**Plan de pruebas de software**

***Multirental Storage***

***Fecha: 14/11/2024***

**Historia de Revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 2024-11-14 | 1.1.0 | Documento inicial | Matías Allendes |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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

1. **Introducción**
   1. **Resumen ejecutivo**

Este plan de pruebas para el sistema **Multirental Storage** tiene como propósito principal validar que el sistema cumpla con los requerimientos funcionales y no funcionales definidos para su implementación. Se trata de un **plan detallado** que abarca pruebas funcionales, pruebas de rendimiento, pruebas de seguridad y usabilidad, entre otras.

El objetivo es garantizar que el sistema opere conforme a las expectativas del cliente y a los objetivos del proyecto, abordando elementos clave como la gestión de herramientas, sucursales, usuarios y transacciones. Además, este plan considera restricciones de recursos, tiempo y presupuesto, así como el alcance del esfuerzo requerido para asegurar la calidad en la fase de producción.

* 1. **Alcance de las pruebas**
     1. *Elementos de pruebas*

Se probarán los siguientes módulos y funcionalidades del sistema:

**Gestión de Herramientas:**

* Registro, actualización y eliminación de herramientas.
* Asignación de estados (disponible, reservada, en reparación).
* Visualización y filtrado de herramientas según atributos como nombre, estado, sucursal y stock.

**Gestión de Sucursales:**

* Creación, edición y eliminación de sucursales.
* Visualización del stock en cada sucursal y consolidado global.

**Gestión de Usuarios:**

* Creación y actualización de perfiles.
* Asignación y validación de roles y permisos.

**Transacciones:**

* Registro y seguimiento de transacciones (reservas, reparaciones).
* Actualización automática del inventario.

**Generación de Reportes:**

* Reportes detallados por herramienta, sucursal y estado.
* Exportación a formatos CSV y Excel.

**Seguridad y Control de Acceso:**

* Autenticación de usuarios y restricción de funciones según roles.
* Notificaciones y alertas (por ejemplo, stock bajo).

**Interfaz y Usabilidad:**

* Compatibilidad multiplataforma y con navegadores modernos.
* Diseño responsivo para diferentes dispositivos.
  + 1. *Pruebas funcionales*

El testing funcional para **Multirental Storage** se realizará utilizando la técnica de **Caja Negra**. Este enfoque busca comprobar el comportamiento de los módulos y componentes del sistema sin examinar sus detalles internos.

Con esta metodología, se ingresarán datos de entrada relevantes a cada funcionalidad del sistema, se ejecutarán los procesos correspondientes y se verificarán los resultados obtenidos.

* **Datos de entrada:** Incluirán datos válidos (dentro del rango esperado) y no válidos (para probar la robustez del sistema).
* **Resultados esperados:** Cada funcionalidad debe devolver el resultado correcto de acuerdo con las reglas del negocio y las especificaciones del sistema.
  + 1. *Riesgos*

Algunos riesgos comunes a considerar son:

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

* + - 1. Matrices de riesgos
         1. Pruebas

| **Nº Riesgo** | **Descripción** | **Gravedad** | **Acción** |
| --- | --- | --- | --- |
| 1 | Caída de Sistema | Grave | Revisar conexión a internet, reiniciar sistema, contactar a soporte. |
| 2 | Error del cliente al solicitar el producto equivocado. | Leve | Devolver productos, generar nota de crédito y reembolso. |
| 3 | Falla en conexión con sistema bancario | Grave | Revisar conexión a internet, contactar a soporte bancario. |
| 4 | Errores en la Exportación de Reportes | Moderado | Revisar la lógica de exportación y realizar pruebas exhaustivas en formatos XLSX y CSV. |
| 5 | Bajo Rendimiento del Sistema | Grave | Optimizar consultas SQL, revisar índices y ajustar el rendimiento del servidor. |
| 6 | Acceso No Autorizado | Crítico | Implementar controles de autenticación y registro de actividades para auditoría. |

* 1. **Definiciones, acrónimos y abreviaciones**

**Herramienta:** Objeto físico gestionado en el sistema, como taladros, martillos, o cualquier otro producto ofrecido en arriendo o reparación.

