**Plan de Gestión de la Configuración**

**Versión 1.0**

te documento es la plantilla base para elaborar el documento Plan de SQA. Los textos que aparecen entre paréntesis rectos son explicaciones de que debe contener cada sección. Dichos textos se deben seleccionar y sustituir por el contenido que corresponda.]

**Historia de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autores |
| 20/09/2019 | 1.0 | Creación del Plan | Chinchayan, Erick |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contenido

[1. INTRODUCCIÓN 3](#_Toc22589061)

[1.1. SITUACIÓN ACTUAL DE LA EMPRESA 3](#_Toc22589062)

[1.2. PROBLEMÁTICA 3](#_Toc22589063)

[1.3. PROPÓSITO DE PLAN 3](#_Toc22589064)

[1.4. APLICABILIDAD 4](#_Toc22589065)

[1.5. GOBIERNO Y ALCANCE 4](#_Toc22589066)

[2. GESTIÓN DE LA SCM 4](#_Toc22589067)

[2.1. ROLES O RESPONSABILIDADES 4](#_Toc22589068)

[2.2. POLITICAS, DIRECTRICES Y PROCEDIMIENTOS 5](#_Toc22589069)

[2.3. HERRAMIENTAS, ENTORNO E INFRAESTRUCTURA 5](#_Toc22589070)

[2.4. CALENDARIO 5](#_Toc22589071)

[3. Actividades de la SCM: 6](#_Toc22589072)

[3.1. IDENTIFICACIÓN DE LA CONFIGURACIÓN, NOMENCLATURA 6](#_Toc22589073)

[3.1.1. DEFINIR NOMENCLATURA (FÓRMULA) 6](#_Toc22589074)

[3.1.2. REALIZAR LA CLASIFICACIÓN DE ÍTEMS Y  APLICAR NOMENCLATURA (INVENTARIO) 7](#_Toc22589075)

[3.2. CONTROL DE LA CONFIGURACIÓN 7](#_Toc22589076)

[3.3. ESTADO DE LA CONFIGURACIÓN 7](#_Toc22589077)

[3.4. AUDITORIA DE LA CONFIGURACIÓN 7](#_Toc22589078)

[3.5. GESTIÓN Y ENTREGA DE RELEASE DE SOFTWARE 7](#_Toc22589079)

# INTRODUCCIÓN

## SITUACIÓN ACTUAL DE LA EMPRESA

La empresa Digital Factory es una consultora de Software para entidades públicas y privadas. Se encuentra a la vanguardia de la tecnología y con amplio conocimientos en diversos lenguajes de programación como: Java, Python, Javascript, PHP, .Net, Kotlin, etc.; contando con profesionales capacitados para la implementación de sistemas innovadores para la implementación de soluciones tecnológicas.

## PROBLEMÁTICA

Actualmente la empresa no cuenta con un repositorio central para los proyectos que está gestionando. Los códigos fuentes se encuentran en diversos repositorios privados a criterio del personal asignado a un proyecto, muchos de los cuáles no cuenta con la última versión ya que no se hizo la sincronización adecuada.

Para los softwares que están en mantenimiento es complicado determinar la release en despliegue de un determinado cliente.

La documentación de gestión de la empresa se encuentra en una carpeta central compartida con varios archivos del mismo documento para poder conservar las versiones anteriores, muchos de los cuales fueron sobre escritos, creando conflictos y pérdida de información.

## PROPÓSITO DE PLAN

En el proceso de 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 traer consigo 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.

El propósito de este plan es plantear lineamientos y estándares para una correcta gestión de la configuración de los proyectos de software.

## APLICABILIDAD

El plan de gestión de configuración debe aplicarse para cualquier proyecto desarrollado por la empresa Digital Factory.

## GOBIERNO Y ALCANCE

El gerente de proyecto es responsable de asegurar que el plan de gestión de la configuración sea desarrollado en conjunto con el plan de gestión del proyecto. El gerente de proyecto asegurará su integración en la planificación general.

# GESTIÓN DE LA SCM

## ROLES O RESPONSABILIDADES

Para el desarrollo del presente plan se contará con la participación de un Gestor de la Configuración, Bibliotecario y Comité de control de cambios.

