# Plan de pruebas de software

***Maipo Grande***

***Fecha: [23/09/2020]***

Integrantes: Miguel Solorza

Patricio Toledo

Ismael Leiva

Héctor Rodriguez

Historia de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 2020-09-23 | 1.0.0 | Documento inicial | Ismael Leiva |
|  | 1.0.1 | Revisión 1 | Patricio Toledo |
|  | 1.1.0 | Revisión 2 | Miguel Solorza |
|  | 2.0.0 | Revisión Final | Patricio Toledo. |

Tabla de Contenidos

Plan de pruebas de software 1

1.1 Resumen ejecutivo 3

1.2 Alcance de las pruebas 3

1.2.1 Elementos de pruebas 3

**1.2.2** **Pruebas funcionales** 3

**1.2.3** **Riesgos** 4

1.3 Definiciones, acrónimos y abreviaciones 5

1.4 Referencias 5

2 Requerimientos para pruebas 5

2.1 Casos de uso 5

**2.1.1** **Vista global** 5

2.2 Requerimientos funcionales 6

**2.2.1** **Componentes comunes** 6

**2.2.2** **Componente 1** 6

**2.2.3** **Componente 2** 6

**2.2.4** **Componente 3** 7

2.3 Requerimientos No-Funcionales 7

3 Estrategia de Pruebas 8

3.1 Tipos de pruebas 8

**3.1.1** **Pruebas funcionales** 8

**3.1.2** **Pruebas de rendimiento (Performance)** 9

**3.1.3** **Pruebas de seguridad y de acceso a datos** 9

3.2 Profesionales 10

3.3 Ambiente de pruebas 11

**3.3.1** **Preparación del ambiente de pruebas** 11

**3.3.2** **Diseño del ambiente de pruebas** 11

**3.3.3** **Diseño ambiente de pruebas** 13

**3.3.4** **Integración del ambiente de pruebas y configuración** 14

**3.3.5** **Generación de datos** 14

4 Actividades e Hitos del Plan de Pruebas 15

5 Entregables 17

5.1 Plan de pruebas 17

5.1.1 Criterio de entrada para el “Plan de pruebas” 17

5.1.2 Criterio de salida para el “Plan de pruebas” 17

5.1.3 Criterio de suspensión y resumisión 17

5.2 Resultados de las pruebas 17

5.3 Reporte de defectos 17

6 Anexos 18

6.1 A: Tareas del proyecto 18

6.2 B: Pruebas de rendimiento (*performance*) 19

6.3 C: Pruebas de seguridad y de control de acceso 20

6.4 D: Diagramas y casos de usos 21

Introducción

## Resumen ejecutivo

El presente documento define el plan maestro de pruebas y la estrategia a seguir para testear la solución propuesta en el ERS presentado para la empresa Maipo Grande, abarcando los casos de uso emanados de aquel análisis previo.

## Alcance de las pruebas

### Elementos de pruebas

Serán objetos de prueba los siguientes componentes y sub componentes del sistema:

|  |  |
| --- | --- |
| **Componente** | **Sub componente** |
| Base de datos | Oracle 11g |
| Lenguaje de Programación | C# sobre .Net Framework  Visual Studio 2015 |
|  |
| Cliente Web | Window 7 o Superior + Chrome |
| Cliente Desktop | Window 7 o Superior + Chrome |

### **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 que correspondan a distintas versiones.
4. Hardware y software con errores de funcionamiento.
5. Herramientas de *testing* automatizado 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 | Desconexión de internet | Baja | Esperar a la reconexión |
| 2 | Perdida o BDD corrompida | Alta | Respaldos automático cada 24hrs |
| 3 | Filtración de seguridad de la API | Alta | Restringir el acceso a la BBDD |
| 4 | Corte de luz en proceso de solicitud de inspección | Media | Cache temporal de los últimos 10 minutos |
| 5 | Aplicación de escritorio se congela | Baja | Reinicio del equipo |

##### 

## Definiciones, acrónimos y abreviaciones

* **CU**: Caso de uso.
* **RF**: Requerimiento funcional.
* **RNF**: Requerimiento no funcional.
* **ERS**: Especificación de Requisitos de Software.
* **PP**: Plan de pruebas.

