**Trabajo Práctico 5.6**

**Seguimiento de desviaciones**

**Integrantes:**

**Emanuel Alberto Márquez.**

**Catedra:**

**Gestión de Calidad.**

**Carrera:**

**Licenciatura En Sistemas.**

**Docentes:**

**Mg. Albert Osiris Sofía.**

**Esp. Eder Dos Santos.**

**Año: 2020.**

**Fecha de presentación: 19 de Mayo de 2020.**

INDICE

[*1.* *Realizar una evaluación de distintas herramientas open source de seguimiento de incidentes (Issue Tracking) y errores (Bug Tracking).* 1](#_Toc40789854)

[*2.* *En base a la evaluación realizada, seleccionar una herramienta para implementar en su proyecto LDS.* 7](#_Toc40789855)

[*3.* *Elaborar un procedimiento para el registro y seguimiento de las desviaciones encontradas en su proyecto, en la que se contemple la utilización de la herramienta seleccionada.* 7](#_Toc40789856)

[*4.* *Implementar la herramienta seleccionada.* 9](#_Toc40789857)

[*5.* *Modificar el plan de SQA elaborado durante la cursada de la asignatura; se debe incorporar el procedimiento y describir su implementación.* 12](#_Toc40789858)

1. *Realizar una evaluación de distintas herramientas open source de seguimiento de incidentes (Issue Tracking) y errores (Bug Tracking).*

**Bugzilla**

Es un sistema de seguimiento de errores. Como tal, permite a los equipo de desarrollo de software realizar seguimiento a los errores, problemas, mejoras y otras solicitudes de cambios en los productos de una forma efectiva. A pesar de ser gratuito, tiene muchas características que son capaces de competir con sus contrapartes pagas. Para poder ser utilizado, debe ser instalado en un servidor.

Características

Para los usuarios ofrece capacidades de búsqueda avanzada, notificaciones por correo electrónico, listados de errores, informes programados por correo electrónico, informes y gráficos. También ofrece seguimientos de tiempo, sistema de solicitudes, visor de parches y almacenamiento de búsqueda.

Para los administradores ofrece mecanismos de seguridad, personalización de campos, flujos de trabajo personalizados, soporte Unicode, interfaz de servicios web, métodos de autenticación y soporte para múltiples motores de bases de datos.

Apoyo o soporte:

* Documentación en línea.
* Soporte gratuito. Se describen opciones de instalación, administración y soporte de personalización ([**https://www.bugzilla.org/support/**](https://www.bugzilla.org/support/)). Este tipo de soporte viene sin garantía.
* Soporte pago. Utiliza consultores que pueden ayudar a ahorrar tiempo de instalación y administración.

([**https://www.bugzilla.org/support/consulting.html**](https://www.bugzilla.org/support/consulting.html))

Requerimientos del sistema

Sobre el sistema operativo

* Para instalar y ejecutar Bugzilla en un servidor, se requiere tener instalada Perl. Esto nos indica que se puede instalar en cualquier plataforma donde se pueda usar Perl, incluidos Windows, Linux y Mac OS.

Sobre el hardware

* Los requerimientos de este tipo dependen en gran medida del tamaño de la base de usuarios. Para un grupo pequeño con pocos errores, cualquier maquina hará el trabajo bien. Se recomienda tener suficiente RAM (4 GB mínimo) y un procesador rápido. Un disco con 50 GB de espacio libre es suficiente.

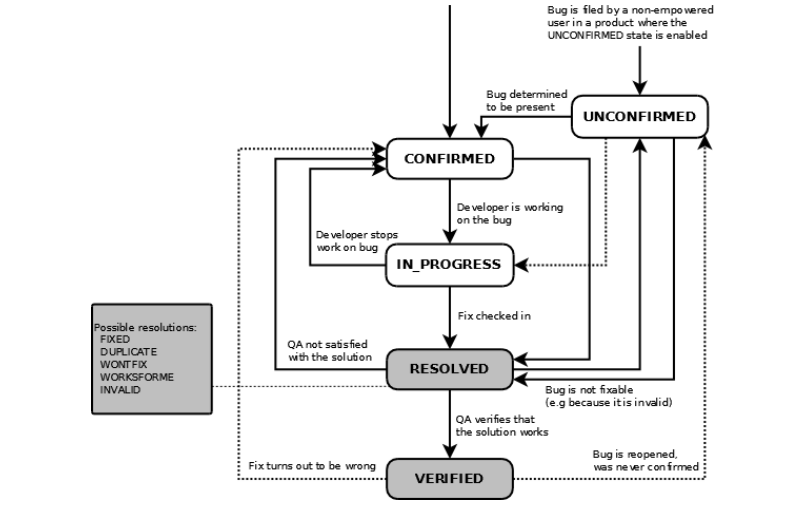
Sobre el software

Se necesita un servidor de base de datos, un servidor web y Perl.

* Perl. Se requiere una versión 5.8.1 o superior. Se recomienda 5.12 o posterior.
* Servidor de base de datos. Es compatible con MySQL, PostgreSQL, Oracle y SQLite. Se recomiendan MySQL (5.5 o posterior) y PostgreSQL (8.3) por el soporte.
* Se recomienda Apache 2.2 o IIS 7 o superior.