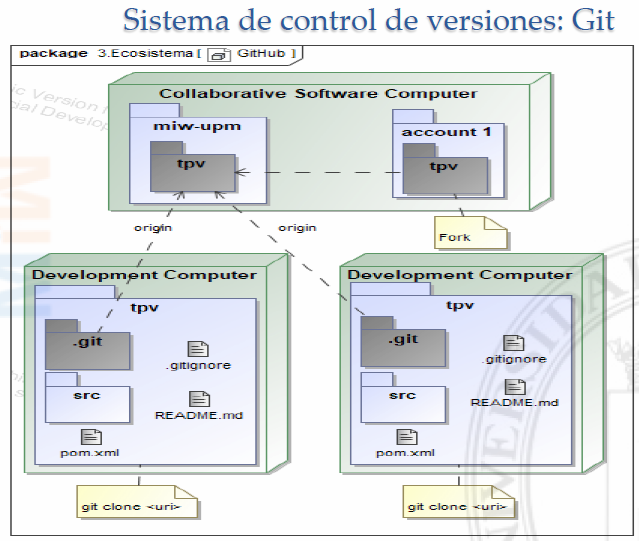
|  |  |  |  |
| --- | --- | --- | --- |
| **ROL** | **PERSONA ASIGNADA** | **RESPONSABILIDADES** | **NIVELES DE AUDITORIA** |
| Gestor de la Configuración | Erick Chinchayan | * Gestionar la planificación, identificación, control, seguimiento y auditoría de todos los elementos de configuración en la base de datos de configuración. * Desarrollar el plan de gestión de configuración. * Monitorizar y reportar los cambios no autorizados sobre los elementos de configuración. * Liderar las actividades de evaluación del proceso: revisar tipos de elementos de configuración, relaciones, atributos y valores asociados, derechos de acceso. | Autoridad para operar las funciones de Gestión de la Configuración |
| Bibliotecario | Mauro Flores | * Asegurar que los elementos de configuración de los que es responsable están registrados en la base de datos de configuración con el estado y datos de configuración apropiados. * Verificar que los cambios sobre los elementos de configuración siguen el proceso de cambios definido. | Audita la Gestión de la Configuración según indique el Project Manager. |
| Comité de control de cambios | Ivan Garcilazo  Miguel Vera | * Evaluar el impacto y riesgo de los cambios. * Asegurar que los responsables de los elementos de configuración actualizan los históricos de estos elementos con los cambios implementados | Depende de cada miembro, se especifica para cada artefacto y cada ítem. |

## POLITICAS, DIRECTRICES Y PROCEDIMIENTOS

* Se usará Git como herramientas de control de versiones.
* Todos los ítem de configuración serán almacenados en la herramienta de control de versiones seleccionada.
* Como repositorio remoto se usará GitHub.
* Se usará flujo de trabajo ramificado
  + Cada integrante crea una rama *su-apellido* desde *master*
  + Se hace los cambios en la rama *su-apellido*
  + Cuando se completa un *su-apellido*, se fusiona con la rama *master*.
* Para la nomenclatura de los archivos de código fuente se usará la guía de estilos de la comunidad oficial del lenguaje de programación seleccionado.
* Todos los miembros del equipo del tendrán que alinearse a éstas políticas.
* No se deberán almacenar archivos comprimidos (archivos .zip, .rar, .ace, etc.).
* Los nombres de los archivos que se carguen al repositorio solo deberán estar conformados por letras y números, es decir, no deberán tener tildes, letras ñ, ni ningún otro carácter especial.

## HERRAMIENTAS, ENTORNO E INFRAESTRUCTURA

* Se usará Git como herramientas de control de versiones.
* Se usará GitHub como repositorio remoto
* Se usará Kotlin como lenguaje de programación para android
* Se usará Android Studio como IDE para Kotlin
* Se usará Gradle como gestor de dependencias para Kotlin
* Se usará MySQL como sistema gestor de base de datos
* Se usará MySQL Workbench como herramienta para administración de base de datos.
* Se usará Angular como framework de desarrollo para el frontend
* Se usará NodeJS + Express para desarrollo para el backend
* Se usará JavaScript como lenguaje de programación del backend y frontend.
* Se usará WebStorm como IDE para angular y NodeJS
* Se usará npm como gestor de dependencias para backend y fronted



El diagrama describe el ecosistema de trabajo con GitHub que consiste en un repositorio remoto del cuál cada uno de los integrantes tiene en su computadora un clon con la misma información. Éstos clones pueden ser modificados y luego se sincronizan. La ventaja de tener clones distribuidos del proyecto disminuye la posibilidad de pérdida del proyecto, además que permite trabajar sin necesidad de conexión a internet y llevando un control eficiente de las versiones de los ítems de la configuración.

