TIPOS DE PRUEBAS DE SOFTWARE

A continuación se describen las principales tipos pruebas que se pueden realizar a cualquier tipo de software. Cada prueba contendrá como mínimo a siguiente información:

- Objetivo de la prueba

- Descripción de la prueba

- Técnica

- Criterio de Completitud

- Consideraciones Especiales

**PRUEBAS UNITARIAS**

**Prueba Unitaria**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Se focaliza en ejecutar cada módulo (o unidad minima a ser probada, ej = una clase) lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores.  Busca asegurar que el código funciona de acuerdo con las especificaciones y que el módulo lógico es válido. |
| **Descripción de la Prueba:** | * Particionar los módulos en pruebas en unidades lógicas fáciles de probar. * Por cada unidad hay que definir los casos de prueba (pruebas de caja blanca). * Para esto los casos de prueba deben diseñarse de forma tal que se recorran todos los caminos de ejecución posibles dentro del código bajo prueba; por lo tanto el diseñador debe construirlos con acceso al código fuente de la unidad a probar. * Los aspectos a considerar son los siguientes: Rutinas de excepción, Rutinas de error, Manejo de parámetros, Validaciones, Valores válidos, Valores límites, Rangos, Mensajes posibles. |
| **Técnica:** | * Comparar el resultado esperado con el resultado obtenido. * Si existen errores, reportarlos. |
| **Criterio de Completitud:** | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identificaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | Para la elaboración de pruebas unitarias en java se puede utillizar el JUNIT y CACTUS. |

**PRUEBAS DE INTEGRACIÓN**

**Prueba de Integración**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Identificar errores introducidos por la combinación de programas probados unitariamente.  Determina cómo la base de datos de prueba será cargada.  Verificar que las interfaces entre las entidades externas (usuarios) y las aplicaciones funcionan correctamente.  Verificar que las especificaciones de diseño sean alcanzadas.  Determina el enfoque para avanzar desde un nivel de integración de las componentes al siguiente. |
| **Descripción de la Prueba:** | Describe cómo verificar que las interfaces entre las componentes de software funcionan correctamente.  Determina cómo la base de datos de prueba será cargada.  Determina el enfoque para avanzar desde un nivel de integración de las componentes al siguiente.  Decide qué acciones tomar cuando se descubren problemas.  Por cada Caso de Prueba ejecutado:   * Comparar el resultado esperado con el resultado obtenido. |
| **Técnica:** | * Utilizar la técnica top-down. Se empieza con los módulos de nivel superior, y se verifica que los módulos de nivel superior llaman a los de nivel inferior de manera correcta, con los parámetros correctos. * Utilizar la técnica down-top. Se empieza con los módulos de nivel inferior, y se verifica que los módulos de nivel inferior llaman a los de nivel superior de manera correcta, con los parámetros correctos. |
| **Criterio de Completitud:** | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identificaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | Ninguna |

**Prueba de Regresión**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Determinar si los cambios recientes en una parte de la aplicación tienen efecto adverso en otras partes. |
| **Descripción de la Prueba:** | En esta prueba se vuelve a probar el sistema a la luz de los cambios realizados durante el debugging, mantenimiento o desarrollo de la nueva versión del sistema buscando efectos adversos en otras partes. |
| **Técnica:** | * La prueba de regresión es una nueva corrida de casos de prueba previos. * Se requiere de políticas para ser creada la prueba de regresión y decidir qué casos de prueba incluir, para probar eficientemente. * La prueba de regresión es un buen candidato para automatización. Desde que estas pruebas se repiten una y otra vez, las herramientas para minimizar el esfuerzo del trabajo son útiles. * La prueba de viejas funcionalidades es más importante que la de nuevas funcionalidades. * Aquellos casos de uso (y los casos de prueba asociados) que descubren defectos tempranamente deben ser incluidos en la prueba de regresión. |
| **Criterio de Completitud:** | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identificaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | Ninguna |

**Pruebas de Humo (Smoke Testing o Ad Hoc)**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Los objetivos son los siguientes:   * Detectar los errores en realeases tempranos y de manera fácil * Probar el sistema constantemente * Garantizar poco esfuerzo en la integración final del sistema * Asegurar los resultados de las pruebas unitarias * Se reducen los riesgos y a baja calidad. |
| **Descripción de la Prueba:** | Toma éste nombre debido a que su objetivo es probar el sistema constantemente buscando que saque “humo” o falle. En algunos proyectos este tipo de prueba va junto con las pruebas funcionales. Permite detectar problemas que por lo regular no son detectados en las pruebas normales. Algunas veces, si las Pruebas ocurren tarde en el ciclo de desarrollo está será una forma de garantizar el buen desarrollo.  Las pruebas de humo ***NO SON***exhaustivas, pero van de extremo a extremo de la aplicación. |
| **Técnica:** | 1. Realizar una integración de todo el sistema cada cierto periodo (se recomienda un día, máximo una semana)  2. Realizar los casos de prueba asignados a los casos de uso finalizados ese día más los realizados en días anteriores  3. Buscar eficientemente errores |
| **Criterio de Completitud:** | Todas las pruebas planeadas han sido ejecutadas.  Todos los defectos que se identifcaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | Cuando se encuentre un error en el release correspondiente al periodo elegido para hacer las integraciones del sistema, se detiente el desarrollo hasta que el error es corregido.  Este tipo de pruebas es útil en la programación extrema (***extremme programming***) y de sistemas complejos.  Es útil el uso de programas de prueba automáticas que se encarguen de probar os casos de prueba ya ejecutados en realeases anteriores. |

