|  |
| --- |
| WhatsApp%20Image%202018-04-03%20at%203.03.05%20PM.jpeg  **PROYECTO**  SPACIA  Sistema de gestión de ambientes  **Plan de Gestión de la Configuración**  **V 1.2** |

**Contenido**

[1. Introducción 3](#_Toc513386425)

[1.1. Propósito del documento 3](#_Toc513386426)

[1.2. Alcance 4](#_Toc513386427)

[1.3. Definiciones 4](#_Toc513386428)

[Abreviaturas 5](#_Toc513386429)

[1.4. Referencias 5](#_Toc513386430)

[2. Gestión de la Configuración de Software 5](#_Toc513386431)

[2.1. Organización 5](#_Toc513386432)

[2.2. Roles y Responsabilidades 7](#_Toc513386433)

[2.3. Políticas, directrices y procedimientos 9](#_Toc513386434)

[2.4. Herramientas, entorno e infraestructura 10](#_Toc513386435)

[Herramientas de control de versiones 11](#_Toc513386436)

[Herramientas de entorno 12](#_Toc513386437)

[2.5. Calendario 13](#_Toc513386438)

[3. Actividades de la SCM 14](#_Toc513386439)

[3.1. Identificación de la configuración 14](#_Toc513386440)

[Ítems de configuración 14](#_Toc513386441)

[Definiciones de nomenclatura de los ítems 14](#_Toc513386442)

[Lista de los ítems con nomenclatura 17](#_Toc513386443)

# Introducción

En el tiempo en que NN-Consulting lleva manejando proyectos de desarrollo de software, han surgido inconvenientes en el manejo de las solicitudes de cambios (tanto previstos como imprevistos) que han llevado a la empresa a realizar múltiples versiones (no controladas) del mismo producto durante todo el ciclo de vida de los proyectos. Con esta mala práctica, la empresa pasaba por momentos de confusión e incertidumbre, debido a que no se contaba con una guía o un documento donde se encontrarán establecidas ciertas reglas para poder llevar a cabo una correcta gestión del versionamiento. Surgían inconvenientes, tales como, que los desarrolladores trabajaban sobre una versión que no era la última aprobada; o que, más de un desarrollador realizaba la misma tarea, pues en la versión que tenían no estaba completa. En consecuencia, era muy complicado realizar un seguimiento y auditoría a las versiones de un mismo producto de software y los documentos relacionados.

Con todos estos inconvenientes identificados, se consideró necesario que las versiones de los productos de software desarrollados por NN-Consulting estén sujetas a mantenimiento y gestión para minimizar los riesgos asociados al tener varias versiones de los distintos ítems, además de la cantidad de los mismos que se maneja.

Ante la necesidad de establecer los roles, responsabilidades, herramientas, políticas y procedimientos para una correcta gestión de la configuración del software; y de la necesidad de llevar a cabo las tareas necesarias para el mantenimiento del software desarrollado, se desarrolló este documento; el cual describe las actividades relacionadas con la gestión de la configuración y el mantenimiento de los productos de software y productos de trabajo de software (documentos de software) desarrollados por NN-Consulting. Este documento permitirá a la empresa gestionar adecuadamente la configuración y mantenimiento de los proyectos a cargo.

## Propósito del documento

Este documento está diseñado para servir de soporte al equipo encargado de la gestión de la configuración o mantenimiento, en el uso de métodos (y los requisitos para el cumplimiento de estos) y en la ejecución de actividades para una adecuada configuración y mantenimiento de los proyectos de software de NN-Consulting.

## Alcance

Este documento tendrá el siguiente alcance:

* Describir los roles, responsabilidades, políticas, procedimientos, herramientas y actividades de gestión de configuración de software que deben ser llevadas a cabo durante el proceso de desarrollo de cualquier proyecto de software de la empresa.
* Definir tanto los productos que se pondrán bajo control de configuración como los procedimientos que deben ser seguidos por los integrantes del equipo de trabajo.
* Identificar y definir la línea base de los elementos para cualquier proyecto de software de la empresa.
* Establecer de los métodos, procedimientos y herramientas para el control y registro del estado de las versiones de cualquier proyecto de software de la empresa.
* Procedimientos para la solicitud de cambios (modificación de cualquier proyecto de software de la empresa).

