**MUCZ**

**Sistema Automatizado de Gestión de** **Solicitudes de Servicio de la Dirección de** **Planta Física USB**

Plan de Pruebas

Versión 1.0

Historia de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 25/02/2016 | 1.0 | Versión Inicial | MCUZ |
| 18/03/2016 | 1.0 | Plan de Pruebas final | MUCZ |
|  |  |  |  |
|  |  |  |  |

Tabla de Contenidos

***Plan de Pruebas* 5**

**1. Introducción 5**

**1.1 Propósito 5**

**1.2 Alcance 5**

**1.3 Audiencia a la que se orienta 6**

**1.4 Terminología del Documento y Acrónimos 6**

**1.5 Referencias 7**

**1.6 Estructura del Documento 7**

**2. Propósito de la evaluación y motivación de las pruebas 7**

**2.1 Antecedentes 7**

**2.2 Propósito de la Evaluación 7**

**2.3 Motivadores de la Prueba 8**

**3. Aspectos a los que se enfoca la prueba 8**

**4. Delimitación de las pruebas previstas 9**

**4.1 Pruebas a Incluir 9**

**4.2 Pruebas a Excluir 10**

**5. Enfoque de las Pruebas 10**

**5.1 Ideas Iniciales y otras Fuentes de Referencia 10**

**5.2 Técnicas y Tipos de Prueba 12**

**6. Criterios de Entrada y Salida 14**

**6.1 Plan de Pruebas 14**

**6.1.1 Criterios de Entrada 14**

**6.1.2 Criterios de Salida 14**

**6.1.3 Criterios de Suspensión y Retomar Actividad 14**

**6.2 Ciclos de Prueba 14**

**6.2.1 Criterios de Entrada 14**

**6.2.2 Criterios de Salida 14**

**6.2.3 Finalización Anormal 14**

**7. Entregables 15**

**7.1 MECAP – Casos de Pruebas 15**

**7.2 Resumen del ciclo de pruebas 15**

**7.3 Reportes de calidad percibida 15**

**7.4 Pruebas de Regresión y soporte de los Scripts de Prueba 15**

**8. Flujos de Trabajo de Pruebas 15**

**9. Necesidades de ambiente 16**

**9.1 Hardware 16**

**9.2 Software 16**

**9.3 Herramientas de Productividad y Soporte 16**

**9.4 Configuraciones del Ambiente de Pruebas 17**

**10. Responsabilidades, Personal y necesidades de entrenamiento 19**

**10.1 Personal y Roles 19**

**10.2 Personal y Necesidades de entrenamiento 20**

**11. Puntos de control en la iteración 20**

**12. Riesgos, Dependencias, Suposiciones y Restricciones 21**

**13. Procedimientos y Gerencia de Procesos 22**

**13.1 Midiendo y determinando el grado de la prueba 22**

**13.2 Determinación de los entregables de éste plan de pruebas 23**

**13.3 Reportes del problema, escalada y resolución 23**

**13.4 Gerenciar Ciclos de Prueba 23**

**13.5 Estrategias de trazabilidad 23**

**13.6 Aprobación y Finalización 23**

Plan de Pruebas

1. Introducción

El presente documento tiene como finalidad listar, definir y analizar el plan con las distintas pruebas que se realizará al sistema SIAGES. A medida que avance el documento se describirá las pruebas que se ejecutarán en cada iteración, estas incluirán el su enfoque y criterios que determinarán si el segmento probado del sistema cumple con los requerimientos o por el contrario se detecte algún fallo que requiera ser corregido.

* 1. Propósito

El propósito del plan de prueba es recopilar toda la información necesaria para planear y controlar el esfuerzo de la prueba para una iteración dada. Describe el enfoque para probar el software y es el plan a nivel superior generado y usado por los encargados para dirigir el esfuerzo de la prueba.

Este plan de prueba para SIAGES apoya los siguientes objetivos:

* Identifica los aspectos a los que se deben orientar las pruebas
* Identifica la motivación y las ideas que respaldan las pruebas que se realizarán.
* Delimitan el enfoque de las pruebas que será usado
* Identifica los recursos requeridos y provee un estimado del esfuerzo
* Lista los entregables de las pruebas del proyecto.
  1. Alcance

En este documento se describen los detalles de los distintos tipos de pruebas a ser aplicadas. Se exponen también las metodologías a utilizar en cada una de estas. Las pruebas que serán realizadas son:

