Plan de Gestión de Configuración

Grupo Nº 1 – Gestión de Calidad de Software

Sartini Nicolás – Muñoz Rut – Cadin Vanesa

**

*Este documento describe todas las actividades de Gestión de Configuración y Cambios que serán realizadas durante todo el ciclo de vida del proyecto.*

*El mismo nos proporciona planificaciones detalladas de las actividades, responsabilidades asignadas, recursos necesarios que incluyen personal, herramientas y equipamiento.*

**Tabla de contenido**

[Introducción 5](#_Toc75184641)

[Propósito 5](#_Toc75184642)

[Alcance 5](#_Toc75184643)

[Gestión de Configuración 5](#_Toc75184644)

[Herramientas, Entorno e Infraestructura 5](#_Toc75184645)

[Herramienta Git - GitHub 5](#_Toc75184646)

[Organización 5](#_Toc75184647)

[Instalación de Git de manera local 6](#_Toc75184648)

[Creación de repositorio remoto 6](#_Toc75184649)

[Clonar el repositorio 7](#_Toc75184650)

[Herramienta Mantisbt 7](#_Toc75184651)

[Organización 7](#_Toc75184652)

[Instalación de Mantisbt de manera local con xampp 8](#_Toc75184653)

[Revisión de parámetros que deben estar instalados 8](#_Toc75184654)

[Opciones de Instalación y parámetros 8](#_Toc75184656)

[Usuario creado por defecto para el ingreso a la aplicacion 9](#_Toc75184657)

[Ubicación del proyecto en la nube 9](#_Toc75184658)

[Control de Configuración 10](#_Toc75184659)

[Identificación de la incidencia 10](#_Toc75184660)

[Dejar registro de la solicitud de cambio 10](#_Toc75184661)

[Solicitud de Cambios 10](#_Toc75184662)

[Aprobación de Cambios 12](#_Toc75184663)

[Implementación de Cambios 12](#_Toc75184664)

[Monitoreo 12](#_Toc75184667)

[Dar por resuelta la incidencia 13](#_Toc75184668)

[Plantilla de Solicitud de cambio 13](#_Toc75184669)

[Estado de la Configuración 14](#_Toc75184670)

[Informes y Auditorías 14](#_Toc75184671)

Plan de Gestión de Configuración

Introducción

Propósito

Este documento describe las actividades de gestión de configuración de software que deben ser llevadas a cabo durante el proceso de desarrollo del proyecto. Aquí se definen tanto los productos que se pondrán bajo control de configuración como los procedimientos que deben ser seguidos por los integrantes del equipo de trabajo.

Alcance

El ámbito de este documento es el proyecto Mueblería y establece un plan para administrar los productos de trabajo del proyecto, incluyendo tanto los entregables de software como la documentación del proyecto.

Gestión de Configuración

Herramientas, Entorno e Infraestructura

Herramienta Git - GitHub

Git es un software de control de versiones desarrollado por Linus Torvalds. Su propósito consiste en llevar un registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de computadora.

Organización

El administrador de la configuración será Nicolás Sartini.

Instalación de Git de manera local

Primero instalamos Git en nuestra máquina, desde el siguiente enlace: <https://gitforwindows.org/>

Una vez descargado el ejecutable (.exe) le daremos siguiente al instalador, hasta finalizar.

Nota: En uno de los pasos, debemos seleccionar VS Code como editor de texto predeterminado, ya que es el que utilizamos para el proyecto.

Una vez instalado Git, configuramos el entorno desde Git Bash (consola integrada de Git) mediante los siguientes comandos, donde en el primero debemos colocar nuestro nombre de usuario y en el segundo el email, ya que ambas credenciales se asociarán con nuestra cuenta de GitHub para su posterior utilización.

git config --global user.name " nombre\_usuario "

git config --global user.email " [ejemplo@dominio.com](mailto:ejemplo@dominio.com) "

Creación de repositorio remoto

Una vez configurado Git en nuestra máquina local, procedemos a crear un repositorio vacío desde <https://github.com/> , el cual en nuestro caso, se llama Proyecto-Muebleria.

