Puerta de Acero

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 24/Abril/19 | 1.0 | Primer revisión | Luis Alberto García Rodríguez |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Intended Audience 3

1.4 Document Terminology and Acronyms 3

1.5 References 3

1.6 Document Structure 3

2. Evaluation Mission and Test Motivation 3

2.1 Background 3

2.2 Evaluation Mission 3

2.3 Test Motivators 3

3. Target Test Items 3

4. Outline of Planned Tests 3

4.1 Outline of Test Inclusions 3

4.2 Outline of other candidates for potential inclusion 3

4.3 Outline of Test Exclusions 3

5. Test Approach 3

5.1 Initial Test-Idea Catalogs and other reference sources 3

5.2 Testing Techniques and Types 3

5.2.1 Data and Database Integrity Testing 3

5.2.2 Function Testing 3

5.2.3 Business Cycle Testing 3

5.2.4 User Interface Testing 3

5.2.5 Performance Profiling 3

5.2.6 Load Testing 3

5.2.7 Stress Testing 3

5.2.8 Volume Testing 3

5.2.9 Security and Access Control Testing 3

5.2.10 Failover and Recovery Testing 3

5.2.11 Configuration Testing 3

5.2.12 Installation Testing 3

6. Entry and Exit Criteria 3

6.1 Test Plan 3

6.1.1 Test Plan Entry Criteria 3

6.1.2 Test Plan Exit Criteria 3

6.1.3 Suspension and resumption criteria 3

6.2 Test Cycles 3

6.2.1 Test Cycle Entry Criteria 3

6.2.2 Test Cycle Exit Criteria 3

6.2.3 Test Cycle abnormal termination 3

7. Deliverables 3

7.1 Test Evaluation Summaries 3

7.2 Reporting on Test Coverage 3

7.3 Perceived Quality Reports 3

7.4 Incident Logs and Change Requests 3

7.5 Smoke Test Suite and supporting Test Scripts 3

7.6 Additional work products 3

7.6.1 Detailed Test Results 3

7.6.2 Additional automated functional Test Scripts 3

7.6.3 Test Guidelines 3

7.6.4 Traceability Matrices 3

8. Testing Workflow 3

9. Environmental Needs 3

9.1 Base System Hardware 3

9.2 Base Software Elements in the Test Environment 3

9.3 Productivity and Support Tools 3

9.4 Test Environment Configurations 3

10. Responsibilities, Staffing and Training Needs 3

10.1 People and Roles 3

10.2 Staffing and Training Needs 3

11. Iteration Milestones 3

12. Risks, Dependencies, Assumptions and Constraints 3

13. Management Process and Procedures 3

13.1 Measuring and Assessing the Extent of Testing 3

13.2 Assessing the deliverables of this Test Plan 3

13.3 Problem Reporting, Escalation and Issue Resolution 3

13.4 Managing Test Cycles 3

13.5 Traceability Strategies 3

13.6 Approval and Signoff 3

# Introduction

## Purpose

El propósito de este plan de pruebas es tener presente todos los módulos que serán sometidos a las diferentes pruebas

## Scope

El alcance de este plan de pruebas es para determiner las posibles fallas que pudieran encontrarse en nuestro Sistema, no sin estar concientes de que las pruebas que se realizarán no quieren decir que nuestro Sistema esté libre al cien por ciento de fallas, sino que gracias a las pruebas que se realizarán en partes específicas, se reducirán considerablemente las principals fallas que puedan presentarse.

## Intended Audience

Las pruebas que relaizaremos estarán enfocadas principalmente al funcionamiento correcto del Sistema, tanto en fucnión de la calidad del Desarrollo, así como enfocado a que las pruebas demuestren que se están cumpliendo con los requerimientos solicitados por nuestro cliente.

## Document Terminology and Acronyms

Software

## References

Referencias

## Document Structure

Seeción para elimiar porque se utilizarán las tablas de contenido

# Evaluation Mission and Test Motivation

Con el objetivo de cumplir con la calidad que caracteriza a nuestra empresa, es nuestra mission realizer la mayor cantidad de pruebas exaustivas que nos permitan demostrar por que somos los mejores en el Desarrollo de sistemas que estén a la altura de las tecnologías actuals.

## Background

Las pruebas que se realizarán en este plan de pruebas estarán enfocadas principalmente a la calidad de los requerimientos solicitados por el cliente pues él es el principal motivo de nuestros esfuerzos en esta empresa.

Este Proyecto está enfocado a la realización de cobros de cuotas de manmtenimiento en el condominio Puerta de Acero, y gracias al Sistema en cuestión, todo este proceso se podrá llevar a cabo de una maner más eficiente como organizada.

## Evaluation Mission

La mission que temenos con este plan de pruebas tiene el objetivo de:

• encontrar tantos errores como sea posible

• Encontrar problemas importantes, evaluar riesgos de calidad percibidos.

• Asesorar sobre los riesgos percibidos del proyecto.

• verificar algunas de las especificaciones de diseño y de requisites.

• Asesorar sobre la calidad del producto, y satisfacer a los interesados.

• asesorar sobre las pruebas

• cumplir con los mandatos del proceso

## Test Motivators

Las pruebas que realizaremos están motivadas por muchas cosas: riesgos de calidad, riesgos técnicos, riesgos de proyectos, casos de uso, requisitos funcionales, requisitos no funcionales, elementos de diseño, fallas o fallos sospechosos, solicitudes de cambio, entre otras cosas, por lo tanto son indispensables para satisfacer las garantías que nuestra empresa ofrece a nuestros clientes.

