



Asignatura:

Planificación y Gestión de Proyectos Informáticos

Título del documento:

Plan de Gestión de Configuración de Software SCMP

Preparado por:

Carlos Company
Javier Torres
Mathias Moser
William Echeverry
Álvaro Guisado

21/03/2022

Nombre

Fecha

Grado
Curso

Ingeniería Informática
4º Grado, A

Nombre de fichero:

v0.1_0007_SCMP_GA1_SGM.pdf

Fecha:

21/03/2022

Edición:

1.0

Página:






1/51

Registro de cambios

1. TÍTULO DEL DOCUMENTO: Plan de Gestión de Configuración de Software (SCMP)			
2. NÚMERO DE REFERENCIA DEL DOCUMENTO: 0007			
3. ISSUE	4. REVISIÓN	5. FECHA	6. RAZÓN DEL CAMBIO
0	1	17/02/2022	Primera versión del documento.
1	0	21/03/2022	Última revisión del documento, entrega a cliente.

* N/A = No aplica

Registro de revisión de documento

Acción	Responsable	Firma	Fecha
Creado por:	William Echeverry <i>Arquitecto de la Solución</i>		17/02/2022
Revisado por:	Javier Torres Solis <i>Consultor Senior Tester</i>		11/03/2022
Revisado por:	Álvaro Guisado <i>Analista de Datos Senior</i>		11/03/2022
Validado por:	Carlos Company Torres <i>Gestor de la Calidad</i>		18/03/2022
Aprobado por:	Mathias Brunkow Moser <i>Jefe del Proyecto</i>		21/03/2022

Resumen

En el presente documento se definirá con detalle la gestión de la configuración para el desarrollo del sistema de gestión muelles. Se detallará las diferentes políticas del proceso de cambios, configuración y responsabilidades relacionadas con el desarrollo y gestión de la configuración a lo largo del proyecto.

Se definirán las pautas para realizar el control de la configuración además de definir como identificar y gestionar los diferentes componentes de la configuración del proyecto.

Índice

1	Introduction	6
1.1	Purpose of the plan	¡Error! Marcador no definido.
1.2	Definición del plan	¡Error! Marcador no definido.
1.3	Glosario	7
1.4	Referencias	8
2	Gestión.....	8
2.1	Organización	9
2.2	Responsabilidades SCM.....	9
2.3	Gestión de interfaces	10
2.4	Implementación SCMP	11
2.5	Políticas, directivas y procedimientos aplicables.....	11
3	Identificación de la configuración.....	12
3.1	Convenciones.....	12
3.2	Líneas de base.....	13
4	Control de configuración.....	15
4.1	Control del código	15
4.1.1	Estructuración de repositorio	15
4.2	Control de Mídia	17
4.3	Control de cambios	17
4.3.1	Niveles de autoridad.....	18
4.3.2	Procedimientos de cambio.....	19
4.3.2.1	Procedimiento de Código:	19
4.3.2.2	Procedimiento de Documentos:.....	20
4.3.3	Comité de revision	20
4.3.4	Control de la interfaz	21
4.3.4.1	Código	21
4.3.4.2	Documentos.....	21
4.3.5	Procedimientos de cambio de software de apoyo.....	21
5	Configuración de Estado de la Contabilidad	22
6	Herramientas, técnicas y métodos para la GCS.....	22
7	Control de proveedores.....	23
8	Recogida y conservación de documentos.....	23

Índice de tablas

Tabla 1: Acrónimos y Definiciones	7
Tabla 2: Referencias	8
Tabla 3: Convenciones	12
Tabla 4: Permisos de Repositorio	15
Tabla 5: Ramas repositorio	16
Tabla 6: Configuración de nombres	18
Tabla 7: Niveles de Autoridad	19
Tabla 8: Comité de Revisión	20

2 Introducción

En el documento Software Configuration Management Plan (SCMP) se especificaran de forma detallada las actividades de la gestión de la configuración del software del proyecto

2.2 Definición del plan

El objetivo de este documento es proporcionar la identificación, control y seguimiento sobre cada una de las iteraciones y de las fases de los componentes y las versiones que constituyen el producto en cada iteración de trabajo, y rastrear los cambios y las razones que los justifican, al pasar de una iteración a otra.

De esta forma se pretende lograr un desarrollo incremental en el que cada iteración se apoya sobre los resultados de la anterior en lugar de ser una construcción completamente nueva, lo que permite ganar en productividad, facilitar la verificación y generar un producto más estable.







