***“SCAE (SIstema de Control de Asistencia de Estudiantes)”***

**Plan de Gestión de**

**Configuración**

***Fecha: 29/04/2017***

ÍNDICE

[1. Introducción 3](#_Toc18546)

[1.1. Propósito 3](#_Toc30303)

[1.2. Alcance 3](#_Toc25340)

[1.3. Abreviaciones 3](#_Toc2918)

[1.4. Resumen Ejecutivo 4](#_Toc4890)

[2. Gestión de la SCM 5](#_Toc22039)

[2.1. Organización 5](#_Toc10245)

[2.2. Roles y responsabilidades 7](#_Toc9720)

[2.3. Políticas y directrices y procedimientos 8](#_Toc17305)

# **Introducción**

## **Propósito**

El propósito de este documento es describir 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 presente plan de configuración está realizando teniendo presente lo siguiente:

* El proyecto dura un aproximado de 12 semanas.
* El modelo de proceso de desarrollo de software a seguir es SCRUM
* Los proyectos tipo Front-end son desarrollados usando el Framework Angular 1.x.
* Los proyectos tipo Móvil son desarrollados en Java, usando el IDE Android Studio.
* Los proyectos tipo Back-end son desarrollados en Node.js, usando el framework Loopback.

## **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

## **Resumen Ejecutivo**

Los siguientes puntos del presente documento van a presentar la siguiente estructura:

* Gestión del SCM: se muestra las responsabilidades y responsables del proyecto
* Actividades del SCM: se identifican las actividades y tareas que se requieren para la configuración del sistema
* Control:Configuración y control de cambios en la configuración del proyecto
* Estado:Registro y seguimiento de las actividades del SCM
* Auditoría: Como nos aseguramos de que los cambios efectuados se han hecho correctamente.
* Gestión de Release: se detalla cómo se hará la entrega final del proyecto al cliente.

# **Gestión de la SCM**

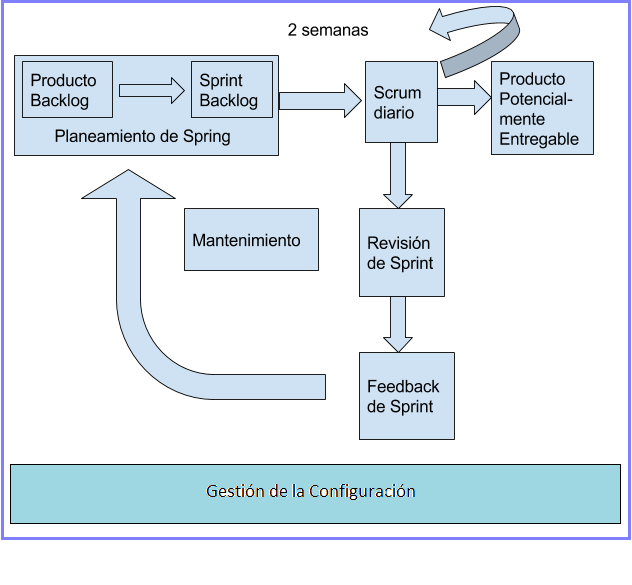
## **Organización**

SP11 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 2015, nuestra propuesta de propuesta de valor se basa en la entrega de soluciones software personalizadas, capaces de soportara procesos clave y actividades de misión crítica, alineados con los objetivos estratégicos de nuestros clientes.

Para lograrlo utilizamos herramientas y metodologías de desarrollo, y de prueba que nos permiten entregar soluciones flexibles, escalables, con una gran facilidad de uso, multiplataforma, en línea y en tiempo real cuando sea requerido, fáciles de integrar y con los máximos niveles de seguridad.

Nuestro objetivo es que nuestros clientes nos consideren un socio estratégico a largo plazo, en todo lo que se refiera de entrega de soluciones tecnológicas que resuelven sus problemas de negocio.

Para cumplir con esta espectativas nuestra empresa como tal, cuenta con un esquema organizacional, donde cada integrante de nuestro equipo conoce sus obligaciones en nuestra organización, el siguiente esquema muestra la forma como está distribuido, manteniendo por supuesto una jerarquía como cualquier otra empresa:



*Relación del modelo de proceso usando, respecto a la gestión de configuraicón y mantenimiento*

La siguiente tabla muestra cómo se relacionan las actividades del proyecto con los roles en el proyecto.

|  |  |
| --- | --- |
| **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 |

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 el. 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. Como 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 estricamente 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)

* 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](https://sass-guidelin.es/es/" \l "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.
* Para los items de proyectos Móvil:
  + ...

**Procedimientos.-**

* Respecto a la modificación de archivos pertenecientes a la gestión de la configuración.

1. Indicar a demás miembros encargados de los documentos qué documento está siendo manipulado, con el fin de evitar modificaciones en el mismo por parte de otro miembros, lo cual repercutiría en un sobreescribimiento del item.
2. Modificar el item apropiadamente.
3. Subir los cambios.
4. Hacer los merge necesarios para que el item modificado se encuentre disponible para todos.
5. Informar que el cambio fue realizado.