

EVERSOFT

**Plan de Gestión de**

**Configuración**

**V 2.0**

***Fecha: 24/05/2018***

ÍNDICE

[**1.** **Introducción** 5](#_Toc514983022)

[**1.1.** **Propósito** 5](#_Toc514983023)

[**1.2.** **Alcance** 5](#_Toc514983024)

[**1.3.** **Aplicabilidad** 6](#_Toc514983025)

[**1.4.** **Abreviaciones** 6](#_Toc514983026)

[**2.** **Gestión de la SCM** 6](#_Toc514983027)

[**2.1.** **Organización** 6](#_Toc514983028)

[**2.2.** **Roles y responsabilidades** 7](#_Toc514983029)

[**2.3.** **Políticas y directrices y procedimientos** 8](#_Toc514983030)

[**2.4.** **Herramientas, entorno e infraestructura** 10](#_Toc514983031)

[**2.4.1.** **Herramientas de control de versiones** 10](#_Toc514983032)

[**2.4.2.** **Herramientas de entorno** 11](#_Toc514983033)

[**2.5. Cronograma de trabajo** 11](#_Toc514983034)

[**3.** **Actividades de la SCM** 12](#_Toc514983035)

[**3.1.** **Identificación:** 12](#_Toc514983036)

[**3.1.1.** **Lista de Clasificación de CI** 12](#_Toc514983037)

[**3.2.** **Control:** 15](#_Toc514983038)

**Lista de Figuras**

Figura 1. Diagrama de actividades dentro de la organización de la SCM 7

Figura 2. Estructura del trabajo de github con la herramienta Visutal Studio Community 10

**Lista de Tablas**

Tabla 1. Roles y responsabilidades de la SCM 7

# **Introducción**

## **Propósito**

Eversoft es una empresa peruana que fue fundada en el 2018, contamos con un personal especializado en desarrollo de nuevas tecnologías de la UNMSM. Desde la fundación estamos orientados a brindar servicios de soluciones usando TI a nuestros clientes capaces de soportar los diferentes modelos de negocio y requerimientos de los mismos, para que a corto plazo aumente la productividad de los procesos de nuestros clientes.

Para lograr la satisfacción de nuestros clientes, usamos las herramientas más adecuadas dependiendo de los diferentes modelos de negocio y las mejores metodologías de desarrollo, a su vez cercioramos la calidad de nuestros productos con un exhaustivo modelo de pruebas, de tal manera que nos aseguramos que nuestra marca ofrezca productos de fácil uso, escalables, flexibles, multiplataforma y seguros.

Nuestro objetivo es establecernos como una empresa firme y reconocida entre las empresas por brindar buenas soluciones y gran trabajo.

El propósito del presente documento es definir la manera en cómo se administrará el gestión de la configuración de nuestros proyectos.

## **Alcance**

El presente plan de la gestión de configuración define como se manejará todas las etapas del ciclo de vida de nuestros proyectos para poder dar una mejor administración de las versiones y revisiones que se llevaran a cabo durante el transcurso del proyecto.

Se definirá estándares y nomenclaturas de los nombres de los ítems que se usaran para el control y gestión de los entregables de documentación y software, esto para poder identificar de manera rápida y eficaz los ítems requeridos dependiendo de la necesidad que se tenga.

Se identificará todos los roles dentro de la gestión de la configuración y se establecerán las responsabilidades que correspondan a cada rol; también se definirá como se realizara el control de versiones de los diferentes documentos, y código de software, y la entrega de los Releases.

## **Aplicabilidad**

El documento presente tiene la intención de definir como se llevará a cabo la gestión de la configuración de todos los proyectos realizados por la EVERSOFT a partir, por tanto, las definiciones, procedimientos, actividades, estándares especificados en el presente documento, serán válidos y aplicables para los proyectos de software de la empresa.

## **Abreviaciones**

* **SCM**: Software Configuration Management, es el documento de la gestión de configuración del software.
* **CMS**: Configuration Mangement System, al sistema de gestión de las configuraciones encargado de soportar las relaciones entre los elementos de configuración.
* **SVR**: Sistema de Ventas de Restaurantes.
* **CI**: Configuration Ítem, elementos de configuración.
* **Cambio**: Adición, modificación, corrección o retiro de un componente de servicio que ha sido autorizado, soportado o planificado por la organización.