## Definiciones

* **Ítem de configuración:** Unidad estructural fundamental de un sistema de gestión de configuración.
* **Línea base:** Punto de referencia en el ciclo de vida del desarrollo de software marcado por la finalización y aprobación formal de un conjunto de productos de trabajo predefinidos
* **Identificación de configuración:** Esta actividad se encarga de identificar los elementos, esquemas y herramientas.
* **Control de configuración:** Gestiona los cambios que se realizarán al software durante su ciclo de vida. Determina que cambio se debe hacer, si es que es correcto y la manera de su implementación.
* **Contabilidad del estado de configuración:** Proceso por el cual en el estado actual del desarrollo puede ser juzgado y la historia del ciclo de vida del desarrollo se puede rastrear.
* **Auditoría de configuración:** Identificar los elementos del sistema que satisfacen los requerimientos del cliente.

## Abreviaturas

A continuación, se procederá a mencionar las abreviaturas utilizadas en el desarrollo de documento. Es importante resaltar, que las siglas se manejan en inglés, tomando en cuenta que este documento podría ser usado por terceros. No obstante, se describe la definición traducida.

* CCB: Tablero de Control de Configuración (Configuration Control Board)
* CI: Ítem de Configuración (Configuration Item)
* CM: Gestión de Configuración (Configuration Management)
* SCM: Gestión de la Configuración de Software (Software Configuration Management)
* CMO: Oficial de Gestión de Configuración (Configuration Management Officer)
* SQA: Aseguramiento de calidad de software (Software Quality Assurance)

## Referencias

[1] ANSI/IEEE Std 828-1983, IEEE Standard for Software Configuration Management Plans

[2] ANSI/IEEE Std 1042-1987, IEEE Guide to Software Configuration Management

# Gestión de la Configuración de Software

## Organización

La figura 1 muestra cómo está organizado el desarrollo de los productos de software de la empresa NN-Consulting. Este modelo está alineado a la metodología de desarrollo para que pueda cubrir las expectativas de nuestros clientes, tomando en cuenta la prevención de las actividades de configuración y mantenimiento, así como, de las demás fases de desarrollo de software. Actualmente la empresa usa la metodología SCRUM. En la figura 1, se observa que, para cada Sprint según el ciclo de desarrollo de software, se tiene alineado la configuración y el mantenimiento de manera transversal. Esto se traduce, en que la SCM estará presente durante cada una de las fases de del proyecto integrada con las actividades del proyecto.

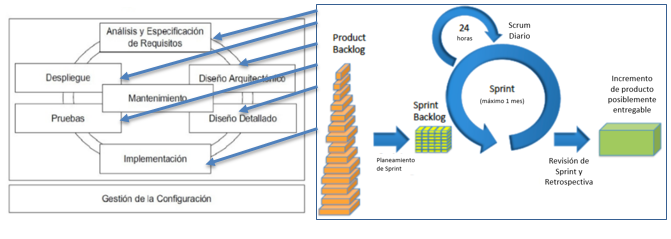


Figura 1: Relación de la metodología de desarrollo usada, respecto a las etapas del proyecto, incluyendo la configuración y mantenimiento.

La siguiente tabla describe la relación que hay entre las actividades específicas de la SCM durante el desarrollo de un proyecto con los roles responsables (Ver Tabla 1).

*Tabla 1: Relación entre las actividades específicas de la SCM y los roles responsables.*

|  |  |
| --- | --- |
| Actividades | Rol Responsable |
| Planificar la SCM | CMO |
| Definir de la Línea del proyecto | CMO |
| Seguir la Línea del proyecto | CMO |
| Definir del plan del contrato | CMO |
| Controlar los Cambios | CCB, CMO |
| Describir la Versión | CMO |
| Realizar informes final de SCM | CMO |

En la figura 2, se observa la relación de las actividades de la SCM con las etapas del proyecto.