## Referencias

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Título** | **Versión** | **Fecha** | **Autor** | **Nombre de archivo** |
| *Plan de Integración* | 1.1.0 |  |  |  |
| *Plan de Pruebas* | 1.1.0 |  | Héctor Rodríguez |  |
| *ERS* | 1.1.0 |  | Patricio Toledo |  |
| CU | 1.0.0 |  | Miguel Solorza |  |

# 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**

Se adjunta al final del documento “Diagramas y Especificación de Casos de uso”, en el cual se puede observar tres diagramas de casos de uso por sistema a desarrollar (Administrador, Productor, Cliente, Transportista y Consultor). Además, se encuentran detallados las especificaciones de cada caso de uso de los diagramas en el archivo Descripción\_Casos\_de\_uso\_Maipo\_Grande.docx.

## Requerimientos funcionales

* Crear y mantener un registro de clientes, productores y de sus contratos.
* El sistema debe tener los roles de administración, productor, cliente externo, cliente interno, transportista y consultor, diferenciando las funcionalidades que cada uno tiene de acuerdo a su rol.
* El administrador es el encargado de crear procesos de venta externa y procesos locales, pudiendo actualizar el estado de los procesos en las etapas que le correspondan.
* Los productores deben ingresar la información de sus productos, los cuales deben asociarse a los procesos de venta que se estén llevando a cabo.
* Los administradores deben reportar un resumen de venta al final, vía correo electrónico, cuando esta se concrete.
* Los clientes finales deben ser registrados para darles acceso al sistema.
* Los clientes finales podrán ingresar solicitudes de compra, las cuales, podrán ser vistas por el administrador.
* Los clientes finales solo deben ver los procesos de ventas en que cada uno esté participando.
* Los administradores deben registrar a transportistas y darles accesos a las subastas que están relacionadas a los procesos de venta externo y local.
* El administrador debe publicar subastas para que los transportistas compitan por hacer el transporte de los productos.
* El sistema debe realizar las subastas recibiendo las ofertas de los transportistas y adjudicando, al que terminado el tiempo tenga el precio menor, y cumpla con las características del transporte que se necesita (tamaño, capacidad carga, refrigeración, etc.).
* El administrador debe publicar los procesos de venta para que los productores puedan ingresar y agregar sus productos para cumplir con el pedido. Una vez se recopile la información, mediante un algoritmo, el sistema determinará los productores y la cantidad de producto que cada uno deberá proveer para el proceso de venta, intentando seleccionar los mejores productos al menor costo posible. Luego, debe comenzar el proceso de recolección de los productos, subastando los transportes y actualizando el estado del proceso.
* El cliente externo debe actualizar el proceso una vez reciba el pedido.
* El sistema evaluará el rendimiento de los procesos mediante la medición de la pérdida de fruta a través del tiempo. Se deben crear reportes para cada rol. El rol de consultor verá la totalidad de reportes.
* El cliente interno ingresará a comprar los saldos, que el administrador publicará en procesos de venta interna.

### **Componentes comunes**

* Maipo Grande.

### **Componente 1**

* Sistema Web: este será utilizado generalmente por clientes de la empresa Natural Verde, se podrán generar solicitudes de compra, consultar estado del mismo.

### **Componente 2**

* Sistema Escritorio: este será utilizado principalmente por personal de la empresa (Administrador), podrá publicar ventas internas, ingresar venta local, distribuir productos, notificar resultados, completar transporte, recibir pago, distribuir pagos, crear informe de venta externa e interna.

### **Componente 3**

* Base de datos: esto será lo principal, donde se guardarán todos los datos ingresados, tales como: productos, productores, clientes, transportistas y consultores entre otros.

## Requerimientos No-Funcionales

* La aplicación debe estar compuesta por un módulo web y un módulo de escritorio. Opcionalmente puede reemplazar el módulo de escritorio por una aplicación móvil.
* El módulo web debe ser construido mediante un modelo de capas, logrando una separación de la interfaz gráfica, reglas de negocio y repositorio de datos.
* La aplicación debe considerar un módulo de administración en ambiente de escritorio, como aplicación satélite, desarrollada en lenguaje java o .net, las funciones del administrador deberán ser implementadas en este módulo con acceso a la base de datos central, por lo cual se trabaja la misma información que la aplicación web.
* Los procesos CRUD se deben efectuar mediante procedimientos almacenados con PL/SQL.
* Considere utilizar PL/SQL para obtener las listas de datos mediante cursores.
* Las notificaciones a los clientes deberán realizarse mediante correo electrónico, o bien, mediante notificaciones a dispositivos móviles.
* La generación de reportes debe considerar el formato PDF.
* El sistema debe incluir medidas de seguridad tales como enmascarar clave y control de sesiones.
* Todas las aplicaciones de usuario deben presentar una interfaz gráfica que considera los elementos de diseño incorporados en las aplicaciones de Windows.
* La autenticación de usuarios debe considerar las medidas de seguridad respectivas, tales como manejo de sesiones y acceso con usuario-clave-perfil a modo de acceder a las funcionalidades de acuerdo al perfil o rol que posee el usuario.
* El sistema debe utilizar base datos Oracle y lenguaje de programación orientado a objetos como Microsoft .NET y J2EE.
* El sistema debe contar con manuales de usuario estructurados adecuadamente.
* El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
* Todas las entradas de datos deben considerar las validaciones correspondientes.
* La aplicación web debe poseer un diseño “Responsive” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.

