

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 1 DE 48
		FECHA	VERSIÓN No. 10

MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE

{fiduprevisora)

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 2 DE 48
		FECHA	VERSIÓN No. 10

0.	LISTA DE VERSIONES .....	3
1.	OBJETO.....	5
2.	ALCANCE.....	5
3.	GLOSARIO.....	5
4.	MARCO NORMATIVO .....	10
5.	DOCUMENTOS RELACIONADOS .....	10
6.	DESCRIPCION DEL MODELO .....	11
7.	APLICABILIDAD .....	11
7.1.	CONSIDERACIONES DE APLICABILIDAD .....	11
7.2.	GENERALIDADES DE LA METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE	14
8.	RESPONSABLES .....	16
9.	FASES O ETAPAS DE LA METODOLOGÍA DE DESARROLLO.....	19
9.1.	SOLICITUD .....	20
9.2.	ANÁLISIS.....	21
9.3.	PLANIFICACIÓN .....	22
9.4.	DISEÑO.....	23
9.5.	CONSTRUCCIÓN .....	23
9.6.	PRUEBAS .....	24
9.7.	IMPLANTACIÓN .....	26
9.8.	ESTABILIZACIÓN.....	26
9.9.	CIERRE .....	27
10.	FLUJOGRAMA .....	29
11.	POLÍTICAS Y LINEAMIENTOS .....	30
11.1.	CONTROL DE CAMBIO .....	30
11.2.	PRIORIZACIÓN DE CASOS DE DESARROLLO .....	30
11.3.	PROCEDIMIENTOS Y PROCESOS OPERATIVOS .....	30
11.4.	ASEGURAMIENTO DE CALIDAD .....	30
11.5.	GARANTÍAS Y SANCIONES .....	31
11.5.1.	MÉTRICAS DE OBLIGATORIO CUMPLIMIENTO.....	32
11.5.2.	CUMPLIMIENTO DE LA METODOLOGÍA DE DESARROLLO Y DOCUMENTACIÓN.....	32
11.5.3.	ETAPAS.....	32
11.5.4.	INDICADOR DE OPORTUNIDAD GENERAL EN LA ATENCIÓN .....	32
11.5.5.	DESCUENTOS POR INCUMPLIMIENTOS.....	33
11.5.6.	CÁLCULO DE DESCUENTOS Y/O SANCIONES POR INCUMPLIMIENTO .....	34
11.6.	EXCEPCIONES.....	34
11.7.	EVALUACIÓN DE PROVEEDORES .....	35
12.	ESTÁNDARES DE PROGRAMACIÓN .....	36
12.1.	OBJETIVO.....	36
12.2.	REGLAS.....	36
12.3.	MANEJO DE DOCUMENTACIÓN. ....	46

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 3 DE 48
		FECHA	VERSIÓN No. 10

## 0. LISTA DE VERSIONES

VERSIÓN	FECHA	RAZON DE LA ACTUALIZACIÓN
1	Feb/2016	Documentación de la metodología de desarrollo de la Dirección de Software mediante anexo versión 1.1.0.
2	Nov/2008	Versión 2.4.0 del anexo “Metodología de Desarrollo” se incluye el mantenimiento de software en el numeral 6. “Metodología de Desarrollo”
3	Ene/2010	Versión 2.5.0 del anexo “Metodología de Desarrollo” se modifica el numeral 9. “Migración de Información” y se consolida por Migración – Esquema de datos diferentes, Migración – Esquema de datos iguales y Desmote de Sistema Anterior.
4	Abr/2012	Versión 2.6.0 del anexo “Metodología de Desarrollo” se incluye el numeral 13. “Modelo de atención para aplicativos cuyo mantenimiento es externo”.
5	Mar/2013	Versión 2.7.0 del anexo “Metodología de Desarrollo”
6	May/2013	Versión 2.8.0 del anexo “Metodología de Desarrollo”
7	Jul/2014	Versión 2.9.0 del anexo “Metodología de Desarrollo”
8	23/May/2016	Se incorpora el anexo denominado Metodología de Desarrollo del manual de políticas ML-GTE-02-001 Manual de Políticas para Gestión Tecnológica a estructura de un manual de políticas y se le asigna un código al documento ML-GTE-02-003.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 4 DE 48
		FECHA	VERSIÓN No. 10

VERSIÓN	FECHA	RAZON DE LA ACTUALIZACIÓN
9	09/07/2017	<p>Se adiciona control de cambio relacionado con requerimientos de documentación que deben ir adjuntos para realizar el paso a producción, <b>Paso 9:</b> Paso a Producción. Se incluye políticas (<i>b, c, g, m, n y p</i>) frente a los controles de cambios relacionados con los siguientes temas:</p> <ul style="list-style-type: none"> <li>- Documentación obligatoria de pruebas.</li> <li>- Indicar que casos requieren o no pasos a producción.</li> <li>- Autorización de funcionarios delegados para registrar casos de nuevos desarrollos.</li> <li>- Acuerdos de servicios.</li> <li>- Atención de casos relacionados con modificación de datos.</li> <li>- Cierre de casos por Presunción.</li> </ul> <p>Adicionalmente se elimina la descripción de los riesgos en el documento y se agrega al glosario la definición de Administrador de Fábrica y Especialista de Producto (analista desarrollador) y el numeral 5.8 “MODELO DE SERVICIO APLICACIONES HCM-ERP”. Por otra parte, se adiciona al glosario la definición de: administrador de fábrica, cierre presunción, directivo líder de producto y especialista de producto.</p>
10		<p>El proceso de gestión de requerimientos se define en el instructivo IN-GTE-02-009.</p> <p>Se establece un nuevo modelo de gestión de desarrollo y mantenimiento articulado con la función de mesa de servicios y basado en metodología ágil. Por tal motivo se cambia el nombre a Modelo operativo de Mantenimiento y Desarrollo de Software con el fin de establecer el marco bajo el cual deben operar las fábricas de software en Fiduprevisora S.A.</p>

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 5 DE 48
		FECHA	VERSIÓN No. 10

## 1. OBJETO

Disponer de un documento guía para la ejecución de las actividades relacionadas con el desarrollo y mantenimiento de software, para el manejo de los requerimientos realizados por Fiduprevisora, como parte del contrato establecido entre la Fábrica de Desarrollo y Fiduprevisora S.A., y que brinde los lineamientos para la implementación de mejores niveles de servicio, calidad y seguridad de la información, disponibilidad y mantenimiento de bases de datos de conocimiento e indicadores de gestión en busca de soluciones de desarrollo efectivas y aplicación eficaz de planes de mejoramiento.

## 2. ALCANCE

Este documento es aplicable a todo el personal involucrado en los servicios de tecnología que presta Fiduprevisora S.A. sin importar su tipo de vinculación contractual (planta, temporales, contratistas, etc.), para todos los aspectos relacionados con el desarrollo de software teniendo en cuenta las fases, actividades, actores involucrados desde el momento de la preconceptualización del software hasta su entrega.

El Modelo de desarrollo, cubre la gestión de las solicitudes de los usuarios correspondientes a mantenimientos y nuevos desarrollos de las aplicaciones (exceptuando los mantenimientos correctivos, ya que éstos son gestionados de acuerdo al procedimiento de gestión de incidentes de ITIL v3.0 y se encuentra en proceso migrar a v4.0).

La seguridad en el desarrollo y mantenimiento de las aplicaciones de la Fiduprevisora se define en el documento (ML-ESC-05-001) Manual de Políticas de Seguridad de la Información de Fiduprevisora basados en la norma ISO 27001:2013, y es apoyado en la lista de verificación de seguridad en el mantenimiento y desarrollo de software FR-ESC-05-053 los cuales se encuentran dispuestos en la herramienta de gestión documental ISOLUCION.

## 3. GLOSARIO

**Acuerdos de Niveles de Servicio (SLA/ANS):** Es un compromiso que se realiza entre el proveedor de servicios de Tecnología de la Información TI y un cliente. Un acuerdo de nivel de servicio describe los objetivos, la lista de servicios de TI, especifica las responsabilidades del proveedor de servicios de TI y el cliente y establece las métricas de rendimiento. Un acuerdo único puede cubrir múltiples servicios de TI o varios clientes.

**Acuerdo de Nivel de Servicios Operativos (OLA):** Es un compromiso entre el proveedor del servicio y otra parte de la misma organización para brindar el servicio a los usuarios; define las responsabilidades de ambas partes, así como las métricas para la medición de la prestación del servicio con una calidad preestablecida. Por ejemplo, podría haber un acuerdo de nivel operativo entre: El proveedor de servicios de TI y un departamento de compras para adquirir hardware. El Centro de Servicio y un grupo de soporte para proporcionar la resolución de incidentes en los tiempos acordados.

**Administrador de Fábrica:** Es el rol del responsable de servir como intermediario entre el usuario funcional del aplicativo y el consultor de la fábrica, de tal forma, que entiende el requerimiento del usuario y gestiona

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 6 DE 48
		FECHA	VERSIÓN No. 10

con la consultoría el levantamiento de información, desarrollo, pruebas, capacitación y puesta en producción de la solución. Es el encargado de controlar los avances de los incidentes y/o desarrollos, asignaciones de trabajo, reuniones de entendimiento y de seguimiento hasta llevar a buen término la ejecución de cualquier tipo de requerimiento de software.

**Ambiente Productivo:** Es el ambiente donde se realiza el procesamiento real de la información y donde finalmente se despliegan las aplicaciones.

**Analista desarrollador:** Profesional que conoce el funcionamiento del aplicativo y que es responsable de diseñar, desarrollar, documenta y probar el producto ofreciendo soluciones adaptadas a las necesidades de la organización.

**Analista de Función Mesa de Servicio:** Es la persona encargada de gestionar (recibir, analizar, categorizar, escalar y actualizar información de solicitudes) y brindar soluciones en primer contacto.

**Aranda:** Herramienta de gestión para el registro y seguimiento a las solicitudes (incidentes o requerimientos) registradas por los usuarios, y que son evaluadas y categorizadas por los Analistas de Mesa de Servicios.

