

Sistema de Administración de Farmacias

SAF

Plan de Verificación y Validación

Versión 1.1

Historia de revisiones

Fecha	Versión	Descripción	Autor
21/08/2014	1.0	Creación del documento	Juan Pablo Sierra
31/08/2014	1.1	Revisión y Formato	Alfonso Methol

Contenido

1.INTRODUCCIÓN.....	3
1.1.PROPÓSITO	3
1.2.PUNTO DE PARTIDA	3
1.3.ALCANCE	4
1.4.IDENTIFICACIÓN DEL PROYECTO.....	6
1.5.ESTRATEGIA DE EVOLUCIÓN DEL PLAN.....	6
2.REQUERIMIENTOS PARA VERIFICAR.....	6
3.ESTRATEGIA DE VERIFICACIÓN	7
3.1.TIPOS DE PRUEBAS	7
3.1.1.Prueba de integridad de los datos y la base de datos.....	7
3.1.2.Prueba de Funcionalidad	8
3.1.3.Prueba de Ciclo del Negocio.....	9
3.1.4.Prueba de Interfase de Usuario.....	9
3.1.5.Prueba de Performance.....	11
3.1.6.Prueba de Carga	12
3.1.7.Prueba de Esfuerzo (stress, competencia por recursos, bajos recursos).....	13
3.1.8.Prueba de Volumen.....	14
3.1.9.Prueba de Seguridad y Control de Acceso	14
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.....	19
4.RECURSOS	19
4.1.ROLES	19
4.2.SISTEMA.....	19
5.HITOS DEL PROYECTO DE VERIFICACIÓN.....	20
6.ENTREGABLES	20
6.1.MODELO DE CASOS DE PRUEBA	20
6.2.INFORMES DE VERIFICACIÓN.....	20
6.3.EVALUACIÓN DE LA VERIFICACIÓN	21
6.4.INFORME FINAL DE VERIFICACIÓN	21
7.DEPENDENCIAS	22
7.1.DEPENDENCIA DE PERSONAL	22
7.2.DEPENDENCIA DE SOFTWARE.....	22
7.3.DEPENDENCIA DE HARDWARE	22
7.4.DEPENDENCIA DE DATOS Y BASE DE DATOS DE PRUEBA	22
8.RIESGOS	23
8.1.PLANIFICACIÓN	23
8.2.TÉCNICO	23
8.3.GESTIÓN.....	23
9.APÉNDICE	25
9.1.NIVELES DE GRAVEDAD DE ERROR	25
9.2.NIVELES DE ACEPTACIÓN PARA LO ELEMENTOS VERIFICADOS	25

1. Introducción

1.1 Propósito

El presente documento brinda detalles acerca de la planificación que guiará el proceso de verificación que estará presente en el proyecto "Proyecto X" y cuyos objetivos, que se encuentran alineados con los de éste último, pueden resumirse en los siguientes puntos:

- Guiar el proceso de estudio de los requerimientos de modo de asegurar las cualidades necesarias para que éstos puedan considerarse válidos.
- Reconocer los componentes del sistema de software cuya calidad necesita asegurarse.
- Establecer las estrategias a seguir para optimizar el uso de los recursos humanos dedicados a las tareas de verificación.
- Definir las herramientas a utilizar para una adecuada gestión del proceso de verificación.
- Explicitar los procedimientos que se han de seguir de modo de cumplir con los estándares de calidad a los que el producto debe adecuarse.
- Definir los entregables que resultan como producto del presente plan.

1.2 Punto de partida

El sistema a construir se compone de dos partes importantes. Por una parte, el componente con el que contarán las farmacias, de aquí en más "cliente". Por otra parte, se reconoce al sistema que ejecutará en la empresa D.U.S.A., de aquí en más "servidor", y que brindará servicios de información al cliente, entre los cuales se destacan la lista de artículos, la actualización de precios y la realización de pedidos y facturas.

El destino de la verificación es la de asegurar la calidad en general del sistema, para lo cual es necesario poder definir pautas que hagan efectivo el proceso, entre las cuales se incluye:

- El diseño de las estrategias para realizar las pruebas de exploración del código y permitir hallazgo de errores tempranamente.
- La inclusión de los estándares de calidad a los que debe ajustarse el sistema dentro del plan de modo que el proceso de desarrollo se rija de acuerdo al mismo y el producto final resulte aceptable desde ese punto de vista.
- La creación de los ambientes de prueba para verificar el sistema en las condiciones más similares posibles a la realidad en la que se ha de desenvolver.
- La selección y utilización de herramientas adecuadas para permitir la optimización de recursos humanos y aumento en las holguras temporales de cada hito.