2.3 Glosario

Tabla 1: Acrónimos y Definiciones

Acrónimo	Definición
SCMP	Software Configuration Management Plan
QR	Quick Response
IoT	Internet de las Cosas
SMTP	Simple Mail Transfer Protocol
CDN	Content Delivery Network
GD	General Data Protection Regulation

2.4 Referencias

Tabla 2: Referencias

Id	Tipo	Nombre	Enlace
REF-01	Especificación	2021-22 Sistema Gestión Muelles.pdf	 2021-22 Sistema Gestión Muelles.pdf
REF-02	Estándar	PSS050.pdf – ESA software engineering standards	 PSS050.pdf
REF-03	Estándar	ESA ECSS-M-ST-10-01C.047918:	 ECSS-M-ST-10-01C.pdf
REF-04	Estándar	PSS0509.pdf - Guide to software configuration management	 PSS0509.pdf
REF-05	Documento	v1.0_0001_URD_GA1_SGM.pdf	 v1.0_0001_URD_GA1_SGM.pdf
REF-06	Documento	v1.0_0003_SPMP_GA1_SGM.pdf	 v1.0_0003_SPMP_GA1_SGM.pdf
REF-07	Documento	V1.0_0006_SVVP_GA1_SGM.pdf	V1.0_0006_SVVP_GA1_SGM.pdf
REF-08	Repositorio	Github	https://github.com/UFV-INGINF/gestor-muelles-feedex
REF-09	Guía de Modelo	Gitflow	https://cleventy.com/que-es-git-flow-y-como-funciona/
REF-10	Imagen	Gitflow Worflow	https://learntutorials.net/es/git/topic/1276/analisis-de-tipos-de-flujos-de-trabajo-

3 Gestión

La gestión del Proyecto se realizará en correspondencia con la guía y prácticas habituales de FeedEx con el fin de garantizar la gestión más óptima basada en una amplia experiencia. Además de las figuras más relevantes se definirán en los equipos dentro de los departamentos responsables que permitirán una vista más amplia del proyecto.

3.2 Organización

La organización del proyecto queda definida por los siguientes roles:

- Jefe del Proyecto – Mantiene contacto con todos los responsables de categoría superior y dispondrá en todo momento de la supervisión y control del proyecto para garantizar su realización según los requisitos definidos.
- Stakeholders – Los stakeholders o partes interesadas verificarán lo propuesto por FeedEx para realizar el proyecto de forma satisfactoria y dispondrán de contacto con el Jefe del Proyecto para conocer las adversidades, novedades y estado del proyecto.
- Arquitecto de la solución – Realiza un diseño de la solución según los requisitos y aporta propuestas de valor con el fin de lograr dar la solución más idónea.
- Analista de Datos – Proporciona al proyecto el análisis de los aspectos influyentes que pueden ser obtenidos directamente de los datos que se tratan en todo el proyecto.
- Gestor de la Calidad – Garantiza en la realización del proyecto que se mantienen los estándares de calidad acordados y realiza sugerencias de valor al respecto.
- Consultor – Realiza labores de asesoramiento en el proyecto de forma profesional para obtener un resultado acorde a las buenas prácticas del sector.
- Responsables de Equipos – En cada equipo de personal dentro del proyecto se delega las funciones de responsable de equipo con el fin de obtener una vista amplia y detallada de cada equipo durante su desarrollo y poder atajar los inconvenientes de forma eficaz.

3.3 Responsabilidades SCM

Las responsabilidades en la gestión de la configuración de la gestión del software en el proyecto de sistema de gestión de muelles se realizan según la especificación:

- Jefe del Proyecto – Comprobará que se cumplan las responsabilidades definidas.
- Arquitecto de la solución – Verificará que se adecua la solución planteada a la desarrollada de forma periódica con el equipo de diseño y desarrollo.
- Analista de Datos – Comprobará que se cumplan las propuestas planteadas que han sido aceptadas en reuniones periódicas con el equipo.
- Gestor de la Calidad – Verificará que se cumpla en el desarrollo del proyecto los estándares y buenas prácticas definidas para la realización del proyecto.

- Consultor – Elaborara la función de consulta y asesoramiento en posibles desviaciones o inconvenientes en el desarrollo del proyecto.
- Responsables de Equipos – Velara por que se ajuste el trabajo encomendado con el plan definido y notificará mediante reportes el estado de forma periodica.

