CAPÍTULO 6

PRUEBAS Y LIBERACIÓN DEL SISTEMA

6.1 TIPOS DE PRUEBAS

La meta de probar las aplicaciones Web es ejercitar cada una de las muchas dimensiones de la calidad en la aplicación con la finalidad de encontrar errores o descubrir conflictos que pudieran conducir a fallas de uso, navegabilidad, desempeño, compatibilidad, capacidad y seguridad. Las pruebas también incorporan revisiones que ocurren conforme se diseña la aplicación. [12]

La estrategia de prueba de la aplicación ejercita cada una de las dimensiones de calidad al examinar inicialmente "unidades" de contenido, funcionalidad o navegación. Una vez que las unidades individuales han sido validadas, el enfoque se cambia a pruebas que ejerciten la aplicación como un todo. Ésto se logra derivando muchas pruebas de las perspectivas de los usuarios y se alimentan con la información contenida en los casos de uso. Se desarrolla un plan de prueba de ingeniería web y se identifican los pasos de prueba, los productos de trabajo y los mecanismos para la evaluación de los resultados de prueba, el proceso de prueba abarca diferentes tipos de pruebas, que se mencionan a continuación:

- La prueba del contenido (y las revisiones) se centra en varias categorías de contenido. La finalidad es descubrir errores tanto semánticos como sintácticos que afecten la precisión del contenido o la forma en la que se presenta al usuario final. [12]

- La prueba de la interfaz ejercita los mecanismos de interacción que permiten que un usuario se comunique con la aplicación y valida los aspectos estéticos de la interfaz. El objetivo es descubrir errores que resulten de mecanismos de interacción mal implementados, u omisiones, inconsistencias o ambigüedades en la semántica de la interfaz. [12]

- La prueba de navegación aplica casos de uso, derivados como parte de la actividad de análisis, en el diseño de casos de prueba que ejercitan cada uno de los escenarios de uso frente al diseño de navegación. Los mecanismos de navegación se prueban para garantizar que se identifican y corrigen los errores que impiden el completar un caso de uso. [12]

- La prueba de componentes ejercita las unidades de contenido y funcionales dentro de la aplicación. Cada página web encapsula contenido, vínculos de navegación y elementos de procesamiento que forman una "unidad" dentro de la arquitectura de la aplicación web. Se deben probar dichas unidades. [12]

- La prueba de configuración intenta descubrir los errores o los problemas de compatibilidad específicos de un ambiente particular de cliente servidor. Entonces se llevan a cabo pruebas para descubrir los errores asociados con cada posible configuración. [12]
- La prueba de la seguridad incorpora una serie de pruebas diseñadas para explotar las vulnerabilidades de la aplicación y en su ambiente. La finalidad es encontrar hoyos de seguridad. [12]

6.2 DOCUMENTACIÓN DE PRUEBAS

Prueba de Contenido

Para este tipo de pruebas los mismos programadores y administradores funcionaron como examinadores del sistema, debido a que, al mismo tiempo que se construía, se comprobaba la calidad y veracidad del contenido. Así mismo se cotejaba que al consultar o modificar la base de datos, la información se almacenara en el lugar correcto de acuerdo a la operación realizada.

Un ejemplo de éstas pruebas, es verificar que los datos se muestren correctamente al agregar una solicitud:

Paso 1: Datos personales del Active Directory de acuerdo al usuario que ingresó al sistema. (Ver figura 6.1).

Figura 6.1 Carga de datos mediante Active Directory

Paso 2: Datos del equipo de acuerdo al no. de inventario proporcionado por el usuario. (Ver figura 6.2).

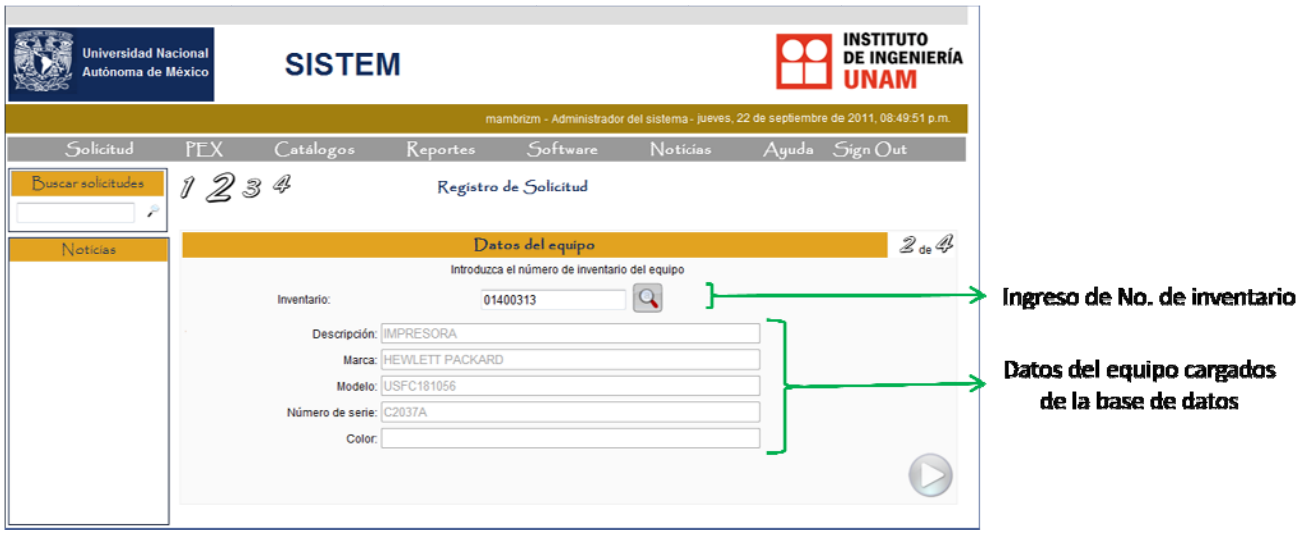

Figura 6.2 Carga de datos por base de datos

Paso 3: Niveles de problemas de acuerdo al servicio requerido. (Ver figura 6.3).

