**Plan de pruebas de software**

***[Nombre del proyecto]***

***Fecha: [dd/mm/aaa]***

[Esta plantilla tiene por finalidad servir de base para la confección del documento “Plan de Pruebas”. El texto entre paréntesis cuadrados y desplegado en azul itálico (estilo = InfoBlue) tiene por finalidad guiar al autor y debe ser borrado antes de la publicación del documento.]

Historia de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| <aaaa-mm-dd> | <1.1.0> | Documento inicial | <Nombre> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabla de Contenidos

1 Introducción 6

1.1 Resumen ejecutivo 6

1.2 Alcance de las pruebas 6

1.2.1 Elementos de pruebas 6

1.2.2 Pruebas funcionales 6

1.2.3 Riesgos 6

1.3 Definiciones, acrónimos y abreviaciones 8

1.4 Referencias 8

2 Requerimientos para pruebas 9

2.1 Casos de uso 9

2.1.1 Vista global 9

2.1.2 Caso de uso 1 9

2.1.3 Caso de uso 2 9

2.1.4 Caso de uso 3 9

2.2 Requerimientos funcionales 9

2.2.1 Componentes comunes 9

2.2.2 Componente 1 9

2.2.3 Componente 2 9

2.3 Requerimientos No-Funcionales (Matriz con Input/Output, solo si aplica) 9

2.3.1 Componente 1 (puede tener uno o más casos de uso) 9

2.3.2 Componente 2 (puede tener uno o más casos de uso) 9

3 Estrategia de Pruebas 10

3.1 Tipos de pruebas 10

3.1.1 Pruebas funcionales 10

3.1.2 Pruebas de rendimiento (Performance) 11

3.1.3 Pruebas de seguridad y de acceso a datos 11

3.1.4 Herramientas involucradas 11

4 Recursos 12

4.1 Profesionales 12

4.2 Ambiente de pruebas 13

4.2.1 Preparación del ambiente de pruebas 13

4.2.2 Diseño del ambiente de pruebas 13

4.2.3 Diseño ambiente de pruebas 15

4.2.4 Integración del ambiente de pruebas y configuración 16

4.2.5 Generación de datos 16

5 Actividades e Hitos del Plan de Pruebas 16

6 Entregables 19

6.1 Plan de pruebas 19

6.1.1 Criterio de entrada para el “Plan de pruebas” 19

6.1.2 Criterio de salida para el “Plan de pruebas” 19

6.1.3 Criterio de suspensión y resumisión 19

6.2 Resultados de las pruebas 19

6.3 Reporte de defectos 19

7 Anexos 20

7.1 A: Tareas del proyecto 20

7.2 B: Pruebas de rendimiento (*performance*) 21

7.3 C: Pruebas de seguridad y de control de acceso 22

# Introducción

## Resumen ejecutivo

*Resumen de todo el contenido del plan de pruebas de software, describe cuál es su propósito, establece si es un plan maestro o un plan detallado, identifica el alcance del plan de pruebas en relación con el plan de proyecto de software, restricciones (por ejemplo de recursos o presupuesto), alcance del esfuerzo de pruebas entre otros aspectos.*

## Alcance de las pruebas

### Elementos de pruebas

Listado de todos los módulos, componentes o elementos que se van a probar. Si es de alto nivel, se listan las áreas funcionales (módulos o procesos que cubre el Testing), por otro lado, si es de un nivel detallado se listan los programas, unidades o módulos.

### Pruebas funcionales

*El testing señalado es del tipo “Caja Negra”. Este tipo de pruebas busca comprobar el comportamiento de un componente, sin inspeccionar sus detalles internos.*

*Con esta finalidad se utilizan datos de entrada (input), se ejecuta un componente de software y se obtiene un resultado (output).*

Los datos de entrada son los utilizados por las transacciones involucradas. Cada argumento de entrada puede seleccionar uno de los siguientes datos de prueba, dependiendo este del resultado que se desea obtener (esperado), verificando así el comportamiento del componente a probar usando distintos valores de entrada:

* Valores normales para cada transacción.
* Valores límites para cada transacción.
* Valores de borde.
* Valores ilegales.