3.4 Gestión de interfaces

En el proceso de gestion de interfaces de un sistema software es preciso definir como se tramitan los procedimientos entre las interfaces hardware y software. A continuación, se detalla como se realizará la gestión de interfaces en este proyecto:

1. Interfaz de gestión con cámara de lectura de matrículas – Se emplearán las técnicas y marcas conocidas por FeedEx con respecto a la experiencia en otros proyectos con esta implementación mediante el uso de dispositivos Internet de las Cosas (IoT). Esta interfaz será gestionada y automatizada por los desarrolladores según fabricante.
2. Interfaz de gestion con lector de códigos Quick Response (QR) - Se incorporarán las técnicas y marcas conocidas por FeedEx con respecto a la experiencia en otros proyectos con esta implementación mediante el uso de dispositivos IoT. Esta interfaz será gestionada y automatizada por los desarrolladores según fabricante.
3. Interfaz de gestión con las pantallas de estado - Se creara un usuario con rol gestor y se proyectara en las pantallas ubicadas en el muelle para que los transportistas puedan conocer el estado de los muelles en tiempo real. Esta interfaz será gestionada por los empleados del muelle según indicaciones.
4. Interfaz de gestión con servidor Simple Mail Transfer Protocol (SMTP) – Se implementarán los servidores SMTP definidos por FeedEx para el envío de correos a los transportistas. Esta interfaz será gestionada por los desarrolladores según la política de FeedEx.
5. Interfaz de gestion con la Base de Datos - Se creará la base de datos definida por FeedEx para la gestión y almacenaje de los datos del sistema. Esta interfaz será gestionada por los desarrolladores según la política de FeedEx.
6. Interfaz de gestion con el servidor y el servicio Content Delivery Network (CDN) - Se integrara el servidor para alojar el sistema según lo definido por FeedEx y se mejorará su desempeño mediante el uso de un CDN para asegurar la disponibilidad y seguridad de los componentes principales. Esta interfaz será gestionada por los desarrolladores según la política y acuerdos establecidos por FeedEx.

3.5 Implementación SCMP

Se dispondrá de un sistema de gestión de la configuración el cual se encontrará disponible para su uso desde el principio del desarrollo, ya que dispondremos de nuestros repositorios de GitHub desde el principio de este para así llevar un control sobre nuestro proyecto, el resto de configuraciones, como por ejemplo, las de seguridad, se irán realizando sobre el desarrollo de cada una de las partes donde se necesite, por ejemplo, en la creación de la base de datos se creará la configuración de seguridad por la cual aplicaremos como base la General Data Protection Regulation (GDPR) como norma de protección de datos europea.

Para la revisión del software se creará una Junta de Revisión del Software en la cual se debatirán los puntos que hayan podido causar problemas a lo largo del desarrollo, como dudas en la definición o nuevas ideas de cara a este desarrollo. Esta junta de revisión se convocará cada dos semanas y al fin de cada uno de los puntos clave del proyecto.

En el establecimiento de los baselines trataremos los hitos que tenemos planteados en nuestro proyecto, pero al ser un plan, no podemos asegurar que el contenido de las baselines pueda cambiar según vaya avanzando el proyecto.

El desarrollo final lo tendremos para el 21 de abril, como podemos ver en el Gantt adjuntado anteriormente, aunque también podemos ver como iremos completando partes de este desarrollo hasta llegar a esa fecha, por lo tanto se pueden ir liberando partes del proyecto de software al cliente para que este pueda revisar o proponer un cambio a este, la entrega final se producirá después de haber realizado pruebas de validación y una limpieza de código para que haya la menor cantidad de problemas.

3.6 Políticas, directivas y procedimientos aplicables

Nombramos las distintas políticas y procedimientos que tendrán un impacto en nuestro proyecto:

- Política de desarrollo de sistema en base a los estándares de desarrollo de software de la ESA.
- Se utilizará como base la GDPR, como norma para la protección de datos europea.
- Dispondremos de una política de cookies para informar al usuario que se podrán guardar datos de la sesión
- Dispondremos de una política de privacidad aclarando que el usuario posee total privacidad a respecto de sus datos personales y su seguridad respecto a terceros.