**PRUEBAS DEL SISTEMA**

**Pruebas del Sistema**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación. |
| **Descripción de la Prueba:** | Las pruebas del sistema deben enfocarse en requisitos que puedan ser tomados directamente de casos de uso y reglas y funciones de negocios. El objetivo de estas pruebas es verificar el ingreso, procesamiento y recuperación apropiado de datos, y la implementación apropiada de las reglas de negocios. Este tipo de pruebas se basan en técnicas de caja negra, ésto es, verificar el sistema (y sus procesos internos), la interacción con las aplicaciones que lo usan via GUI y analizar las salidas o resultados.  En esta prueba se determina qué pruebas de Sistema (usabilidad, volumen, desempeño, etc.) asegurarán que la aplicación alcanzará sus objetivos de negocio.  La prueba de Sistema incluye:  Prueba funcionalidad  Prueba de Usabilidad  Prueba de Performance  Prueba de Documentación y Procedimientos  Prueba de Seguridad y Controles  Prueba de Volumen  Prueba de Esfuerzo (Stress)  Prueba de recuperación  Prueba de múltiples sitios  Para sistemas web se recomienda especialmente ***realizar mínimo*** el siguiente grupo de pruebas de sistema:  Humo.  Usabilidad  Performance  Funcionalidad  Para capitalizar el trabajo hasta ahora completado, los casos de prueba de las pruebas previas realizadas pueden frecuentemente ser reorganizados y rehusados durante la prueba de sistema. No obstante, deben ser desarrollados casos de prueba adicionales para aquellos aspectos del sistema, tales como documentación, procedimientos y desempeño que no han sido probados durante la prueba unitaria y de integración.  La prueba de sistema es compleja porque intenta validar un número de características al mismo tiempo, a diferencia de otras pruebas que sólo se centran en uno o dos aspectos del sistema al mismo tiempo. |
| **Técnica:** | Ejecute cada caso de uso, flujo básico o función utilizando datos válidos e inválidos, para verificar que:  Los resultados esperados ocurren cuando se utiliza un dato válido.  Los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido.  Cada regla de negocios es aplicada adecuadamente. |
| **Criterio de Completitud:** | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identifcaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | * Identifique o describa aquellos aspectos (internos o externos) que impactan la implementación y ejecución de las pruebas del Sistema |

**Pruebas de Desempeño**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Validar el tiempo de respuesta para las transacciones o funciones de negocios bajo las siguientes dos condiciones:   * Volumen normal anticipado * Volumen máximo anticipado. |
| **Descripción de la Prueba:** | Las pruebas de desempeño miden tiempos de respuesta, índices de procesamiento de transacciones y otros requisitos sensibles al tiempo. El objetivo de las pruebas de desempeño es verificar y validar los requisitos de desempeño que se han especificado (en este caso, el desempeño ofrecido por el proponente).  Las pruebas de desempeño usualmente se ejecutan varias veces, utilizando en cada una, carga diferente en el sistema. La prueba inicial debe ser ejecutada con una carga similar a la esperada en el sistema. Una segunda prueba debe hacerse utilizando una carga máxima.  Adicionalmente, las pruebas de desempeño pueden ser utilizadas para perfilar y refinar el desempeño del sistema como una función de condiciones tales como carga o configuraciones de hardware  Las principales actividades son:   * Comparar el desempeño del sistema actual con los requisitos, * Poner a punto el sistema para mejorar las métricas de desempeño y proyectar la capacidad futura de carga del sistema.   Los objetivos de nivel de servicio definidos deben guiar la prueba de performance. Algunas características que afectan el desempeño son:   * Errores lógicos * Procesamiento ineficiente * Diseño pobre: muchas interfases, instrucciones y entradas/salidas. * Cuellos de botella en discos, CPU ó canales de entrada/salida * Salidas del sistema * Tiempos de respuesta * Capacidad de almacenamiento * Tasa de entrada/salida de datos * Número de transacciones que pueden ser manejadas simultáneamente.   Las pruebas de desempeño utilizan las técnicas de caja blanca y caja negra. |
| **Técnica:** | * Utilice los procedimientos de prueba desarrollados para las pruebas del modelo del negocio (Pruebas del Sistema).· * Modifique archivos de datos (para incrementar el número de transacciones) o los scripts para incrementar el número de veces que ocurre cada transacción.· * Los scripts deben ser ejecutados en una máquina y deben ser repetidos con múltiples clientes (virtuales o actuales). (Ver consideraciones especiales). |
| **Criterio de Completitud:** | * Un Usuario / Una Transaccion. Se completaron las pruebas sin ninguna falla y dentro del tiempo esperado o requerido por transacción. * Múltiples transacciones, múltiples usuarios. Se completaron las pruebas de los scripts sin ninguna falla y dentro del tiempo esperado. |
| **Consideraciones Especiales:** | Unas pruebas de desempeño integrales incluyen tener una carga en background en el servidor. Hay varios métodos que pueden ser utilizados para hacer ésto:  o Transacciones dirigidas directamente al servidor, usualmente en forma de sentencias SQL.·  o Creación de usuarios virtuales para simular muchos clientes (usualmente varios cientos). Se utilizan herramientas de Emulación de Terminales Remotas para obtener esta carga. Esta técnica también puede ser utilizada para cargar de tráfico la red.  o Use múltiples clientes físicos, cada uno corriendo los scripts de prueba.  Las pruebas de desempeño deben ser ejecutadas en una máquina dedicada o en un tiempo dedicado. Esto permite control total y medidas precisas.  La Base de datos utilizada para pruebas de desempeño debe ser de un tamaño real o proporcionalmente más grande que la diseñada. |

