UsacBook

Documento de Plan de Pruebas

Version 2.0

Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 14/12/2013 | 1.0 | Versión de preliminar del documento de Plan de Pruebas | Erick Dávila |
| 21/12/2013 | 2.0 | Inclusión de Pruebas de Aceptación, Sistema, Funcionales y Regresión | Erick Dávila |

Tabla de Contenido

[1. Introducción 4](#_Toc375330349)

[1.1 Propósito 4](#_Toc375330350)

[1.2 Alcance 4](#_Toc375330351)

[2. Estrategia de Pruebas 7](#_Toc375330352)

[3. Herramientas a Utilizar 8](#_Toc375330353)

[4. TIPOS DE PRUEBAS 9](#_Toc375330354)

[**4.1.1.** **PRUEBAS UNITARIAS** 9](#_Toc375330360)

[**4.1.2.** **PRUEBAS DE INTEGRACIÓN** 10](#_Toc375330361)

[**4.1.3.** **PRUEBAS DEL SISTEMA** 11](#_Toc375330362)

[**4.1.4.** **PRUEBAS DE RECUPERACIÓN A FALLAS** 12](#_Toc375330363)

[**4.1.5.** **PRUEBAS DE SEGURIDAD** 13](#_Toc375330364)

[**4.1.6.** **PRUEBAS FUNCIONALES** 13](#_Toc375330365)

[**4.1.7.** **PRUEBAS DE ACEPTACIÓN** 14](#_Toc375330366)

[**4.1.8.** **PRUEBAS DE REGRESIÓN** 14](#_Toc375330367)

[**4.1.9.** **PRUEBAS DE USABILIDAD** 15](#_Toc375330368)

[4.2 ENTREGABLES DE PRUEBAS 16](#_Toc375330369)

[5.1 RECURSO HUMANO 17](#_Toc375330372)

[5.2 RECURSO DEL SISTEMA 17](#_Toc375330373)

[5.3 ADMINISTRACIÓN DE VERSIONES 17](#_Toc375330374)

[**5.3.1.** **HERRAMIENTAS** 18](#_Toc375330378)

[6.1 CRITERIOS DE INICIO DE EJECUCIÓN 18](#_Toc375330381)

[6.2 CRITERIOS DE EVALUACION 18](#_Toc375330382)

Documento de Plan de Pruebas

# Introducción

El contenido de este documento de plan de pruebas hace parte integral de la metodologia de pruebas, el contenido de este documento se encuentran fundamentados en estándares de calidad que no solo permiten el seguimiento y correcciones a tiempo del software sino que además se encuentra definido por etapas, facilitando el seguimiento y control de los procesos del proyecto en desarrollo y proporcionando a la aplicación UsacBook la garantía del correcto funcionamiento de la solución desarrollada.

Acorde con el enfoque del desarrollo de la solución, el plan de pruebas está basado en la metodología de RUP, lo que hace que este plan de pruebas tenga como propósito establecer las técnicas, herramientas y actividades relacionadas con la ejecución y validación de cada una de las pruebas, incluyendo responsabilidades de cada una de las actividades, los recursos y los prerequisitos que deben ser considerados en el esfuerzo de cada una de las pruebas.

## Propósito

Este documento tiene como propósito establecer las técnicas, herramientas y actividades relacionadas con la ejecución y validación del plan de pruebas; incluye responsabilidades de cada una de las tareas, los recursos y los prerrequisitos que deben ser considerados en el esfuerzo de cada una de las pruebas, permitiendo garantizar el cumplimiento de los requerimientos planteados en el marco del desarrollo del proyecto denominado “UsacBook”.

## Alcance

Este documento de PLAN DE PRUEBAS DETALLADO, se convierte en una guía para desarrollar de una forma organizada las diferentes actividades que se realizarán en el proceso del plan de pruebas en el desarrollo del proyecto denominado “UsacBook”.

La metodología de pruebas y este documento de plan de pruebas permitirán al equipo profesionales expertos que participan en el frente de pruebas del proyecto denominado “UsacBook, evaluar aspectos como: la lógica estructural, la seguridad, la interconexión, el soporte conceptual, las herramientas de apoyo y sobretodo la independencia de aspectos técnicos del desarrollo de la solución tecnológica contratada, tales como: la plataforma tecnológica o la arquitectura de la solución a probar, sin embargo a continuación se describen las diferentes pruebas a ser aplicadas:

|  |  |  |
| --- | --- | --- |
| **TIPO DE PRUEBA** | **DEFINICIONES** | **FASE DE RUP** |
| UNITARIAS | **Unitarias:** Permite verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado. | **ELABORACIÓN** |
| INTEGRACIÓN | **Integración:** Permite verificar el correcto ensamblaje entre los distintos módulos que componen el sistema desarrollado. | **ELABORACIÓN** |
| SISTEMA:   * Carga * Volumen * Estress * Robustez * Concurrencia, * Interfaz de Usuario * Recuperación a Fallas * Rendimiento * Seguridad * Integridad de las BD * Interoperabilidad * Desempeño * Configuración | **Sistema:** Estas pruebas buscan diferencias entre la solución desarrollada y los requerimientos, con el fin de identificar errores que se puedan generar entre la especificación funcional y el diseño del sistema.  **Carga:** Valida aquellos volúmenes de datos máximos especificados en los requerimientos no Funcionales  **Volumen:** Esta prueba somete el software a grandes cantidades de datos para determinar si se alcanzan límites que causen la falla del software  **Estrés:** Valida aquellos volúmenes de datos máximos que resiste el sistema antes de comenzar con errores.  **Robustez:** Valida si el sistema se mantiene estable y consistente después de circunstancias adversas  **Concurrencia:** Valida la capacidad del sistema de atender múltiples solicitudes de parte de los usuarios que acceden a un mismo recurso.  **Interfaz de usuario**: Permite verificar que la navegación a través de los elementos que se están probando, reflejen las funciones del negocio y los requerimientos funcionales.    **Recuperación a fallas**: Estas pruebas aseguran que el que el software pueda recuperarse a fallas de hardware, software o mal funcionamiento de la red sin pérdida de datos o de integridad de los datos.  **Rendimiento:** Permite validar si la aplicación cumple los criterios de tiempos de respuesta establecidos.  **Seguridad**: Verifica el cumplimiento de las políticas de seguridad acordadas para el sistema.  **Integridad de las bases de datos:** Consiste en asegurar que los métodos y procesos de acceso a la base de datos funcionan correctamente y sin corromper datos.  **Interoperabilidad:** Esta prueba permite verificar todos los artefactos de la solución desarrollada, su arquitectura base, los protocolos de la solución, las interfaces y los módulos del sistema, funcionando en forma conjunta.  **Desempeño:** Este tipo de prueba es un aspecto fundamental en una aplicación, ya que si ésta no responde en el debido tiempo, se pueden perder clientes, o dañar la imagen ante los usuarios.  **Configuración:** Establece y mantiene la integridad de los productos de software a través del ciclo de vida del proceso del mismo. | **CONSTRUCCIÓN** |
| FUNCIONALES | **Funcional:** La prueba funcional es un proceso para procurar encontrar discrepancias entre el programa y la especificación funcional.  **Caja Negra:** Estas pruebas permiten obtener conjuntos de condiciones de entrada que ejecutan todos los requisitos funcionales de un programa.  **Ciclo de Negocio:** Esta prueba tiene por objeto garantizar que el proceso de negocio esta adecuadamente soportado por el software desarrollado y que éste dispone de la funcionalidad adecuada para ejecutar todas las tareas incorporadas en el proceso de negocio.  **Usabilidad:**  Esta prueba permite encontrar problemas de factores humanos, o usabilidad.  **Instalación:** Esta prueba permite verificar la instalación y desinstalación de la aplicación en diferentes entornos de hardware y software |
| ACEPTACIÓN | Es la prueba final basada en las especificaciones del usuario o basada en el uso del programa por el usuario final luego de un periodo de tiempo | **TRANSICIÓN** |
| REGRESIÓN | En esta prueba se valida que el sistema mantenga su correcta funcionalidad debido a la incorporación de un ajuste, corrección o nuevo requerimiento. Es una prueba funcional y técnica que valida que el sistema siga funcionando perfectamente después de que las correcciones sean aplicadas. |

# Estrategia de Pruebas

* 1. **TÉCNICAS DE ESPECIFICACIÓN DE LAS PRUEBAS**

La estrategia del proceso del plan de pruebas se implementara siguiendo un ciclo de vida que consta de cinco actividades esenciales para las pruebas de la aplicación a desarrollar.

* + 1. **CICLO DE PRUEBAS**

El ciclo de pruebas comprende cinco actividades las cuales deberán ser desarrolladas de la siguiente manera:

* + - 1. **PLANIFICACIÓN**

Para el desarrollo de la solución de UsacBook, es importante la ejecución del plan de pruebas, haciéndose necesario la planificación de las mismas, lo que en consecuencia hace necesario tener claro los siguientes planteamientos:

* Uso de herramientas.
* Criterios de aceptación.
* Recursos involucrados.

Se tendrá en cuenta que:

* Las pruebas estarán presentes a lo largo de todo el ciclo de vida del desarrollo, de la solución.
* Siempre hay errores.
* No es recomendable que el programador pruebe sus propios programas.
* Se puede disponer de herramientas.
* Se debe considerar la importancia de actualización del plan de pruebas con el fin de reflejar los cambios que se produzcan en los requisitos y/o proceso de desarrollo del producto.
  + - 1. **DISEÑO DE LAS PRUEBAS**

Para el diseño de las pruebas, se tendrán en cuenta aspectos que permitirán encontrar defectos en el periodo de desarrollo del software, la realización de pruebas propias y pruebas automatizadas.

Se realizaran pruebas automatizadas haciendo uso de herramientas que son compatibles con el IDE a utilizar y que se integran con él, esto con el fin de tener una forma concreta de demostrar las pruebas y por el hecho de la reutilización de las mismas, estas las realizaran los programadores.

Las pruebas propias serán realizadas por los miembros del equipo quienes harán el papel de tester y se encargaran de evaluar la funcionalidad de la aplicación e identificar puntos vulnerables que pueda tener.

* + - 1. **EJECUCIÓN**

Mientras se realiza la fase de implementación se van diseñando las pruebas del mismo nivel. En esta etapa se llevan a cabo las pruebas unitarias y al llegar a la etapa de pruebas se inicia la ejecución de lo diseñado desde las pruebas unitarias hasta las pruebas funcionales.

* + - 1. **EVALUACIÓN Y CIERRE**

Para la evaluación y cierre de las pruebas se presentará el informe de pruebas donde se documentará el resultado de cada una de las diferentes pruebas ejecutadas. El informe debe presentarse de manera formal, esto ya que el informe de pruebas es un entregable independiente.

* + - 1. **SEGUIMIENTO Y CONTROL**

Para el seguimiento y control de las pruebas se nombraran al encargado quien se encargara de la gestión del informe de pruebas e informaran al resto del equipo de trabajo los resultados y si es necesario convocar a reuniones para tratar temas críticos sobre los estados de las pruebas.

Los puntos a cubrir serian:

* Estado y resultado de las pruebas realizadas
* Estado del proyecto según las fechas indicadas en la planificación de actividades
* Seguimiento a las incidencias reportadas según la ejecución de pruebas.

# Herramientas a Utilizar

**2.1.1. PHPUNIT**

|  |  |
| --- | --- |
| **Características** | **Tipo de prueba** |
| PHPUnit es un entorno para realizar pruebas unitarias en el lenguaje de programación PHP. El objetivo de las pruebas unitarias es aislar cada parte del programa y demostrar que las partes de forma individual son correctas. Una prueba unitaria proporciona un contrato escrito que la pieza de código debe satisfacer. Como resultado, las pruebas unitarias encuentran problemas en las fases iniciales del desarrollo de software.  En la actualidad las herramientas de desarrollo como NetBeans y Eclipse cuentan con plugins que permiten que la generación de las plantillas necesarias para la creación de las pruebas se realice de manera automática, facilitando al programador enfocarse en la prueba y el resultado esperado, y dejando a la herramienta la creación de las clases que permiten coordinar las pruebas. La versión de la herramienta que se utilizará será: release 3.6.10 | Esta herramienta será utilizada para la ejecución de:   * Pruebas unitarias. * Pruebas de Sistema. * Pruebas de Integración. * Pruebas de Seguridad * Pruebas de Aceptación |

* + 1. **JENKINS**

|  |  |
| --- | --- |
| **Características** | **Tipo de prueba** |
| Jenkins es una herramienta utilizada para la integración continua que permite realizar tareas periódicas y de manera automática como generar y compilar el proyecto.  Detecta problemas de forma temprana haciendo que sea más fácil la programación entre un amplio número de personas de manera simultánea. | Esta herramienta será utilizada para la ejecución de:   * Pruebas de Integración Continua |

# TIPOS DE PRUEBAS

Las pruebas que se realizarán son las necesarias para garantizar la calidad de la aplicación.

El objetivo principal de la ejecución de las pruebas esta dado a:

• Descubrir todos los errores posibles.

• Identificar rápidamente inconsistencias

• Identificar falencias funcionales de la aplicación, enmarcadas en grados de usabilidad ya definidos.

• Evaluar la calidad del producto y señalar un indicador de aceptación del mismo.

• Asegurar el cumplimiento de los requisitos definidos.

• Cumplir con los requerimientos específicos del cliente, en cuanto a la ejecución de las pruebas.

4. 1. 1. **PRUEBAS UNITARIAS**

Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente, función, método, etc. individualmente del sistema una vez que ha sido codificado.

Es una Prueba técnica que permitirá:

* Verificar que los módulos carecen de errores
* Que todos los caminos lógicos principales deben ejecutarse correctamente en cada módulo de la aplicación.
* Probar todas las transacciones de la aplicación.
* Todos los tipos de registro de entrada válidos deben ser procesados
* Todos los tipos de registro de entrada inválidos deben ser procesados correctamente
* Códigos de vuelta no nulos.
* Excepciones a tratamiento normal.
* Todas las salidas válidas son procesadas.
* Rasgos de Control son probados y documentados.

|  |  |
| --- | --- |
| Objetivo de la Prueba: | * Validar las piezas individuales del software como una unidad independiente. |
| Estrategia: | * Generar casos de pruebas necesarios que permitan identificar:   + Que al menos cada sentencia o instrucción del programa se ejecute al menos una vez correctamente.   + Que cada condición tenga por lo menos una vez un resultado positivo y/o negativo.   + Que las transacciones a la BDD sean correctas. |
| Herramienta requeridas: | PHPUnit |
| Observaciones | La prueba se realizará por Módulo entendiéndose por tal:   * Bloque básico de programa * Implementa función independiente y simple * Puede probarse por separado. |

* + 1. **PRUEBAS DE INTEGRACIÓN**

El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos módulos que componen la solución una vez que han sido probados unitariamente provocadas por la programación de varias personas en el mismo proyecto, con el fin de comprobar que los módulos interactúan correctamente a través de sus interfaces internas y externas, que cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes.

Estas partes pueden ser módulos, aplicaciones individuales, es decir esta prueba válida la integración entre los diferentes módulos que componen la solución con el fin de garantizar que su operación integrada es correcta, teniendo en cuenta los siguientes temas técnicos:

• El funcionamiento integrado de módulos interdependientes debe estar libre de errores

• Probar todas las dependencias entre módulos

• Probar el flujo de control y el flujo de datos a través de todas las capas

|  |  |
| --- | --- |
| Objetivo de la Prueba: | Validar la integración entre los diferentes módulos que componen la solución con el fin de garantizar que su operación integrada es correcta |
| Estrategia: | Pruebas de Integración   * Combinación de módulos de bajo nivel en grupos que realicen una misma función o sub función específica, con el fin de reducir el número de pasos de integración. * Se escribe para cada módulo un módulo impulsor o conductor, con el fin de simular la llamada a los módulos, introducir datos de pruebas y recoger resultados. * Se prueba cada módulo mediante su impulsor. * Se eliminan los módulos impulsores y se sustituyen por los módulos de nivel superior en la jerarquía. |
| Herramienta requeridas: | JENKINS |
| Observaciones: |  |

* + 1. **PRUEBAS DEL SISTEMA**

Las pruebas de sistema buscan diferencias entre la solución desarrollada y los requerimientos, enfocándose en la identificación de los errores que se puedan generar entre la especificación funcional y el diseño del sistema, así como, el negocio objeto de la aplicación.

|  |  |
| --- | --- |
| Objetivo de la Prueba: | * Validar aquellos volúmenes de datos máximos (por lo general las transacciones o informes) que pueden ser completados dentro de un período específico en el tiempo, y con un nivel de concurrencia dado (carga, concurrencia y desempeño). * Validar los requerimientos no funcionales de cada proyecto. |
| Estrategia: | * Realizar Set de Pruebas a partir de los Requerimientos no funcionales. * Realizar pruebas de rendimiento básico. Consiste en probar la aplicación simulando la carga esperada en el entorno de producción. * Realizar las pruebas de concurrencia: verificar el comportamiento de la aplicación en condiciones de sobrecarga de usuarios, que supone permitirá identificar potenciales problemas de rendimiento o cuellos de botella, antes de su pase a producción. * Realizar pruebas de requerimientos no funcionales: Consiste en probar la aplicación con cada uno de los requerimientos no funcionales establecidos en el proyecto. * Identificar posibles cuellos de botella o problemas de rendimiento. * Realizar pruebas de carga: Altos volúmenes de información. |
| Herramienta requeridas: | PHPUNIT |
| Observaciones: | Estas pruebas se emularan mediante métodos creados dentro de pruebas unitarias. |

* + 1. **PRUEBAS DE RECUPERACIÓN A FALLAS**

Estas pruebas aseguran que el software pueda recuperarse a fallas de hardware, software o mal funcionamiento de la red sin pérdida de datos o de integridad de los datos. El objetivo de esta prueba es verificar que los procesos de recuperación (manual o automáticos) se realice apropiadamente: las base de datos, las aplicaciones y el sistema a un estado conocido y deseado.

En la prueba se incluyen los siguientes tipos de condiciones:

* Interrupción de energía al cliente
* Interrupción de energía al servidor
* Interrupción de comunicaciones mediante los servidores de la red
* Interrupción de comunicación o pérdida de energía de los discos del servidor o con los controladores
* Ciclos incompletos (procesos de filtro de datos interrumpidos, procesos de sincronización de datos interrumpidos)

|  |  |
| --- | --- |
| Objetivo de la Prueba: | * Validar la capacidad de recuperación a fallas de:   + Hardware   + Software   + Mal funcionamiento de Red. |
| Estrategia : | * Interrumpir la energía del cliente: apagar el PC. * Interrumpir la energía del servidor: simular o iniciar el proceso de apagado del servidor. * Interrupción por medio de los servidores de red: simular o iniciar la pérdida de comunicación con la red * (desconectar físicamente la comunicación o apagar el servidor de red o router) * Interrumpir la comunicación o quitar la energía de los discos del servidor o sus controladores: * simular o eliminar físicamente la comunicación con uno o más controladores de disco o los discos.] * Una vez que se lograron o simularon estas condiciones, se deben invocar los procedimientos de * recuperación. |
| Herramientas Requeridas: | Ninguna |
| Observaciones: |  |

* + 1. **PRUEBAS DE SEGURIDAD**

Estas pruebas tienen dos enfoques:

* Pruebas de seguridad de la aplicación; donde se verifica que un actor solo pueda acceder a las funciones y datos que su usuario tiene permitido.
* Pruebas de seguridad del sistema; donde se verificar que solo los actores con acceso al sistema y a la aplicación están habilitados para accederla.

|  |  |
| --- | --- |
| Objetivo de la Prueba: | * Que los usuarios están restringidos a funciones específicas o su acceso está limitado únicamente a los datos que está autorizado a acceder. * Que solo aquellos usuarios autorizados a acceder al sistema son capaces de ejecutar las funciones del sistema. * Que el cortafuego oculte apropiadamente la aplicación. * Que los puertos clave solo puedan ser accedidos de la forma establecida para su acceso. * Que los puertos restringidos efectivamente no se encuentren accesibles. |
| Estrategia : | * Identificar cada tipo de usuario y las funciones y datos a los que se debe autorizar. * Crear pruebas para cada tipo de usuario y verificar cada permiso, creando transacciones específicas para cada tipo de usuario. * Modificar tipos de usuarios y volver a ejecutar las pruebas. |
| Herramientas Requeridas: | * Nessus * Pruebas funcionales de seguridad. |
| Observaciones: | Para el reporte de incidencias se utilizará una herramienta para el registro y seguimiento. |

* + 1. **PRUEBAS FUNCIONALES**

La prueba funcional es un proceso para procurar encontrar discrepancias entre el software desarrollado y la especificación funcional. La prueba funcional normalmente es una actividad de caja negra. Esta prueba permite validar:

* Los procesos y reglas de negocio establecidas,
* Que se cumplan los requerimientos funcionales establecidos

En esta prueba se validan los Casos de Uso que fueron aprobados por el cliente, y a partir de ellos se diseñan y ejecutan los set de pruebas correspondientes. Se deben elaborar los casos de pruebas necesarios que permitan asegurar el funcionamiento de todos los flujos normales y alternos de dichos casos de uso.

|  |  |
| --- | --- |
| Objetivo de la Prueba: | Se asegura el trabajo apropiado de los requisitos funcionales, Incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados. |
| Estrategia : | * Validación y ejecución de Set de Pruebas y escenarios definidos, teniendo en cuenta flujo normal y flujos alternativos, usando datos validos e inválidos para verificar lo siguiente:   + Los resultados esperados ocurren cuando se usan datos válidos.   + Se despliegan mensajes de error cuando se usan datos inválidos.   + Cada regla de negocio es propiamente aplicada.   + Realizar set de pruebas de los requerimientos mínimos para el adecuado funcionamiento de la aplicación |
| Herramientas Requeridas: | * PHPUNIT cuando aplique * Formato de casos de prueba funcionales |
| Observaciones: |  |

* + 1. **PRUEBAS DE ACEPTACIÓN**

El objetivo de las pruebas de aceptación es validar que la solución desarrollada cumpla con el funcionamiento esperado y permitir al usuario de dicho sistema determine su aceptación, desde el punto de vista de su funcionalidad y de su rendimiento. Estas pruebas son realizadas por el cliente, donde comprueba que el sistema cumple con lo definido y se obtiene la conformidad del cliente. Esta prueba se realiza mediante el proceso de validación de caja negra.

Estas pruebas corresponden a la ejecución de las siguientes pruebas por parte de los usuarios funcionales o cliente:

* Pruebas Funcionales.
* Pruebas de Usabilidad.
  + 1. **PRUEBAS DE REGRESIÓN**

En esta prueba se valida que el sistema mantenga su correcta funcionalidad después de la incorporación de un ajuste, corrección o nuevo requerimiento. Es una prueba funcional y técnica que valida que el sistema siga funcionando perfectamente después de que las correcciones sean aplicadas.

|  |  |
| --- | --- |
| Objetivo de la Prueba: | Validar que el sistema siga funcionando perfectamente después de que las acciones correctivas sean aplicadas. |
| Estrategia: | * Repetir las pruebas (unitarias, de integración, funcionales y de carga) que se hicieron antes de corregir defectos o de añadir nuevas funcionalidades, para comprobar que las modificaciones no provocan errores donde antes no los había. |
| Herramienta requeridas: | * Utilizar las mismas herramientas usadas para las pruebas según sea el caso. |
| Observaciones | Los responsables de las Pruebas de Regresión se establecen dependiendo del momento en el que se realicen las modificaciones. |

* + 1. **PRUEBAS DE USABILIDAD**

Las pruebas de usabilidad consisten en seleccionar a un grupo de usuarios de una aplicación y solicitarles que lleven a cabo las tareas para las cuales fue diseñada, en tanto el equipo de diseño, desarrollo y otros involucrados toman nota de la interacción, principalmente de los errores y dificultades con las que se encuentren los usuarios para resolverlas lo más pronto posible.

No es necesario que se trate de una aplicación completamente terminada, pudiendo tratarse de un prototipo.

|  |  |
| --- | --- |
| Objetivo de la Prueba: | * Validar el grado de usabilidad empírico del sistema. * El grado de usabilidad se medirá en tres aspectos clave: * Facilidad de aprendizaje: facilidad con la que nuevos usuarios desarrollan una interacción efectiva con el sistema o producto. * Flexibilidad: relativa a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. * Robustez: es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos. |
| Estrategia : | * Se usarán cuatro métricas principales para medir la usabilidad del sistema * [Exactitud](http://es.wikipedia.org/wiki/Exactitud): Número de errores cometidos por los sujetos de prueba y si estos fueron recuperables o no al usar los datos o procedimientos adecuados. * [Tiempo](http://es.wikipedia.org/wiki/Tiempo) requerido para concluir la actividad. * [Recuerdo](http://es.wikipedia.org/wiki/Recuerdo): Qué tanto recuerda el usuario después de un periodo sin usar la aplicación. * Respuesta [emocional](http://es.wikipedia.org/wiki/Emoci%C3%B3n): Cómo se siente el usuario al terminar la tarea (bajo tensión, satisfecho, molesto, etcétera). * Estas métricas será implementadas para cada uno de los aspectos clave señalados en el objetivo de la prueba. * La forma de evaluación será mediante el uso de encuestas; donde cada pregunta evaluará un aspecto clave de usabilidad y aportará valor a una o varias métricas dentro del aspecto clave evaluado. * Las encuestas se realizarán a los usuarios utilizando los prototipos del sistema; para así poder realizar cambios de forma temprana al diseño de la capa de presentación. |
| Herramientas Requeridas: | * Encuesta * Prototipos del sistema. |
| Observaciones: |  |

## ENTREGABLES DE PRUEBAS

De acuerdo al tipo de pruebas ejecutadas puede que el entregable del mismo sea diferente, en el siguiente cuadro se señalan los diferentes entregables por tipo de prueba.

|  |  |
| --- | --- |
| **TIPO DE PRUEBAS** | **ENTREGABLES** |
| Pruebas Unitarias | * Resumen de validación de la prueba. |
| Pruebas de Sistema | * Se entregará un documento de resultados de las pruebas realizadas, que incluya resultados de la ejecución de los scripts de pruebas y análisis de los errores o defectos encontrados durante el proceso de realización de las pruebas. |
| Pruebas de Integración | * Se entregará un documento de pruebas de integración que incluye resultados de la ejecución de los scripts de pruebas y análisis de los defectos encontrados durante el proceso de pruebas |
| Pruebas de Regresión | * Se entregará un documento de pruebas de regresión, que incluye resultados de la ejecución de los scripts de pruebas, análisis de los defectos encontrados durante el proceso de pruebas y solicitud de las correcciones recibidas. |
| Pruebas Funcionales | * Se entregará un documento de pruebas de regresión, que incluye resultados de la ejecución de los scripts de pruebas y análisis de los defectos encontrados durante el proceso de pruebas y solicitud de las correcciones recibidas. |
| Pruebas de Usabilidad | * Indicadores de Usabilidad |
| Pruebas de Seguridad | * Informe de vulnerabilidades * Resultado de pruebas funcionales de seguridad. |
| Pruebas de Recuperación a Fallas | * Resumen de validación de la prueba. |
| Pruebas de Aceptación | * Resumen de validación de la prueba. |



## RECURSO HUMANO

|  |  |
| --- | --- |
| **TIPO DE PRUEBAS** | **PERFIL DEL RECURSO HUMANO** |
| Pruebas Unitarias | * Ingeniero Desarrollador. * Analista de Pruebas. |
| Pruebas de Sistema | * Ingeniero Desarrollador. * Analista de Pruebas. |
| Pruebas de Integración | * Ingeniero Desarrollador. * Analista de Pruebas |
| Pruebas de Regresión | * Ingeniero Desarrollador. * Analista de Pruebas |
| Pruebas Funcionales | * Analista de Pruebas |
| Pruebas de Usabilidad | * Analista de Pruebas * Usuario Funcional. |
| Pruebas de Seguridad | * Ingeniero Desarrollador. * Analista de Pruebas * Arquitecto de Desarrollo. |
| Pruebas de Recuperación a Fallas | * Ingeniero Desarrollador. * Analista de Pruebas |
| Pruebas de Aceptación | * Analista de Pruebas. * Usuario Funcional. |

## RECURSO DEL SISTEMA

A continuación se describen las características de la infraestructura del ambiente de pruebas.

|  |  |  |
| --- | --- | --- |
| **DESCRIPCION** | **FUNCIONALIDAD** | **CANTIDAD** |
| Servidor | Montar ambiente de Pruebas con la solución en proceso de desarrollo | 3 |
| Estaciones de Trabajo | Con acceso al Servidor de Pruebas a través de la red HAMACHI | 3 |
| Herramientas de pruebas de sistemas | PHPUNIT, JENKINS | 1 |

## ADMINISTRACIÓN DE VERSIONES

La administración de versiones que se probarán será el mecanismo ideal, para controlar los release de pruebas y los cambios que estos sufrirán en la etapa de corrección de incidencias reportadas.

De acuerdo a lo anterior la administración de versiones contempla las siguientes etapas:

1. **Entrega de la Versión para Pruebas** (Release)
   * La versión debe venir con el Release Note.
   * Lista de Chequeo (si aplica)
2. **Creación de Incidencias en la herramienta**.
   * Se debe especificar a que Release pertenecen las incidencias.
3. **Análisis y Desarrollo de Incidencias**.
   * Se realiza la clasificación de las incidencias.
   * Se empieza el desarrollo de las correcciones.
   * Se integran los desarrollos en el release correspondiente
4. **Pruebas de Correcciones de Incidencias** según el release.
5. **Aprobación del Release**.


   3. 1. **HERRAMIENTAS**

**SUBVERSION**

Es un software de sistema de control de versiones, Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo

Ventajas

* Se sigue la historia de los archivos y directorios a través de copias y renombrados.
* Las modificaciones (incluyendo cambios a varios archivos) son atómicas.
* La creación de ramas y etiquetas es una operación más eficiente.
* Se envían sólo las diferencias en ambas direcciones
* Maneja eficientemente archivos binarios.
* Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.



## CRITERIOS DE INICIO DE EJECUCIÓN

A continuación se señalan las condiciones mínimas que se deben presentar para iniciar la ejecución de las pruebas:

* Se poseen los set de pruebas aprobadas con escenarios claros.
* El entorno de pruebas es el adecuado para el tipo de pruebas a iniciar.
* Todos los artefactos requeridos se encuentran disponibles.
* Se recibió la Versión del Software para pruebas con su correspondiente Release Note y Lista de Chequeo cuando esta aplique.
* Todos los recursos humanos y técnicos necesarios se encuentran disponibles.

## CRITERIOS DE EVALUACION

Los criterios de evaluación estarán dados de forma independiente para cada tipo de pruebas; el siguiente cuadro muestra los criterios de evaluación generales de las pruebas ejecutadas.

|  |  |
| --- | --- |
| **TIPO DE PRUEBAS** | **CRITERIOS DE EVALUACION** |
| Pruebas Unitarias | * Detectar errores en la ejecución de las pruebas. * El 90% de las pruebas realizadas deben ser exitosas. |
| Pruebas de Sistema | * Detectar errores en la ejecución de las pruebas * Que los reportes generados contengan las mínimas variables que permitan un análisis acertado de cada una de las pruebas realizadas. * Tener en cuenta todos los escenarios posibles. * El 90% de las pruebas realizadas deben ser exitosas. |
| Pruebas de Integración | * La totalidad de los puntos de control probadas debe ser mayor al 75% del total de los componentes que integran la solución. * Detectar errores en la ejecución de las pruebas * El 90% de las pruebas realizadas deben ser exitosas. |
| Pruebas de Regresión | * Para realizar esta prueba se debe tomar como base los criterios de aceptación de las pruebas que se volverán a realizar. |
| Pruebas Funcionales | * El resultado de cada caso de prueba debe ser igual al resultado de salida esperado. * Encontrar fallas al ejecutar los diferentes casos de pruebas. * La aplicación cumple con los requerimientos funcionales especificados en la fase de análisis * La aplicación cumple con los requerimientos mínimos para el funcionamiento |
| Pruebas de Usabilidad | * El resultado de cada caso de prueba debe ser igual al resultado de salida esperado. * Se deben incluir los datos de entrada válidos y esperados como no validos e inesperados Encontrar los errores al ejecutar los diferentes casos de pruebas. * La aplicación debe cumplir con los requerimientos funcionales especificados en la fase de análisis. * La aplicación debe cumplir con los requerimientos mínimos para el funcionamiento. |
| Pruebas de Seguridad | * El resultado de cada caso de prueba debe ser igual al resultado de salida esperado. * La aplicación debe cumplir con los requerimientos mínimos de seguridad. |
| Pruebas de Recuperación a Fallas | * Considerar todos los escenarios posibles * Qué el sistema funcione de acuerdo a lo esperado después de las pruebas. |
| Pruebas de Aceptación | * Para realizar esta prueba se debe tomar como base los criterios de aceptación de las pruebas que se volverán a realizar. |

Para cada una de las pruebas se tendrá en cuenta:

* **Pruebas Unitarias:** Las pruebas unitarias se evalúan por medio de la siguiente tabla o lista de chequeo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Elemento a Revisar | SI | NO | No Aplica | Observaciones |
| ¿Se realizaron las Pruebas Unitarias con alguna herramienta especializada? |  |  |  |  |
| ¿Con las pruebas realizadas, cuál fue el porcentaje de cobertura del sistema? |  |  |  |  |
| ¿Existe constancia de la realización de las pruebas mencionadas? |  |  |  |  |
| ¿El funcionamiento de la prueba unitaria respeta el diseño establecido? |  |  |  |  |
| ¿Existe un manejo de errores adecuado? |  |  |  |  |
| ¿Se cumplió con la estrategia de ejecución de la prueba? |  |  |  |  |

* **Pruebas del Sistema:** El resultado de las pruebas del sistema se verá reflejado en el siguiente informe:

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de Uso** | <Identificador del Caso de uso> | **Descripción del escenario** | <Número total de casos de prueba ejecutados de acuerdo al escenario> |
| **Número de pruebas exitosas** | <Del total de pruebas ejecutadas, cuantas pruebas fueron exitosas> | **Número de pruebas Fallidas** | <Del total de pruebas ejecutadas, cuantas pruebas fueron fallidas> |
| **Tiempo máximo de ejecución de una prueba** | <Tiempo máximo que duró en ejecución una Prueba> | **Tiempo Promedio de ejecución de las pruebas** | <Tiempo promedio de ejecución de las pruebas> |
| **Número de peticiones exitosas** | <Número de peticiones http exitosas> | **Número de Peticiones Fallidas** | <Número de peticiones http fallidas> |
| **Número de Errores** | <Numero de errores ocurridos durante las pruebas> | **Tipo de errores** | <Descripción del tipo de errores presentados> |
| **% de Utilización del Procesador** | <Porcentaje de consumo de utilización de CPU durante la ejecución de la prueba> | **Cantidad de Memoria utilizada** | <Cantidad de MB de memoria utilizada en la prueba> |
| **Promedio de bytes enviados** | <Promedio de bytes enviados> | **Promedio de bytes recibidos** | <Promedio de bytes recibidos> |

* **Pruebas de Integración:** El resultado de las pruebas de integración se verá reflejado en el siguiente informe o lista de chequeo:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Elemento a Revisar** | **SI** | **NO** | **No Aplica** | **Observaciones** |
| ¿Se realizaron las pruebas de Integración con alguna herramienta especializada? |  |  |  |  |
| ¿Cuál fue el porcentaje de cobertura de la prueba con relación al sistema total? |  |  |  |  |
| ¿Existe constancia de la ejecución de las pruebas? |  |  |  |  |
| ¿Qué capas o componentes de la arquitectura se cubrió con la ejecución de las pruebas? |  |  |  |  |
| ¿Se estableció un criterio para la ejecución de las pruebas? ¿Cuál? |  |  |  |  |
| ¿Se cumplió la estrategia de ejecución de la prueba? |  |  |  |  |

* **Pruebas de Regresión:** El resultado de las pruebas de regresión se verá reflejado de acuerdo a los tipos de pruebas seleccionados.
* **Pruebas Funcionales:** El resultado de las pruebas funcionales se verá reflejado de acuerdo al formato de set de pruebas, ver anexos.
* **Pruebas de interfaz de Usuario:** El resultado de las pruebas de interfaz de usuario se verá reflejado en el siguiente informe o lista de chequeo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Elemento a Revisar** | **SI** | **NO** | **No Aplica** | **Observaciones** |
| ¿Se realizaron las pruebas de interfaz de usuario con alguna herramienta especializada? |  |  |  |  |
| ¿Cuál fue el porcentaje de cobertura de la prueba con relación al sistema total? |  |  |  |  |
| ¿Existe constancia de la ejecución de las pruebas? |  |  |  |  |
| ¿Qué páginas se cubrió con la prueba? |  |  |  |  |
| ¿Se estableció un criterio para la ejecución de las pruebas? ¿Cuál? |  |  |  |  |
| ¿Se cumplió la estrategia de ejecución de la prueba? |  |  |  |  |

* **Pruebas de Seguridad:** El resultado de las pruebas de seguridad se verá reflejado en el siguiente informe o lista de chequeo:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Elemento a Revisar** | **SI** | **NO** | **No Aplica** | **Observaciones** |
| ¿Se realizaron las pruebas de seguridad con alguna herramienta especializada? |  |  |  |  |
| ¿Cuál fue el porcentaje de cobertura de la prueba con relación al sistema total? |  |  |  |  |
| ¿Existe constancia de la ejecución de las pruebas? |  |  |  |  |
| ¿Qué capas o componentes de la arquitectura se cubrió con la ejecución de las pruebas? |  |  |  |  |
| ¿Se estableció un criterio para la ejecución de las pruebas? ¿Cuál? |  |  |  |  |
| ¿Se cumplió la estrategia de ejecución de la prueba? |  |  |  |  |

* **Pruebas de Recuperación a Fallas:** El resultado de las pruebas de Recuperación a Fallas se verá reflejado en el siguiente informe o lista de chequeo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Elemento a Revisar** | **SI** | **NO** | **No Aplica** | **Observaciones** |
| ¿Se realizaron las pruebas de recuperación a fallas con alguna herramienta especializada? |  |  |  |  |
| ¿Cuál fue el porcentaje de cobertura de la prueba con relación al sistema total? |  |  |  |  |
| ¿Existe constancia de la ejecución de las pruebas? |  |  |  |  |
| ¿Qué páginas se cubrió con la prueba? |  |  |  |  |
| ¿Se estableció un criterio para la ejecución de las pruebas? ¿Cuál? |  |  |  |  |
| ¿Se cumplió la estrategia de ejecución de la prueba? |  |  |  |  |

* **Pruebas de Aceptación:** El resultado de las pruebas de aceptación se verá reflejado de acuerdo a los tipos de pruebas seleccionados.

1. **PLANTILLA DE PRUEBAS**

|  |  |
| --- | --- |
| ID Caso de Prueba | Identificador único para el caso de prueba |
| Nombre Caso de Prueba | Aspecto a probar |
| Descripción | Se probara la respuesta del sistema cuando se presenta X escenario… |
| Precondiciones | Condiciones que se deben cumplir para la ejecución del caso de prueba |
| Relaciones CDU | Indicar si el caso de uso al cual se le están derivando los casos de prueba tiene conexiones con casos de uso mediante relaciones de include, extend o generalización |
| Pasos y Condiciones de Ejecución | Pasos detallados del caso de prueba y los datos con los que se probara. |
| Resultado Esperado | Resultado ideal de la aplicación de acuerdo a los pasos ejecutados. |
| Estado Caso de Prueba | Ejecutado: Exitoso, Fallido, Frenado  Pendiente de ejecución  En Construcción |
| Resultado Obtenido | Se diligencia luego de la ejecución del caso de prueba y de acuerdo a la reacción de la aplicación. |
| Errores Asociados | Al ejecutar el caso de prueba, en este campo se indican los identificadores únicos de los errores presentados |
| Responsable de Diseño | Nombre del encargado que diligencio la plantilla |
| Responsable de Ejecución | Nombre del Tester que ejecutó el caso de prueba. |
| comentarios |  |