4 Identificación de la configuración

Para el desarrollo de un proyecto de estas dimensiones, vamos a utilizar distintas configuraciones para el control del proyecto. La configuración del proyecto esta definida en dos elementos: la documentacion y el código.

Documentación:

- Todos los documentos asociados al proyecto dispondrán de un control de versiones indicado en el índice de este, para tener un control y seguimiento de los cambios asociados al documento. El control de versiones de los documentos seguirá una convención estipulada en el punto siguiente de la documentación 3.1 *Convenciones*.

Código (software):

- El código del proyecto estará gestionado a través del sistema de control de versiones GitHub. El flujo de trabajo se desarrollara a través de ramas de trabajo definidas en el punto 3.1 *Convenciones*.

4.2 Convenciones

Convención sobre el control de versiones de los documentos:

Tabla 3: Convenciones

Nombre	Tipo	Confidencialidad
Version del documento y nombre del nuevo documento	Estado del documento, documentación de los cambios realizados	Estado condifencial del proyecto
...

Ramas utilizadas en **GitHub** (especificadas en el punto 4.1.1 *Estructuración del repositorio de este documento*):

- **Master:** Rama principal del proyecto, el cual alojará una versión estable del proyecto en todo momento. Gestionado a través de *tags**.
- **Release:** Rama para efecutar merges de la rama develop y verificar su estabilidad para subir a producción. Gestionado a través de *tags*.
- **Develop:** Rama de trabajo secundaria con el codigo desarrollado mas reciente y pendiente de pruebas.

- **Feature:** Rama para el desarrollo de nuevas funcionalidades del proyecto.
- **Bugfix:** Rama para el desarrollo de soluciones de código defectuoso originado en la rama develop.
- **Hotfix:** Rama de depuración de código en estado crítico encontrado en la rama master.

**tags: etiquetas para establecer el estado del repositorio.*

4.3 Líneas de base

4.3.1 Documentos

Id: LB-01	Nombre: Documentos
Estructura Nombre	v[VersiónEstable.VersiónTrabajo_NumReferencia_NomDoc_NomGrupo_Nom Proyecto]
Descripción	Documentos como URD, SPMP, SVVP, SCMP, SQAP.
Herramientas Utilizadas	Word, Excel, Visio, Powerpoint
Componentes	Titulo, Control de Versiones, Resumen, Indice, Indice de Tablas, Apartados, Capítulos, Subcapitulos, Referencias, Anexos
Pruebas	N/A

4.3.2 Requisitos

Id: LB-02	Nombre: Requisitos
Estructura Nombre	idModulo_idRequisito
Descripción	Pueden ser de tipo Restricción (No Funcionales) o de Capacidad (Funcionales)
Herramientas Utilizadas	Word, Plan de Pruebas, CI/CD Pipelines (Para validar)
Componentes	ID, Módulo/Categoría, Pedido Por, Nombre, Descripción, Quiero que (Acción), Para que (Finalidad), Prioridad, Estable. Claro, Verificable, Comentarios, Riegos, Preguntas
Pruebas	Plan de Pruebas, Test Unitarios

4.3.3 GAN

Id: LB-03	Nombre: GAN
Estructura Nombre	v[VersiónEstable.VersiónTrabajo_NumReferencia_NomDoc_NomGrupo_Nom Proyecto]

Id: LB-03	Nombre: GAN
Descripción	Plan de proyecto
Herramientas Utilizadas	Gan Project o Microsoft Project
Componentes	Tareas, Recursos, Presupuesto.
Pruebas	N/A

4.3.4 Modulo de Requisito

Id: LB-05	Nombre: Modulo de Requisitos
Estructura Nombre	idModulo
Descripción	Tipo de requisitos para la función.
Herramientas Utilizadas	Excel, Word
Componentes	Nombre, Tipo
Pruebas	Pruebas Funcionales

4.3.5 Repositorio

Id: LB-06	Nombre: Repositorio
Estructura Nombre	Nombre Repositor
Descripción	Para almacenar al código y realizar la colaboración.
Herramientas Utilizadas	Word, Plan de Pruebas, CI/CD Pipelines (Para validar)
Componentes	Ramas, Tags, Releases
Pruebas	Pipelines

4.3.6 Código Fuente

Id: LB-07	Nombre: Código Fuente
Estructura Nombre	NombreClase.extensión
Descripción	El código de la aplicación.
Herramientas Utilizadas	Visual Studio Code, Compiladores, Navegador Web, Entorno Desarrollo
Componentes	Clases, Funciones, Componentes, Imágenes.
Pruebas	Pruebas unitarias, pruebas de aceptación, pruebas funcionales.

