Plan de Testing

Versión 1.1

Proyecto: Entropy

|  |  |  |
| --- | --- | --- |
| **Universidad Tecnológica Nacional**  **Facultad Regional Córdoba**  **Ingeniería en Sistemas de Información**  **Cátedra:** Proyecto Final  Plan de Testing V1.1  **Curso:** 5K2  *GRUPO 5 - EntropyTeam*  **Año 2014** | INTEGRANTES | |
| Alumno | Legajo |
| CUNIBERTTI, Lucas | 57476 |
| DONNET, Matías | 55017 |
| GIUSTO, Denise | 55192 |
| NOVES, Gastón | 51496 |
| RUIZ LOPEZ, José | 55230 |

Información del Documento

|  |  |
| --- | --- |
| **Título del Documento** | Plan de Testing |
| **Nombre del Archivo del Documento** | Plan de Testing\_v1.1.docx |
| **Número de Versión** | 1.1 |
| **Autor** | GIUSTO Denise |
| **Contribuyentes** | CUNIBERTTI Lucas  DONNET Matías  NOVES Gastón  RUIZ LOPEZ José |
| **Fecha de Creación** | 24/05/2014 |

Historia de Cambios

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 24/05/2014 | 0.1 | Borrador | GIUSTO Denise |
| 24/05/2014 | 1.0 | Versión Inicial | CUNIBERTTI Lucas  DONNET Matías  GIUSTO Denise  NOVES Gastón  RUIZ LOPEZ José |
| 13/06/2014 | 1.1 | Correcciones: Estrategia de Prueba. | GIUSTO Denise |

Referencias

| **ID Archivo de Documento** | **Título del Documento** | **Fecha de Publicación** |
| --- | --- | --- |
| Plantilla\_Reporte de Inspeccion\_v1.0.xlsx | Plantilla de Reporte de Inspección | 24/05/2014 |
| Plantilla\_Casos de Prueba\_v1.0.xlsx | Plantilla de Casos de Prueba | 24/05/2014 |

Tabla de Contenidos

|  |
| --- |
| [Introducción 6](#_Toc390442556)  [Propósito del Plan 6](#_Toc390442557)  [Objetivos 6](#_Toc390442558)  [Alcance 6](#_Toc390442559)  [Criterios de inicio y finalización 7](#_Toc390442560)  [Prueba del Sistema 8](#_Toc390442561)  [Estrategia de Prueba 8](#_Toc390442562)  [Criterios de aceptación 8](#_Toc390442563)  [Categorías de resultados de prueba 9](#_Toc390442564)  [Identificación de la prueba 10](#_Toc390442565)  [Errores de prueba 11](#_Toc390442566)  [Documentación de la prueba 12](#_Toc390442567)  [Ambiente de Prueba 13](#_Toc390442568)  [Hardware 13](#_Toc390442569)  [Software 13](#_Toc390442570)  [Otros 13](#_Toc390442571)  [Responsabilidades 14](#_Toc390442572) |
|  |

# Introducción

## Propósito del Plan

El propósito de este plan es planificar, estructurar y documentar la planificación de las pruebas del sistema Entropy, así como la estrategia a utilizar para su ejecución.

## Objetivos

El Plan de Pruebas del Sistema especifica los procesos de test y de verificación que se realizaran con el objeto de:

* Identificar defectos y fallas.
* Medir rendimiento.
* Evaluar la calidad
* Determinar el cumplimiento de los requerimientos.

Los objetivos de este plan son:

* Definir y detallar todas las tareas que se desarrollarán para probar el sistema.
* Definir el plan y la persona o grupo responsable de cada tarea.
* Definir las herramientas de prueba y el ambiente necesario a la conducción de las actividades de test.
* Definir los ítems y funcionalidades que serán probados.

## Alcance

El Plan de Pruebas del Sistema es una especificación de alto nivel de los requerimientos funcionales y de calidad que serán probados, del ambiente de testing, de la estrategia de testing, de las responsabilidades y de los criterios de éxito.

El comportamiento de un producto bajo testing será comparado con las especificaciones de los requerimientos que fueron usados para implementar el sistema, incluyendo todos los cambios que han sido aprobados e implementados.

Los casos de prueba y los criterios de éxito serán derivados de este plan general y serán especificados en el documento de Especificaciones de Prueba del Sistema.

El alcance del test del sistema es probar la funcionalidad completa y el rendimiento del sistema Entropy.

## Criterios de inicio y finalización

### Criterios de Inicio

Se manejaran los siguientes criterios de inicio de las pruebas de aceptación:

* Test unitarios realizados y completados para cada componente del sistema.
* Disponer de una línea base del sistema.

