



GI-LABSI-005

PÁGINA 1 DE 17

METODOLOGIA PARA LA ADMINISTRACION DE
PROYECTOS DE SOFTWARE

Versión: 09

Contenido

1	INTRODUCCIÓN	2
2	OBJETIVO	2
3	ALCANCE	2
4	DESCRIPCIÓN	3
4.1	Los Requisitos del SIAM y los Objetivos de la Metodología de Administración de Proyectos de Software	3
4.2	Metodología aplicable a la administración de proyectos software para el SIAM	4
4.3	INICIO DEL PROYECTO	5
4.4	DESARROLLO DEL PROYECTO	10
4.5	CIERRE DEL PROYECTO	12
4.6	MODELO DE EQUIPO DE TRABAJO	14
4.7	GLOSARIO	16
4.8	DOCUMENTOS DE REFERENCIA	16

		GI-LABSI-005
PÁGINA 2 DE 17	METODOLOGIA PARA LA ADMINISTRACION DE PROYECTOS DE SOFTWARE	Versión: 09

1 INTRODUCCIÓN

El uso adecuado de una metodología de administración de proyectos de software proporciona un orden a las actividades del grupo, especifica qué es lo que se debe construir, permite dirigir y planear las tareas que se desarrollan por el equipo, proporciona criterios para hacer seguimiento y medir productos y actividades. Adicionalmente simplifica el mantenimiento de la aplicación, el control de la calidad del producto, y la reutilización de componentes de software.

2 OBJETIVO

En general una metodología para la administración de proyectos de software busca establecer un procedimiento sistemático para optimizar los beneficios y minimizar los riesgos asociados a un proyecto de desarrollo de software considerando apropiadamente los requisitos de los usuarios, la estructura organizacional y la disponibilidad de recursos. La metodología establece roles para las personas involucradas en el desarrollo, proporciona un plan para la puesta en marcha, establece los mecanismos de comunicación necesarios, determina de manera simple cual es la relación entre los componentes o cambios al sistema y en general provee la información requerida para hacer las tareas de construcción y mantenimiento del sistema.

3 ALCANCE

Este documento responde a las necesidades relacionadas con la gestión de proyectos de desarrollo de software inherentes a la función misional asignada a

		GI-LABSI-005
PÁGINA 3 DE 17	METODOLOGIA PARA LA ADMINISTRACION DE PROYECTOS DE SOFTWARE	Versión: 09

INVEMAR relacionada con la consolidación del Sistema de Información Ambiental Marina de Colombia – SIAM y tiene como finalidad racionalizar y optimizar el uso de los recursos y asegurar la calidad de los productos de software que se generen en su ciclo completo. Los lineamientos y normas expuestas se aplicaran, de preferencia, sobre otras normas similares en todas las etapas del ciclo de desarrollo de los proyectos de software. Siempre que sea necesario el documento mismo se someterá a evaluación y actualización.

4 DESCRIPCIÓN

4.1 LOS REQUISITOS DEL SIAM Y LOS OBJETIVOS DE LA METODOLOGÍA DE ADMINISTRACIÓN DE PROYECTOS DE SOFTWARE

Las necesidades del SIAM en el área del desarrollo de sistemas y/o servicios de información con apoyo de tecnologías de la informática actualmente se centran en dos grandes áreas: la primera concierne a las actividades misionales propiamente, la segunda se relaciona con ofrecer la infraestructura operativa que permita la operación del SIAM, para lo cual se hace uso de paquetes de software desarrollados por terceros que requieren de acciones concretas de parametrización y mantenimiento que es indispensable ejecutar siguiendo un proceso metodológico que asegure la calidad de los resultados.

La complejidad y diversidad de las metas fijadas para cada uno de los proyectos de desarrollo de software, a lo que es necesario añadir la especialidad de las temáticas abordadas, hace indispensable implementar una metodología para su administración a fin de garantizar la uniformidad de los productos y asegurar que

		GI-LABSI-005
PÁGINA 4 DE 17	METODOLOGIA PARA LA ADMINISTRACION DE PROYECTOS DE SOFTWARE	Versión: 09

los mismos cumplan con los requisitos planteados por los usuarios. Los beneficios de la aplicación de la metodología en el desarrollo de software son:

- Gestión regular de las expectativas del cliente.
- Resultados anticipados.
- Flexibilidad y adaptación al cambio.
- Mitigación de riesgos.
- Productividad y calidad.
- Alineamiento entre el cliente y el equipo.
- Equipo motivado.