## CALENDARIO

|  |  |  |
| --- | --- | --- |
| **ACTIVIDAD** | **TIEMPO EN DIAS** | **ROL** |
| 1. **Planificación** | **5** | **Gestor de la Configuración** |
| * 1. Describir la situación actual de la empresa | 1 | Gestor de la Configuración |
| * 1. Plantear la problemática | 1 | Gestor de la Configuración |
| * 1. Identificar el propósito del plan | 1 | Gestor de la Configuración |
| * 1. Describir la aplicabilidad | 1 | Gestor de la Configuración |
| * 1. Describir el gobierno y alcance | 1 | Gestor de la Configuración |
| 1. **Identificación** | **6** | **Gestor de la Configuración** |
| * 1. Identificar roles y responsabilidad | 1 | Gestor de la Configuración |
| * 1. Plantear políticas, directrices y procedimientos | 1 | Gestor de la Configuración |
| * 1. Describir herramientas, entorno e infraestructura | 1 | Gestor de la Configuración |
| * 1. Clasificar los ítems de configuración | 1 | Gestor de la Configuración |
| * 1. Identificar los ítems de configuración | 1 | Gestor de la Configuración |
| * 1. Definir nomenclatura | 1 | Gestor de la Configuración |
| 1. **Control** | **3** | **Comité de control de cambios** |
| * 1. Definir Línea Base | 2 | Comité de control de cambios |
| * 1. Definir la estructura de la biblioteca | 1 | Comité de control de cambios |
| 1. **Estado** |  | **Bibliotecario** |
| 1. **Auditoría** |  | **Comité de control de cambios** |
| 1. **Entrega y gestión de release** |  | **Gestor de la Configuración** |

# ACTIVIDADES DE LA SCM

## IDENTIFICACIÓN DE LA CONFIGURACIÓN, NOMENCLATURA

### CLASIFICACIÓN DE LOS ÍTEMS DE LA CONFIGURACIÓN

La clasificación de ítem de la configuración son los siguientes:

* **Ítems en evolución,** tales como documentos, los que están sujetos a una o más revisiones y nuevas liberaciones durante el ciclo de vida del software. Los Ítems en evolución son de dos tipos: Documentos, y archivos ejecutables o de soporte.
* **Ítems fuente,** generalmente código fuente y archivos objeto utilizados para compilar una aplicación de software para ambiente de producción, los cuales son generalmente numerosos y cambian frecuentemente;
* **Ítems de soporte,** como sistemas operativos y software base, de los cuales el proyecto requiere ciertas versiones para su operación exitosa.

### IDENTIFICACIÓN DE LOS ÍTEMS DE CONFIGURACIÓN

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo**  **(E= Evolución**  **F=Fuente**  **S=Soporte)** | **Nombre del Ítem** | **Extensión** | **Proyecto** |
| E | Plan de Gestión de la configuración | docx | - |
| E | Acta de constitución del proyecto | docx | SFA |
| F | Modelo Cliente en angular | ts | SFA |
| S | Instalador de Android Studio | exe | SFA |
| S | Instalador de Git | exe | SFA |

