

Pedidos Online - DUSA

Plan de Verificación y Validación

Versión 1.3

Historia de revisiones

Fecha	Versión	Descripción	Autor
25/08/2013	1.0	Comienzo del Plan	Patricia Rolandi
26/08/2013	1.1	Actualización del documento	Patricia Rolandi
29/08/2013	1.2	Nueva actualización a partir de reuniones con cliente	Patricia Rolandi
29/08/2013	1.3	Revisión final Responsable SQA	Verónica Gamarra

Índice

1. Introducción	4
1.1 Propósito	4
1.2 Punto de partida	4
1.3 Alcance	5
1.4 Identificación del proyecto.....	7
1.5 Estrategia de evolución del Plan.....	7
2. Requerimientos para verificar	8
3. Estrategia de Verificación.....	11
3.1 Tipos de pruebas	11
3.1.1 Prueba de integridad de los datos y la base de datos	11
3.1.2 Prueba de Funcionalidad.....	11
3.1.3 Prueba de Ciclo del Negocio	12
3.1.4 Prueba de Interfaz de Usuario	12
3.1.5 Prueba de Performance.....	13
3.1.6. Prueba de Carga.....	14
3.1.7. Prueba de Esfuerzo (stress, competencia por recursos, bajos recursos).....	14
3.1.8. Prueba de Volumen	15
3.1.9. Prueba de Seguridad y Control de Acceso	15
3.1.10. Prueba de Fallas y Recuperación.....	16
3.1.11. Prueba de Configuración.....	17
3.1.12. Prueba de Instalación	17
3.1.13. Prueba de Documentos	18
3.2 Herramientas.....	18
4. Recursos.....	19
4.1 Roles.....	19
4.2 Sistema.....	19
5. Hitos del proyecto de Verificación	20
6. Entregables.....	21
6.1 Modelo de Casos de Prueba	21
6.2 Informes de Verificación	21
6.3 Evaluación de la verificación	22
6.4 Informe final de verificación	23
7. Dependencias.....	23
7.1. Dependencia de personal.....	23
7.2. Dependencia de software	23
7.3. Dependencia de hardware.....	24
7.4. Dependencia de datos y base de datos de prueba.....	24
8. Riesgos	24
8.1. Planificación.....	24
8.2. Técnico	24
8.3. Gestión.....	24
9. Apéndice	24

9.1. Niveles de gravedad de error24

9.2. Niveles de aceptación para los elementos verificados25

1. Introducción

1.1 Propósito

Este Plan de Verificación para el proyecto Pedidos Online – DUSA, soporta los siguientes objetivos:

- Construir una aplicación web que permita a las farmacias realizar pedidos en forma ágil y sencilla, como una alternativa al sistema de llamado telefónico utilizado actualmente. La aplicación deberá funcionar de manera análoga a cómo funciona actualmente el servicio de Call center de DUSA, por medio del cual los farmacéuticos realizan pedidos diariamente.
Además existe un software en la empresa que está automatizado para realizar el despacho del pedido. Por lo que la idea es que a partir de nuestra aplicación web, se pueda realizar un pedido y para esto se deberá verificar la correcta comunicación entre nuestra aplicación y la que DUSA tiene actualmente, así como también su base de datos sobre clientes y stock de medicamentos y perfumería, entre otros.
- Para la realización de la verificación se constará de 3 etapas:
 - Las pruebas unitarias las ejecutarán los propios implementadores utilizando la técnica de caja blanca mediante la herramienta JUNIT.
 - Las pruebas de integración serán dirigidas por el responsable de las mismas utilizando el software que él considere apropiado.
 - Las pruebas del sistema las realizarán todos los Asistentes de Verificación los cuales van a ejecutar las pruebas creadas por el Responsable de Verificación, esto también se hará con la utilización de las herramientas apropiadas, se utilizará una planilla Excel para llevar el registro de la gestión de pruebas unitarias realizadas.
- Para la tarea de verificación se contará con los siguientes participantes:
 - Responsable de Verificación: Patricia Rolandi
 - Asistentes de Verificación:
 - Laura Font
 - Verónica Gamarra
 - Joana Miller
 - Juan Miguel Álvarez
 - Los implementadores del proyecto que realizarán pruebas unitarias.
- La documentación que se tiene que entregar en lo que respecta a verificación son los siguientes:
 - Plan de Verificación y Validación
 - Plan de Verificación de la Iteración
 - Modelo de Casos de Prueba
 - Informe de Verificación de Documento
 - Informe de Verificación de Integración
 - Informe de Verificación de Sistema
 - Documento de evaluación y ajuste al plan de VyV
 - Evaluación de Verificación
 - Informe Final de Verificación

1.2 Punto de partida

El objetivo principal de la verificación y validación es intentar detectar la mayor parte de los defectos para que luego puedan ser corregidos, antes de la liberación del producto de software.

El destino de la verificación del proyecto es el correcto funcionamiento del producto a construir en base a lo que se acuerde con el cliente. Se deben verificar que los

requerimientos funcionales y no funcionales que sean especificados por el cliente se cumplan para lograr así su satisfacción.

Se propone como objetivo, por medio de las pruebas, entregar al cliente un sistema robusto dentro de lo posible, de manera de minimizar la probabilidad de manifestaciones de fallas.

1.3 Alcance

En esta sección se describen los diferentes estados de la verificación:

- Pruebas Unitarias.
- Pruebas de Integración.
- Pruebas del Sistema: Pruebas funcionales, Pruebas de desempeño, Pruebas de aceptación.

Tipo de Prueba	Pruebas Unitarias
Responsable de su ejecución	Estas pruebas las realizarán los implementadores, siendo preferible (aunque no obligatorio) que para cada módulo a probar, las pruebas las realice la persona que lo desarrollo, pues posee un conocimiento más detallado de su funcionamiento.
Descripción	Se verificarán las funciones de cada módulo, buscando descubrir diferencias entre los requerimientos y su funcionamiento; priorizando los módulos que se consideren más críticos.
Salida	Se generarán informes de verificación donde se registran los defectos encontrados, detallando los que se solucionaron y los que no se lograron solucionar (incidentes conocidos que se solucionarán para la siguiente fase).
Consideraciones	Si el implementador lo cree necesario puede contar con la ayuda de alguno de los integrantes del equipo de verificación para realizar las pruebas y/o el informe correspondiente.
Ambiente	El ambiente de desarrollo será donde se ejecuten las pruebas, pudiendo utilizar herramientas como JUNIT.