* ***Pruebas Unitarias:*** Tienen por objeto probar que los fragmentos individuales o unidades del sistema cumplen las especificaciones y presentan el comportamiento esperado.
* ***Pruebas de Integración:*** Se prueban las funcionalidades, rendimiento y fiabilidad del sistema una vez integrados sus diferentes módulos y las relaciones del sistema con el exterior.
* ***Pruebas de Aceptación:*** Representan el proceso de revisión que verifica que el sistema de software producido cumple con las especificaciones y que logra su cometido. Es normalmente una parte del proceso de pruebas de software de un proyecto, que también utiliza técnicas tales como evaluaciones, inspecciones y tutoriales. La validación es el proceso de comprobar que lo que se ha especificado es lo que el usuario realmente quería. Se trata de evaluar el sistema o parte de este durante o al final del desarrollo para determinar si satisface los requisitos iniciales.
  1. Audiencia a la que se orienta

Este plan de pruebas está dirigido a todas aquellas personas involucradas en la planeación, aprobación y ejecución del mismo. Tiene también como audiencia al equipo de trabajo del sistema, en especial a aquellos que cumplen el rol de “probadores”, debido a que les servirá como una especie de guía o manual para poder entender los distintos tipos de pruebas que serán necesarios para esta iteración, además de poder de llevar un control sobre los parámetros que deben cumplir cada una de las pruebas y los debidos reportes que deben generarse a la hora de conseguir un resultado.

* 1. Terminología del Documento y Acrónimos
* **Cliente**: Para efectos del Sistema, se refiere a la UAI.
* **DEX:** Decanato de Extensión. Esta unidad proveerá el servidor en donde estará almacenado el software del sistema.
* **DPF:** Departamento Planta Física*.*
* **POT:** Plan Operativo Trimestral*.*
* **PP:**Plan de Pruebas*.*
* **RUP:** *Rational Unified Process*. Se trata de una metodología para describir el proceso de desarrollo de software.
* **SIAGES:** Siglas para Sistema Automatizado de Gestión de Solicitudes.
* **Stakeholders:** Interesados o involucrados en cierto problema que necesitan una solución óptima. Desde el punto de vista del desarrollo de sistemas, son personas o entidades que están interesadas en la realización de un proyecto, auspiciando el mismo ya sea mediante su poder de decisión o de financiamiento.
* **UAI:** Unidad de Atención e Inspección. Esta unidad está adscrita a la Dirección de Planta Física de la USB.
* **USB:** Universidad Simón Bolívar.
  1. Referencias

Se tiene como referencia a los documentos de:

* Documento de Visión del sistema.
* Documento de Especificaciones de Requerimientos de Software.
  1. Estructura del Documento

El documento se estructura en 13 secciones que abarcan desde la importancia de realizar el plan de pruebas, pasando por la especificación de las pruebas tales como las pruebas que están o no previstas, el enfoque que tomaran y los criterios para definir cuando una prueba puede considerarse exitosa o no. Luego se describirá el personal encargado de realizar dichas pruebas, además del ambiente tecnológico donde se realizarán y el flujo de trabajo a seguir.

1. Propósito de la evaluación y motivación de las pruebas

Es de vital importancia realizar una serie de pruebas, ya que permite descubrir una gran parte de los errores funcionales que pueda presentar el sistema, gracias a esto se puede evaluar la calidad actual del producto y de ser necesario hacer las correcciones necesarias para poder cumplir con los requerimientos y expectativas del cliente.

* 1. Antecedentes

Se pretende plantear una solución tecnológica con el fin de optimizar la gobernabilidad, monitoreo y eficiencia, tanto a nivel técnico como funcional de estos procesos de negocio que constituyen y representan valor en las objetivos estratégicos de la organización.

El problema clave a ser solucionado se centra en la integridad de la información en el manejo de solicitudes que se realizan a la UAI, que a nivel sistémico debe funcionar de manera modular e integral.

Los documentos vinculados a ello son, Documento de Visión, Documento ERS, Documento DAS.

* 1. Propósito de la Evaluación

El propósito de las pruebas es encontrar distintos fallos en el sistema que puedan perjudicar el correcto funcionamiento del mismo y por ende genere insatisfacción del cliente con el producto. Este se define enfocado al aseguramiento de la calidad de los componentes y artefactos tecnológicos desarrollados, de manera que estos cumplan con la especificación de los requerimientos del cliente.

El proceso de evaluación y pruebas debe permitir detectar problemas desde el inicio de la especificación de requerimientos, antes de que sean de gran impacto en fases más adelantadas del proyecto, esto con el fin de disminuir los riesgos y de obtener un producto con calidad logrando mayor satisfacción del cliente.