**Branch (Ramificar):** Es una copia de un conjunto de objetos en un lugar diferente del repositorio de código fuente. Esta copia permite a dos o más equipos trabajar en paralelo en la misma parte de un proyecto.

**Caso de uso:** Documento requerido por la metodología de desarrollo de software, donde se explica una determinada Funcionalidad del desarrollo, siendo el documento de entendimiento del cliente y el desarrollador, siendo una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

**Cierre por Presunción:** Evento que se presenta cuando el administrador de la fábrica de software se ve en la necesidad de cerrar una solicitud al no contar con la aprobación de alcance funcional de la solución dada al caso o la aceptación de pruebas o producción por parte de los usuarios que registran los casos, dentro de los cinco (5) días hábiles siguientes a la notificación que se realiza por correo al funcionario responsable.

**Código fuente:** Es el texto que contiene las instrucciones del programa, escritas en el lenguaje de programación. Se trata de un archivo de texto legible que se puede copiar, modificar e imprimir sin dificultad.

**Control de versión:** Contiene los detalles de una Versión y documenta la historia de la misma desde las etapas iniciales de la planificación hasta su cierre.

**Definición de requerimiento funcional:** Se considera aquellos tiquetes o casos en los cuales no se tiene claridad en su alcance.

**Despliegue/Liberación (Deployment/Release):** Sinónimo de implementación. La mayoría de las veces se refiere a Implementaciones complejas o divididos en fases, o a Implementaciones en múltiples ubicaciones.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 7 DE 48
		FECHA	VERSIÓN No. 10

**Diccionario de datos:** Conjunto de definiciones que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

**Directivo líder del proceso:** Es el rol responsable en cada área de solicitar y autorizar de acuerdo con los procesos y procedimientos bajo su responsabilidad, el desarrollo de controles de cambios o nuevos desarrollos para automatizar actividades manuales y para colocar controles que mitiguen los riesgos. También tiene la responsabilidad en casos de incidentes de software que no se puedan solucionar de forma funcional por el aplicativo, de autorizar la actualización o corrección de datos directamente en base de datos.

**Gestión de incidentes:** Su objetivo es dar solución de la manera más rápida y eficaz a un incidente que cause anomalías o interrupciones en el servicio.

**Herramienta de control de versiones (Github 2.22.0):** Herramienta provista por la Fábrica de software para el registro y control de las versiones de código, además de toda la documentación de los desarrollos y mantenimientos de software.

**Herramienta de gestión de incidentes de pruebas (Mantis):** Herramienta provista por la Fábrica de software para el registro y gestión de los incidentes que se presenten en la fase de pruebas.

**Historia de Usuario:** Es la explicación breve que hace cada usuario, describiendo lo que requiere haga el sistema.

**Incidente:** Evento que se presenta en las pruebas y/o periodo de estabilización de la solicitud construida que no refleja el funcionamiento correcto del software.

**ITIL (Librería de integración de tecnología de información):** Conjunto de buenas prácticas que ayuda a conseguir una mayor calidad en los servicios. Ofrece una descripción detallada de los procesos más importantes en una organización de TI.

**Mantenimiento de aplicaciones:** Corresponden a las mejoras, adiciones o adecuaciones de aplicaciones existentes.

**Mantenimiento adaptativo o evolutivo:** Es el que permite adecuar o ajustar el software a un cambio normativo o de procedimiento.

**Mantenimiento perfectivo:** Es el orientado a mejorar el rendimiento (performance) u otras propiedades, entendiéndose características de los sistemas de información.

**Mantenimiento preventivo:** Es el orientado a modificar el software o aplicativo para mejorar sus propiedades en términos de calidad, mantenibilidad u otra característica del sistema de información.

**Mantenimiento correctivo:** Es el orientado a corregir defectos de funcionamiento del software.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 8 DE 48
		FECHA	VERSIÓN No. 10

**Manual de explotación:** Documento que tiene como objetivo proporcionar información básica de apoyo a los ingenieros de sistemas que requieran implantar nuevas funcionalidades dentro del mismo. Presenta de manera gráfica el modelo con el cual fue diseñada la base de datos, describiendo además la infraestructura tanto de sus tablas y procedimientos como de los datos que internamente procesa para el óptimo funcionamiento diario.

**Manual de usuario:** Documento que recibe el usuario junto con el sistema de composición, ya sea como referencia del hardware o software, que le permite aprender las particularidades del equipo o del sistema y la forma de utilizarlo.

**Metodología de desarrollo:** Es un marco de trabajo estructurado, iterativo e incremental por fases y actividades que permiten planificar y controlar el ciclo de vida del software mediante la colaboración de grupos auto-organizados y multidisciplinarios.

**Modelo entidad relación (MER):** Es un diagrama que permite representar las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades. Formalmente, los diagramas ER son un lenguaje gráfico para describir conceptos.

**Notas de Entrevistas:** Corresponden a comentarios adicionales que surgen luego de realizar las entrevistas para recolección de información.

**Nuevos desarrollos o proyectos:** Cuando el requerimiento de software adiciona una funcionalidad al aplicativo o si el requerimiento corresponde a una funcionalidad de un aplicativo, componente o módulo que no existe en el sistema de información de Fiduprevisora.

**Plan Detallado de Trabajo (PDT):** Documento detallado donde se incluyen actividades, recursos y tiempo invertido por cada actividad a realizar.

**Plan de pruebas:** Documento que tiene como propósito explícito el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas.

**Product backlog:** Corresponde a la lista de objetivos/requisitos priorizada, la cual representa la visión y expectativas del cliente respecto a los objetivos y entregas del producto o proyecto para una solicitud específica.

**Prototipo de interfaz gráfica:** Es un diagrama que especifica los elementos, controles gráficos y las relaciones que existen entre ellos dentro de un caso de uso que hace la especificación de un sistema de información.

**Pruebas de Caja Blanca:** Son pruebas que garantizan que todas las instrucciones del código se ejecuten por lo menos una vez, para garantizar que todas las partes funcionen correctamente.

**Pruebas de Caja Negra:** Son pruebas funcionales donde solo interesa que el programa funcione de acuerdo con los requerimientos previamente especificados.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 9 DE 48
		FECHA	VERSIÓN No. 10

**Pruebas de calidad:** Son las pruebas basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. Las pruebas de Calidad se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático. Estas pruebas son realizadas por el grupo de calidad de la fábrica de desarrollo para asegurar que la entrega del desarrollo al usuario se depura con el fin de minimizar la ocurrencia de errores que no fueron detectados en las pruebas unitarias.

**Pruebas de Defectos:** Son pruebas donde el profesional simula situaciones anormales. La idea es generar entradas que generen un comportamiento anormal y generar salidas defectuosas que confirmen las posibles fallas del sistema.

**Pruebas de Desempeño:** Son pruebas que garantizan que el sistema puede procesar la carga sugerida. Se debe manejar la cantidad de información necesaria que produzca la caída del sistema.

**Pruebas de Integración:** Inicialmente se hacen pruebas de cada parte del programa por separado. Las pruebas de integración consisten en probar todas las partes en conjunto.

**Pruebas de Interfaces de Parámetros:** Los pruebas donde se pasan datos o mensajes de un sistema a otro.

**Pruebas de seguridad:** Son pruebas realizadas por el equipo de seguridad de la fábrica de desarrollo y/o de la Fiduprevisora, con el fin de identificar las fallas de seguridad y/o vulnerabilidades del desarrollo.

**Pruebas de Trayectorias:** Consiste en identificar todas las posibles trayectorias de un programa y probarlas (las condicionales).

**Pruebas de usuario:** Son las pruebas realizadas por los usuarios (finales o funcionales de la subcuenta) de la Fiduprevisora o terceros funcionales a cargo, con el fin de validar que un sistema cumple con el funcionamiento esperado. Buscan la aceptación desde el punto de vista de su funcionalidad y rendimiento.

**Pruebas unitarias:** Son pruebas realizadas por el equipo de desarrolladores, realizadas con el fin de probar el correcto funcionamiento de un módulo de código. El propósito de estas pruebas es asegurar que cada uno de los módulos funcione correctamente por separado.

**Requerimientos:** Son solicitudes realizadas por los clientes en los cuales describen las necesidades que contiene la SRF. Inclusión, eliminación o modificación de un servicio o componente de un servicio soportado, planificado o autorizado, y su documentación asociada. p.e. gestión de usuarios, reportes y consultas, movimientos de equipos.

**Reuniones de entendimiento:** Espacio donde se reúne el Usuario Directo y Funcional con el Líder Técnico y Funcional, en donde cuyo objetivo es mejorar, modificar o adicionar alguna funcionalidad al sistema de información de Fiduprevisora.

**Revisiones técnicas formales (RTF):** Se deben principalmente a que el trabajo debe ser revisado; debido a que hay errores que son percibidos más fácilmente por otras personas que por los creadores.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 10 DE 48
		FECHA	VERSIÓN No. 10

**Software:** Componentes lógicos de la infraestructura informática que permiten la operación de componentes de hardware, como aplicaciones de negocio, programas antivirus, etc., es un recurso que facilita las labores de procesamiento de datos, es utilizado como herramienta de productividad y como ayuda en la realización de trabajos.

**SRF (Solicitud de Requerimiento Funcional):** Es el formato de solicitud donde el usuario (cliente) define y describe de forma clara, precisa y completa el comportamiento y funcionamiento específico del sistema requerido de acuerdo a sus necesidades.

**Usuario directo (Fiduprevisora):** Es la persona que se encuentra autorizada para registrar las necesidades ante la Mesa de Servicios a través de la Solicitud de Requerimiento Funcional o Product Backlog.

**Usuario funcional (Fiduprevisora):** El usuario es el individuo o la organización que utiliza directamente el producto del proyecto o los servicios de TI.