Tipo de Prueba	Pruebas de Integración
Responsable de su ejecución	Serán realizadas por el equipo de desarrollo, pues para este tipo de pruebas se necesita tener conocimiento de las interfaces y funciones.
Descripción	Se evaluará que los componentes trabajan juntos como un sistema integrado.
Salida	Se generarán informes de verificación donde se registran los defectos encontrados, detallando los que se solucionaron y los que no se lograron solucionar (incidentes conocidos que se solucionarán para la siguiente fase).
Consideraciones	Si el implementador lo cree necesario puede contar con la ayuda de alguno de los integrantes del equipo de verificación para realizar las pruebas y/o el informe correspondiente.
Ambiente	El ambiente de desarrollo será donde se ejecuten las pruebas.

Tipo de Prueba	Pruebas de Sistema
Tipo de Prueba concreta	Pruebas funcionales
Responsable de su ejecución	Las pruebas del sistema serán realizadas por el equipo de verificadores (responsable y asistentes de verificación).
Descripción	Tienen como finalidad, verificar que el sistema integrado cumple con los requerimientos definidos. Las pruebas de sistema que se realizarán son funcionales.
Salida	Se generarán informes de verificación donde se registran los defectos encontrados, detallando los que se solucionaron y los que no se lograron solucionar (incidentes conocidos que se solucionarán para la siguiente fase).
Consideraciones	Las pruebas se realizarán en base a los casos de uso, identificando los diferentes escenarios y llegando a encontrar las distintas condiciones para la ejecución de las pruebas. Se tendrá en cuenta priorizar aquellos casos que se consideren críticos sobre los que no lo sean.
Ambiente	El ambiente de verificación brindado por DUSA, será donde se ejecutarán dichas pruebas, utilizando las herramientas que el equipo de verificación considere necesarias.

Tipo de Prueba	Pruebas de Sistema
Tipo de Prueba concreta	Pruebas de desempeño
Responsable de su ejecución	Las pruebas del sistema serán realizadas por el equipo de verificadores (responsable y asistentes de verificación).
Descripción	Tienen como finalidad, verificar que el sistema integrado cumple con los requerimientos definidos. Las pruebas de sistema que se realizarán son de desempeño, verificando que se cumplan los requerimientos no funcionales.
Salida	Se generarán informes de verificación donde se registran los defectos encontrados, detallando los que se solucionaron y los que no se lograron solucionar (incidentes conocidos que se solucionarán para la siguiente fase).
Consideraciones	Se deberá definir junto con el cliente que tipo de pruebas desea que se realicen (carga, volumen, stress, etc.)
Ambiente	El ambiente de verificación será donde se ejecutarán dichas pruebas, utilizando las herramientas que determine el equipo de verificación.

Tipo de Prueba	Pruebas de Sistema
Tipo de Prueba concreta	Pruebas de aceptación
Responsable de su ejecución	Las pruebas del sistema serán realizadas por el equipo de verificadores (responsable y asistentes de verificación).
Descripción	Tienen como objetivo, verificar que el sistema integrado cumple con los requerimientos definidos por el cliente y con estas pruebas se valida el sistema construido.
Salida	Se generarán informes de verificación donde se registran los defectos encontrados, detallando los que se solucionaron y los que no se lograron solucionar (incidentes conocidos que se solucionarán para la siguiente fase).
Consideraciones	Se deberá definir junto con el cliente que tipo de pruebas son de su interés, para lograr la validación correspondiente.
Ambiente	El ambiente de verificación será donde se ejecutarán dichas pruebas, utilizando las herramientas que el equipo de verificación considere necesarias.

Debido a que el tiempo para la realización del proyecto y la consiguiente creación del producto, está limitado a 14 semanas, cualquier desviación en el calendario impactará en la verificación, pues si la construcción del sistema toma más tiempo del esperado, menor será el tiempo disponible para la verificación.

Además como el proyecto será iterativo e incremental, el plan irá sufriendo algunas modificaciones a lo largo de cada iteración.

1.4 Identificación del proyecto

Los documentos usados para elaborar el Plan de Verificación fueron los siguientes:

- Especificación de Requerimientos
- Descripción arquitectura
- Casos de Uso
- Glosario
- Modelo de dominio
- Documentación que surge a partir de las reuniones con el cliente (actas de requerimientos y otros).
- Material sobre el plan de V y V de otros años, que se encuentra en la Memoria Organizacional.

1.5 Estrategia de evolución del Plan

Especificación de la estrategia para realizar cambios agendados y no agendados al Plan de Verificación y Validación:

- El encargado de monitorear el Plan de V y V es el Responsable de Verificación. Deberá planificar la verificación que se hará a lo largo del proyecto, evaluando y ajustando el plan mencionado.
- El plan se revisará periódicamente y será modificado en caso de ser necesario, sobre todo en las primeras semanas y a medida que vayan cambiando los otros documentos de los cuales depende el mismo.
- El plan será ajustado en cada iteración por todo el equipo de Verificación, cuando sea necesario un cambio, este será evaluado y aprobado por todos. En el caso

- de que sea un cambio menor éste podrá ser aprobado por el Responsable de Verificación o por un asistente designado por él.
- Cualquier cambio será informado al resto del equipo de Verificación mediante los medios de comunicación definidos.
- La última versión del mismo siempre se encontrará en el repositorio del grupo, a disposición de todo el equipo del proyecto para ser consultado cuando sea necesario.

2. Requerimientos para verificar

En la lista a continuación se presentan los elementos, casos de uso, requerimientos funcionales y requerimientos no funcionales, que serán verificados:

- **Requerimientos Funcionales:**

Requerimientos identificados referidos a los Pedidos:

ID	Nombre	Descripción
R00	Iniciar Pedido	El sistema permitirá al Usuario-Web comenzar un pedido a través del sitio web.
R01	Modificar Pedido	El sistema permitirá al Usuario-Web modificar la lista de productos, el reparto asignado, fecha de entrega y condiciones de compra de un pedido activo.
R02	Almacenar Pedido	El sistema permitirá al Usuario-Web guardar un pedido para finalizarlo en una sesión posterior.
R03	Retomar Pedido	El sistema permitirá al Usuario-Web reactivar un pedido guardado para su finalización/ cancelación.
R04	Cancelar Pedido	El sistema permitirá al Usuario-Web cancelar un pedido activo.
R05	Confirmar Pedido	El sistema permitirá al Usuario-Web confirmar un pedido activo.