Existen una serie de aspectos a considerar a la hora de hacer pruebas entre ellas podemos mencionar:

* Encontrar tantos errores como sea posible
* Encontrar los problemas importantes que pongan en riesgo la calidad
* Aconsejar sobre riesgos percibidos del proyecto
* Verificar la especificación (los requerimientos, los diseños o los reclamos)
* Cumplimiento de las necesidades del proceso

Cada aspecto mencionado proporciona un contexto diverso al esfuerzo de la prueba y altera la manera en la cual la prueba debe ser enfocada.

* 1. Motivadores de la Prueba

Algunos de los principales motivadores de pruebas del proyecto, son, la necesidad del cliente de optimizar y gestionar la ejecución de sus procesos de negocio, verificar la confiabilidad de la información que posee sobre sus clientes y dar trámites ágiles y efectivos a las solicitudes que ellos generan a la organización.

Adicionalmente existen unos motivadores puntuales que van a contribuir a que se construya un software que satisfaga los requerimientos del cliente de la manera más óptima posible y siguiendo un proceso adecuado para conseguirlo. Estos son:

* Aseguramiento de la calidad.
* Solicitudes de cambios.
* Riesgos de calidad.
* Verificación de los casos de uso.
* Comprobación de los requerimientos funcionales y no funcionales.

Para esta primera iteración, las pruebas irán enfocadas principalmente hacia el correcto funcionamiento de todos los módulos que componen al sistema y su correcta integración, por lo tanto las pruebas serán motivadas a enfocarse en los requerimientos funcionales, los 4 casos de uso primordiales especificados en otros documentos. También las pruebas buscarán poner en evidencia sospechas de fallas o posibles irregularidades en el diseño del sistema.

1. Aspectos a los que se enfoca la prueba

El plan de pruebas se basará en su totalidad en pruebas unitarias de integración y aceptación.

* ***Pruebas Unitarias:***Las estrategias para realizar estas pruebas consiste en generar casos de prueba necesarios: Para que cada sentencia o instrucción del programa se ejecute al menos una vez correctamente. Para que cada condición tenga por lo menos una vez un resultado verdadero y al menos una vez uno falso. Para probar varias veces el mismo bucle (en donde aplique) considerando los siguientes casos: Ignorar el bucle, pasar una vez, pasar dos veces, pasar n veces, pasar n-1 veces y n+1 veces.

* ***Pruebas de Integración:*** La estrategia para realizar estas pruebas consiste en la elaboración y ejecución de Pruebas, teniendo en cuenta flujo normal y flujos alternativos entre los módulos funcionando de manera integrada, usando datos validos e inválidos
* ***Pruebas de Aceptación:*** Debido a que las pruebas de aceptación buscan validar que un sistema cumple con el funcionamiento esperado, estas pruebas corresponden al usuario o cliente quien es el que mejor puede validar la funcionalidad y rendimiento y verificar que el alcance es correcto.

1. Delimitación de las pruebas previstas

Para este apartado se realizarán pruebas de verificación, específicamente se abarcará el nivel de pruebas unitarias y pruebas de integración para verificar la correctitud funcional e integral del sistema, evaluando los posibles casos en donde el sistema pueda fallar mediante el análisis de casos frontera, esquina y malicia para cada método involucrado en los casos de uso. Para el caso de las pruebas de validación, éstas se realizarán cada vez que se le muestren avances del producto al cliente, validando si el sistema cumple con los requerimientos especificados.

Las pruebas que serán realizadas para este 30% del sistema no cubrirán pruebas de carga, ni pruebas de stress ni pruebas de regresión ya que no está contemplado la verificación de la funcionalidad del sistema bajo condiciones anormales fuera de las que estén previstas en la visión del mismo.

* 1. Pruebas a Incluir

Se realizarán pruebas para verificar la fluidez de los datos, la independencia entre los módulos, el soporte del software y la interfaz con el usuario, se deben realizar:

* Pruebas Unitarias.
* Pruebas de Integración.
* Pruebas de Aceptación.

Estas pruebas sirven para corroborar diferentes aspectos del sistema tales como la fluidez de los datos que serán utilizados en el sistema por los diferentes usuarios. Además, permitirá verificar la independencia entre los diferentes módulos generados por los casos de uso, permitirá comprobar el estilo de la interfaz y dar un soporte al software a nivel general.

* 1. Pruebas a Excluir

Para esta iteración no se tomarán en cuenta pruebas que tengan que ver con los siguientes aspectos:

* Errores relacionados con el tiempo.
* Condiciones de error no detectadas.
* Condiciones especiales de los datos.
* Invalidez de la información mostrada por pantalla.
* Interacción con tareas en background.
* Fallos de configuración o compatibilidad con software.
* Incapacidad de soportar el volumen de carga o fallos de hardware.