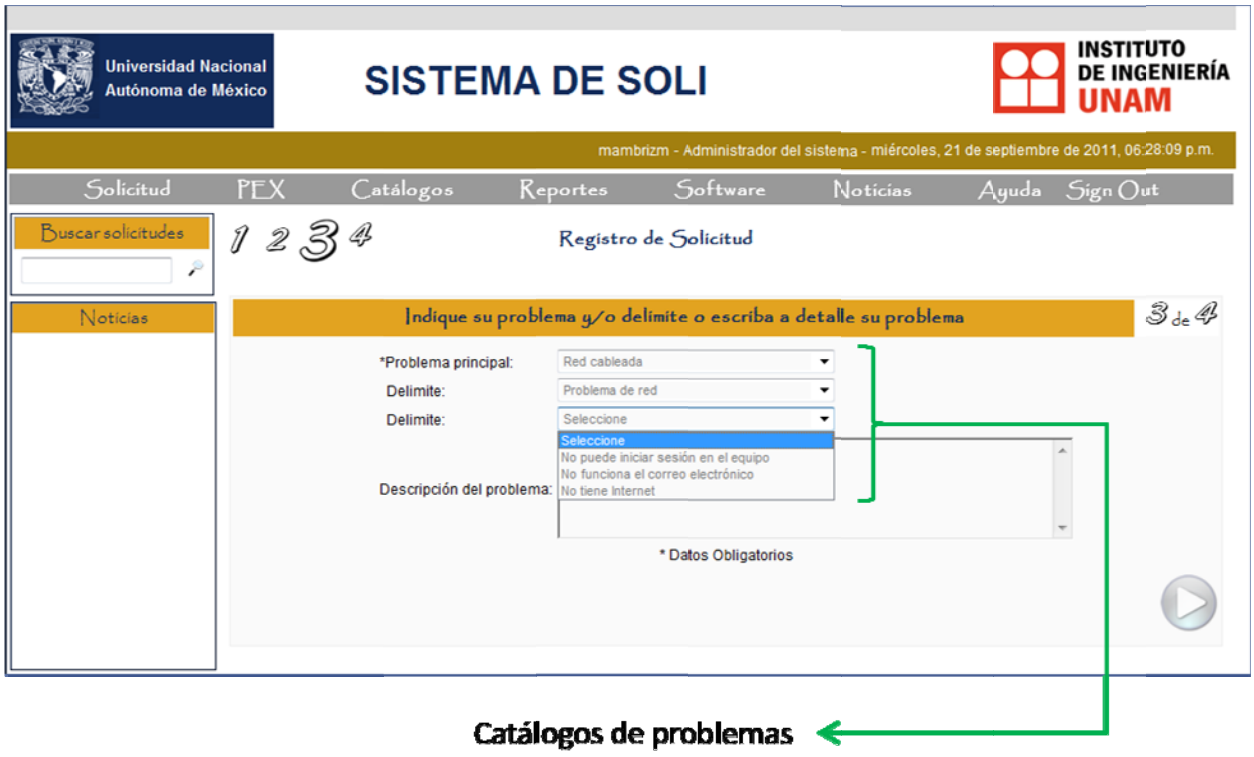

Figura 6.3 Niveles de problemas en registro de solicitud

Paso 4: Resumen de los datos recabados en los pasos anteriores y guardados en la tabla Tbl_Solicitud. Se envía un correo con la información a los jefes de área correspondientes y al usuario. (Ver Figura 6.4)