**Usuario:** Es la persona que utiliza los servicios de TI en el día a día. Los usuarios son distintos de los clientes, ya que algunos clientes no utilizan el servicio de TI directamente

#### 4. MARCO NORMATIVO

Norma Técnica Colombiana, **NTC-ISO 9001**, Sistemas de Gestión de la Calidad

Norma Técnica Colombiana, **NTC-ISO 14001**, Sistemas de Gestión Ambiental

Norma **ISO 27001** Tecnología de la Información. Técnicas de Seguridad. Sistema de Gestión de la Seguridad de la Información (SGSI)

Demás normas que lo adicionen, sustituyan y/o modifiquen.

#### 5. DOCUMENTOS RELACIONADOS

ML-GTE-02-001 - Manual de Políticas para Gestión Tecnológica

ML-ESC-05-001 - Manual de Políticas de Seguridad de la Información

MP-GTE-02-009 - Función mesa de servicios

IN-GTE-02-009 - Gestión de requerimientos

IN-GTE-02-011 Gestión de Cambios

IN-GTE-02-013 Gestión de Versiones y Despliegues

IN-GAD-01-006 Aplicación Clausula Penal Pecuniaria

FR-GTE-01-002 - Formato de Solicitud de Requerimientos

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 11 DE 48
		FECHA	VERSIÓN No. 10

FR-GTE-01-001 - Formato de pruebas

FR-ESC-05-053 - Lista de verificación de seguridad en el mantenimiento y desarrollo de software

## 6. DESCRIPCIÓN DEL MODELO

El modelo que se presenta a lo largo de este documento se enfoca en “La implementación de requerimientos de software”, aplicado a la recepción y atención de dos (2) tipos de solicitudes:

- a) **MANTENIMIENTOS:** Aplica para requerimientos de modificación de software preexistente en los Sistemas de Información de Fiduprevisora, estos mantenimientos incluyen:
  - **Mantenimiento perfectivo:** Es el orientado a mejorar el rendimiento (performance), seguridad u otras propiedades, entiéndase características de los sistemas de información.
  - **Mantenimiento preventivo:** Es el orientado a modificar el software o aplicativo para mejorar sus propiedades en términos de calidad, mantenibilidad, seguridad u otra característica del sistema de información.
  - **Mantenimiento evolutivo o adaptativo:** Es el que permite adecuar o ajustar el software a un cambio normativo o de procedimiento.
  - **Mantenimiento correctivo:** Es el orientado a corregir defectos de funcionamiento y seguridad del software. Este tipo de mantenimiento se reporta a través de la herramienta de “Gestión de Incidentes” para lo cual el líder del proceso asigna al equipo de soporte. Este tipo de mantenimiento se atenderá bajo el proceso de Gestión de Incidentes ITIL.
- b) **NUEVOS DESARROLLOS O PROYECTOS:** Cuando el requerimiento de software adiciona una funcionalidad al aplicativo o si el requerimiento corresponde a una funcionalidad, componente o módulo que no existe en aplicativo o sistema de información de Fiduprevisora.

## 7. APLICABILIDAD

El modelo operativo de desarrollo y mantenimiento de software define las etapas para la Metodología Ágil.

### 7.1. CONSIDERACIONES DE APLICABILIDAD

- Para la realización de todo nuevo desarrollo o mantenimiento de software a excepción de mantenimientos correctivos, se debe hacer la solicitud del servicio a Fiduprevisora, quien realizará la gestión necesaria para iniciar el ciclo de desarrollo basado en la metodología acordada y aprobada.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 12 DE 48
		FECHA	VERSIÓN No. 10

- No se dará inicio a cualquier etapa de la Metodología de desarrollo y mantenimiento de software, hasta tanto no se cumplan los requisitos mínimos definidos en la solicitud de requerimiento funcional (SRF) o en el Product Backlog de solicitud (metodología Ágil), validados y aprobados por el comité de requerimientos.
- La Gerencia de Tecnología, es el único canal o interlocutor entre el equipo de la Fábrica de Desarrollo y Usuario Funcional.
- Es necesario que la Dirección de Software, cuente con la aprobación de los Product Backlog de solicitud o SRF.
- La estimación se basa en la precisión y completitud de la Solicitud de Requerimiento Funcional (SRF) o Product Backlog de solicitud. Cualquier cambio de alcance en el proyecto, en particular en los casos de uso, si se altera la funcionalidad o complejidad impactará el cronograma. Dichos cambios serán incluidos en el control de cambios y estos serán recibidos hasta la fase de construcción (máximo 40% de avance de la fase) y deberán entrar en fase de priorización.
- La Fábrica de Desarrollo utilizará técnicas o métricas de estimación de tiempo para aquellas tareas como acompañamiento, documentación, formación, requerimientos no funcionales, investigación o innovación tecnológica, pruebas de concepto y/o seguridad, análisis de datos, ambientación, creación de scripts y controles de seguridad para desarrollo. Como técnicas de estimación se contemplarán: Puntos de Función, juicio de expertos (el cual no puede basarse en más del 60% del juicio de expertos) y las demás que se consideren necesarias. Estas técnicas o métricas de estimación tendrán como fin la definición de los planes de trabajo (PDT).
- El tiempo de elaboración o actualización de casos de uso es aplicable únicamente a la actividad del analista de requerimientos, no incluye tiempos de revisión y aprobación; estas actividades deberán acordarse individualmente con el usuario dependiendo del número de casos de uso que resulten de cada requerimiento de software, realizando un cronograma de reuniones de revisión y aprobación del plan de trabajo. Por otra parte, es importante garantizar la disponibilidad de tiempo del Usuario Funcional.
- La arquitectura es un proceso transversal, que se da a lo largo de la metodología de desarrollo y mantenimiento de software, para que el producto se alinee con los objetivos del negocio.
- Los tiempos de aprobación de las solicitudes (SRF, Product Backlog) y Plan de Trabajo en las etapas establecidas son responsabilidad de Fiduprevisora.
- En la metodología no se tendrán equipos de desarrollo por cada iniciativa (tiquete o caso), sino por especialidades o disciplinas. Por ejemplo: Equipo de Requerimientos, Diseño, Desarrollo, y/o de Pruebas.
- En aquellos casos donde el desarrollo sea suspendido o anulado, se elaborará el informe de cierre incluyendo el motivo por el cual la solicitud fue cancelada o suspendida y se dará cierre al tiquete o caso. Por consiguiente, la fábrica podrá facturar hasta la fase en la cual se encuentre el caso.

{fiduprevisora}	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 13 DE 48
		FECHA	VERSIÓN No. 10

- Los desarrollos, nuevas funcionalidades y controles de cambio por parte de la Fábrica de Desarrollo deben contar con un módulo de administración funcional, y que a su vez asigne y capacite el Usuario Funcional idóneo para su manejo y gestión.
- La Fábrica de Desarrollo debe monitorear permanentemente los requerimientos de sistemas de información asignado a su objeto contractual, con el fin de su atención oportuna y soluciones eficaces para la mesa de servicios y/o desarrollos asignados.
- La Fábrica de Desarrollo es responsable de informar a Fiduprevisora mediante la vigencia del contrato las oportunidades de mejora; investigar, analizar y diseñar cuando se requieran nuevas tecnologías, herramientas, metodologías y opciones que permitan optimizar el funcionamiento, seguridad y desempeño de los aplicativos y sistemas de información de Fiduprevisora acordes al cumplimiento del objeto del contrato o aplicaciones asociados a dicho funcionamiento y en general la infraestructura lógica de soporte tecnológico de Fiduprevisora como parte integral del cumplimiento del contrato y sin generar pago adicional o consumo de horas del contrato vigente, con una periodicidad mínima de 2 veces por año de contrato (de acuerdo a la calidad de este entregable se tendrá en cuenta para la calificación del proveedor).
- La Fábrica de Desarrollo revisará y evaluará el impacto cuando se generen nuevos proyectos de software en los sistemas existentes y en general en la infraestructura física y lógica de la Entidad, para cumplir los requerimientos de los funcionarios sobre las aplicaciones y determinar su viabilidad para todos los cambios del sistema de información que enmarca el objeto del contrato.
- La Fábrica de Desarrollo deberá cumplir permanentemente los Acuerdos de Niveles de Servicio (ANS) establecidos para la atención de requerimientos de tecnología y pactados con el contrato en el marco del manual operativo.
- La Fábrica de Desarrollo gestionará el desarrollo de módulos de registro de actividades (logs) dentro de los aplicativos de acuerdo con los requerimientos funcionales y normativos que se requieran por nuevas funcionalidades o control en el sistema de información de Fiduprevisora frente al marco del objeto del contrato.
- La Fábrica de Desarrollo coordinará y velará por la elaboración y actualización de manuales, documentación técnica y de Usuario Funcional de los sistemas de información de Fiduprevisora, para que estos se documenten adecuadamente. A su vez, velar y aplicar las mejores prácticas de seguridad de la información de la industria, aplicadas a la tecnología informática según las políticas de seguridad de la información (ML-ESC-05-001) Manual de Políticas de Seguridad.
- La metodología de software de la Fábrica de Desarrollo debe estar alineada a la política de software seguro vigente de la entidad (ML-ESC-05-001) Manual de Políticas de Seguridad de la Información, todo desarrollo debe plantear la protección y mitigación de vulnerabilidad del código y permitir siempre la autenticidad y seguridad de los desarrollos relacionando a los roles y perfiles que debe estar asociado siempre a una nueva funcionalidad o ajustado al control cambio solicitado por la entidad.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 14 DE 48
		FECHA	VERSIÓN No. 10

- La fábrica de software debe definir e incorporar las pruebas que certifiquen el cumplimiento de las condiciones de seguridad del desarrollo, validadas conjuntamente con la fiduciaria y basadas en estándares, guías, frameworks, técnicas, metodologías, etc. reconocidos por la industria.

## 7.2. GENERALIDADES DE LA METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE

El modelo de desarrollo y mantenimiento de software incorpora los procesos que intervienen en el ciclo de vida de los requerimientos registrados por parte de los usuarios de Fiduprevisora en la Gestión de Tecnologías de la Información, con el objeto de controlar y asegurar la calidad del software en el proceso.