Workflow

El ciclo de vida de un error o workflow se puede personalizar según las necesidades de la organización. En forma predeterminada Bugzilla presenta el siguiente ciclo de vida para un bug:



**GLPI**

Es un sistema de seguimiento de incidencias que se encuentra desarrollado en PHP bajo una distribución GPL. Es una aplicación web que permite a las organizaciones construir un inventario de los recursos y de gestionar tareas administrativas. Sus funcionalidades ayudan a administrar activos y de proporcionar historiales de intervenciones.

Última versión estable: 9.4.5 (18/12/2019)



*Imagen: Demo GLPI*

Apoyo o soporte:

* Comunidad. Permite contribuir o participar de proyectos además de recibir ayuda de terceros. ([**https://glpi-project.org/community/**](https://glpi-project.org/community/))
* Listas de correo para preguntas sobre la utilización o instalación. ([**https://mail.ow2.org/wws/info/glpi-users**](https://mail.ow2.org/wws/info/glpi-users))
* Redes sociales.

Requerimientos del sistema

GLPI es una aplicación web que necesita: un servidor web, PHP y una base de datos.

Sobre el servidor web

* Apache 2 o posterior.
* Nginx.
* Microsoft IIS.

Sobre PHP

* A partir de la versión 9.2 de GLPI se requiere PHP 5.6 o superior. Se recomienda usar la versión más estable.
* Extensiones obligatorias: CURL, fileinfo, gd, json, mbstring, mysqli, sesión, zlib, simplexml y xml.
* Extensiones opcionales: CLI, domxml, imap, ldap, openssl, xmlrpc, APCu.

Sobre la base de datos

* Actualmente solo admite servidores de base de datos MySQL 5.6 o posterior y MariaDB (mínimo 10.0)

**Mantis Bug Tracker**

Mantis Bug Tracker es un sistema de seguimiento de errores basado en la web que se encuentra desarrollado en PHP y con soporte a múltiples bases de datos (MySQL, MS SQL y PostgreSQL). Mantis BT se puede ejecutar sobre cualquier sistema operativo que tenga compatibilidad con PHP y que soporte el DBMS seleccionado.

Apoyo o soporte:

* Foros.
* Lista de correos. Con previa suscripción.
* Gitter. Es un chat en línea.
* Wiki. El Wiki de Mantis contiene preguntas frecuentes, requisitos, complementos, etc.

Requerimientos del sistema

Sobre los requisitos del hardware para el servidor:

* El servidor puede ser un servidor web público compartido o uno dedicado.
* CPU y memoria debe ser dimensionada en función del tráfico en el sitio.
* El código de la aplicación tiene menos de 30 MB. Se debe analizar el espacio en disco para la base de datos.

Sobre los requisitos de software para el servidor:

* Mantis BT se ejecuta en Windows, macOS, Linux, Solaris y casi todo que admita el software del servidor.
* Se espera que funcione con cualquier software de servidor web.
* Mantis solo usa archivos PHP.
* Sobre las extensiones de PHP se debe contar en forma obligatoria con: extensión para el RDBMS que se utiliza y mbstring para el soporte Unicode. En forma opcional se puede contar con: Curl, GD, LDAP.
* Mantis BT requiere de una base de datos.

Sobre los requisitos del clientes

* Navegador como: Firefox, Internet Explorer 11, Borde, Safari, Opera.

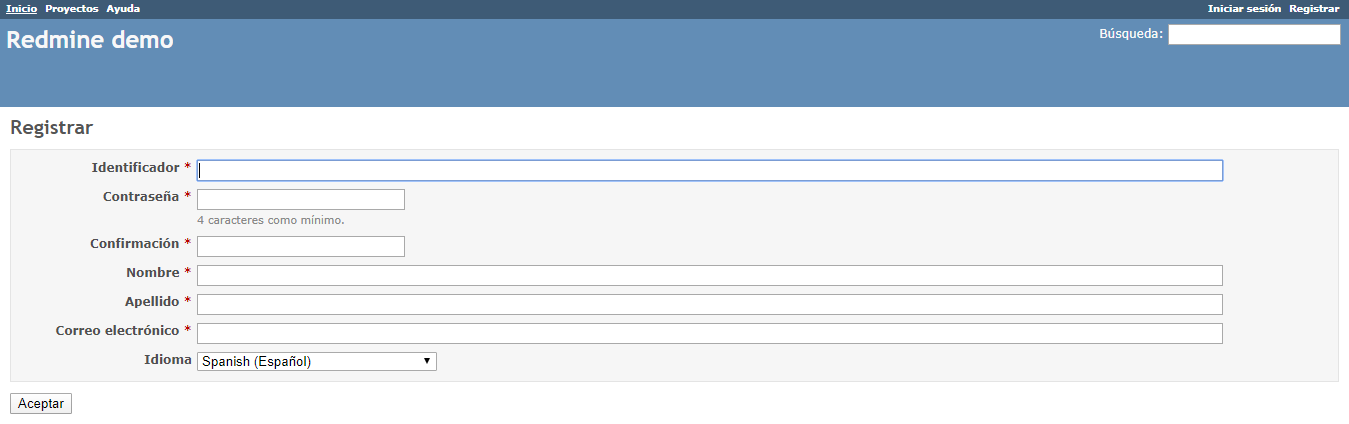
Gestión de usuarios

En Mantis BT no hay límite en la cantidad de cuentas de usuarios que se puedan crear. La acción de creación de usuarios es sencilla dado que solo se necesita ingresar el nombre, email y tipo de usuario. Esto desencadena dos acciones, almacenar datos en la base y notificación por correo electrónico al usuario (en caso de estar activadas). Al momento de quitar un usuario solo se debe deshabilitar para mantener el registro histórico.