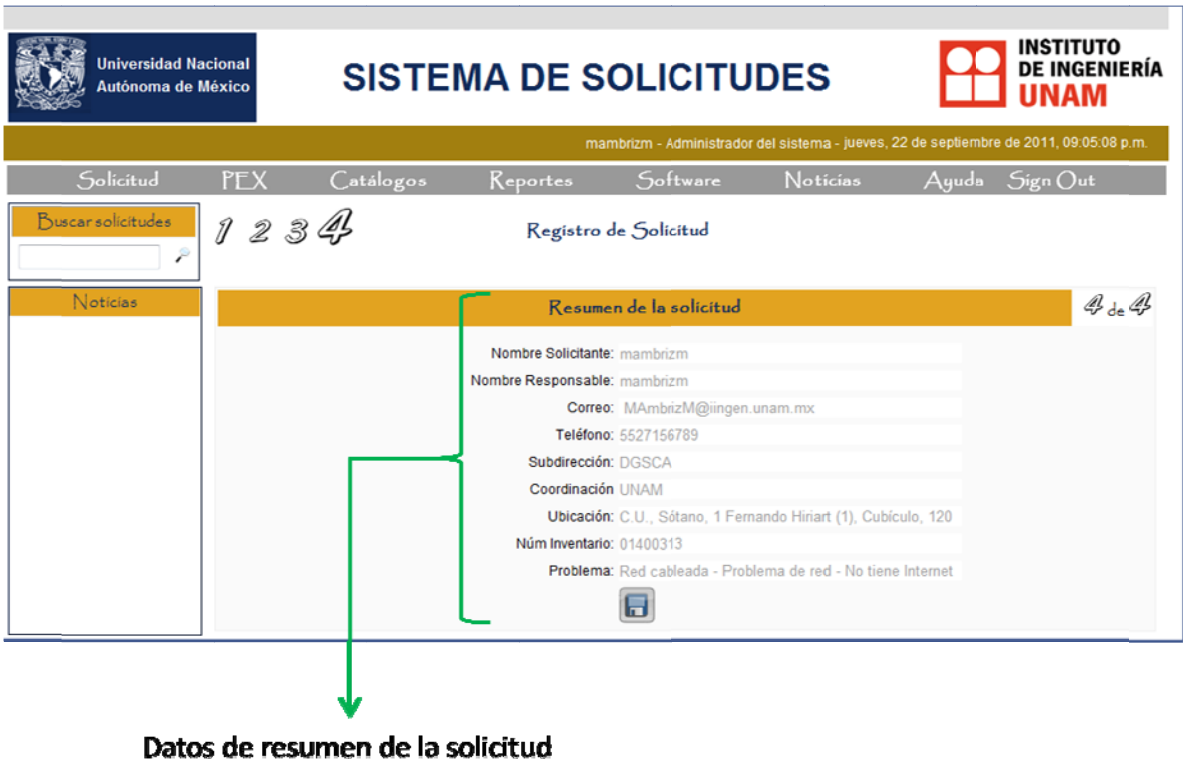

Figura 6.4 Resumen de la solicitud

Al realizar y completar este tipo de pruebas en cada uno de los módulos del sistema, se logró verificar que la información solicitada es concreta y sólo la necesaria, consultando y obteniendo datos personales del Active Directory, lo que reduce la captura de información sin sentido para el usuario.

Prueba de Interfaz

Para esta prueba hay varios factores a considerar, y no solo al final, sino que en tres etapas distintas, del desarrollo: en el análisis y formulación de requerimientos, en el diseño y en las pruebas finales, ésto con el objetivo de llevar un buen control de lo que el usuario observara y conducirá en el sistema. Los factores a considerar son:

Mecanismos de la interfaz: Cuando se interactúa con el sistema se hace mediante uno o más mecanismos de interfaz, los cuales pueden ser: vínculos de navegación, formatos, ventanas pop-up, mapas de sitio, etc. Los cuales se verifican probándolos para asegurarse que se alcance el objeto de contenido o función debida. En el caso de los vínculos redirigir la pagina al lugar correcto, en el tema de los formatos garantizar que el usuario identifique etiquetas, campos obligatorios, valores por defecto y el sistema cumpla con el envío correcto de la información, que las funciones del navegador no corrompan la información, que los menús se muestren completos y que las verificaciones de error de datos sean correctas. Ver figura 6.4

Semántica de la Interfaz: Conforme se prueba cada caso de uso, se revisa la información en pantalla que el actor obtendrá, principalmente de los menús y formularios, que sea entendible y coherente con el actor que hace uso del sistema. Un ejemplo de ello es que el usuario no vea información sólo permitida para los Administradores. Ver figura 6.4

Facilidad de Uso: Es una prueba similar a la de semántica, cuya diferencia radica en que son para determinar el grado de facilidad de uso para el usuario, ya sea que se pruebe por mecanismos, por páginas completas o por el sistema entero. Se pueden dividir las pruebas en categorías y establecer objetivos específicos como: interactividad (mecanismos fáciles de entender y usar), plantillas (mecanismos localizables rápidamente), legibilidad (texto e imágenes comprensibles), estética (colores, tipografía y características relacionadas), personalización (según el tipo de actor: Administrador 1, Administrador 2 ó Usuario, sean las pantallas correspondientes), etc. (Ver Figura 6.4)

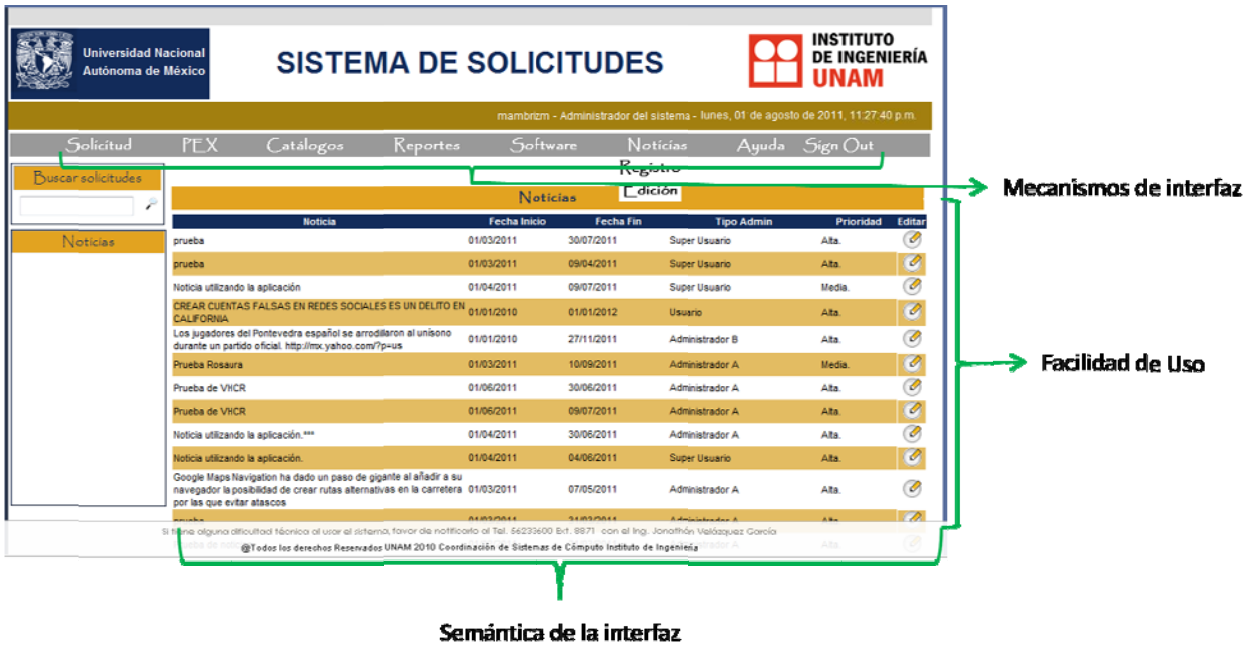

Figura 6.4 Factores a considerar en las pruebas de interfaz

En la formulación y análisis de requerimientos se planteó la manera en que el usuario se desenvolverá en la interfaz, así como los datos que se le pedirán para llenar los formularios de petición de servicios y con lo que podría llevar el seguimiento de su servicio, que se determinó como un numero consecutivo llamado ticket. En esta etapa se realizaron pruebas de mecanismos y semántica de interfaz puesto que se definió la estructura del sistema y, los casos de uso, lo cuales sirvieron para definir los vínculos, formularios, ventanas, etc., de los diferentes actores que intervienen en el sistema, dándonos como resultado una idea clara de cómo diseñar el sistema y los casos de uso. Ver tema 4.2, como ejemplo de casos de uso definidos.