El modelo de desarrollo y mantenimiento de software que se presenta a lo largo de este documento se enfoca en el “desarrollo de requerimientos”, aplicado a la recepción y atención de las siguientes fases (cada vez que apliquen dependiendo de la decisión de metodología a aplicar):

- **SOLICITUD:** Se basa en que el Usuario Directo documentará en la Solicitud de Requerimiento Funcional (SRF) o Product Backlog las necesidades para mejorar, modificar, ajustar o adicionar alguna funcionalidad de los sistemas de información de Fiduprevisora, posteriormente registrará en la herramienta de gestión de solicitudes la respectiva necesidad. La Dirección de Software de Fiduprevisora liderará un Comité de requerimientos interno en el que se definirá si la solicitud registrada se aprueba y asigna para desarrollo.
- **ANÁLISIS:** El objetivo de la presente fase es detallar la solicitud de requerimiento funcional realizada por el Usuario Directo, con respecto a la situación actual Fiduprevisora, estudiando aspectos relacionados con la operatividad del negocio, estrategias, sistemas, involucrados dentro del requerimiento.
- **PLANIFICACIÓN:** El objetivo de esta fase es realizar la elaboración del cronograma, la planeación y organización del tiquete o caso, que conlleva a la validación de los recursos necesarios (humanos, tecnológicos y logísticos) proporcionando los medios que aseguren el éxito del mismo: métricas, informes, herramientas. Durante esta fase se determina el tipo de Metodología Ágil a aplicar.
- **DISEÑO:** El objetivo de esta fase es realizar el diseño de los componentes y módulos que van a ser desarrollados para satisfacer los requerimientos, para lo cual se debe establecer la arquitectura y diseño técnico teniendo en cuenta la infraestructura tecnológica, conforme al ciclo de vida de la Metodología Ágil en la etapa de planificación. Así mismo se debe considerar los requisitos de seguridad que sean solicitados por parte de la dirección de seguridad de la fiduciaria y proponer las medidas adicionales que en este sentido se encuentren identificadas como mejores prácticas en la industria.
- **CONSTRUCCIÓN:** El objetivo es implementar la solución de acuerdo con los requerimientos de software siguiendo las especificaciones generadas en el diseño funcional y no funcional del tiquete o caso.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 15 DE 48
		FECHA	VERSIÓN No. 10

- **PRUEBAS:** El objetivo es verificar el funcionamiento del software que satisfaga los requerimientos del Usuario Funcional, evaluando y valorando la calidad del producto de software, generado en la etapa de construcción. La fase de pruebas, es un proceso iterativo donde se ejecuta la prueba, se notifican incidencias de falla para su corrección y se liberan nuevas versiones de software en el ambiente de pruebas, así, el ciclo se repite hasta obtener el software con la calidad y funcionamiento apropiado para disponerlo en ambiente de producción con la aceptación formal del sistema por parte del Usuario Funcional.

- ✓ **PRUEBAS DE CALIDAD (Fábrica):** Son las pruebas basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. Las pruebas de Calidad se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático. Estas pruebas son realizadas por el grupo de calidad para asegurar que la entrega del desarrollo al usuario se depura con el fin de minimizar errores que no fueron detectados en las pruebas unitarias.

Estas pruebas incluyen las pruebas de:

- **Rendimiento:** Se centran en determinar la velocidad con la que el sistema bajo pruebas realiza una tarea en las condiciones particulares del escenario de pruebas.
- **Carga:** Se utilizan para verificar el comportamiento de un producto, servicio o aplicación bajo una cantidad de peticiones esperada. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dure cierta actividad.
- **Estrés:** Son pruebas de carga o rendimiento, pero superando los límites esperados y/o determinados en las pruebas, con el propósito de encontrar el volumen de datos o de tiempo en que la aplicación comienza a fallar o es incapaz de responder a las peticiones.
- **Usabilidad:** Su objetivo es validar la facilidad de uso de un producto, servicio o de una aplicación en un entorno real con usuarios reales.
- **Fiabilidad o Portabilidad:** Se utilizan para verificar la tolerancia de fallos, capacidad de recuperación, capacidad de ser reemplazado, capacidad de adaptación e instalabilidad.

Por tanto, se centran en características del software que establecen *“cómo trabaja el sistema”*.

- ✓ **PRUEBAS DE SEGURIDAD Y HACKING ÉTICO:** Son pruebas enfocadas a detectar las vulnerabilidades y o fallas de seguridad del desarrollo. Teniendo en cuenta las políticas de seguridad de la Información de Fiduprevisora se ejecutan pruebas de hacking ético a lo largo del año, por consiguiente, las vulnerabilidades que se detecten durante dichas pruebas deberán ser atendidas y corregidas por la Fábrica de Desarrollo sin generar cobro alguno y se atenderán como garantía del software.
- ✓ **PRUEBAS UNITARIAS (Fábrica):** Son pruebas realizadas por el equipo de desarrolladores, con el fin de probar el correcto funcionamiento de un módulo de código. El propósito de estas pruebas es asegurar que cada uno de los módulos funcione correctamente por separado.

{fiduprevisora}	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 16 DE 48
		FECHA	VERSIÓN No. 10

✓ **PRUEBAS DE USUARIO:** Son las pruebas realizadas por los usuarios (Directos o Funcionales) de Fiduprevisora o terceros funcionales a cargo, con el fin de validar que un sistema cumple con el funcionamiento esperado. Buscan la aceptación desde el punto de vista de su funcionalidad y rendimiento.

- **IMPLANTACIÓN:** Hace referencia a la estrategia de puesta en producción del mantenimiento y/o desarrollo de software, en el ambiente productivo.
- **ESTABILIZACIÓN:** Hace referencia a un periodo de tiempo después de la puesta en producción en la cual el Usuario Funcional verifica el correcto funcionamiento de acuerdo a la solución brindada y a lo establecido en la solicitud de requerimiento funcional.
- **CIERRE:** Hace referencia al cierre general del ticket o caso que conforma el desarrollo de software, el objetivo de esta fase es formalizar la entrega del software a producción junto con los artefactos definidos y a satisfacción del Usuario Directo y Funcional de Fiduprevisora.

## 8. RESPONSABLES

- **Usuario Directo (Fiduprevisora)**
  - ✓ Crear la solicitud de definición de requerimiento funcional y asistir a las reuniones de iniciativas para la elaboración o ajuste del requerimiento funcional.
  - ✓ Elaborar la Solicitud de Requerimiento Funcional (SRF o Product Backlog). La responsabilidad del diligenciamiento del SRF o Product Backlog correspondientes es del usuario de Fiduprevisora. Cuando la iniciativa afecta transversalmente el negocio o proviene de líneas de negocio que no están a cargo Fiduprevisora o parten de cambios normativos o de Ley, la SRF o Product Backlog la debe diligenciar cada uno de los actores involucrados.
  - ✓ Registrar el ticket o caso correspondiente al requerimiento funcional (SRF o Product Backlog) creado ante la mesa de servicios.
- **Director de Software (Fiduprevisora)**
  - ✓ Funcionario que coordina, supervisa y vigila el servicio que presta la Fábrica de Desarrollo a Fiduprevisora. Participa en el aseguramiento de las tareas a cargo del personal de Fiduprevisora.
- **Gerente de Proyecto (Fábrica de Desarrollo)**
  - ✓ Gerente designado por la Fábrica de Desarrollo para liderar y gestionar el servicio prestado a Fiduprevisora.
  - ✓ Funcionario designado para gestionar el buen funcionamiento la Fábrica de Desarrollo ante Fiduprevisora. Encargado de coordinar las iniciativas de su área y los proyectos de software en su generalidad.
  - ✓ Funcionario designado para ejecutar las labores de preventa y estimaciones ante Fiduprevisora cuando sea requerido, sin que esto genere costos adicionales o sobrecostos a la bolsa de horas.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 17 DE 48
		FECHA	VERSIÓN No. 10

- **Comité Asesor de Cambios (CAB - Change Advisory Board)**

- ✓ Grupo de personas que aconseja al Gestor de Cambios en la evaluación, gestiona el establecimiento de prioridades y programación de los cambios. Se llevarán al CAB las solicitudes de paso a producción de los nuevos desarrollos y mantenimientos de software, siendo éste el organismo para la aprobación formal del paso a producción.

- **Comité de cambios de emergencia (ECAB - Emergency Change Advisory Board)**

- ✓ Grupo de personas que son convocadas cuando el CAB requiere tomar decisiones de emergencia que están fuera de la programación. Se llevarán al ECAB las solicitudes de paso a producción de los controles de cambio y demás desarrollos y mantenimientos de software manejados bajo metodología ágil, siendo éste el organismo para la aprobación formal del paso a producción de los mismos.

- **Líderes / Especialistas Técnicos (Fábrica de Desarrollo)**

- ✓ Funcionarios asignados por el Gerente de Proyecto con el fin de gestionar, informar y garantizar el cumplimiento de los resultados esperados de las solicitudes inscritas en la Mesa de Servicio. Es responsable de las tareas de gestión y técnicas de los tiquetes a su cargo y de llevar el Plan de Trabajo del tiquete.
- ✓ Mantiene actualizado el porcentaje de avance del Plan de Trabajo en la Herramienta para control de Planes de Trabajo.
- ✓ Adicionalmente actualiza la documentación relacionada con: manual de explotación, diccionario de datos, modelo entidad-relación, en el repositorio documental.

- **Líder Funcional (Fiduprevisora)**

- ✓ Líder responsable de los procesos operados por los usuarios funcionales en los Sistemas de Información de Fiduprevisora.
- ✓ Es la persona encargada de generar los casos de prueba de usuario para su utilización en las pruebas de usuario.
- ✓ Realizar las pruebas de usuario, validar y recibir el producto final.
- ✓ Reportar las incidencias de las pruebas realizadas en la herramienta Mantis de registro de incidentes de pruebas y de estabilización suministrada por la Fábrica de software.
- ✓ Responsable de elaborar y socializar los procesos a los usuarios funcionales después de la entrega del software en producción.

