**Plan de Gestión de**

**Configuración**

**V 1.0**

***Fecha: 08/09/2017***

ÍNDICE

[1. Introducción 4](#_Toc493843870)

[1.1. Propósito 4](#_Toc493843871)

[1.3. Objetivos 4](#_Toc493843872)

[1.4. Abreviaciones 5](#_Toc493843873)

[1.5. Resumen Ejecutivo 5](#_Toc493843874)

[2. Gestión de la SCM 6](#_Toc493843875)

[2.1. Organización 6](#_Toc493843876)

[2.2. Roles y responsabilidades 6](#_Toc493843877)

[2.3. Políticas y directrices y procedimientos 7](#_Toc493843878)

[2.4. Herramientas, entorno e infraestructura 9](#_Toc493843879)

[2.4.1. Herramientas de control de versiones 9](#_Toc493843880)

[2.4.2. Herramientas de entorno 10](#_Toc493843881)

**Lista de Tablas**

[Tabla 1 : Relación de Actividades con los roles de proyecto 6](#_Toc493843859)

[Tabla 2: Calendario del Plan de Gestión de la Configuración 10](#_Toc493843860)

# **Introducción**

Consulting Asociados es una empresa peruana que, desde su creación desarrolla software a la medida con personal especializado que reside en Perú, nuestra empresa fue fundada en el año 2014, nuestra propuesta de propuesta de valor se basa en la entrega de soluciones software personalizadas, capaces de soportar procesos clave y actividades de misión crítica, alineados con los objetivos estratégicos de nuestros clientes.

En el desarrollo de software, los cambios debidos principalmente a modificaciones de requisitos y fallos son inevitables. Normalmente se trabaja en equipo por lo que es preciso llevar un control y registro de los cambios con el fin de reducir errores, aumentar la calidad y la productividad y evitar los problemas que puede acarrear una incorrecta sincronización en dichos cambios, al afectar a otros elementos del sistema o a las tareas realizadas por otros miembros del equipo de proyecto.

El objetivo de la gestión de la configuración es mantener la integridad de los productos que se obtienen a lo largo del desarrollo de los sistemas de información, garantizando que no se realizan cambios incontrolados y que todos los participantes en el desarrollo del sistema disponen de la versión adecuada de los productos que manejan. Así, entre los elementos de configuración software, se encuentran no únicamente ejecutables y código fuente, sino también los modelos de datos, modelos de procesos, especificaciones de requisitos, pruebas, etc.

La gestión de configuración se realiza durante todas las actividades asociadas al desarrollo del sistema, y continúa registrando los cambios hasta que éste deja de utilizarse.

## **Problemática**

**\*** La primera dolencia que sufre la empresa indica es que no tiene sus repositorios con sus documentación, la cual hace que los desarrolladores no puedan consultar a la documentación los requerimientos del proyecto u otras especificaciones

\* También al momento de actualizar alguna funcionalidad nueva integrada en el sistema, no se esta actualizando continuamente estos requerimientos

\* La empresa al desarrollar muchos proyectos en simultaneo, se le es difícil lleva una adecuada gestión de los proyectos

\* Los developers que ya no están laborando en la empresa, todavía tiene los permisos en los repositorios

* 1. **Propósito**

\* Para las problemáticas presentadas, se va a llevar un plan para el gestionamiento de cambios en el cual se actualizaran los cambios, en periodos de tiempo o según se vayan aumentando las funcionalidad del proyecto que se esta realizando

\* Se actualizaran los permisos para personas que ingresen al proyecto o personan que ya no están laborando en la empresa, para así poder evitar algunos problemas éticos en la empresa

## **Aplicación**

\* De esta manera vamos a lograr optimizar los recursos por parte de lado de informática.

\* Optimizaremos el retraso del proyecto, que puede surgir al no dejar algunas funcionalidades poco claras

## **Abreviaciones**

* SCMR: Software Configuration Management Responsible, Gestor de configuración del software.
* CCB: Configuration Contorl Board, Comité del Control de la Configuración.
* PL: Program Librarian, Bibliotecario
* soluciones software personalizadas: productos desarrollados específicamente para mejorar procesos del negocio del cliente.
* Mantener Integridad de los productos: con esto se quiere decir que se asegura que los cambios realizados en un sistema no afectaran negativamente a los otros sistemas
* sistemas de información: conjunto de elementos que interactúan entre si con el fin de apoyar las actividades de la empresa o negocio
* Ejecutable: archivo que tiene la capacidad de ser ejecutado independiente mente.
* Código fuente: archivo con las las instrucciones necesarias, realizadas en un lenguaje de programación, para ejecutar el programa.
* Modelo de procesos: expresión abstracta de los procesos principales de la empresa o negocio.
* Especificación de requisitos: son todos los requerimientos detallados que el sistema debe cumplir.