**Pruebas de Carga**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Verificar el tiempo de respuesta del sistema para transacciones o casos de uso de negocios, bajo diferentes condiciones de carga. |
| **Descripción de la Prueba:** | Las pruebas de carga miden la capacidad del sistema para continuar funcionando apropiadamente bajo diferentes condiciones de carga.  La meta de las pruebas de carga es determinar y asegurar que el sistema funciona apropiadamente aún más allá de la carga de trabajo máxima esperada. Adicionalmente, las pruebas de carga evalúan las características de desempeño (tiempos de respuesta, tasas de transacciones y otros aspectos sensibles al tiempo). |
| **Técnica:** | * Use los scripts desarrolladas para Pruebas del Negocio. * Modifique archivos de datos (para incrementar el número de transacciones o veces que cada transacción ocurre). |
| **Criterio de Completitud:** | * Múltiples transacciones, múltiples usuarios. Se completaron las pruebas de los scripts sin ninguna falla y dentro del tiempo esperado. |
| **Consideraciones Especiales:** | * Las pruebas de carga deben ser ejecutadas en una máquina dedicada o en un tiempo dedicado. Esto permite control total y medidas precisas.· * La Base de datos utilizada para pruebas de desempeño debe ser de un tamaño real o proporcionalmente más grande que la diseñada. |

**Pruebas de Stress**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Verificar que el sistema funciona apropiadamente y sin errores, bajo estas condiciones de stress:   * Memoria baja o no disponible en el servidor. * Máximo número de clientes conectados o simulados (actuales o físicamente posibles) * Múltiples usuarios desempeñando la misma transacción con los mismos datos. * El peor caso de volumen de transacciones (ver pruebas de desempeño).   NOTAS: La meta de las pruebas de stress también es identificar y documentar las condiciones bajo las cuales el sistema FALLA. |
| **Descripción de la Prueba:** | Las pruebas de stress se proponen encontrar errores debidos a recursos bajos o completitud de recursos. Poca memoria o espacio en disco puede revelar defectos en el sistema que no son aparentes bajo condiciones normales. Otros defectos pueden resultar de incluir recursos compartidos, como bloqueos de base de datos o ancho de banda de la red. Las pruebas de stress identifican la carga máxima que el sistema puede manejar.  El objetivo de esta prueba es investigar el comportamiento del sistema bajo condiciones que sobrecargan sus recursos. No debe confundirse con las pruebas de volumen: un esfuerzo grande es un pico de volumen de datos que se presenta en un corto período de tiempo.  Puesto que la prueba de esfuerzo involucra un elemento de tiempo, no resulta aplicable a muchos programas, por ejemplo a un compilador o a una rutina de pagos.  Es aplicable, sin embargo, a programas que trabajan bajo cargas variables, interactivos, de tiempo real y de control de proceso.  Aunque muchas pruebas de esfuerzo representan condiciones que el programa encontrará realmente durante su utilización, muchas otras serán en verdad situaciones que nunca ocurrirán en la realidad. Esto no implica, sin embargo, que estas pruebas no sean útiles.  Si se detectan errores durante estas condiciones “imposibles”, la prueba es valiosa porque es de esperar que los mismos errores puedan presentarse en situaciones reales, algo menos exigentes. |
| **Técnica:** | * Use los scripts utilizados en las pruebas de desempeño. * Para probar recursos limitados, las pruebas se deben correr en un servidor con configuración reducida (o limitada). * Para las pruebas de stress restantes, deben utilizarse múltiples clientes, ya sea corriendo los mismos scripts o scripts complementarios para producir el peor caso de volumen de transacciones. |
| **Criterio de Completitud:** | Todas las pruebas planeadas han sido ejecutadas y excedidas sin que el sistema falle. ( O si las condiciones en que el sistema falle ocurren por fuera de las condiciones especificadas). |
| **Consideraciones Especiales:** | Producir stress en la red puede requerir herramientas de red para sobrecargarla de tráfico.  El espacio en disco utilizado para el sistema debe ser reducido temporalmente para limitar el espacio disponible para el crecimiento de la Base de datos.  Sincronización de varios clientes accediendo simultáneamente los mismos registros. |