### Criterios de Finalización

La prueba finalizará cuando el 95% de los Casos de Prueba hayan arrojado un resultado de éxito.

# Prueba del Sistema

## Estrategia de Prueba

### Estrategia General

Las pruebas se realizarán con una filosofía bottom-up, comenzando por las pruebas unitarias las cuales estarán a cargo del desarrollador, basadas en JUnits y deberán encontrarse pasadas al momento de ejecutar los Casos de Prueba. Luego seguirán las pruebas de integración y sistema.

### Estrategia de Inspección de Código

Siempre que alguno de los miembros del team lo crea necesario, puede convocar una inspección de software para rever funcionalidad que considere complicada. Quedará bajo su responsabilidad la organización de la reunión. La inspección demandará no más de dos horas de preparación previa y no podrá durar más de cuatro horas. Las horas necesarias se acordarán fuera del calendario estipulado a fin de no afectar el desarrollo del Sprint. Se emitirá un reporte según el formulario *Plantilla\_Reporte de Inspección\_v1.0.xlsx*.

### Estrategia de Regresión

Las pruebas de regresión se realizarán sobre funcionalidades críticas afectadas por bugs que han sido corregidos. Se ejecutarán luego de que hayan finalizado las pruebas correspondientes al Sprint.

### Estrategia de Integración

Las pruebas de integración se realizarán luego de que hayan sido ejecutadas las pruebas correspondientes a los Sprints de Release.

## Criterios de aceptación

Para que un ítem supere la prueba es necesario que los errores de severidad 1, 2 y 3 que hayan sido encontrados sean removidos.

Identificamos los siguientes criterios los cuales deben ser evaluados progresivamente.

* Requerimientos de Prueba: ¿Todos los requerimientos del sistema han sido probados?
* Pruebas Cubiertas: ¿Todas las partes del software han sido probadas, incluyendo manejo de errores?
* Medida de Casos de Prueba: ¿Cuántos Casos de Prueba han sido planeados, diseñados, implementados, ejecutados y pasaron con éxito o falla?
* Defectos detectados en Casos de Prueba: Es importante tener un ratio de los defectos encontrados en los casos de prueba, y de los defectos corregidos y mantenidos.

## Categorías de resultados de prueba

Esta sección describe las categorías que pueden ser asignadas los resultados de prueba en un Caso de Prueba.

| Nivel de Severidad | Denominación | Descripción | Ejemplo |
| --- | --- | --- | --- |
| 1 | Grave | Falla en el sistema. No es posible continuar con el procesamiento. | Un error crítico ha sido encontrado y no permite que se continúe con la operación de la aplicación. |
| 2 | Mayor | No es posible continuar con el proceso de la función seleccionada | El componente clave no está disponible o la funcionalidad está incorrecta. |
| 3 | Menor | Funciones restringidas, pero el procesamiento puede continuar | Componentes no críticos no están disponibles o la funcionalidad es incorrecta; cálculos incorrectos en funcionalidad clave. |
| 4 | Cosmético | Cambio de forma menor | Errores de usabilidad, pantallas o reportes de errores que no afectan la calidad, el uso ni la funcionalidad del sistema, por ejemplo, cambio en una etiqueta, en un mensaje, etc. |

## Identificación de la prueba

### Casos de Prueba

Cada UserStory deberá tener un Caso de Prueba ligado, cuyo formato se especifica en el documento *Plantilla\_Casos de Prueba\_v1.0.xlsx*. Dicha plantilla deberá instanciarse para cada Caso de Prueba, nombrándose acorde a las reglas prefijadas.

Se describirán los pasos y los resultados esperados de cada prueba individual. En particular:

* Identificador del proyecto.
* Descripción del objetivo de la prueba.
* Historial de cambios.
* Referencias.
* Identificador del Caso de Prueba.
* Funcionalidad a testear.
* Evento que desencadena la funcionalidad.
* Descripción de la funcionalidad.
* Descripción del estado de la aplicación antes de la prueba o pre-condiciones de la misma.
* Pasos precisos y no ambiguos para ejecutar la prueba.
* Descripción de los resultados esperados.
* Datos requeridos como entrada para la prueba.
* Responsable.
* Prioridad.
* Categoría (Regresión o no).