**Redmine**

Redmine es una herramienta para la gestión de proyectos que permite organizar y hacer seguimiento de los mismos. Junto con esto ofrece un sistema de seguimientos de incidentes con seguimiento de errores. Es un sistema de software libre y de código abierto que está desarrollado usando el framework Ruby On Rails.

Última versión estable: 4.1.1 (06/04/2020).



*Imagen: Demo Redmine*

Apoyo o soporte:

* Foros ([**https://www.redmine.org/projects/redmine/boards**](https://www.redmine.org/projects/redmine/boards))
* Wiki ([**https://www.redmine.org/projects/redmine/wiki**](https://www.redmine.org/projects/redmine/wiki))

Características generales

* Gestión de usuarios, roles y permisos.
* Gestión de proyectos.
* Autenticación mediante LDAP.
* Notificaciones por correo electrónico.
* Seguimiento de peticiones, planificación, controles de tiempo, diagramas de Gantt, documentos, estadísticas y registro de cambios.

Requerimientos del sistema

Sobre el sistema operativo

* Se puede ejecutar en Unix, Linux, MacOS y Windows. La única condición es que Ruby esté disponible en la plataforma.
* Interprete de Ruby. Las versiones requeridas son las 2.2 o superior para una versión 4.0 o 4.1 de Redmine. Una aclaración importante es que Redmine no admite JRuby porque no se soportan Rails.

Sobre la base de datos

* MySQL 5.5 – 5.7.
* PostgreSQL 9.2 o superior.
* Microsoft SQL Server 2012 o superior.
* SQLite 3 pero no para uso en producción multiusuario.

Componentes opcionales

* Binarios de SCM.
* ImageMagick (Para diagramas de Gantt)
* Ghostscript.

**Trac**

Trac es un Wiki mejorado y un sistema de seguimientos de problemas para proyectos de desarrollo de software. Además, proporciona un interfaz para sistemas de control de versiones como Git, un Wiki integrado y medios de notificaciones. Trac es desarrollado por Edgewall Software con software libre y código abierto escrito en Python.

Última versión estable: 1.4.1.

Apoyo o soporte:

* Listas de correos ([**https://trac.edgewall.org/wiki/MailingList**](https://trac.edgewall.org/wiki/MailingList)).
* Preguntas frecuentas ([**https://trac.edgewall.org/wiki/TracFaq**](https://trac.edgewall.org/wiki/TracFaq))

Requerimientos del sistema

Para Trac el servidor web es opcional porque se envía con un servidor incluido. En forma alternativa se puede configurar para que se ejecute en cualquiera de los siguientes entornos:

* Apache.
* Un servidor FastCGI.
* Microsoft IIS con FastCGI.
* Un servidor con capacidad CGI.

Sobre los prerrequisitos para la versión estable de Trac

* Tener Python 2.7
* Jinja2. Para el procesamiento HTML.
* Genshi.
* Babel.

Sobre la base de datos

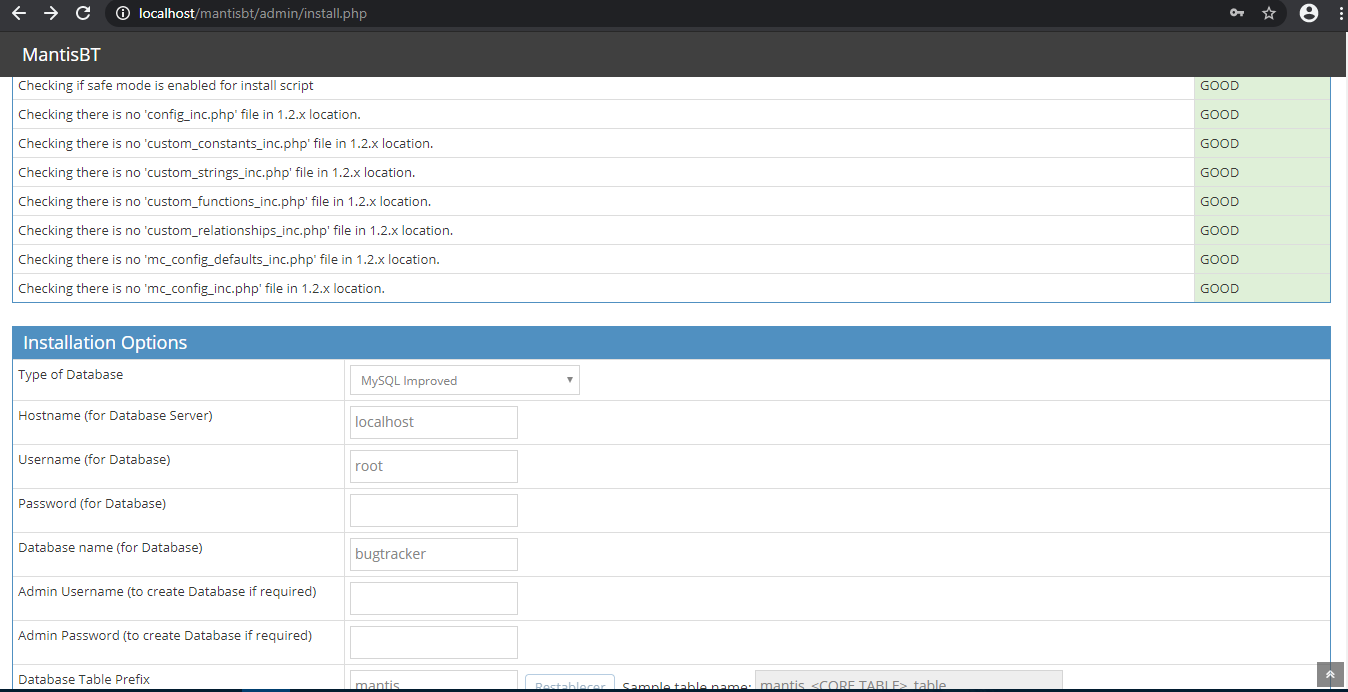
* MySQL. Version 5.0 o posterior.
* PostgreSQL. Version 9.1 o posterior.
* SQLite.