4.2 METODOLOGÍA APLICABLE A LA ADMINISTRACIÓN DE PROYECTOS SOFTWARE PARA EL SIAM

En general una metodología para la administración de proyectos de software busca asignar roles a las personas involucradas en el desarrollo, proporcionar un plan para la puesta en marcha y uso del sistema, establecer los mecanismos de comunicación dentro del equipo de desarrollo, determinar de manera simple cuál es la relación entre los componentes, o cambios al sistema. También, tiene en cuenta la estructura organizacional actual y las políticas operativas del INVEMAR, la aplicación de las técnicas de documentación basadas en diagramas del Lenguaje de Modelado Unificado (UML). En general la metodología provee la información requerida para hacer las tareas de construcción y mantenimiento del sistema.

La metodología que se aplicará en el contexto del SIAM para la construcción, adaptación, mantenimientos o reingeniería de aplicaciones a la medida, servicios

		GI-LABSI-005
PÁGINA 5 DE 17	METODOLOGIA PARA LA ADMINISTRACION DE PROYECTOS DE SOFTWARE	Versión: 09

y sistemas de información, es una adaptación del marco de trabajo SCRUM, metodología de desarrollo incremental e iterativo (Scrum.org, 2015).

Las metodología se divide en las siguientes fases:

- Inicio del proyecto.
- Desarrollo del proyecto.
- Cierre el proyecto.

4.3 INICIO DEL PROYECTO

Si se trata de un nuevo módulo complementario, servicio o sistema de información o se recomienda ejecutar las siguientes actividades:

- Establecer una visión y una misión.
- Plantear un modelo conceptual que se implementa a través de diagrama conceptual.
- Definir componentes gruesos que se desarrollan por fases.
- Establecer perfiles de usuarios.
- Desarrollar servicios en los planos alfanumérico, espacial, documental, de imágenes, de participación ciudadana y el desarrollo de intercambio de datos.
- Generar un marco normativo, incluyendo la adopción de estándares y protocolos.

Si se trata de un mantenimiento de software define una nueva versión basada en los mejoras y problemas detectadas por los usuarios. Se recomienda para el proceso de mantenimiento de software ejecutar las siguientes actividades:



GI-LABSI-005

PÁGINA 6 DE 17

METODOLOGIA PARA LA ADMINISTRACION DE
PROYECTOS DE SOFTWARE

Versión: 09

- Corregir y depurar los errores del sistema que no hayan sido detectados o que aparezcan como resultado de la evolución en las tecnologías del ambiente de trabajo de la aplicación.
- Adicionar funciones de acuerdo a requisitos de los usuarios.
- Mantener actualizada el área tecnológica, mediante la investigación con el propósito de promover el uso de tecnología adecuada para el desarrollo informático.

La fase de inicio del proyecto se divide en dos grandes actividades: **Planificación** y **arquitectura del sistema** (Scrum Manager, 2015).

4.3.1 Planificación

La actividades de la fase de planificación son las siguientes:

1. Creación del Documento de Requisitos de Software, el cual contendrá los requisitos funcionales y no funcionales (Historias de Usuario) que el producto de software debe cumplir al cierre del proyecto. Solo es requerido para proyectos nuevos de software.
2. Con base al **Documento de Requisitos de Software** para proyectos nuevos de software, se desarrollará el Product Backlog donde se ordenarán y priorizarán los requisitos funcionales y no funcionales en tareas y/o sub-tareas que se desarrollarán en los ciclos de desarrollo (sprints) de la fase de Desarrollo. Este documento siempre estará en constante evolución mediante cambios y/o actualizaciones (Scrum.org, 2015). Para actualización y/o mantenimiento de productos de software, solo se actualizará el Product Backlog del proyecto.

