CAPÍTULO 4

PRUEBAS E IMPLEMENTACIÓN DE LA APLICACIÓN

CONCEPTOS DE PRUEBAS DE APLICACIÓN

El departamento de "Testing" se encarga de diseñar, planear y aplicar el rol de pruebas a los sistemas que el PROVEEDOR tiene a su cargo, con el fin de asegurar la calidad y seguridad de los productos que ofrece a sus clientes. Aunque el equipo de desarrollo no tuvo participación en esta etapa del desarrollo del sistema LMP, aplicamos varias técnicas de testing de software, que aunque no fueron a nivel de las que aplica el departamento de "Testing", nos permitieron validar la operación del software e ir mejorando la aplicación.

Probar es el proceso de ejecución de software con la intención de encontrar y corregir errores. Ya que los sistemas web residen en red y son accesibles desde muchos sistemas operativos, navegadores de varios dispositivos, plataformas de hardware la búsqueda de errores representa un reto significativo. Las pruebas se enfocan en contenido, función, estructura, usabilidad, navegabilidad, rendimiento, compatibilidad, interacción, capacidad y seguridad. Estas pruebas ocurren mientras se diseña y desarrolla la aplicación y pruebas que se llevan a cabo una vez que se implementa.

Como parte de las pruebas al sistema LMP se examinaron las siguientes dimensiones de calidad.

Contenido: Evalúa tanto en el nivel sintáctico como en el semántico. En el primero, se valora el vocabulario, puntuación y gramática de los textos en la aplicación. En el segundo se valora la consistencia de la información.

Función: Se prueba para descubrir errores que indican falta de conformidad con los requerimientos del cliente.

Estructura: Se valora para garantizar que entrega adecuadamente el contenido y la función de la aplicación.

Usabilidad: Se prueba para asegurar que la interfaz soporta cada tipo de usuario.

Navegabilidad: Se prueba para asegurar que toda la sintaxis y la semántica de navegación se ejecutan para descubrir cualquier error de navegación (por ejemplo, vínculos muertos, inadecuados y erróneos).

Rendimiento: Se prueba bajo condiciones operativas, configuraciones y cargas diferentes a fin de asegurar que el sistema responde a la interacción con el usuario y que maneja la carga operativa con un nivel aceptable.

Compatibilidad: Se prueba al ejecutar la aplicación en varias configuraciones del cliente como en el servidor.

Interoperabilidad: Se prueba para garantizar que la aplicación tiene la interfaz adecuada con otras aplicaciones y/o bases de datos.

Seguridad: Se prueban al valorar las vulnerabilidades potenciales e intenta explotar cada una. Cualquier intento de penetración exitoso se estima como un fallo de seguridad.

El procedimiento básico de pruebas de aplicaciones Web se basa en definir entradas (información a partir de la cual la aplicación generara una serie de contenidos Web) y realizar comprobaciones sobre las salidas (contenidos Web generados: documentos HTML, JavaScript, documentos XML, etc.). Se trata, por tanto, de pruebas funcionales que se realizan mediante un enfoque de caja negra, es decir, los casos de prueba se construyen sin tener en cuenta la estructura interna de la aplicación sino únicamente sus entradas y salidas esperadas. Estas entradas y salidas son típicamente peticiones HTTP y documentos Web respectivamente.

En el caso de la aplicación LMP se realizaron diversas pruebas por parte del equipo de testing de la empresa, mi tarea en esta etapa como tal no fue realizar estas pruebas puntualmente pero mientras se desarrollo la aplicación utilizamos herramientas como:

Zero Day Scan es un servicio online gratuito que detecta las vulnerabilidades de una Web, como por ejemplo el Cross-Site Scripting (XSS), inyección SQL, y otras. A partir de este análisis, genera un resumen de resultados. Requiere la confirmación del propietario de la página.

Características:

- No requiere instalación, es un servicio gratuito.
- Detecta ataques de Cross Site Scripting (XSS)
- Detecta directorios ocultos y archives de backup.
- Comprueba vulnerabilidades de seguridad conocidas.
- Busca vulnerabilidades de SQL injections.
- Realiza Website Fingerprinting

Además de Powerfuzzer que permite crear tests personalizados para de aplicaciones Web con la intención de detectar agujeros de seguridad. En esencia, se trata de un simple escáner de aplicaciones. Algunos de los tests que ofrece son los siguientes:

- Código de páginas cruzadas (XSS)
- Inyecciones (SQL, LDAP, código, comandos, y XPATh)
- CRLF
- Estados HTTP 500

Por ejemplo en la Figura 4.1 se muestra cómo la aplicación Powerfuzzer intenta hacer diversos ataques a la aplicación, para detectar alguna vulnerabilidad en el sistema de login, sin éxito.