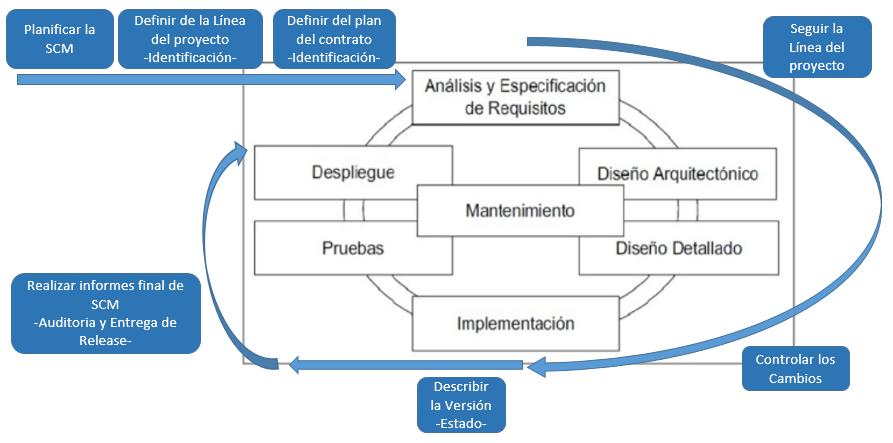


Figura 2: Relación actividades SCM con el desarrollo del proyecto de software

## Roles y Responsabilidades

La unidad de SCM de la NN-Consulting es liderada por el responsable de SCM (Configuration Management Officer, CMO), responsable por las tareas de gestión de la unidad. En general, la unidad actúa como un equipo en el cual sus miembros cooperan para llevar a cabo las tareas de SCM. A continuación, se presenta la tabla 2 con los roles y sus responsabilidades correspondientes.

*Tabla 2: Roles y Responsabilidades*

|  |  |
| --- | --- |
| Rol | Responsabilidades |
| Oficial de Gestión de Configuración | * Gestionar la unidad de SCM. * Definir una estrategia para la implantación de las prácticas de SCM. * Proveer el personal de SCM requerido para las actividades de desarrollo. * Planificar las actividades de SCM. * Informar a los niveles superiores sobre el estado del proceso y las actividades de SCM en los proyectos. * Garantizar la capacitación del personal de SCM y los recursos necesarios para el desarrollo de sus actividades. * Promover el mejoramiento continuo del proceso de SCM. * Gestionar el desarrollo de herramientas para facilitar el proceso de SCM. |
| Tablero de Control de Configuración | * Revisar el hardware o solicitudes de cambio de clientes, en términos del cronograma del proyecto, el costo y el impacto en los clientes. * Recomienda la aprobación o desaprobación del cambio. * Priorización de la incorporación de cambios aprobados. * Reenvía la decisión de aprobación o desaprobación del cambio al CMO * Informa por escrito al cliente de la decisión de cambio. |
| Ingenieros | * Desarrollar y coordinar las actividades de SCM planificadas para el proyecto. * Establecer y participar de la identificación, control, auditoría e informe sobre el estado de la configuración. * Participar en el establecimiento de la biblioteca del software. * Monitorear el correcto funcionamiento de la biblioteca del software. * Interactuar con los desarrolladores y la unidad de SQA. |
| Bibliotecarios | * Diseñar y establecer la biblioteca del software para cada proyecto de desarrollo durante la etapa de planificación. * Establecer y mantener el software y la documentación de cada proyecto de acuerdo con un proceso documentado * Proveer a los desarrolladores las copias de las líneas bases requeridas para sus diferentes tareas. * Mantener y distribuir un índice con el contenido de cada biblioteca. * Entregar la copia original para la implementación de los cambios aprobados por el CCB. * Informar a los desarrolladores sobre los cambios a los ítems * Registrar y mantener copias de las antiguas versiones. * Apoyar la elaboración de informe sobre el estado de la configuración. |
| Personal de apoyo | * Desarrollar tareas administrativas para los representantes de SCM. * Completar los informes de SCM. * Interactuar con las diferentes herramientas utilizadas para SCM durante un proyecto. * Interactuar con el personal de SCM según sea necesario. |

## Políticas, directrices y procedimientos

Respecto a los documentos en el repositorio de GitHub, todos tienen derechos de administrador, es decir, 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 estrictamente prohibida.
* Correcta comunicación entre los miembros encargados de la gestión de la configuración.
* Toda dependencia en los proyectos deberá ser estrictamente documentada.

**Directrices**

Las directrices se encuentran en un documento aparte. Ver repositorio: Documentos/DirectricesDeSCM.docx.

**Procedimientos**

* Respecto a la modificación de archivos pertenecientes a la gestión de la configuración:
  + - 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 otros miembros, lo cual repercutiría en un sobreescribimiento del ítem.
    - Modificar el ítem apropiadamente.
    - Subir los cambios.
    - Hacer los merge necesarios para que el ítem modificado se encuentre disponible para todos.
    - Informar que el cambio fue realizado.