# **Gestión de la SCM**

## **Organización**

Para cumplir con las expectativas de nuestra distinguida cliente nuestra empresa como tal, cuenta con un esquema organizacional, donde cada integrante de nuestro equipo conoce sus obligaciones en nuestra organización

|  |  |
| --- | --- |
| **Actividad** | **Responsable** |
| Planificar la Configuración del SCM | SCMR |
| Definición de la línea base del proyecto | SCMR, CCB, PL |
| Seguimiento del proyecto de la línea base | SCMR |
| Definición del ambiente | SCMR |
| Control de cambios | SCMR, CCB |
| Descripción de la versión | SCMR |
| Realizar informe final del SCM | SCMR |

Tabla 1 : Relación de Actividades con los roles de proyecto

Los integrantes usan google docs para estar vinculados a las actividades del SCM y posteriormente subidas al repositorio de github.

## **Roles y responsabilidades**

* **SCMR**
  + Debe proveer la infraestructura y el entorno de configuración para el proyecto. Debe preocuparse porque todos los integrantes del grupo entiendan y puedan ejecutar las actividades de SCM que el Plan les asigna, así como asegurar que éstas sean llevadas a cabo. Seguir la línea base, controlando las versiones y cambios de ella, son tareas correspondientes a él. Debe definir y construir el Ambiente Controlado e informar al resto del equipo sobre la manera de usarlo.
  + Personal necesario: 1
* **Bibliotecario (*Program Librarian*, PL)**
  + Encargado de respaldar con las tareas de almacenamiento de los *baselines*. Un *baseline* o línea base es uno o más documentos formalmente diseñados y corregidos en un tiempo específico del ciclo de vida de los ítems de configuración, estos a su vez, son el conjunto de versiones de un item aceptado por el cliente.
  + Personal necesario: 1
* **Comité de Control de Configuración (Configuration Control Board, CCB)**,
  + Máxima autoridad en la autorización de cambios.
  + Tiene la autoridad para aceptar o rechazar las propuestas de cambio a componentes de configuración. Cómo estos cambios tienen sentido controlarlos una vez que se crean las líneas base, el comité de control de cambios tiene la autoridad para gestionar las líneas base del producto y asegurar que los cambios son adecuadamente considerados y coordinados.
  + Personal necesario: 2

## **Políticas y directrices y procedimientos**

Respecto a los documentos en el repositorio de github, todos tienen derechos de administrador, osea, permiso de lectura, escritura y creación de nuevos documentos. Se tiene como política la total confianza en todos los integrantes del equipo.

**Políticas.-**

* Seguir los estándares de nomenclatura plasmados en el presente documento.
* Cada rol debe cumplir las actividades designadas en el documento.
* Cada miembro debe trabajar únicamente y exclusivamente sobre su branch.
* La filtración de información a agentes externos se encuentra extrictamente prohibida.
* Correcta comunicación entre los miembros encargados de la gestión de la configuración.
* Toda dependencia en los proyectos deberá ser estrictamente documentada.

**Directrices.-**

* Se encuentra estrictamente prohibido el versionamiento en el nombre del item. Ejm: No se permite el nombre “SCAE\_AC\_v1.1.docx”
* Para los items de gestión de la configuración:
  + Los items deberán llevar el siguiente estándar de nombre:

<INICIALES DEL DOCUMENTO>.\*

Ejm: PGC.docx (Plan de gestión de la configuración)

* En caso dos items de gestión de la configuración llegasen a tener la misma nomenclatura, el ítem más reciente tendrá la nomenclatura:
  + <INICIALES DEL DOCUMENTO>\_<TAG>.\*

Donde <TAG> es un identificador para dar soporte a la identificación del documento más reciente.