5 Control de configuración

5.2 Control del código

5.2.1 Estructuración de repositorio

Todo código generado por los miembros del equipo deberá estar subido a un repositorio en Github [REF-09] que permita la colaboración de todo el equipo.

Existirá un nivel de permisos vinculado al nivel de autoridad de cada miembro del equipo. Los permisos existentes en el repositorio son:

Tabla 4: Permisos de Repositorio

#idPermiso	Permiso	Descripción	Niveles de Autoridad*
PER-01	Owner	Es el dueño del repositorio, posee todos los permisos.	RG-1
PER-02	Maintainer	Posee la mayoría de los permisos y puede modificar información no sensible en el repositorio. Al igual que el owner puede autorizar pull requests.	RG-1 y RG-2
PER-03	Developer	Puede realizar modificaciones en el código en desarrollo, puede realizar commits y clonar el repositorio en su local para poder trabajar sobre el. Necesita permiso para realizar un pull request	RG-3 y RG-4
PER-04	Reviewer	Tiene acceso a la información estadística del repositorio, pero no puede modificar ninguna rama. Solo tiene permitido realizar la descarga de versiones estables (releases o tags).	RG-5**

* Los rangos asignados a cada miembro del equipo se especifican en el SPMP [REF-06] (0003) en el apartado (3.5) de plan de personal.

** El nivel de autoridad RG-5 no se encuentra en la tabla de personal del documento SPMP [REF-06] (3.5) pues su perfil es externo al equipo de trabajo.

Se deberá realizar un cambio de versión en producción cuando:

- Se produzca un gran cambio en la estructura del código.
- Se desarrolle complemente una feature y esté probado en develop.
- Se requiera una resolución en caliente (bugfix) para estabilizar el código en producción.
- Se identifique una versión estable.

Para el desarrollo y la organización del repositorio utilizaremos el modelo de organización de ramas GitFlow [REF-03]:

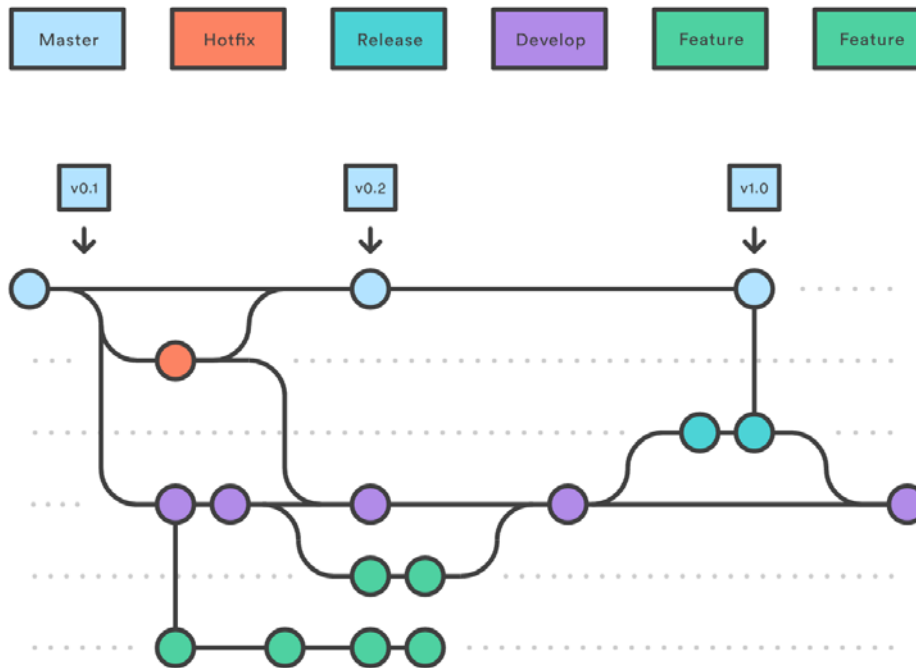


Figura 4.1: Gitflow Workflow [REF-10]