Requerimientos identificados referidos a los Productos:

ID	Nombre	Descripción
R06	Quitar Producto a Pedido	El sistema permitirá al Usuario-Web quitar productos de la lista de productos asociada a pedidos activos.
R07	Agregar Producto a Pedido	El sistema permitirá seleccionar una cantidad determinada de un producto a ser agregado en un pedido activo.
R08	Consultar detalle Producto	El sistema permitirá ver los detalles de un producto, como por ejemplo: Laboratorio, nombre, descripción, precio y ofertas/bonificaciones relacionadas al mismo.
R09	Buscar Producto	El sistema brindará la posibilidad de buscar un producto de acuerdo a distintos criterios aún no establecidos.

R10	Consultar Disponibilidad	Consulta la disponibilidad en stock de un producto dado.
-----	--------------------------	--

Requerimientos identificados referidos a los Usuarios:

ID	Nombre	Descripción
R11	Iniciar sesión	El sistema permitirá al usuario ingresar al sistema mediante el uso de un Usuario-Web. El mismo está definido por nombre de usuario y contraseña.
R12	Finalizar sesión	El sistema permitirá al Usuario-Web finalizar la sesión.
R13	Consultar detalle Cuenta	Consulta detalle de condiciones de compra que aplican a su cuenta, cuentas corrientes, repartos asignados, historial de pedidos, estado de pedidos, límite de crédito y saldo disponible.
R14	Modificar contraseña	El sistema permitirá al Usuario-Web modificar la contraseña con la cual debe ingresar al sitio.
R15	Notificaciones	El sistema notificará al Usuario-Web de promociones existentes acerca de los productos, así como también mensajes privados asociados a su Usuario-Web, generados por el Administrador del sitio.
R16	Enviar catálogo	El sistema permitirá enviar el catálogo de productos de interés para el Usuario-Web. Se definirá el medio de envío en una próxima instancia.
R17	Agendar Pedido	El sistema permitirá al Usuario-Web agendar pedidos activos, siempre que no supere más de una semana respecto a la fecha actual.

Requerimientos identificados referidos al administrador del sitio:

ID	Nombre	Descripción
R19	Iniciar sesión	El sistema permitirá al usuario administrador ingresar al sistema mediante el ingreso de nombre de usuario y contraseña.
R20	Finalizar sesión	El sistema permitirá al administrador finalizar la sesión.
R21	Deshabilitar Usuarios-Web	El sistema permitirá al administrador bloquear un

		Usuario-Web de forma tal que no pueda acceder a la aplicación web para realizar pedidos.
R22	Habilitar Usuarios-Web	El administrador podrá desbloquear a un Usuario-Web (bloqueado) de modo que pueda acceder nuevamente a la aplicación web para realizar pedidos.
R23	Resetear contraseña de Usuario Web	El sistema permitirá al administrador del sitio modificar la contraseña de un Usuario-Web. Será D.U.S.A. quien se encargará de notificar dicho cambio.
R24	Configuración de Time-Out de sesión.	El sistema permitirá al administrador configurar el tiempo máximo en el que una sesión de un Usuario-Web puede estar inactiva. Una vez se supere el mismo, se finalizará la sesión automáticamente y en caso de que exista un pedido activo en ese momento se guardará el estado del mismo para que pueda ser retomado por el usuario en una sesión posterior.
R25	Configuración de notificaciones.	El sistema brindará la posibilidad de que el administrador configure determinadas notificaciones que desea que el usuario vea la próxima vez que ingrese al sitio, como por ejemplo: ofertas del día o mensajes privados.

- **Requerimientos No Funcionales:**

Interfaces externas

- El sistema deberá poder integrarse con el existente Sistema de Despacho de Pedidos.
- Se deberán documentar las interfaces implementadas previendo futuras interacciones con otros sistemas externos.

Rendimiento

- Se requiere una aplicación ágil y liviana con buen tiempo de respuesta. A su vez el sistema deberá ser capaz de soportar la carga actual de datos (1.200 farmacias registradas en el sistema, 14.000 productos y se estiman 200 pedidos online por día).

Diseño

- La arquitectura del sistema deberá ser lo suficientemente flexible para poder utilizar el sistema final como base para desarrollo de servicios adicionales en un futuro, así como también permitir la integración con otros sistemas que pudieran ser de interés para el cliente.

Seguridad

- El tráfico de información entre los sistemas externos y la aplicación web deberá ser a través de un medio de comunicación seguro a elección.

Ambiente y otros requerimientos

- El producto deberá poder correr sobre los servidores de D.U.S.A.
- El sistema deberá estar disponible en un régimen 24/7.
- El sistema deberá correr en Windows XP Service Pack 3, IE versión 6.0 y superior y Firefox versión 12 en adelante.

Estos requerimientos fueron obtenidos a partir del documento de Especificación de Requerimientos.

3. Estrategia de Verificación

Esta sección presenta el enfoque recomendado para la verificación. Describe cómo se verificarán los elementos.

3.1 Tipos de pruebas

3.1.1 Prueba de integridad de los datos y la base de datos

1. Objetivo de la prueba

Asegurar que los métodos y procesos de acceso a la base de datos funcionan correctamente y sin corromper datos.

2. Técnica

Se invocará cada método o proceso de acceso a la base de datos con datos válidos y no válidos. También se realizarán consultas para asegurar que se han guardado los datos correctos, que todos los eventos de la base de datos ocurrieron correctamente, o repasar los datos devueltos para asegurar que se recuperaron datos correctos por la vía correcta.

3. Criterio de aceptación

Todos los métodos y procesos de acceso a la base de datos funcionan como fueron diseñados y sin datos corruptos.

4. Consideraciones especiales

La prueba requiere un entorno de administración de DBMS o controladores para ingresar o modificar información directamente en la base de datos.

Los procesos deben ser invocados manualmente.

Se utilizará una base de datos pequeña, proporcionada por el cliente, para aumentar la facilidad de inspección de los datos verificando que no sucedan eventos no aceptables.

3.1.2 Prueba de Funcionalidad

La prueba de funcionalidad se enfoca en requerimientos para verificar que se corresponden directamente a casos de usos o funciones y reglas del negocio. Los objetivos de estas pruebas son verificar la aceptación de los datos, el proceso, la recuperación y la implementación correcta de las reglas del negocio. Este tipo de prueba se basa en técnicas de caja negra, que consisten en verificar la aplicación y sus procesos interactuando con la aplicación por medio de la interfaz de usuario y analizar los resultados obtenidos.

1. Objetivo de la prueba

Asegurar la funcionalidad apropiada del objeto de prueba, incluyendo la navegación, entrada de datos, proceso y recuperación.