Luego, para subir nuestros archivos locales al repositorio remoto, nos posicionamos en el interior de nuestro proyecto y ejecutamos los siguientes comandos:

git init → Genera un repositorio local.

git add . → Agrega todos los archivos al Staging Area (Área de preparación).

Git commit -m “mensaje de commit inicial” → Realiza un commit (lleva los archivos al repositorio para el control de cambios).

git branch -M main → Renombra la rama “Master” por “Main”.

git remote add origin <https://github.com/Nicolas-Sartini/Proyecto-Muebleria.git> → Agrega nuestro repositorio local (origen) al repositorio remoto (destino)

git push -u origin main → Carga el contenido de nuestro repositorio local al remoto (confirma el envío)

Por último, desde VS Code crearemos una nueva rama, llamada “responsive”, que será una nueva versión de nuestro proyecto, la cual incorporará Bootstrap. Para finalizar, realizamos un commit y push para subir la nueva rama al repositorio remoto.

Clonar el repositorio

Este apartado es para quienes deseen colaborar en el desarrollo del proyecto. Para esto, deben dirijirse a la página de GitHub, donde en la raíz del repositorio “Proyecto-Muebleria” deben clickear el botón “Code” donde se desplegarán opciones para clonar el repositorio via Https, SSL, GitHub CLI.

En este caso, lo clonaremos mediante Https. Para ello, copiamos el enlace del repositorio y nos dirijiremos a VS Code, donde elegiremos la opción “Clone Git Repository” que se muestra al inicio. A continuación, seleccionaremos la ubicación donde desearemos clonar el repositorio (C:\xampp\htdocs) y procedemos a pegar el enlace copiado y presionaremos enter.

Con esto, ya contaremos con el repositorio clonado en nuestra máquina, listo para colaborar.

Herramienta Mantisbt

Mantisbt es una aplicación Open Source que sirve para controlar los errores aparecidos en el software y que permite a desarrolladores, testers o clientes reportar fallos y realizar el seguimiento de los mismos hasta su resolución.

También funciona como unaplataforma de colaboración para proyectos en los que participan diversos perfiles a los que se les asigna distintos roles para gestionar permisos.

Basado en acceso web y adaptable a todo tipo de pantallas**,** tanto móviles como de escritorio, Mantis está desarrollado en tecnologías abiertas como PHP y da soporte a clientes web populares como Chrome, Firefox, Safari e Internet Explorer desde la versión 7 en adelante.

Organización

El equipo de trabajo está estructurado de la siguiente forma:

|  |  |  |  |
| --- | --- | --- | --- |
| Persona  Rol | Rut Muñoz | Vanesa Cadin | Nicolás Sartini |
| Informador | X |  |  |
| Supervisor |  | X |  |
| Desarrollador |  |  | X |

Instalación de Mantisbt de manera local con xampp

* Descarga: <https://sourceforge.net/projects/mantisbt/>
* Descomprimir el archivo descargado.
* Copiar contenido en la dirección C:\xampp\htdocs
* Iniciar la instalación en <http://localhost/mantisbt/>admin/install.php