# Target Test Items

Para llenado.

# Outline of Planned Tests

Para nuestro Sistema se realizarán diferentes tipos de pruebas tanto de caja blanca como de caja negra, o pruebas estáticas, como dinámicas, esto con el fin de cubrir la mayor parte de las posibilidades de falla que pudieran presentarse en la vida útil del Sistema.

## Outline of Test Inclusions

Se realizarán pruebas estáticas, como dinámicas, esto con el fin de cubrir la mayor parte de las posibilidades de fallas que pudieran presentarse en la vida útil del Sistema. La pruebas que se realizarán son pruebas de aceptación, pruebas de Sistema, priuebas de integración, y pruebas de components.

## Outline of Other Candidates for Potential Inclusion

Se evaluarán las secciones de movimientos de usuarios, tanto de registro como de edición y eliminación pues a veces son partes que no se consideran tanto y es una buena área de oportunidad para realizer las diversas pruebas y probar la calidad del Sistema.

## Outline of Test Exclusions

No se realizarán exclusiones de pruebas en ningún área del Sistema pues es un Sistema relativamente pequeño y existen los recursos tnto humanos como técnicos para realizer las ´ruebas necesarias en todas las áreas de dicho Sistema.

# Test Approach

Se utilizará la técnica de pruebas unitarias para determiner de la manera más sencilla las posibles fallas en general sobre los módulos del Sistema, de la misma manera se utilizará la técnica de puebas o análisis de valores límites porquecon esta técnica, como su nombre lo indica, se analizan los valores límite que generalmente son en donde más fallas se pueden encontrar en los sistemas pues en ocasiones los programadores no realizan estas pruebas de valores límite sobre sus propios desarrollos por falta de tiempo.

## Initial Test-Idea Catalogs and Other Reference Sources

## Testing Techniques and Types

### Data and Database Integrity Testing

|  |  |
| --- | --- |
| Technique Objective: | Verificar la integridad y la consistencia de los datos |
| Technique: | Realizar consultas comunes para determiner si da los resultados esperados  Realizar consultas con left y right join para determiner si los valores nulos los muestra correctamente |
| Oracles: |  |
| Required Tools: | Necesitaremos:   * XAMPP como servidor web y de bases de datos ya que incluye PHP y MySQL * El editor de Código Sublime Text * Scripts para realizer las pruebas de las consultas a la base de datos |
| Success Criteria: | Principalmente se realizarán consultas de selección de datos, ya que son las que más se realizarán para obtener la información deseada cuando el Sistema se encuentre en funcionamiento |
| Special Considerations: | * Se requerirá del servidor MySQL administrado con Phpmyadmin para realizer las consultas en un entorno gráfico y que sea más clara la visualización de los resultados obtenidos. * Es necesario que la base de datos cuente con pocos registros para también poder observer y comparer los diferentes resultados obtenidos a través de las prubeas. |

### Function Testing

|  |  |
| --- | --- |
| Technique Objective: | En esta técnica se realizarán entradas de datos en la interfaz grafica del Sistema para comprobar us funcionalidad de acuerdo a los requerimientos solicitados por el cliente |
| Technique: | Ejecutaremos los flujos o funciones y funciones individuales de cada caso de uso del escenario de uso, utilizando datos válidos e inválidos, para verificar que:  • los resultados esperados ocurren cuando se usan datos válidos  • se muestran los mensajes de error o advertencia apropiados cuando se utilizan datos no válidos  • cada regla de negocio se aplica correctamente |
| Oracles: | Para empezar se propone capturer datos numéricos en campos donde se debería de admitir solo texto y viceversa para determinar que el Sistema responde de una manera adecuada a esta entrada de datos incorrecta y se recupera o no falla. |
| Required Tools: | Necesitaremos:   * Scripts para ingresar datos automaticamente * Una bases de datos de respaldo para realizer las pruebas |
| Success Criteria: | Los criterios de evaluación correcta serán válidos cuando el sistema no presente fallas y se compruebe que avisa que las entradas de datos no son válidas y pida una correrción al usuarios de los datos que ingresará |
| Special Considerations: | Puede ser una limitamte el no poder realizer pruebas a todo el Sistema pero es poco el impacto |

### Business Cycle Testing

|  |  |
| --- | --- |
| Technique Objective: | Ejerza los procesos de objetivo de prueba y de fondo de acuerdo con los modelos de negocio y programas requeridos para observar y registrar el comportamiento objetivo |
| Technique: | Las pruebas simularán varios ciclos económicos al realizar lo siguiente:  • Las pruebas utilizadas para las pruebas de función del objetivo de la prueba se modificarán o mejorarán para aumentar el número de veces que se ejecuta cada función para simular varios usuarios diferentes durante un período específico.  • Todas las funciones sensibles a la hora o la fecha se ejecutarán utilizando fechas o períodos de tiempo válidos e inválidos.  • Todas las funciones que se producen en un horario periódico se ejecutarán o lanzarán en el momento adecuado.  • Las pruebas incluirán el uso de datos válidos e inválidos para verificar lo siguiente:  o Los resultados esperados ocurren cuando se usan datos válidos.  o Los mensajes de error o advertencia apropiados se muestran cuando se utilizan datos no válidos.  o Cada regla de negocio se aplica correctamente. |
| Oracles: | Se realizarán verificaciones periódica e incrementales a lo largo del periodo de desarrollo |
| Required Tools: | La técnica requiere las siguientes herramientas:  • Herramienta de automatización de secuencias de comandos de prueba  • Configurador de imágenes de base y restaurador.  • herramientas de copia de seguridad y recuperación  • Herramientas de generación de datos |
| Success Criteria: | La técnica es compatible con la prueba de todos los ciclos comerciales críticos |
| Special Considerations: | Las fechas y eventos del sistema pueden requerir actividades de apoyo especiales.  • Se requiere un modelo de negocio para identificar los requisitos y procedimientos de prueba apropiados. |