Para las pruebas durante el diseño se formularon los posibles bocetos de las páginas que interactúan con los usuarios así como su formato y seguimiento, esto es que los formularios a rellenar sean congruentes y tengan una secuencia lógica. Ofreciendo la oportunidad de hacer pruebas de mecanismos, en su mayoría, y semántica, pero con una mayor idea sobre el sistema en desarrollo ya que en este punto se tienen diseños más en forma y estructurados. Las pruebas nos corrigieron errores en el diseño físico del sistema, dándonos la oportunidad de re-establecer vínculos y formularios que pudieran ser incongruentes según el actor y el proceso que llevara, por ejemplo, que los

administradores 1 en general pudieran validar personal externo, siendo función solamente del Coordinador de la CSC. De la misma manera se empezó a llevar un control de la semántica de la interfaz para así poder definir correctamente los roles e información correspondiente a cada actor, según los casos de uso. Ver tema 4.3, para ejemplificar el diseño de la interfaz.

En el caso de las pruebas finales se plantearon guiones para los posibles usuarios, tomando como referencia los casos de uso que se diseñaron al principio, ésto contempla errores de lógica en los datos introducidos para verificar las pantallas de alerta y su reacción al error. Por lo que podemos implementar las 3 pruebas más importantes en ésta etapa (mecanismo, semántica de la interfaz y facilidad de uso) en una sola prueba, como por ejemplo el siguiente guión:

Proyecto: Sistema de Solicitud de Servicios para la CSC

CASO DE PRUEBA	Registra solicitud		
CASO DE USO de referencia	Realizar Solicitud		
PROPÓSITO	El usuario captura la información para registrar una solicitud de forma correcta.		
PRE – REQUISITOS	1. Se cuenta con perfil de usuario. 2. El usuario accedió al sistema de solicitudes. 3. El usuario cuenta con un número de inventario del equipo para el que solicita servicio. 4. El usuario seleccionó el menú principal “Solicitud”→”Registrar Solicitud”		
DATOS DE PRUEBA	Usuario de dominio. Número de inventario del equipo.		
PASOS	**Actividad**	**Resultado**	
		✓	X
	1. Paso 1: La pantalla está integrada por los componentes que a continuación se listan:		
	1.1. Casilla para realizar **solicitud a nombre de otra persona**. Aparece deshabilitada. Al habilitar casilla, se muestra lista **Académico responsable.**	✓	
	1.2. Caja de texto donde aparece el **Nombre** del usuario que ingresó al sistema. Aparece inhabilitada.	✓	
	1.3. Caja de texto donde aparece el **Teléfono** del usuario que ingresó al sistema. Aparece inhabilitada.	✓	
	1.4. Caja de texto donde aparece la **Subdirección** del usuario que ingresó al sistema. Aparece inhabilitada.	✓	
	1.5. Caja de texto donde aparece el **Correo** del usuario que ingresó al sistema. Aparece inhabilitada.	✓	
	1.6. Caja de texto para capturar el nombre del **Responsable Alterno** para la solicitud. Aparece habilitada.	✓	
	1.7. Lista **Campus** aparece habilitada y no presenta		

	información seleccionada. **1.8.** Lista **Edificio** aparece habilitada y no presenta información seleccionada. **1.9.** Lista **Piso** aparece habilitada y no presenta información seleccionada. **1.10.** Lista **Cubículo** aparece habilitada y no presenta información seleccionada. **1.11.** Botón **Siguiente**, aparece habilitado. (Ver paso 2)	✓ ✓ ✓ ✓	
	2. Paso 2: La pantalla está integrada por los componentes que a continuación se listan: **2.1.** Caja de texto para capturar el **Número de Inventario** para la solicitud. Aparece habilitada. Ingresar número de inventario existente. **2.2.** Botón **Buscar** para consultar los datos del equipo con el número de inventario proporcionado. **2.3.** Caja de texto donde aparece la **Descripción** del equipo con el número de inventario proporcionado. Aparece inhabilitada. **2.4.** Caja de texto donde aparece la **Marca** del equipo con el número de inventario proporcionado. Aparece inhabilitada. **2.5.** Caja de texto donde aparece el **Modelo** del equipo con el número de inventario proporcionado. Aparece inhabilitada. **2.6.** Caja de texto donde aparece el **Número de Serie** del equipo con el número de inventario proporcionado. Aparece inhabilitada. **2.7.** Caja de texto donde aparece el **Color** del equipo con el número de inventario proporcionado. Aparece inhabilitada. **2.8.** Botón **Siguiente**, aparece habilitado. (Ver paso 3)	 ✓ ✓ ✓ ✓ ✓ ✓ ✓	
	3. Paso 3: La pantalla está integrada por los componentes que a continuación se listan: **3.1.** Lista **Problema Principal** aparece habilitada y no presenta información seleccionada. Seleccionar para	 ✓	

