**Bug Manager**

Sistema de Administración de Calidad de Proyectos de Software

Pruebas de Sistema

Rev 0.1

Fecha: 6 de Febrero de 2011

Autor: MR, HG, EC

Distribución:

|  |  |  |  |
| --- | --- | --- | --- |
| **Autoridad** | **Nombre** | **Firma** | **Fecha** |
| Director de Tesis | Óscar Mondragón (OM) |  |  |
| Tesista | Marco Rangel (MR) |  |  |
| Tesista | Eduardo Campos (EC) |  |  |
| TEsista | Humberto García (HG) |  |  |

# Contenidos

|  |  |
| --- | --- |
|  | [Contenidos 2](#_Toc316304563)  [1. Control del Documento 3](#_Toc316304564)  [1.1. Propósito y alcance 3](#_Toc316304565)  [1.2. Audiencia 3](#_Toc316304566)  [1.3. Visión General del Documento 3](#_Toc316304567)  [1.4. Organización del Documento 3](#_Toc316304568)  [1.5. Definiciones 3](#_Toc316304569)  [1.6. Control de Versiones 4](#_Toc316304570)  [1.7. Visión General del Sistema 4](#_Toc316304571)  [2. Plan de Acción 5](#_Toc316304572)  [2.1. Objetivo de las Pruebas 5](#_Toc316304573)  [2.2. Criterios de Entrada y de Salida 5](#_Toc316304574)  [2.2.1. Ambiente de Prueba 5](#_Toc316304575)  [2.3. Proceso de Pruebas 6](#_Toc316304576)  [2.4. Ejemplo de Caso de Prueba 6](#_Toc316304577)  [3. Casos de Prueba 7](#_Toc316304578) |

# Control del Documento

## Propósito y alcance

El presente documento refiere a las pruebas del sistema de administración de calidad de proyectos de software, cuyo nombre clave es “Bug Manager” (BM).

## Audiencia

Este documento está dirigido a:

* Oscar Mondragón (Director de tesis) para su revisión y aceptación.
* Equipo de desarrollo.

## Visión General del Documento

El documento incluye una serie de casos de prueba donde se asegurará la funcionalidad del sistema.

A su vez se tendrán los resultados de la ejecución de estos casos de prueba.

## Organización del Documento

|  |  |
| --- | --- |
| Sección 1 | Contiene información introductoria que incluye el propósito y los alcances del documento, la audiencia a la que está dirigido, definiciones de los términos utilizados en el documento, un control de versiones y una pequeña descripción de cada sección. |
| Sección 2 | Contiene el plan de acción de las pruebas del sistema. |
| Sección 3 | Contiene los casos de prueba con sus resultados. |

## Definiciones

|  |  |
| --- | --- |
| **Término** | **Significado** |
| BM | Bug Manager |

Tabla1: Definiciones

## Control de Versiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versión** | **Fecha** | **Editor** | **Comentarios** | **Aprobación** |
| 0.1 | Junio 24 | MR | Versión inicial de las Pruebas de Sistema |  |

Tabla2: Control de Versiones

## Visión General del Sistema

El documento establece las metas, características, ambiente operacional y funcionalidad del BM. El BM será utilizado por cualquier empresa de desarrollo de software que busque implementar técnicas de control de calidad en su proceso de desarrollo de software. Este producto va enfocado a pequeñas y medianas empresas, principalmente mexicanas.

El BM permitirá que los individuos, equipos de desarrollo, o empresas que lo utilicen puedan tener control sobre el avance y la calidad de sus proyectos por medio de lo siguiente:

* Guía en la elaboración del plan de calidad.
* Definición del ciclo de vida y actividades de desarrollo.
* Registro y seguimiento de actividades de aseguramiento de calidad.
* Registro y seguimiento de defectos.
* Generación de estadísticas personales, por proyecto, por equipo y de empresa.

# Plan de Acción

Esta sección describe el enfoque general del plan de pruebas e incluye lo siguiente:

* Contexto de Prueba.
* Requerimientos de Prueba.
* Estrategia de Prueba.
* Reporte de Errores.

## Objetivo de las Pruebas

El objetivo es la validación de las distintas funciones y tareas que el sistema debe de desempeñar, para poder comenzar a utilizarlo.

## Criterios de Entrada y de Salida

Se comenzará a probar la versión funcional del sistema. Se realizarán un grupo de casos de prueba, para que el sistema pueda acreditar positivamente las pruebas debe de haber conseguido un resultado positivo en cada uno de los casos.

Si se encuentra algún defecto en algún caso de prueba se deberá agregar el error al documento de reporte de errores, realizar la corrección del software y volver a realizar la prueba. Se realizará este proceso con cada uno de los casos de prueba hasta que todas sean positivas.

Una vez que todos los casos de prueba acrediten positivamente se considerará que el sistema está terminado.

### Ambiente de Prueba

Para realizar las pruebas debe de existir al menos una computadora personal, en la cual estará montado el servidor, y desde la cual se accederá al sistema desde un navegador de internet como un usuario normal.