### User Interface Testing

|  |  |
| --- | --- |
| Technique Objective: | Navegaremos a través de los módulos creados para verificar que dicha navegación se realiza correctamente usando diferentes métodos de acceso a las diferentes seccioines. |
| Technique: | Crear o modificar las pruebas para cada ventana para verificar la navegación adecuada y los estados del objeto para cada ventana y objeto de la aplicación. |
| Oracles: | Resumir una o más estrategias que la técnica puede usar para observar con precisión los resultados de la prueba. |
| Required Tools: | Utilizaremos scripts automatizados para esta tarea |
| Success Criteria: | Se considera válido cuando la navegación se realiza correctamente por las diferentes ventanas y funciones del sistema |
| Special Considerations: |  |

### Performance Profiling

|  |  |
| --- | --- |
| Technique Objective: | Ejercicios de comportamiento para transacciones funcionales designadas o funciones comerciales bajo las siguientes condiciones para observar y registrar el comportamiento objetivo y los datos de rendimiento de la aplicación:  • carga de trabajo normal anticipada  • carga de trabajo en el peor de los casos prevista |
| Technique: | Use los procedimientos de prueba desarrollados para pruebas de función o de ciclo comercial.  • Modifique los archivos de datos para aumentar la cantidad de transacciones o los scripts para aumentar la cantidad de iteraciones que ocurren en cada transacción.  • Los scripts deben ejecutarse en una máquina (el mejor de los casos para evaluar un solo usuario, una sola transacción) y deben repetirse con varios clientes (virtual o real, consulte Consideraciones especiales a continuación). |
| Oracles: |  |
| Required Tools: | La técnica requiere las siguientes herramientas:  • Herramienta de automatización de secuencias de comandos de prueba  • una herramienta de perfiles de rendimiento de aplicaciones, como Rational Quantify  • herramientas de supervisión de la instalación (registro, disco duro, CPU, memoria, etc.)  • herramientas de restricción de recursos; por ejemplo, Calor Enlatado |
| Success Criteria: | La técnica soporta pruebas:  • Transacción única o usuario único: emulación exitosa de los scripts de transacción sin fallas debido a problemas de implementación de prueba.  • Transacciones múltiples o usuarios múltiples: emulación exitosa de la carga de trabajo sin fallas debido a problemas de implementación de prueba. |
| Special Considerations: | Las pruebas de rendimiento integrales incluyen tener una carga de trabajo de fondo en el servidor.  Hay varios métodos que pueden usarse para realizar esto, incluyendo:  • “Conducir transacciones” directamente al servidor, generalmente en forma de llamadas de lenguaje de consulta estructurado (SQL).  • Cree una carga de usuarios "virtual" para simular muchos clientes, generalmente varios cientos. Las herramientas de emulación de terminal remota se utilizan para lograr esta carga. Esta técnica también se puede utilizar para cargar la red con "tráfico".  • Utilice varios clientes físicos, cada uno de los scripts de prueba en ejecución, para colocar una carga en el sistema.  Las pruebas de rendimiento se deben realizar en una máquina dedicada o en un tiempo dedicado. Esto permite un control total y una medición precisa.  Las bases de datos utilizadas para las Pruebas de rendimiento deben tener un tamaño real o una escala igual |

### Load Testing

|  |  |
| --- | --- |
| Technique Objective: | Ejecute transacciones designadas o casos de negocios bajo diferentes condiciones de carga de trabajo para observar y registrar el comportamiento objetivo y los datos de rendimiento del sistema. |
| Technique: | Utilice las secuencias de comandos de prueba de transacción desarrolladas para pruebas de función o ciclo de negocios como base, pero recuerde eliminar las interacciones y demoras innecesarias.  • Modifique los archivos de datos para aumentar la cantidad de transacciones o las pruebas para aumentar la cantidad de veces que ocurre cada transacción.  • Las cargas de trabajo deben incluir (p. Ej., Cargas diarias, semanales, mensuales, etc.).  • Las cargas de trabajo deben representar tanto la carga media como la máxima.  • Las cargas de trabajo deben representar picos instantáneos y sostenidos.  • Las cargas de trabajo deben ejecutarse bajo diferentes configuraciones de entorno de prueba |
| Oracles: | Resuma una o más estrategias que la técnica puede usar para observar con precisión los resultados de la prueba. El oráculo combina elementos del método mediante el cual se puede realizar la observación y las características de un resultado específico que indican un éxito o fracaso probable. Lo ideal es que los oráculos se auto-verifiquen, permitiendo que las pruebas automatizadas realicen una evaluación inicial de la aprobación o el fracaso de la prueba; sin embargo, tenga cuidado de mitigar los riesgos inherentes a la determinación automática de resultados. |
| Required Tools: | La técnica requiere las siguientes herramientas:  • Herramienta de automatización de secuencias de comandos de prueba  • Programación de la carga de la transacción y herramienta de control  • herramientas de supervisión de la instalación (registro, disco duro, CPU, memoria, etc.)  • herramientas de restricción de recursos (por ejemplo, Canned Heat)  • Herramientas de generación de datos |
| Success Criteria: | La técnica admite la prueba de Emulación de carga de trabajo, que es la emulación exitosa de la carga de trabajo sin fallas debido a problemas de implementación de prueba. |
| Special Considerations: | Las pruebas de carga se deben realizar en una máquina dedicada o en un tiempo dedicado. Esto permite un control total y una medición precisa.  • Las bases de datos utilizadas para las pruebas de carga deben tener un tamaño real o una escala igual. |