	mostrar la siguiente lista.		
	3.2. Lista **Delimite** aparece habilitada y no presenta información seleccionada. Seleccionar para mostrar la siguiente lista.	✓	
	3.3. Lista **Delimite** aparece habilitada y no presenta información seleccionada.	✓	
	3.4. Caja de texto para capturar la **Descripción del problema** para la solicitud. Aparece habilitada.	✓	
	3.5. Botón **Siguiente**, aparece habilitado. (Ver paso 4)	✓	
	4. Paso 4 "Resumen de la Solicitud": La pantalla está integrada por los componentes que a continuación se listan:		
	4.1. Caja de texto donde aparece **Nombre de Solicitante**. Aparece inhabilitada.	✓	
	4.2. Caja de texto donde aparece **Nombre de Responsable**. Aparece inhabilitada.	✓	
	4.3. Caja de texto donde aparece **Correo**. Aparece inhabilitada.	✓	
	4.4. Caja de texto donde aparece **Teléfono**. Aparece inhabilitada.	✓	
	4.5. Caja de texto donde aparece **Subdirección**. Aparece inhabilitada.	✓	
	4.6. Caja de texto donde aparece **Coordinación**. Aparece inhabilitada.	✓	
	4.7. Caja de texto donde aparece **Ubicación**. Aparece inhabilitada.	✓	
	4.8. Caja de texto donde aparece **Número de Inventario**. Aparece inhabilitada.	✓	
	4.9. Caja de texto donde aparece **Problema**. Aparece inhabilitada.	✓	
	4.10. Botón **Guardar**, aparece habilitado.	✓	
	4.11. Se guarda la solicitud y muestra mensaje de solicitud guardada.		X
	4.12. Botón **Word**, aparece habilitado. Para consultar el resumen de la solicitud en archivo de tipo Word		X
NOTAS Y PREGUNTAS			

Prueba de Navegación

En las pruebas de navegación al igual que la mayoría de las demás pruebas se pueden realizar conforme se desarrolla el sistema, en este caso las pruebas se le realizan a la congruencia y facilidad de manejo del sistema, es decir, una secuencia lógica y amigable para los usuarios durante el uso del sistema.

La primera fase de la prueba de navegación comienza durante la prueba de interfaz. los mecanismo de navegación se prueban para asegurar que cada uno realiza la función que se busca, como ya se vio en la prueba de interfaz.

La prueba de la semántica de navegación se realiza para probar los conjuntos de estructuras de información y navegación relacionadas que colaboran en el cumplimiento de los requisitos de usuario, se pueden realizar mediante los casos de uso, esto es, con guiones formados de los casos de uso.

Para probar el sistema, se determinaron guiones para posibles usuarios y diferentes problemáticas se implementaron dichos guiones y se verifico que tan coherente era el desarrollo de una solicitud, en el caso de un usuario y una asignación de servicio en el caso de un administrador, en ambos casos mostrando problemas de navegación y re direccionamiento, para los links, y otros componentes webs. Se puede ejemplificar la prueba mediante el guión utilizado en el punto anterior.

Prueba de Configuración

La labor de probar la configuración no es ejercitar toda posible configuración del lado del cliente. Más bien, es probar un conjunto de probables configuraciones de los lados del cliente y del servidor para garantizar que la experiencia del usuario será la misma en todos ellos y para aislar errores que puedan ser especificados de una configuración particular.

Esencialmente, la aplicación se instaló en el ambiente del servidor y se probó con la intención de encontrar errores relacionados con la configuración. Por lo que podemos decir que se probaron servidores de Aplicaciones y de base de datos, dándonos como resultado:

- Compatibilidad de la aplicación con el sistema operativo del servidor.
- Creación correcta de los archivos de sistema, directorios y datos de sistema relacionados cuando la aplicación es operativa.
- Las medidas de seguridad como cortafuegos, permiten la ejecución del sistema sin interferencia ni bajo rendimiento.
- El sistema se integró correcta y completamente con la base de datos.

Cabe resaltar que los servidores son manejados y administrados por el área de servidores Windows de la CSC, por lo que constantemente son sometidos a pruebas de configuración, desempeño y errores, lo que nos asegura el correcto funcionamiento de nuestro sistema del lado del servidor.

En el lado del cliente las pruebas de configuración se centran principalmente en la compatibilidad de la aplicación con las configuraciones que contienen los siguientes elementos:

- Hardware: CPU, memoria, almacenamiento y dispositivos de impresión. (Ver tema 3.2.3)
- Sistemas operativos y software de navegación. (Ver tema 3.2.3 y 4.3)
- Conectividad. Red proporcionada por el II.

Pruebas de Seguridad

Las pruebas de seguridad están diseñadas para probar las vulnerabilidades en el ambiente del lado del cliente, las comunicaciones de red y el ambiente del lado del servidor

Nos corresponden las del lado del cliente y parte de las del servidor, debido a que el servidor es administrado por el área de Servidores Windows de la CSC. En el lado del cliente, las vulnerabilidades con frecuencia provienen de errores preexistentes en los navegadores, programas de correo electrónico o software de comunicación. Por ejemplo desbordamiento de buffer, acceso no autorizado a cookies, simulación de páginas web.

En el lado del servidor, las vulnerabilidades incluyen ataques de negación de servicio y guiones maliciosos que pueden pasar al lado del cliente o empleados para deshabilitar las operaciones del servidor y acceso, no autorizado, a la base de datos.

La protección contra estas vulnerabilidades requiere implementar los siguientes elementos de seguridad.

- Cortafuegos: Mecanismo de filtrado que examina cada paquete de información entrante para garantizar que llega de una fuente legítima y bloquea cualquier dato sospechoso (Firewall Administrado por el área de Servidores Windows de la SCS).

- Autentificación: Mecanismo de verificación que valida la identidad de todos los clientes y servidores, y permite que la comunicación ocurra solo cuando ambos lados son verificados. La autentificación se realiza mediante el Active Directory y solo permite el acceso a miembros del II con cuenta y contraseña validadas por el área de Servidores Windows.