- La definición las prioridades de cada componente a modo de acelerar la verificación y eventual liberación de aquellos elementos del sistema que se necesiten con mayor anticipación.

1.3 Alcance

A continuación se definen cada una de las fases que compondrán la verificación.

Unitaria

El procedimiento es considerar cada componente como una unidad que debe ser verificada y ha de funcionar como se espera que lo haga cuando funcione en conjunto con los demás componentes.

Este tipo de verificación será realizada por quienes implementen el sistema, basándose en que éstos son quienes conocen más a fondo el sistema y podrán erradicar el mayor número de errores en esta etapa. A modo de permitir a los desarrolladores información más útil en la detección de errores y control sobre el sistema, las pruebas se desarrollaran en un entorno que éstos controlen, típicamente el ambiente de desarrollo.

Finalizada la etapa se detallan, en un informe, la información relativa a los errores que se encontraron, aquellos que quedaron resueltos y los que no han podido resolverse por dificultades técnicas, agenda o simplemente han sido postergados porque su prioridad lo ameritaba.

Pruebas de integración

Habiendo verificado que cada componente funciona de una manera que aparenta ser la correcta, es necesario verificar que el sistema como un todo funcione de la forma esperable.

En esta etapa se verifica si cada elemento del sistema de software funciona correctamente cuando es puesto a funcionar de forma colaborativa con el resto del sistema. En este momento las entradas a cada módulo serán, posiblemente, provistas por otros módulos, mientras que las salidas que generan serán utilizadas por otras partes del sistema. En esta etapa se pretende verificar que todos los elementos se corresponden a su especificación y que el sistema efectivamente funciona de forma integral.

Nuevamente éstas serán llevadas a cabo por el equipo de desarrollo, dado que es éste quien tiene un conocimiento más técnico y puede obtener una información más útil frente a los problemas típicos de ésta fase de pruebas. Para brindar la información más detallada, se utilizará el ambiente de desarrollo, por las mismas razones que en las pruebas unitarias.

Pruebas del sistema

Las pruebas de sistema son las pruebas que se realizan sobre el producto de software, considerado de forma integral, y que corresponden al aseguramiento de la calidad en el sentido de incluir todos los requerimientos funcionales que se han acordado y además adecuarse a las consideraciones impuestas por el cliente tales como los requerimientos no funcionales. Éstas son llevadas a cabo por el equipo de verificación, quien trabajará en un entorno de pruebas, independiente del entorno de desarrollo, y en donde se contará con un sistema completo para ser evaluado.

Pruebas funcionales

Las pruebas funcionales corresponden a asegurar que el sistema responde a las necesidades especificadas por el cliente, entre lo que se incluye:

- El sistema debe incluir las funcionalidades que el cliente ha solicitado, tales como control de stock, facturación, actualización de precios y estadísticas que guíen al farmacéutico.
- Verificar que las reglas de negocio se cumplan de forma íntegra.

Para verificar los anteriores puntos, se considerará la creación de casos de prueba que contemplen las funcionalidades y que además aseguren que en todo momento se impongan las reglas de negocio que apliquen para el sistema.

Pruebas no funcionales

En las pruebas no funcionales, se consideran los siguientes aspectos:

- El diseño de la interfaz de usuario debe ser amigable considerando el público objetivo.
- El sistema debe ajustarse a especificaciones de rendimiento, especialmente en las búsquedas de artículos y sincronización de precios, además de poder ejecutar en sistemas con capacidad de cómputo muy variable y posiblemente limitada.
- Seguridad y control del flujo de información en aspectos sensibles tales como el stock de mercadería.
- Capacidad de manejo de volúmenes de información relativamente importantes, según la manifestación del cliente en lo que respecta al número de artículos disponibles.

Para lo anterior se considera ejecutar el sistema en condiciones similares, tanto de cantidad de información como de carga, considerando una farmacia con varios puntos de venta.

1.4 Identificación del proyecto

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

- Especificación de requerimientos

- Presentaciones de la asignatura "Introducción a la Ingeniería de Software"
- Historia organizacional del año 2011

1.5 Estrategia de evolución del Plan

El plan de verificación será gestionado por el Responsable de Verificación y Validación, asegurando que el mismo se ajuste al proyecto o si existen incompatibilidades que las mismas sean solucionadas realizando las adecuaciones necesarias, que se cumpla con el mismo y que el mismo acompañe la evolución del proyecto.