### Riesgos

Algunos riesgos comunes a considerar son:

1. Documentación de especificación errónea o incompleta.
2. Lista de requerimientos inconsistente con los casos de uso.
3. Componentes a probar y componentes comunes correspondan a distintas versiones.
4. Hardware y software no funcionen correctamente.
5. Herramientas de *testing* automatizado estén mal configuradas.

Los riesgos serán identificados de acuerdo a un concepto de Bajo, Medio o Alto, dependiendo de la importancia del caso de uso para el cual se está desarrollando el *testing*.

Así mismo, se han identificado una serie de riesgos (calificados entre 1 y 10 dependiendo de su gravedad) los que están detallados en el artefacto “Lista de riesgos” con sus alcances y acciones, en el presente plan, son enunciados para sugerir algunas acciones.

#### Matrices de riesgos

##### Pruebas

| **Nº Riesgo** | **Descripción** | **Gravedad** | **Acción** |
| --- | --- | --- | --- |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |

##### 

## Definiciones, acrónimos y abreviaciones

[Esta subsección provee las definiciones de todos los términos, acrónimos, y abreviaciones requeridas para interpretar correctamente el documento. Esta información puede entregarse a modo de referencia a documento “Glosario” del proyecto.]

## Referencias

[Esta subsección provee una lista completa de todos los documentos referenciados en cualquier parte de este artefacto. Cada documento debe ser identificado por título, edición/versión, fecha, autor y nombre de archivo. Especificar las fuentes de donde se pueden obtener estas referencias. Esta información puede ser entregada como referencia a un apéndice o a otro documento.]

# Requerimientos para pruebas

La siguiente lista identifica los ítems (casos de uso, requerimientos funcionales y requerimientos no funcionales) que se han identificado como requerimientos a ser probados.

## Casos de uso

### Vista global

[Agregar o eliminar según corresponda]

### Caso de uso 1

### Caso de uso 2

### Caso de uso 3

## Requerimientos funcionales

### Componentes comunes

[Agregar o eliminar según corresponda]

### Componente 1

### Componente 2

## Requerimientos No-Funcionales (Matriz con Input/Output, solo si aplica)

### Componente 1 (puede tener uno o más casos de uso)

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 |  |  |  |

### Componente 2 (puede tener uno o más casos de uso)

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
|  |  |  |  |

# Estrategia de Pruebas

[A continuación se describe la estrategia de testing a ser utilizada, en otras palabras, se indica cómo se realizará el testing de los requerimientos analizados en la sección precedente (Requerimientos para pruebas) que identifica qué se va a testear.]

*[Se proponen varios tipos de pruebas, de los cuales la Organización aplicará los que considere necesarios y suficientes.]*

## Tipos de pruebas

### Pruebas funcionales

El *testing* funcional se realizará sobre los requerimientos funcionales antes descritos y sus casos de uso. Estas pruebas tienen por finalidad comprobar la funcionalidad de la aplicación a partir de datos válidamente seleccionados sobre las transacciones del sistema.

Este tipo de comprobación se basa en las técnicas de caja negra, que permiten probar la aplicación (y sus procesos interinos) vía GUI.