2. Técnica

Se ejecutará cada caso de uso, flujo de caso de uso, o función usando datos válidos y no válidos, para verificar lo siguiente:

- Obtener los resultados esperados cuando se usan datos válidos.
- Cuando se usan datos no válidos se despliegan los mensajes de error o advertencia apropiados.
- Se aplica apropiadamente cada regla del negocio.
- Se deberá contar con casos de prueba que cuenten con entradas, condiciones de ejecución y resultados esperados. Estos casos de prueba deben ser diseñados aplicando alguna técnica de Testing que asegure un tipo de cubrimiento (por ejemplo, utilizando la técnica de partición de clases de equivalencia, valores límite, entre otras).

3. Criterio de aceptación

Todas las pruebas planificadas se realizaron. Todos los defectos encontrados han sido debidamente identificados.

4. Consideraciones especiales

No hay.

3.1.3 Prueba de Ciclo del Negocio

Esta prueba debe simular las actividades realizadas en el proyecto en el tiempo. Se debe identificar un período, que puede ser un año, y se deben ejecutar las transacciones y actividades que ocurrirían en el período de un año. Esto incluye todos los ciclos diarios, semanales y mensuales y eventos que son sensibles a la fecha.

1. Objetivo de la prueba

Asegurar que la aplicación funciona de acuerdo a los requerimientos del negocio.

2. Técnica

La prueba debe simular ciclos de negocios realizando lo siguiente:

Las pruebas de funcionalidad se deben modificar para aumentar la cantidad de veces que se ejecuta cada función, simulando varios usuarios diferentes en un período determinado.

Todas las funciones sensibles a la fecha se deben ejecutar con fechas válidas y no válidas o períodos de tiempo válidos y no válidos.

Para cada prueba realizada verificar lo siguiente:

- Se obtienen los resultados esperados cuando se usan datos válidos.
- Cuando se usan datos no válidos se despliegan los mensajes de error o advertencia apropiados.
- Se aplica apropiadamente cada regla del negocio.

3. Criterio de aceptación

Todas las pruebas planificadas se realizaron. Todos los defectos encontrados han sido debidamente identificados.

4. Consideraciones especiales

Las fechas del sistema y eventos requieren actividades de soporte especiales. Se requieren las reglas del negocio para identificar apropiadamente los requerimientos y procedimientos a ser verificados.

3.1.4 Prueba de Interfaz de Usuario

Esta prueba verifica que la interfaz de usuario proporcione al usuario el acceso y navegación a través de las funciones apropiada. Además asegura que los objetos presentes en la interfaz de usuario se muestren como se espera y conforme a los estándares establecidos por la empresa.

1. Objetivo de la prueba

Verificar que: la navegación a través de los elementos que se están probando reflejen las funciones del negocio y los requerimientos, incluyendo manejo de ventanas, campos y métodos de acceso; los objetos de las ventanas y características, como menús, tamaño, posición, estado, funcionen de acuerdo a los estándares.

2. *Técnica*

Crear o modificar pruebas para cada ventana verificando la navegación y los estados de los objetos para cada ventana de la aplicación y cada objeto dentro de la ventana.

3. *Criterio de aceptación*

Cada ventana ha sido verificada exitosamente siendo consistente con una versión de referencia o estándar establecido.

4. *Consideraciones especiales*

No todas las propiedades de los objetos se pueden acceder.

3.1.5 Prueba de Performance

En esta prueba se miden y evalúan los tiempos de respuesta, los tiempos de transacción y otros requerimientos sensitivos al tiempo. El objetivo de la prueba es verificar que se logren los requerimientos de performance. La prueba de performance es implementada y ejecutada para poner a punto los destinos de pruebas de performance como función de condiciones de trabajo o configuraciones de hardware.

1. *Objetivo de la prueba*

Verificar la performance de determinadas transacciones o funciones de negocio bajo ciertas condiciones:

- condiciones de trabajo normales conocidas.
- peores casos de condiciones de trabajo conocidas. En nuestro proyecto en particular se hará hincapié en los diferentes navegadores sobre los cuales deberá correr la aplicación web.

2. *Técnica*

- Usar procedimientos de prueba desarrollados para verificar funciones o ciclos de negocio.
- Modificar archivos de datos para aumentar el número de transacciones o los procedimientos de prueba para aumentar el número de iteraciones de ocurrencia de transacciones.
- Las pruebas se deben ejecutar en una máquina (mejor caso de prueba un solo usuario, una sola transacción) y se debe repetir con múltiples usuarios (virtuales o reales).

3. *Criterio de aceptación*

Con una transacción o un usuario: Éxito completo de la prueba sin fallas y dentro del tiempo esperado o requerido.

Con múltiples transacciones y varios usuarios: Éxito completo de la prueba sin fallas y dentro de un tiempo aceptable.

4. *Consideraciones especiales*

Las pruebas de performance deben incluir un trabajo de fondo en el servidor. Esto se puede realizar de distintas formas:

- Enviar transacciones directamente al servidor, generalmente en la forma de consultas (SQL).
- Crear usuarios virtuales para simular muchos clientes, generalmente varios cientos. Se pueden usar herramientas de Emulación de Terminal Remota para lograr este objetivo. Esta técnica también se usa para cargar la red con "tráfico".
- Usar muchos clientes físicos, cada uno corriendo procedimientos de prueba.

La prueba de performance se debe realizar en una máquina dedicada para permitir control total y medición exacta.

Las bases de datos usadas para las pruebas de performance deben tener un tamaño similar a las reales y serán provistas por el cliente.

Los tiempos de espera estimados también se coordinarán con el cliente, en principio se habló de que la aplicación web sea lo más ágil posible y que se mantenga la sesión

abierta durante una media hora o una hora, en caso que el usuario no realizará ningún acción luego de ese tiempo, la sesión se cierra.

3.1.6. Prueba de Carga

La prueba de carga somete los objetos a verificar a diferentes cargas de trabajo para medir y evaluar los comportamientos de performance y la habilidad de los objetos de continuar funcionando apropiadamente bajo diferentes cargas de trabajo. El objetivo es determinar y asegurar que el sistema funciona apropiadamente en circunstancias de máxima carga de trabajo esperada. Además evaluar las características de performance, como tiempos de respuesta, tiempos de transacciones y otros elementos sensitivos al tiempo.

1. Objetivo de la prueba

Verificar el comportamiento de performance de determinados componentes del software bajo condiciones de trabajo diferentes.