El plan de verificación se evaluará en cada instancia que lo permita, tales como reuniones del equipo o conversaciones con el cliente que presenten nuevas necesidades que el mismo deba contemplar, especialmente en lo que a niveles de calidad o plazos respecta. En función de las evaluaciones que el responsable realice, podrá considerar realizar ajustes al plan, comunicando a cada uno de los asistentes de la propuesta y evaluando entre otras cosas la factibilidad del cambio.

De considerarse conveniente, el responsable realizará los ajustes necesarios y transmitirá mediante las vías oficiales de comunicación, que al presente día es el correo electrónico, que una nueva revisión del plan se encuentra disponible en el repositorio de documentos que se utiliza para dicho fin.

2. Requerimientos para verificar

Los elementos a verificar son los que se encuentran presentes en el documento "Especificación de Requerimientos", cuyo código es RQDRQ, y detalla en profundidad los aspectos que el cliente ha solicitado para el producto.

El mencionado documento se encuentra en revisión, por lo que se hace referencia al mismo para evitar posibles errores de comunicación.

3. Estrategia de Verificación

Las pruebas de verificación, tal como se describe anteriormente, se componen de dos grandes grupos.

En primer lugar, las unitarias y de integración, que estarán a cargo del equipo de desarrollo y serán por tanto diseñadas por éstos. El equipo de verificación estará disponible para auxiliarles en la tarea, aunque han de ser éstos quienes tengan el conocimiento más profundo sobre la implementación del sistema y por tanto quienes han de poder realizar las pruebas más efectivamente.

En segundo lugar, están las pruebas de sistema, que serán llevadas a cabo por el equipo de verificación, y cuya estrategia se describe a continuación.

- Confeccionar los casos de prueba para verificar el sistema.
- Validar los casos de prueba, de modo que se efectivamente sean correctos.
- Preparar un ambiente de pruebas, que podrá ser confeccionado con asistencia del equipo de desarrollo, y el cual sea utilizado por el equipo de

verificación únicamente. El mismo debe contar con las siguientes particularidades:

- Debe ser exclusivo al equipo de desarrollo, tal como se mencionó.
- Debe contar con una versión del sistema instalada tal como se instalaría el producto final, o que al menos permitan el desarrollo de las pruebas tales como se realizarían en la infraestructura del cliente.
- Debe poder replicar el ambiente de producción en lo que respecta a carga y volumen.
- Llevar a cabo la ejecución de los casos de prueba. Para esto se utilizará el sistema tal como sería utilizado por un cliente. Se evaluarán las salidas en función de las entradas.
- Reportar los resultados obtenidos y discutirlos con el equipo de desarrollo.

Tanto en el caso de las pruebas de sistema como en las que lleva a cabo el equipo de desarrollo, será necesaria la aprobación de las mismas por parte del encargado de verificación, lo que hace explícita la necesidad de que el equipo de desarrollo y el de verificación trabajen en conjunto.

3.1 Tipos de pruebas

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

3.1.1.1 Objetivo de la prueba

El objetivo de la prueba es asegurar que los datos no son afectados de forma inesperada, en otras palabras, que solamente se afecte la información cuando deba hacerse, y en caso de afectarse, que la información sea modificada tal como sería esperable y que en ningún momento los datos puedan dañarse.

3.1.1.2 Técnica

La estrategia es ejecutar todos los procesos que puedan desencadenar en modificación de la instancia de la base de datos. Combinar ingresos de modificación con cancelaciones de ingreso, ingresar datos inválidos que no deberían ingresarse, verificar casos adicionales como caracteres especiales en campos de texto, etcétera.

Luego de ejecutar las pruebas es necesario verificar el estado de la base de datos, que podrá consultarse de forma directa o utilizando el sistema, dependiendo del caso, dado que no toda la información podría ser visible en desde el mismo.

3.1.1.3 Criterio de aceptación

Para que las pruebas se consideren satisfactorias no deben existir situaciones como las descritas en el punto "Objetivo de la prueba".

3.1.1.4 Consideraciones especiales

La prueba requiere acceso directo a la base de datos para evaluar el estado de la misma, base de datos que convenientemente será pequeña y controlada para poder evaluar más fácilmente los resultados de las operaciones.