		GI-LABSI-005
PÁGINA 7 DE 17	METODOLOGIA PARA LA ADMINISTRACION DE PROYECTOS DE SOFTWARE	Versión: 09

3. Determinación de la fecha de entrega del producto de software y de las funcionalidades de una o más versiones.
4. Se evalúan y definirán los framework para el desarrollo de software y los componentes de la solución de software que serán administrados por paquetes de software desarrollados por terceros que formaran parte del sistema. Ejemplos de herramientas de desarrollo son Java, Sprint Frameworks, Python, Django; Ejemplos de software de terceros son Alfresco, Geonode, Geonetwork.
5. Conformación del equipo o equipos para desarrollar el producto de software. Los desarrolladores del producto quedaran registrados en la plataforma Jira.
6. Descripción y evaluación de los riesgos, junto con las acciones de control.

4.3.2 Arquitectura del sistema

Esta fase consiste en la conceptualización y el análisis. Los diseñadores y arquitectos dividen el proyecto en paquetes basándose en los ítems del Product Backlog. Los paquetes son los componentes que necesitan cambiarse en cada iteración o sprint. (Caso, 2015).

Las actividades de la fase de Diseño o Arquitectura son las siguientes:

1. Revisión y ajuste de los requisitos del Product Backlog.
2. Análisis del dominio de la aplicación para incluir los requisitos que incluye el desarrollo de la mejora o actualización.
3. Identificar problemas del desarrollo o modificaciones.
4. Reunión de revisión del diseño.

		GI-LABSI-005
PÁGINA 8 DE 17	METODOLOGIA PARA LA ADMINISTRACION DE PROYECTOS DE SOFTWARE	Versión: 09

Entregables y/o artefactos que se generan en esta fase

Artefacto	Responsable	Aplicabilidad	Medio de registro y seguimiento
Diagrama Conceptual	Administrador del proyecto o Scrum Master	Requerido para proyectos nuevos	Plataforma Confluence
Diagramas complementarios	Administrador del proyecto o Scrum Master	Opcional	Plataforma Confluence
Documento de Requisitos de Software	Administrador del proyecto o Scrum Master	Requerido para proyectos nuevos de software	Plataforma Confluence
Product Backlog (para registro y seguimiento de requisitos)	Administrador del proyecto o Scrum Master	Requerido	Plataforma Jira
El plan de acción personal	Administrador del proyecto o Scrum Master, Desarrollador	Requerido	(FT-TAL-001).

4.3.3 Entregables de inicio del proyecto

Diagrama conceptual: El diagrama conceptual describe la arquitectura del sistema, también puede ser utilizado para describir la arquitectura de información del sistema. Para el diseño de este modelo se recomienda utilizar el diagrama de bloques, diagramas UML o diagramas ADL (Lenguaje de Descripción de Arquitectura) de la solución a implementar y se debe registrar en la plataforma Confluence.

Diagrama complementarios: En el proceso de planificación o diseño de arquitectura también pueden usarse diagramas adicionales que ayuden a diseñar, construir y documentar el sistema. Como por ejemplo, el diagrama BPMN.

		GI-LABSI-005
PÁGINA 9 DE 17	METODOLOGIA PARA LA ADMINISTRACION DE PROYECTOS DE SOFTWARE	Versión: 09

Documento de Requisitos de Software: La documentación de los requisitos funcionales y no funcionales de software se realizará en la plataforma CONFLUENCE.

Product Backlog: Para el registro y seguimiento de los requisitos de software contenidos en el Product Backlog se utilizará la plataforma JIRA.

Plan de acción personal: Es un documento que aplica los formatos y estándares institucionales adoptados por el grupo de Talento Humano para llevar registro y control de las actividades laborales del recurso humano.

4.3.4 Recomendaciones para tener en cuenta en el diseño de arquitectura

Si de la lista de requisitos se concluyera la necesidad de implementar el componente para su ejecución en ambiente Web, se recomienda considerar el diseño visual y gráfico, los vínculos, gráficas e imágenes, técnica de navegación entre páginas, esquema de seguridad, formularios, la aplicación de estándares CSS, HTML5, de accesibilidad y usabilidad, para lo cual se seguirán las indicaciones estipuladas en los documentos "*Manual para la implementación de la Estrategia de Gobierno en Línea*" y "*Lineamientos y Metodologías en Usabilidad para Gobierno en Línea*" en su versión más reciente. El resultado de este proceso analítico se puede representar en diagramas UML del tipo **mapa de navegación** o mediante **wireframes**. Un wireframe es una representación esquemática de una página web sin elementos gráficos que muestran el contenido y comportamiento de las páginas. Sirve como herramienta de comunicación y discusión entre arquitectos de información, programadores, diseñadores y clientes.

4.4 DESARROLLO DEL PROYECTO

En esta etapa se realiza la codificación o adaptación del software, dividiéndose el desarrollo en iteraciones o **sprints** que proveen como resultado funcionalidades incrementales al fin de cada uno de ellas (Caso, 2015). Los sprints son eventos de tiempo que tienen una duración de 30 días o menos (Glosary, 2015). Cada sprint incluye las siguientes actividades: planeamiento, desarrollo, despliegue, revisión y ajuste. Ver figura 1.

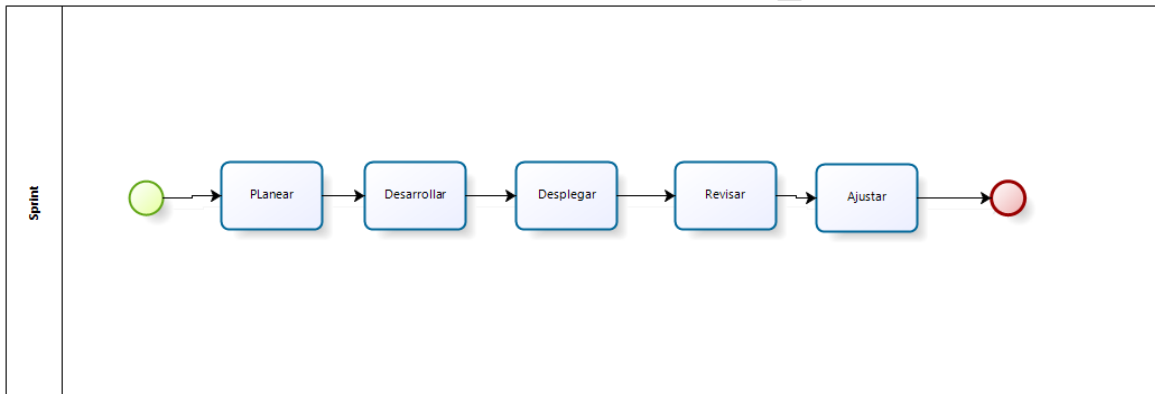


Figura 1. Ciclo de vida de un Sprint

Entregables y/o artefactos que se generan en esta fase

Artefacto	Responsable	Aplicabilidad	Medio de registro y seguimiento
Código fuente de la aplicación	Desarrollador o Scrum Developer	Requerido	Plataforma Bitbucket
Descripción detallada de procesos con notación BPMN	Desarrollador	Opcional	Plataforma Confluence

		GI-LABSI-005
PÁGINA 11 DE 17	METODOLOGIA PARA LA ADMINISTRACION DE PROYECTOS DE SOFTWARE	Versión: 09

Modelo lógico de datos	Desarrollador	Opcional	Plataforma Confluence
Wireframes	Desarrollador o diseñador gráfico	Opcional	Plataforma Confluence
Diagramas UML complementarios	Desarrollador	Opcional	Plataforma Confluence
Cambios de Requisitos de Software	Desarrollador y/o propietario del producto	Opcional	Plataforma Jira

4.4.1 Entregables de desarrollo

Código fuente de la aplicación: Son un conjunto de archivos que implementa las funcionalidades o requisitos de software solicitados por el propietario del producto. El control de versiones del código fuente de las aplicaciones quedará registrado en la plataforma Bitbucket.

Descripción detallada de los procesos con notación BPMN: Algunos sistemas implementan o automatizan procesos, estos procesos quedarán documentados mediante diagramas BPMN y registrados en la plataforma Confluence. El diligenciamiento de este documento es opcional, dependerá del tipo de proyecto o necesidad de documentación.

Módulo lógico de datos: Los sistemas que requieran documentar sus modelos de datos podrán utilizar los diagramas UML y se registrarán en la plataforma Confluence. Es recomendable acompañar este diagrama con el diccionario de datos del modelo.

Wireframes: Es un esquema de página o plano de pantalla que representa el esqueleto o estructura visual de un sitio web.