# Estrategia de Pruebas

## 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: | Considerar Usuarios para Cliente Web como para Clientes y Trabajadores para aplicación de escritorio |
| Observaciones: |  |

### **Pruebas de rendimiento (Performance)**

Pendiente

### **Pruebas de seguridad y de acceso a datos**

Pendiente**Recursos**

## Profesionales

| **Recursos Humanos** | | |
| --- | --- | --- |
| **Rol** | **Recursos mínimos recomendados**  **(Número de personas full-time)** | **Responsabilidades específicas / Comentarios** |
| Diseñador de casos de prueba | 2 personas | Responsabilidades   * Identificar, priorizar e implementar los casos de prueba. * Evaluar de forma el esfuerzo de *testing*. |
| Testeador | 1 persona | 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 | 1 persona | 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 | 1 persona | 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 | 2 personas | Responsabilidades:   * Identificar y definir las operaciones, atributos y asociaciones de las clases de prueba. * Identificar y definir los paquetes de prueba. |
| Implementador | 2 personas | 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.

### **Preparación del ambiente de pruebas**

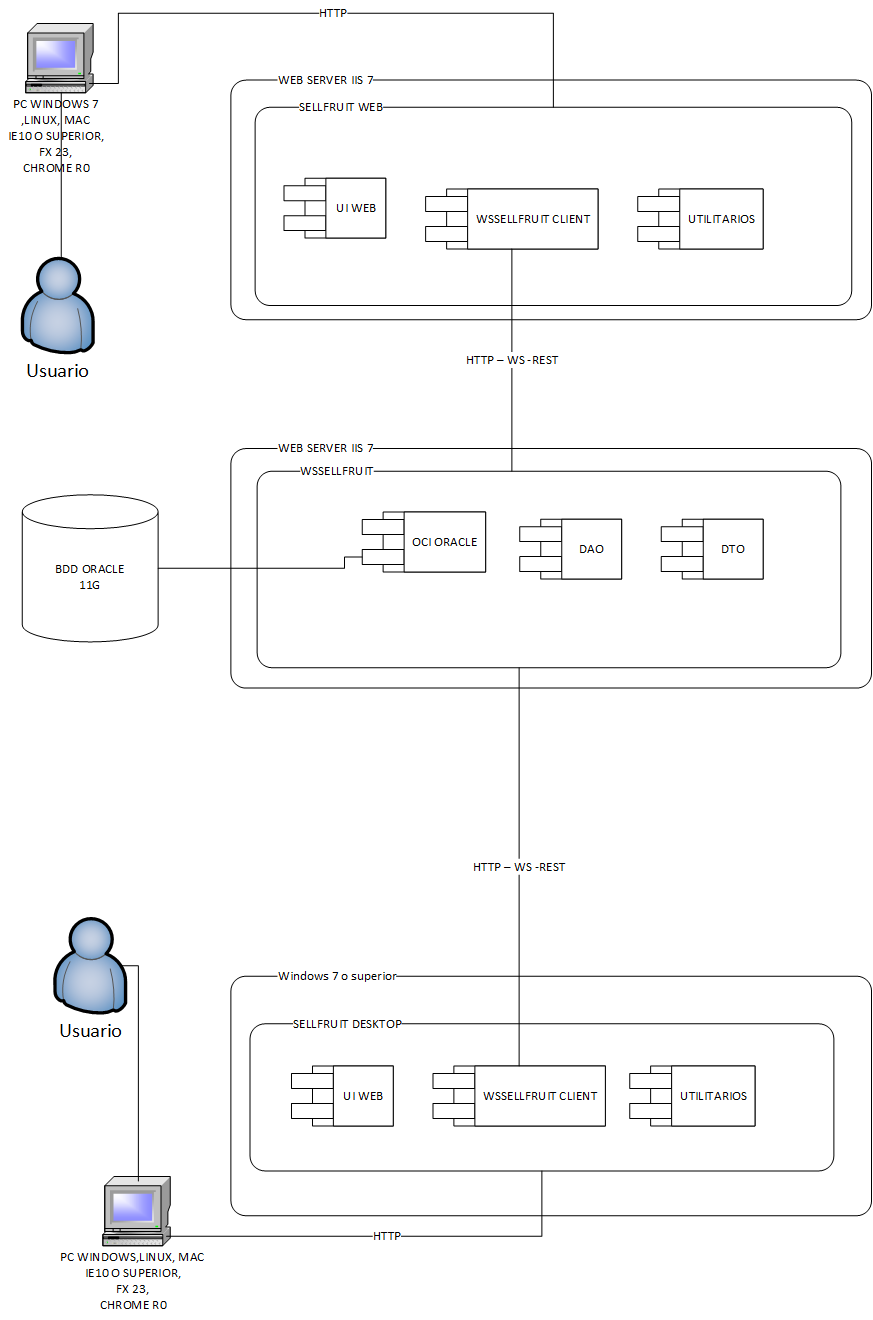
Las pruebas unitarias y de regresión deberán ser ejecutadas dentro del Ambiente de desarrollo, las pruebas de aceptación del usuario y del sistema se ejecutarán en este Ambiente de pruebas. Este ambiente deberá representar una configuración idéntica al Ambiente de producción o al menos, una versión en 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**