|  |  |
| --- | --- |
| Objetivo de la prueba: | Asegurar la funcionalidad del conjunto de casos, incluyendo la navegación en la aplicación, el ingreso de datos, el proceso y la recuperación (resultados).  Que la navegación a través de los casos de prueba refleje apropiadamente las reglas del negocio y los requerimientos, incluyendo ventana a ventana, campo a campo y usando los métodos de acceso correctamente (tecla *tab*, movimiento del *mouse*, etc.)  Que los objetos de las ventanas y sus características, tales como menús, tamaño, posición, estados y foco, estén de acuerdo a los estándares. |
| Técnica a utilizar: | Ejecutar cada caso de uso, su flujo y funcionalidad usando tanto datos válidos como inválidos para comprobar lo siguiente:   * Que los resultados esperados ocurren cuando los datos válidos son utilizados. * Que el mensaje de error es apropiado cuando se utilizan datos inválidos. * Que cada regla de negocio se utiliza apropiadamente. * Crear y modificar los procedimientos de prueba para cada ventana, para comprobar los estados de los objetos y de la aplicación. |
| Criterio de validación: | * Todas las pruebas planificadas se ejecutaron correctamente. * Todos los defectos identificados han sido asignados. * Cada ventana debe ser verificada para mantener la consistencia con la versión maestra y comprobar que esté dentro de los estándares aceptables. |
| Consideraciones especiales: | [Puede que no todas las propiedades sean verificadas, considerar las más importantes y/o definidas y no incluir todos los objetos de terceras partes.] |
| Observaciones: |  |

### Pruebas de rendimiento (Performance)

Un detalle de este tipo de *testing*, adjunta en el Anexo B de este documento.

### Pruebas de seguridad y de acceso a datos

Un detalle de este tipo de *testing*, adjunta en el Anexo C de este documento.

### Herramientas involucradas

[Esta sección es solo opcional y aplica en el caso que se utilicen herramientas de testing.]

*[La siguiente tabla contiene algunos ejemplos.]*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Herramienta** | **Vendedor/Tienda** | **Versión** |
| Guía de Referencia | Rational Unified Process | Rational Software Corp. | 5.5 |
| Administración de Testing | Rational TestManager incluida en el Rational TeamTest | Rational Software Corp. | 7.5 |
| Seguimiento de Defectos | Rational ClearQuest TT Edition incluida en el Rational TeamTest | Rational Software Corp. | 2.0 |
| Testing Funcional | Rational Robot incluido en el Rational TeamTest | Rational Software Corp. | 7.5 |
| ASQ Tool (performance testing) | No definido |  |  |
| Test Coverage Monitor or Profiler | No definido |  |  |
| Herramienta de DBMS | Microsoft Access | Microsoft Software Corp. | 97 |
| SQL Server | Microsoft Software Corp. | 7.0 |

# Recursos

## Profesionales

| **Recursos Humanos** | | |
| --- | --- | --- |
| **Rol** | **Recursos mínimos recomendados**  **(Número de personas full-time)** | **Responsabilidades específicas / Comentarios** |
| Diseñador de casos de prueba |  | Responsabilidades   * Identificar, priorizar e implementar los casos de prueba. * Evaluar de forma el esfuerzo de *testing*. |
| Testeador |  | Responsabilidades:   * Ejecutar los casos de prueba. * Guardar estado de los resultados. * Recuperación de errores. * Generar peticiones de cambios en la documentación. |
| Administrador de sistema del pruebas |  | Responsabilidades   * Administrar el sistema de control de pruebas. * Instalar / administrar el acceso al sistema de pruebas. |
| Administrador de la base de datos / Encargado de la base de datos |  | Responsabilidades:   * Administra los datos del prueba (Base de Datos) * Asegurar que el entorno de datos de prueba (base de datos) y los valores que contiene son controlados y mantenidos. |
| Diseñador |  | Responsabilidades:   * Identificar y definir las operaciones, atributos y asociaciones de las clases de prueba. * Identificar y definir los paquetes de prueba. |
| Implementador |  | Responsabilidades:   * Implementar las clases de prueba y los paquetes de prueba. |

## Ambiente de pruebas

Se identifican los requerimientos de hardware, software y de comunicación necesarios para crear y dar soporte permanente al Ambiente de pruebas. Las actividades de instalación y configuración para el conjunto de los componentes del Ambiente de pruebas, deberán ser planificadas y calendarizadas. Se requiere que este ambiente sea seguro, estable y dedicado exclusivamente para las pruebas del sistema.