- **Líder Técnico (Fiduprevisora)**

- ✓ Líder responsable de los procesos de los Sistemas de Información de Fiduprevisora, es el encargado de aprobar las solicitudes de requerimientos y análisis funcional que permitan el desarrollo de los proyectos de software.
- ✓ Apoya metodológicamente la reunión de iniciativa y la elaboración de la (SRF o Product Backlog) y asegura que ésta se encuentre correctamente detallada.

{fiduprevisora}	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 18 DE 48
		FECHA	VERSIÓN No. 10

- ✓ Revisa y aprueba la SRF o Product Backlog de acuerdo a las necesidades para dar continuidad al proceso.
- ✓ Coordina y asegura la ejecución de la reunión de entendimiento.
- ✓ Realiza las actividades de empalme cuando hay cambio de fábrica de software.
- ✓ Aprueba la estimación del desarrollo.

- **Consultor/Analista Negocios y Funcional (Fábrica de Desarrollo)**

- ✓ Persona encargada de realizar un análisis detallado y funcional de los procesos del sistema basado en los lineamientos y estándares establecidos.
- ✓ Actualiza la documentación relacionada con: Listado de Casos de Uso, Casos de Uso y Actas de Reunión de Entendimiento en el repositorio documental.
- ✓ Responsable de definir el apoyo metodológico para realizar un análisis detallado de los procesos de negocio. Coordina el apoyo técnico y metodológico para la reunión de iniciativas y por ende para la creación del requerimiento funcional (en ocasiones de manera conjunta con el líder / especialista técnico). Solicita reunión de entendimiento cuando la SRF o Product Backlog lo requiera.

- **Arquitecto de Software (Fábrica de Desarrollo)**

- ✓ Persona asignada para realizar el diseño de arquitectura de los diferentes requerimientos de software, alineadas al modelo general de los componentes de los Sistemas de Información de Fiduprevisora. Asegura la aplicación de estándares y buenas prácticas de codificación.
- ✓ Realiza un diseño que cumpla con los estándares de desarrollo seguro con el fin de eliminar cualquier vulnerabilidad.
- ✓ Actualiza el documento de arquitectura, modelos y diagramas en el repositorio documental.
- ✓ Persona asignada para realizar los modelos y diagramas necesarios donde se especifica detalladamente la interacción de los objetos del negocio. Elabora el modelado de datos, la especificación de interfaces, adopta patrones de diseño adecuados con la arquitectura base, entre otras actividades, lo cual sirve de insumo para la fase de construcción.
- ✓ Actualiza la documentación relacionada con el diseño de la aplicación, en el repositorio documental.
- ✓ Propone acciones de mejora y actualización a la infraestructura lógica a los sistemas de información de la entidad de los que tiene acceso.

- **Analista de Calidad de Software (Fábrica de Desarrollo)**

- ✓ Persona asignada que realiza el aseguramiento del proceso y del producto, mediante verificaciones del debido proceso del proyecto, ejecuciones de pruebas estáticas de documentación y pruebas de calidad sobre el software y la documentación.
- ✓ Garantiza el aseguramiento de calidad de productos de software mediante la aplicación de la Metodología de Desarrollo, la ejecución de los procesos de calidad de software de acuerdo con el set de pruebas y estándares en los diferentes proyectos.
- ✓ Es el responsable de asegurar la realización de todas las pruebas del sistema y su documentación.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 19 DE 48
		FECHA	VERSIÓN No. 10

- ✓ Realizar acompañamiento al Usuario Funcional en el diseño de los planes de pruebas y de los casos de pruebas.
- ✓ Es responsable de asegurar y revisar la realización oportuna de las pruebas de usuario y conseguir las actas de aceptación de los mismos, antes de su paso a producción.
- ✓ Actualiza el manual de usuario en el repositorio documental; adicionalmente publica en este repositorio la información de plan de pruebas, diseños de pruebas, ejecuciones de pruebas y evidencias de pruebas.
- ✓ Reporta las incidencias de las pruebas realizadas en la herramienta Mantis de registro de incidentes de pruebas.

- **Oficial de Seguridad de la Información (Fiduprevisora)**

- ✓ Líder asignado por Fiduprevisora para verificar la correcta alineación del proyecto cumpliendo con la Política de Seguridad de la Información y los requerimientos de seguridad establecidos.
- ✓ Persona asignada para controlar la implementación de los lineamientos de la Política de Seguridad de la información que apliquen para los desarrollos de software.

- **Consultor/Ingeniero de Desarrollo (Fábrica de Desarrollo)**

- ✓ Persona asignada para analizar, implementar, estructurar y elaborar la construcción del proyecto siguiendo los estándares de la metodología de desarrollo de software. El desarrollador toma como insumo lo generado por el Diseñador y sigue con los lineamientos y documentación del Arquitecto. Así mismo, elabora o refina los manuales de explotación de los sistemas, elabora guías de despliegue, ejecuta y evidencia las pruebas unitarias, lista de chequeo de desarrollo, entre otras actividades.

Dentro de las actividades a realizar se encuentra:

- Implementación de los elementos de diseño.
- Estructura el modelo de implementación.
- Realiza o actualiza la codificación de los componentes.
- Realiza integraciones de sistemas - subsistemas.
- Realiza pruebas unitarias.
- Elabora o Refina los Manuales de explotación de los sistemas.
- Elabora guías de despliegue.
- Elabora evidencias de pruebas unitarias, lista de chequeo de desarrollo y la publica en el repositorio documental.
- Publica en la herramienta para control de versiones Github 2.22.0 los objetos propios del desarrollo realizado y actualiza la versión del mismo.

## 9. FASES O ETAPAS DE LA METODOLOGÍA DE DESARROLLO

La metodología de desarrollo a ejecutar en los distintos mantenimientos y nuevos desarrollos de software debe tener en cuenta:

{fiduprevisora}	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 20 DE 48
		FECHA	VERSIÓN No. 10

1. Número de Usuarios impactados por el desarrollo/mantenimiento: (Fiduprevisora; Usuarios externos).
2. Las entregas parciales y anticipadas que se requiere en el desarrollo.
3. Las pruebas especializadas o adicionales a las contempladas metodológicamente.
4. La definición de los requerimientos asociados que permitan tener una única especificación sin cambios a futuro.

### 9.1. SOLICITUD

Esta etapa se enfoca en la comprensión y análisis de la iniciativa (aquellos tiquetes o casos en los que aún no se tiene claridad de su alcance) o necesidad por parte de Fiduprevisora, con el apoyo del Líder Técnico de la Gerencia de Tecnología.

Dado el caso puntual que se necesite acompañamiento Técnico por parte de la Fábrica de Desarrollo, debe ser solicitado a la Gerencia del Proyecto con su respectiva justificación y este tiempo será incluido dentro la etapa de Planificación.

- ✓ **Identificar solicitud de Requerimiento:** Surge como consecuencia de una nueva Ley, Decreto, resolución, reforma, obligación, mejora, que origina un mantenimiento o nuevo desarrollo de software y que satisfaga dicha necesidad.
- ✓ **Documentar la Necesidad:** Una vez definido el alcance del Requerimiento, el Usuario Directo documentará la Solicitud de Requerimiento Funcional (SRF) o Product Backlog,
- ✓ **Registrar la Solicitud de Requerimiento Funcional o Product Backlog:** Esta se registra en la herramienta de gestión con la categoría de Mantenimiento o Nuevo Desarrollo con la respectiva solicitud de requerimiento funcional (SRF) o Product Backlog, para así ser programadas reuniones lideradas por el Líder Técnico de Fiduprevisora. Al registrar el caso debe contener como mínimo lo siguiente: Nombre del módulo del sistema de información, Nombre del Requerimiento, breve descripción de la solicitud (indicando el tipo de apoyo que requiere).
- ✓ **Reuniones definición de Requerimiento Funcional o Product Backlog:** El Usuario Directo y Funcional (Fiduprevisora) en compañía con el equipo funcional de la Dirección de Software analizan, validan, evalúan el alcance que tendrá el requerimiento para delimitar el ámbito de la solicitud. Bajo la gestión del equipo Funcional Fiduprevisora se convocan a las reuniones de entendimiento necesarias, resultados que serán recopilados en actas de reunión que serán diligenciadas por los consultores y/o analistas de negocio o funcionales de la Fábrica de Desarrollo.

Se llevarán a cabo las reuniones de entendimiento necesarias antes de dar inicio a la fase de planificación.

- ✓ **Entregables:**
  - ❖ Solicitud de Requerimiento Funcional SRF o Product backlog de solicitud (Aprobado)
  - ❖ Historias de Usuario en caso de ser Product backlog en el punto anterior.
  - ❖ Acta de Suspensión (si aplica)

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 21 DE 48
		FECHA	VERSIÓN No. 10

❖ Acta de Cancelación (si aplica)

## 9.2. ANÁLISIS

El objetivo de la presente fase es analizar en detalle la situación actual de Fiduprevisora respecto a la solicitud radicada, estudiando aspectos relacionados con la operatividad como estrategias, infraestructura tecnológica y sistemas, seguridad, entre otros, con el objetivo de llevar a cabo un diseño o rediseño de los procesos a los que va a ofrecer cobertura el nuevo cambio al Sistema y elaborar o ajustar el Diseño Conceptual del mismo.