Estas pruebas no serán contempladas debido a que no ayudan a alcanzar la misión de la evaluación, puesto que el sistema se ha ido desarrollando en base a las especificaciones y lineamientos establecidos en la visión de dicho sistema. Además de haber deficiencia en cuanto a los recursos para verificar distintas configuraciones de hardware y software del sistema.

1. Enfoque de las Pruebas

Para la verificación del sistema se realizarán pruebas unitarias con el fin de comprobar la correcta funcionalidad de los métodos implementados de cada módulo. Para ello se debe realizar un estudio de casos de prueba, como los son casos frontera, esquina y malicia, los cuales verifican que los métodos funcionen correctamente según los valores introducidos, ya sea que estén dentro o fuera del dominio de dichas funciones. Este tipo de pruebas enriquecen al sistema ya que maximizan la posibilidad de encontrar errores en el mismo, pudiendo ser corregidos a tiempo.

* 1. Ideas Iniciales y otras Fuentes de Referencia

SIAGES cuenta con cuatro módulos, y a su vez cada uno con diferentes funcionalidades:

* + Gestionar Solicitud
    - [Crear Solicitud](#_Toc445932772)
    - [Modificar Solicitud](#_Toc445932773)
    - [Buscar Solicitud](#_Toc445932774)
    - [Imprimir Solicitud](#_Toc445932772)
  + Gestionar Inventario
    - [Crear Material](#_Toc445932772)
    - [Modificar Material](#_Toc445932773)
    - [Buscar Material](#_Toc445932774)
  + Gestionar Empleados
    - [Crear Empleado](#_Toc445932776)
    - [Modificar Empleado](#_Toc445932777)
    - [Buscar Empleado](#_Toc445932778)
    - [Paginación de Empleados](#_Toc445932779)
  + Gestionar Catálogo
    - [Catálogo de Áreas de Trabajo](#_Toc445932781) 
      * [Agregar Área de Trabajo](#_Toc445932782)
      * [Consultar Área de Trabajo](#_Toc445932783)
      * [Modificar Área de Trabajo](#_Toc445932784)
    - [Catálogo de Edificios](#_Toc445932785) 
      * [Agregar Edificio](#_Toc445932786)
      * [Consultar Edificio](#_Toc445932787)
      * [Modificar Edificio](#_Toc445932788)
    - [Catálogo de Unidad](#_Toc445932789) 
      * [Agregar Unidad](#_Toc445932790)
      * [Consultar Unidad](#_Toc445932791)
      * [Modificar Unidad](#_Toc445932792)
    - [Catalogo de Lugares](#_Toc445932793) 
      * [Agregar Lugar](#_Toc445932794)
      * [Consultar Lugar](#_Toc445932795)
      * [Modificar Lugar](#_Toc445932796)
    - [Catálogo de Empleados](#_Toc445932797)
    - [Catálogo de Supervisores](#_Toc445932798)
    - [Catálogo de Materiales](#_Toc445932799)
    - [Catálogo de Prioridad](#_Toc445932800) 
      * [Agregar Prioridad](#_Toc445932801)
      * [Consultar Prioridad](#_Toc445932802)
      * [Modificar Prioridad](#_Toc445932803)
    - [Catálogo de Unidad Métrica](#_Toc445932804) 
      * [Agregar Unidad Métrica](#_Toc445932805)
      * [Consultar Unidad Métrica](#_Toc445932806)
      * [Modificar Unidad Métrica](#_Toc445932807)
    - [Catálogo de Estatus de Solicitud](#_Toc445932808) 
      * [Agregar Estatus de Solicitud](#_Toc445932809)
      * [Consultar Estatus de Solicitud](#_Toc445932810)
      * [Modificar Estatus de Solicitud](#_Toc445932811)
  + Gestionar Notificaciones
  1. Técnicas y Tipos de Prueba

|  |  |
| --- | --- |
| Tipo de prueba a documentar: | Prueba Unitaria. |
| Técnica: | Se realiza un estudio de casos de frontera, esquina y malicia para las funcionalidades del sistema, por medio de caja negra. |
| Objetivo de la Técnica: | Verificar el correcto funcionamiento de cada uno de los métodos de los módulos del sistema desarrollados. |
| Oráculos: | Observación de los resultados tanto en la vista como en la base de datos indicando el correcto funcionamiento de los módulos a probar. |
| Herramientas requeridas: | Este tipo de pruebas se realizarán de forma manual. |
| Criterios de Éxito: | Funcionalidades pasen todas las pruebas planteadas. |
| Consideraciones Especiales: | Las funcionalidades deben pasar las pruebas planteadas para asegurarse de tener un producto fiable. |

