



MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE

Aprobado con Resolución Administrativa
AGETIC/RA/0042/2022, de 26 de mayo de 2022

**UNIDAD DE PRODUCCIÓN Y
ACTUALIZACIÓN TECNOLÓGICA
(UPAT)**



FICHA DE CONTROL DE CAMBIOS

Nombre del documento: Procedimientos, Políticas y Lineamientos de Desarrollo de Software”.

Código del Documento: AGETIC/RA/0029/2019.

CONTROL DE CAMBIOS		
REF.	VERSIÓN ANTERIOR	VERSIÓN ACTUAL
1	Documento "Procedimientos, Políticas y Lineamientos de Desarrollo de Software" aprobado con la Resolución Administrativa AGETIC/RA/0029/2019 de fecha 30 de mayo de 2019.	Documento editado a partir de los puntos de políticas y lineamientos del documento "Procedimientos, Políticas y Lineamientos de Desarrollo de Software"
2		Esta nueva versión incluye el punto 9. Estándares y se reorganizó los puntos de políticas y lineamientos
3		El nuevo documento no incluye la definición de procedimientos.

RESPONSABLES DE ELABORACIÓN, REVISIÓN Y DE CONFORMIDAD

Elaborado por:

Alfredo Torrez Garay
Cargo: Responsable del Área
de Desarrollo Tecnológico

Revisado por:

Esteban Saavedra
Cargo: Jefe Unidad de
Producción y Actualización
Tecnológica

Revisado por:

Adriana Flores Orihuela
Cargo: Responsable de
Planificación
Firma Digital

Conformidad:

Vladimir Terán
Cargo: Director General
Ejecutivo
Firma Digital

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

CONTENIDO

CAPÍTULO I	4
ASPECTOS GENERALES	4
1. Objeto	4
2. Marco Normativo	4
3. Alcance y/o Ámbito de Aplicación	4
4. Previsión	4
5. Definiciones	5
6. Aprobación, Vigencia, Difusión e Implementación del Reglamento	5
7. Revisión y Actualización del Reglamento	5
CAPÍTULO II	6
POLÍTICAS, LINEAMIENTOS Y ESTÁNDARES DE DESARROLLO Y ACTUALIZACIÓN	6
8. Políticas	6
9. Estándares	7
10. Lineamientos	9

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

CAPÍTULO I ASPECTOS GENERALES

1. Objeto

Definir las políticas, lineamientos y estándares aplicables a los procesos de desarrollo y actualización de productos de software dentro del Área de Desarrollo Tecnológico (ADET) dependiente de la Unidad de Producción y Actualización Tecnológica (UPAT) de la AGETIC.

2. Marco Normativo

Las políticas, lineamientos y estándares, tiene como marco normativo las siguientes disposiciones:

- a. Constitución Política del Estado, 7 de febrero de 2009.
- b. Ley N° 1178, de 20 de julio de 1990, de Administración y Control Gubernamentales.
- c. Decreto Supremo N° 23318-A, de 3 de noviembre de 1992, Reglamento de la Responsabilidad por la Función Pública.
- d. Decreto Supremo N° 25350, de 8 de abril de 1999, Manual de Técnicas Normativas.
- e. Decreto Supremo N° 26237, de 29 de junio del 2001, Reglamento de Responsabilidad por la Función Pública y sus modificaciones.
- f. Decreto Supremo N° 2514, de 9 de septiembre de 2015, crea la Agencia de Gobierno Electrónico y Tecnologías de Información y Comunicación - AGETIC.
- g. Resolución Suprema N° 217055, de 20 de mayo de 1997, Normas Básicas del Sistema de Organización Administrativa.
- h. Resolución Administrativa AGETIC/RA/0030/2021, de 05 de julio de 2021, que aprueba el Reglamento Específico del Sistema de Organización Administrativa de la AGETIC.
- i. Resolución Administrativa AGETIC/RA/0029/2019, de 30 de mayo de 2019, que aprueba el documento "Procedimientos, Políticas y Lineamientos de Desarrollo de Software".

3. Alcance y/o Ámbito de Aplicación

El presente Manual es de aplicación para todo el personal con ítem y consultores de línea de la Unidad de Producción y Actualización Tecnológica (UPAT) cuyas actividades están relacionadas con los procesos de desarrollo y actualización de software del Área de Desarrollo Tecnológico (ADET).