- ✓ **Análisis de Requerimientos y solicitud de reuniones:** Primera identificación de la solicitud y requisitos especificados para el desarrollo de un Sistema de Gestión o Producto de software que cubra sus necesidades. Dichos requisitos serán tanto de carácter técnico como funcional. El Analista de Negocio de la Fábrica de Desarrollo, solicitará la reunión de entendimiento al Líder Técnico de Fiduprevisora.
- ✓ **Reuniones de entendimiento:** El Equipo Funcional de Fiduprevisora cita la reunión de entendimiento y asegura su realización. Un Analista de Calidad de la Fábrica de Desarrollo participará de las reuniones de entendimiento. De manera opcional o cuando sean requeridos participarán: El Líder/Especialista Técnico (Fábrica) y Arquitecto de la Fábrica de Desarrollo.
- ✓ **Modelado Conceptual:** Modelo en el que se reflejan los objetos que interactúan en el negocio. Se especifica en esta tarea, de forma inicial, cuál es la composición de cada objeto y qué relación existe entre cada objeto y el resto. Esto tiene especial importancia, debido a que una vez identificados los procesos, así como la información relativa a las entradas y salidas, será necesario identificar la relación existente entre los objetos y los procesos que los transforman.
- ✓ **Especificación de casos de uso:** Identificación más detallada de los requerimientos iniciales de Usuario Directo, valoración y refinamiento de los mismos con el objetivo de verificar que dichos requerimientos son nuevos, completos, consistentes y que permiten la trazabilidad. Así mismo se debe realizar el aseguramiento del documento de caso de Uso, en cuanto a: forma, consistencia, integralidad, ambigüedad y ortografía.
- ✓ **Definición de prototipo de Interfaz gráfica:** En esta actividad se definen, las interfaces gráficas, formularios e informes de la necesidad a desarrollar.

En cualquier caso, es fundamental la definición de los estándares a utilizar con anterioridad al diseño de las entradas y salidas. Por tanto, si los estándares existentes no se adaptan a las características del Sistema, se definirán unos particulares que se tendrán en cuenta al diseñar.

Una vez realizadas las anteriores actividades, el Líder Técnico de Fiduprevisora aprueba los casos de uso para así continuar con la estimación final o definitiva. De no ser aprobado, nuevamente se valida la especificación de los casos de uso.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 22 DE 48
		FECHA	VERSIÓN No. 10

- ✓ **Especificación de Planes y casos de Prueba:** Se diseñan los casos de prueba para comprobar que las operaciones estén de acuerdo con las especificaciones y que los componentes internos funcionan de manera adecuada.
- ✓ **Formalización Aprobación Estimación Final:** El Líder/Especialista Técnico (Fábrica) de la Fábrica de Desarrollo formaliza la aprobación de la estimación final mediante firma del Acta de aprobación con el Líder Técnico de Fiduprevisora.
- ✓ **Entregables:**
  - ❖ Listado de casos de Uso
  - ❖ Especificación de casos de uso
  - ❖ Planes y casos de prueba
  - ❖ Documento de línea base (Glosario, Requerimientos Funcionales, Requerimientos No funcionales y Visión del Sistema)
  - ❖ Aprobación o cierre de caso de uso.

**Nota:** El Usuario Funcional debe entregar el Plan de pruebas de usuario, desde el momento en que es aprobado la estimación final y máximo en el 40% de la fase de construcción.

### 9.3. PLANIFICACIÓN

El objetivo de esta fase es analizar los requerimientos iniciales del Usuario Directo, por parte del Equipo de la Fábrica de Desarrollo.

- ✓ **Elaboración del Cronograma del Tiquete o Caso:** Preparación de la propuesta para el desarrollo/implantación del sistema incluyendo la planificación de las distintas fases a realizar, composición y funciones de las figuras del proyecto y presupuesto para la realización del mismo, esto incluye una etapa de estimación cuyo método será por Puntos de Función, juicio de expertos (el cual no puede basarse en más del 60% del juicio de expertos) y las demás que se consideren necesarias. Antes de la entrega del cronograma del tiquete o caso a Fiduprevisora, se valida la metodología de acuerdo al resultado adquirido.
- ✓ **Organización del Proyecto:** Establecimiento del Equipo de desarrollo que participará en el Proyecto (que comprenderá tanto los integrantes en el equipo de trabajo como los externos). Así mismo, se determinarán las tareas asociadas a la ejecución del proyecto, planificación y costo de la ejecución; así como la asignación de recursos y la matriz de riesgos del proyecto detallando estándares a aplicar y productos a obtener (bien sean tangibles o no). Una vez finalice la organización del proyecto, El Líder/Especialista Técnico (Fábrica) ejecutará la fase de planificación, con el propósito de asegurar que las distintas tareas de la fase se hayan realizado correctamente, antes de continuar a la Fase de Diseño.
  - ❖ El Líder Técnico (Fiduprevisora) y Director de Software aprueban la estimación o planeación preliminar.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 23 DE 48
		FECHA	VERSIÓN No. 10

✓ **Entregables:**

- ❖ Estimación
- ❖ Plan de Trabajo (PDT) y/o Cronograma.
- ❖ Listado de Riesgos (Funcionales y no Funcionales)

#### 9.4. DISEÑO

El objetivo de esta etapa es validar, refinar y verificar la correcta alineación de los requisitos del proyecto con las necesidades del negocio, para así dar una directriz para la siguiente etapa de desarrollo alineado con la arquitectura.

- ✓ **Analizar Artefactos Requerimientos:** El Arquitecto, El Líder/Especialista Técnico (Fábrica) analizan los artefactos generados en las fases previas.
- ✓ **Aclarar dudas:** El analista de requerimientos explica y resuelve las dudas generadas por la Fábrica de Desarrollo.
- ✓ **Elaborar o ajustar documento de Arquitectura:** Especificar la arquitectura y elaborar el documento de acuerdo a las necesidades del negocio por parte del Arquitecto de la Fábrica.
- ✓ **Diseñar y Elaborar artefactos de diseño:** El Arquitecto de la Fábrica junto con el Equipo de Trabajo elaboran los artefactos de salida de esta fase.
- ✓ **Revisión y aprobación del diseño:** Líder Técnico (Fiduprevisora) y Director de Software de Fiduprevisora validan el diseño técnico para su aprobación.

✓ **Entregables:**

- ❖ Documento de Arquitectura
- ❖ Diseño Técnico: El Diseño debe contemplar los siguientes niveles:
  - Arquitectura: Componentes del sistema y sus interconexiones.
  - Diseño del Código: Algoritmos y Estructuras de Datos.
  - Diseño de Ejecución: Asignación de memoria, tiempos de ejecución y optimización de código.

#### 9.5. CONSTRUCCIÓN

El objetivo de la presente fase es plasmar en un producto Software las especificaciones técnicas y funcionales derivadas del Diseño.

- ✓ **Construcción:** El Arquitecto dirige el desarrollo de la arquitectura, difunde ante El Líder/Especialista Técnico (Fábrica) y el Equipo de Desarrollo los patrones a implementar, rigiéndose por el estándar de Arquitectura. Soporta decisiones técnicas del Equipo de Trabajo.
- ✓ **Documentación del código fuente:**

{fiduprevisora}	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 24 DE 48
		FECHA	VERSIÓN No. 10

- ✓ **Ejecución Pruebas Unitarias:** El ingeniero desarrollador de la Fábrica de Desarrollo prepara la codificación de las unidades software a implementar y acomete así las pruebas unitarias de cada una de las unidades, modificando el código. Esto se repetirá de forma iterativa para cada una de las unidades software.
- ✓ **Elaboración y Ejecución de la guía de despliegue:** El Líder/Especialista Técnico (Fábrica) verifica la guía de despliegue elaborada por el ingeniero desarrollador. Así mismo, se asegura que se tenga la ambientación necesaria para su despliegue. Si cumple con los criterios de aceptación, procede a implantar la guía de despliegue en el ambiente de pruebas.

Seguido a esto se debe realizar (de ser necesario) el análisis de código estático con la ayuda de la herramienta definida por la Fábrica de Desarrollo. Siempre que haya observaciones o modificaciones que impacten la calidad del desarrollo, éstas serán corregidas por el ingeniero de desarrollo. Esta revisión de código es realizada por el arquitecto y/o el equipo de calidad de la Fábrica de Desarrollo.

- ✓ **Entregables:**
  - ❖ Guía de Despliegue
  - ❖ Evidencia Pruebas unitarias
  - ❖ Código fuente de la solución documentado

## 9.6. PRUEBAS

Tiene como propósito revisar y validar el cumplimiento de las especificaciones detalladas de requerimientos y casos de uso de sistema frente al producto de software.

- ✓ **Revisión estrategia de pruebas:** El equipo de pruebas de la Fábrica realiza revisión de la estrategia definida para llevar a cabo las actividades de pruebas y la tipología de pruebas a realizar, tales como Pruebas de Calidad, seguridad, de rendimiento y de integración.
- ✓ **Aclaración de dudas:** El Analista de Calidad junto con El Líder/Especialista Técnico (Fábrica) aclaran dudas funcionales y técnicas generadas para llevar a cabo la ejecución de pruebas.
- ✓ **Alistamiento Ambiente de pruebas:** La Fábrica de Desarrollo debe garantizar el alistamiento del ambiente y del conjunto de datos de pruebas, roles, permisos y configuraciones, para la ejecución de Casos de Prueba.

Una vez realizado el alistamiento de Datos de Prueba, Líder/Especialista Técnico valida la guía de despliegue proporcionada por el desarrollador y de cumplir con el entorno de calidad, procede a solicitar la implementación de la guía en el ambiente de pruebas al Gestor de Versiones y despliegues de Fiduprevisora.

- ✓ **Ejecución Casos de Pruebas:** El equipo de Calidad de la Fábrica de Desarrollo realiza el paso a paso de cada escenario de prueba, revisa que el resultado en el objeto de prueba corresponda

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 25 DE 48
		FECHA	VERSIÓN No. 10

con el criterio de aceptación de la prueba (producto de la especificación detallada de requerimientos o caso de uso). Así mismo toma la evidencia de la prueba y determina si la prueba cumple con los criterios de aceptación. En caso de cumplir con los criterios de aceptación, publica la evidencia en el repositorio documental del proyecto.

En caso de que no cumpla se debe registrar la incidencia de pruebas en la herramienta Mantis proporcionada por la fábrica de Desarrollo.