## Herramientas, entorno e infraestructura

En la figura 3, se muestra el mapeo del uso de las diferentes herramientas que se usan para el manejo de las versiones de los productos de los proyectos. Actualmente, NN-Consulting trabaja con las siguientes herramientas:

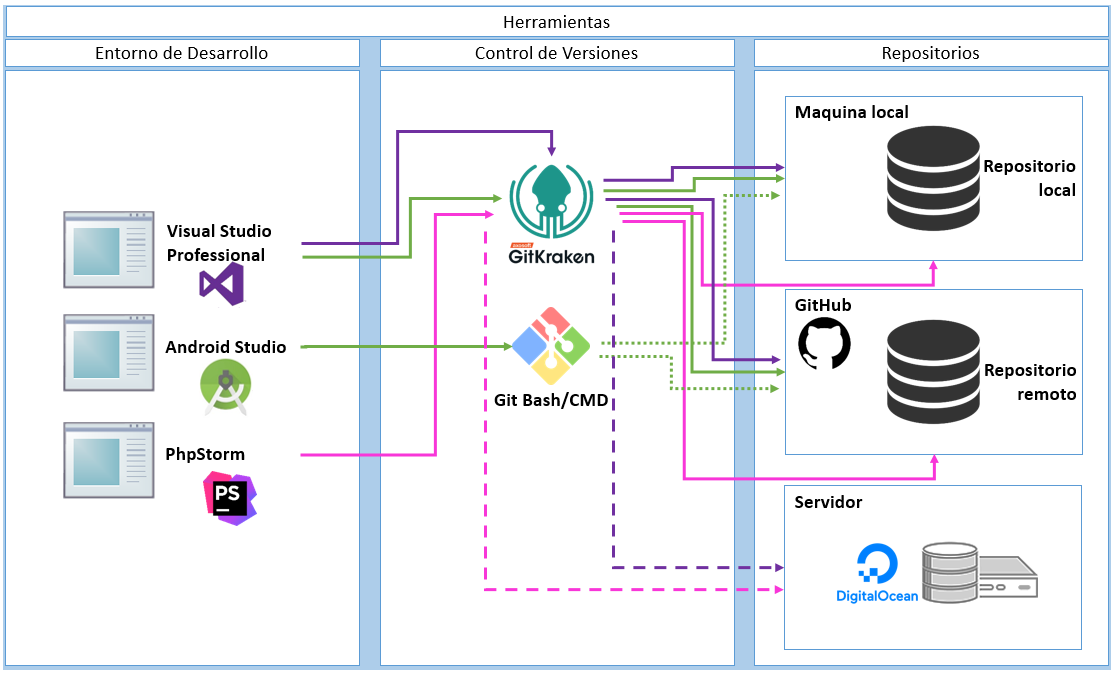


Figura 3: Mapeo del uso de las herramientas de control de versiones y entorno

## Herramientas de control de versiones

* + - Github

Es una plataforma de desarrollo colaborativo online, utiliza control de versiones git para el mantenimiento y versionado del código fuente, añadiendo una serie de servicios extras para la gestión del proyecto y el código fuente. Permite alojar nuestro código en repositorios públicos, la versión Premium permite tener repositorios privados, pero para el desarrollo de este proyecto se usará la versión gratuita. 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 hacer mejoras, sugerencias y optimizaciones.
* Herramienta de versión de código, donde puedes añadir anotaciones en cualquier punto del proyecto.
* Un visor de ramas o branch: donde se puede conocer el progreso que llevamos en nuestro proyecto, mostrando las actualizaciones o commits realizados a partir de nuestro master.
  + - Git

Es una herramienta de tipo consola que posee múltiples comandos que ayudan a tener un buen control de versiones, permite manipular y gestionar todo el proceso a realizarse en el proyecto. Posee la misma funcionalidad tanto en Windows, Linux o Mac, permite un desarrollo no lineal lo que permite una gran rapidez en la gestión de ramas. La herramienta será usada tanto en el desarrollo Front End, Back End, Android y documentación. Git Bash es la herramienta para consola más utilizada por los desarrolladores de NN-Consulting.

* + - Git Kraken

Proporciona una interfaz gráfica multiplataforma para git. Es bastante amigable y nos permite visualizar y gestionar de una forma más sencilla y hacer un seguimiento a nuestro historial en el repositorio y sus distintas ramas al igual que git esta herramienta será usada tanto en el desarrollo FrontEnd, BackEnd, Android y documentación.