[Se puede referenciar el artefacto “Especificación de Ambientes”.]

### Preparación del ambiente de pruebas

Las pruebas unitarias y de regresión deberán ser ejecutadas dentro del Ambiente de desarrollo, las pruebas de aceptación del usuario y del sistema se ejecutarán en este Ambiente de pruebas. Este ambiente deberá representar una configuración idéntica al Ambiente de producción o al menos, una versión en menor escala. Esto se requiere debido a que se debe replicar el rendimiento de la línea base y las medidas de mejoramiento relacionadas.

### Diseño del ambiente de pruebas

*[La siguiente tabla corresponde a un ejemplo de identificación de los recursos de sistema necesarios para el testing.]*

| **ITEM** | **DESCRIPCIÓN** | **OBSERVACIONES** |
| --- | --- | --- |
| **HARDWARE** |  |  |
| **Estación de Pruebas (Cliente)** |  |  |
| Procesador | *Pentium II 300 MHz o superior* |  |
| Memoria RAM | *64 MB (128 MB ideal)* |  |
| Espacio en Disco | *600 MB* |  |
| Tipo Monitor y Resolución | *SVGA .28 I 800x600* |  |
| Unidad de Disquete | *1.44 MB* |  |
| Tarjeta de red |  |  |
| Modem |  |  |
| Mouse |  |  |
| Tipo Enlace |  |  |
| **Servidores** |  |  |
| ***Base de Datos (pruebas)*** |  |  |
| Procesador |  |  |
| Memoria RAM |  |  |
| Espacio en Disco |  |  |
| Tipo Monitor y Resolución |  |  |
| Unidad de Disquete |  |  |
| Tarjeta de red |  |  |
| Modem |  |  |
| Mouse |  |  |
| Tipo Enlace |  |  |
| ***Aplicación (pruebas)*** |  |  |
| Procesador |  |  |
| Memoria RAM |  |  |
| Espacio en Disco |  |  |
| Tipo Monitor y Resolución |  |  |
| Unidad de Disquete |  |  |
| Tarjeta de red |  |  |
| Módem |  |  |
| Mouse |  |  |
| Tipo Enlace |  |  |
| **Impresoras** |  |  |
| Marca y Modelo |  |  |
| Tipo |  |  |
| Resolución |  |  |
| Rendimiento |  |  |
| Dedicación |  |  |
| **RED** |  |  |
| Topología |  |  |
| Medio |  |  |
| Velocidad |  |  |
| Protocolo |  |  |
| Módems |  |  |
| Conexión Internet |  |  |
| **Sistema de Respaldo / Restauración** |  |  |
| Unidad (Modelo y Marca) |  |  |
| Capacidad |  |  |
| Ubicación |  |  |
|  |  |  |
| **SOFTWARE** |  |  |
| **Estación de Pruebas (Cliente)** |  |  |
| Sistema Operativo |  |  |
| Herramienta de *testing* | *Rational TeamTest compuesta por Robot, ClearQuest TT, Test Manager* |  |
| Herramienta de Modelamiento | *Rational Rose 2000* |  |
| BDMA | *Access* |  |
| Browser | *Microsoft Explorer* |  |
| Software de Escritorio | *Microsoft Office* |  |
| **Base de Datos (pruebas)** |  |  |
| Sistema Operativo |  |  |
| Software de Red |  |  |
| Dominio/Cuenta |  |  |
|  |  |  |
| **Aplicación (pruebas)** |  |  |
| Sistema Operativo |  |  |
| Software de Red |  |  |
| Dominio/Cuenta |  |  |
|  |  |  |
| **Repositorios** |  |  |
| Servidor |  |  |
| Dominio / Cuenta |  |  |
|  |  |  |
| **Seguridad** |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### Diseño ambiente de pruebas

El siguiente diagrama muestra la arquitectura del Ambiente de pruebas requerido para realizar las pruebas. La arquitectura del ambiente de pruebas debe ser, en la medida de lo posible, similar a la arquitectura definida para el sistema en producción.