		GI-LABSI-005
PÁGINA 12 DE 17	METODOLOGIA PARA LA ADMINISTRACION DE PROYECTOS DE SOFTWARE	Versión: 09

Diagramas UML complementarios: En el proceso de desarrollo se pueden generar diferentes diagramas o vistas del sistema que ayudan a diseñar, construir y documentar el proceso, se recomienda para ello los diagramas UML (Lenguaje de modelado unificado). Estos diagramas deben quedar registrados en la plataforma Confluence para el proyecto específico que documentan.

Cambios de Requisitos de Software: Las solicitudes de cambios de requisitos de software durante el proceso de desarrollo se documentan en la Plataforma Jira.

4.5 CIERRE DEL PROYECTO

Esta etapa comienza cuando el equipo de desarrollo decide que todos los requisitos se han completado. En esta etapa se genera la documentación final, se realiza el testing, pre-lanzamiento y el lanzamiento del sistema (Caso, 2015).

Entregables y/o artefactos que se generan en esta fase

Artefacto	Responsable	Aplicabilidad	Medio de registro y seguimiento
Plan de prueba	Dueño del producto	Requerido	RE-LABSIS-001
Manual de usuario	Desarrollador	Requerido	Multimedia o impreso
Manual técnico del sistema	Desarrollador	Opcional	Medio electrónico o impreso
Registro de la instalación o actualizaciones de software de terceros	Desarrollador	Requerido	FT-LABSIS-013

		GI-LABSI-005
PÁGINA 13 DE 17	METODOLOGIA PARA LA ADMINISTRACION DE PROYECTOS DE SOFTWARE	Versión: 09

4.5.1 Entregables de cierre

Plan de Prueba (Plantilla RE-LABSIS-01): Indica los criterios que se aplicaran para evaluar la calidad del producto, asignando las personas responsables de ejecutarlo. El plan debe incluir el registro detallado de las falencias encontradas en el producto.

Manual de Usuario: Este manual tiene como objetivo orientar al usuario sobre la manera óptima de utilizar la aplicación, guiarlo sobre los procedimientos que debe seguir para instalarla y resolver las dudas que surjan al momento de operar el sistema. Un manual de usuario puede desarrollarse en cualquier formato (impreso, video, PDF) y preferiblemente estará disponible para su acceso dinámico siempre que el usuario lo requiera, debido a lo cual todo aplicativo tendrá una sección de ayuda. Se recomienda incluir en el manual de usuario los siguientes apartados:

- Introducción: Presentación del sistema, su origen, propósito y proyecciones.
- Objetivos del software
- Personas a quienes va dirigido.
- Requisitos para el funcionamiento del software.
- Descripción de la interfaz gráfica completa, preferiblemente siguiendo secuencia de funcionamiento de los módulos del sistema.
- Los procedimientos a seguir para la extracción o inserción de datos a partir de archivos planos, de tipos de documentos diferentes o de otras bases de datos.

		GI-LABSI-005
PÁGINA 14 DE 17	METODOLOGIA PARA LA ADMINISTRACION DE PROYECTOS DE SOFTWARE	Versión: 09

- Los mensajes de error, su significado y los procedimientos a seguir cuando se presenten.
- Los procedimientos a seguir para acceder a soporte técnico
- Glosario de términos técnicos del software y de la aplicación.

Manual técnico del sistema: Este documento recopila y documenta todos los diagramas o vistas UML y procedimientos necesarios para construir o reconstruir e instalar todo el sistema. Será generado en modo digital o impreso y registrado en la plataforma Confluence.

Registro de la instalación o actualizaciones de software de terceros: Los paquetes de software desarrollados por terceros deberán llevar un registro de su instalación y la versión en funcionamiento. Para su registro y control de seguimiento se aplica el formato FT-LABSIS-013.

4.6 MODELO DE EQUIPO DE TRABAJO

Para el desarrollo de un producto de software en equipos Scrum tiene tres roles principales:

SCRUM MASTER

Es el responsable de asegurar que el equipo siga las prácticas de Scrum. Sus principales funciones son:

- Ayuda a que el equipo de desarrollo y la organización adopten Scrum.
- Liderar el equipo Scrum, buscando la mejora en la productividad y calidad de los entregables.
- Ayudar a la autogestión del equipo.