4. Previsión

En caso de presentarse dudas, omisiones, contradicciones y/o diferencias en la interpretación de las Políticas, Lineamientos y Estándares del presente Manual, éstas serán solucionadas en los alcances y previsiones establecidas en las disposiciones legales y normativas pertinentes.

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

5. Definiciones

- a. **Producto de Software**, es una colección integrada de componentes lógicos que se originan a partir de la ejecución de un conjunto de fases -denominado ciclo de vida- a una lista de requerimientos. La característica principal de un producto de software es que puede ser compartido y empaquetado para su distribución y explotación.
- b. **Desarrollo de Software**, se entiende al proceso por el cual se aplica un conjunto de etapas -denominado **ciclo de vida**- a una lista de requerimientos para analizar, diseñar, programar, probar y producir un producto de software.
- c. **Actualización de Software**, es el proceso que se aplica a los productos de software desarrollados, una vez realizada la implementación y despliegue en un ambiente de producción. Normalmente, luego de analizar y dimensionar los requerimientos se procede con el desarrollo, siguiendo el ciclo de vida definido a partir de la etapa de análisis, diseño o programación. La dinámica de actualización no solo se aplica al producto de software funcional sino se aplica también a los entregable, artefactos y componentes creados o modificados por este proceso.
- d. **Tipos de Actualización**, la actualización de un producto de software se realiza debido a las siguientes razones: corrección de errores, mejora del rendimiento, incremento de funcionalidades y/o atención casos de soporte.
- e. **Marco de Trabajo**, para el desarrollo y actualización de software se adopta un marco de trabajo el cual permite definir la dinámica de producción, los entregables, los artefactos y componentes a crear por el equipo de desarrollo.
- f. **Estándares**, establecen un conjunto de instrucciones que se deben seguir para lograr un desarrollo y actualización uniforme de los componentes y artefactos de un productos de software.

6. Aprobación, Vigencia, Difusión e Implementación del Manual

El presente manual deberá ser aprobado por el Director General Ejecutivo de la AGETIC mediante Resolución Administrativa.

La vigencia del manual será a partir de la fecha establecida en la Resolución Administrativa de aprobación.

La difusión del manual será realizada por el Área de Planificación (AP) en coordinación con la Unidad de Producción y Actualización Tecnológica (UPAT), siendo de conocimiento general por el personal de la AGETIC.

La implementación del manual será efectuada por la Unidad de Producción y Actualización Tecnológica (UPAT).

7. Revisión y Actualización

El presente manual deberá ser ajustado y/o actualizado cuando se produzcan cambios o ajustes en el marco normativo, o cuando por razones internas y/o del entorno se justifique realizar modificaciones.

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

El Área de Desarrollo Tecnológico (ADET) dependiente de la Unidad de Producción y Actualización Tecnológica (UPAT) es responsable de la actualización del presente manual y se realizará en base a:

- El análisis de la experiencia derivada de su aplicación.
- Las necesidades institucionales que se presenten.
- Los cambios que se hacen en las disposiciones legales.
- Los cambios generados por el Consejo para las Tecnologías de Información y Comunicación del Estado Plurinacional de Bolivia (CTIC-EPB) sobre el documento *“Lineamientos y Estándares de Desarrollo de Software para el rol Desarrollador”*.

El Área de Desarrollo Tecnológico (ADET) en coordinación con el AP realizará el ajuste y actualización del manual cuando se produzcan los cambios señalados.

Toda vez que el manual sea actualizado, deberá darse cumplimiento al numeral precedente de Aprobación, Vigencia, Difusión e Implementación.

CAPÍTULO II POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE

8. Políticas

Para la realización de los procesos de desarrollo y actualización de software de la UPAT se deben considerar las siguientes políticas: Alta disponibilidad, Escalabilidad, Uso Repositorio y Versionamiento, Seguridad, Pruebas Automatizadas, Documentación y Entornos de Operación.

8.1 Alta disponibilidad

La alta disponibilidad está relacionada con la habilidad de un grupo de usuarios para acceder al sistema, por tanto, el diseño del software debe contemplar la replicación asegurando la característica de redundancia, y por ende que el servicio que proporciona el software sea continuo y esté siempre disponible.