- Detección de intrusos: Es el seguimiento y la notificación de una actividad no autorizada en un equipo o red monitorizados. Implica funcionalidades como: alertas, registro, elaboración de informes y prevención.

La finalidad de esta prueba es exponer hoyos en dichos elementos de seguridad que podrían ser explotadas por aquellos que tengan intenciones maliciosas. En muchos casos, las pruebas de seguridad se subcontratan con firmas que se especializan en dichas tecnologías. En el caso del II, la misma CSC se ha visto en la necesidad de adquirir tecnologías y desarrollar políticas de seguridad, administradas por algunas áreas de la coordinación; un ejemplo de ello es el área de servidores Windows que, como se ha mencionado, mantiene mucha de la seguridad en lo que a cómputo y redes se refieren.

6.3 NAVEGADORES MÁS UTILIZADOS

Los tres navegadores más utilizados en la actualidad son: Internet Explorer (Microsoft), Firefox (Mozilla) y Chrome (Google).

Conforme avanza la tecnología, las aplicaciones web son cada vez más complejas, por lo que se ha buscado reducir su tiempo de desarrollo y a su vez, tener la seguridad de que las aplicaciones funcionan igual de bien en los navegadores más populares. Es por esta razón que han surgido algunas librerías y frameworks específicos para el desarrollo de aplicaciones con JavaScript, y un ejemplo de ésto es el Framework .Net 3.5 de Microsoft.

Microsoft .NET 3.5 incluye la biblioteca ASP.NET AJAX para desarrollar aplicaciones web más eficientes, interactivas y altamente personalizadas que funcionen para la mayoría de los navegadores y utilicen las últimas tecnologías y herramientas Web, incluyendo Silverlight y Popfly. [22]

A continuación presentamos una lista de los navegadores que mantienen una compatibilidad con la tecnología AJAX, debe tenerse en cuenta que su soporte dependerá de las características que el navegador permita.

- Microsoft Internet Explorer para Windows versión 6.0 ó superiores.
- Navegadores basados en Gecko como Mozilla Firefox versión 1.5 ó superiores.
- Navegadores basados en WebKit como Google Chrome y Apple Safari versiones 2.0 ó superiores.
- Navegadores basados en Presto como Opera versión 9.0 ó superiores.

En el desarrollo del Sistema de Control de Solicitudes se utilizó la biblioteca de AJAX incluida en el Framework 3.5 de Visual Studio 2008, por lo que el sistema fue probado y diseñado para su uso en el navegador más utilizado, Windows Internet Explorer, y que recomendamos para la correcta visualización del sistema. En el caso de los otros dos navegadores más populares se recomienda verificar su funcionamiento y de ser necesario adecuar el sistema conforme a las necesidades de los usuarios.

6.4 LIBERACIÓN

Una aplicación Web está compuesta por un conjunto de archivos, páginas, módulos y código ejecutable, que se invocan o ejecutan dentro del ámbito de un directorio virtual (y sus subdirectorios) en el servidor Web de aplicaciones IIS.

Para poner en marcha el sistema debemos seguir las políticas del II, las cuales incluyen, la coordinación y apoyo con el área de servidores Windows, que como se detalló en el capítulo es la encargada de administrar servidores de bases de datos y aplicaciones desarrolladas para y por el II, pasos que se describen a continuación.

1.- Base de datos

Para cargar la base de datos “CMP_SISOL”, se creó un script con la información necesaria para generar la base de datos que incluye tablas, vistas, procedimientos almacenados, funciones, etc.; así como creación de usuarios y sus correspondientes permisos. Para ejemplificar este paso, la imagen siguiente muestra el script ejecutado, en el entorno SQL Server 2008 del servidor de pruebas.