Tabla 5: Ramas repositorio

Nombre Rama	Descripción	Nivel de Aprobación*
master	Rama principal del repositorio, contiene una versión estable y funcional del código. Está normalmente asociado a una versión (tag)	RG-1 + RG-2
release	Previa a una subida a master se deberá de crear una rama release que permita verificar si el código es estable para subirlo a producción y crear una tag.	RG-2
develop	Rama secundaria única, posee el código más reciente del repositorio. Debe integrar de manera funcional las diferentes funcionalidades, permite la realización de tests previos a la subida a máster.	RG-2
feature	Ramas individuales o grupales, deberá de haber un responsable por rama que autorice la subida del código al repositorio.	RG-3
bugfix	Ramas temporales que permiten el trabajo de manera independiente con enforque a la solución de un error identificado en develop.	RG-2 + RG-3
hotfix	Ramas temporales para indenficar y solucionar errores críticos encontrados en la rama master.	RG-2

*Los niveles de aprobación definen que rangos como mínimo deben aprobar un pull request a la rama en cuestión.

5.3 Control de Mídia

Los documentos deberán indicar en su nombre la versión, el número de referencia, la sigla del documento, el grupo de trabajo y el nombre del proyecto.

v[VersiónEstable.VersiónTrabajo_NumReferencia_NomDoc_NomGrupo_NomProyecto]

Todos los documentos deberán de tener un número de referencia único que permita la fácil diferenciación de los documentos y de su orden de realización. Cada documento deberá incluir la fecha de su realización así como los responsables de su realización. El título del documento se redatará de manera completa y la sigla del nombre del documento será utilizada a lo largo del mismo y en otros documentos como código identificativo de tipo.

Todos los documentos sin excepción serán guardados en la plataforma cloud de Onedrive en un Sharepoint compartido accesible por medio de un grupo de Microsoft Teams para el equipo de desarrollo. Todos los miembros del equipo tendrán acceso a los documentos necesarios para cada función. Cada documento principal deberá estar incluido en un directorio personalizado para él, donde se encuentren los demás documentos secundarios/adjunto relacionados con el.

En el periodo de desarrollo y confección del documento se deberán de guardar copias de seguridad de manera regular para evitar la perdida de información. Una vez finalizado el desarrollo se deberá realizar al menos dos copias de seguridad del documento y guardarlas en locales separados.

5.4 Control de cambios

Cada documento deberá contener un control de versiones para facilitar a los miembros del equipo identificar cual versión es la más reciente.

Las versiones entregadas al cliente serán siempre números enteros y englobarán una serie de versiones anteriores consideradas de trabajo o borradores. El cliente recibirá un documento de extensión PDF (.pdf) al menos que sea necesario que el cliente revise y pueda realizar cambios en el documento en caso de revisión para aprobación, en este caso el cliente también recibirá un documento de extensión tipo Word (.docx) de manera adicional al documento pdf.

A continuación constan dos ejemplo de documentos en diferentes versiones:

Tabla 6: Configuración de nombres

Nombre	Tipo	Confidencialidad
v0.3_0007_SCMP_GA1_SGM.docx	Borrador de trabajo	Confidencial Interno
V1.0_0007_SCMP_GA1_SGM.pdf	Entregue al cliente	Confidencial Externo
v1.0_0007_SCMP_GA1_SGM_Review0.7.docx	Revisión del comité	Confidencial Interno

Una versión debe incrementarse siempre que:

- El documento sea transferido de un miembro del equipo a otro.
- El documento sea entregado a un cliente.
- El documento sufra un cambio sustancial que diferencie a el de los demás.

En caso de que una versión sea inválida o rechazada, se permitirá la restauración a una versión anterior, siempre que el documento no haya sido entregado y aprobado por el cliente y se encuentre en una versión estable.

5.4.1 Niveles de autoridad

Cada nivel de autoridad concede más o menos permisos dentro del correspondiente repositorio del proyecto, además de obtener acceso y interactuar con diferentes personas dependiendo del nivel de autoridad existente. Los niveles van de 1 (Más poder) a 5 (Menos poder).