Las especificaciones de software para que una computadora personal pueda fungir como servidor son las siguientes:

* Java JDK 7 instalado.
* Tomcat 6.0.29 instalado.
* MySQL Server 5.1.49 instalado.

Las especificaciones de software para que una computadora personal pueda fungir como cliente son las siguientes:

* Java JRE 7 instalado.
* Firefox 10.0 instalado.

## Proceso de Pruebas

El primer paso de la prueba será elaborar los casos de prueba. Una vez que se tengan los casos de prueba se seguirá el siguiente proceso:

1. Realizar las pruebas detalladas en el caso.
2. En caso de aparezcan defectos, se hará el reporte de errores para su futura corrección.
3. Pasar al siguiente caso de prueba.

## Ejemplo de Caso de Prueba

El siguiente es un ejemplo de caso de prueba y será como se elaborarán todos los del sistema:

|  |  |
| --- | --- |
| **Propósito:** | Asegurar la funcionalidad del login. |
| **Identificación gráfica:** | Botón con texto entrar. |
| **Prerrequisitos:** |  |
| **Casos de Prueba:** | * Ingresar a la página principal del Compocohua. * Escribir los siguientes datos en la pantalla de login:  1. ID Usuario: *en blanco* Contraseña: *en blanco* 2. ID Usuario: *un usuario válido* Contraseña: *en blanco* 3. ID Usuario: *un usuario inválido* Contraseña: *en blanco* 4. ID Usuario: *en blanco* Contraseña: *una contraseña inválida* 5. ID Usuario: *un usuario válido* Contraseña: *una contraseña inválida* 6. ID Usuario: *un usuario inválido* Contraseña: *una contraseña inválida* 7. ID Usuario: *en blanco* Contraseña: *una contraseña válida* 8. ID Usuario: *un usuario válido* Contraseña: *una contraseña válida* 9. ID Usuario: *un usuario inválido* Contraseña: *una contraseña válida* |
| **Resultados esperados:** | 1.- *Usuario o password incorrectos.* 2.- *Usuario o password incorrectos.* 3.- *Usuario o password incorrectos.* 4.- *Usuario o password incorrectos.* 5.- *Usuario o password incorrectos.* 6.- *Usuario o password incorrectos.* 7.- *Usuario o password incorrectos.* 8.- Se ingresa al sistema y te manda a la página de selección de proyectos. 9.- *Usuario o password incorrectos.* |
| **Resultados obtenidos** |  |
| **Notas y Preguntas:** |  |

# Casos de Prueba

|  |  |
| --- | --- |
| **Propósito:** | Probar funcionalidad de los proyectos |
| **Identificación gráfica:** | Botón de Proyectos |
| **Prerrequisitos:** | 1. Ninguno. 2. Proyecto creado. 3. Proyecto creado vacío y proyecto creado no vacío. |
| **Casos de Prueba:** | 1. Agregar proyecto:    1. Dejar en blanco cada uno de los campos obligatorios a la vez. 2. Modificar proyecto:    1. Intentar dejar en blanco un campo obligatorio 3. Borrar proyecto:    1. Intentar borrar un proyecto vacío (sin tareas, ni plan de vida, ni defectos)    2. Intentar borrar un proyecto no vacío (con tareas y/o defectos) |
| **Resultados esperados:** | 1. Agregar proyecto    1. Evitar creación de proyecto 2. Modificar proyecto:    1. Evitar modificación de los datos 3. Borrar proyecto:    1. Borrado exitoso.    2. Evitar borrado de proyecto. |
| **Resultados obtenidos** | 1. Agregar proyecto:    1. El esperado 2. Modificar proyecto:    1. El esperado 3. Borrar proyecto:    1. Excepción al momento de borrarlo: Could not execute JDBC batch update.    2. El esperado |
| **Notas y Preguntas:** | Ninguno. |

|  |  |
| --- | --- |
| **Propósito:** | Probar funcionalidad de los tipos de defectos |
| **Identificación gráfica:** |  |
| **Prerrequisitos:** | 1. Ninguno 2. Tipo de defecto creado 3. 1. Tipo de defecto creado y sin defectos    2. Tipo de defecto creado y con defectos |
| **Casos de Prueba:** | 1. Agregar tipo de defecto    1. Dejar cada uno de los campos obligatorios vacíos 2. Modificar tipo de defecto    1. Dejar cada uno de los campos obligatorios vacíos 3. Borrar tipo de defecto    1. Borrar tipo de defecto sin defectos    2. Borrar tipo de defecto con defectos |
| **Resultados esperados:** | 1. Agregar tipo de defecto    1. Evitar creación 2. Modificar tipo de defecto:    1. Evitar modificación 3. Borrar tipo de defecto:    1. Borrado exitoso.    2. Evitar borrado. |
| **Resultados obtenidos** | 1. Agregar tipo de defecto:    1. El esperado 2. Modificar tipo de defecto:    1. El esperado 3. Borrar tipo de defecto:    1. El esperado    2. Excepción al momento de borrarlo: Could not execute JDBC batch update. |
| **Notas y Preguntas:** | Ninguno. |