2. Técnica

Usar pruebas desarrolladas para funciones o ciclos de negocios y modificar archivos de datos para aumentar el número de transacciones o las pruebas para aumentar la cantidad de ocurrencia de transacciones.

3. Criterio de aceptación

Para múltiples transacciones y múltiples usuarios: Realización exitosa de las pruebas sin fallas y dentro del tiempo aceptable.

4. Consideraciones especiales

La prueba de carga debe realizarse en una máquina dedicada para tener control total y exactitud de mediciones.

Las bases de datos usadas para la prueba deben tener un tamaño similar a las reales.

3.1.7. Prueba de Esfuerzo (stress, competencia por recursos, bajos recursos)

La prueba de esfuerzo en un tipo de prueba de performance implementada y ejecutada para encontrar errores cuando hay pocos recursos o cuando hay competencia por recursos. Poca memoria o poco espacio de disco pueden revelar fallas en el software que no aparecen bajo condiciones normales de cantidad de recursos. Otras fallas pueden resultar al competir por recursos compartidos como bloqueos de bases de datos o ancho de banda de red. La prueba de esfuerzo también puede usarse para identificar el trabajo máximo que el software puede manejar.

1. Objetivo de la prueba

Verificar que el software funciona apropiadamente y sin error bajo condiciones de esfuerzo, como son:

- poca memoria o sin disponibilidad de memoria en el servidor
- cantidad máxima de clientes conectados (alrededor de 200 usuarios a la vez)
- múltiples usuarios realizando la misma operación sobre los mismos datos
- peor caso de volumen de operaciones.

El objetivo de la prueba de esfuerzo es también identificar y documentar las condiciones bajo las cuales el sistema falla y no continua funcionando apropiadamente.

2. Técnica

Usar las pruebas desarrolladas para Performance y Prueba de Carga.

Para probar recursos limitados, las pruebas se deben ejecutar en una sola máquina, y se debe reducir o limitar la memoria en el servidor.

Para las pruebas de esfuerzo restantes, debe usarse múltiples clientes, cualquiera que ejecute las mismas pruebas o pruebas complementarias para producir el peor caso de volumen de operaciones.

3. Criterio de aceptación

Todas las pruebas planeadas se ejecutaron y se alcanzaron o excedieron los límites del sistema sin que el software fallara o las condiciones bajo las que ocurre una falla en el software están fuera de las condiciones especificadas.

4. Consideraciones especiales

Se tendrá en cuenta la cantidad de 1200 usuarios como tope máximo. Pero que estén conectados a la vez, se predice junto con el cliente, un número de usuarios alrededor de 200.

3.1.8. Prueba de Volumen

La Prueba de Volumen somete el software a grandes cantidades de datos para determinar si se alcanzan límites que causan la falla del software. La Prueba de Volumen identifica la carga máxima continua que puede manejar el software a prueba en un período dado.

1. *Objetivo de la prueba*

Verificar que el software funciona correctamente con volúmenes de datos grandes:

- Máximo (real o físicamente posible) número de clientes conectados, o simulados, todos realizando la misma operación (peor caso de operación) por un período de tiempo extenso.
- Máximo tamaño de base de datos y múltiples consultas ejecutadas simultáneamente.

2. *Técnica*

Usar pruebas desarrolladas para Prueba de Performance y Prueba de Carga.

- Se deben usar múltiples clientes, ejecutando las mismas pruebas o pruebas complementarias para producir el peor caso de volumen de operaciones o mezcla en un período de tiempo extenso.
- Se debe crear el tamaño máximo de base de datos (real, escalado o con datos representativos) y múltiples clientes ejecutando consultas simultáneamente por un período de tiempo extenso.

3. *Criterio de aceptación*

Todas las pruebas planificadas se ejecutaron y se han alcanzado o excedido los límites especificados sin que el software falle.

4. *Consideraciones especiales*

¿Qué período de tiempo se considera aceptable para condiciones de gran volumen?, es otro punto interesante a coordinar con el cliente.

3.1.9. Prueba de Seguridad y Control de Acceso

La Prueba de Seguridad y Control de Acceso se enfoca en dos áreas de seguridad:

- Seguridad en el ámbito de aplicación, incluyendo el acceso a los datos y a las funciones de negocios.
- Seguridad en el ámbito de sistema, incluyendo conexión, o acceso remoto al sistema.

La seguridad en el ámbito de aplicación asegura que, basado en la seguridad deseada los actores están restringidos a funciones o casos de uso específicos o limitados en los datos que están disponibles para ellos.

La seguridad en el ámbito de sistema asegura que, solo los usuarios con derecho a acceder al sistema son capaces de acceder a las aplicaciones y solo a través de los puntos de ingresos apropiados.

1. *Objetivo de la prueba*

Seguridad en el ámbito de aplicación: Verificar que un actor pueda acceder solo a las funciones o datos para los cuales su tipo de usuario tiene permiso.

En nuestro caso, si se trata de un Usuario-Web, este tendrá permisos de realizar pedidos y consultas a su historial de pedido así como también ver su información web. En el caso que el administrador del sitio bloquee a dicho Usuario-Web, este no podrá acceder a realizar pedidos.

Seguridad en el ámbito de sistema: Verificar que sólo los actores con acceso al sistema y a las aplicaciones, puedan acceder a ellos. Es decir, solo aquellas farmacias que dispongan de nombre de usuario y contraseña, podrá conectarse a la aplicación.

2. *Técnica*

Seguridad en el ámbito de aplicación: Sólo habrá dos tipos de usuario, un usuario administrador, perteneciente a la empresa D.U.S.A. y por otro lado, los Usuarios-Web de

las farmacias asociadas, que son clientes de la empresa actualmente y están registradas en su base de datos.

Acceso en el ámbito de sistema: Ingresar con datos de usuario y contraseña inexistentes, el sistema no puede permitir el acceso a la aplicación para ese usuario desplegando un mensaje de aviso.

3. Criterio de aceptación

Para cada tipo de actor conocido las funciones y datos apropiados están disponibles, y todas las operaciones funcionan como se espera y ejecutan las pruebas de Funcionalidad de la aplicación.

4. Consideraciones especiales

No hay.

3.1.10. Prueba de Fallas y Recuperación

Las Pruebas de Fallas y Recuperación aseguran que el software puede recuperarse de fallas de hardware, software o mal funcionamiento de la red sin pérdida de datos o de integridad de los datos.