- ✓ **Reporte y Gestión de Incidencia:** El equipo de Calidad de la Fábrica de Desarrollo y/o el Usuario Funcional, registrarán las incidencias detectadas durante las pruebas en la herramienta de gestión de incidentes de pruebas Mantis suministrada por la Fábrica de Desarrollo, incluyendo el paso a paso y la evidencia de la prueba.
- ✓ **Control de Calidad:** El analista de Calidad de la Fábrica de Desarrollo o quien haga sus veces, y/o el Usuario Funcional, valida que la ejecución de pruebas funcionales haya sido satisfactoria y genera el informe para la ejecución de Plan de Pruebas de usuario y realiza la actualización en el repositorio documental.

Una vez realizado el control de calidad, el Gestor de Versiones y despliegues de Fiduprevisora valida la guía de despliegue con los cambios que se hayan generado después de realizadas las pruebas de calidad. De cumplir con los respectivos lineamientos, procede a implantar la guía en el ambiente de pruebas de usuario.

- ✓ **Control de seguridad:** El analista de seguridad de la Fábrica de Desarrollo o quien haga sus veces, y/o la dirección de seguridad de la información, ejecutarán la pruebas de seguridad sobre la aplicación y generará el informe de los resultados obtenidos. La fábrica de software deberá atender la remediación de las vulnerabilidades identificadas aportando a más tardar a los quince (15) días calendario el plan de remediación que incluirá como mínimo:
  - Identificación y descripción de la vulnerabilidad
  - Criticidad.
  - Actividad a ejecutar para la remediación
  - Fechas de inicio y cierre de la remediación.
  - Responsable de la ejecución.
- ✓ **Ejecutar Plan de Pruebas de Usuario:** El Usuario Funcional (Fiduprevisora), realiza la prueba formal para determinar si el producto de software satisface o no los criterios de aceptación del área funcional, revisa que el software esté preparado y lo puedan utilizar los usuarios Funcionales y Directos que realizan las funciones y tareas que fueron determinadas por el mismo. De encontrarse fallas o errores, se reportarán y gestionarán en la herramienta de incidencias de pruebas Mantis suministrada por la Fábrica de Desarrollo.
- ✓ **Entregables:**
  - ❖ Acta de entrega de producto
  - ❖ Documento de pruebas de Calidad

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 26 DE 48
		FECHA	VERSIÓN No. 10

- ❖ Plan de Pruebas de usuario y evidencias
- ❖ Acta de Aceptación de pruebas de usuario

### 9.7. IMPLANTACIÓN

El objetivo es describir las actividades asociadas a garantizar que el producto de software esté disponible en ambiente de producción para los Usuarios Funcionales.

- ✓ **Planificación de la Implantación:** Decidir la estrategia de implantación del Producto de Software y desarrollar la planificación detallada de todas las tareas a realizar para la correcta implantación del Sistema. Se debe convocar al Comité Asesor de Cambios (CAB) para su validación y aprobación para el despliegue en el entorno de Producción. Para los casos que se requiera (dado por un carácter de emergencia) se convocará en su defecto un ECAB.
- ✓ **Despliegue en producción:** Iniciar actividades para la implantación del nuevo Sistema y transferir la aplicación al entorno de producción. Se realiza el despliegue hacia ambiente de producción en la herramienta para control de versiones Github 2.22.0.
- ✓ **Entrega y aceptación del Sistema:** Realizar el cierre de la puesta en producción y obtener la aceptación formal del Sistema por parte del Usuario Directo y Funcional de Fiduprevisora.
- ✓ **Entregables:**
  - ❖ Formato Listado de asistencia capacitaciones
  - ❖ Disponer el código fuente en la herramienta de control de versiones Github 2.22.0
  - ❖ Manual de explotación
  - ❖ Manual de Usuario
  - ❖ RFC- Formato de Solicitud de Cambio
  - ❖ Acta de comité de aceptación de paso a producción

### 9.8. ESTABILIZACIÓN

Tiene por objetivo resolver incidencias y/o ajustar defectos encontrados en el producto de software instalado en ambiente productivo.

El tiempo de estabilización corresponde a dos meses después de la puesta en producción, contando a partir de la primera aplicación efectiva de la guía de despliegue en el ambiente productivo. En caso de que al finalizar el tiempo determinado para la estabilización, se encuentren incidencias abiertas pendientes de solución, se extenderá el tiempo de estabilización a un mes correspondiente al tiempo de resolución de cada incidencia.

- ✓ **Actualizar documentación:** El Líder Técnico de Fiduprevisora, con base en los criterios definidos para incluir un tiquete en la fase de estabilización, determina si la incidencia reportada debe ser gestionada en esta fase. De ser así, El Líder/Especialista Técnico (Fábrica) de la Fábrica de Desarrollo asigna a los responsables para solucionar la incidencia. Así mismo, asigna al Equipo de

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 27 DE 48
		FECHA	VERSIÓN No. 10

Trabajo la actualización (si lo requiere), de la documentación producto de la incidencia reportada.

- ✓ **Elaborar el Acta de Estabilización y Garantía:** El Líder/Especialista Técnico (Fábrica) de la Fábrica de Desarrollo, elabora el Acta de Garantía y Soporte. Incluyendo la fecha de instalación en ambiente productivo y fecha de finalización de la garantía para el producto de software instalado.
- ✓ **Formalizar Aprobación Entregables:** El Líder/Especialista Técnico (Fábrica) de la Fábrica de Desarrollo, elabora el Acta de Entrega de Artefactos y el Acta de Garantía del Producto de Software. Se formaliza la aceptación de las actas, mediante firma del Director de Software de Fiduprevisora y El Líder/Especialista Técnico (Fábrica).
- ✓ **Entregables:**
  - ❖ Acta de Entrega de Artefactos
  - ❖ Acta de Garantía del Producto de Software

## 9.9. CIERRE

Su objetivo es formalizar la finalización del proyecto ejecutado. Adicionalmente, a partir de este punto, se transfiere la responsabilidad del desarrollo al equipo de soporte, notificándose a los grupos interesados (Desarrollo, usuarios, Integrador, etc.).

De presentarse incidencias nuevas posteriores al cierre del ticket, éstas deberán ser reportadas a través de la Mesa de Servicio como un incidente de acuerdo al proceso de gestión de incidentes (ITIL V3.0).

- ✓ **Solicitar cierre:** El Analista de Calidad de la Fábrica de Desarrollo solicita el cierre de proyecto al Líder Técnico de Fiduprevisora.
- ✓ **Aprobación cierre:** El Líder Técnico de Fiduprevisora aprueba al Analista de Calidad de la Fábrica de Desarrollo, realizar el cierre del ticket o caso, el cual debe darse en la herramienta de gestión donde fue reportado el caso y con la debida aceptación del Usuario Funcional y Directo.
- ✓ **Formalizar cierre:** El Líder Técnico de Fiduprevisora y el Analista de Calidad de la Fábrica de Desarrollo formalizan el cierre mediante la firma del Acta de Cierre del ticket o caso.
- ✓ **Entregables:**
  - ❖ Acta de Cierre
  - ❖ Formalización de Cierre en la herramienta de gestión de solicitudes con el debido soporte.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 28 DE 48
		FECHA	VERSIÓN No. 10

**Nota:** En aquellos casos donde el desarrollo sea cancelado o anulado, se elaborará el informe de cierre incluyendo el motivo por el cual la solicitud fue cancelada o suspendida y se dará cierre al ticket o caso.

**10. FLUJOGRAMA**

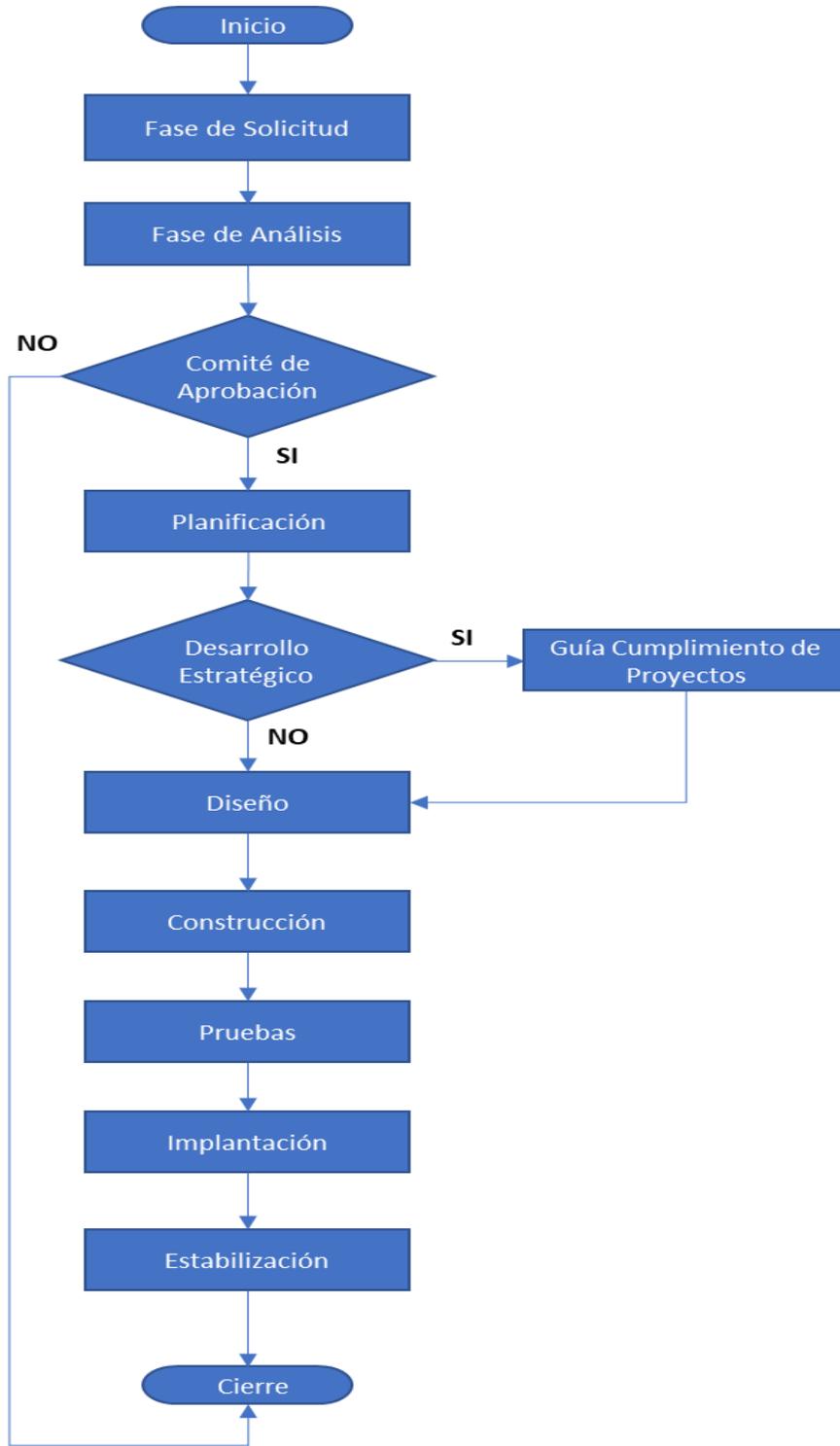


Imagen 1. Diagrama de flujo de implementación de una solución.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 30 DE 48
		FECHA	VERSIÓN No. 10