### Stress Testing

|  |  |
| --- | --- |
| Technique Objective: | Ejercite las funciones de objetivo de prueba en las siguientes condiciones de estrés para observar y registrar el comportamiento objetivo que identifica y documenta las condiciones bajo las cuales el sistema no puede continuar funcionando correctamente  • poca o ninguna memoria disponible en el servidor (RAM y espacio de almacenamiento persistente)  • número máximo real o físicamente capaz de clientes conectados o simulados  • usuarios múltiples que realizan las mismas transacciones con los mismos datos o cuentas  • "sobrecargar" el volumen de transacciones o la mezcla (consulte el perfil de rendimiento más arriba) |
| Technique: | Pruebas de uso desarrolladas para perfiles de rendimiento o pruebas de carga.  • Para probar recursos limitados, las pruebas deben ejecutarse en una sola máquina, y la RAM y el espacio de almacenamiento persistente en el servidor deben reducirse o limitarse.  • Para las pruebas de estrés remanentes, se deben usar múltiples clientes, ya sea ejecutando las mismas pruebas o pruebas complementarias para producir el volumen o la combinación de transacciones del peor de los casos. |
| Oracles: |  |
| Required Tools: | La técnica requiere las siguientes herramientas:  • Herramienta de automatización de secuencias de comandos de prueba  • Programación de la carga de la transacción y herramienta de control  • herramientas de supervisión de la instalación (registro, disco duro, CPU, memoria, etc.)  • herramientas de restricción de recursos (por ejemplo, Canned Heat)  • Herramientas de generación de datos |
| Success Criteria: | La técnica soporta la prueba de emulación de estrés. El sistema se puede emular con éxito en una o más condiciones definidas como condiciones de estrés y se puede capturar una observación del estado del sistema resultante durante y después de que la condición haya sido emulada. |
| Special Considerations: | • Hacer hincapié en la red puede requerir herramientas de red para cargar la red con mensajes o paquetes.  • El almacenamiento persistente utilizado para el sistema debe reducirse temporalmente para restringir el espacio disponible para que la base de datos crezca.  • Sincronizar los clientes simultáneos que acceden a los mismos registros o cuentas de datos. |

### Volume Testing

|  |  |
| --- | --- |
| Technique Objective: | Ejecute el objetivo de la prueba en los siguientes escenarios de gran volumen para observar y registrar el comportamiento objetivo:  • Número máximo (real o físicamente capacitado) de clientes conectados, o simulados, todos desempeñando la misma función empresarial en el peor de los casos (desempeño) durante un período prolongado.  • Se ha alcanzado el tamaño máximo de la base de datos (real o escalado) y se ejecutan simultáneamente múltiples consultas o transacciones de informes. |
| Technique: | • Use las pruebas desarrolladas para el rendimiento de rendimiento o pruebas de carga.  • Se deben usar varios clientes, ya sea ejecutando las mismas pruebas o pruebas complementarias para producir el volumen o la mezcla de transacciones en el peor de los casos (ver Pruebas de estrés) durante un período prolongado.  • Se crea el tamaño máximo de la base de datos (real, escalado o se llena con datos representativos) y se utilizan varios clientes para ejecutar consultas e informar transacciones de forma simultánea durante períodos prolongados. |
| Oracles: |  |
| Required Tools: | La técnica requiere las siguientes herramientas:  • Herramienta de automatización de secuencias de comandos de prueba  • Programación de la carga de la transacción y herramienta de control  • herramientas de supervisión de la instalación (registro, disco duro, CPU, memoria, etc.)  • herramientas de restricción de recursos (por ejemplo, Canned Heat)  • Herramientas de generación de datos |
| Success Criteria: | [La técnica soporta la prueba de Emulación de Volumen. Se pueden emular con éxito grandes cantidades de usuarios, datos, transacciones u otros aspectos del uso del sistema bajo volumen y se puede capturar una observación de los cambios de estado del sistema durante la duración de la prueba de volumen. |
| Special Considerations: | La técnica soporta la prueba de Emulación de Volumen. Se pueden emular con éxito grandes cantidades de usuarios, datos, transacciones u otros aspectos del uso del sistema bajo volumen y se puede capturar una observación de los cambios de estado del sistema durante la duración de la prueba de volumen. |