1. *En base a la evaluación realizada, seleccionar una herramienta para implementar en su proyecto LDS.*

Para implementar en el proyecto de desarrollo de software se ha seleccionado la herramienta **Mantis Bug Tracker**. Las razones por las cuales se hace elección de Mantis es que se encuentra desarrollado en PHP y por qué se disponen de los requerimientos para su configuración.

Una de las ventajas de utilizar Mantis, es que el proyecto Tempus de LDS también se encuentra desarrollado bajo el mismo lenguaje de programación. Acceder al código fuente de este Bug Tracker nos da la oportunidad de interpretar de una forma más sencilla su funcionamiento y observar cómo terceros desarrollan software.

En la siguiente imagen observamos la pantalla de configuración de Mantis Bug Tracker una vez que se ha desplegado de forma local en nuestro XAMPP:



1. *Elaborar un procedimiento para el registro y seguimiento de las desviaciones encontradas en su proyecto, en la que se contemple la utilización de la herramienta seleccionada.*

En esta secciona se detallan dos procedimientos. El primer procedimiento es el que se debe utilizar para la creación de nuevas incidencias sobre nuestro proyecto. El segundo, es el procedimiento para el seguimiento y resolución de las incidencias creadas.

**Procedimiento de creación de incidencias**:

1. Ingresar a Mantis Bug Tracker.
2. Seleccionar el proyecto “Tempus” sobre el menú superior “Todos los proyectos”.
3. Reportar una nueva incidencia
   * Establecer la categoría (Proyecto tempus)
   * Establecer la severidad (Funcionalidad, trivial, texto, ajuste, menor, mayor, fallo o bloqueo)
   * Establecer la prioridad (Ninguna, baja, normal, alta, urgente, inmediata)
   * Asignar a quien corresponda.
   * Resumen de la incidencia.
   * Descripción detallada de la incidencia.
4. Crear incidencia.

*Nota*: *Se recomienda bridar el máximo detalle que permita al agente resolver la incidencia en forma rápida y satisfactoria. Se pueden incluir imágenes o documentación de apoyo.*

**Procedimiento para resolución de incidencias**

1. Ingresar a Mantis Bug Tracker.
2. Seleccionar el proyecto “Tempus” sobre el menú superior “Todos los proyectos”.
3. Sobre el menú lateral seleccionar “Ver incidencias”
4. Seleccionar la incidencia asignada al agente o sin asignar. En caso de no estar asignada, se debe indicar “Asignar a” y auto-seleccionarse (si corresponde).