### DEFINIR NOMENCLATURA (FÓRMULA)

A. Identificación de Ítems en evolución

* Ítems que no son específicos a un proyecto único, tales como políticas, descripciones de procesos y guías, son identificados únicamente por su acrónimo. Ejemplo: **PGPS** (Política de gestión de proyectos de software).
* Los ítems que son específicos de un PY pero no están asociados con un componente del PY utilizan un identificador de dos partes: ACRÓNIMO\_PROYECTO y ACRÓNIMO derivado del tipo de artefacto. Ejemplo, para identificar el plan de gestión de la configuración del proyecto SFA, tenemos: **SFA\_PGC**.
* Los ítems en evolución que son específicos de un PY y están asociados con un componente específico, utilizan un identificador de tres partes: ACRÓNIMO\_PROYECTO, ACRÓNIMO\_COMPONENTE, y, ACRÓNIMO derivado del tipo de artefacto. Ejemplo: para identificar el documento de especificación de requerimientos, del componente B2B (Business to Business) del proyecto SFA, tenemos: **SFA\_B2B\_DER**.
* El nivel de versión de cada ítem se mantiene como un identificador separado. Esto permite que el identificador principal sea utilizado como parte del nombre del ítem, para acceder a la versión más actualizada sin necesidad de requerir cambios a todos los ítems referenciados.
* El nivel de versión se mantiene como un identificador numérico con dos componentes: Versión.Revisión. Ejemplo: 1.1; Versión 1, Revisión 1
* El número de versión cambia únicamente cuando la arquitectura principal del ítem ha cambiado, o cuando el ítem es completamente reconstruido, con cambios internos sustanciales. En este caso la versión 1.1 se convertirá en versión 2.0.
* El número de revisión cambia cuando el contenido ha cambiado, pero la estructura principal y el flujo del ítem se mantiene igual. La secuencia normal de las revisiones es: 1.0, 1.1, 1.2, etc.

B. Identificación de ítems fuente

* Se maneja basándose en las guías de estilos del fabricante.

C. Identificación de ítems de soporte

* Son identificados por su nombre y el número de versión necesario para soportar el entorno de producción o desarrollo. Por ejemplo: Para el sistema operativo Centos 7 tendremos: **Centos v7**

Ejemplos:

|  |  |  |
| --- | --- | --- |
| **Proyecto** | **Nombre del ítem** | **Nomeclatura** |
| - | Plan de Gestión de la Configuración | PGC |
| SFA | Acta de Constitución del Proyecto | SFA\_ACP |
| SFA | Requisitos funcionales y no funcionales | SFA\_RFNF |
| SFA | Modelo Cliente en angular | customer.model.ts |
| SFA | Servicio medidor en angular | meter.service.ts |
| SFA | Controlador alquiler en PHP | RentsController.php |
| SFA | Sistema Operativo Centos 7 | Centos v7 |
| SFA | Gestor de base de datos MaríaDB 10.4 | MariaDB v10.4 |

## CONTROL DE LA CONFIGURACIÓN

### DEFINIR LÍNEA BASE

|  |  |
| --- | --- |
| **Línea Base** | **Ítems** |
| Línea Base Planificación | * Acta de constitución del proyecto * Plan de Pruebas de Calidad |
| Línea Base de Análisis | * Documento de levantamiento de información * Modelado de negocio * Requisitos funcionales y no funcionales * Especificación de casos de uso * Matriz de perfiles y accesos * Prototipos * Casos de prueba |
| Línea Base de Diseño | * Arquitectura de software * Arquitectura de tecnología de información * Diseño detallado de software |
| Línea Base de Construcción | * Código fuente backend * Código fuente frontend * Código fuente mobile |
| Línea Base de Pruebas | * Resultados de pruebas * Acta de aceptación de pruebas |
| Línea Base Cierre de Proyecto | * Acta de cierre |

### DEFINIR LA ESTRUCTURA DE LA BIBLIOTECA

Las cuatro principales bibliotecas de nuestro repositorio son:

* **Documentos:** Se encuentran los documentos de la empresa que son generales, es decir, que no pertenecen específicamente a un proyecto.
* **Línea Base:** Está distribuida por sub directorios que representan los proyectos que a su vez contienen las líneas bases de los mismo.
* **Desarrollo:** Se encuentra la ítems de configuración de los proyectos que están en desarrollo (base de datos, código fuente, documentos)
* **Clientes:** Podemos encontrar a los clientes de la empresa con las versiones de cada software que usan.

## ESTADO DE LA CONFIGURACIÓN

## AUDITORIA DE LA CONFIGURACIÓN

## GESTIÓN Y ENTREGA DE RELEASE DE SOFTWARE