|  |  |
| --- | --- |
| **Propósito:** | Probar la funcionalidad del ciclo de vida |
| **Identificación gráfica:** |  |
| **Prerrequisitos:** | 1. Proyecto creado. 2. Proyecto y ciclo de vida creados.    1. Proyecto y ciclo de vida creados.    2. Proyecto y ciclo de vida creados, con tareas y/o defectos. |
| **Casos de Prueba:** | 1. Agregar fase al ciclo de vida    1. Dejar vació cada uno de los campos obligatorios 2. Modificar fase al ciclo de vida    1. Dejar vacío cada uno de los campos obligatorios    2. Modificar el orden de las fases 3. Borrar fase al ciclo de vida    1. Borrar fase sin tareas ni defectos    2. Borrar fase con tareas y/o defectos |
| **Resultados esperados:** | 1. Agregar fase al ciclo de vida    1. Evitar creación 2. Modificar fase al ciclo de vida    1. Evitar modificación    2. Actualización del ciclo de vida correctamente 3. Borrar fase del ciclo de vida    1. Borrado exitoso    2. Evitar borrado |
| **Resultados obtenidos** | 1. Agregar fase al ciclo de vida    1. El controller regresa una excepción 2. Modificar fase al ciclo de vida    1. El controller regresa una excepción    2. El esperado 3. Borrar fase del ciclo de vida    1. El esperado    2. Excepción al momento de borrarla: Found exception mx.itesm.gda.bm.biz.BizException: Cannot delete non-empty phase! |
| **Notas y Preguntas:** | Ninguno. |

|  |  |
| --- | --- |
| **Propósito:** | Probar la funcionalidad de las actividades de un proyecto |
| **Identificación gráfica:** |  |
| **Prerrequisitos:** | 1. Proyecto creado y recursos asignados. |
| **Casos de Prueba:** | 1. Agregar actividad a un proyecto:    1. Dejar vacío cada uno de los campos obligatorios 2. Modificar actividad de un proyecto:    1. Dejar vació cada uno de los campos obligatorios    2. Cambiar estatus de tarea a completada con esfuerzo restante mayor a 0.    3. Cambiar datos de tareas COMPLETADAS    4. Cancelar tareas no iniciadas 3. Borrar actividad de un proyecto:    1. Borrar una actividad no iniciada    2. Intentar borrar una actividad iniciada |
| **Resultados esperados:** | 1. Agregar actividad a un proyecto:    1. Evitar creación 2. Modificar actividad de un proyecto:    1. Evitar modificación    2. Evitar modificación mostrando mensaje    3. Solo permitir cambios en los campos especificados    4. Permitir cancelación 3. Borrar actividad de un proyecto:    1. Borrado exitoso    2. Evitar borrado |
| **Resultados obtenidos** | 1. Agregar actividad a un proyecto:    1. Cuando no se llena la descripción el controller manda una excepción 2. Modificar actividad de un proyecto:    1. Cuando no se llena la descripción el controller manda una excepción    2. El controller envía la siguiente excepción: No se puede completar una tarea si las horas restantes son mayores a 0!    3. SIEMPRE se envía la siguiente excepción: MissingServletRequestParameterException: Required String parameter 'status' is not present!    4. El esperado 3. Borrar actividad de un proyecto:    1. El esperado    2. El esperado |
| **Notas y Preguntas:** | Ninguno. |

|  |  |
| --- | --- |
| **Propósito:** | Probar la funcionalidad de los defectos |
| **Identificación gráfica:** |  |
| **Prerrequisitos:** | 1. Proyecto creado, recursos asignados y tareas creadas. |
| **Casos de Prueba:** | 1. Agregar un defecto:    1. Dejar vacío cada uno de los campos obligatorios 2. Modificar defecto:    1. Dejar vacío cada uno de los campos obligatorios    2. Poner una fase de remoción anterior a la fase de inyección    3. Poner una fase de remoción anterior a la fase de detección    4. Poner una fase de detección anterior a la fase de inyección 3. Borrar defecto:    1. Borrar defecto en estatus ENVIADO    2. Borrar defecto en estatus ACEPTADO    3. Borrar defecto en estatus CORREGIDO |
| **Resultados esperados:** | 1. Agregar un defecto:    1. Evitar creación 2. Modificar defecto:    1. Evitar modificación    2. Evitar modificación    3. Evitar modificación    4. Evitar modificación 3. Borrar defecto:    1. Permitir borrado    2. Evitar borrado    3. Evitar borrado |
| **Resultados obtenidos** | 1. Agregar un defecto:    1. Se envía una de las dos siguientes excepciones: No se admiten campos vaciós! O IllegalArgumentException! 2. Modificar defecto:    1. Cuando se deja la descripción vacía se envía la siguiente excepción: ControllerException: La descripción del defecto no puede estar vacía!    2. Permite modificación    3. Permite modificación    4. Permite modificación 3. Borrar defecto:    1. El esperado    2. El esperado    3. El esperado |
| **Notas y Preguntas:** | Ninguno. |