|  |  |
| --- | --- |
| Tipo de prueba a documentar: | Prueba de Integración. |
| Técnica: | Se seleccionan una serie de pruebas que verifiquen que haya correspondencia entre la base de datos de los módulos y las vistas de todo el sistema. |
| Objetivo de la Técnica: | Verificar el correcto funcionamiento del sistema, haciendo uso de todos sus módulos y vistas. |
| Oráculos: | Observación de las respuestas del sistema ante los estímulos inducidos por el usuario durante el uso del mismo. |
| Herramientas requeridas: | Para realizar este tipo de pruebas, se pueden realizar tanto de forma manual por medio del uso de la aplicación, así como de una herramienta automatizada como lo es Selenium. |
| Criterios de Éxito: | No haya discrepancia con respecto a la base de datos entre las vistas del sistema.  Funcionen adecuadamente las vistas y las funcionalidades de cada módulo del sistema, verificando su uso. |
| Consideraciones Especiales: | Tanto las funcionalidades como la interacción con las vistas deben ejecutarse correctamente para poder ser utilizado por el usuario final. |

|  |  |
| --- | --- |
| Tipo de prueba a documentar: | Prueba de Aceptación. |
| Técnica: | Se le muestra al cliente los avances del producto verificando si cumple con los requerimientos propuestos. |
| Objetivo de la Técnica: | Hacer una validación del sistema, comprobando que los requerimientos provistos en el documento visión se vean reflejados en el producto. |
| Oráculos: | Observación de las respuestas del sistema ante los estímulos inducidos por el usuario durante el uso del mismo. |
| Herramientas requeridas: | Para realizar este tipo de pruebas, se debe realizar una demostración ante algún Stakeholder para validar el sistema en cuestión. |
| Criterios de Éxito: | Cumplimiento de los requerimientos propuestos por los Stakeholders. |
| Consideraciones Especiales: | Tanto las funcionalidades como la interacción con las vistas deben cumplir con los requerimientos establecidos para satisfacer a los Stakeholders. |

1. Criterios de Entrada y Salida
   1. Plan de Pruebas
      1. **Criterios de Entrada**

Se puede desarrollar una suite de pruebas completo donde:

Para el caso de las pruebas unitarias, debido a que el sistema se trabaja por módulos, se pueden aplicar diversos casos de prueba para verificar el correcto funcionamiento de los métodos del mismo sin depender de los otros módulos.

Para las pruebas de integración se requiere que los módulos trabajen en conjunto bajo el mismo entorno, conformando el 30% del sistema, una vez realizada dicha integración se procederá a realizar las pruebas respectivas para comprobar el funcionamiento adecuando de la interacción del usuario con el sistema.

Por último, para las pruebas de aceptación, se han de realizar al final de cada iteración de desarrollo del sistema mediante la presentación de los avances al cliente.

A través de estos criterios el plan de pruebas se puede ejecutar.

* + 1. **Criterios de Salida**

Las suites de pruebas establecidas para verificar cada caso de uso se deben ejecutar de manera exitosa, cumpliendo los criterios de aceptación definidos por los Stakeholders.

* + 1. **Criterios de Suspensión y Retomar Actividad**

Para la ejecución satisfactoria de este plan de pruebas, se debe poseer el sistema actual ejecutándose en el servidor, en vista de que la implantación no se ha dado a cabo, no se pueden hacer por completo pruebas de integración.

* 1. Ciclos de Prueba
     1. **Criterios de Entrada**

En base a lo establecido, se puede realizar diversos tipos de pruebas una vez implantado correctamente el sistema, una vez hecho esto se pueden realizar pruebas exhaustivas en cuanto al desempeño, condicionales anormales para el sistema, operaciones de carga, entre otros.

* + 1. **Criterios de Salida**

Al momento en realizar pruebas unitarias y de integración al momento de montar el sistema en el servidor, se pueden realizar distintos tipos de pruebas pertinentes, asegurando que cumplan con las características del sistema

* + 1. **Finalización Anormal**

El plan de pruebas podría finalizar de manera anormal principalmente debido a la falta de tiempo para ejecutar las pruebas.

1. Entregables
   1. MECAP – Casos de Pruebas

Todos los casos de pruebas realizados para el 30% del sistema están adjuntos en el documento Casos\_de\_Prueba\_Equipo\_MUCZ.pdf

* 1. Resumen del ciclo de pruebas

La evaluación de las pruebas se encuentra estipulada en el documento Casos\_de\_Prueba\_Equipo\_MUCZ.pdf. En este documento se evalúan todas las pruebas realizadas sobre todos los casos de usos que fueron realizados para la entrega del 30% del sistema.