*<Incluir diagrama de la arquitectura del Ambiente de pruebas>*

### Integración del ambiente de pruebas y configuración

Para esta actividad se requerirá la participación de profesionales de *<nombre de cliente>* en cuanto a instalación, configuración y puesta en marcha del Ambiente de Pruebas. Principalmente se requiere del responsable de la Red y Administración de Bases de Datos, de tal forma de obtener un ambiente lo más consistente y similar al de producción, con las bases de datos creadas y el software configurado para asegurar que el sistema funciona de acuerdo a diseño.

Las actividades generales a ser consideradas son:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Actividad** | **Responsable** | **Fecha Estimada** | **Fecha Real** | **Observaciones** |
|  |  |  |  |  |

### Generación de datos

La generación de los datos para las pruebas consideran los siguientes aspectos, que se deben definir de acuerdo a los requerimientos y posibilidades de obtención. Los aspectos que se describen a continuación, buscan que los datos sean los correctos y que cubran todos los riesgos y situaciones necesarias.

#### Muestra de producción

Para que la muestra de datos sea realmente representativa, se deberá elegir una fecha testigo adecuada y que posibilite la mayor cobertura de datos. En este sentido se ha elegido como fecha testigo el

*<Indicar fecha testigo para obtención de datos>*

Para la obtención de datos por esta vía, se deberán definir las restricciones (por motivos de confidencialidad) y generar algún utilitario para filtrar los datos de tal forma de obtener la mayor variabilidad de datos posible.

Además se deben considerar los siguientes aspectos para asegurar que estos datos funcionen correctamente en el Ambiente de Pruebas, si corresponde:

* Archivos Maestros al Inicio del Día
* Tablas de Parámetros
* Interfaces de Entrada
* Archivos de Movimientos del día o del periodo

Todos estos aspectos se deben considerar en los distintos ambientes donde los datos van a ser utilizados en las transacciones o actualizaciones.

# Actividades e Hitos del Plan de Pruebas

[A continuación se enumeran las actividades a realizar.]

|  | **Tarea** | **Responsable** | **Esfuerzo** | **Fecha Inicio** | **Fecha Término** |
| --- | --- | --- | --- | --- | --- |
|  | **Plan de Pruebas** |  |  |  |  |
|  | Identificar el Proyecto |  |  |  |  |
|  | Definir Estrategia |  |  |  |  |
|  | Estimar Actividades |  |  |  |  |
|  | Identificar Recursos |  |  |  |  |
|  | Documentar el “Plan de Pruebas” |  |  |  |  |
|  | Agendar de Actividades |  |  |  |  |
|  | Revisar el “Plan de Pruebas” |  |  |  |  |
|  | **Diseño de las Pruebas** |  |  |  |  |
|  | Analizar Requerimientos |  |  |  |  |
|  | Especificar Procedimientos de prueba |  |  |  |  |
|  | Especificar casos de prueba |  |  |  |  |
|  | Revisar Cobertura de los requerimientos de prueba |  |  |  |  |
|  | **Implementación de las Pruebas** |  |  |  |  |
|  | Establecer Ambiente de Implementación |  |  |  |  |
|  | Desarrollar los Procedimientos de Prueba |  |  |  |  |
|  | Probar y depurar los procedimientos de prueba |  |  |  |  |
|  | Modificar los procedimientos de prueba |  |  |  |  |
|  |  |  |  |  |  |
|  | **Ejecución de las pruebas** |  |  |  |  |
|  | Ejecutar pruebas |  |  |  |  |
|  | Comprobar resultados esperados |  |  |  |  |
|  | Investigar resultados inesperados |  |  |  |  |
|  | Registrar defectos (log) |  |  |  |  |
|  | Re-Ejecutar las pruebas |  |  |  |  |
|  | **Evaluación de las pruebas** |  |  |  |  |
|  | Revisar el Log de pruebas |  |  |  |  |
|  | Evaluar cobertura de los casos de prueba |  |  |  |  |
|  | Evaluar defectos |  |  |  |  |
|  | Reportar defectos |  |  |  |  |

# Entregables

[En esta sección se deben listar los documentos, herramientas, y reportes que serán creados con autor, destinatario y fecha de entrega.]