La Prueba de Recuperación es un proceso en el cual la aplicación o sistema se expone a condiciones extremas, o condiciones simuladas, para causar falla, como fallas en dispositivos de Entrada/Salida o punteros a la base de datos inválidos. Los procedimientos de recuperación se invocan y la aplicación o sistema es monitoreado e inspeccionado para verificar que se recupera apropiadamente la aplicación o sistema y se logre la recuperación de datos.

1. Objetivo de la prueba

Verificar que los procesos de recuperación (manual o automáticos) recuperen apropiadamente la base de datos, aplicaciones y sistema a un estado conocido y deseado. En la prueba se incluyen los siguientes tipos de condiciones:

- Interrupción de energía al cliente
- Interrupción de energía al servidor
- Interrupción de comunicaciones mediante los servidores de la red
- Interrupción de comunicación o pérdida de energía de los discos del servidor o con los controladores
- Ciclos incompletos (procesos de filtro de datos interrumpidos, procesos de Sincronización de datos interrumpidos)
- Punteros a la base de datos o claves inválidos
- Elementos de datos en la base de datos inválidos o corruptos.

2. Técnica

Se deben usar las pruebas creadas para probar Funcionalidad y Ciclos de negocio para crear una serie de operaciones. Una vez logrado el punto de comienzo deseado, se deben realizar o simular las siguientes acciones, individualmente:

- Interrumpir la energía del cliente: apagar el PC.
- Interrumpir la energía del servidor: simular o iniciar el proceso de apagado del servidor.
- Interrupción por medio de los servidores de red: simular o iniciar la pérdida de comunicación con la red (desconectar físicamente la comunicación o apagar el servidor de red o router)
- Interrumpir la comunicación o quitar la energía de los discos del servidor o sus controladores: simular o eliminar físicamente la comunicación con uno o más controladores de disco o los discos.
- Una vez que se lograron o simularon estas condiciones, se deben invocar los procedimientos de recuperación.
- Las pruebas de ciclos incompletos utilizan la misma técnica excepto que los procesos de bases de datos deben ser abortados a sí mismos o terminados prematuramente.
- Las últimas dos pruebas requieren que se logre un estado conocido de la base de datos. Se deben corromper manualmente campos de la base de datos, punteros y claves trabajando directamente sobre la base de datos (utilizando herramientas para la base de datos). Se deben ejecutar las pruebas de Funcionalidad y Ciclo de negocio y verificar que los ciclos se completen.

3. Criterio de aceptación

En todos los casos, la aplicación, la base de datos y el sistema deben, en la realización procedimientos de recuperación, volver a un estado conocido y deseable. Este estado incluye corrupción de datos limitada a los campos, punteros o claves corruptos conocidos, y reportes indicando los procesos u operaciones que no se completaron debido a las interrupciones.

4. Consideraciones especiales

Los procedimientos para desconectar cables (simulando falta de energía o pérdida de comunicación) no son deseables o factibles. Se pueden requerir métodos alternativos, como software de diagnóstico. Se requieren los grupos de recursos de Sistemas, Bases de datos y Red.

Estas pruebas deben ejecutarse fuera del horario de trabajo normal o en una máquina aislada.

Nota: Llegado el momento de realizar estas pruebas se consultará al cliente, para saber cuál es el resultado que espera.

3.1.11. Prueba de Configuración

La Prueba de Configuración verifica el funcionamiento del software con diferentes configuraciones de software y hardware.

1. Objetivo de la prueba

Verificar que el software funcione apropiadamente en las configuraciones requeridas de hardware y software.

2. Técnica

Usar las pruebas de Funcionalidad.

- Abrir y cerrar varias sesiones de software que no son objeto de prueba, como parte de la prueba o antes de comenzar la prueba.
- Ejecutar operaciones seleccionadas para simular la interacción del actor con el software objeto de prueba y con el software que no es objeto de prueba.
- Repetir los procedimientos anteriores minimizando la memoria convencional disponible en la máquina cliente.

3. Criterio de aceptación

Por cada combinación de software objeto de prueba y software que no es objeto de prueba, todas las operaciones son completadas exitosamente sin fallas.

4. Consideraciones especiales

Todo el software que no es objeto de prueba que es necesario y debe estar accesible.

¿Qué aplicaciones se usan normalmente?

¿Qué información se maneja en las aplicaciones que se usan normalmente, y que tamaño de información?

Los sistemas, red, servidores de red, bases de datos, deben ser documentados como parte de esta prueba.

3.1.12. Prueba de Instalación

La Prueba de Instalación tiene dos propósitos. Uno es asegurar que el software puede ser instalado en diferentes condiciones (como una nueva instalación, una actualización, y una instalación completa o personalizada) bajo condiciones normales y anormales. Condiciones anormales pueden ser insuficiente espacio en disco, falta de privilegios para crear directorios, etc. El otro propósito es verificar que, una vez instalado, el software opera correctamente. Esto significa normalmente ejecutar un conjunto de pruebas que fueron desarrolladas para Prueba de Funcionalidad.

1. Objetivo de la prueba

Verificar que el software objeto de prueba se instala correctamente en cada configuración de hardware requerida bajo las siguientes condiciones:

- Instalación nueva, una nueva máquina, nunca instalada previamente con Pedidos Online – DUSA.

- Actualización, máquina previamente instalada con la Aplicación Web, con la misma versión
- Actualización, máquina previamente instalada con la Aplicación Web, con una versión anterior.

2. *Técnica*

Manualmente o desarrollando programas, para validar la condición de la máquina destino (nueva, nunca instalado, misma versión, versión anterior ya instalada).

Realizar la instalación. Ejecutar un conjunto de pruebas funcionales ya implementadas para la Prueba de Funcionalidad.

3. *Criterio de aceptación*

Las pruebas de funcionalidad de la Aplicación Web se ejecutan exitosamente sin fallas.

4. *Consideraciones especiales*

¿Qué operaciones se deben seleccionar para realizar una prueba confiable de que la Aplicación Web ha sido exitosamente instalada sin dejar fuera ningún componente importante? En principio se tendrán en cuenta todas las operaciones que tengan relación con los requerimientos funcionales y no funcionales.

3.1.13. Prueba de Documentos

La Prueba de Documentos debe asegurar que los documentos relacionados al software que se generen en el proceso sean correctos, consistentes y entendibles. Se incluyen como documentos los Materiales para Soporte al Usuario, Documentación Técnica, Ayuda en Línea y todo tipo de documento que forme parte del paquete de software.