**Sucursal:** Punto físico donde se encuentran disponibles las herramientas y se realizan transacciones.

**Transacción:** Operación registrada en el sistema que implica cambios en el stock de herramientas, como reservas, reparaciones, devoluciones o retiros.

**Stock:** Cantidad disponible de herramientas en una sucursal específica.

**Estado de Herramienta:** Condición actual de una herramienta, que puede ser "Disponible", "Reservada" o "En Reparación".

**Usuario:** Persona que interactúa con el sistema, pudiendo ser administrador o usuario regular.

**Reporte:** Documento generado por el sistema que presenta información consolidada sobre herramientas, transacciones y estados.

#### **Acrónimos**

**CRUD:** Create, Read, Update, Delete (Crear, Leer, Actualizar, Eliminar).

**CSV:** Comma-Separated Values, formato para exportación de datos.

**XLSX:** Formato de archivo de Microsoft Excel utilizado para reportes.

**DBMS:** Database Management System, sistema de gestión de bases de datos.

**UI/UX:** User Interface / User Experience, diseño de la interfaz y experiencia del usuario.

* 1. **Referencias**

**Título:** Multirental Storage - Especificación del Sistema

* **Edición/Versión:** 1.0
* **Fecha:** 2024-11-01
* **Autor:** Matias Allendes
* **Archivo:** Multirental\_Storage\_Especificación.pdf

**Título:** Metodología Kanban - Guía de Implementación

* **Edición/Versión:** 2.0
* **Fecha:** 2024-10-15
* **Archivo:** Guía\_Kanban.pdf

**Título:** Plantilla de Plan de Pruebas

* **Edición/Versión:** 1.1.0
* **Fecha:** 2023-05-09
* **Autor:** Matias Allendes
* **Archivo:** Plantilla\_Plan\_de\_Pruebas.docx

**Título:** Documentación de Herramientas de Gestión de Inventario

* **Edición/Versión:** 1.0
* **Fecha:** 2024-10-20
* **Autor:** Augusto Alonso
* **Archivo:** Gestión\_Inventario.pdf

**Título:** Manual Técnico de MariaDB

1. **Edición/Versión:** 10.5
2. **Fecha:** 2024-09-01
3. **Autor:** MariaDB Foundation
4. **Archivo:** MariaDB\_Manual.pdf
5. **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.

* 1. **Casos de uso**
     1. *Vista global*
     2. *Ingresar herramienta*
     3. *Actualizar herramienta*
     4. *Revisar herramienta*
     5. Eliminar herramienta
     6. Ingresar usuario
     7. Modificar usuario
     8. Eliminar usuario
     9. Visualizar usuario
     10. Realizar transacciones
     11. Visualizar stock consolidado
  2. **Requerimientos funcionales**
     1. *Componentes comunes*
     2. *Agregar una herramienta*
     3. *Eliminar una herramienta*
     4. Actualizar herramienta
     5. Visualizar herramienta
     6. Registrar transacciones
     7. Notificaciones
     8. Generación de reportes
  3. **Requerimientos No-Funcionales (Matriz con Input/Output, solo si aplica)**
     1. *Componente 1 (puede tener uno o más casos de uso)*

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | Registrar una herramienta | Ingresar nombre, marca, | La herramienta es registrada correctamente en el sistema. |
| 2 | Actualizar una herramienta | Seleccionar una herramienta, modificar atributos como estado o sucursal | Los datos de la herramienta se actualizan en el sistema. |
| 3 | Eliminar una herramienta | Seleccionar herramienta a eliminar | La herramienta es eliminada del sistema si no tiene transacciones activas. |

* + 1. *Componente 2 (puede tener uno o más casos de uso)*

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | Registrar una transacción | Seleccionar herramienta, tipo de transacción (reserva, disponible, reparación) | La transacción se registra y actualiza el stock y estado de la herramienta. |
| 2 | Generar un reporte | Seleccionar criterios (herramienta, sucursal, estado, fecha) | El reporte se genera en formato CSV o Excel con los datos filtrados. |
| 3 | Notificar stock bajo | Cambios en stock por debajo del umbral establecido | Notificación enviada automáticamente al administrador. |