3.1.2 Prueba de Funcionalidad

La prueba de funcionalidad se enfoca en los aspectos más relevantes y específicos de las reglas de negocio y los requerimientos del sistema. La intención es verificar que los procesos funcionales dentro del sistema se corresponden con las peticiones del cliente.

De esta forma, los datos son validados tal como se explicitó, que la información es procesada adecuadamente y que la información que el sistema brinda es también la requerida. También puede considerarse la interacción con otros sistemas de software o hardware.

Son típicamente llevadas a cabo por evaluación de las entradas y salidas del sistema durante su ejecución, "Pruebas de caja negra", interactuando con el sistema mediante su interface tal como se realizaría en un ambiente de producción. En base a lo que se obtiene se evalúa si el sistema responde como se esperaba o si se ha presentado una falla.

3.1.2.1 *Objetivo de la prueba*

El objetivo es verificar que, a nivel funcional, el sistema se comporta como era de esperarse, y se respetan las reglas de validación de los datos, el procesamiento se realiza adecuadamente y el sistema presenta la información requerida, entre otros.

3.1.2.2 *Técnica*

Dependiendo de lo que se desea verificar se tienen distintas posibilidades.

- Validación de los datos de entrada
 - Se deben ingresar distintas combinaciones de datos correctos e incorrectos para comprobar que el sistema esté validando la información.
 - Ingresar la información en formatos que podrían no ser los esperados. Por ejemplo, proveer al sistema de un archivo con un formato distinto o brindar una dirección de correo que no tenga el carácter "@".
 - Verificar que cuando se ingresan datos erróneos, el sistema notifica del problema y que cuando los mismos son correctos, se obtienen los resultados esperables.
- Validación de las reglas de negocio.
 - Considerar en las pruebas la mayor cantidad de situaciones, casos bordes y evaluar que el sistema lleva a cabo el procesamiento de los datos de forma correcta.
- Verificación de la información provista
 - Considerar la mayor cantidad de casos posibles, y para cada uno llevar a cabo la visualización de la información. Por ejemplo, en el caso de que existan multiplicidades, por ejemplo "Un producto puede tener múltiples presentaciones", considerar verificar cero presentaciones, una presentación o múltiples presentaciones, siempre ajustándose a lo que sea posible en función de los requerimientos (dado que podría ocurrir que algún caso no sea válido).
 - Verificar que en todos los casos la información es completa y correcta.
- Validación de la integración con otros sistemas
 - Verificar que la información recibida desde otros sistemas es correcta, completa e íntegra.
 - Verificar que la información se visualiza adecuadamente en caso de que sea una salida a pantalla, o se envía correctamente a la impresora o el servicio externo recibe correctamente la información enviada.

3.1.2.3 Criterio de aceptación

Para la aceptación es necesario cubrir todas las pruebas consideradas inicialmente, y que todos los defectos que se hayan podido presenciar sean identificados.

3.1.2.4 Consideraciones especiales

En el caso de la interacción con D.U.S.A. será importante verificar que la información que se recibe se realiza de forma completa, dado que el volumen de información es grande, y dado que no se tiene control sobre ese sistema.

3.1.3 Prueba de Ciclo del Negocio

Para contar con información útil se debe ingresar en la base de datos información que simule una cierta actividad en el negocio. Esto incluye todos los ciclos diarios, semanales y mensuales y eventos que son sensibles a la fecha. De esta forma, existirá un conjunto de datos que permitirá validar que el sistema funciona de la forma esperada, según lo indican los requerimientos del negocio.

3.1.3.1 Objetivo de la prueba

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

3.1.3.2 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.

3.1.3.3 Criterio de aceptación

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

3.1.3.4 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 Interface de Usuario

En esta prueba se determina si la interface que el sistema presenta al usuario se ajusta a los requerimientos especificados y además permite una navegación adecuada por las distintas funcionalidades que el sistema provee.

3.1.4.1 *Objetivo de la prueba*

Se busca determinar que la interface cumple con los atributos que el cliente ha definido, entre los que se encuentran:

- Usabilidad
- Uso de imágenes (logo)
- Configurabilidad
- Uso de colores adecuado

Además es necesario verificar que la interfaz gráfica no supone un obstáculo al momento de utilizar el sistema y que en cambio permite la utilización del mismo al nivel que se ha especificado. Esto supone permitir la navegación hacia aquellas secciones que deban accederse y no permitirlo para aquellas que no.