### Reporte de resultados

Se denominará ciclo de prueba a cada una de las ejecuciones de un Caso de Prueba determinado. Los resultados de cada ciclo de prueba son registrados en el mismo formulario en que estipula el Caso de Prueba, e incluye la siguiente información:

* Resultado para cada paso del Caso de Prueba.
* Número de defecto.
* Responsable de la ejecución.
* Observaciones.
* Nombre y versión de la aplicación a prueba.
* Fecha de Prueba.
* Categoría de resultado de prueba.
* Descripción del error.

## Errores de prueba

Esta sección especifica los procesos para alcanzar la corrección de los errores observados y registrados durante la prueba.

Para cada error observado que requiera corrección de la aplicación o de la especificación de funcionalidades, los miembros del team deben de estar de acuerdo en lo siguiente:

* El ámbito de trabajo adicional y escalas de tiempo para la corrección.
* El Caso de Prueba requerido para ser re-ejecutado después de la corrección.
* Dada una falla, el principal responsable de realizar la corrección es el que se encargó de desarrollar dicho componente.
* Establecer prioridades de acuerdo a una serie de fallas.

En la plantilla del Caso de Prueba deberá ser actualizada también, cuando sea pertinente, la siguiente información:

* Estado de corrección.
* Responsable de corrección.

Los errores encontrados deberán ser cargados como bugs por el responsable de la prueba en la herramienta de gestión de proyectos ágiles, Rally, dentro del Sprint Backlog, para ser corregidos o incluidos en las Notas de Release.

## Documentación de la prueba

Esta sección describe los documentos que deben ser generados durante la actividad de prueba. Estos documentos son los siguientes:

* Casos y resultados de Pruebas siguiendo el formato especificado.
* Reporte consolidado de pruebas por módulo.

Los documentos serán guardados en el repositorio en la carpeta [directorio\_raíz]/producto/tests/sprint-[número\_de\_sprint\_correspondiente].

# Ambiente de Prueba

Esta sección establece los componentes del ambiente de testing.

## Hardware

El número de computadoras con que se contará será de acuerdo al tipo de prueba, entre una y diez computadoras.

Los equipos de prueba deberán poseer las siguientes características:

* 2GB de RAM o superior
* Procesador de al menos doble núcleo
* 10 GB de disco libres
* Windows 7, Windows 8, Linux Ubuntu 12 o superior, o similar

## Software

Las aplicaciones y servicios necesarios para completar correctamente las pruebas incluyen:

* SQLite - JDBC4 - 3.8.2.jar
* SQLiteMan - 1.2.2 - Windows x86.zip
* SQLite - Shell Windows x86 - 3.8.4.3.zip
* NetBeans 8.0 - Windows.exe
* Java SE - JDK 7u45 - Windows x64.exe
* Tortoise SVN 1.8.6.25419 - Windows x64.msi
* Rally
* Google Code

## Otros

### Datos de Prueba

Se desarrollarán y especificarán conjuntos de datos de prueba, tomando las muestras necesarias para la ejecución de las pruebas, de manera que se verifique que cumple con diversos tipos de datos.

# Responsabilidades

|  |  |  |
| --- | --- | --- |
| Recursos humanos | | |
| Cargo | Recursos mínimos necesarios | Responsabilidades específicas / comentarios |
| Administrador de pruebas | DONNET Matías | Proporcionar atención especial al funcionamiento correcto de las tareas principales del sistema.  Responsabilidades:   * Proporcionar dirección técnica. * Adquirir los recursos apropiados. * Administración de reportes. |
| Diseñador de pruebas | CUNIBERTTI Lucas  DONNET Matías  GIUSTO Denise  NOVES Gastón  RUIZ LOPEZ José | Identificar, asignar la prioridad, e implementar los casos de la prueba a ejecutar.  Responsabilidades:   * Generar el plan de prueba. * Generar la especificación de los distintos tipos de prueba. * Generar el modelo de prueba. * Evaluar la eficacia del esfuerzo en la prueba. |
| Ejecutores de prueba | CUNIBERTTI Lucas  DONNET Matías  GIUSTO Denise  NOVES Gastón  RUIZ LOPEZ José | Realizar las pruebas.  Responsabilidades:   * Ejecutar pruebas. * Registrar resultados. * Recuperación después de errores. * Documentación de errores. |
| Administrador del sistema de pruebas | CUNIBERTTI Lucas | Asegurar el ambiente de prueba, así como mantener y manejar sus ventajas.  Responsabilidades:   * Administrar el manejo de pruebas del sistema. * Controlar el acceso de los integrantes del equipo a los sistemas de prueba. |
| Administrador de la Definición de Datos | DONNET Matías | Asegurar el ambiente de los datos de prueba, así como manejar y mantener sus ventajas.  Responsabilidades:   * Administrar los datos de prueba. |