* Para los items de gestión de proyectos:
  + Los items deberán llevar el siguiente estándar de nombre:

<INICIALES DEL PROYECTO>\_<INICIALES DEL DOCUMENTO>.\*

Ejm: SCAE\_AC.docx

* Los nombres de los branch deberá ser hecho en snake\_case de los nombres de los responsables en minúsculas. Ejm: Josafat Vara -> branch “josafat\_vara”
* Para los items de desarrollo de proyectos FrontEnd:
  + Todo folder o item que cuente con más de 2 palabras, deberá ser nombrado en Kebab-case. Ejm: dashboard home -> dashboard-home
  + Los componentes:
    - Los items que expongan los componentes deberán ser nombrados de la siguiente manera:

<NOMBRE DEL COMPONENT O FEATURE>.component.js

* + Las vistas o templates:
    - Deberán ir en el folder de su respectivo componente o feature (característica de la aplicación).
    - Deberán ser nombradas de la siguiente manera:

<NOMBRE DEL COMPONENT O FEATURE>.template.html

* + Los controladores
    - Deberán ir en el folder de su respectivo componente o feature (característica de la aplicación).
    - Deberán ser nombradas de la siguiente manera:

<NOMBRE DE COMPONENT O FEATURE>.controller.js

* + Los services(servicios)
    - Deberán ir en el folder src/app/core del proyecto.
    - Dependiendo de la utilidad del servicio, este deberá ir en la siguiente carpetas:
      * Utils.- servicios que utilizan componentes.
      * Helpers.- servicios para evitar la repetición de código o snippets.
      * Services.- servicios con lógica de procesos.
    - Los items que expongan los servicios, deberán ser nombrados de la siguiente manera:

<NOMBRE DEL SERVICIO>.service.js

* + Los resources(recursos de datos)
    - Deberán ir en el folder src/app/core/resources del proyecto.
    - Dependiendo de su fuente, deberán ir en las carpetas:
      * Local.- Extracción de datos almacenados en el navegador.
      * Test.- Para mock.
      * Remote.- Extracción de datos de fuentes remotas(APIs).
  + Los estilos de la aplicación
    - Deberán ir en src/assets.
    - Se deberá seguir el patrón de diseño 7-1. Ref: <https://sass-guidelin.es/es/#el-patron-7-1>
  + Imágenes y documentos
    - Deberán ir en las carpetas src/images y src/docs respectivamente.
* Para los items de proyectos BackEnd:
  + Todo item con más de dos palabras deberá ser nombrado en Kebab-case.

Ejm: logging middleware -> logging-middleware

* + Será usado como estructura del proyecto el generado por el Framework loopback.

## **Herramientas, entorno e infraestructura**

### **Herramientas de control de versiones**

* Github

Un hosting online para nuestros repositorios que utiliza git para el mantenimiento y versionado del código fuente, añadiendo una serie de servicios extras para la gestión del proyecto y el código fuente. La parte gratuita de este hosting permite alojar nuestro código en repositorios públicos, si queremos repositorios privados entramos a la parte “premium”. En la actualidad ofrecen varias herramientas útiles para el trabajo en equipo, entre ellos cabe destacar

* Una wiki: para el mantenimiento de distintas versiones de las páginas.
* Un sistema de seguimiento de problemas, que permite a un miembro de tu equipo detallar el problema con tu software.
* Herramienta de versión de código, donde puedes añadir anotaciones en cualquier punto de un fichero.
* Un visor de ramas: donde se puede comparar los progresos realizados en las distintas ramas de nuestro repositorio.
* Git CLI

Git es un sistema rápido, escalable, distribuido de control de versiones con un número muy grande de comandos que proporciona las operacione de alto nivel y acceso completo al nivel inferior. La herramienta a usar para el caso de proyectos Front-end y Back-end será Git CLI(Command Line).

### **Herramientas de entorno**

AÑADIR Q HERRAMIENTAS SE USARÁ

**2.5 Calendario**

Las diferentes actividades que se tomara en cuenta durante todo el proceso de gestión de la configuración y mantenimiento del software (Ver Tabla 2.0).

Tabla 2: Calendario del Plan de Gestión de la Configuración

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Depende de** | **Comienzo** |
| Realizar el plan de gestión de la configuración | Nada | Semana 1: 01/09/2017 |
| Realizar la actividad de identificación de SCM | Plan de gestión de la configuración | Semana 3: 15/09/2017 |
| Realizar la definición de la línea base y estructura de librerías | Plan de gestión de la configuración | Semana 7: 13/10/2017 |
| Realizar el plan de gestión de cambios | Plan de gestión de la configuración | Semana 8: 20/10/2017 |
| Realizar los reportes de estado | Plan de gestión de la configuración y plan de gestión de cambios | Semana 9: 27/10/2017 |
| Realizar los reportes de auditorias | Plan de gestión de la configuración y plan de gestión de cambios | Semana 10: 03/10/2017 |
| Realizar la gestión de release | Plan de gestión de la configuración | Semana 11: 10/10/2017 |