**PLAN DE GESTIÓN DE LA CONFIGURACIÓN**

**PROYECTO SISTEMA DE INVENTARIO PARA BODEGA “EL AMIGO”**

**VERSIÓN 1.0**

**Historial de Versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 30/04/2015 | 1.0 | Versión preliminar como una propuesta de desarrollo. | Equipo GoSoft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contenido

[**1.** **Introducción** 4](#_Toc449722307)

[**1.1.** **Propósito** 4](#_Toc449722308)

[**1.2.** **Definiciones, siglas y abreviaturas** 4](#_Toc449722309)

[**1.3.** **Referencias** 5](#_Toc449722310)

# **Introducción**

MR&JP es una empresa de consultoría en tecnologías de información, especializada en el desarrollo de aplicaciones y soluciones de negocio para medianas y grandes empresas en los sectores educativo, industrias diversas, empresas de servicio y gobierno. Tenemos más de 7 años de experiencia en el desarrollo e implementación de aplicaciones y soluciones tecnológicas.

Buscamos cumplir siempre con las expectativas de nuestros clientes en cada proyecto, por lo que proporcionamos un servicio con un alto sentido de responsabilidad y compromiso. Para ello nos enfocamos en la mejora continua de nuestras aplicaciones, las cuales son revisadas por nuestro equipo para ser entregadas con los más altos estándares de calidad.

## **Propósito**

< debe incluir información sobre el propósito específico de las actividades de SCM que serán definidas en el plan, por ejemplo, si el énfasis está dado en un control riguroso, en una rápida respuesta a los cambios, en la documentación, entre otros >

## **Alcance**

< debe establecerse brevemente el alcance de las tareas de SCM, identificando intereses y responsabilidades específicas, lo que se incluye en el plan y lo que no se incluye, información sobre los ítems en la configuración, tipo de control sobre cada ítem, etc. >

## **Definiciones, siglas y abreviaturas**

< incluye las definiciones de los términos necesarios para entender el Plan de SCM que ayuden a la comunicación entre los integrantes del grupo>

## **Referencias**

< incluye la lista de documentos que son referenciados en el Plan de SCM >

# **Gestión de la Configuración del Software**

< el tema de esta sección es relacionar los elementos de la disciplina de SCM con las actividades específicas del proyecto y/o de SCM en la institución. Se especificarán organización, responsabilidades, agenda y recursos >

* 1. **Organización de la SCM**

< se especifican las funciones que debe cumplir cada entidad en la organización, teniendo en cuenta la estructura y como asignar y coordinar de la mejor forma posible las actividades de SCM que serán desarrolladas>

* 1. **Roles y Responsabilidades de la SCM**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del Rol** | **Persona asignada** | **Responsabilidades** | **Niveles de Autoridad** |
| Gestor de la configuración |  | Ejecutar todas las tareas de Gestión de la configuración |  |
| Bibliotecario |  | Diseñar y establecer la biblioteca del software para cada proyecto de desarrollo durante la etapa de planificación y la documentación respectiva. |  |
| Miembro del comité de cambio |  | Analizar las solicitudes de cambio, teniendo en cuenta el impacto y los riesgos que puede causar. |  |
| Aseguramiento de la calidad |  | Desarrollar y poner en práctica estándares, procesos, herramientas y métodos de evaluación para el aseguramiento de calidad para los diferentes procesos incluidos en la gestión de la configuración del software |  |
| Auditoría |  | asegurar que la información registrada en la CMDB (Configuration Management DataBase) coincide con la configuración real de la estructura TI de la organización |  |

**Tabla 1:** Tabla de Roles y responsabilidades

* 1. **Políticas**

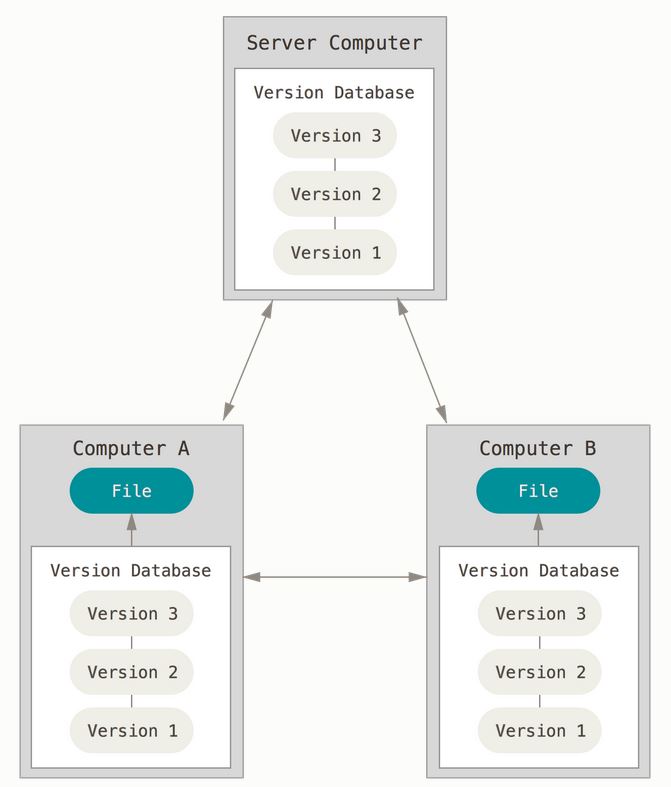
* 1. **Herramientas, entorno e Infraestructura**

**Git Hub**

Es una plataforma de **desarrollo colaborativo de software** para alojar proyectos utilizando el sistema de control de versiones **Git**.

**Git**

Es un sistema de control de versiones, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

El git trabaja de una forma distribuida para que cada usuario pueda tener actualizado los archivos que contiene el repositorio en un repositorio local, de forma que si por algún motivo el servidor central tiene una falla, el contenido del repositorio esta copiado en cada usuario conectado.

Podemos observar como es que se va actualizando las versiones de los archivos, Si un archivo es modificado, el nuevo contenido cambia de versión.