**Pruebas de Volumen**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Verificar que la aplicación funciona adecuadamente bajo los siguientes escenarios de volumen:  o Máximo (actual o físicamente posible) número de clientes conectados (o simulados), todos ejecutando la misma función (peor caso de desempeño) por un período extendido.  o Máximo tamaño de base de datos (actual o escalado) y múltiples consultas ejecutadas simultáneamente |
| **Descripción de la Prueba:** | Las pruebas de volumen hacen referencia a grandes cantidades de datos para determinar los límites en que se causa que el Sistema falle. También identifican la carga máxima o volumen que el sistema puede manejar en un período dado. Por ejemplo, si el sistema está procesando un conjunto de registros de Base de datos para generar un reporte, una prueba de volumen podría usar una Base de datos de prueba grande y verificar que el sistema se comporta normalmente y produce el reporte correctamente.  El objetivo de esta prueba es someter al sistema a grandes volúmenes de datos para determinar si el mismo puede manejar el volumen de datos especificado en sus requisitos.  Algunos ejemplos de escenarios de prueba de volúmenes:  Un compilador puede ser alimentado por un programa para compilar que sea absurdamente grande.  Un editor de nexos puede recibir un programa que contenga miles de módulos.  Un simulador de circuito electrónico puede recibir un circuito diseñado con miles de componentes.  Puesto que obviamente, la prueba de volumen es una prueba costosa, tanto en tiempo de máquina como en personal, se debe tratar de no exceder los límites. Sin embargo, todo programa debería ser expuesto, al menos, a algunas pruebas de volumen. |
| **Técnica:** | Utilice los scripts diseñados para las pruebas de desempeño.  Deben usarse múltiples clientes, ya sea corriendo las mismas pruebas o pruebas complementarias para producir el peor caso de volumen (ver pruebas de stress) por un período extendido.  Se utiliza un tamaño máximo de Base de datos. (actual, escalado o con datos representativos) y múltiples clientes para correr consultas simultáneamente para períodos extendidos. |
| **Criterio de Completitud:** | Todas las pruebas planeadas han sido ejecutadas y los límites especificados en el sistema se han conseguido o excedido sin que el sistema falle. |
| **Consideraciones Especiales:** | * Qué período de tiempo debería considerarse como aceptable para condiciones de volumen alto? |

**Pruebas de Recuperación y Tolerancia a fallas**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Verificar que los procesos de recuperación (manual o automática) restauran apropiadamente la Base de datos, aplicaciones y sistemas, y los llevan a un estado conocido o deseado. Los siguientes tipos de condiciones deben incluirse en la prueba:  Interrupción de electricidad en el cliente.  Interrupción de electricidad en el servidor  Interrupción en la comunicación hacia el servidor (caídas de red)  Interrupción en la comunicación con los controladores de disco.  Ciclos incompletos (procesos de consultas interrumpidos, procesos de sincronización de datos interrumpidos)  Llaves o apuntadores de base de datos inválidos  Elementos corruptos o inválidos en la base de datos. |
| **Descripción de la Prueba:** | Estas pruebas aseguran que una aplicación o sistema se recupere de una variedad de anomalías de hardware, software o red con pérdidas de datos o fallas de integridad.  Las pruebas de tolerancia a fallas aseguran que, para aquellos sistemas que deben mantenerse corriendo, cuando una condición de falla ocurre, los sistemas alternos o de respaldo pueden tomar control del sistema sin pérdida de datos o transacciones.  Las pruebas de recuperación son contrarias a las pruebas en que la aplicación o sistema es expuesto a condiciones extremas (o condiciones simuladas), tales como fallas en dispositivos en entrada/salida o llaves o apuntadores inválidos de base de datos. Los procesos de recuperación se invocan y la aplicación es monitoreada y/o inspeccionada para verificar que éstos mecanismos se han ejecutado en forma apropiada.  El objetivo de esta prueba es determinar la habilidad del sistema para recuperarse de una falla de hardware o software. Esta prueba evalúa las características de contingencia construidas en el sistema para procesar interrupciones y para volver a puntos específicos en el ciclo de procesamiento del sistema. La recuperación debe ser considerada en el proceso de diseño. Errores de programación o de datos pueden ser incorporados ex profeso en un sistema para determinar si se puede recuperar de ellos. Las fallas de equipo (por ejemplo errores de paridad en memoria, errores en dispositivos de entrada/salida) pueden ser simuladas. |
| **Técnica:** | Se deben utilizar las pruebas creadas para la Funcionalidad del sistema y Procesos de Negocios para crear una serie de transacciones. Una vez se alcanza el punto inicial de las pruebas de recuperación, se deben realizar o simular las siguientes acciones:  Interrupción de electricidad en el cliente.  Interrupción de electricidad en el servidor: simular o iniciar procedimientos de pérdida de energía para el servidor.  Interrupción de la comunicación en la red. (desconectar físicamente los cables o apagar los hubs o routers)  Interrupción de la comunicación con los controladores de disco: simular o eliminar físicamente la comunicación con uno o mas controladores o dispositivos.  Una vez se realizan estas acciones, se deben ejecutar series de transacciones, y luego, una vez alcanzado el segundo punto de pruebas, se deben invocar los procedimientos de recuperación.  Las pruebas para ciclos incompletos utilizan la misma técnica que se describe arriba, excepto que los procesos de Base de datos deban ser abortados o terminados prematuramente.  Las pruebas para estas condiciones requieren que se llegue a un estado conocido previamente en la Base de datos. Algunos campos, apuntadores y llaves deben ser modificados manualmente, valiéndose de las herramientas que ofrezca la SSPD. Las transacciones adicionales deben ser ejecutadas utilizando las pruebas para Funcionalidad de la aplicación y para Procesos de Negocios. |
| **Criterio de Completitud:** | En todos los casos mencionados, la Base de datos, aplicación y otros sistemas deben retornar a un estado conocido y deseado, una vez se completan los procedimientos de recuperación. Este estado podría incluir que el daño de datos se limite solamente a los campos, llaves o apuntadores que se sabe que fueron alterados, y reportes indicando los procesos o transacciones que no fueron completadas debido a las interrupciones producidas. |
| **Consideraciones Especiales:** | Las pruebas de recuperación pueden llegar a ser molestas. Los procedimientos para desconectar cables o simular pérdida de electricidad pueden ser poco factibles o deseables. Podrían llegar a requerirse métodos alternativos, como herramientas de diagnóstico.  Se requiere la participación de personal de la red, administradores de la base de datos y del sistema.  Estas pruebas deben ser ejecutadas en horas no laborables o en máquinas aisladas. |