*6.3.3 Componente 3*

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | Registrar una transacción | Seleccionar herramienta, tipo de transacción (reserva, disponible, reparación) | La transacción se registra y actualiza el stock y estado de la herramienta. |
| 2 | Generar un reporte | Seleccionar criterios (herramienta, sucursal, estado, fecha) | El reporte se genera en formato CSV o Excel con los datos filtrados. |
| 3 | Notificar stock bajo | Cambios en stock por debajo del umbral establecido | Notificación enviada automáticamente al administrador. |

*6.3.4 Componente 4*

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | Registrar una sucursal | Ingresar nombre, ubicación y capacidad | La sucursal se registra correctamente en el sistema. |
| 2 | Actualizar datos de una sucursal | Seleccionar sucursal y modificar datos | Los datos de la sucursal se actualizan en el sistema. |
| 3 | Eliminar una sucursal | Seleccionar sucursal sin herramientas asignadas | La sucursal es eliminada del sistema. |

*6.3.5 Componente 5*

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | Iniciar sesión | Ingresar correo y contraseña | El sistema autentica al usuario y lo redirige a su panel correspondiente. |
| 2 | Recuperar contraseña | Ingresar correo registrado | Se envían las instrucciones para recuperar la contraseña. |
| 3 | Asignar roles a usuarios | Seleccionar usuario y rol | El sistema asigna los permisos correspondientes al rol seleccionado. |

*6.3.6 Componente 6*

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | Consultar stock por sucursal | Seleccionar sucursal | El sistema muestra el stock de herramientas disponibles en esa sucursal. |
| 2 | Consolidar stock global | No requiere input | El sistema presenta el stock consolidado de todas las sucursales. |
| 3 | Filtrar stock por estado | Seleccionar estado (disponible, reservada, en reparación) | El sistema filtra y muestra herramientas según el estado seleccionado. |

*6.3.7 Componente 7*

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | Generar reporte por herramienta | Seleccionar herramienta | El sistema genera un reporte detallado de la herramienta seleccionada. |
| 2 | Generar reporte por sucursal | Seleccionar sucursal | El sistema genera un reporte con el stock de la sucursal seleccionada. |
| 3 | Exportar reporte a Excel o CSV | Seleccionar formato | El sistema exporta el reporte en el formato elegido por el usuario. |

1. **Estrategia de Pruebas**

Estrategia de caja negra, ya que se evaluarán aspectos funcionales del sistema en cuestión.

* 1. **Tipos de pruebas**
     1. *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: |  |

* + 1. *Pruebas de rendimiento (Performance)*

Las pruebas de rendimiento buscan evaluar cómo el sistema Multirental Storage responde bajo diferentes condiciones de carga y uso. Se medirán aspectos como tiempos de respuesta, tasa de transacciones procesadas, y estabilidad del sistema bajo escenarios normales y de estrés.

#### **Objetivo de la Prueba**

1. Validar que las consultas, generación de reportes y actualizaciones en el sistema se completen en un tiempo razonable (< 1 minuto).
2. Garantizar que el sistema se mantenga estable y funcional bajo condiciones de carga elevada.
3. Identificar posibles cuellos de botella en la base de datos o en la arquitectura del sistema.

#### **Técnica a Utilizar**

1. Escenarios con carga normal y carga elevada.
2. Simulación de múltiples usuarios concurrentes ejecutando consultas, transacciones y generación de reportes.
3. Herramientas de prueba para emular usuarios y capturar métricas de rendimiento.
   * 1. *Pruebas de seguridad y de acceso a datos*

Este tipo de pruebas evalúa la seguridad del sistema en dos niveles:

1. **Seguridad de la aplicación:** Verifica que el acceso a funcionalidades y datos esté restringido según los roles de usuario.
2. **Seguridad del sistema:** Garantiza que solo los usuarios autenticados puedan acceder al sistema y que la información sensible esté protegida.

#### **Objetivo de la Prueba**

* Validar que las restricciones de acceso funcionan correctamente según los roles definidos (administrador, usuario regular).
* Proteger datos sensibles, como credenciales de usuarios y registros de herramientas, contra accesos no autorizados.
* Asegurar que las operaciones críticas sean rastreadas mediante logs de auditoría.

**Recursos**

* 1. **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 el 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. |

* 1. **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.

* + 1. *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.