Tabla 7: Niveles de Autoridad

Id de Nivel de Autoridad/Rango	Descripción	Tipo
RG-1	Responsable de coordinar al equipo y informar al cliente del estado actual del proyecto. Es el responsable de la gestión del proyecto. Posee la última palabra y tiene la autoridad para cambiar el rumbo del proyecto.	Coordinador Interno o Externo.
RG-2	Responsable directo del equipo tiene autoridad para modificar las funcionalidades técnicas del sistema para poder diseñarlo o actualizarlo. Depende del responsable del proyecto.	Responsable Interno
RG-3	Responsable de un área específica del desarrollo, centraliza todas las comunicaciones entre los niveles inferiores dentro de un área y los responsables.	Interno
RG-4	Encargado de desarrollar una funcionalidad y realizar un servicio específico dentro de un área de desarrollo. Depende directamente del responsable del área.	Interno
RG-5	Al ser externo al proyecto no tiene autoridad dentro del equipo de trabajo, pero puede afectar al desarrollo del proyecto con cambios de rumbo inesperados o un cambio de expectativas. En general suelen ser los clientes, pero su nivel de autoridad varía dependiendo del proyecto realizado.	Externo

5.4.2 Procedimientos de cambio

Antes de la realización de un cambio se deberá de consultar con al menos un miembro del equipo que posea un rango más que el que posea el miembro.

5.4.2.1 Procedimiento de Código:

Para realizarse un cambio de una rama estable se deberá de seguir el siguiente flujo:

1. Crear una rama bugfix/hotfix si se trata de un bug detectado o release si el cambio añade una nueva funcionalidad a partir de la rama en cuestión.
2. Realizar commits de los cambios necesarios.
3. Solicitar un pull request de la rama creada.
4. Si el pull request es aprobado se añadirá (merge) a la rama en cuestión.
5. Si se producen incidencias, se abrirán ramas para solucionarlas.

6. Se subirá el código a producción para que el cliente pueda acceder a él.

5.4.2.2 Procedimiento de Documentos:

Para realizarse un cambio a una versión estable se deberá seguir el siguiente flujo:

1. Informar al responsable directo de que se ha realizado un cambio.
2. Incrementar una versión al documento de trabajo.
3. Realizar una revisión general del documento.
4. El comité de revisión deberá aprobar y firmar el documento.
5. Se añadirá la versión estable a la tabla de control de versiones.
6. Se entregará al cliente en formato PDF la última versión estable.

5.4.3 Comité de revisión

En todos los documentos se deberá incluir una tabla de revisiones del documento donde se identifiquen a cada miembro del comité con su nombre y función dentro del proyecto. Todo el comité de revisión deberá firmar el documento, indicando la fecha de revisión.

Tabla 8: Comité de Revisión

Id*	Nombre	Responsabilidad
REC-01	Jefe de Proyecto	Aprobar el documento, no se podrá entregar sin la aprobación del Jefe de Proyecto.
REC-02	Arquitecto de la Solución	Crear el documento y revisar los detalles técnicos.
REC-03	Gestor de la Calidad	Verificar la calidad del documento revisando por detalles de gestión e identificando incoherencias.

* El id de recurso es conforme con el apartado de responsabilidades (2.4) del documento SPMP [REF-06] (0003) y con la tabla de responsabilidades en el SVVP [REF-07] (0006) (4.4)

Se podrán añadir revisores al comité en cada documento, siempre que sea necesario demostrar que el responsable en cuestión ha revisado el documento pues el documento se encuentra en el área de conocimiento de un miembro del equipo. Los miembros incorporados al igual que los miembros principales del comité deberán firmar el documento. Los miembros añadidos deberán ser de Rango RG-3 hacia arriba. Cuando se realice una revisión se deberá añadir al final del documento la identificación de versión revisada por ejemplo:

v1.0_0007_SCMP_GA1_SGM_Review0.7.docx*

*Las versiones entregadas al cliente no indicarán la versión de revisión.

5.4.4 Control de la interfaz

5.4.4.1 Código

Para el elemento de configuración de código cuando se necesite realizar un cambio o se necesite añadir una funcionalidad nueva, primero se deberá crear una rama independiente el proceso de desarrollo que contenga el nombre el nombre de la funcionalidad o el tema, por ejemplo:

feature/tests
feature/contacto

Dentro de las funcionalidades o “features” también se podrán incluir sub ramas tipo feature que engloben una subfuncionalidad específica, por ejemplo:

feature/tests/testunitarios
feature/contacto/envioemails

5.4.4.2 Documentos

Para que se realice un cambio en los documentos se deberá crear una nueva versión del mismo y comprobar que los demás miembros del equipo en especial el comité de revisión, están de acuerdo con el cambio.

Si se autoriza de se deberá de especificar en la tabla de control de versiones la razón del cambio que se ha realizado, describiendo la modificación y en que apartado(s).