**Prueba de Múltiples Sitios**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Detectar fallas en configuraciones y comunicaciones de datos entre múltiples sitios. |
| **Descripción de la Prueba:** | El propósito de esta prueba es evaluar el correcto funcionamiento del sistema o subsistema en múltiples instalaciones. |
| **Técnica:** | Realizar casos de prueba que verifiquen mínimo lo siguiente:   1. Consistencia de las opciones de configuración para el sistema a través de los sitios 2. Empaquetamiento del sistema para múltiples instalaciones 3. Sincronización de datos entre sitios 4. Comunicación de datos entre sistemas en diferentes sitios 5. Rompimiento de funciones de sistema a través de los sitios. 6. Consistencia de controles y seguridad a través de los sitios |
| **Criterio de Completitud:** | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identificaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | Ninguna |

**Prueba de Compatibilidad y Conversión**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Buscar problemas de compatibilidad y conversión en los sistemas. |
| **Descripción de la Prueba:** | El propósito es demostrar que los objetivos de compatibilidad no han sido logrados y que los procedimientos de conversión no funcionan.  La mayoría de los programas que se desarrollan no son completamente nuevos; con frecuencia son reemplazos de partes deficientes, ya sea de sistemas de procesamiento de datos, o sistemas manuales.  Como tales, los programas tienen a menudo objetivos específicos con respecto a su compatibilidad y a sus procedimientos de conversión con el sistema existente. |
| **Técnica:** | Desarrollar casos de prueba que permitan detectar deficiencias con:  Compatibilidad entre programas  Conversión de datos |
| **Criterio de Completitud:** | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identificaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | Ninguna |

**Pruebas de Integridad de Datos y Base de Datos**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Asegurar que los métodos de acceso y procesos funcionan adecuadamente y sin ocasionar corrupción de datos. |
| **Descripción de la Prueba:** | La Base de datos y los procesos de Base de datos deben ser probados como sistemas separados del proyecto . Estos sistemas deberían ser probados sin usar interfaces de usuario (como las interfaces de datos). Se necesita realizar investigación adicional en el DBMS para identificar las herramientas y técnicas que podrían existir para soportar las pruebas identificadas más adelante. |
| **Técnica:** | * Invoque cada método de acceso y proceso de la Base de datos, utilizando en cada uno datos válidos e inválidos. * Analice la Base de datos, para asegurar que los datos han sido grabados apropiadamente, que todos los eventos de Base de datos se ejecutaron en forma correcta y revise los datos retornados en diferentes consultas. |
| **Criterio de Completitud:** | Todos los métodos de acceso y procesos de la Base de datos funcionan como fueron diseñados y sin corrupción de datos |
| **Consideraciones Especiales:** | * Las pruebas pueden requerir un ambiente de DBMS o controladores para ingresar o modificar datos directamente en la Base de datos. * Se debe utilizar un conjunto pequeño de datos para incrementar la visibilidad de cualquier evento anormal o inesperado. * Los procesos pueden ser invocados manualmente. |