[La documentación de casos de prueba y reportes puede hacerse con los artefactos “Planilla de Casos de Prueba” y “Sumario de Evaluación de Pruebas”.]

## Plan de pruebas

[Esta sección identifica los reportes del “Plan de pruebas” que serán creados y distribuidos.]

### Criterio de entrada para el “Plan de pruebas”

*[Especifica el criterio que se usará para determinar si la ejecución del “Plan de pruebas” puede comenzar.]*

### Criterio de salida para el “Plan de pruebas”

*[Especifica el criterio que se usará para determinar si la ejecución del “Plan de pruebas” se ha completado o si continua en ejecución.]*

### Criterio de suspensión y resumisión

[Especifica el criterio que se usará para determinar si la ejecución del “Plan de pruebas” debe ser suspendida prematuramente o finalizada antes de ser completada, y bajo qué criterio puede ser reasumido.]

## Resultados de las pruebas

[Describe los métodos y herramientas usadas para registrar y reportar los resultados y el estado de la ejecución del “Plan de pruebas”.]

## Reporte de defectos

[Define los métodos y herramienta usados para registrar, ajustar y reportar los incidentes en la ejecución del “Plan de pruebas”.]

# Anexos

## A: Tareas del proyecto

La siguiente lista muestra las tareas relacionadas con el “Plan de pruebas”:

|  |  |
| --- | --- |
| *✓* | “Plan de pruebas” (preliminar, al inicio del proyecto) |
|  | Identificar requerimientos para el testing |
|  | Identificar los riesgos, cuantificar impacto |
|  | Desarrollar la estrategia de pruebas |
|  | Identificar los recursos para las pruebas |
|  | Generar “Plan de Pruebas” detallado |
|  | Diseño general de las pruebas |
|  | Análisis de carga |
|  | Identificar y describir los casos de prueba |
|  | Identificar y estructurar los procedimientos de prueba |
|  | Revisar y accesar la cobertura de las pruebas |
|  | Implementar las pruebas |
|  | Grabar o programar los *scripts* de las pruebas, si aplica |
|  | Identificar las funcionalidades a probar, específicos en el modelo de diseño e implementación |
|  | Establecer el conjunto de datos externos |
|  | Ejecutar las pruebas |
|  | Ejecutar los procedimientos de prueba |
|  | Evaluar la ejecución de las pruebas |
|  | Comprobar los resultados |
|  | Investigar los resultados inesperados |
|  | Registro de defectos, Informe de Resultados |
|  | Evaluar las pruebas |
|  | Evaluar la cobertura de los casos de prueba |
|  | Evaluar la cobertura del código |
|  | Analizar defectos |

## B: Pruebas de rendimiento (*performance*)

[El testing de rendimiento es una prueba para comprobar los tiempos de respuestas, la tasa de transacciones y otros tiempos requeridos de desempeño. Su es comprobar que los tiempos de respuestas requeridos son los óptimos, y que el desempeño del sistema y la configuración del hardware son adecuados frente a una serie de transacciones bajo ciertas condiciones de carga.]