5.4.5 Procedimientos de cambio de software de apoyo

Para que se apruebe la realización del cambio de software de apoyo se deberá seguir el siguiente procedimiento:

1. Identificar las razones y documentar los fallos/deficiencias encontradas en el software de apoyo actual.
2. Realizar una investigación inicial sobre alternativas al software de apoyo actual y documentar las especificaciones.
3. Realizar al menos 2 presupuestos que incluirán:
 - a. Descripción detallada del producto
 - b. Cálculo de Riegos
 - c. Cálculo de Costes
 - d. Ventajas

- e. Desventajas
 - f. Una demo del producto.
4. Realizar una reunión con todos los miembros del equipo para:
- a. Evaluar la necesidad real.
 - b. Elegir cual presupuesto elegir.
 - c. Verificar la calidad del producto
 - d. Formar a los integrantes como utilizarlo enseñando una demo.
5. Realizar una reunión con el cliente para firmar el presupuesto y integrar en el proyecto

Pasos 1 al 3 se deberán de realizar en un mínimo de 3 días para que no afecte el proyecto dependiendo de la gravedad de la deficiencia del software de apoyo.

Los pasos 4 y 5 se deberán realizar en menos de 2 días, para que en menos de una semana laboral se resuelva la deficiencia del anterior software de apoyo.

Los responsables de firmar el presupuesto serán el comité de revisión y el cliente, que deberá aprobar el presupuesto si es necesario.

Los responsables de elaborar los documentos indicados en el procedimiento serán el Jefe de Proyecto [REC-01] y el Arquitecto de la Solución [REC-02]

6 Configuración de Estado de la Contabilidad

El seguimiento de los cambios dentro de la documentacion, como se explica en los puntos 3 y 4 de este documento, se realiza a traves de copias de seguridad de los documentos alojados en la plataforma OneDrive, tanto a nivel local como en la nube.

EL registro de estos cambios se notifica dentro del mismo, en la tabla anexada dentro con el nombre de "Control de versiones" según el convenio indicado en la tabla anexada en el punto 4.3 *Convenciones*. Todos los documentos deberán de tener un número de referencia único que permita la contabilidad y gestión de las distintas versiones.

Respecto a la contabilidad del estado del software, como indicado en los puntos 3 y 4, se utiliza el gestor de versiones GitHub, utilizando etiquetas para saber el estado del software antes de ser llevado a producción.

7 Herramientas, técnicas y métodos para la GCS

Este desarrollo se basara en la utilización de un conjunto de herramientas para la realización de nuestra aplicación.

Estas herramientas serán principalmente software, como PHP, HTML, Bootstrap, CSS, Javascript, jQuery, MySQL, XAMPP, servidor Apache, Chrome y Firefox, siendo la mayoría de estas lenguajes que utilizaremos para desarrollar nuestro programa.

Para el desarrollo del sistema utilizaremos los estándares de la ESA, y para la definición del módulo legal, se utilizará como base la GDPR, como norma para la protección de dato europea.

El seguimiento de las tareas se hará mediante la herramienta Microsoft Teams y la funcionalidad que esta proporciona para crear, asignar y marcar tareas como completadas.

Las reuniones de seguimiento semanales también se llevarán a cabo mediante la herramienta Microsoft Teams, así como la gestión de la documentación generada durante el proyecto.

La gestión de versiones y cambios del software se llevará a cabo mediante la herramienta Github, de forma que todos los participantes puedan aportar cambios, mejoras y será posible llevar un registro ordenado de esto.

8 Control de proveedores

La gestión del software se hará mediante la plataforma Github, tal y como se explicó de forma detallada en el [Punto 4.1 : Control del código](#).

Para asegurar la integridad del proyecto y su correcta ejecución se seguirán los procedimientos de control periódicos habituales de FeedEx en función de los cargos definidos para cada proyecto en correspondencia con las necesidades del mismo.

9 Recogida y conservación de documentos

La conservación de los documentos generados para este proyecto se hará mediante la plataforma Microsoft Teams. Todos los documentos se subirán a esta herramienta de manera ordenada y separada para tener un fácil acceso a ellos.

Se guardará una versión borrador del documento, la versión final en formato .docx y .pdf y una copia de seguridad en otra carpeta. Las versiones antiguas del mismo documento se almacenarán en una carpeta llamada "Old".

Todos los documentos generados se archivarán y conservarán hasta la finalización del proyecto.