# **Gestión de la SCM**

## **Organización**

Para poder identificar y controlar los diferentes ítems de configuración de los diferentes proyectos, se va seguir una serie de actividades:

* Planeamiento: Antes de realizar la identificación, control y demás actividades sobre los ítems de configuración de los proyectos, se definirá el contexto organizacional, las restricciones que puedan afectar la gestión de la SCM y la naturaleza de los proyectos.
* Identificación: Se identifica los elementos que se utilizaran como por ejemplo la herramienta a usar que en este caso se usará GitHub para la gestión de la configuración de los proyectos de la empresa, a su vez también se identificará la línea base que concuerde con los requerimientos del cliente.
* Control: En esta actividad se evaluará los cambios que se realizaran a los proyectos, por ejemplo si se desea cambiar una funcionalidad, se verificará si el cambio es correcto y la manera de cómo implementarlo.
* Estado de la Contabilidad: El cambio de funcionalidad realizado se debe documentar para que otros desarrolladores puedan estar al tanto de la evolución del proyecto.
* Auditoria: Se verifica que la funcionalidad realizada este acorde con lo especificado del proyecto, también se verifica que el cambio en la funcionalidad desarrollada sea igual a lo definido en la documentación.
* Gestión y Entrega de Releases: La funcionalidad modificada se combinará con los demás elementos para crear en conjunto un ejecutable y entregarlo al cliente.

Fig. 1. Diagrama de actividades dentro de la organización de la SCM

## **Roles y responsabilidades**

Esta sección describe los roles clave que respaldan el proyecto.

| **Role** | **Responsabilidades** |
| --- | --- |
| Gerente del Proyecto | Persona que realiza la gestión diaria del proyecto y tiene responsabilidad específica por la gestión del proyecto dentro de las limitaciones aprobadas de alcance, calidad, tiempo y costo, con los requisitos, entregas y satisfacción del cliente especificada. |
| Gestor de la Configuración | Persona encargada de ejecutar las tareas de la gestión de la configuración. |
| Comité de Control de Cambios | Encargados de recibir y evaluar las solicitudes de cambios sobre el proyecto. |
| Inspector del Aseguramiento de la Calidad | Persona encargada de verificar que los entregables estén acorde con lo requerido por el cliente. |
| Bibliotecario | Define y da mantenimiento a las bibliotecas usadas durante la gestión de la configuración y concede accesos al repositorio. |
| Miembros del Equipo del Proyecto | Son aquellas personas que pueden consultar los documentos y archivos de código para realizar modificaciones, dependiendo de los niveles de autorización que tengan. |

Tabla 1. Roles y responsabilidades de la SCM

## **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 rama.
* 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\_SVR\_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.

* Los nombres de las ramas deberá tener los nombres o apellidos del responsable. Ejm: Kevin Avalos Ocaña-> Rama “kavaloso”
* Para los items de desarrollo de proyectos:
  + Vistas
    - Todos los archivos de .aspx irán dentro de la carpeta Vista
    - Todo item con más de dos palabras deberá ser nombrado con las palabras juntas en la cual la primera letra de la primera palabra en miniscula y la primera letra de la segunda en Mayuscula

Ejm: Reporte Ventas -> reporteVentas.cs

* + Los controladores
    - Deberán ir en el folder de su respectivo componente.
    - Deberán ser nombradas de la siguiente manera:

<NOMBRE DE COMPONENT O FEATURE>.controller.cs

* + Los resources(recursos de datos)
    - Deberán ir en el folder src/resources del proyecto.
    - Dependiendo de su fuente, deberán ir en las carpetas:
  + Los estilos de la aplicación
    - Deberán ir en /styles.
  + Imágenes y documentos
    - Deberán ir en las carpetas src/resources/images y src/resources/docs respectivamente.
* Para los items de proyectos BackEnd:
  + Todo item con más de dos palabras deberá ser nombrado con las palabras juntas en la cual la primera letra de la primera palabra en miniscula y la primera letra de la segunda en Mayuscula.

Ejm: logging middleware -> loggingMiddleware.cs

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

## **Herramientas, entorno e infraestructura**

Se usará la plataforma de desarrollo colaborativo GitHub, teniendo una preferencia por esta por su uso sencillo y colaborativo, así como un manejo y combinación de ramas del proyecto, la opción gratuita que brinda, además de las ventajas ofrecidas por las soluciones empresariales de las que provee, así mismo cuenta con una curva de aprendizaje rápida.

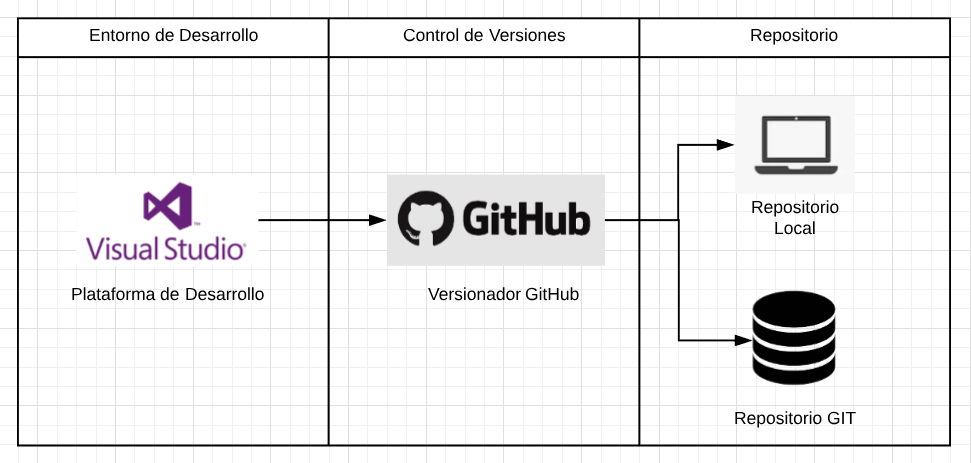


Figura 2. Estructura del trabajo de github con la herramienta Visutal Studio Community

### **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

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**

* Visual Studio Community (Frontend y Backend)

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades online bajo Windows Azure en forma del editor Monaco.

Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y consolas, entre otros.

### **2.5. Cronograma de trabajo**

****

# **Actividades de la SCM**

## **Identificación:**

### **Lista de Clasificación de CI**

En la tabla 3 se especifican los ítems con su respectiva clasificación:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tipo Ítem | Nombre Ítem | Origen | Extensión | Proyecto |
| Evolución | Plan de Gestión de la Configuración | Empresa | DOCX | - |
| Evolución | Lista de Requisitos | Proyecto | XLSX | SVR |
| Evolución | Cronograma de Proyecto | Proyecto | MPP | SVR |
| Evolución | Documento de Negocio | Proyecto | DOCX | SVR |
| Evolución | Plan de Proyecto | Proyecto | DOCX | SVR |
| Soporte | Visual Studio Community | Proyecto | EXE | SVR |
| Soporte | MS Server | Proyecto | EXE | SVR |
| Fuente | Formulario Pedido | Proyecto | ASPX | SVR |
| Fuente | Reporte de Ventas | Proyecto | ASPX | SVR |
| Fuente | Formulario de Cobro | Proyecto | ASPX | SVR |

**Tabla 3. Lista de Clasificación de CI.**

* + 1. **Definiciones de Nomenclatura de los Items:**

Teniendo en cuenta la clasificación tomada a los ítems de configuración se tomará la siguiente nomenclatura:

* **Ítems en evolución**:
  + Sea el caso de ítems de empresa → [Iniciales del documento].[docx|\*]

Ejemplo: PGC.docx

* + Sea el caso de ítems de proyecto → [Iniciales del proyecto]\_[Iniciales del documento].[docx|\*]

Ejemplo: SVR\_DN.docx

* **Ítems fuente:**
  + Para los ítems de desarrollo de proyectos FrontEnd:
    - Todo ítem deberá ser nombrado en CamelCase, si son varias palabras todas deberán estar juntas y la primera letra de cada palabra en mayúsculas a excepción de la primera letra.
    - Los componentes: [NOMBRE DEL COMPONENT O FEATURE].[Extensión]

Ejemplo: estilosFormularios.js

* + - Las vistas: [NOMBRE DEL COMPONENT O FEATURE].[Extensión]

Ejemplo: reporteVenta.aspx

* + - Los controladores [NOMBRE DE COMPONENT O FEATURE].[Extension]

Ejemplo: controller.cs

* + Para los items de proyectos BackEnd:
    - Todo ítem deberá ser nombrado en CamelCase, si son varias palabras todas deberán estar juntas y la primera letra de cada palabra en mayúsculas a excepción de la primera letra: [nombre del ítem en CamelCase].{cs}

Ejemplo: articulosOperaciones.cs

* + Para los items de Base de Datos:
    - Todo ítem deberá ser nombrado en CamelCase, si son varias palabras todas deberán estar juntas y la primera letra de cada palabra en mayúsculas a excepción de la primera letra: [iniciales del proyecto]\_[nombre del ítem en CamelCase].{cs}

Ejemplo: SVR\_listarVentasPorCliente.sql

* **Ítems de soporte:**
  + Para los proyectos de Front-end y Back-end archivos de configuración serán nombrados: [nombre del archivo en CamelCase].[extesion]
* **Ítem de Archivo:**
  + Serán nombrados: [Iniciales del proyecto]\_[Iniciales del documento].docx
    1. **Lista de Ítems con la Nomenclatura**

En la siguiente tabla 4 se dará a conocer la nomenclatura de algunos ítems teniendo en cuenta la definición de nomenclatura tomada (se tomará como ejemplo el proyecto “Sistema de Ventas de Restaurantes” SVR):

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del Ítem | Extensión | Nomenclatura | Proyecto |
| Plan de Gestión de la Configuración | DOCX | PGC.docx | - |
| Lista de Requisitos | XLSX | SVR\_LR.xlsx | SVR |
| Documento de Negocio | DOCX | SVR\_DN.docx | SVR |
| Plan de Proyecto | DOCX | SVR\_PP.docx | SVR |
| Especificación de Casos de Uso | DOCX | SVR\_ECU.docx | SVR |
| Documento de Análisis | DOCX | SVR\_DA.docx | SVR |
| Formulario Pedido | ASPX | formularioPedido.aspx | SVR |
| Reporte de Ventas | ASPX | reporteVentas.aspx | SVR |
| Formulario de Cobro | ASPX | formularioCobro.aspx | SVR |
| Login | ASPX | Login.aspx | SVR |

**Tabla 4. Lista de Ítems con nomenclatura**