Para esto, se definen las transacciones de acuerdo a los casos de uso específicos que se espera que un actor del sistema realice usando un conjunto de datos para agregar o modificar transacciones.

|  |  |
| --- | --- |
| Objetivo de la prueba: | Comprobar la conducta de rendimiento para las transacciones seleccionadas o funcionalidades bajo las siguientes condiciones:  - Una carga de trabajo normal.  - Una sobrecarga de trabajo. |
| Técnica a usar: | Usar los procedimientos de pruebas desarrollados para el testing funcional.  Modificar los archivos de datos para aumentar las transacciones o los *script* de robotización para incrementar el número de iteraciones de cada transacción.  Los *script* deberán correr en una máquina (la mejor referencia es un solo usuario y una única transacción) y repetirla con múltiples clientes (virtuales o reales). |
| Criterio de validación: | Una Transacción / Un Usuario: La finalización exitosa de los *scripts* de prueba sin ninguna falla dentro del tiempo esperado (por transacción en forma independiente).  Múltiples Transacciones / Múltiples Usuarios: La finalización exitosa de los *scripts* de prueba sin ninguna falla dentro tiempo estimado. |
| Consideraciones especiales: | La extensión del testing de rendimiento requiere tener en *background* la carga de trabajo en el servidor.  Existen varios métodos que se pueden usar para realizar esto como por ejemplo:  Gatillar transacciones directamente al servidor, normalmente en forma de llamadas de SQL.  Crear una carga de usuarios virtuales para simular (normalmente varios cientos) los clientes. Para esto se utilizan herramientas de emulación de terminales remotas para lograr esta carga. Esta técnica también puede usarse para someter a la red a un alto tráfico.  Usar múltiples clientes físicos, cada uno corriendo los Test scripts para agregar una carga al sistema.  El testing de rendimiento debería realizarse en una máquina dedicada o en un tiempo dedicado. Esto permite un control total y una exacta medición.  Las bases de datos utilizadas para realizar el testing de rendimiento deberán ser del tamaño equivalente a las de producción o a escala similar. |
| Observaciones: |  |

## C: Pruebas de seguridad y de control de acceso

Se recomienda que el Administrador de la Red y del Sistema planifiquen algunas pruebas en este sentido.

Este *testing* se enfoca en dos áreas claves de la seguridad:

* Seguridad a nivel de la Aplicación, incluyendo acceso a los datos o funciones de negocio, y
* Seguridad a nivel del Sistema, incluyendo el autenticación (*login*) y/o acceso remoto al sistema.
* La seguridad a nivel de la aplicación, asegura que, sobre la base de la seguridad deseada, se restringen a los usuarios a ciertas funciones o casos de uso específicos o se les limita el acceso a datos disponible para ellos.
* La seguridad a nivel de sistema, asegura que sólo los usuarios definidos en el sistema son capaces de acceder a la aplicación y sólo a través de entradas apropiadas.

|  |  |
| --- | --- |
| Objetivo de la prueba: | Seguridad a Nivel de Aplicación: comprobar que un usuario puede acceder sólo a las funcionalidades y datos para las cuales ese tipo de usuario tiene permiso.  Seguridad a Nivel de Sistema: comprobar que sólo esos usuarios con acceso al sistema y aplicación tienen permitido el acceso. |
| Técnica a usar: | Nivel de Aplicación: Identifique y liste cada tipo de usuario y las funcionalidades y datos de cada tipo para las cuales tiene permiso.  Cree pruebas para cada tipo de usuario y verifique cada permiso creando transacciones específicas para cada usuario.  Modifique los tipos de usuarios y vuelva a ejecutar los casos de prueba para los mismos usuarios. En cada caso verifique si las funcionalidades y los datos están correctamente disponibles o denegados.  Acceso a Nivel de Sistema: vea las consideraciones especiales más abajo. |
| Criterio de validación: | Para cada tipo de usuario conocido, las funcionalidades y los datos correctos debieran estar disponibles y todas las transacciones ejecutadas debieran ejecutarse de acuerdo a lo esperado. |
| Consideraciones especiales: | El acceso al sistema debería ser comprobado con el administrador de la red o del sistema.  Este *testing* quizás pueda requerir de la participación del administrador de la red o del sistema. |
| Observaciones: |  |