**Pruebas de Seguridad y Control de Acceso**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Nivel de seguridad de la aplicación: Verifica que un actor solo pueda acceder a las funciones y datos que su usuario tiene permitido.  Nivel de Seguridad del Sistema: Verificar que solo los actores con acceso al sistema y a la aplicación están habilitados para accederla. |
| **Descripción de la Prueba:** | Las pruebas de seguridad y control de acceso se centran en dos áreas claves de seguridad:   * Seguridad del sistema, incluyendo acceso a datos o Funciones de negocios y * Seguridad del sistema, incluyendo ingresos y accesos remotos al sistema.   Las pruebas de seguridad de la aplicación garantizan que, con base en la seguridad deseada, los usuarios están restringidos a funciones específicas o su acceso está limitado únicamente a los datos que está autorizado a acceder. Por ejemplo, cada usuario puede estar autorizado a crear nuevas cuentas, pero sólo los administradores pueden borrarlas. Si existe seguridad a nivel de datos, la prueba garantiza que un usuario “típico” 1 puede ver toda la información de clientes, incluyendo datos financieros; sin embargo, el usuario 2 solamente puede ver los datos institucionales del mismo cliente.  Las pruebas de seguridad del sistema garantizan que solamente aquellos usuarios autorizados a acceder al sistema son capaces de ejecutar las funciones del sistema a través de los mecanismos apropiados.  Debido a la creciente preocupación de la sociedad por la privacidad de la información, muchos programas tienen objetivos específicos de seguridad.  El objetivo de esta prueba es evaluar el funcionamiento correcto de los controles de seguridad del sistema para asegurar la integridad y confidencialidad de los datos. El foco principal es probar la vulnerabilidad del sistema frente a accesos o manipulaciones no autorizadas. Una manera de encontrar esos casos de prueba es estudiar problemas conocidos de seguridad en sistemas similares y tratar de mostrar la existencia de problemas parecidos en el sistema que se examina.  Algunas consideraciones de prueba son:   * Controles de acceso físico * Acceso a estructuras de datos específicas a través de los programas de aplicación. * Seguridad en sitios remotos * Existencia de datos confidenciales en reportes y pantallas * Controles manuales, incluyendo aquellos para autorización y aprobación, formularios, documentación numerada, transmisión de datos, balances y conversión de datos. * Controles automáticos, incluyendo aquellos para edición de datos, chequeo de máquinas, errores del operador, acceso a datos elementales y archivos, acceso a funciones, auditoría, entre otros. |
| **Técnica:** | Funciones / Seguridad de Datos: Identificar cada tipo de usuario y las funciones y datos a los que se debe autorizar.  Crear pruebas para cada tipo de usuario y verificar cada permiso, creando transacciones específicas para cada tipo de usuario.  Modificar tipos de usuarios y volver a ejecutar las pruebas. En cada caso, verificar si los datos o funciones adicionales quedan correctamente permitidos o denegados.  Acceso al sistema (ver consideraciones especiales) |
| **Criterio de Completitud:** | Para cada tipo de usuario conocido, las funciones y datos apropiados y todas las transacciones funcionan como se esperaba. |
| **Consideraciones Especiales:** | El acceso al sistema debe ser revisado y discutido con los administradores de la red y/o del sistema. Esta prueba puede no ser requerida como tal, sino como una función de los administradores de red o del sistema. |

**PRUEBAS DE VALIDACIÓN A SISTEMAS A LA MEDIDA**

**Pruebas del Ciclo del Negocio**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Asegurar que el sistema funciona de acuerdo con el modelo de negocios emulando todos los eventos en el tiempo y en función del tiempo. |
| **Descripción de la Prueba:** | Las pruebas del ciclo de negocio deberían emular las actividades ejecutadas en el a través del tiempo. Debería identificarse un periodo, como por ejemplo un año, y las transacciones y actividades que podrían ocurrir durante un periodo de un año deberían ejecutarse. Incluyendo todos los ciclos y eventos diarios, semanales y mensuales que sean datos sensitivos, como las agendas. |
| **Técnica:** | Ejecute cada caso de uso, flujo básico o función utilizando datos válidos e inválidos, para verificar que:  Incremente el número de veces en que una función es ejecutada para simular diferentes usuarios sobre un periodo especificado  Todas las fechas o funciones que involucren tiempos serán probadas con datos válidos e inválidos de fechas o periodos de tiempo.  Todas las funciones ocurren en un periodo de tiempo serán ejecutadas en el tiempo apropiado.  Los resultados esperados ocurren cuando los datos válidos son usados.  Los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido.  Cada regla de negocios es aplicada adecuadamente. |
| **Criterio de Completitud:** | Todas las pruebas planeadas han sido ejecutadas.  Todos los defectos que se identificaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | Las fechas y eventos del sistema pueden requerir actividades especiales de soporte.  Se requiere un modelo de negocios para identificar requisitos y procedimientos de prueba apropiados. |

**Pruebas de GUI**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Verifica lo siguiente:   * La navegación a través de los objetos de la prueba reflejan las funcionalidades del negocio y requisitos, se realiza una navegación ventana por ventana, usando los modos de acceso (tabuladores, movimientos del mouse, teclas rápidas, etc) * Los objetos de la ventana y características, tales como menús, medidas, posiciones, estados y focos se verifican conforme a los estándares. |
| **Descripción de la Prueba:** | La prueba de interfaz de usuario verifica la interacción del usuario con el software. El objetivo es asegurar que la interfaz tiene apropiada navegación a través de las diferentes funcionalidades. Adicionalmente, las pruebas de interfaz aseguran que los objetos de la interfaz a ser probada se encuentra dentro de los estandares de la industria |
| **Técnica:** | Pruebas de crear / modificar cada ventana para verificar la adecuada navegación y estado de los objetos. |
| **Criterio de Completitud:** | Cada ventana elegida será totalmente verificada y comparada con similares en el mercado logrando una buena aceptación dentro del estándar. |
| **Consideraciones Especiales:** | Ninguna |