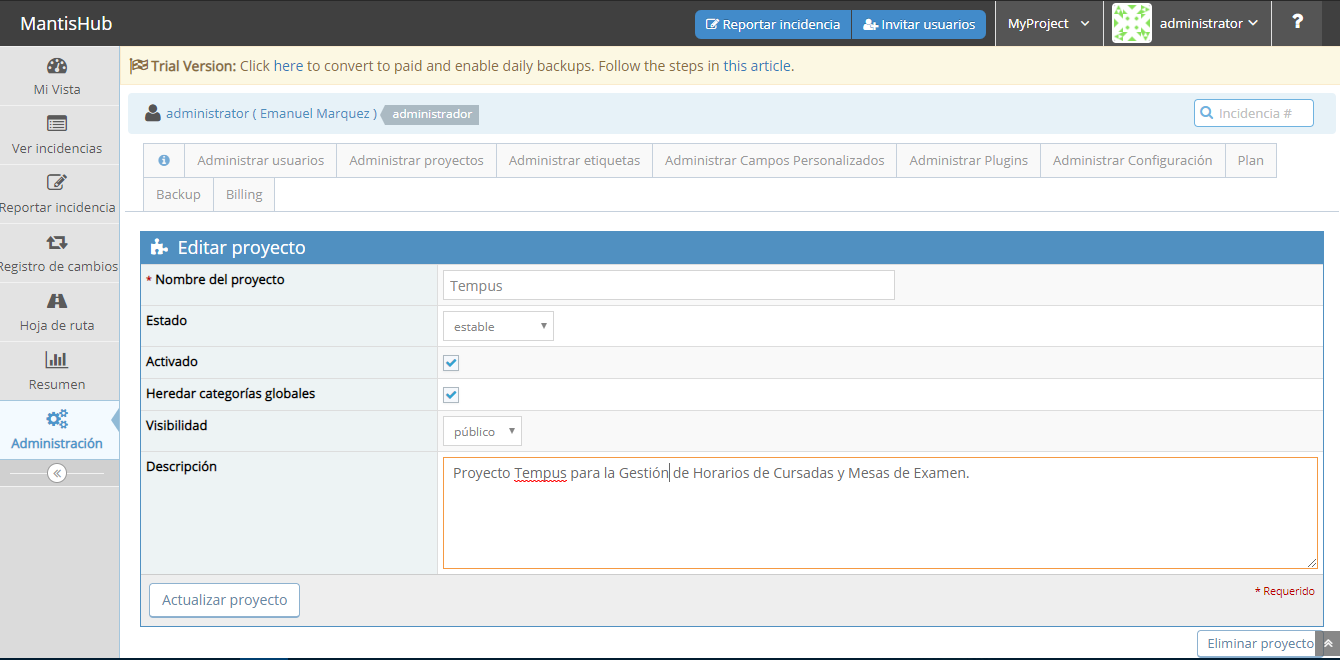
Sobre el cambio de estado:

* + Se necesitan más datos. No se puede proceder con el incidente porque faltan datos que detallen la misma.
  + Aceptada. El responsable acepta la incidencia. Comienza a trabajar en ella.
  + Confirmada. Se confirman los detalles que se sugieren por parte del creador de la incidencia.
  + Asignada. Se refleja que la incidencia se ha asignado a algún miembro del equipo y que ese miembro está trabajando en ella.
  + Resuelta. Cuando la incidencia ha sido resuelta.
  + Cerrada. La incidencia está completamente cerrada y no se requieren medidas con respecto a ella.

1. *Implementar la herramienta seleccionada.*

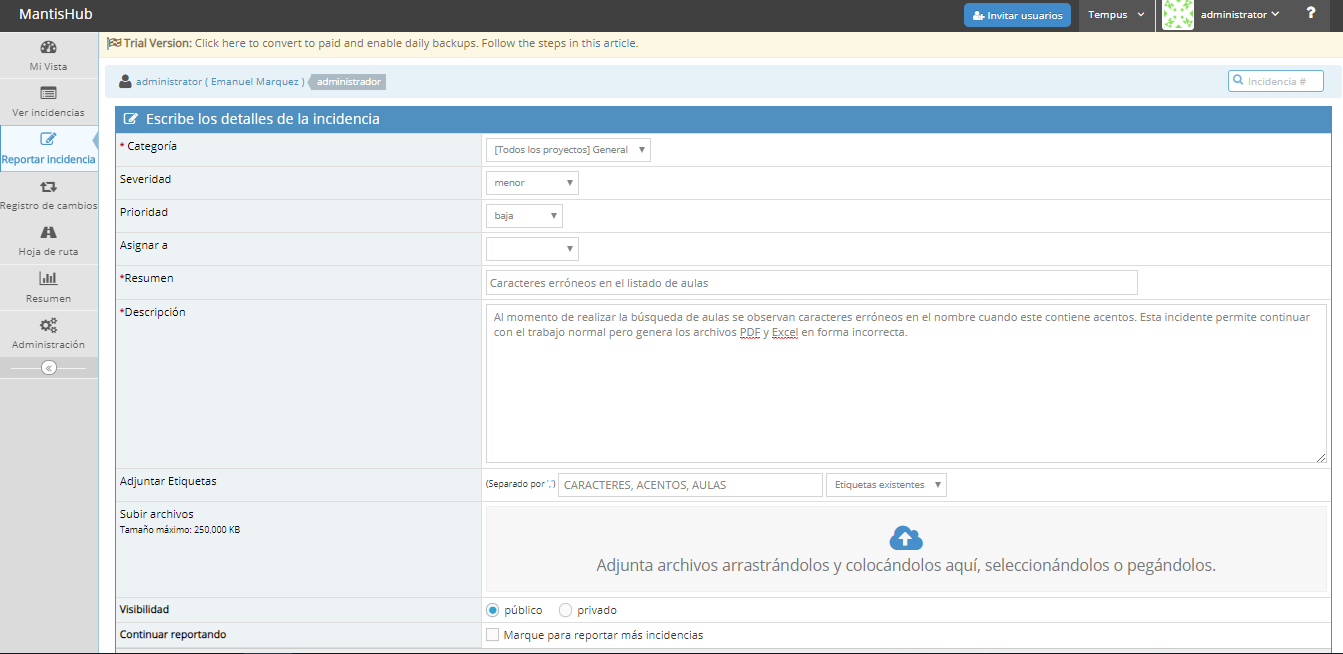
El código adecuado se adjunta como parte de la entrega. A continuación observamos algunas de las modificaciones incorporadas:

Configuración de nuestro proyecto. Realizamos la edición de nuestro proyecto denominándolo “Tempus”, en estado activo y con una breve descripción.



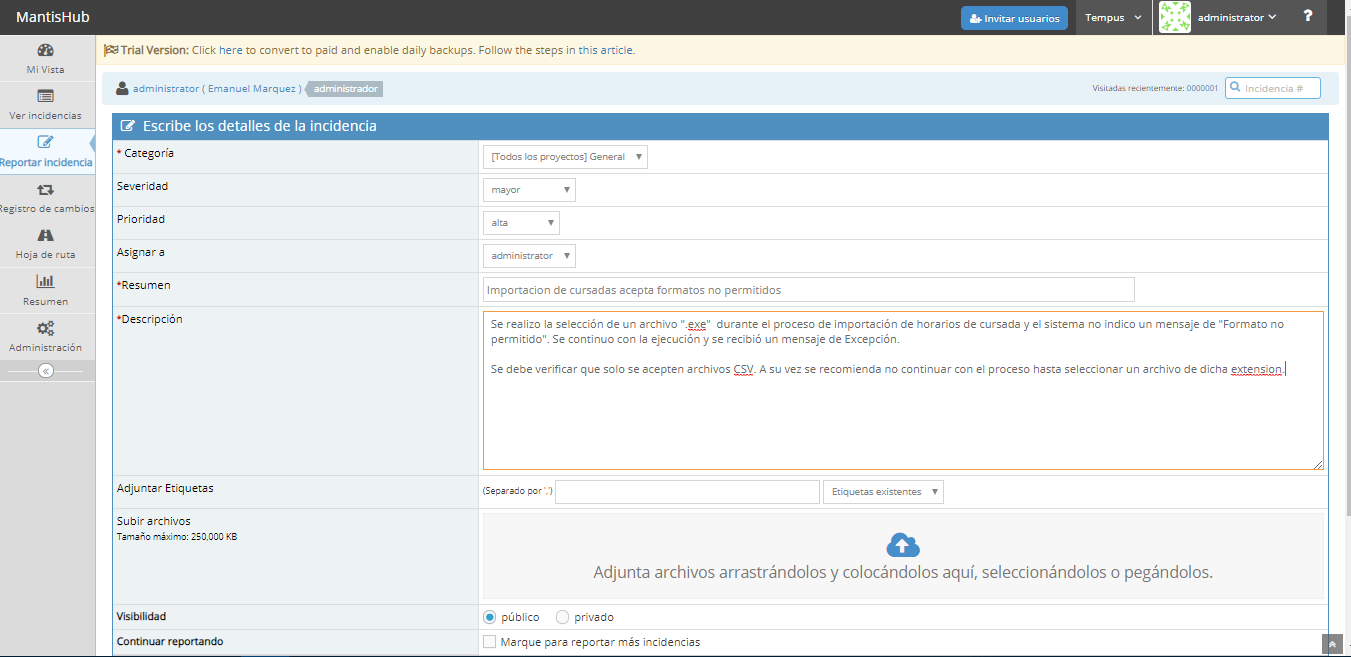
*Imagen: Editar proyecto*

En la siguiente imagen observamos la pantalla de creación de una nueva incidencia. Se puede ver que se sigue el procedimiento detallado anteriormente. Para esta nueva incidencia establecemos la categoría **General**, un nivel de severidad **Menor**, prioridad **Baja**, un resumen y descripción detallada.



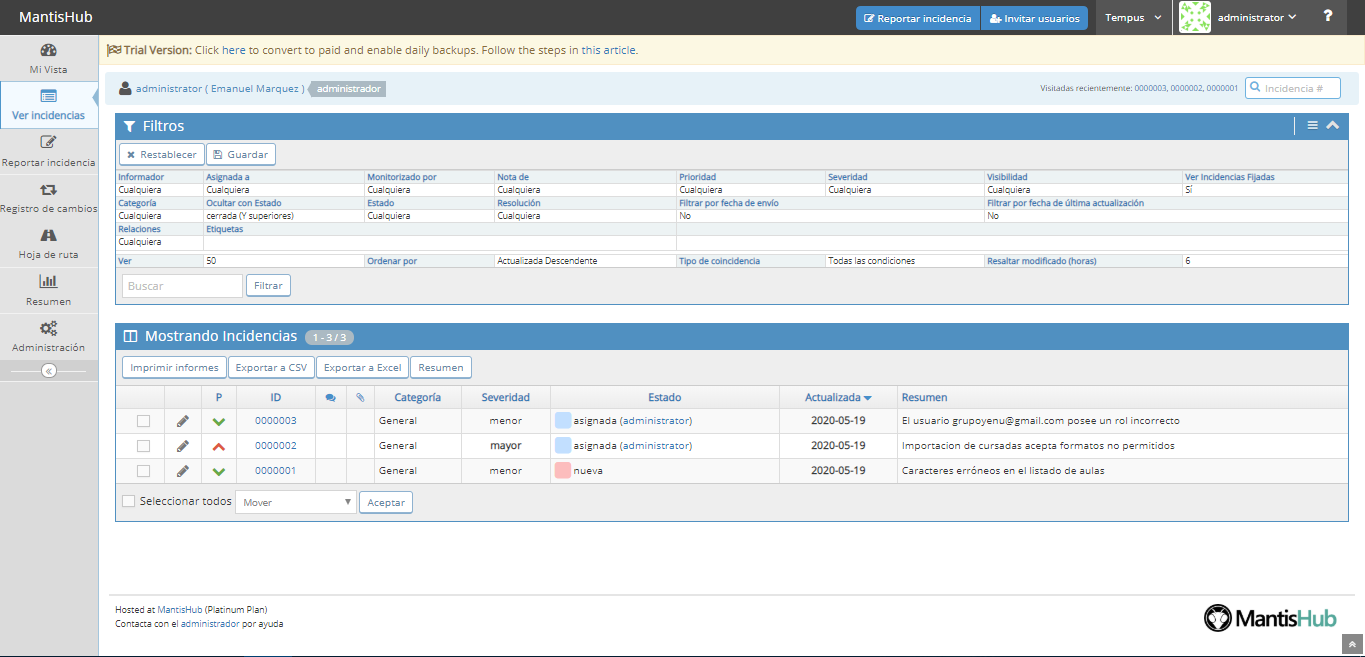
*Imagen: Nueva incidencia*

En el siguiente caso, realizamos la creación de una nueva incidencia pero asignándola a un miembro del equipo de desarrollo. En este caso, solo tenemos configurado a un administrador.



*Imagen: Nueva incidencia con asignación*

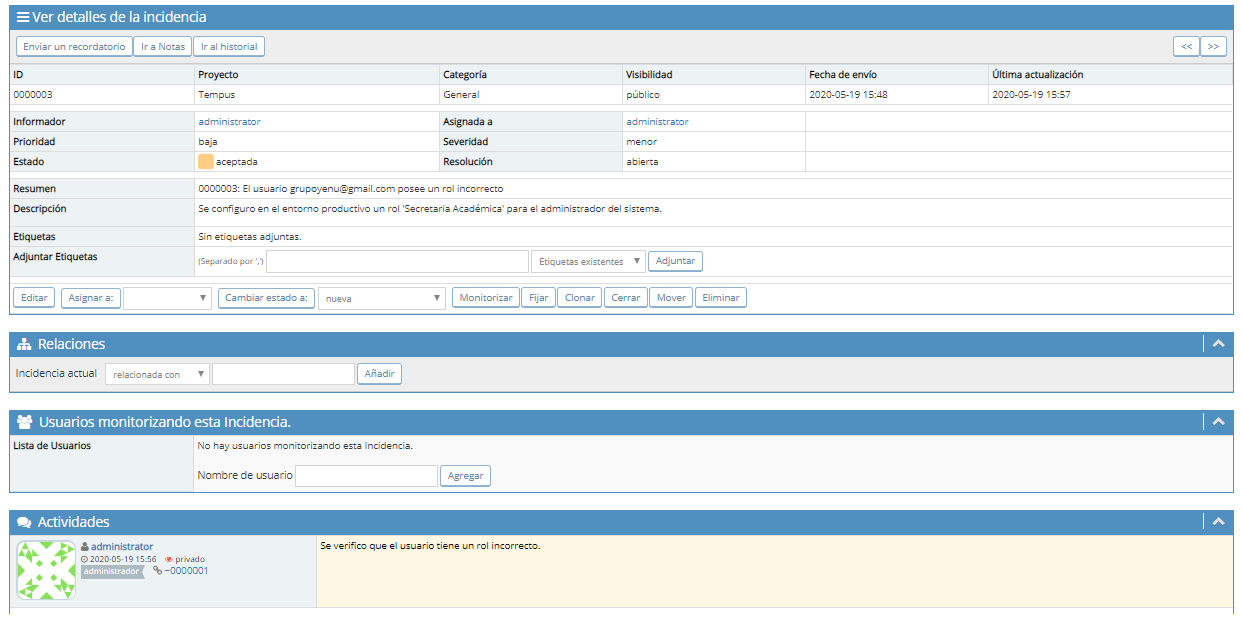
Utilizando el procedimiento para la resolución de incidencias realizamos lo siguiente. Sobre el menú lateral de “Ver dependencias” nos aparece la tabla con el listado de incidencias que se han creado.



*Imagen: Listado de incidencias*

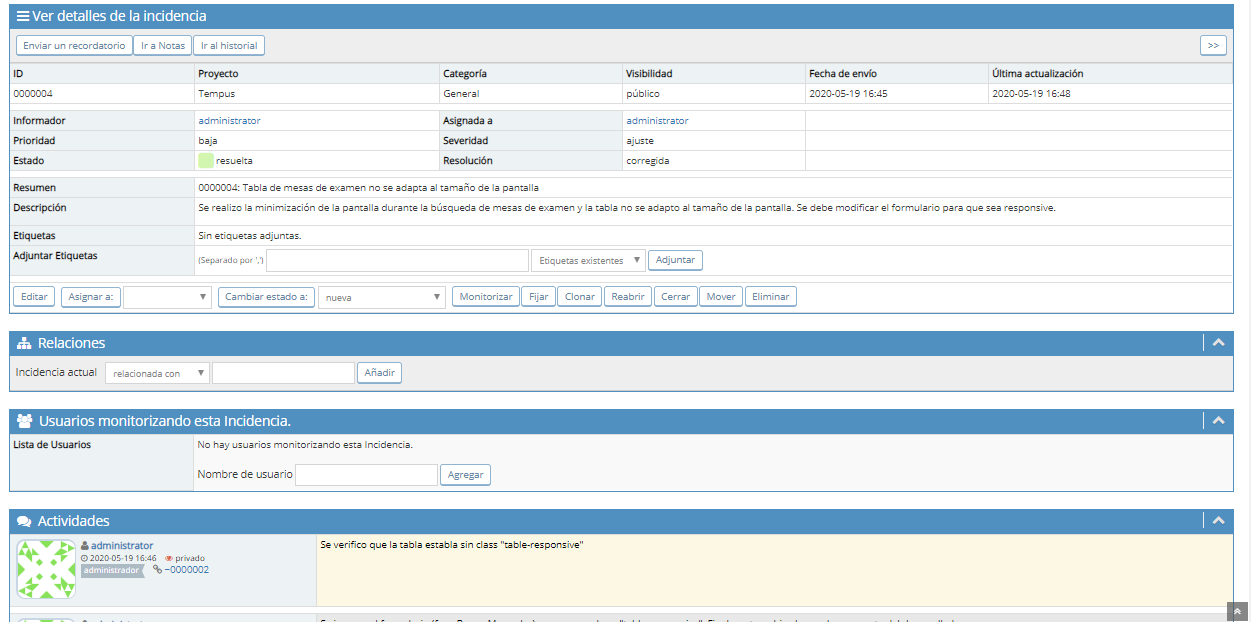
En caso de ver solicitudes asignadas a mi deberíamos abrirla para indicar si es “Aceptada” o si “Se necesitan más datos”.

Seleccionamos un incidente y aceptamos la información brindada por el creador:



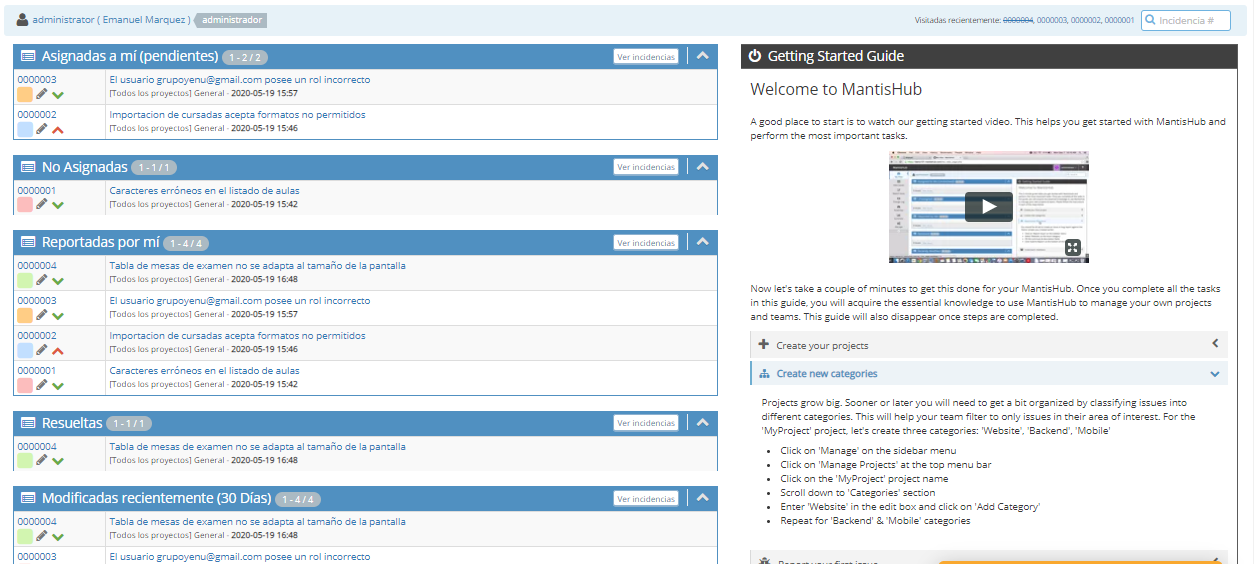
*Imagen: Incidencia aceptada*

Ahora observamos una solicitud que fue “Aceptada”, se trabajó en su solución y paso a un estado de “Resuelta”:



*Imagen: Incidencia resuelta*

Mantis Bug Tracker nos brinda un panel de acceso rápido para identificar las incidencias asignadas o creadas:



*Imagen: Panel Mi vista*

1. *Modificar el plan de SQA elaborado durante la cursada de la asignatura; se debe incorporar el procedimiento y describir su implementación.*

Se adjunta el Plan SQA en la entrega del presente trabajo práctico.

.