1. *Objetivo de la prueba*

Verificar que el documento objeto de prueba sea:

- Correcto, esto es, que cumpla con el formato y organización para el documento establecido en el proyecto.
- Consistente, esto es, que el contenido del documento sea fiel a lo que hace referencia. Si el documento es Documentación de Usuario, que la explicación de un procedimiento sea exactamente como se realiza el procedimiento en el software, si se muestran pantallas que sean las correctas.
- Entendible, esto es, que al leer el documento se entienda correctamente lo que expresa y sin ambigüedades, además que sea fácil de leer.

2. *Técnica*

Para verificar que el documento es correcto se debe comparar con el estándar definido si existe o con las pautas de documentación y ver que el documento cumple con ellas.

Para verificar que el documento es Consistente se debe ejecutar el programa siguiendo el documento en caso de los Materiales de Soporte al Usuario y comprobar que lo que se explica en estos documentos es exactamente lo que se ejecuta en el programa. En caso de Documentación Técnica se debe revisar el código al cual corresponde la documentación y comprobar que dicha describe el código.

Para verificar que el documento es entendible, debe comprobar que se entiende correctamente, que no tiene ambigüedades y que sea fácil de leer.

3. *Criterio de aceptación*

El documento expresa exactamente lo que debe expresar, no hay diferencias entre lo que está escrito y el objeto de la descripción (operación de software, código de programa, decisiones técnicas) y se entiende fácilmente.

4. *Consideraciones especiales*

No hay.

3.2 Herramientas

A continuación se mencionan las herramientas utilizadas en el proyecto:

- Google Drive, Gmail y Asana : Para la comunicación entre el equipo del proyecto. La comunicación con el cliente D.U.S.A. será por email.
- JUNIT: Para pruebas unitarias.
- Mantis o utilizar issues de Github: Para el sistema de gestión de incidentes.

- Github: Repositorio para compartir código fuente y reporte de bugs.
- IDE Netbeans o Eclipse.
- La aplicación será deployada en un sistema CentOS 6.4 con servidor Tomcat 6 y JDK 7.

Nota: Resta declarar algunas otras herramientas para el resto de las pruebas.

4. Recursos

En esta sección se presentan los recursos recomendados para la Aplicación Web, sus principales responsabilidades y su conocimiento o habilidades.

4.1 Roles

En la tabla a continuación se muestra la composición de personal para la Aplicación Web en el área Verificación del Software.

Rol	Cantidad mínima de recursos recomendada	Responsabilidades
Responsable de verificación	1	Identifica, prioriza e implementa los casos de prueba. <ul style="list-style-type: none"> • Genera el Plan de Verificación. • Genera el Modelo de Prueba. • Evalúa el esfuerzo necesario para verificar. • Proporciona la dirección técnica. • Adquiere los recursos apropiados. • Proporciona informes sobre la verificación.
Asistente de verificación	4	<ul style="list-style-type: none"> • Ejecuta las pruebas • Registra los resultados de las pruebas. • Recuperar el software de errores. • Documenta los pedidos de cambio.
Administrador de Base de Datos	1	<ul style="list-style-type: none"> • Realiza la gestión y mantenimiento del entorno de los datos (base de datos) de prueba y los recursos. • Administra la base de datos de prueba.

4.2 Sistema

En la siguiente tabla se establecen los recursos de sistema necesarios para realizar la verificación.

Recurso	Nombre/Tipo
Servidor de base de datos	
Red o subred	Internet
Nombre del servidor	Servidor 11 de la empresa

Nombre de la base de datos	Oracle 11G Release 1- Manejador de BD JDBC
PC Cliente para pruebas	
Requerimientos especiales	Windows XP con Internet
Repositorio de pruebas	
Red o subred	A definir.
Nombre del servidor	A definir.

5. Hitos del proyecto de Verificación

La verificación la Aplicación Web debe incorporar actividades de prueba para cada verificación identificada en las secciones anteriores. Se deben identificar los hitos del proyecto de verificación separados para comunicar los logros de estado de proyecto.

Actividad que determina el hito	Esfuerzo (Hs/semana)	Fecha de comienzo	Fecha de finalización
Fase Inicial - Iteración 1 – 1 y 2		19/08/2013	01/09/2013
Planificar la verificación	15-20	25/08/2013	01/09/2013
Fase Inicial - Iteración 2 – 3 y 4		02/09/2013	15/09/2013
Elaborar casos de prueba	10-15	06/09/2013	15/09/2013
Ajuste y Control de Verificación	5-8	06/09/2013	15/09/2013
Ejecutar la verificación	10-15	06/09/2013	15/09/2013
Evaluar la verificación	8-10	08/09/2013	15/09/2013
Fase Elaboración - Iteración 1 – 5 y 6		16/09/2013	29/09/2013
Elaborar casos de prueba	10-15	20/09/2013	29/09/2013
Ajuste y Control de Verificación	5-8	20/09/2013	29/09/2013
Ejecutar la verificación	10-15	20/09/2013	29/09/2013
Evaluar la verificación	8-10	22/09/2013	29/09/2013
Fase Elaboración - Iteración 2 – 7 y 8		30/09/2013	13/10/2013
Elaborar casos de prueba	10-15	04/10/2013	13/10/2013
Ajuste y Control de Verificación	5-8	04/10/2013	13/10/2013

Ejecutar la verificación	10-15	04/10/2013	13/10/2013
Evaluar la verificación	8-10	06/10/2013	13/10/2013
Fase Construcción - Iteración 1 - 9 y 10		14/10/2013	27/10/2013
Elaborar casos de prueba	10-15	18/10/2013	27/10/2013
Ajuste y Control de Verificación	5-8	18/10/2013	27/10/2013
Ejecutar la verificación	10-15	18/10/2013	27/10/2013
Evaluar la verificación	8-10	20/10/2013	27/10/2013
Fase Construcción- Iteración 2 - 11 y 12		28/10/2013	10/11/2013
Elaborar casos de prueba	10-15	03/11/2013	10/11/2013
Ajuste y Control de Verificación	5-8	03/11/2013	10/11/2013
Ejecutar la verificación	10-15	03/11/2013	10/11/2013
Evaluar la verificación	8-10	05/11/2013	10/11/2013
Fase Transición - Iteración 1 - 13 y 14		11/11/2013	24/11/2013
Ajuste y Control de Verificación	5-8	11/11/2013	24/11/2013
Ejecutar la verificación	10-15	11/11/2013	24/11/2013
Evaluar la verificación	8-10	13/11/2013	24/11/2013

6. Entregables

6.1 Modelo de Casos de Prueba

Documento	Modelo de Casos de Prueba
Creado por	El Responsable de verificación, Patricia Rolandi.
Para quien	Es la guía para realizar las pruebas del sistema y lo usarán los Asistentes de verificación y el Responsable de verificación cuando se ejecuten las pruebas del sistema.
Fecha de liberación	Será liberado al final de la semana 4.