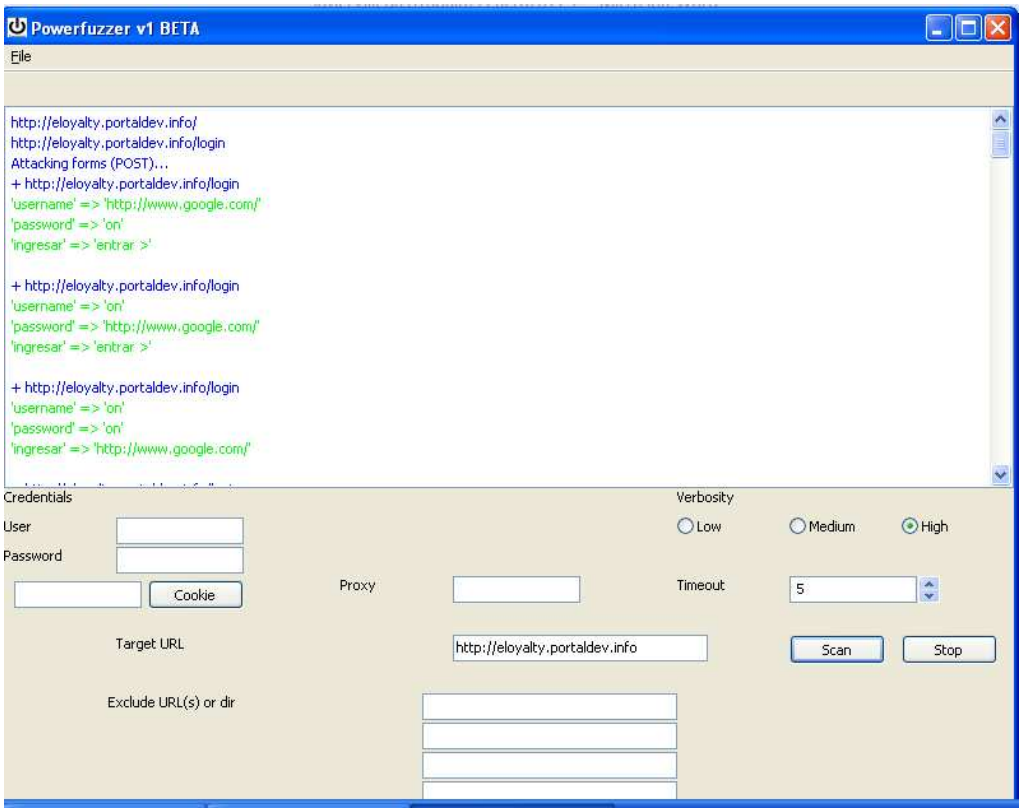

Figura 4.1 Test de seguridad a login del sistema LMP

En el Anexo 4 se listan una serie de herramientas que nos ayudan a realizar pruebas a un sistema.

PRUEBAS DE CAJA NEGRA

El sistema de pruebas de caja negra no considera la codificación dentro de sus parámetros a evaluar, es decir, que no están basadas en el conocimiento del diseño interno del programa. Estas pruebas se enfocan en los requerimientos establecidos y en la funcionalidad del sistema.

Mientras se desarrolló la aplicación realizamos pruebas de caja negra por ejemplo en la pantalla de login mostrar los mensajes de error correctamente como se muestra en la Figura 4.2. En la que se muestra el mensaje de error en caso de ingresar con un usuario/contraseña incorrectos.

Figura 4.2 Error de login en sistema LMP

PRUEBAS DE CAJA BLANCA

Al contrario de las pruebas de caja negra, éstas se basan en el conocimiento de la lógica interna del código del sistema. Las pruebas contemplan los distintos caminos que se pueden generar gracias a las estructuras condicionales, a los distintos estados del mismo, etc.