8.2 Escalabilidad

El diseño del software debe contemplar este concepto, de tal manera que cuando haya un aumento significativo del número de usuarios, la adición de nuevas funcionalidades, u otros cambios, el rendimiento del software debe mantenerse sin afectar su capacidad de adaptación y respuesta.

8.3 Uso Repositorio y Versionamiento

El código fuente del software desarrollado debe ser resguardado y versionado mediante el

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

Sistema de Control de Versiones de la AGETIC, el cual almacenará un registro histórico de la creación, modificación y eliminación en los archivos y carpetas relacionados a cada proyecto de software.

8.4 Seguridad

Durante la ejecución de los proyectos de software se deben contemplar los aspectos de seguridad relacionados fundamentalmente al resguardo de la información y al uso adecuado de las aplicaciones. Se adopta el modelo de autenticación para confirmar la identidad de los usuarios y la gestión de roles y permisos para controlar el acceso a la información y ejecución de funcionalidades en cada aplicación.

8.5 Pruebas automatizadas

El desarrollo de software debe contar con pruebas automatizadas que evalúen las funcionalidades principales del software, mismas que deben estar debidamente registradas en el Sistema de Control de Versiones de la AGETIC.

8.6 Documentación

Cada proyecto de software debe contar con la documentación necesaria y registrada en el repositorio oficial de código y documentación para entender lo que hace, desde el análisis y diseño hasta el despliegue. La documentación se debe completar a lo largo de las etapas de desarrollo.

8.7 Entornos de operación

Los proyectos de software se deben contemplar para que estos sean desplegados en diferentes entornos de operación (p. e. *desarrollo, pruebas, preproducción y producción*) para garantizar que el software se pruebe rigurosamente antes de implementarlo y ponerlo a disposición de los usuarios finales.

9. Estándares

En los procesos de desarrollo y actualización de software se deben aplicar los siguientes estándares: Entornos de Operación, Denominación de Grupos y Repositorios, Nombre para Manejo de Versiones.

9.1 Entornos de Operación - Estándar E001

Para el desarrollo y/o actualización de productos de software se deben considerar los siguientes entornos de operación:

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

- *Desarrollo (development)*. Es el entorno donde se realiza el desarrollo y actualización de funcionalidades, corrección de errores, optimización y actualizaciones de versiones.
- *Pruebas (test)*. Es el entorno utilizado para aplicar las pruebas automatizadas y de calidad.
- *Preproducción (sandbox)*. Es el entorno utilizado para aplicar las pruebas de aceptación por parte de los interesados del producto, similar al de producción.
- *Producción (production)*. Es el entorno utilizado cuando el software está listo para ser puesto a disposición de los usuarios finales, por tanto es un entorno optimizado y de operación real.

9.2 Denominación de Grupos y Repositorios

La denominación de grupos y repositorios está basada en el uso de la sigla de la entidad, tipo de servicio, nombre del proyecto y sigla del proyecto.

Variable	Valor
<sigla_entidad>	Para entidades públicas, se utilizará la sigla registrada en el clasificador de entidades del MEFP. Para otros casos, se asignará una sigla en base al nombre de la entidad.
<tipo_servicio>	{backend, frontend}
<nombre_proyecto>	Denominación asignada a cada proyecto.
<sigla_proyecto>	Denominación corta asignada a cada proyecto.
<nombre_componente>	Nombre corto de módulo, componente, elemento o parte del proyecto

Estándar E002: Denominación de Grupo

<denominacion_grupo> ::= agetic/<sigla_entidad>/<nombre_proyecto>

Ejemplo de denominación de grupo (ruta de proyecto):

agetic/mdpyep/registro_comercio

Estándar E003: Denominación de Repositorio

<denominacion_repositorio> ::= <sigla_entidad>-<sigla_proyecto>-<nombre_componente>-<tipo_servicio>

Ejemplo de denominación de repositorio:

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

mdpyep-seprec-homonimia-backend

Ruta Completa de Repositorio

agetic/mdpyep/registro_comercio/mdpyep-seprec-homonimia-backend

9.3 Nombre para Manejo de Versiones - Estándar E004

El manejo de versiones se realiza mediante la asignación de un nombre, código o número único al conjunto de código fuente desarrollado hasta un momento determinado.

Se debe considerar los siguientes criterios para realizar la asignación de nombre para el manejo de versiones de los proyectos de software:

- El uso de un código único como identificador de las versiones.
- Dicho código será el resultado de la sigla o nombre del proyecto más la numeración X.Y.Z o solo la numeración X.Y.Z.