Revisión de parámetros que deben estar instalados



Opciones de Instalación y parámetros

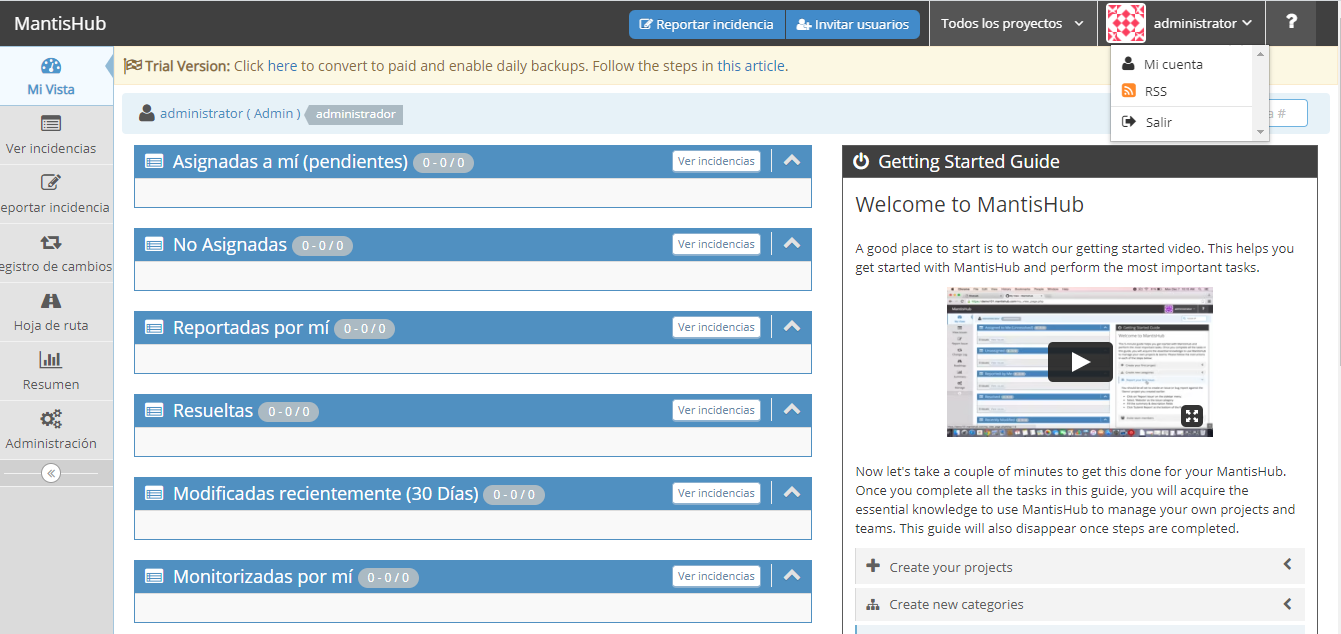
* Type of Database: elegir motor de base de datos (MySQL, ORACLE, Postgre SQL).
* Hostname(for Database Server): localhost.
* Username(for Database): root
* Password(for Database): contraseña
* Database name(for Database): bugtracker
* Admin Username(for Database): opcional
* Admin Password(for Database): opcional
* Database Plugin table Prefix: mantis
* Database table suffix: \_table
* Default Time Zone : Argentina

Usuario creado por defecto para el ingreso a la aplicacion

* User: Administrator
* Password: root

Ubicación del proyecto en la nube

* Ingresar a la página: <https://www.mantisbt.org/>
* Crear un usuario en la sección hosting: <https://www.mantisbt.org/hosting.php>
* Ingresar al servicio: <https://gestion2021.mantishub.io/>