## 11. POLÍTICAS Y LINEAMIENTOS

### 11.1. CONTROL DE CAMBIO

Los cambios al requerimiento en desarrollo serán evaluados en primera instancia por el Líder Técnico de Fiduprevisora, el cual notificará al Analista Requerimientos de la Fábrica de Desarrollo y este a su vez al Líder/especialista técnico de la Fábrica. De ser aprobado, se incluirá en la actual fase del Proyecto e iniciará su proceso a través de una Solicitud Formal y cumplirá el debido proceso de la Metodología de desarrollo y mantenimiento de software.

Los controles de cambio aplicarán a partir de la fase de Construcción, es decir, si el Usuario Directo realiza un alcance al SRF, tiene hasta la fase de Diseño para hacerlo.

### 11.2. PRIORIZACIÓN DE CASOS DE DESARROLLO

La priorización de casos o tickets de desarrollos y mantenimientos de software se realizará en base a que los cambios y nuevas funcionalidades Normativas o de Ley (Regulatorios) tendrán prioridad frente a las solicitudes que sean diferentes a las anteriormente descritas. Las Solicitudes que requieran ser priorizadas serán evaluadas por el comité requerimientos para su posterior presentación en el comité de Presidencia de Fiduprevisora para su respectiva aprobación o negación de priorización.

### 11.3. PROCEDIMIENTOS Y PROCESOS OPERATIVOS

Después cumplirse con las respectivas fases de desarrollo y su respectiva entrega del producto en el ambiente de producción al usuario Definitivo y Funcional, es de total responsabilidad de estos usuarios la definición, documentación y socialización de los procesos y procedimientos operativos frente a la utilización del sistema de información sobre el cual se realizó el mantenimiento o desarrollo de software.

### 11.4. ASEGURAMIENTO DE CALIDAD

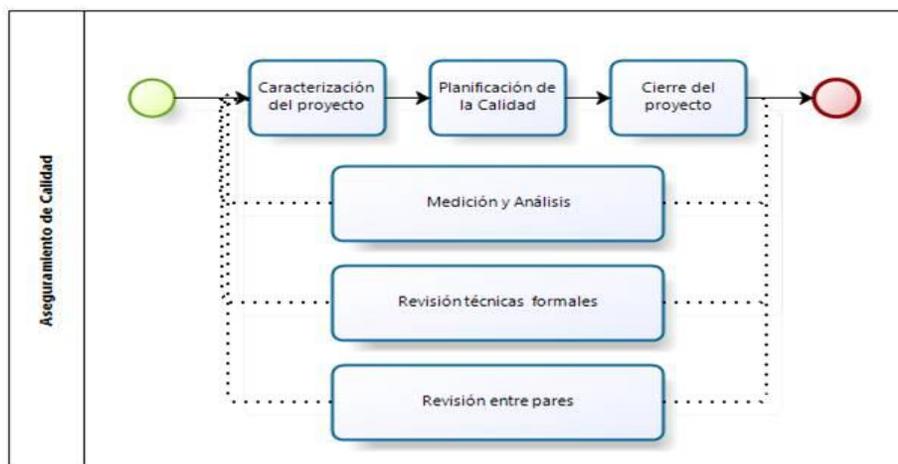


Imagen 2. Diagrama de flujo del Aseguramiento de Calidad.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 31 DE 48
		FECHA	VERSIÓN No. 10

El objetivo de este segmento es contextualizar el proyecto sirviendo de guía, en cuanto a las actividades relativas al aseguramiento de la calidad y desarrollar una infraestructura para la captura de indicadores y análisis de los mismos.

- Caracterización del Proyecto: Análisis del contexto del proyecto y tipificación del mismo.
- Planificación de la Calidad: Definición del Ciclo de Vida Ágil e identificación del método de desarrollo pactado con la Fábrica de Desarrollo, patrones y aceptación de la nomenclatura, documentación del código fuente y los diferentes estándares a emplear en la solución; así como establecer las prácticas específicas, los recursos y la secuencia de actividades relacionadas con la calidad de los productos o procesos.
- Cierre del Proyecto: Análisis del proyecto una vez finalizado, con el objeto de realizar el análisis de las lecciones aprendidas y generar una base de conocimiento de experiencias previas a proyectos futuros.
- Documentación: La Fábrica de Desarrollo es responsable de la existencia, control y actualización de un repositorio con la documentación y versiones de código de cada una de las aplicaciones en Github 2.22.0.
- Medición y Análisis: Como consecuencia del aseguramiento de la calidad del proyecto, mediante la gestión de listas de chequeo de cada etapa, se entregará como resultados indicadores de alerta que permitan tomar decisiones a tiempo.
- Revisiones Técnicas Formales: realización de comprobaciones técnicas con el objeto de asegurar los beneficios técnicos del desarrollo.
- Revisiones y seguimiento: Realización de una serie de comprobaciones técnicas a lo largo del ciclo de vida del Proyecto, en las cuales los Líderes de Fiduprevisora versus los Líderes de la Fábrica de Desarrollo, llegan a un acuerdo y aprobación de cada etapa.
- Entrega: En el momento de finalización del contrato la Fábrica de Desarrollo debe entregar la documentación y versiones de código de cada una de las aplicaciones.

### 11.5. GARANTÍAS Y SANCIONES

Como complemento a lo presentado en este documento, los acuerdos de nivel de servicio para cada una de las etapas en la metodología ágil deben tener en cuenta:

- Los tiempos de los ANS son posibles teniendo en cuenta que se cumpla y aplique la metodología definida en acuerdo entre la Fábrica de desarrollo y Fiduprevisora.
- Los desarrollos y/o mantenimientos no pueden realizarse fuera de metodología.
- No se dará inicio a cualquier etapa de la metodología de desarrollo hasta tanto se formalicen los requisitos mínimos documentales: Solicitud de Requerimiento Funcional o Product backlog debidamente diligenciados.
- La entrega del listado de Casos de Uso se basa en el requerimiento funcional, la precisión de la proyección depende directamente de la calidad de este documento. El listado de casos de uso es susceptible de refinación o afinamiento, pero es importante tener en cuenta que cualquier ajuste en la cantidad y/o complejidad de los casos de uso impacta el cronograma.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 32 DE 48
		FECHA	VERSIÓN No. 10

### 11.5.1. MÉTRICAS DE OBLIGATORIO CUMPLIMIENTO

**Objetivo:** Garantizar la correcta y completa aplicación de la metodología de desarrollo de Software y manual operativo de Fiduprevisora por parte del proveedor (Fabrica de Desarrollo) a utilizarse en todo nuevo desarrollo de software, mantenimiento adaptativo, perfectivo, preventivo o evolutivo. El porcentaje de cumplimiento no deberá ser inferior al 90,00% en el periodo de medición.

### 11.5.2. CUMPLIMIENTO DE LA METODOLOGÍA DE DESARROLLO Y DOCUMENTACIÓN

El proveedor deberá tener en cuenta y dar estricto cumplimiento a cada una de las Etapas de la metodología, mediante la entrega de todos los artefactos y/o entregables definidos para cada una de las etapas en los tiempos establecidos y estimados en el PDT y Cronograma definidos, con lo cual Fiduprevisora certificará que dicho cumplimiento se ha dado de manera efectiva, que será parte integral para proceder con el pago de la facturación por parte de Fiduprevisora.

### 11.5.3. ETAPAS

**Etapas o fases de la metodología Comprometidas (ET):** Son las etapas definidas y establecidas para la ejecución del desarrollo o mantenimiento de software.

**Etapas o fases de la metodología cumplidas (EC):** Se entenderá como etapas cumplidas de la metodología, cuando cada fase o etapa fue entregada en los tiempos acordados y establecidos en el PDT, Cronograma o metodología definida y que a su vez hayan entregado cada uno de los artefactos, y/o entregables definidos en cada una de las fases o etapas.

**Etapas o fases de la metodología incumplidas (EI):** Se entenderá como etapas incumplidas de la metodología, cuando se presente incumplimiento en los tiempos de entrega acordados y establecidos en el PDT, Cronograma o metodología definida, y/o en el momento que no se haya entregado la totalidad de los artefactos y/o entregables definidos en cada una de las etapas o fases.

### 11.5.4. INDICADOR DE OPORTUNIDAD GENERAL EN LA ATENCIÓN

Como resultado del cálculo se dispondrá del porcentaje de oportunidad en la atención de las actividades el cual no deberá ser inferior al 90,00%. (Ver ejemplo:)

**EC:** Etapas o fases de la metodología cumplidas

**EI:** Etapas o fases de la metodología incumplidas

**ET:** