Módulos de Organización y Generalidades

Plan de Gestión de la Configuración

Versión 1.1

Andres Fernando López Avila

Juan Esteban Moreno Rodríguez

Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 02/Septiembre/2014 | 1.0 | Se da inicio con la plantilla y su organización. | Andres Fernando López Avila. |
| 03/Septiembre/2014 | 1.1 | Se da inicio a la realización de los primeros ítems de la plantilla | Andres Fernando López Avila. |
| 04/Septiembre/2014 | 1.2 | Se da inicio a llenar el punto 2 de la plantilla | Juan Esteban Moreno Rodriguez |
|  |  |  |  |

Tabla de contenidos

1. Introducción 3

1.1 Propósito 3

1.2 Alcance 3

1.3 Definiciones, acrónimos y abreviaturas 3

1.4 Referiencia 3

2. Gestión de la Configuración del Software

2.1 Organización, responsabilidades e Interfaces 3

2.2 Desarrollo, ambiente e infraestructura 3

3. Programa de Gestión de la Configuración 3

3.1 Identificación de la Configuración 3

3.1.1 Identificación de métodos 3

3.1.2 Línea base del proyecto 3

3.2 Configuración y control de cambios 3

3.2.1 Procedimientos y Aprobaciones para las Solicitudes de Cambios 3

3.2.2 Junta de control de cambios (CCB) 3

3.3 Estimación del Estado de la Configuración

3.3.1 Medios de Almacenamientos del Proyecto y Procesos de Distribución 3

3.3.2 Reportes y aditorias 3

4. Hitos 3

5. Formación y Recursos 3

6. Subcontratista y proveedor de software de control 3

Plan de Gestión de la Configuración

# Introducción

## Propósito

En este documento Plan de Gestión de la Configuración (SCM) se describirán las actividades de gestión de configuración de software que deben ser llevadas a cabo durante el proceso de desarrollo del proyecto, nos proporciona una visión general de la organización, actividades, tareas y los objetivos de gestión de la configuración, es una actividad de garantía de calidad que se aplica en todas las fases del proceso del software.

## Alcance

Este Plan de Gestión de la Configuración involucrara todas las fases del proyecto y estará basado en algunos lineamientos que se especificaran:

* El tiempo de realización del proyecto está restringido a 16 semanas, por lo que se necesitaran respuestas rápidas a los cambios.
* El desarrollo del proceso se realizara de manera incremental por lo que se debe llevar un control sobre cada una de las entregas y de los cambios sugeridos, evaluados y aprobados.
* Se elegirán los elementos de configuración en base a las entregas realizadas.

## Definiciones, acrónimos y abreviaturas

* **SCM** (Software Configuration Management) Gestión de Configuración del Software.
* **Línea Base**: Es documento o producto oficial aprobado que sirve como punto de partida para futuras versiones.
* **CI**: Item de configuración.

## Referencias

[1] ANSI/IEEE Std 828-1990, IEEE Standard for Software Configuration Management Plans.

[2] CMMI para desarrollo, Versión 1.3 Noviembre de 2010 – [www.sei.cmu.edu](http://www.sei.cmu.edu).

# Gestión de la Configuración del Software

## Organización, responsabilidades e Interfaces

|  |  |
| --- | --- |
| Responsable | Actividad |
| Andres Fernando López Avila | - Revisar todas las peticiones de cambio.  - Asignar las peticiones de cambio aprobadas una fecha de implementación.  - Controlar y facilitar los cambios a las características de un CI.  - Administrar la base de datos de administración de configuración. |
| Juan Esteban Moreno Rodríguez | - Asegurarse de que la acción se toma en las solicitudes de cambio de manera oportuna.  - Mantener el Plan de Gestión de la Configuración.  - Realizar auditorías para verificar el cumplimiento del Plan. |

## Desarrollo, ambiente e infraestructura

Los sistemas de control de versiones (SVC – System Version Control) son herramientas que facilitan la administración de distintas versiones de cada producto que se esta desarrollando, brindan la posibilidad de hacer cambios sobre los elementos guardados y al mismo tiempo permite devolver los cambios o guardar la nueva versión de cada uno de estos elementos para no perder el control del proyecto a desarrollar.

Para realizar el proyecto del modulo administrativo se ha elegido como SVC el sistema de control de versiones GIT, son algunas características de esta herramienta las siguientes:

* Manejo de ramas en el repositorio para dar la opción de mezclarlas entre si cuando sea necesario
* Cada una de las nuevas ramas que se van formando y los nuevos cambios van quedando guardados en la nube, por lo que no ocupa mucho espacio en el disco duro.
* Libre y de código abierto

El repositorio del proyecto del modulo administrativo se puede visitar por medio del siguiente enlace:

* <https://github.com/andresflopez13/Proyecto-Software-III>

|  |  |  |
| --- | --- | --- |
| **Nombre de Herramienta** | **Tipo** | **Descripción** |
| Git | CVS | Git es un sistema de control de versiones distribuido de código abierto. |
| Github | Repositorio Remoto | Es la herramienta que se utiliza para alojar los proyectos utilizando el sistema de control de versiones GIT |
| Mantis | Bug Tracker | Es el sistema que permite realizar las peticiones de cambios al proyecto, también permite notificar los problemas hallados. |
| Netbeans | Entorno de desarrollo | Permite crear los proyectos con JAVA primordialmente pero también se puede instalar varios modulos, para el caso de este proyecto se instala el modulo de PHP que es el lenguaje de programación que se va a utilizar. |
| YII | Framework | YII es el framework eligido para desarrollar el proyecto, es orientado a objetos, lo que ayuda a mejorar la curva de aprendizaje. |
| MySql | SGBD | Es un SGBD multihilo y multiusuario, utilizado para almacenar datos y comunicar modelos. |
| Apache | Servidor Web | Es un servidor web de código abierto usado para desarrollar de forma optima el proyecto. |

# Programa de Gestión de la Configuración

## Identificación de la Configuración

### Identificación de métodos

[Describe how project or product artifacts are to be named, marked, and numbered. The identification scheme needs to cover hardware, system software, Commercial-Off-The-Shelf (COTS) products, and all application development artifacts listed in the product directory structure; for example, plans, models, components, test software, results and data, executables, and so on.]

### Línea base del proyecto

[Baselines provide an official standard on which subsequent work is based and to which only authorized changes are made.

Describe at what points during the project or product lifecycle the baselines are to be established. The most common baselines would be at the end of each of the Inception, Elaboration, Construction, and Transition phases. Baselines could also be generated at the end of iterations within the various phases or even more frequently.

Describe who authorizes a baseline and what goes into it.]

## Configuración y control de cambios

### Procedimientos y Aprobaciones para las Solicitudes de Cambios

[Describe the process by which problems and changes are submitted, reviewed, and dispositioned.]

### Junta de control de cambios (CCB)

[Describe the CCB membership and the procedures for processing change requests and approvals to be followed by the CCB.]

## Estimación del Estado de la Configuración

### Medios de Almacenamientos del Proyecto y Procesos de Distribución

[Describe retention policies, and the back-up, disaster, and recovery plans. Also describe how the media is to be retained—online, offline, media type, and format.

The release process describes what is in the release, who it is for, and whether there are any known problems and any installation instructions.]

### Reportes y auditorias

[Describe the content, format, and purpose of the requested reports and configuration audits.

Reports are used to assess the “quality of the product” at any given time in the project or product lifecycle. Reporting on defects based on change requests may provide some useful quality indicators and, thereby, alert management and developers to particularly critical areas of development. Defects are often classified by criticality (high, medium, and low) and could be reported on the following basis:

* Aging (Time-based Reports): How long have defects of the various kinds been open? What is the “lag time’’ between when defects are found in the lifecycle and when they are fixed?
* Distribution (Count Based Reports): How many defects are there in the various categories by owner, priority or state of fix?
* Trend (Time-related and Count-related Reports): What is the cumulative number of defects found and fixed over time? What is the rate of defect discovery and fix? What is the “quality gap” in terms of open as opposed to closed defects? What is the average defect resolution time?]

# Hitos

[Identify the internal and customer milestones related to the project or product CM effort. This section includes details on when the **Configuration Management Plan** itself is to be updated.]

# Formación y Recursos

A continuación se describe el personal y la formación que se debe de tener para usar las herramientas anteriormente mencionadas y así cumplir con las actividades propuestas.

**Github:** Para manejar esta herramienta es supremamente necesario que cada una de las personas que la utilicen sepan cómo subir, bajar, registrar y combinar las diferentes versiones, también que sepan utilizar la interfaz gráfica pero al mismo tiempo que utilicen la línea de comandos.

**Mantis:** También es necesario que esta herramienta pueda ser usada por cualquier miembro del equipo de trabajo para corregir los errores e implementar las modificaciones en el menor tiempo posible y por la persona o personas que sean las más adecuadas para trabajar en ello.

**Netbeans:** Al igual que las herramientas mencionadas anteriormente, se necesita que todo el equipo de trabajo conozca muy bien esta herramienta, para instalar los diferentes componentes que hagan falta y para que se agilice la elaboración del código fuente, además de esto también es necesario conocimiento en el lenguaje de programación PHP ya que es el lenguaje usado en la elaboración de todo el proyecto.

**YII:** YII es un framework usado en PHP, por lo tanto se necesita que por lo menos una persona sepa usarlo, conozca sus características para no causar conflictos y para aprovechar al máximo sus características.

**MySql:** Es muy importante saber utilizar esta herramienta para poder realizar cada una de las consultas y para poder crear un buen modelo.

**Apache:** Es indispensable que usemos este servidor para el proyecto, ya que se necesita para poder ver el proyecto como una web, así que por este motivo debe de haber por lo menos una persona que conozca sus servicios y como funciona para que no hayan problemas.

# Subcontratista y proveedor de software de control

A continuación se especifica como se incorporan al proyecto los diferentes softwares desarrollados fuera del entorno del proyecto:

[Describe how software developed outside of the project environment will be incorporated.]