará los log de resultados, archivo que contendrá el registro de las acciones realizadas durante la generación del script de ejecución.

Fase 2: Cargar script de prueba. Esta fase se inicia al momento que el usuario da clic en el botón "Confirmar". Posteriormente se valida la estructura general del script de prueba mediante el cargado del mismo y llevando a cabo un recorrido que identifique unas características predeterminadas propias del esquema de metadatos.

Fase 3: Generación script de prueba. En esta fase se inicia la lectura por filas de cada paso en el script de prueba de forma que se identifique las palabras reservadas y genere el código de la acción. La anterior tarea podrá tardar dependiendo del volumen de información en cada paso, del volumen de pasos incluidos y de la complejidad de los mismos. Durante el proceso, toda actividad se registrara en el log, con el objetivo de que estos sirvan como seguimiento del control de errores del script de ejecución.

5. RESULTADOS

5.1. Implementación y validación de la propuesta mediante SOA

En esta sección se presenta en detalle la implementación de la estrategia propuesta. Para cada una de las fases se implementa un servicio web. La implementación de la estrategia consta de tres servicios principales propuestos en el siguiente orden:

Fase 1: Se implementa mediante el Servicio web de transporte de archivos. Este servicio consta de pequeñas subrutinas que tienen como objetivo realizar la validación de la dirección de origen y la dirección de destino. Así mismo se encarga de realizar el cambio de nombre de archivo.

Fase 2: Se lleva a cabo mediante el *Servicio web de validación*. Dicho servicio consta de pequeñas subrutinas que tienen como objetivo realizar la validación de los elementos que se exponen a continuación, dicho servicio es activado por cada registro detectado en la tabla presentada en la aplicación.

Para el desarrollo de este servicio web es necesario crear un archivo que contenga la información de los elementos principales identificables en el aplicativo web y su correspondencia con la acción permitida a ejecutar con los mismos, por lo tanto, se crea un archivo con dichos datos mostrado a continuación, en la Figura 3.

Fase 3: Se desarrolló un *Servicio web de codificación*. El servicio de codificación consta de pequeñas subrutinas que tienen como objetivo realizar la validación de palabras reservadas y realización de transcripción. Este servicio es activado por cada registro detectado en la tabla presentada en la aplicación. Una vez el servicio web recibe la respuesta la expone al elemento invocador, quien para este caso corresponde a la interfaz.

Código	Código Elemento	Código Acción
1	Texto	Buscar
2	Texto	Leer
3	Texto	Copiar
4	Tabla	Buscar
5	Fila	Buscar
6	Columna	Buscar
7	Título	Buscar

Figura 3. Equivalencia de elementos y acciones

Para la validación se presenta las características del caso de estudio seleccionado para la aplicación de la estrategia y que sirve como base para la creación del script de prueba, insumo principal del sistema de generación automática de scripts de ejecución.

5.1.1. Características.

- La página <http://www.tata.com>, sitio web de TATA Consultancy Services, emplea hipervínculos que permite acceder a nuevas URL diferentes a la seleccionada principalmente, brindando la posibilidad de validar la interacción del sistema con otras ventanas y otros aplicativos web.

- La posibilidad de realizar pruebas de integración entre aplicativos permite el diseño de casos de prueba más robustos y extensos con mayor variedad de objetos por validar y mayor cantidad de código fuente por generar.

- La estructura del aplicativo y el diseño de los formularios de ingreso de información, permite visualizar en detalle los atributos, evitando que el analista deba suponer o consultar con el equipo de desarrollo los atributos necesarios. De esta manera se facilita el proceso de diseño del script de prueba.

- La página <http://www.tata.com> presenta una estabilidad en el ambiente que permite realizar diseños de casos prueba viables en un lapso de tiempo prolongado, ya que no modifica su estructura continuamente y los artículos consultados permanecen en sus rutas originales por largo tiempo.

- La página se encuentra desarrollada en HTML básico compatible con la versión del prototipo desarrollado, permitiendo la fácil integración con el mismo y la rápida lectura de los objetos que la componen para la validación de las acciones ejecutadas y de igual forma, la traducción del script de prueba al código fuente, o script de ejecución.

- Se presenta la posibilidad de segmentar el aplicativo por módulos o funcionalidades fácilmente debido a la gran cantidad de productos y servicios con los que cuenta la compañía, y al desarrollo de un ambiente diferente para cada uno de éstos, facilitando el proceso de pruebas y diseño de las mismas.

- Alcance: Comprobar la correcta generación del código fuente correspondiente con algunos objetos preseleccionados encontrados en una página web.

- A continuación se determinan los módulos tomados en cuenta para el diseño del script de prueba con la URL seleccionada.