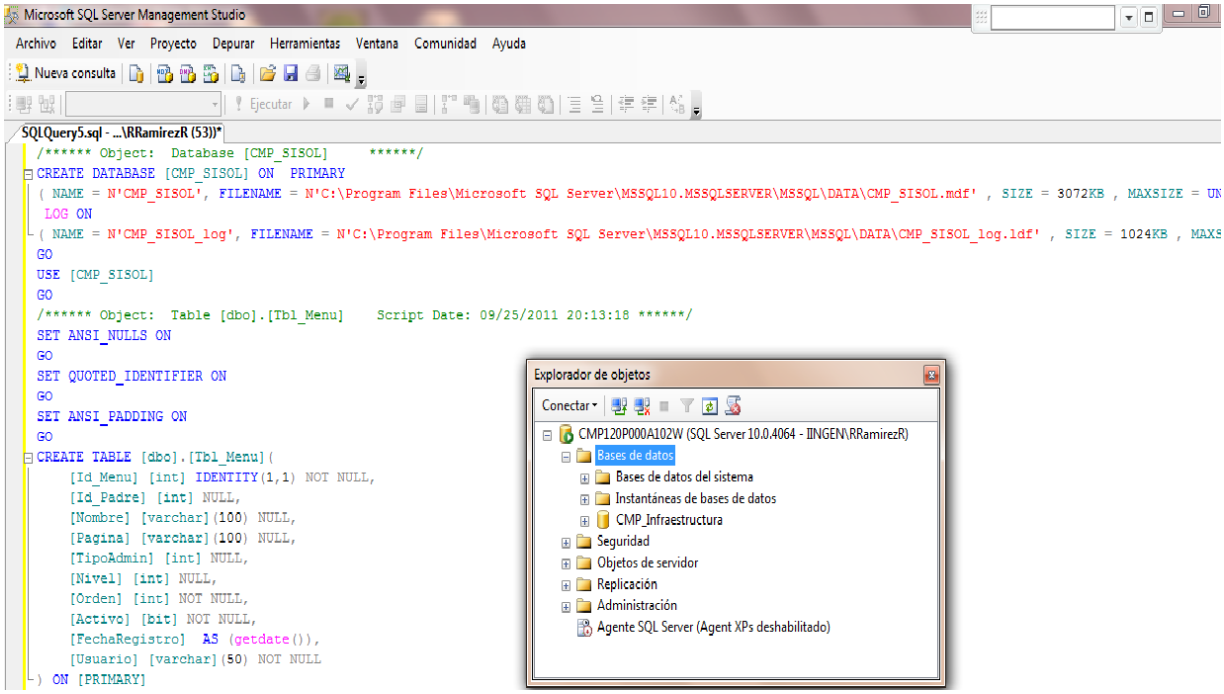

Figura 6.5 Script para la creación de la base de datos

2.- Probar conexión

La cuenta de usuarios necesaria para el acceso a la base de datos es: II_CMP_SISOL, la cual es creada y especificada en el script del paso 1. Para probar la conexión debemos conectarnos al motor de base de datos indicando tipo de autenticación de SQL Server, cuenta y contraseña de usuario II_CMP_SISOL para inicio de sesión. Ver figura 6.6 para ejemplificar este punto.

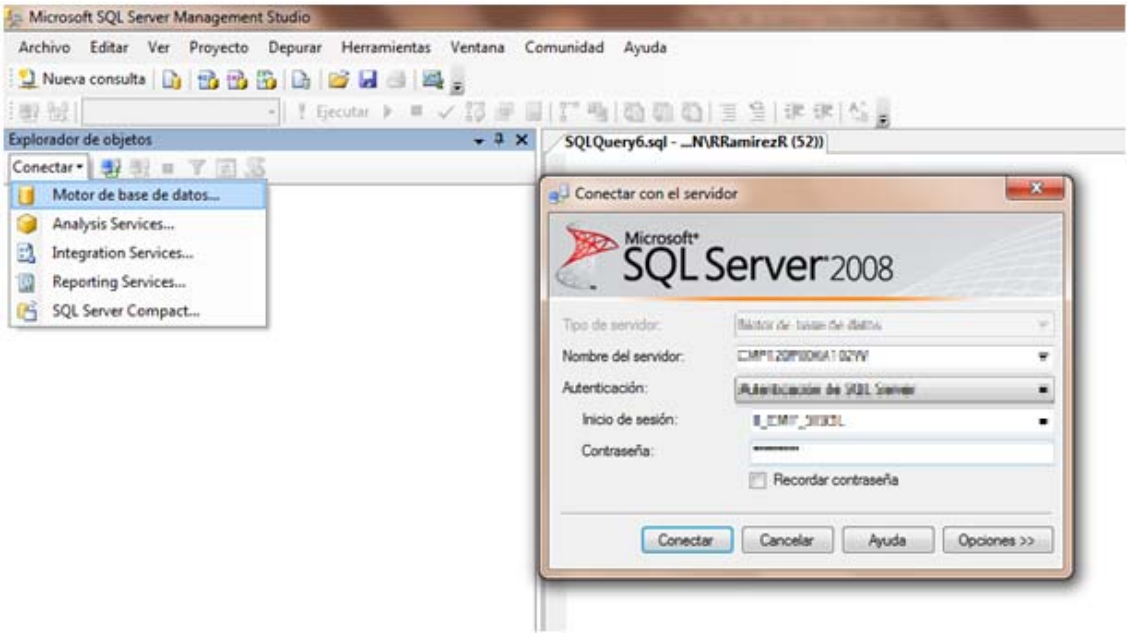

Figura 6.6 Acceso y autenticación a la base de datos

La conexión exitosa se confirma con una pantalla similar a la figura 6.7.

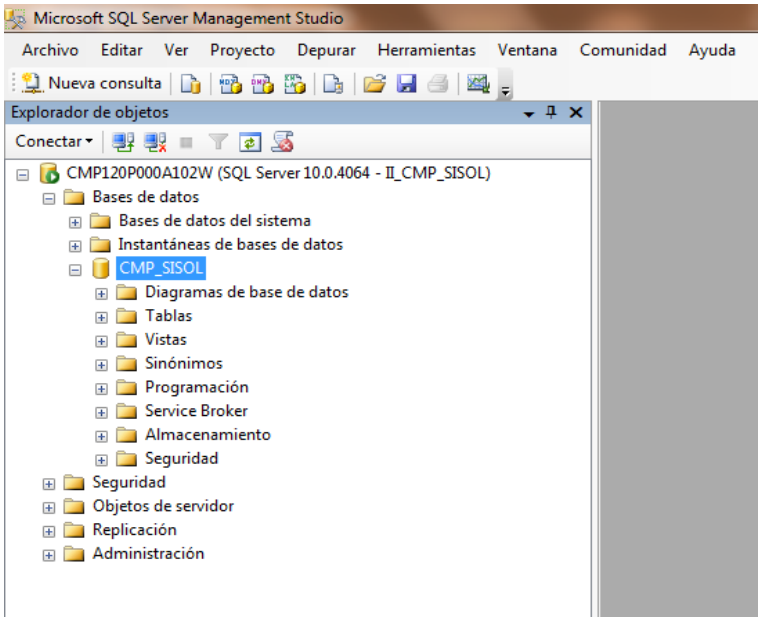

Figura 6.7 Prueba de conexión a la base de datos

3.- Organización de aplicación en servidor de producción

La organización de carpetas y permisos para nuestra aplicación, se detalla en el documento de especificaciones requerido por el área de Windows para su adecuada creación y administración del sitio virtual y grupos de control de usuarios. Dicho documento se puede consultar en el anexo B.

4.- Mantenimiento

Una vez realizados los puntos anteriores, el sistema está liberado y es necesario establecer un plan de mantenimiento de acuerdo con las políticas del II y los integrantes de la CSC responsables de su administración.