Ejemplos:

- <nombre_proyecto> v[X . Y . Z]
- v[X . Y . Z]

Donde:

- X : se refiere a los grandes cambios y mejoras que se realiza en el software, y/o crea una incompatibilidad con la versión anterior.
- Y : se refiere a las nuevas funcionalidades que son incorporadas al software, realizando cambios menores.
- Z : se refiere a la corrección de errores en las funcionalidades del software.

El manejo de versiones se realiza mediante el uso de ramas o etiquetas (tags) de lanzamiento.

10. Lineamientos

El proceso de desarrollo y actualización de software se debe realizar considerando la segmentación y agrupación de tareas en las siguientes etapas del ciclo de vida del software: *Análisis y diseño; Modelado de Datos; Modelado de Roles y Permisos; Desarrollo y Actualización; Pruebas; Implementación y Monitoreo.*

10.1 Análisis y Diseño

Esta etapa tiene por objetivo analizar los requerimientos y plantear un diseño para la

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

arquitectura de la solución basado en software. Se proponen posibles escenarios de solución, en base a la especificación de requerimientos y plasmados en los artefactos de análisis y diseño.

Se recomienda utilizar el Modelo C4 para generar la documentación (entregables, artefactos y componentes) de registro de la etapa de análisis y diseño.

Modelo C4

El modelo C4 es una técnica de notación gráfica ajustada para describir la arquitectura de sistemas del software.

Los niveles de descripción considerados en el modelo C4 son los siguientes:

- *Diagramas de contexto (nivel 1)*: muestran el alcance del sistema y su relación con los usuarios y otros sistemas.
- *Diagramas de contenedores (nivel 2)*: descomponen un sistema en contenedores interrelacionados. Un contenedor representa una aplicación o un almacén de datos.
- *Diagramas de componentes (nivel 3)*: se descomponen los contenedores en componentes interrelacionados y se relacionan los componentes con otros contenedores u otros sistemas.
- *Diagramas de código (nivel 4)*: proporcionan detalles adicionales sobre el diseño de los elementos arquitectónicos que se pueden asignar al código.

De acuerdo a la complejidad de los proyectos de software se realiza una descripción y descomposición en los niveles que sean necesarios para el Modelo C4.

La documentación (de especificaciones, diagramas y otros) generada se debe almacenar y versionar en la carpeta *docs/* en la raíz del proyecto de software, mismos que se debe realizar conforme a la técnica diseño utilizada.

10.2 Modelado de Datos

El modelado de datos permite establecer la estructura, organización, relaciones y restricciones que existen entre las entidades lógica y física de información. El modelado de datos establece la manera cómo serán almacenados y cómo se llegaran a manipular y acceder los datos. Lo que finalmente desemboca en la selección de un tipo de base de datos.

Se recomienda que la base de datos se diseñe para que pueda ser utilizada por un periodo de tiempo largo e independiente de la cantidad y volumen de datos que vaya a almacenar, acceder o recuperar.

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

Durante el modelado de la base de datos, se deben considerar los siguientes aspectos:

- velocidad de acceso requerido;
- volumen de datos a almacenar;
- tipos de datos utilizados;
- facilidad de acceso a los datos;
- tiempo de permanencia en la base de datos; y
- migración/inicialización de los datos necesarios.

El modelado de datos es necesario solamente en sistemas que registran, recuperan y utilizan datos almacenados en una base de datos.

En el ámbito tecnológico existen dos tendencias en el uso de bases de datos, las *relacionales* y las *no relacionales*. Para el modelado de datos es necesario adoptar uno o combinar ambos, dependiendo de los requerimientos y las soluciones tecnológicas que se quieran alcanzar.

Las bases de datos relacionales se basan en la estructuración de los datos en tablas y columnas con las cuales se llegan a representar entidades, atributos y relaciones. Sobre las tablas se establecen restricciones de integridad y de referencia. Para representar un modelo de datos relacional se utilizan los diagramas entidad-relación y en la comunidad de desarrollo es el modelo más extendido y utilizado.