### Security and Access Control Testing

|  |  |
| --- | --- |
| Technique Objective: | Ejercite el objetivo de la prueba en las siguientes condiciones para observar y registrar el comportamiento objetivo:  • Seguridad a nivel de la aplicación: un actor puede acceder solo a aquellas funciones o datos para los cuales su tipo de usuario cuenta con permisos.  • Seguridad a nivel del sistema: solo los actores con acceso al sistema y las aplicaciones pueden acceder a ellos |
| Technique: | Seguridad a nivel de la aplicación: identifique y enumere cada tipo de usuario y las funciones o datos para los que cada tipo tiene permisos].  o Cree pruebas para cada tipo de usuario y verifique cada permiso creando transacciones específicas para cada tipo de usuario.  o Modificar el tipo de usuario y volver a ejecutar las pruebas para los mismos usuarios. En cada caso, verifique que las funciones o datos adicionales estén disponibles o denegados correctamente.  • Acceso a nivel del sistema: Ver Consideraciones especiales a continuación |
| Oracles: |  |
| Required Tools: | La técnica requiere las siguientes herramientas:  • Herramienta de automatización de secuencias de comandos de prueba  • "Hacker" herramientas de violación de seguridad y sondeo  • Herramientas de administración de seguridad del sistema operativo |
| Success Criteria: | La técnica admite la prueba para cada tipo de actor conocido, se pueden probar las funciones apropiadas o los datos afectados por la configuración de seguridad. |
| Special Considerations: | El acceso al sistema debe ser revisado o discutido con la red o el administrador de sistemas apropiado. Es posible que estas pruebas no sean necesarias, ya que puede ser una función de la administración de redes o sistemas. |

### Failover and Recovery Testing

|  |  |
| --- | --- |
| Technique Objective: | Simule las condiciones de falla y ejecute los procesos de recuperación (manual y automatizado) para restaurar la base de datos, las aplicaciones y el sistema a un estado conocido y deseado. Los siguientes tipos de condiciones se incluyen en las pruebas para observar y registrar el comportamiento objetivo después de la recuperación:  • Interrupción de energía al cliente.  • Interrupción de energía al servidor.  • interrupción de la comunicación a través de servidores de red  • interrupción, comunicación o pérdida de alimentación a DASD (Dispositivos de almacenamiento de acceso dinámico) y controladores DASD  • ciclos incompletos (procesos de filtrado de datos interrumpidos, procesos de sincronización de datos interrumpidos)  • punteros o claves de base de datos no válidos  • elementos de datos inválidos o dañados en la base de datos |
| Technique: | [Las pruebas que ya se crearon para las pruebas de funciones y ciclos de negocios se pueden utilizar como base para crear una serie de transacciones para respaldar las pruebas de recuperación y recuperación de fallos, principalmente para definir las pruebas que se ejecutarán para probar que la recuperación fue exitosa.  • Interrupción de energía al cliente: apaga la PC.  • Interrupción de alimentación en el servidor: simule o inicie procedimientos de apagado para el servidor.  • Interrupción a través de servidores de red: simule o inicie una pérdida de comunicación con la red (desconecte físicamente los cables de comunicación o apague los servidores o enrutadores de la red).  • Interrupción, comunicación o pérdida de alimentación a los controladores DASD y DASD: simule o elimine físicamente la comunicación con uno o más DASD o controladores.  Una vez que se logran las condiciones anteriores o las condiciones simuladas, se deben ejecutar transacciones adicionales y, al alcanzar este segundo estado de prueba, se deben invocar los procedimientos de recuperación.  La prueba de ciclos incompletos utiliza la misma técnica que se describe anteriormente, excepto que los procesos de la base de datos deben abortarse o terminarse prematuramente.  La prueba de las siguientes condiciones requiere que se logre un estado de base de datos conocido.  Varios campos de la base de datos, punteros y claves deben corromperse manual y directamente dentro de la base de datos (a través de herramientas de base de datos). Las transacciones adicionales deben ejecutarse utilizando las pruebas de Pruebas de función de aplicación y Ciclo comercial y los ciclos completos ejecutados. |
| Oracles: |  |
| Required Tools: | La técnica requiere las siguientes herramientas:  • Configurador de imágenes de base y restaurador.  • herramientas de supervisión de la instalación (registro, disco duro, CPU, memoria, etc.)  • herramientas de copia de seguridad y recuperación |
| Success Criteria: | La técnica soporta la prueba de:  • Uno o más desastres simulados que involucran una o más combinaciones de la aplicación, la base de datos y el sistema.  • Una o más recuperaciones simuladas que involucran una o más combinaciones de la aplicación, la base de datos y el sistema a un estado deseado conocido. |
| Special Considerations: | • Las pruebas de recuperación son altamente intrusivas. Los procedimientos para desconectar el cableado (simulando la pérdida de energía o comunicación) pueden no ser deseables o factibles. Se pueden requerir métodos alternativos, como herramientas de software de diagnóstico.  • Se requieren los recursos de los grupos de Sistemas (o Operaciones de computadora), Base de datos y Redes.  • Estas pruebas deben realizarse después de horas o en una máquina aislada. |

### Configuration Testing

|  |  |
| --- | --- |
| Technique Objective: | Ejecute el objetivo de la prueba en las configuraciones de hardware y software requeridas para observar y registrar el comportamiento del objetivo en diferentes configuraciones e identificar los cambios en el estado de la configuración. |
| Technique: | • Usar scripts de prueba de función.  • Abra y cierre varios programas no relacionados con el objetivo de la prueba, como Microsoft Excel y las aplicaciones de Word, ya sea como parte de la prueba o antes del inicio de la prueba.  • Ejecutar transacciones seleccionadas para simular actores que interactúan con el objetivo de la prueba y el software que no es el objetivo de la prueba.  • Repita el proceso anterior, minimizando la memoria convencional disponible en la estación de trabajo cliente. |
| Oracles: |  |
| Required Tools: | La técnica requiere las siguientes herramientas:  • configuración de base de imágenes y restauración  • herramientas de supervisión de la instalación (registro, disco duro, CPU, memoria, etc.) |
| Success Criteria: | La técnica admite la prueba de una o más combinaciones de los elementos de prueba de destino que se ejecutan en los entornos de implementación admitidos y esperados. |
| Special Considerations: | • ¿Qué software que no es el objetivo de la prueba es necesario, está disponible y qué se puede acceder desde el escritorio?  • ¿Qué aplicaciones se usan típicamente?  • ¿Qué datos están ejecutando las aplicaciones? por ejemplo, ¿una hoja de cálculo grande abierta en Excel o un documento de 100 páginas en Word?  • La red, los servidores de red, las bases de datos, etc. de todo el sistema también deben documentarse como parte de esta prueba. |

### Installation Testing

|  |  |
| --- | --- |
| Technique Objective: | Ejecute la instalación del objetivo de prueba en cada configuración de hardware requerida en las siguientes condiciones para observar y registrar el comportamiento de la instalación y los cambios de estado de la configuración:  • nueva instalación: una nueva máquina, nunca instalada previamente con <Nombre del proyecto>  • actualización: una máquina previamente instalada <Nombre del proyecto>, misma versión  • actualización: una máquina previamente instalada <Nombre del proyecto>, versión anterior |
| Technique: | • [Desarrolle scripts automatizados o manuales para validar la condición de la máquina de destino.  o nuevo: nunca instalado  o la misma versión o la anterior ya está instalada  • Iniciar o realizar la instalación.  • Utilizando un subconjunto predeterminado de scripts de prueba de función, ejecute las transacciones. |
| Oracles: |  |
| Required Tools: | La técnica requiere las siguientes herramientas:  • Configurador de imágenes de base y restaurador.  • herramientas de supervisión de la instalación (registro, disco duro, CPU, memoria, etc.) |
| Success Criteria: | La técnica admite la prueba de la instalación del producto desarrollado en una o más configuraciones de instalación. |
| Special Considerations: | ¿Qué transacciones de <Nombre del proyecto> deben seleccionarse para incluir una prueba de confianza de que la aplicación <Nombre del proyecto> se ha instalado correctamente y no faltan componentes de software importantes? |

# Entry and Exit Criteria

## Test Plan

### Test Plan Entry Criteria

El plan de pruebas comenzará una vez los requerimientos del Sistema esté bien definidos pues de ello dependeremos para poder comenzar con las pruebas de aceptación, y conforme se vaya avanzando el Desarrollo de los módulos del Sistema

### Test Plan Exit Criteria

El plan de pruebas terminará una vez que las pruebas demuestren que el Sistema en Desarrollo no de señales de fallas cuando las pruebas se hayan realizado exaustivamente.

### Suspension and Resumption Criteria

Las pruebas se suspenderán en caso de que estén demostrando o sacando a la luz fallas en exceso, pues con esto se sobreentiende que el Desarrollo del Sistema debe mejorar. De la misma manera una causal de suspension de las pruebas haya demoras en el Desarrollo y/o que el cliente haya decidido suspender o cancelar el Proyecto.

## Test Cycles

### Test Cycle Entry Criteria

En caso de que las pruebas se hallan suspendido por que el Desarrollo se haya visto retrasado, se reanudarán las pruebas una vez que haya suficiente avance en dicho Desarrollo pues los módulos o las configuraciónes para las pruebas dependen directamente de dichos avances.

### Test Cycle Exit Criteria

Para este plan de pruebas se determinará que es momento de darlo por concluido cuando las pruebas demuestrén que las fallas en el Sistema, con las pruebas determinadas ya no existen, una ves integrados todos los módulos de el ciclo correspondiente.

### Test Cycle Abnormal Termination

En caso de que las pruebas demuestren que hay demasiadas fallas, se deberá considerar cambiar los módulos a probar o verificar si las configuraciones de la pruebas son las adecuadas de acuerdo al avance del Desarrollo delo Sistema.

# Deliverables

Se entregarán los módulos de configuración de creación, eliminación, edición y visualización de los usuarios del Sistema, pues gracias a ello se considerará como realizado este esfuerzo de pruebas de este ciclo.

## Test Evaluation Summaries

Los resumenes de evaluación de las pruebas se entregarán de manera semanal para de esta forma continuar con el aseguramiento de la calidad del Desarrollo del Sistema y darle seguimiento a dicho avance progresivo.

## Reporting on Test Coverage

Como se dijo en el punto anterior, los reports de los avances de las pruebas se realizará de manera seminal para darle el debido seguimiento a las pruebas, y estos avances se enviarán de manera electronica al correo corporativo y al repositorio o base de datos de avance de las pruebas realizadas.

## Perceived Quality Reports

El contenido de los informes de aseguramiento de la calidad contendrá el author que realiza las pruebas, el que lo documenta, se realizará de manera mensual, se incluirán los datos comprobables obtenidos de las pruebas, se enviará a los correos corporativos y se realizará el registro en las base de datos de seguimiento a las pruebas para llevar el control de los avances.

## Incident Logs and Change Requests

Para entregar los resultados de las pruebas y darle el debido seguimiento se deberá entregar la documentación en matrices de pruebas, así como el registro en la base de datos de los avances de las pruebas para darle continuidad al seguimiento.

## Smoke Test Suite and Supporting Test Scripts

Para esta sección aún no hay especificación

## Additional Work Products

No aplica

### Detailed Test Results

Se entregarán los resultados mediante los repositories para llevar el seguimiento de los casos de pruebas.

### Additional Automated Functional Test Scripts

Estos serán una recopilación de los archivos de código fuente para los scripts de prueba automatizados, o el repositorio de código fuente y ejecutables compilados para los scripts de prueba mantenidos por el producto de automatización de prueba.

### Test Guidelines

Directrices de examen abarcan un amplio conjunto de categorías, incluyendo catálogos Test-Idea, Guía de Buenas Prácticas, patrones de prueba, fallo y problema modelos, estándares de automatización de diseño, y así sucesivamente.

### Traceability Matrices

Al usar una herramienta como Rational RequisitePro o MS Excel, proporcione una o más matrices de relaciones de trazabilidad entre los elementos rastreados

# Testing Workflow

Proporcione un resumen del flujo de trabajo a seguir por el equipo de prueba en el desarrollo y ejecución de este plan de prueba.

El flujo de trabajo de prueba específico que utilizará debe documentarse por separado en el Caso de desarrollo del proyecto. Debería explicar cómo el proyecto ha personalizado el flujo de trabajo de prueba RUP base (generalmente en una fase por fase). En la mayoría de los casos, le recomendamos que coloque una referencia en esta sección del Plan de prueba en la sección correspondiente del Caso de desarrollo. Puede ser útil y suficiente para incluir simplemente un diagrama o una imagen que muestre su flujo de trabajo de prueba.

Los detalles más específicos de las tareas de prueba individuales se definen de diferentes maneras, dependiendo de la cultura del proyecto; por ejemplo:

• definido como una lista de tareas en esta sección del plan de prueba, o en un apéndice adjunto

• definido en una programación central del proyecto (a menudo en una herramienta de programación como Microsoft Project)

• documentado en listas de tareas individuales, "dinámicas" para cada miembro del equipo, que generalmente son demasiado detalladas para ser incluidas en el plan de prueba

• documentado en una pizarra de ubicación central y actualizado dinámicamente

• no documentado formalmente en absoluto

En función de la cultura de su proyecto, debe enumerar aquí sus tareas de prueba específicas o proporcionar un texto descriptivo que explique el proceso que utiliza su equipo para manejar la planificación detallada de las tareas y proporcionar una referencia de dónde se almacenan los detalles, si corresponde.

Para los planes de prueba maestros, recomendamos evitar la planificación detallada de tareas, que a menudo es un esfuerzo improductivo si se realiza como una actividad cargada al principio del proyecto. Un plan de prueba maestro podría describir provechosamente las fases y el número de iteraciones, y dar una indicación de qué tipos de pruebas generalmente se planifican para cada fase o iteración.

Nota: Cuando el proceso y la información detallada de la planificación se registren de manera centralizada y separada de este Plan de prueba, deberá gestionar los problemas que surjan al tener copias duplicadas de la misma información. Para evitar que los miembros del equipo hagan referencia a información desactualizada, sugerimos que, en esta situación, coloque la cantidad mínima de información de proceso y planificación dentro del Plan de Pruebas para facilitar el mantenimiento continuo y simplemente hacer referencia al material fuente "Maestro".

# Environmental Needs

Esta sección presenta los recursos no humanos requeridos para el Plan de prueba.

## Base System Hardware

| **System Resources** | | |
| --- | --- | --- |
| **Resource** | **Quantity** | **Name and Type** |
| Database Server | 1 | MySQL |
| —Network or Subnet |  | 192.168.41.0 |
| —Server Name |  | windows |
| —Database Name |  | Is\_puertadeacero |
| Client Test PCs |  | Windows 10 |
| —Include special configuration requirements |  | XAMPP |
| Test Repository |  | Github |
| —Network or Subnet |  | Github.com |
| —Server Name |  | github |
| Test Development PCs |  | Windows 10 |

## Base Software Elements in the Test Environment

The following base software elements are required in the test environment for this *Test Plan*.

[Note: Add or delete items as appropriate.]

| **Software Element Name** | **Version** | **Type and Other Notes** |
| --- | --- | --- |
| Windows 10 |  | Operating System |
| Internet Explorer |  | Internet Browser |
| MS Outlook |  | eMail Client software |
| Network Associates McAfee Virus Checker |  | Virus Detection and Recovery Software |

## Productivity and Support Tools

The following tools will be employed to support the test process for this *Test Plan*.

| **Tool Category or Type** | **Tool Brand Name** | **Vendor or In-house** | **Version** |
| --- | --- | --- | --- |
| Test Management | phpunit | opensource | 1.0 |
| Defect Tracking | phpunit | opensource |  |
| ASQ Tool for functional testing |  |  |  |
| ASQ Tool for performance testing |  |  |  |
| Test Coverage Monitor or Profiler |  |  |  |
| Project Management |  |  |  |
| DBMS tools |  |  |  |

## Test Environment Configurations

The following Test Environment Configurations needs to be provided and supported for this project.

| **Configuration Name** | **Description** | **Implemented in Physical Configuration** |
| --- | --- | --- |
| Average user configuration |  |  |
| Minimal configuration supported |  |  |
| Visually and mobility challenged |  |  |
| International Double Byte OS |  |  |
| Network installation (not client) |  |  |

# Responsibilities, Staffing, and Training Needs

Desarrolladores, testers con conocimientos en PHP, MySQL, HTML, Javascript y AJAX, Además del manejo de software XAMMP y sublime text con el Framework de pruebas PhpUnit

## People and Roles

| **Human Resources** | | |
| --- | --- | --- |
| **Role** | **Minimum Resources Recommended**  **(number of full-time roles allocated)** | **Specific Responsibilities or Comments** |
| Test Manager | 500 | Provides management oversight.  Responsibilities include:   * planning and logistics * agree mission * identify motivators * acquire appropriate resources * present management reporting * advocate the interests of test * evaluate effectiveness of test effort |
| Test Analyst | 400 | Identifies and defines the specific tests to be conducted.  Responsibilities include:   * identify test ideas * define test details * determine test results * document change requests * evaluate product quality |
| Test Designer | 400 | Defines the technical approach to the implementation of the test effort.  Responsibilities include:   * define test approach * define test automation architecture * verify test techniques * define testability elements * structure test implementation |
| Tester | 400 | Implements and executes the tests.  Responsibilities include:   * implement tests and test suites * execute test suites * log results * analyze and recover from test failures * document incidents |
| Test System Administrator | 400 | Ensures test environment and assets are managed and maintained.  Responsibilities include:   * administer test management system * install and support access to, and recovery of, test environment configurations and test labs |
| Database Administrator, Database Manager | 400 | Ensures test data (database) environment and assets are managed and maintained.  Responsibilities include:   * support the administration of test data and test beds (database). |
| Designer | 400 | Identifies and defines the operations, attributes, and associations of the test classes.  Responsibilities include:   * defines the test classes required to support testability requirements as defined by the test team |
| Implementer | 400 | Implements and unit tests the test classes and test packages.  Responsibilities include:   * creates the test components required to support testability requirements as defined by the designer |

## Staffing and Training Needs

Se necesitará capacitar al personal en las herramientas de pruebas de software antes mencionadas

# Iteration Milestones

| **Milestone** | **Planned Start Date** | **Actual Start Date** | **Planned End Date** | **Actual End Date** |
| --- | --- | --- | --- | --- |
| Iteration Plan agreed |  |  |  |  |
| Iteration starts |  |  |  |  |
| Requirements baselined |  |  |  |  |
| Architecture baselined |  |  |  |  |
| User Interface baselined |  |  |  |  |
| First Build delivered to test |  |  |  |  |
| First Build accepted into test |  |  |  |  |
| First Build test cycle finishes |  |  |  |  |
| [Build Two will not be tested] |  |  |  |  |
| Third Build delivered to test |  |  |  |  |
| Third Build accepted into test |  |  |  |  |
| Third Build test cycle finishes |  |  |  |  |
| Fourth Build delivered to test |  |  |  |  |
| Fourth Build accepted into test |  |  |  |  |
| Iteration Assessment review |  |  |  |  |
| Iteration ends |  |  |  |  |

# Risks, Dependencies, Assumptions, and Constraints

[List any risks that may affect the successful execution of this **Test Plan**, and identify mitigation and contingency strategies for each risk. Also indicate a relative ranking for both the likelihood of occurrence and the impact if the risk is realized.]

| **Risk** | **Mitigation Strategy** | **Contingency (Risk is realized)** |
| --- | --- | --- |
| Prerequisite entry criteria is not met. | <Tester> will define the prerequisites that must be met before Load Testing can start.  <Customer> will endeavor to meet prerequisites indicated by <Tester>. | * Meet outstanding prerequisites * Consider Load Test Failure |
| Test data proves to be inadequate. | <Customer> will ensure a full set of suitable and protected test data is available.  <Tester> will indicate what is required and will verify the suitability of test data. | * Redefine test data * Review Test Plan and modify * components (that is, scripts) * Consider Load Test Failure |
| Database requires refresh. | <System Admin> will endeavor to ensure the Database is regularly refreshed as required by <Tester>. | * Restore data and restart * Clear Database |

[List any dependencies identified during the development of this **Test Plan** that may affect its successful execution if those dependencies are not honored. Typically these dependencies relate to activities on the critical path that are prerequisites or post-requisites to one or more preceding (or subsequent) activities You should consider responsibilities you are relying on other teams or staff members external to the test effort completing, timing and dependencies of other planned tasks, the reliance on certain work products being produced.]

| **Dependency between** | **Potential Impact of Dependency** | **Owners** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

[List any assumptions made during the development of this **Test Plan** that may affect its successful execution if those assumptions are proven incorrect. Assumptions might relate to work you assume other teams are doing, expectations that certain aspects of the product or environment are stable, and so forth].

| **Assumption to be proven** | **Impact of Assumption being incorrect** | **Owners** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

[List any constraints placed on the test effort that have had a negative effect on the way in which this **Test Plan** has been approached.]

| **Constraint on** | **Impact Constraint has on test effort** | **Owners** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# Management Process and Procedures

[Outline what processes and procedures are to be used when issues arise with the **Test Plan** and its enactment.]

## Measuring and Assessing the Extent of Testing

[Outline the measurement and assessment process to be used to track the extent of testing.]

## Assessing the Deliverables of this Test Plan

[Outline the assessment process for reviewing and accepting the deliverables of this **Test Plan**]

## Problem Reporting, Escalation, and Issue Resolution

[Define how process problems will be reported and escalated, and the process to be followed to achieve resolution.]

## Managing Test Cycles

[Outline the management control process for a test cycle.]

## Traceability Strategies

[Consider appropriate traceability strategies for:

* Coverage of Testing against Specifications — enables measurement the extent of testing
* Motivations for Testing — enables assessment of relevance of tests to help determine whether to maintain or retire tests
* Software Design Elements — enables tracking of subsequent design changes that would necessitate rerunning tests or retiring them
* Resulting Change Requests — enables the tests that discovered the need for the change to be identified and re-run to verify the change request has been completed successfully]

## Approval and Signoff

[Outline the approval process and list the job titles (and names of current incumbents) that initially must approve the plan, and sign off on the plans satisfactory execution.]