* + 1. *Diseño del ambiente de pruebas*

| **ITEM** | **DESCRIPCIÓN** | **OBSERVACIONES** |
| --- | --- | --- |
| **HARDWARE** |  |  |
| **Estación de Pruebas (Cliente)** |  |  |
| Procesador | *Pentium II 300 MHz o superior* | Compatible con simulación de múltiples usuarios. |
| Memoria RAM | *64 MB (128 MB ideal)* | Para ejecutar pruebas simultáneas y herramientas de testing. |
| Espacio en Disco | *600 MB* | Almacenar datos de pruebas y registros generados. |
| Tipo Monitor y Resolución | *SVGA .28 I 800x600* | Para pruebas de UI/UX en alta resolución. |
| Unidad de Disquete | *1.44 MB* |  |
| Tarjeta de red | Ethernet Gigabit | Para garantizar una conexión estable durante pruebas de carga. |
| Modem |  |  |
| Mouse | *USB estándar* |  |
| Tipo Enlace |  |  |
| **Servidores** |  |  |
| ***Base de Datos (pruebas)*** |  |  |
| Procesador | Intel Xeon o equivalente | Para manejar múltiples consultas concurrentes. |
| Memoria RAM | 4 GB o más | Optimizar el rendimiento de consultas en MariaDB. |
| Espacio en Disco | 250 GB (SSD preferido) | Almacenar bases de datos y logs de auditoría. |
| Tipo Monitor y Resolución | *SVGA .28 I 800x600* | Para pruebas de UI/UX en alta resolución. |
| Unidad de Disquete | *1.44 MB* |  |
| Tarjeta de red | Ethernet Gigabit | Conexión estable con el sistema y estaciones de prueba. |
| Modem |  |  |
| Mouse |  |  |
| Tipo Enlace |  |  |
| ***Aplicación (pruebas)*** |  |  |
| Procesador | Intel Core i7 o superior | Soporte para pruebas de carga y simulación de usuarios. |
| Memoria RAM | 4 GB |  |
| Espacio en Disco | 250 GB |  |
| Tipo Monitor y Resolución |  |  |
| Unidad de Disquete |  |  |
| Tarjeta de red |  |  |
| Módem |  |  |
| Mouse |  |  |
| Tipo Enlace |  |  |
| **Impresoras** |  |  |
| Marca y Modelo | Epson WorkForce o similar | Para impresión de reportes. |
| Tipo | 600 dpi o superior |  |
| Resolución |  |  |
| Rendimiento |  |  |
| Dedicación |  |  |
| **RED** |  |  |
| Topología | LAN | Para pruebas locales en red interna. |
| Medio |  |  |
| Velocidad | 1 Gbps | Para transferencias rápidas y pruebas de rendimiento en red. |
| Protocolo |  |  |
| Módems |  |  |
| Conexión Internet |  |  |
| **Sistema de Respaldo / Restauración** |  |  |
| Unidad (Modelo y Marca) | NAS Synology o similar | Respaldo automatizado de la base de datos y configuraciones de pruebas. |
| Capacidad | 2 TB |  |
| Ubicación |  |  |
|  |  |  |
| **SOFTWARE** |  |  |
| **Estación de Pruebas (Cliente)** |  |  |
| Sistema Operativo | Windows 10 Pro/Linux Ubuntu | Compatible con herramientas de testing y conectividad con el sistema. |
| Herramienta de *testing* | *Rational TeamTest compuesta por Robot, ClearQuest TT, Test Manager* | Simulación de carga, pruebas funcionales y seguridad. |
| Herramienta de Modelamiento | *Rational Rose 2000* |  |
| BDMA | *Access* |  |
| Browser | *Chrome, Firefox, Edge* | Pruebas de compatibilidad en navegadores modernos. |
| Software de Escritorio | *Microsoft Office 365* | Generación y validación de reportes (XLSX, CSV). |
| **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** |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* + 1. *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.