| **ITEM** | **DESCRIPCIÓN** | **OBSERVACIONES** |
| --- | --- | --- |
| **HARDWARE** |  |  |
| **Estación de Pruebas (Cliente)** |  |  |
| Procesador | Intel Core i3 o superior |  |
| Memoria RAM | 2GB |  |
| Espacio en Disco | 128GB |  |
| Periféricos | Mouse y teclado | USB o inalámbricos |
| **Servidores** |  |  |
| ***Base de Datos (pruebas)*** |  |  |
| Procesador | Inte Core i3 |  |
| Memoria RAM | 8GB |  |
| Espacio en Disco | 1TB |  |
| Periféricos | Mouse y teclado | USB o inalámbricos |
| ***Aplicación (pruebas)*** |  |  |
| Procesador | Intel Core i3 o superior |  |
| Memoria RAM | 2GB |  |
| Espacio en Disco | 128GB |  |
| Mouse | Mouse y teclado | USB o inalámbricos |
| **Sistema de Respaldo / Restauración** |  |  |
| Unidad (Modelo y Marca) |  |  |
| Capacidad | 2TB |  |
| Ubicación |  |  |
|  |  |  |
| **SOFTWARE** |  |  |
| **Estación de Pruebas (Cliente)** |  |  |
| Sistema Operativo | Windows 10 Profesional |  |
| Herramienta de *testing* | Manuales |  |
| Herramienta de Modelamiento | Data Modeler |  |
| BDMA | Oracle 11g |  |
| Browser | IE o Google Chrome | Ver 11 / Ver 74.0 |
| Software de Escritorio | Microsoft Office |  |
| **Base de Datos (pruebas)** |  |  |
| Sistema Operativo | Windows 10 Profesional |  |
|  |  |  |
| **Aplicación (pruebas)** |  |  |
| Sistema Operativo | Windows 10 Profesional |  |
|  |  |  |
| **Repositorios** |  |  |
| Servidor |  |  |
|  |  |  |
| **Seguridad** | Respaldo 2 veces al mes |  |
|  |  |  |
|  |  |  |

### **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.



### **Integración del ambiente de pruebas y configuración**

Para esta actividad se requerirá la participación de profesionales de diferentes áreas tecnológicas 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** |
| Instalación | Equipo de desarrollo | 01/10/2020 |  |  |
| Configuración | Equipo técnico | 02/10/2020 |  |  |
| Puesta en Marcha | Equipo de desarrollo | 05/10/2020 |  |  |

### **Generación de datos**

La generación de los datos para las pruebas considera 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 10/10/2020.

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.

# Actividades e Hitos del Plan de Pruebas

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

# Entregables

## Plan de pruebas

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

*Equipos y aplicativos instalados correctamente y conexiones a la base de datos probadas y funcionando correctamente*

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

*Funcionamiento de la aplicación de escritorio al 100%, tanto como generar, aceptar o rechazar solicitudes funcionando correctamente*

### Criterio de suspensión y resumisión

Las Pruebas se suspenderán cuando la base de datos se desconecte o algún servicio externo deje de funcionar.

## Resultados de las pruebas

Para desplegar el resultado de pruebas se usara la herramienta web Jira.

## Reporte de defectos

Para desplegar el reporte de defectos se usara la herramienta web Jira.

# 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*)

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 planifique 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 la 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: |  |

## D: Diagramas y casos de usos

Diagramas