Las bases de datos no relacionales, se caracterizan principalmente por organizar los datos en estructuras no definidas. Los datos no se almacenan en tablas por lo tanto su estructura no está basada en filas y columnas, están orientadas a la gestión de grandes volúmenes de datos. El modelado no relacional estructura los datos en esquemas de información, el cual es una especificación textual con el cual se describe la estructura y los datos que se almacenan, normalmente tiene la forma jerárquica.

Uno de los objetivos de la etapa de modelado de datos es establecer el tipo de base de datos a utilizar para cada proyecto de software. En la actualidad, las bases de datos no relacionales son más populares que las relacionales lo que permite considerar, al momento de diseño, si para un proyecto de software es necesario modelar los datos con la rigidez o flexibilidad en el manejo de su estructura.

La documentación generada de la etapa de modelado de datos se debe almacenar y versionar en la carpeta *docs/* en la raíz del proyecto de software, mismos que se debe realizar conforme la técnica diseño utilizada.

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

10.3 Modelado de Roles y Permisos

El modelado de roles y permisos tiene el propósito de establecer los privilegios que se van a otorgar a los usuarios para manipular la información y utilizar las funcionalidades de aplicaciones, sistemas y/o servicios. Los permisos que se otorgan a los usuarios pueden ser de acceso (para lectura, actualización o eliminación) o de ejecución (para opciones y funcionalidades) de los sistemas, aplicaciones y/o servicios.

Conceptualmente, un rol agrupa un conjunto de permisos los cuales son asignados a usuarios específicos lo que le otorga a cada uno la capacidad de completar tareas con el apoyo de las funcionalidades de los sistemas y/o aplicaciones.

El modelado de roles y permisos tiene el objetivo de establecer la responsabilidad de trabajo para los usuarios dentro los sistemas. Existen muchas técnicas de modelamiento de roles y permisos, uno de ellos es a través de la adecuación de las Matrices RACI/RASCI (*Responsible, Accountable, Support, Consulted and Informed*) con el cual en un solo arreglo se puede definir: las tareas que se ejecutan, los roles que están involucrados y en cada caso es posible asociar las tareas con sus responsables de realización, sus sustitutos en caso de ausencia y el lugar donde se realiza cada tarea.

En una Matriz RACI/RASCI las filas representan las tareas, las columnas representan los roles y las intersecciones entre tarea y roles permiten representar las responsabilidades, entregables, sustitutos y lugares donde los roles desarrollan sus tareas en los sistemas.

Normalmente, los entregables del Modelado de Roles y Permisos se registra en un esquema de tablas que involucra las siguientes entidades de información:

- sujetos (ROLES)
- objetos (APIs)
- acciones (VERBO)
- permisos (acceso, ejecución)

10.4 Desarrollo y Actualización

Conforme a las políticas definidas, todos los proyectos de software desarrollados o actualizados por la UPAT deben ser registrados, documentados y versionados. Se establece como Sistema de Control de Versiones y de registro documental el software GIT utilizando la herramienta GitLab.

Cada proyecto (sistema, aplicación y/o servicio) conlleva el desarrollo de componentes/artefactos en los siguientes ámbitos: presentación (*frontend*), servicios (*backend*) y base de datos (*data model*).

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

10.4.1 Creación de Grupos y Repositorios

El Jefe de Unidad o los Responsables de Áreas son los encargados de crear los grupos y repositorios en el Sistema de Control de Versiones. Además, éstos son también responsables de asignar los roles de participación a los dependientes designados en cada proyecto. La denominación de grupos y repositorios debe estar de acorde a los estándares **E002 y E003**.

10.4.2 Creación de Ramas Principales

master

Constituye la rama principal del proyecto y debe contener únicamente versiones del producto desarrollado/actualizado, con todas sus funcionalidades completamente operativas. Es importante que cada vez que se haga un pull/merge request a la rama master, esta tenga un tag de versión para restituir en caso de falla.

sandbox

Considerada la rama de pre release o preproducción, es la rama que se usará para desplegar y revisar la aplicación entre partes, la parte interesada (generalmente considerada el cliente) y el equipo de gestión de proyectos, el objetivo principal de esta rama es realizar las revisiones de todas las características a implementarse posteriormente en producción (mediante un pull/merge request a la rama master), en común acuerdo entre partes.

test

Pese a que el concepto de prueba continúa está implícito en todos las ramas y procesos de integración y despliegue continuo, la rama de test está directamente vinculada con los equipos de calidad de software y seguridad donde se realizan pruebas más especializadas para garantizar el correcto comportamiento de las funcionalidades en los ambientes de sandbox y producción.