## Herramientas de entorno

* + - Visual Studio Professional (Frontend-Web)

Visual Studio Professional es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows y **MacOS** (Actualmente-2018 Sistema Operativo más utilizado por los desarrolladores en NN-Consulting). Soporta múltiples lenguajes de programación, tales como C++, **C#** (Actualmente-2018 el lenguaje más utilizado por los desarrolladores en NN-Consulting), Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como **ASP.NET MVC** (Actualmente-2018 el más utilizado por los desarrolladores en NN-Consulting), Django, etc. Permite administrar el código en repositorios GIT que hospede cualquier proveedor, como GitHub y Visual Studio Team Services.

* + - Android Studio (Móvil)

Es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014. Está basado en el software IntelliJ IDEA de JetBrains, y es publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

* + - PhpStorm (Back-End)

PhpStorm es un IDE de programación desarrollado por JetBrains. Es uno de los entornos de programación más completos de la actualidad, permite editar código no sólo del lenguaje de programación php como lo indica su nombre. PhpStorm permite trabajar con Symfony, Drupal, WordPress, Zend Framework, **Laravel** (Actualmente-2018 framework más utilizado por el equipo de desarrollo en NN-Consulting), Magento, Joomla!, CakePHP, Yii, y otros frameworks.

* + - Servidor-Digital Ocean

DigitalOcean es un proveedor estadounidense de servidores virtuales privados, basado en la ciudad de Nueva York. La compañía alquila instalaciones de centros de cómputo existentes, incluyendo sitios como Nueva York, Toronto, Bangalore, Ámsterdam, San Francisco, Londres y Singapur. El servidor con el que cuenta NN-Consulting tiene las siguientes características: VPS Linux (Ubuntu 16.04) 1 CPU, 1GB RAM, 25 GB SSD.

## Calendario

A continuación, se presenta la tabla de desarrollo de actividades (hitos) incluidas dentro del Proceso de Gestión de la Configuración de Software para NN-Consulting (Ver Tabla 3).

Tabla 3: Calendario del Plan de Gestión de la Configuración

|  |  |  |
| --- | --- | --- |
| Actividad | Rol | Duración |
| Realizar el plan de gestión de la configuración | CMO, Ingenieros | 1 semana |
| Realizar la actividad de identificación de SCM | CMO, Ingenieros | 1 semana |
| Realizar la definición de la línea base y estructura de librerías | CMO, Bibliotecarios, Ingenieros | 1 semana y media |
| Realizar el plan de gestión de cambios | CMO, CCB, Ingenieros | 1 semana y media |
| Realizar los reportes de estado | CMO, CCB, Bibliotecarios, Personal de apoyo | 1 semana |
| Realizar los reportes de auditorias | CMO, CCB, Ingenieros, Personal de apoyo | 1 semana |
| Realizar la gestión de release | CMO, CCB, Personal de apoyo | 1 semana |

# Actividades de la SCM

## Identificación de la configuración

## Ítems de configuración

En la tabla 4, se procede a listar los ítems clasificados por su tipo, fuente y extensión.

Tabla 4: Lista de Ítems de configuración

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo  E: Evolución F: Fuente  S:Soporte | Nombre de Ítem | Fuente  E: Empresa  P: Proyecto  C: Cliente  V:Proveedor | Extensión |
| E | Plan de la gestión de la configuración | E | DOC |
| E | Documento de especificación de requisitos | P | DOC |
| E | Documento de diseño y arquitectura de software | P | DOC |
| E | Documento de Pruebas | P | DOC |
| E | Manual de usuario | P | DOC |
| E | Aplicación móvil | P | APK |
| F | Código fuente de la aplicación móvil | P | ZIP |
| F | Script de la base de datos | P | ZIP |
| F | Scripts de front-end | P | ZIP |
| F | Vistas del front-end | P | ZIP |
| F | Compilados del código fuente front-end | P | ZIP |
| E | Documentación del Api | P | DOC |
| F | Código fuente back-end | P | ZIP |
| F | Manual de instalación | P | DOC |

## Definiciones de nomenclatura de los ítems

**Ítems en evolución**

Los ítems en evolución son de dos tipos: documentos, y archivos ejecutables o de soporte.

***Documentos***

Los ítems en evolución tipo documento se asignan identificadores únicos que permiten identificar el proyecto y componente (si es aplicable) con el cual está asociado, junto con el nivel de revisión actual. El identificador consiste en una a tres partes separadas por guion bajo en el formato: PROYECTO\_ACRÓNIMO, o, PROYECTO\_ACRÓNIMO\_COMPONENTE.