6.2 Informes de Verificación

Documento	Se genera un documento Informe de Verificación Unitaria por cada prueba unitaria que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas.
Para quien	Es el retorno para los implementadores de la tarea de verificación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Será liberado luego de cada verificación unitaria.

Documento	Se genera un documento Informe Consolidación por cada consolidación que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas.
Para quien	Es el retorno para los implementadores de la tarea de consolidación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Será liberado luego de cada consolidación.

Documento	Se genera un documento Informe de Verificación de Integración por cada prueba de integración que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas.
Para quien	Es el retorno para los implementadores de la tarea de verificación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Será liberado luego de cada verificación de integración.

Documento	Se genera un documento Informe de Verificación de Sistema por cada prueba de sistema que se realice.
Creado por	Las personas que ejecutan las pruebas.
Para quien	Es el retorno para los implementadores de la tarea de verificación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Será liberado luego de cada verificación de sistema.

6.3 Evaluación de la verificación

Documento	Se genera un documento Evaluación de la verificación por cada prueba que se realice al sistema. Este documento contiene las fallas encontradas en el sistema, la cobertura de la verificación
-----------	--

	realizada y el estado del sistema.
Creado por	El Responsable de verificación, que toma como fuente de su trabajo los Informes de verificación.
Para quien	Es el resumen de la tarea de verificación y es el retorno para todo el equipo de trabajo del estado del sistema.
Fecha de liberación	Será liberado luego de cada verificación, unitaria, de integración y de sistema.

6.4 Informe final de verificación

Documento	El documento Informe final de verificación es el resumen de la verificación final del sistema antes de que sea liberado al entorno del usuario.
Creado por	El Responsable de verificación, que toma como fuente de su trabajo los Informes de verificación.
Para quien	Indica el estado del sistema.
Fecha de liberación	Será liberado luego de la verificación final del sistema.

7. Dependencias

En esta sección se detallan las dependencias, si existen, de las actividades de verificación respecto a otros elementos del sistema.

7.1. Dependencia de personal

Como los asistentes de Verificación tienen a su vez asignado uno o más roles, deben realizar otras tareas, por lo tanto la asignación de personal depende de la iteración dentro del proyecto donde nos encontremos y las actividades que se deban realizar como prioritarias y/o críticas.

Para construir las pruebas sobre los prototipos (y sobre la aplicación final) y ejecutarlas se estima un máximo de dos personas, el responsable de verificación y un asistente. Luego para etapas más avanzadas de verificación del sistema, se intentará utilizar a todo el equipo de verificación, haciendo una división de los distintos tipos de pruebas, siendo el responsable de verificación, el único que participe en todas las instancias de prueba dentro de sus posibilidades.

7.2. Dependencia de software

Se debe verificar el software por medio de las pruebas unitarias y de integración, dichas pruebas deberán ser realizadas por el equipo de implementadores, y deben ser entregadas dentro de las fechas fijadas, para poder luego realizar las pruebas de sistema correspondientes, sin retrasos.

7.3. Dependencia de hardware

Se utilizarán equipos que puedan soportar la aplicación a probar, creando un ambiente similar al que tendrán los Usuarios-Web.

7.4. Dependencia de datos y base de datos de prueba

Se tendrán en cuenta los datos proporcionados por la base de datos de la empresa D.U.S.A..

8. Riesgos

En esta sección se detallan los riesgos detectados que pueden afectar la correcta realización de las tareas de Verificación.

8.1. Planificación

Si suceden atrasos en la liberación del producto de software, habrá menos tiempo para verificar. Y en caso de que esto suceda, posiblemente no se podrá cumplir con el cubrimiento de los casos de prueba preestablecidos.

8.2. Técnico

Se debe lograr separar el ambiente de verificación y de desarrollo, para lograr cumplir con los objetivos de la verificación.

Asimilar a tiempo los conocimientos necesarios para realizar las pruebas utilizando las herramientas necesarias.

Cantidad de máquinas apropiadas para la realización de las pruebas.

8.3. Gestión

Realizando una correcta gestión de proyecto y cumpliendo con el cronograma acordado, se puede lograr mitigar los riesgos, para llevar a cabo una correcta verificación siguiendo las estimaciones realizadas y cumpliendo con el esfuerzo en horas/semana pensado previamente.

Correcta distribución de actividades entre el responsable y los asistentes de Verificación, para que todos podamos cumplir con las demás actividades de los otros roles que también tenemos en el proyecto.

9. Apéndice

9.1. Niveles de gravedad de error

En muchas actividades del proceso de verificación se deben clasificar los errores según su nivel de gravedad. Se asigna un nivel de gravedad a los errores para poder capturar de alguna manera su impacto en el sistema. Además para poder evaluar la verificación y el sistema.

A continuación se da una sugerencia de cuatro niveles diferentes de gravedad de error:

- **Catastrófico:** un error cuya presencia impide el uso del sistema.

- **Crítico:** un error cuya presencia causa la pérdida de una funcionalidad crítica del sistema. Si no se corrige el sistema no satisfará las necesidades del cliente.
- **Marginal:** un error que causa un daño menor, produciendo pérdida de efectividad, pérdida de disponibilidad o degradación de una funcionalidad que no se realiza fácilmente de otra manera.
- **Menor:** un error que no causa perjuicio al sistema, pero que requiere mantenimiento o reparación. No causa pérdida de funcionalidades que no se puedan realizar de otra manera.

9.2. Niveles de aceptación para los elementos verificados

Se debe establecer un nivel de aceptación para los elementos verificados para poder establecer el estado en el que se encuentra el proyecto.

En esta sección se definen niveles de aceptación y los criterios de pertenencia a cada nivel.

Los niveles de aceptación son los siguientes:

- **No aprobado:** el elemento verificado tiene errores catastróficos (uno o varios) que impiden su uso o tiene errores críticos (uno o varios) que hacen que el elemento verificado no sea confiable. El usuario no puede depender de él para realizar el trabajo.
- **Aprobado con Observaciones:** el elemento verificado no tiene errores catastróficos, ni errores críticos, pero tiene errores marginales (uno o varios) que hacen que el elemento de software se degrade en algunas situaciones.
- **Aprobado:** el elemento verificado no tiene errores o tiene errores menores que no afectan el normal funcionamiento del elemento.