<http://www.tata.com>

Módulo 1: intreviews – Jaguar Land Rover

Módulo 2: intreviews – Indian hotels

6. CONCLUSIONES

- Este documento presenta la propuesta de una estrategia de generación de scripts de ejecución de pruebas unitarias web, mediante el desarrollo de un esquema de casos de prueba y su posterior implementación en una solución basada en SOA. A partir de dicho trabajo de puede concluir lo siguiente:

- La utilización de campos adicionales en el script de prueba tales como: pantalla, objeto, acción, valor, parámetro 1 y parámetro 2, permiten una mejor interacción con el aplicativo y la creación de script automáticos de ejecución.

- El sistema expone una novedosa manera de abordar la generación de scripts de pruebas, mediante la propuesta de una estrategia por fases, que fácilmente puede ser implementada por tecnologías orientadas a servicios web.

- La implementación de una arquitectura orientada a servicios para el diseño del sistema brinda un mejor aprovechamiento del mismo para su

acoplamiento con diferentes proyectos y posterior escalabilidad.

- El sistema propuesto permite realizar la validación de diferentes casos de estudio tales como los escenarios de lógica de negocio, validación de obligatoriedad y GUI (Graphical User Interface), mediante el diseño de casos prueba personalizados por proyecto.

7. AGRADECIMIENTOS

Este artículo es el resultado del trabajo de grado "Sistema de generación automática de scripts de ejecución para pruebas unitarias en aplicaciones web". Los autores agradecen al Politécnico Colombiano Jaime Isaza Cadavid y al comité de proyectos y trabajos de grado, del programa de Ingeniería Informática que con sus valiosos aportes contribuyeron a la mejora de este proyecto.

8. REFERENCIAS BIBLIOGRÁFICAS

- [1] Gutiérrez, Javier; Mejía, Manuel, Torres, Arturo & TORRES, Jesús. Implementación de pruebas del sistema. Un caso práctico. Departamento de Lenguajes y Sistemas Informáticos Universidad de Sevilla. (2007)
- [2] Gutiérrez Javier, Villadiego, Darío; Escalona María & MEJÍA Manuel. Aplicación de la programación orientada a aspectos en el diseño e implementación de pruebas funcionales. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. (2008)
- [3] Díaz, Javier; Banchoff, Claudia, Rodríguez, Anahí & Soria, Valeria. Herramientas open source para testing de aplicaciones Web. Evaluación y usos. Facultad de Informática. Universidad de La Plata. Buenos Aires. Argentina. (2009)
- [4] Yagüe, Agustin & Garbajosa Juan. Las pruebas en metodologías ágiles y convencionales. Universidad Politécnica de Madrid (UPM). System and Software Technology Group (SYST). (2009)
- [5] Marin, Beatriz; Vos, Tanja; Giachetti, Giovanni; Baars, Arthur; Tonella, Paolo, "Towards testing future Web applications," Research Challenges in Information Science (RCIS), 2011 Fifth International Conference. (2011)
- [6] Esmite, Ignacio; Farías, Mauricio; Farías, nicolás & Pérez, Beatriz. Automatización y Gestión de las Pruebas Funcionales usando Herramientas Open Source. Centro de Ensayos de Software (CES), Universidad de la República. Montevideo, Uruguay. (2007)
- [7] Rodriguez, Gloria & Camarillo Juan. Herramientas CAST. Investigación sobre las distintas herramientas CAST. (2012).
- [8] Villa, Alejandro & Giraldo Plaza Jorge. Estrategia de diseño para la automatización de pruebas unitarias de códigos PHP utilizando el framework PHP UNIT. Ingeniería informática, Politécnico Colombiano Jaime Isaza Cadavid. (2012).
- [9] QUINTANA, Gerardo & SOLARI Martin. Estudio de Mapeo Sistemático sobre Experimentos de Generación Automática de Casos de Prueba Estructurales. Universidad ORT Uruguay. (2006)
- [10] De La Cal Marín, Enrique; García Marco & Suárez-Cabal María. Generación automática de pruebas de validación de prácticas de sistemas distribuidos. Dpto. de Informática. Universidad de Oviedo. (2006).
- [11] Vallespir, Diego. Generación Automática de Casos de Prueba Unitarios para Objetos, Universidad de la República, Grupo de Ingeniería de Software (Gris), Montevideo, Uruguay. (2009)
- [12] Marback, Aaron; Hyunsook Do & Ehresmann, Nathan. An Effective Regression Testing Approach for PHP Web Applications. Software Testing, Verification and Validation (ICST), 2012 IEEE Fifth International Conference. (2012).