* 1. Reportes de calidad percibida

El sistema pasó casi todas las pruebas de funcionalidad e integridad realizadas, esto nos indica que el mismo posee un nivel de calidad alto en el ámbito de la usabilidad. Algunas de las pruebas realizadas mostraron resultados negativos, sin embargo los desarrolladores del equipo se pusieron de acuerdo para solventarlos de inmediato. Seguimiento de registros de cambios sugeridos

* 1. Pruebas de Regresión y soporte de los Scripts de Prueba

Junto con este documento serán anexados todos los scripts de pruebas realizados durante el desarrollo de 30% del sistema. Los scripts que se adjuntan son programas en python generados por Selenium. Estos programas están centrados en probar la integridad y la funcionalidad del sistema.

1. Flujos de Trabajo de Pruebas

El 30% del sistema consta de cuatro módulos, el módulo de solicitudes, el módulo de catálogo, el módulo de notificaciones y el módulo de recursos que se subdivide en inventario y empleados.

Las pruebas fueron realizadas básicamente por cuatro probadores. Cada uno de los probadores probó la funcionalidad y la integridad de un módulo diferente. Al final estos cuatro probadores se unieron para probar la integración de los cuatro módulos.

Cada uno se centró principalmente en probar los casos de uso que pertenecían a sus módulos.

Todas las pruebas de la tercera iteración se realizaron una semana antes de la entrega final.

1. Necesidades de ambiente
   1. Hardware

| **Recursos del Sistema** | | |
| --- | --- | --- |
| **Recurso** | **Cantidad** | **Nombre y Tipo** |
| Servidor | 1 | Servidor anfitrión |
| Computadora cliente | Sin limite |  |

* 1. Software

| **Nombre del Software** | **Versión** | **Tipo y otras notas** |
| --- | --- | --- |
| Web2py | 2.13.4 | Framework . |
| Postgres | 9.3 | Manejador de base de datos. |
| Python | 2.7 | Lenguaje de programación. |
| Debian | 8.0 | Sistema operativo. |

* 1. Herramientas de Productividad y Soporte

| **Categoría de la Herramienta o Tipo** | **Nombre de la marca Herramienta** | **Vendedor o In-house** | **Versión** |
| --- | --- | --- | --- |
| Framework de pruebas | Selenium | Software libre | 2.1. |
| Herramienta para el manejo de versiones | GitHub | Git | - |

* 1. Configuraciones del Ambiente de Pruebas

| **Nombre de Configuración** | **Descripción** | **Implementación de la Configuración Física** |
| --- | --- | --- |
| Servidor anfitrión activo. | El servidor anfitrión en la cual está alojada la aplicación debe estar encendido y disponible. | Encender y acondicionar el servidor. |

1. Responsabilidades, Personal y necesidades de entrenamiento

Esta sección presenta los recursos requeridos para tratar el esfuerzo de la prueba especificado en el Plan de Pruebas, las responsabilidades principales y los sistemas de conocimiento o habilidad requeridos deesos recursos.

* 1. Personal y Roles

Esta tabla muestra supuestos que proveen de personal para el esfuerzo de la prueba.

| **Recursos Humanos** | | |
| --- | --- | --- |
| **Roles** | **Recursos Mínimos Recomendados**  **(número de roles a tiempo completo asignados)** | **Responsabilidades Específicas o Comentarios** |
| Administrador de Pruebas | 2 | * Administrar el esfuerzo de las pruebas. * Aprobar los criterios de entrada y salida a las pruebas. * Monitorear avance del esfuerzo de pruebas. * Aprueba los casos de prueba, gestiona el alcance y misión de las pruebas. * Certifica el nivel de calidad del producto construido. |
| Diseñadores de pruebas | 2 | * Diseñar los set de pruebas (estructura y enfoque) que se realizarán al sistema para una certificar que se construyó un producto que satisface los requerimientos definidos. * Definir las herramientas automatizadas que se utilizarán. |
|  |  |  |
| Probadores | 2 | * Ejecutar los casos de pruebas y registrar los resultados |
| Ana**l**istas de pruebas | 2 | * Realizar documentación técnica de las pruebas. * Análisis de resultados de casos de prueba, y realización de modificaciones en caso de que en las pruebas se encuentren problemas, fallas, riesgos o vulnerabilidades. |

* 1. Personal y Necesidades de entrenamiento

La manera de trabajar del equipo de desarrollo de pruebas y su entrenamiento variará de proyecto en proyecto. Dicho equipo estará integrado por Administradores de Prueba, Diseñadores de prueba, Probadores y analistas de pruebas.