3.1.4.2 *Técnica*

Considerar un conjunto de pruebas que permitan verificar que la navegación hacia cada una de las secciones se permite de forma razonable, utilizando distintos mecanismos, entre los que se destacan:

- Uso de menús
 - Utilización de botones de confirmación y de cancelación
- Se debe verificar que los atributos de calidad que el cliente ha especificado se encuentran presentes en el software, comprobando que la navegación es intuitiva¹ y que la visualización es acorde a lo especificado.

3.1.4.3 *Criterio de aceptación*

Todas las secciones del sistema han sido visitadas, accedidas de distintas formas, y han mostrado cumplir con los atributos de calidad especificados.

3.1.4.4 *Consideraciones especiales*

N/a.

3.1.5 Prueba de Performance

En la prueba de rendimiento interesa evaluar los tiempos de respuesta o procesamiento de datos que se obtienen al utilizar el sistema. De esta manera, se intenta emular la realidad que podría tenerse en un servidor con una cierta carga o un equipo con capacidad de cómputo limitada y verificar que el sistema completa las acciones requeridas dentro de los límites establecidos.

3.1.5.1 *Objetivo de la prueba*

El objetivo de esta prueba es verificar que el rendimiento del sistema es satisfactorio al realizar las distintas operaciones en las condiciones en las que podría encontrarse en un ambiente de producción. Esto incluye verificar que el sistema responde adecuadamente en:

- condiciones de trabajo normales conocidas.
- peores casos de condiciones de trabajo conocidas.

¹Aún resta definir qué se entiende por esto

3.1.5.2 *Técnica*

- Considerar múltiples clientes accediendo a la base de datos, por ejemplo realizando consultas de stock.
- 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).

3.1.5.3 *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.

3.1.5.4 *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.

3.1.6 **Prueba de Carga**

En la prueba de carga, los elementos que componen el sistema son sometidos a distintas cargas de trabajo a modo de evaluar el desempeño de los mismos en función de la cantidad de procesamiento requerido 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, tanto en situaciones normales como en circunstancias críticas con una carga de trabajo esperada máxima. Se aprovecha la instancia demás para evaluar las características de performance, como tiempos de respuesta, tiempos de transacciones y otros elementos sensitivos al tiempo.

3.1.6.1 *Objetivo de la prueba*

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

3.1.6.2 *Técnica*

En el caso de las búsquedas de artículos, el procedimiento a seguir es, considerar una base de datos con una cantidad de artículos similar a la esperable y ejecutar una cantidad intensiva de búsquedas, esperando que las mismas se completen y en el tiempo esperado.

3.1.6.3 Criterio de aceptación

Será aceptado si las operaciones se completan de la forma esperable y en una cantidad de tiempo dentro de los límites establecidos.

3.1.6.4 Consideraciones especiales

Para evitar alteración de los datos, es necesario que la máquina en la que se han de ejecutar las pruebas se esté usando exclusivamente para ese fin, a modo de permitir un 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 es 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.

3.1.7.1 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
- 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.

3.1.7.2 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, deber usarse múltiples clientes, cualquiera que ejecute las mismas pruebas o pruebas complementarias para producir el peor caso de volumen de operaciones.

3.1.7.3 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.

3.1.7.4 Consideraciones especiales

Las pruebas de esfuerzo de red pueden requerir herramientas de red para cargar la red con mensajes o paquetes.

La cantidad de disco del servidor usada por el sistema debe ser reducida temporalmente para restringir el espacio disponible para crecimiento de la base de datos.

Sincronizar el acceso simultáneo de varios clientes accediendo a los mismos datos.

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 causen la falla del software. La Prueba de Volumen identifica la carga máxima continua que puede manejar el software a prueba en un período dado.

3.1.8.1 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.

3.1.8.2 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.]

3.1.8.3 Criterio de aceptación

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

3.1.8.4 Consideraciones especiales

N/A

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.

En el caso del presente proyecto, la seguridad en la aplicación se guiará por una serie de roles cuyos permisos pueden definirse, limitando así las acciones que un usuario puede realizar sobre el sistema. Por otra parte, la seguridad de la base de datos se guiará por una serie de pautas que limitarán el acceso a la misma.

3.1.9.1 Objetivo de la prueba

Seguridad en el ámbito de aplicación

Dado un usuario, verificar que el mismo sólo puede acceder a las funcionalidades o visualización de datos que cuyos permisos, que se le han otorgado, se lo permitan.

Seguridad en el ámbito de sistema

Verificar que la seguridad del sistema no se puede comprometer por un agente externo al mismo.

3.1.9.2 Técnica

Seguridad en el ámbito de aplicación

Para verificar este tipo de seguridad es necesario:

- Considerar el conjunto de privilegios que es posible conceder a un usuario
- Estudiar las distintas maneras en las que los mismos pueden ser combinados
- Evaluar el conjunto de acciones que los usuarios pueden realizar, en función de los privilegios
- Verificar que cada usuario pueda realizar las operaciones que se encuentren en el conjunto anterior, y no pueda realizar aquellas que no.
- Actualizar los permisos y verificar que los mismos se han hecho efectivos.

Acceso en el ámbito de sistema

Este caso es más difícil de asegurar, pero es posible verificar algunos aspectos que hacen al sistema más seguro, como:

- La base de datos no está expuesta al público (internet)
- La aplicación está disponible sólo para aquellos usuarios autorizados

3.1.9.3 Criterio de aceptación

Se verifica que los usuarios acceden únicamente a las funcionalidades permitidas, y recíprocamente.

3.1.9.4 Consideraciones especiales

No es posible verificar razonablemente el criterio de la seguridad a nivel de sistema. Es necesario tomar en cuenta consideraciones del acceso a la red, entre otras.

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.]

3.1.10.1 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.]

3.1.10.2 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 a 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.]

3.1.10.3 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.

3.1.10.4 Consideraciones especiales

N/A

3.1.11 Prueba de Configuración

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

3.1.11.1 Objetivo de la prueba

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

3.1.11.2 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.

3.1.11.3 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.

3.1.11.4 Consideraciones especiales

N/A

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.

3.1.12.1 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 SAF
- actualización, máquina previamente instalada con SAF, con la misma versión
- actualización, máquina previamente instalada con SAF, con una versión anterior.

3.1.12.2 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.

3.1.12.3 Criterio de aceptación

Las pruebas de funcionalidad de SAF se ejecutan exitosamente sin fallas.

3.1.12.4 Consideraciones especiales

Es necesario validar que, en el caso de una actualización, las nuevas funcionalidades funcionen de acuerdo a la nueva versión instalada. En el caso de una nueva instalación, es necesario verificar las funcionalidades que requieren de recursos externos, tales como acceso a datos almacenados en el servidor, información que pueda necesitar ser consultada desde otros dispositivos, tales como lectores de código de barras o POS y que la instalación se comunica adecuadamente con dispositivos de salida, tales como impresoras.

En el caso de una reinstalación es necesario verificar que la nueva instalación se realiza de forma completa, y no encuentra conflictos al detectar una instalación idéntica ya presente.

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.

3.1.13.1 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.²
- 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.

3.1.13.2 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. Limitando el formato de la documentación al uso de hojas de estilo, es necesario verificar que los estilos se encuentran en uso en todo momento y que los mismos coinciden con los brindados en la plantilla base para el documento.

²Debe adecuarse al que se encuentra presente en el sitio de MUM.

Para verificar que el documento es consistente se debe analizar en profundidad lo que el mismo contiene. 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, implica que en ese caso sí existe una concordancia por lo cual sí existe consistencia. En el caso de Documentación Técnica se debe revisar el código al cual corresponde la documentación y comprobar que la misma describe el código presente.

Para verificar que el documento es entendible, debe comprobar que se entiende correctamente, que no tiene ambigüedades y que sea fácil de leer. Para validar esto hay que considerar el público objetivo y evaluar desde ese punto de vista, dado que no es razonable esperar que los documentos técnicos sean comprensibles para cualquier persona.

3.1.13.3 Criterio de aceptación

Para cada documento, se verifica que se encuentran presentes las tres cualidades descritas en el punto anterior.

3.1.13.4 Consideraciones especiales

No todas las personas tienen la misma facilidad de comprensión, ni los mismos conocimientos ni el mismo involucramiento en el proyecto, por lo que es necesario evaluar la facilidad de comprensión del documento de forma cuidadosa.

3.2 Herramientas

- Motor de base de datos: No definido aún.
- Registro de defectos: Trac
- Medición de cubrimiento de código: junit.
- Herramienta de control de versiones: Git.

4. Recursos

4.1 Roles

En la tabla a continuación se muestra la composición de personal para el proyecto [Nombre del proyecto] 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. <ul style="list-style-type: none"> • Proporciona la dirección técnica. • Adquiere los recursos apropiados. • Proporciona informes sobre la verificación.
Asistente de verificación	2	<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	- resta definir -
Red o subred	- resta definir -
Nombre del servidor	- resta definir -
Nombre de la base de datos	- resta definir -
PC Cliente para pruebas	PC - Windows o Linux
Requerimientos especiales	-
Repositorio de pruebas	- resta definir -
Red o subred	- resta definir -
Nombre del servidor	- resta definir -

5. Hitos del proyecto de Verificación

Actividad que determina el hito	Esfuerzo	Fecha de comienzo	Fecha de finalización
Planificar la verificación	15 horas aprox.	21/8/2014	30/8/2014
Elaborar casos de prueba	-	-	-
Ajuste y Control de Verificación	-	-	-
Ejecutar la verificación	-	-	-
Evaluar la verificación	-	-	-

6. Entregables

6.1 Modelo de Casos de Prueba

Documento	Modelo de Casos de Prueba
Creado por	El Responsable de verificación, Juan Pablo Sierra.
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 el 13/09/2014

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

7.1 Dependencia de personal

Considerando que el sistema ha de interactuar con el usuario mediante una interfaz web, no se requiere entrenamiento especial, dado que es esperable que quienes lleven a cabo las pruebas tengan conocimientos básicos de cómo interactuar con un sistema web.

Para las pruebas es necesario contar con recursos de acuerdo al siguiente cronograma:

- Llevar a cabo la verificación de los prototipos: No más de dos personas
- Creación de casos de prueba para el sistema: No más de dos personas
- Ejecución de los casos de prueba del sistema: 4 personas.

No se necesita exclusividad de los asistentes de verificación, por lo que es posible que los mismos puedan desempeñar múltiples tareas simultáneamente.

7.2 Dependencia de software

La agenda de la verificación estará estrechamente ligada a los ciclos de liberación desde el ambiente de desarrollo, dado que para verificar la calidad del producto de software es necesario que el mismo se encuentre en la versión final, dado que en caso contrario se estaría verificando un sistema distinto al que se espera liberar.

No es menor la dependencia que existe con los servicios externos, en este caso los provistos por D.U.S.A., cuya no disponibilidad puede afectar negativamente la verificación, causando retardos cuya resolución se encuentra fuera del alcance del equipo que lleva a cabo el presente proyecto.

7.3 Dependencia de hardware

Deben existir sistemas físicos que den soporte al ambiente de pruebas sobre el cual se van a llevar a cabo las pruebas de verificación.

7.4 Dependencia de datos y base de datos de prueba

Tal como se explicitó en el punto 7.2, la no disponibilidad del servicio externo puede afectar la obtención de información en una forma que podría resultar bloqueante a las pruebas, por ejemplo si falla la sincronización inicial de productos. Para llevar a cabo las pruebas es necesario contar con un conjunto de datos similar al que el cliente posee, por lo que si no es posible realizar la sincronización con D.U.S.A. será necesario que éste último facilite el conjunto de datos por otra vía.

8. Riesgos

A continuación se describen los principales riesgos que se han detectado y podrían afectar al proceso de verificación.

8.1 Planificación

- Plazos establecidos incorrectos
- Falta de documentación clara
- Mala gestión de los recursos
- Mala coordinación (por ejemplo aún no se cuenta con los casos de prueba o no está disponible el entorno de verificación)

8.2 Técnico

- Servicios externos no disponibles (D.U.S.A.)
- Múltiples usuarios realizando distintas pruebas sobre el mismo equipo.
- Problemas de conexión al servidor

8.3 Gestión

Para la mitigación de los riesgos se propone:

- Planificación evolutiva y correctiva. Realizar la planificación inicial, monitorear el avance, obtener conclusiones, realizar los ajustes a la planificación acorde con lo observado.
- Reservar los recursos humanos cuando se estime una mayor carga de trabajo.
- Considerar holguras en la planificación
- Establecer pautas que guíen el proceso de verificación y la comunicación dentro del equipo.
- Establecer fechas límites para contar con los recursos informáticos adecuados previo al comienzo de las pruebas. Esto incluye efectivamente asegurarse que los aspectos básicos de configuración funcionan correctamente.

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 defina niveles de aceptación y los criterios de pertenencia a cada nivel.

Como ejemplo de niveles de aceptación:

- **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.