develop

Esta rama es considerada de integración, en la cual los desarrolladores irán incorporando los cambios realizados por cada uno de ellos, de manera que se puedan verificar periódicamente conforme a un esquema de integración continua. Se puede considerar a esta rama de naturaleza inestable por la actividad recurrente que presenta entre cambios

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

10.4.3 Creación de Ramas Secundarias

feature & hotfix branches

Las feature branches son ramas que se utilizan para desarrollar características específicas en determinados sprints, los hotfix branches se utilizan para solucionar bugs críticos detectados en versiones de producción del producto y que necesitan ser solucionados urgentemente, sin existir planificación previa.

10.4.4 Aprobación Entre Ramas

El jefe de unidad podrá realizar el *merge request* a cualquiera de las ramas, para el caso de los responsables se detalla el siguiente flujo:

develop

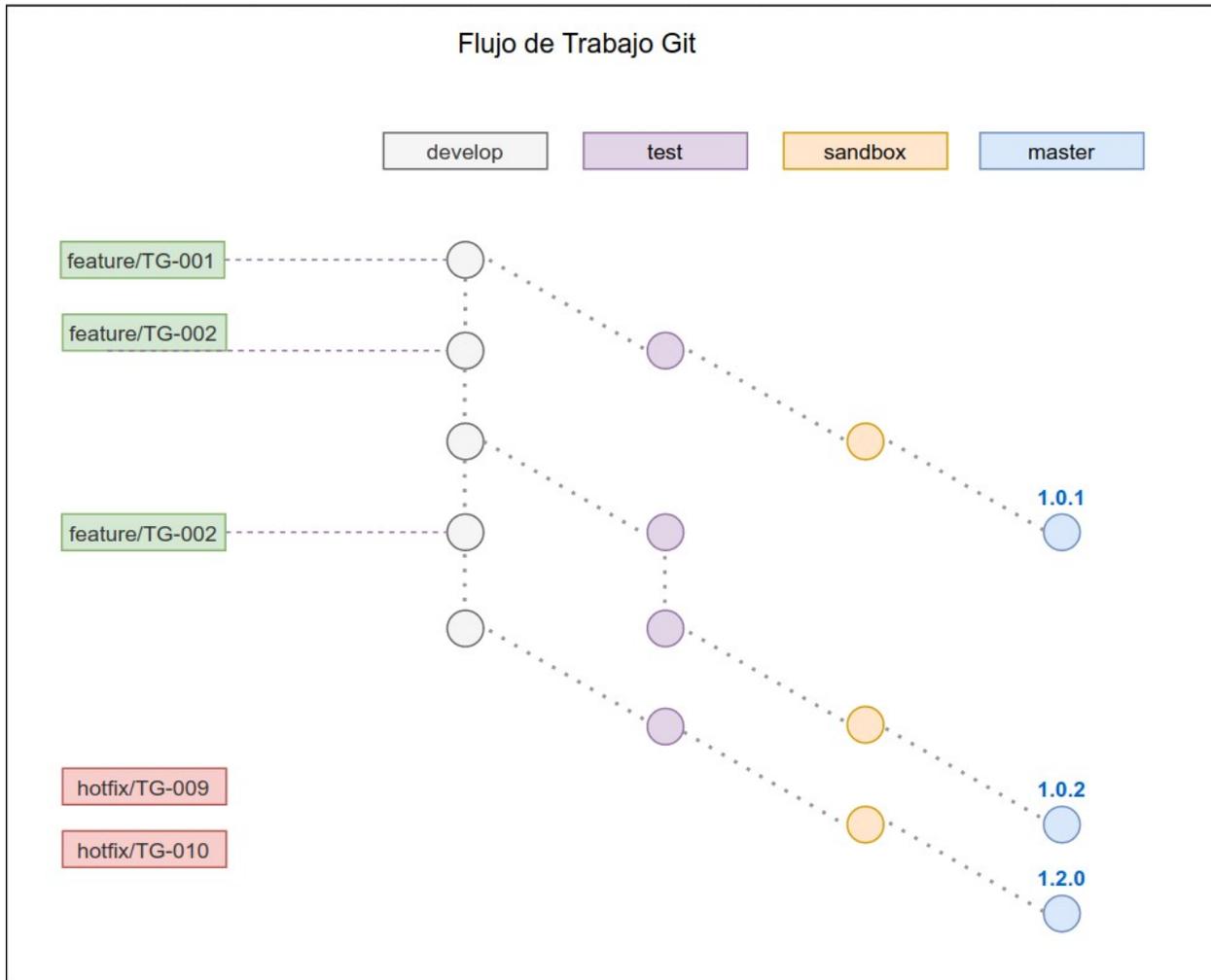
El responsable de desarrollo, realizará el etiquetado y hará un merge request a la rama de test y colocara el tag con la versión.