**Pruebas de Configuración**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Validar y verificar que el cliente del sistema funciona apropiadamente en las estaciones de trabajo recomendadas. |
| **Descripción de la Prueba:** | Estas pruebas verifican la operación del sistema en diferentes configuraciones de hardware y software. En la mayoría de los ambientes de producción, las especificaciones para las estaciones de trabajo, equipos de red y servidores pueden variar. Las estaciones pueden tener diferentes versiones de software instaladas (Sistemas Operativos, Drivers, etc) y en cualquier momento, pueden llegar a utilizarse diferentes combinaciones.  Con frecuencia, el número de configuraciones posibles es demasiado grande para intentar una prueba de cada una de ellas, pero el programa debe probarse al menos con cada tipo de dispositivo y con las configuraciones mínima y máxima posibles. |
| **Técnica:** | Use los scripts para Integración y Pruebas del Sistema.  Incluya la apertura o cierre de varias aplicaciones Microsoft, como Excel y Word (o algun tipo de software similar a la que se esta probando ) como una parte de la prueba, ya sea al comienzo o en algún momento intermedio.  Ejecute algunas transacciones para simular actividades cotidianas del usuario, dentro y fuera de las aplicaciones que interactúan con la Base.  Repita estos pasos, minimizando la cantidad de memoria convencional disponible en los clientes. |
| **Criterio de Completitud:** | Para cada combinación de aplicaciones que interactúan con la Base de datos a probar, las transacciones deben ser ejecutadas sin fallas. |
| **Consideraciones Especiales:** | Qué aplicaciones están disponibles para los clientes?  Qué aplicaciones se utilizan normalmente?  Qué tipos de datos manejan estas aplicaciones? (ej. Una larga hoja de cálculo, o un documento de 100 pág. En Word.)  Los sistemas, software de red, servidores, bases de datos también deben ser incluidas como parte de estas pruebas. |

**Prueba de Estilo**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Comprobar que la aplicación sigue los estándares de estilo propios del cliente. |
| **Descripción de la Prueba:** | Se entienden como tales el formato de las ventanas, colores corporativos, tipos de letra etc. |
| **Técnica:** | Se realiza una navegación por la aplicación verificando si se cumplen con los estándares de GUI del cliente.  Validar objetos gráficos contra el manual de estilos del cliente. |
| **Criterio de Completitud:** | Todas las pruebas planeadas han sido ejecutadas.  Todos los defectos que se identificaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | * Solicitar al cliente el manual de estilos, en caso de no existir, hacer un levantamiento preliminar de este con base en la información corporativa existente. |

**Prueba de Aceptación**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Determinación por parte del cliente de la aceptación o rechazo del sistema desarrollado. |
| **Descripción de la Prueba:** | La prueba de aceptación es ejecutada antes de que la aplicación sea instalada dentro de un ambiente de producción. La prueba de aceptación es generalmente desarrollada y ejecutada por el cliente o un especialista de la aplicación y es conducida a determinar como el sistema satisface sus criterios de aceptación validando los requisitos que han sido levantados para el desarrollo, incluyendo a documentación y procesos de negocio.  Basado en esta prueba el cliente determina si acepta o rechaza el sistema  Estas pruebas están destinadas a probar que el producto está listo para el uso operativo. Suelen ser un subconjunto de las Pruebas de Sistema.  Sirve para que el usuario pueda validar si el producto final se ajusta a los requisitos fijados, es decir, si el producto está listo para ser implantado para el uso operativo en el entorno del usuario.  Esta prueba es complementada por la prueba de estilo. |
| **Técnica:** | Realización de los documentos de planes de prueba de aceptación y especificación de los mismos, basados en los criterios de aceptación del cliente.  Los casos prueba de aceptación han de ser planificados, organizados y formalizados de manera que se determine el cumplimiento de los requisitos del sistema. Para la realización de estas pruebas se necesita disponer de los siguientes documentos:   * Especificación de requisitos del sistema. * Manual de usuario. * Manual de administrador.   Realizar Pruebas de estilo |
| **Criterio de Completitud:** | Todas las pruebas planeadas han sido ejecutadas.  Todos los defectos que se identificaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | Las Pruebas de Aceptación se suelen realizar en un entorno de pre-producción. |

**Prueba de Instalación**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Verificar y validar que el sistema se instala apropiadamente en cada cliente, bajo las siguientes condiciones:  Instalaciones nuevas, nuevas máquinas a las que nunca se les ha instalado el sistema.  Actualizar máquinas previamente instaladas con el sistema.  Instalar versiones viejas en máquinas previamente instaladas con el sistema. |
| **Descripción de la Prueba:** | Las pruebas de instalación tienen dos propósitos. El primero es asegurar que el sistema puede ser instalado en todas las configuraciones posibles, tales como nuevas instalaciones, actualizaciones, instalaciones completas o personalizadas, y bajo condiciones normales o anormales; estas últimas incluyen insuficiente espacio en disco, falta de privilegios para algunas tareas, etc.  El segundo propósito es verificar que, una vez instalado, el sistema opera correctamente. Esto usualmente implica correr un número significativo de pruebas de Funcionalidad. |
| **Técnica:** | Diseñar sripts para validar las condiciones de la máquina a instalar .  Realizar la instalación |
| **Criterio de Completitud:** | Las transacciones de la aplicación se ejecutan sin fallas. |
| **Consideraciones Especiales:** | Qué transacciones del sistema se deben seleccionar para realizar una prueba confiable de que el sistema ha sido instalado exitosamente y no hace falta ningún componente del sistema? |

**Pruebas Funcionales**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Se asegura la trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados |
| **Descripción de la Prueba:** | Las pruebas Funcionales deben enfocarse en los requisitos funcionales, las pruebas pueden estar basadas directamente en los Casos de Uso (o funciones de negocio), y las reglas del negocio. Las metas de estas pruebas son:   * Verificar la apropiada aceptación de datos, * Verificar el procesamiento y recuperación y la implementación adecuada de las reglas del negocio.   Este tipo de pruebas están basadas en técnicas de caja negra, que es, verificar la aplicación (y sus procesos internos) mediante la interacción con la aplicación vía GUI y analizar la salida (resultados). Lo que se identifica a continuación es un diseño preliminar de las pruebas recomendadas para cada aplicación. |
| **Técnica:** | Se ejecuta cada caso de uso, flujo de caso de uso, o función, usando datos válidos e inválidos, para verificar lo siguiente:   * Que los resultados esperados ocurran cuando se usen datos válidos. * Que sean desplegados los mensajes apropiados de error y precaución cuando se usan datos inválidos. * Que se aplique apropiadamente cada regla de negocio. |
| **Criterio de Completitud:** | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identificaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | Identifique o describa aquellos aspectos (internos o externos) que impactan la implementación y ejecución de las pruebas de funcionalidad. |