* Creación del Proyecto en la aplicación



* Incorporación de usuarios



Control de Configuración

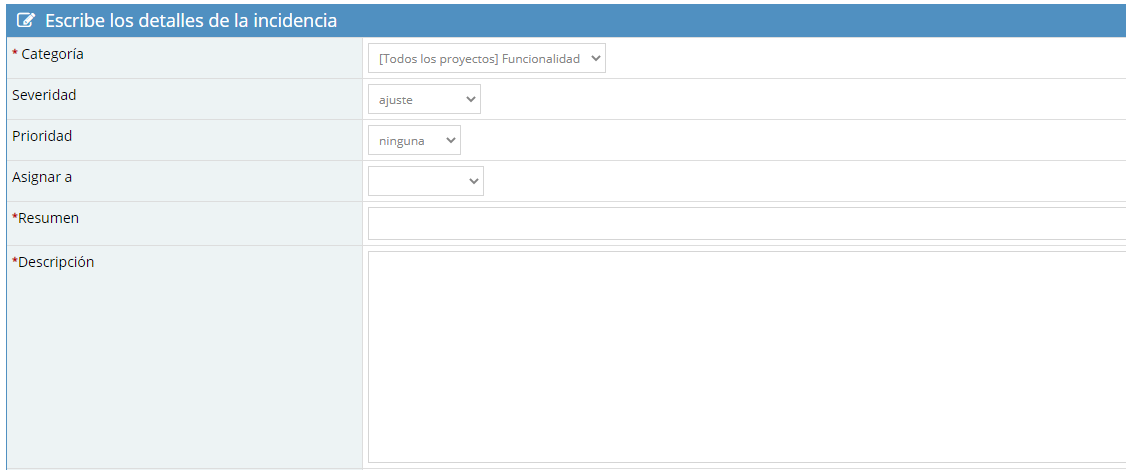
Identificación de la incidencia

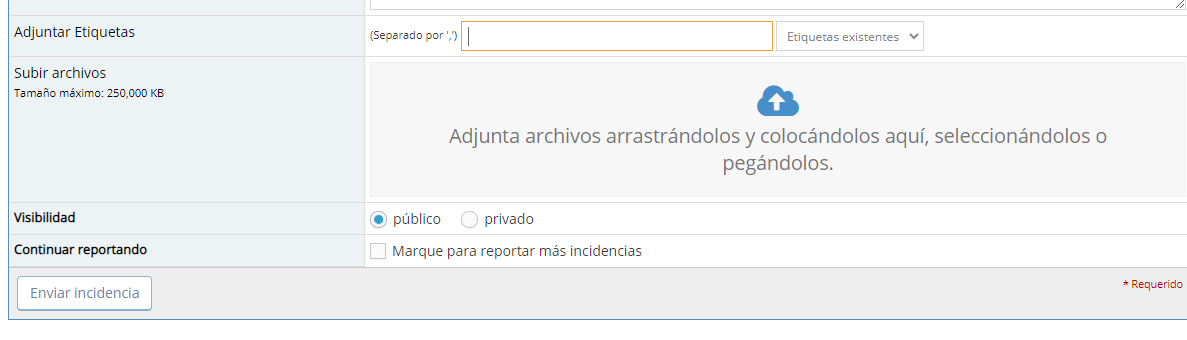
Una vez que se identifica la incidencia se informa al sector correspondiente y a la persona encargada para iniciar el procedimiento de control.

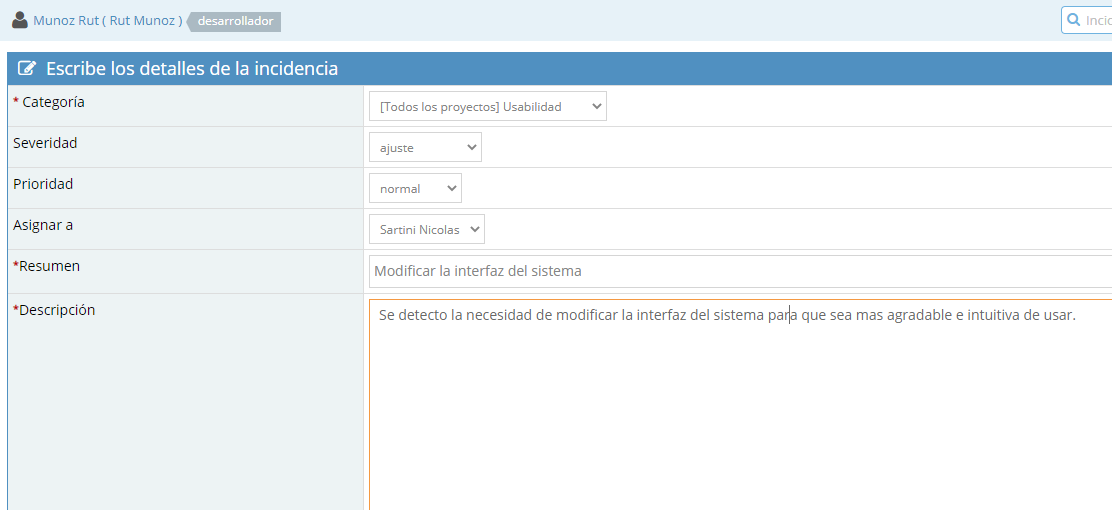
Dejar registro de la solicitud de cambio

En el sistema cargamos el reporte de incidencia.