Según la terminología ANSI-IEEE, el mantenimiento del software es: "la modificación de un producto software después de su entrega al cliente o usuario para corregir defectos, para mejorar el rendimiento u otras propiedades deseables, o para adaptarlo a un cambio de entorno". [30]

Tipos de Mantenimiento

En la definición de mantenimiento aparecen indicados, directa o indirectamente, cuatro tipos de mantenimiento:

- Corregir defectos → Mantenimiento correctivo
- Mejorar el rendimiento → Mantenimiento preventivo/perfectivo
- Adaptar a un cambio de entorno → Mantenimiento adaptativo

Mantenimiento Correctivo

A pesar de las pruebas y verificaciones que aparecen en etapas anteriores del ciclo de vida del software, los programas pueden tener defectos. El mantenimiento correctivo tiene por objetivo localizar y eliminar los posibles defectos de los programas.

Un defecto es una característica del sistema con el potencial de causar un fallo. Un fallo ocurre cuando el comportamiento de un sistema es diferente del establecido en las especificaciones. Entre otros, los fallos en el software pueden ser de:

- Procesamiento, por ejemplo, salidas incorrectas del sistema.
- Rendimiento, por ejemplo, tiempo de respuesta demasiado alto en una búsqueda de información.
- Programación, por ejemplo, inconsistencias en el diseño del sistema.
- Documentación, por ejemplo, inconsistencias entre la funcionalidad del sistema y el manual de usuario.

Mantenimiento Adaptativo

Este tipo de mantenimiento consiste en la modificación del sistema debido a cambios en el entorno (hardware o software) en el cual se ejecuta.

La envergadura del cambio necesario puede ser muy diferente: desde un pequeño retoque en la estructura de un módulo hasta tener que reescribir prácticamente toda la programación del sistema.

Los cambios pueden afectar a:

- El sistema operativo (cambio a una versión actual).
- El entorno de desarrollo del software (incorporación de nuevos elementos o herramientas), los cambios pueden ser de dos clases: en el entorno de los datos y de los procesos.

Este tipo de mantenimiento es cada vez más frecuente debido principalmente al cambio, cada vez más rápido, en los diversos aspectos de la informática: nuevas generaciones de hardware, nuevos sistemas operativos, y mejoras en los periféricos o en otros elementos del sistema (frente a ésto, la vida útil de un sistema de software puede superar fácilmente los diez años).

Mantenimiento Perfectivo

Cambios en la especificación, normalmente debidos a cambios en los requerimientos de un producto de software, implican un nuevo tipo de mantenimiento llamado perfectivo, estos casos son muy variados. Desde algo tan simple como cambiar el formato de impresión de un informe, hasta la incorporación de un nuevo módulo funcional.

Entonces podemos definir el mantenimiento perfectivo como el conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas por el usuario.

Mantenimiento Preventivo

Consiste en la modificación del software para mejorar sus propiedades (por ejemplo, aumentando su calidad y/o su mantenibilidad) sin alterar sus especificaciones funcionales.

Algunas maneras de hacerlo son:

- Incluir sentencias que comprueben la validez de los datos de entrada.
- Reestructurar los programas para mejorar su legibilidad.
- Incluir nuevos comentarios que faciliten la posterior comprensión del programa.

En algunos casos se ha planteado el Mantenimiento para la Reutilización, consistente en modificar el software (buscando y modificando componentes para incluirlos en bibliotecas) para que sea más fácilmente reutilizable. En realidad este tipo de mantenimiento es preventivo, especializado en mejorar la propiedad de reusabilidad del software.

Las actividades de mantenimiento de software se pueden agrupar en tres categorías funcionales:

- Comprensión del software y de los cambios a realizar: para poder modificar un programa, los programadores necesitan conocer su funcionalidad y objetivos, su estructura interna y los requisitos de operación. De no ser así, se corre un gran riesgo de introducir nuevos defectos que en el futuro supondrán un coste de mantenimiento adicional.

- Modificación del software: para incorporar los cambios necesarios se deben crear y modificar las estructuras de datos, la lógica de los procesos, las interfaces y la documentación. Los programadores deben conocer lo mejor posible las repercusiones que tienen en el sistema los cambios que están realizando, con el fin de evitar al máximo posible los efectos secundarios.

- Realización de pruebas: para validar los cambios se deben realizar pruebas selectivas que nos permitan comprobar la corrección del software. Esta actividad es necesaria siempre, ya que incluso un cambio muy pequeño no verificado puede producir defectos en el software que reduzcan su calidad y fiabilidad.

Una vez definidos los puntos anteriores, recomendamos establecer un programa de mantenimiento correctivo, adaptativo, preventivo y perfectivo, de acuerdo al siguiente plan:

- Correctivo: A pesar de haber tenido un plan de pruebas y corrección de errores, el sistema podría presentar alguna falla. Por lo que los desarrolladores deberán resolver las incidencias lo más pronto posible durante un periodo de 3 meses a partir de su liberación.

- Adaptativo: El sistema fue probado en diferentes equipos, la mayoría con el sistema operativo Windows Vista (una de las tecnologías más actuales) y navegador Internet Explorer 8.0. Se recomienda evaluar el aprovechamiento para nuevos sistemas operativos y versiones posteriores de Internet Explorer, en caso de ser necesario realizar su correspondiente mantenimiento adaptativo.

- Perfectivo: Se realizará cuando el sistema sea conocido a la perfección por los usuarios y llegue a la madurez correspondiente para poder realizar cambios que hagan mucho más eficiente al sistema. Es recomendable que el sistema sea usado aproximadamente por 1 año para que los usuarios se familiaricen y puedan aportar mejoras o peticiones de integración con tecnología móvil u otras.

- Preventivo: Durante el año de madurez propuesto, se recomienda realizar el monitoreo de la actividad en el sistema para poder identificar, analizar y llevar a cabo las modificaciones necesarias, reestructuraciones de código y diseño del sistema.