Los ítems en evolución que son ítems no específicos a un proyecto único, tales como políticas, descripciones de procesos y guías, son identificados únicamente por su acrónimo, por ejemplo: PGC (Plan de Gestión de Configuración)

Los ítems en evolución que son específicos de un proyecto, pero no asociados con un componente del proyecto, utilizan un identificador de dos partes: ACRÓNIMO\_PROYECTO y ACRÓNIMO derivado del tipo de artefacto. Por ejemplo, para identificar el plan de proyecto para el proyecto SPC, tenemos: SPC\_PP.

Los ítems en evolución que son específicos de un proyecto y están asociados con un componente específico, utilizan un identificador de tres partes: ACRÓNIMO\_PROYECTO, ACRÓNIMO\_COMPONENTE, y, ACRÓNIMO derivado del tipo de artefacto. Por ejemplo, para identificar el documento de especificación de requerimientos, del componente B2B (Business to Business) del proyecto ACME, tenemos: ACME-B2B-SRD.

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 de archivo o URL 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.

***Número de versión***

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.

***Número de revisión***

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.

***Archivos ejecutables y de soporte***

Los ejecutables del software y los archivos de soporte son identificados generalmente por el nombre y el número de versión, tales como “Main DB v1.1a"

El esquema de numeración de versiones consiste en tres componentes: Versión.RevisiónActualización. Ej: 1.1a.

***Número de versión***

El número de versión cambia únicamente cuando la arquitectura principal del ítem de software cambia, cuando migramos de un nivel de herramienta de desarrollo a otro, cuando una aplicación es totalmente reconstruida, o cuando se producen cambios sustanciales en la GUI. En este caso, la versión 1.1a se convierte en la versión 2.0

***Número de revisión***

El número de revisión es actualizado cuando se añade nuevas características, funcionalidad y otro contenido, o estas han cambiado significativamente. Normalmente la arquitectura principal o la GUI ha sido extendida o limitada de alguna manera. La razón más común de cambiar el número de revisión es cuando añadimos un nuevo módulo u otra funcionalidad al ítem de software. La secuencia normal de revisión es 1.0, 1.1, 1.2, etc.

***Carácter de actualización***

El carácter de actualización se incrementa cuando el único cambio al ítem de software es corregir uno o más defectos, sin añadir ninguna nueva funcionalidad. Las actualizaciones evolucionan 1.1a, 1.1b, etc. Esta actualización se sobrescribe cuando una revisión combinada, que incluye arreglar defectos y añadir nuevas características, se lleva a cabo. En tal caso, se incrementa el número de revisión y se descarta el carácter, es decir: 1.1b a 1.2.

**Identificación de ítems fuente**

Esto se maneja en base a la herramienta de gestión de la configuración utilizada, o de las utilidades del entorno de desarrollo para versionamiento.

**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, si un editor se actualiza de la versión 2.1 a 2.2a, el rango de versión del ítem de configuración será 2.1 – 2.2a.

## Lista de los ítems con nomenclatura

En esta sección, a cada uno de los ítems que ya fueron identificados

anteriormente en una tabla, se les asigna una nomenclatura en base a las reglas que fueron definidas. (Tabla 5)

Tabla 5: Nomenclatura de los ítems durante el ciclo del desarrollo del software

|  |  |  |
| --- | --- | --- |
| Nomenclatura | Descripción del Item(CI) | Fuente  E: Empresa  P: Proyecto  C: Cliente  V:Proveedor |
| SPC\_PGC | Plan de la gestión de la configuración | E |
| SPC\_DER | Documento de especificación de requisitos | P |
| SPC\_DDAS | Documento de diseño y arquitectura de software | P |
| SPC\_DP | Documento de Pruebas | P |
| SPC\_MU | Manual de usuario | P |
| SPC\_AM | Aplicación móvil | P |
| SPC\_AMCF | Código fuente de la aplicación móvil | P |
| SPC\_SBD | Script de la base de datos | P |
| SPC\_SFE | Scripts de front-end | P |
| SPC\_VFE | Vistas del front-end | P |
| SPC\_CFE | Compilados del código fuente front-end | P |
| SPC\_DAPI | Documentación del Api | P |
| SPC\_CFBE | Código fuente back-end | P |
| SPC\_MI | Manual de instalación | P |