test, sandbox y master

El responsable de automatización y calidad hará el *merge request* sobre la versión entregada por desarrollo y cambiará entre las ramas de *test, sandbox y production*. En el caso de que se encuentren errores críticos que ameritan un cambio de versión o se determinen por el jefe de unidad y/o responsables que las pruebas y automatización no debe continuar entonces se desechara la versión actual.

10.4.5 Flujo de Versionamiento

Los proyectos a desarrollarse en la AGETIC deberán seguir el siguiente flujo de versionamientos o ramificaciones.



10.4.6 Configuración de Variables de Entorno

Para el desarrollo y/o actualización de un producto de software se debe contemplar información de configuración de manera que ésta información conduzca a que el sistema, aplicación y/o servicio funcione en cualquiera de los entornos de operación definidos.

Cada repositorio de software vinculado al frontend, backend o base de datos debe almacenar en su carpeta raíz el archivo de configuración *env.sample*. Este archivo debe contener, si es necesario, la definición de variables de entorno para la configuración de:

- Acceso a la base de datos
- Refresh token
- Dominios
- Correos

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

- Encoladores
- URLs y tokens de servicios de interoperabilidad
- Parámetros para integrar (p.e. Ciudadanía Digital, Aprobación de Documentos, etc.)
- Directorios donde se guardan archivos (Logs, Imágenes, PDFs, etc.)
- Permisos (acceso, ejecución)
- Archivos (Logs, Temp, etc.)

10.4.2 Seguridad

Para cada proyecto se debe contemplar aspectos de seguridad en los ámbitos de la capa de presentación (frontend), servicios (backend) y base de datos. La seguridad se debe orientar al resguardo de la información y al uso adecuado de las funcionalidades de una aplicación.

Seguridad en el ámbito del frontend

Se debe adoptar el modelo de autenticación para confirmar la identidad de los usuarios.

La base para habilitar accesos a la información y uso de funcionalidades de una aplicación es la definición de roles y permisos por roles.

Definición de política de CORS (cross origin resource security)

Seguridad en el ámbito del backend

La seguridad en el ámbito del backend se debe implementar en base a:

- Uso de token mediante JWT
- Definición de política de CORS (cross origin resource security)
- Verificación de permisos a recursos en el Middleware

Seguridad en el ámbito de la base de datos

Para resguardar los datos importantes y sensibles de cada proyecto se debe considerar el uso de método de encriptación en la base de datos. Siempre que sea posible y adecuado hacerlo, se debe "cifrar" los datos de modo que ni el personal de desarrollo pueda conocer el mismo. La prioridad es prevenir que usuarios no autorizados, tengan acceso a datos sensibles. Normalmente, las instituciones que patrocinan el desarrollo de proyectos de software hacen conocer sus requerimientos de seguridad de la información.

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

10.4.6 Documentación

En el directorio raíz del proyecto de software se debe registrar los siguientes documentos:

README.md

Contiene la descripción general del proyecto, la lista de los módulos con los que cuenta u otros aspectos del software. Debe contener los siguientes puntos:

- Descripción general del proyecto
- Documentación
- Lista de módulos o dependencias
- Pasos para configurar y ejecutar el software en entorno de desarrollo
- Pasos para configurar y ejecutar las pruebas automatizadas en entorno de pruebas
- Referencia al archivo `INSTALL.md` para el despliegue del software en entorno de producción

INSTALL.md

Describe los pasos y comandos detallados para la instalación y configuración del software. Es necesario que contenga mínimamente el siguiente contenido:

- Requerimientos de hardware mínimos
- Pasos para la instalación de los paquetes/programas necesarios (Ejemplos: node, npm, librerías, pm2, etc.)
- Pasos para clonar el proyecto desde el Sistema de control de versiones.
- Si el software lo requiere, se debe explicar la instalación de los siguientes elementos:
 - Instalación y configuración necesaria de la base de datos.
 - Comandos para Inicializar la Base de Datos (creación de tablas, migraciones, seeders y demás necesario si corresponde)
 - Describir la configuración necesaria de todos los archivos de configuración considerando el entorno en el que se desplegará el software
 - Configuración de datos de acceso a la base de datos
 - Configuración de acceso a los servicios de interoperabilidad que necesita el software
 - Demás variables de configuración
 - Configuración de correo.
- Comandos para iniciar la aplicación en los distintos entornos de desarrollo (development, test, y production)

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

UPDATE.md (Opcionalmente)

En caso de existir actualizaciones al software en producción, este documento describe los pasos y comandos detallados para la actualización. En este documento se debe indicar cómo realizar la actualización en el ambiente de producción, para esto se debe generar este documento con los pasos detallados para la actualización.

Estos documentos se deben elaborar en formato Markdown (.md) y describir con el mayor detalle posible los complementos, librerías, servicios de interoperabilidad y otros que sean necesarios, para que el software pueda ser desplegado con éxito.

Los nombres de estos archivos deben estar descritos en inglés para mantener una nomenclatura internacionalmente utilizada con el propósito de permitir la compatibilidad con proyectos internacionales.

CHANGELOG.md

Este documento muestra los cambios más importantes realizados en un periodo de tiempo. Es importante para facilitar a los usuarios y colaboradores ver exactamente qué cambios reseñables se han realizado entre cada versión del proyecto.

- Este documento se debe elaborar en formato Markdown.
- Cada registro debe estar ordenado cronológicamente
- La última versión va primero
- Debe mostrar la fecha de publicación de cada versión. Usar el formato 2022-03-14 por ser un estándar ISO y ordenar implícitamente por año, mes y día.
- Se debe agrupar y ordenar los mismos tipos de cambios según esta lista:
 - Añadido (Added) para funcionalidades nuevas.
 - Modificado (Changed) para los cambios en funcionalidades existentes.
 - Obsoleto (Deprecated) para indicar que una funcionalidad está obsoleta, se queda sin soporte y se eliminará en próximas versiones.
 - Eliminado (Removed) para las funcionalidades en desuso que se eliminaron en esta versión.
 - Corregido (Fixed) para corrección de errores.
 - Seguridad (Security) en caso de vulnerabilidades.

10.5 Pruebas

De acuerdo a las políticas de desarrollo y actualización cada proyecto de software debe contar con pruebas unitarias y/o de integración que se ejecutan de manera automática en el entorno de pruebas.

	MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE	
Código: UPAT/ADET-M01	Versión: 3	Aprobado: R.A. AGETIC/RA/0042/2022, de 26/05/2022

Las pruebas deben estar versionadas en una carpeta *test/* o *tests/* del código fuente en el repositorio oficial.

En el archivo README.md o INSTALL.md se debe indicar cómo deben ser ejecutadas las pruebas unitarias y/o de integración que se hayan implementado.

Para aquellos desarrollos que contemplen sitios web, blogs, o similares no es necesario realizar este tipo de pruebas.

Pruebas de funcionalidad

En conjunto con las Unidades Organizacionales solicitantes, se realizará las pruebas y validaciones de funcionalidad. En caso de existir observaciones se realizarán los ajustes correspondientes.

Pruebas de seguridad

A la conclusión del desarrollo se debe contemplar la realización de pruebas de seguridad mismas que deben ser coordinadas con el Área de Centro de Gestión de Incidentes Informáticos (ACGII). En caso de detectar vulnerabilidades en el software la UPAT debe subsanar las mismas.

10.6 Implementación

Comprende la coordinación con las Unidades Organizacionales responsables de la puesta en producción de los productos de software. Para tal caso, se debe contar con el documento de instalación (INSTALL.md) con los pasos detallados para el despliegue.

En el manual INSTALL.md de instalación se debe incluir una descripción gráfica de la arquitectura de implementación para el despliegue del software.

En el caso de requerimientos de actualización de software el documento *UPDATE.md* debe describir los pasos y comandos detallados para realizar la actualización. En este documento se debe indicar cómo realizar la actualización en el ambiente de producción, se debe generar este documento con los pasos detallados minuciosamente para la actualización.