* + 1. *Integración del ambiente de pruebas y configuración*

Para esta actividad se requerirá la participación de profesionales de *Sodimac* 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** |
| --- | --- | --- | --- | --- |
| Configuración de la base de datos (MariaDB) | Administrador de Bases de Datos | 15-10-2024 | 17-20-2024 | Configurar esquemas y cargar datos iniciales necesarios para las pruebas. |
| Instalación del sistema en el servidor | Administrador de Sistemas | 18-10-2024 | 19-20-2024 | Instalar el backend (Flask) y frontend en el servidor de pruebas. |
| Configuración de red y acceso remoto | Responsable de Redes | 20-10-2024 | 23-10-2024 | Configurar acceso seguro y establecer conexiones remotas para los testers. |
| Verificación de compatibilidad multiplataforma | Responsable de QA | 25-10-2024 | 26-10-2024 | Probar el sistema en navegadores modernos y dispositivos móviles. |
| Validación del ambiente de pruebas | Líder de Proyecto | 27-10-2024 | 28-10-2024 | Validar que todos los componentes del sistema funcionen correctamente juntos. |
| Generación de datos de prueba | Equipo de QA | 29-10-2024 | 30-10-2024 | Crear escenarios y datos representativos para ejecutar las pruebas planificadas. |

* + 1. *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.

* + - 1. 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

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.

1. **Actividades e Hitos del Plan de Pruebas**

|  | **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 |  |  |  |  |

1. **Entregables**

| Nombre del Documento/Reporte | Descripción | Autor | Destinatario |
| --- | --- | --- | --- |
| Plan de Pruebas | Documento detallado del proceso de pruebas para evaluar el sistema **Multirental Storage**. | Equipo QA | Equipo de Desarrollo |
| Planilla de Casos de Prueba | Documento con la definición de casos de prueba, datos de entrada, resultados esperados y reales. | Equipo QA | Líder de Proyecto |
| Sumario de Evaluación de Pruebas | Reporte de resultados generales de las pruebas, incluyendo métricas y conclusiones. | Equipo QA | Gerencia de Proyecto |
| Reporte de Defectos | Listado de defectos encontrados, clasificados por gravedad y prioridad. | Equipo QA | Equipo de Desarrollo |
| Reporte de Rendimiento | Análisis del rendimiento del sistema bajo distintas cargas de trabajo. | Equipo QA | Equipo de Desarrollo |
| Reporte de Seguridad | Detalle de vulnerabilidades encontradas y sus soluciones. | Equipo QA | Gerencia de Proyecto |

* 1. **Plan de pruebas**

*Documento en el que se detalla el proceso de pruebas a las que se someterá el sistema para determinar su nivel de calidad.*

* + 1. *Criterio de entrada para el “Plan de pruebas”*
* El sistema debe encontrarse en una fase avanzada de desarrollo o completamente terminado.
* Las funcionalidades principales deben estar implementadas y accesibles para pruebas.
* Los datos de prueba deben estar disponibles y correctamente cargados en el sistema.
* El ambiente de pruebas debe estar configurado y validado.
  + 1. *Criterio de salida para el “Plan de pruebas”*
* Todas las pruebas planificadas deben haberse ejecutado con resultados concluyentes.
* Los defectos críticos encontrados deben estar resueltos o mitigados.
* Los reportes finales de pruebas deben ser generados y entregados.
* El sistema debe cumplir con los requisitos funcionales y no funcionales definidos.
  + 1. *Criterio de suspensión y resumisión*

**Suspensión:**

* Falla crítica en el sistema que impida la ejecución de pruebas (p. ej., caída del servidor).
* Ausencia de funcionalidades esenciales para los casos de prueba.

**Resumisión:**

* Resolución de problemas críticos o implementación de las funcionalidades faltantes.
* Validación de que el sistema está nuevamente en condiciones para continuar las pruebas.
  1. **Resultados de las pruebas**
* **Método:** Se utilizarán pruebas de **caja negra**, verificando el comportamiento del sistema ante entradas específicas y evaluando sus salidas.
* **Registro:** Los resultados serán documentados en un archivo de Microsoft Word con una tabla comparativa entre resultados esperados y reales.
  1. **Reporte de defectos**

**Método:** Los defectos serán registrados en matrices creadas en Microsoft Excel, incluyendo:

* Identificador único del defecto.
* Descripción detallada del problema.
* Pasos para reproducirlo.
* Gravedad y prioridad.
* Estado actual (Abierto, En revisión, Resuelto, Cerrado).

**Documentación:** Los reportes serán consolidados en un archivo de Microsoft Word para su presentación al equipo de desarrollo.

1. **Anexos**
   1. **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 |

* 1. **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: |  |

* 1. **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 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 disponibles 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 deberían 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: |  |