Ellos se encargarán de administrar y monitorear el esfuerzo de las pruebas, aprobar los casos de prueba, gestionar el alcance y misión de las pruebas, certificar el nivel de calidad del producto construido, diseñar los set de pruebas, definir las herramientas automatizadas que se utilizarán, ejecutar los casos de pruebas y registrar los resultados, realizar documentación técnica de las pruebas, etc., del sistema y subsistema, teniendo cuidado de no hacer pruebas sobre un mismo subsistema. Este problema se evitará asignando las pruebas en una reunión del equipo y revisando constantemente el control de versiones de las clases involucradas de manera tal que siempre se trabaje con la versión actualizada.

Para poder hacer las pruebas, dado que los encargados de dicha actividad pertenecen al mismo equipo de desarrollo, no es necesaria ninguna destreza que ya no posea un programador en el lenguaje. Sin embargo, el equipo de prueba requiere a menudo la ayuda y habilidades de otros miembros del equipo de desarrollo que no son parte de las pruebas, entre ellos los administradores de la base de datos y los desarrolladores.

1. Puntos de control en la iteración

| **Puntos de Control** | **Fecha de Inicio Planificada** | **Fecha de Inicio Actual** | **Fecha Final Planificada** | **Fecha Final Actual** |
| --- | --- | --- | --- | --- |
| Acuerdos de Plan de Iteración | 04-03-2016 | 07-03-2016 | 11-03-2016 | 11-03-2016 |
| Inicio de la Iteración | 11-03-2016 | 11-03-2016 | 18-03-2016 | 18-03-2016 |
| Primera construcción a ser entregada para pruebas | 11-03-2016 | 11-03-2016 | 13-03-2016 | 13-03-2016 |
| Primera construcción aceptada por pruebas | 13-03-2016 | 13-03-2016 | 15-03-2016 | 14-03-2016 |
| Segunda construcción entregada a pruebas | 14-03-2016 | 14-03-2016 | 16-03-2016 | 16-03-2016 |
| Segunda construcción que finaliza ciclo de pruebas | 16-03-2016 | 16-03-2016 | 17-03-2016 | 18-03-2016 |
| Revisión de la Iteración | 17-03-2016 | 17-03-2016 | 17-03-2016 | 18-03-2016 |
| Fin de la Iteración | 18-03-2016 | 18-03-2016 | 18-03-2016 | 18-03-2016 |

1. Riesgos, Dependencias, Suposiciones y Restricciones

| **Riesgos** | **Estrategias de Mitigación** | **Contingencias**  **(Riesgo es realizado)** |
| --- | --- | --- |
| Tiempo insuficiente. | Identificar los casos críticos en el esquema de los casos de prueba, a fin de probarlos. Especificar aquellos casos que no pudieron ser probados. | Asignar un mayor número de recursos |
| Atraso en cuanto a corrección de errores. | Dar prioridad a errores funcionales y obstáculos que impidan la el desarrollo y la continuación de las pruebas | Asignar un mayor número de personal de pruebas |
| Plan de Pruebas deficiente. | En el caso en que algún probador ejecutase mal el plan de pruebas, se debe contar con un segundo probador que valide la correcta aplicación de las estrategias de pruebas. | En la medida de lo posible, tratar de negociar una adición de tiempo a la fase de pruebas inicialmente planteada |
| Base de datos incorrecta. | Los desarrolladores deben asegurarse que la base de datos utilizada esté completa y actualizada. Además, debe informarle al equipo de pruebas cómo instalarla correctamente para que no existan problemas con la conexión de la misma. | Asignar un experto en base de datos, para que pueda comprobar constantemente el estatus de la base de datos. |

| **Dependencia entre** | **Impacto Potencial o Dependencia** | **Dueños** |
| --- | --- | --- |
| Sección de catálogos y los módulos de inventario, empleados y solicitudes. | Los catálogos son usados de manera constantes por los otros módulos, un fallo de esto puede causar un fallo en los otros módulos. | Módulo de Catálogos |
| Notificaciones y solicitudes. | Una vez generada la solicitud, el módulo de notificaciones debe informar al (la) jefe(a) de la UAI que se ha realizado una nueva solicitud y así ésta realizar las acciones pertinentes según sea el caso. | Módulo de Notificaciones |
| Solicitudes e Inventario. | Al momento de asignar los materiales que son necesarios para solventar cierta solicitud. El inventario tiene que arrojar los datos correctos, si no esto puede causar un retraso en el cumplimiento de esta solicitud o de otras futuras. | Módulo de Inventario |

| **Supuestos a ser probados** | **Impacto de los supuestos incorrectos** | **Dueños** |
| --- | --- | --- |
| Modificar elementos que se encuentren ya creados en el catálogo. Y no se guarde en la base de datos. | Los otros módulos pueden realizar una búsqueda usando como palabra clave este elemento modificado, y no se dé el resultado esperado. | Módulo de Catálogos |
| Enviar un correo al usuario una vez una nueva solicitud sea atendida por la UAI con el número de esta. | El correo enviado al usuario presenta una información incorrecta. Esto pude causar que el seguimiento a la solicitud no se dificulte. | Módulo de Notificaciones |
| La creación del archivo PDF que se generara para su posterior impresión, con toda la información que tiene una nueva solicitud. | El archivo PDF presente campos mal colocados dentro del archivo o en su defecto la información que debe pasar en un campo es incorrecta. | Módulo de Solicitudes |

| **Limitaciones** | **Impacto que tienen las limitaciones y el esfuerzo de las Pruebas** | **Dueños** |
| --- | --- | --- |
| El tiempo estipulado para poner en marcha el plan de pruebas. | Por el tiempo requerido y la fecha de entrega del Sistema, las pruebas no pueden cumplirse en su totalidad. | Todos los grupos de desarrollo. |
| Correcta integración del módulo de solicitudes con el resto de los módulos, puesto que este depende del resto. | Las pruebas pueden dar resultados erróneos en las pruebas realizadas, puesto que la integración no se realizó correctamente, | Módulo de Solicitudes |
| Las fallas de conexión en internet. | La falla de conexión a internet representa un retraso constante en las pruebas puesto que se pueden quedar a mitad o al ser mínimos puede generar resultados erróneos para su análisis | Externo al sistema |

1. Procedimientos y Gerencia de Procesos

El software posee un gran impacto sobre la calidad de un sistema, por lo que es muy importante estar en capacidad de tomar decisiones acertada sobre esas situaciones que se presentan. Es por eso el uso de este plan de pruebas. Para darle calidad al sistema.

* 1. Midiendo y determinando el grado de la prueba

Dependiendo del paradigma de AyD, y programación que se esté utilizando, se pueden proponer diferentes conjuntos de métricas. Entre ellas destacan las de estructura jerárquica, para las clases, para las cartas de navegación, entre otras.

* 1. Determinación de los entregables de éste plan de pruebas

Este plan de prueba será entregado al final del avance del 30% del proyecto de levantamiento del Sistema Automatizado de Gestión de Solicitudes de Planta Física (SIAGES), junto con otros documentos, como el documento visión o plan creativo, por ejemplo, que conforman la información de todo lo que se hizo para levantar, hasta esta fase, el proyecto, y una vez así los subsiguientes desarrolladores puedan llegar a un 100%.

* 1. Reportes del problema, escalada y resolución

Al trabajar con el Framework web2py, los reportes de errores se hacían fáciles de entender y por lo tanto fáciles de corregir al momento. Además que los desarrolladores trabajaron de manera modular, lo cual hace que el programa sea mantenible.

* 1. Gerenciar Ciclos de Prueba

Los desarrolladores junto con los probadores establecieron ciclos de pruebas con sus respectivos comentarios que permitieron una mejora importante al sistema de cara a realizar la presentación al cliente. Para tener una información más detalla, se invita al lector a volver a la sección número 12, donde se encuentran especificadas las fechas claves para el desarrollo del sistema.

* 1. Estrategias de trazabilidad

Las pruebas fueron diseñadas para que cumplieran lo esperado por cada funcionalidad desarrollada dentro del sistema.

Las motivaciones para probar, estas permiten resaltar la importancia de pruebas para ayudar a determinar si se mantienen o se retiraran las pruebas. Los elementos del diseño del software, activan el seguimiento de los cambios subsecuentes del diseño que harían necesario la reejecución de pruebas o retirarlas

Requerimientos resultantes de los Cambios, activa las pruebas que descubrieron la necesidad para el cambio a ser identificado y reejecuta para verificar que la petición del cambio se ha terminado con éxito.

* 1. Aprobación y Finalización

Una vez leído el presente informe sobre el plan de pruebas, y presentado su conformidad con lo establecido en este plan de pruebas. La Licenciada Yesiree Barrios, jefa de la Unidad de Atención e Inspección de Planta Física, como representante de este ente de la Universidad Simón Bolívar, acepta que el 30% del sistema ha sido probado satisfactoriamente.