Debido a que el equipo de desarrollo conoce todas las condiciones del sistema LMP podemos realizar pruebas con diferentes escenarios y tipos de datos de entrada de información en el sistema por ejemplo en el alta de usuarios podemos probar el ingreso de la información con diferentes textos de entrada y en caso de tener algún error corregirlo de manera rápida y ágil. Por ejemplo la Figura 4.3 muestra los errores posibles en el alta de un usuario.

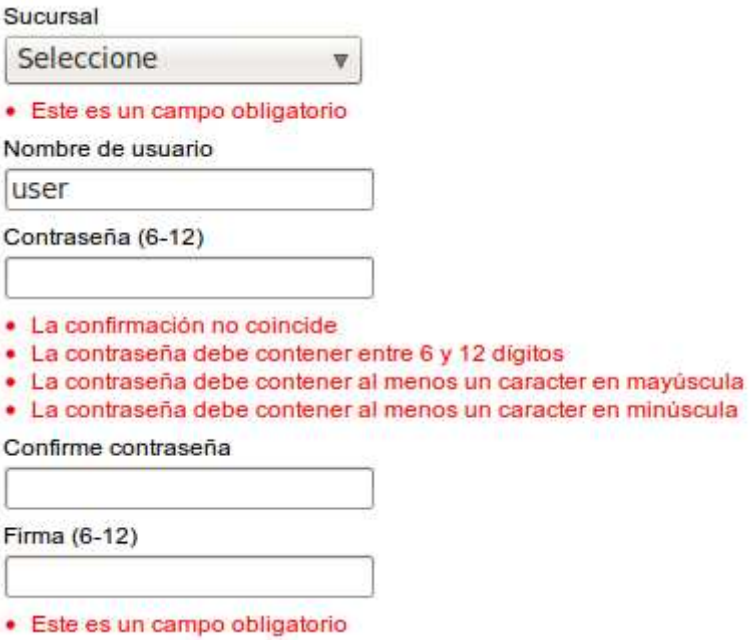

Figura 4.3 Errores en alta de Usuarios en el Sistema LMP

IMPLEMENTACIÓN Y MANTENIMIENTO DE SOFTWARE

Después de realizar las pruebas en el sistema, el PROVEEDOR realizó la instalación de la aplicación en producción con el equipo de software y hardware solicitado al CLIENTE. Durante esta etapa el equipo de instalación realizó el deployment de la aplicación ya previamente probada, aun así se realizaron pruebas de caja negra en la aplicación final para asegurar su correcto funcionamiento.

Éste comienza casi de inmediato. El software se libera a los usuarios finales y, en cuestión de días, los reportes de errores se filtran de vuelta hacia la organización de ingeniería de software. En semanas una clase de usuarios indica que el software debe cambiarse de modo que pueda ajustarse a las necesidades especiales de su entorno. Y en meses, otro grupo corporativo, que no quería saber nada del software cuando se liberó, ahora reconoce que puede ofrecerle beneficios inesperados. Necesitará algunas mejoras para hacer que funcione en su mundo.

El reto del mantenimiento del software comienza. Uno se enfrenta con una creciente lista de corrección de errores, peticiones de adaptación y mejoras categóricas que deben planearse, calendarizarse y, al final de cuentas, lograrse. Mucho antes, la fila creció bastante y el trabajo que implica amenaza con consumir los recursos disponibles.

Otra razón del problema de mantenimiento de software es la movilidad del personal. Es probable que el equipo o la persona responsable del software que hizo el trabajo original ya no esté más por ahí. O peor aun otras generaciones de personal de software modificaron el

sistema y se mudaron. Y puede ser que ya no quede alguien que tenga algún conocimiento del sistema heredado.

En el caso de LMP desde que se puso en producción fue sujeto de modificaciones y adaptaciones a procesos que otras áreas de la empresa del cliente necesitaba. La ventaja de la aplicación LMP es que está diseñada para poder realizar cambios de forma rápida y legible para los desarrolladores debido al uso del modelo MVC proporcionado por ZF.

En esta etapa de mantenimiento a la aplicación el equipo del PROVEEDOR da soporte a la aplicación a la fecha debido a que han surgido requerimientos de nuevas funcionalidades al sistema además de cuidar la estabilidad del mismo.

Durante este periodo he tenido la oportunidad de aplicar mis conocimientos en desarrollo de software especialmente para aplicar mejoras de funcionalidad las cuales se han acoplado a las nuevas necesidades de la empresa.