**Prueba de Documentación Y Procedimiento**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Evaluar la documentación del usuario |
| **Descripción de la Prueba:** | Evaluar la exactitud y claridad de la documentación del usuario y para determinar si el manual de procedimientos trabajará correctamente como una parte integral del sistema.  Muchos defectos son identificados cuando un probador competente chequea totalmente los manuales y documentación del usuario.  Muchos programas son parte de sistemas mayores, no completamente automatizados, que contienen procedimientos realizados por operadores. Cualquier procedimiento humano, tal como los que llevan a cabo el operador, el administrador de la base de datos, o el usuario de terminal, debe ser probado durante la prueba de sistemas. |
| **Técnica:** | Revisar la documentación del proyecto contra las funcionalidades del sistema y su configuración física. |
| **Criterio de Completitud:** | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identificaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | Ninguna. |

**Prueba de Usabilidad**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Determinar la usabilidad del sistema. |
| **Descripción de la Prueba:** | Determina cuán bien el usuario podrá usar y entender la aplicación. Identifica las áreas de diseño que hacen al sistema de difícil uso para el usuario.  La prueba de usabilidad detecta problemas relacionados con la conveniencia y practicidad del sistema desde el punto de vista del usuario. |
| **Técnica:** | Verificar que la aplicación no presenta los siguientes problemas de usabilidad típicos:   * El sistema es demasiado complejo y difícil de usar. * Es difícil instalar y entender el sistema * La recuperación de errores es pobre y los mensajes de error no tienen significado * La sintaxis de los comandos es difícil de aprender y recordar * El sistema obliga al usuario a recordar formatos y secuencias fijas * Los procedimientos no son simples ni obvios * El sistema no tiene instrucciones de ayuda por computadora y tiene manuales pobres. * Los diagramas, pantallas, reportes y gráficos son de calidad y apariencia pobre * El sistema carece de herramientas de construcción adecuadas y requiere múltiples comandos * La lógica y conveniencia de los botones, switches, displays y mensajes de ayuda deben ser testeados. (La prueba de usabilidad puede ser conducida por un grupo separado si es posible.   Se deben crear casos de prueba para comprobar que se puede operar en el sistema de forma adecuada. |
| **Criterio de Completitud:** | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identificaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | Ninguna |

**Prueba de Campo**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Correr el sistema en el ambiente real para encontrar errores y validar el producto contra sus especificaciones originales. |
| **Descripción de la Prueba:** | Realizar un subconjunto válido de pruebas de sistema. |
| **Técnica:** | Determinar que pruebas de sistema serán corridas para validar el sistema en producción. |
| **Criterio de Completitud:** | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identificaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | Ninguna |

**PRUEBAS DE VALIDACIÓN A APLICACIONES GENÉRICAS**

**Pruebas Alfa**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Prueba de aceptación para detectar errores en el sistema bajo un ambiente controlado. |
| **Descripción de la Prueba:** | La verificación involucra la ejecución de partes o todo del sistema en ambientes simulados, con el fin de encontrar errores.  La retroalimentación de esta fase produce cambios en el software para resolver los errores y fallas que se descubren. |
| **Técnica:** | Realizar las pruebas de sistema bajo las siguientes características:   * se llevan a cabo en el lugar en donde fue desarrollado el sw, * en un ambiente controlado, en el cual el desarrollador está presente.   Se selecciona un grupo de usuarios para el alpha test y se les pide trabajen con el sistema como parte de las pruebas. |
| **Criterio de Completitud:** | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identificaron han sido tenidos en cuenta. |
| **Consideraciones Especiales:** | Ninguna |

**Pruebas Beta**

|  |  |
| --- | --- |
| **Objetivo de la Prueba:** | Realizar la validación del sistema por parte del usuario. |
| **Descripción de la Prueba:** | Prueba de aceptación donde La validación (o pruebas beta) involucra el uso del software en un ambiente real. |
| **Técnica:** | Se selecciona un grupo de usuarios que ponen a trabajar el sistema en un ambiente real. Usan el sistema en sus actividades cotidianas, procesan transacciones y producen salidas normales del sistema.  Las transacciones y personas que usan el sistema son reales y trabajan en su área de trabajo real.  El desarrollador no esta presente.  Los usuarios están advertidos de que están usando un sistema que puede fallar.  Los usuarios realizan pruebas a su antojo realizando uso de la aplicación. |
| **Criterio de Completitud:** | Se establece un periodo de pruebas beta en el que los errores detectados no sean de carácter crítico para el sistema. |
| **Consideraciones Especiales:** | Se deben considerar mecanismos de comunicación entre los desarrolladores y los usuarios de manera que los errores detectados puedan ser corregidos. |