		GI-LABSI-005
PÁGINA 15 DE 17	METODOLOGIA PARA LA ADMINISTRACION DE PROYECTOS DE SOFTWARE	Versión: 09

- Gestiona e intenta resolver los impedimentos con los que el equipo se encuentra para cumplir con las tareas del proyecto.

PROPIETARIO DEL PRODUCTO (PRODUCTOWNER)

Es la persona responsable de gestionar las necesidades que serán satisfechas por el proyecto y asegurar el valor del trabajo que el equipo lleva a cabo. Su aporte al equipo se basa en:

- Recolectar y ordenar las necesidades o requisitos de usuarios.
- Aceptar el producto software al finalizar cada iteración o sprint.

EQUIPO DE DESARROLLO

El equipo está formado por los desarrolladores, que convertirán las necesidades del ProductOwner en un conjunto de nuevas funcionalidades, modificaciones o incrementos del producto de software final. El equipo de desarrollo tiene características especiales que son:

- Auto-gestionado: El mismo equipo supervisa su trabajo. En Scrum se potenciarán las reuniones del equipo, aumentando la comunicación.
- No distribuidos: Es conveniente que el equipo se encuentre en el mismo lugar físico. Esto facilita la comunicación y la autogestión que nace del mismo equipo.
- Tamaño óptimo: Un equipo de desarrollo Scrum (sin tener en cuenta al ProductOwner y al Scrum Master) estaría compuesto por al menos un desarrollador y máximo nueve desarrolladores.

		GI-LABSI-005
PÁGINA 16 DE 17	METODOLOGIA PARA LA ADMINISTRACION DE PROYECTOS DE SOFTWARE	Versión: 09

4.7 GLOSARIO

- **Artefacto:** Es un producto tangible resultante del proceso de desarrollo de software.
- **BPMN:** Es una notación estándar de modelado de procesos de negocio para mejorar la comunicación en la organización. (OMG, 2015)
- **CSS:** Las hojas de estilo en cascada son un estándar para aplicar estilos visuales a las páginas web.
- **HTML:** Lenguaje de marcado de hipertexto, es un estándar para la construcción de sitios web.
- **Plataforma Confluence:** Es un software para la documentación y colaboración en equipo. La URL de la plataforma que usa LABSIS es: <https://www.atlassian.com/software/confluence> . Todo el equipo de trabajo tiene acceso a esta plataforma.
- **Plataforma Bitbucket:** Servicio de alojamiento basado en la web para llevar control de versiones de código fuente de aplicaciones software. URL: <https://bitbucket.org/>.
- **Plataforma Jira:** Es un software de gestión de proyectos. La URL de la plataforma que usa LABSIS es: <https://siaminvemar.atlassian.net/secure/Dashboard.jspa>. Todo el equipo de desarrollo tiene acceso a esta plataforma.

4.8 DOCUMENTOS DE REFERENCIA

- Caso, N. G. (7 de 12 de 2015). SCRUM development process. Buenos aires, Argentina.
- glosary, S. o. (7 de 12 de 2015). <https://www.scrum.org/>. Obtenido de <https://www.scrum.org/>: <https://www.scrum.org/Resources/Scrum-Glossary>
- OMG. (15 de 12 de 2015). *Business Process Model and Notation*. Obtenido de Business Process Model and Notation: <http://www.bpmn.org/>
- Scrum, m. (7 de 12 de 2015). *scrummanager*. Obtenido de Scrummanager: http://www.scrummanager.net/bok/index.php?title=Modelo_original_de_Scrum_para_desarrollo_de_software
- Scrum.org*. (7 de 11 de 2015). Obtenido de Scrum.org: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf#zoom=100>



GI-LABSI-005

PÁGINA 17 DE 17

**METODOLOGIA PARA LA ADMINISTRACION DE
PROYECTOS DE SOFTWARE**

Versión: 09

Elaborado por: Daniel Ricardo González Polo Carlos Oliveros Villanueva	Cargo: Investigador Científico Investigador Científico
Revisado por: Julián Pizarro Pertúz Paula Cristina Sierra Correa	Cargo: Jefe Laboratorio de Servicios de Información Coordinadora GEZ
Aprobado por: Jesús A. Garay Tinoco	Cargo: Subdirector de Coordinación de Investigaciones

Fecha de aprobación del documento: 2019-01-31