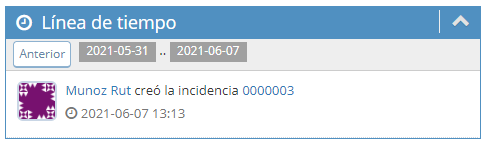
Solicitud de Cambios







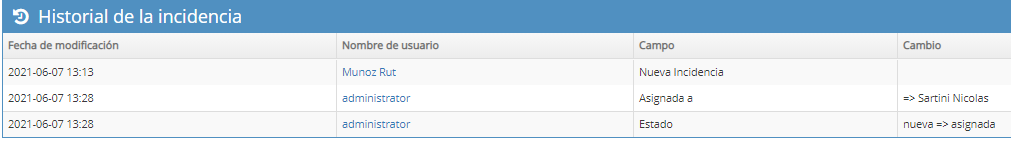


En el sistema se ve reflejado la creación de la incidencia.

Aprobación de Cambios

Revisar la incidencia para darle curso o no. En caso de proceder por la primera opción se asigna un responsable para su resolución.





Implementación de Cambios





Monitoreo

Una vez resuelta la incidencia e implementada, se inicia un control sobre el cambio realizado.



Dar por resuelta la incidencia

En caso de no surgir ningún inconveniente, concluye el monitoreo y el administrador cambia el estado de la incidencia como corregida y se concluye el proceso.





Plantilla de Solicitud de cambio

A continuación, se propone una plantilla de ejemplo para realizar las solicitudes de cambio pertinentes.

|  |  |
| --- | --- |
| ***Número:*** *[xxxx]* | **Estado:** Se actualiza el estado del cambio: SOLICITADO, APROVADO, RECHAZADO, IMPLEMENTADO, CANCELADO |
| ***Fecha de Solicitud*** | Indicar la fecha en la que se solicita el cambio. |
| ***Solicitante*** | Nombre y Rol de la persona que solicita el cambio. |
| ***Nombre del Elemento*** | Indicar el nombre del elemento que se solicita cambiar. |
| ***Urgencia*** | Indicar la urgencia del cambio en cuestión: ALTA, MEDIA, BAJA. Se debe ser muy objetivo en esta asignación, ya que las divisiones de desarrollo y producto deben aplazar actividades urgentes por resolver estas solicitudes |
| ***Descripción Detallada*** | Escribir claramente el detalle del error o la solicitud de mejora, indicando la mayor cantidad de detalles que describan el error: Ej.: el mensaje mostrado, la pantalla, la base de datos usada, etc. |
| ***Necesidad*** | Indicar la necesidad de llevar a cabo el cambio solicitado. |
| ***Impacto*** | Indicar la influencia y magnitud del efecto generado por el cambio propuesto. |
| ***Observaciones*** | Mencionar cualquier detalle significativo relacionado con la solicitud, si fuese necesario. |

Estado de la Configuración

Informes y Auditorías

Cada cierto tiempo, el gestor de configuración y cambios realizará un informe para el jefe de proyecto con el fin de revisar la evolución de los defectos que se vayan registrando en el mismo.

En este informe (“Informe de Evolución de Defectos”) se detallará los defectos detectados, sus prioridades, responsables de corregirlos, su estado y el procedimiento que se ha seguido o se va a seguir a la hora de resolverlos.

Este nos permitirá obtener una serie de indicadores para determinar la calidad del producto que se está desarrollando, permitiendo al jefe de proyecto informar al cliente sobre este aspecto.

Los tipos de informes que se van a desarrollar para el estado actual de los defectos son los siguientes:

• Tabla(s) que muestre(n) el estado actual de cada defecto y su severidad.

• Tabla(s) que muestre(n) el estado actual de cada defecto y su prioridad.

Las auditorías que se llevarán a cabo para comprobar si los cambios se han realizado correctamente serán al final de cada iteración, pero antes de que se cree una línea base. En éstas se revisarán tanto los requisitos funcionales y de rendimiento, como que el producto cumpla con las especificaciones detalladas en las que se define. Tomarán parte en éstas el cliente, el jefe de proyecto y el gestor de configuración.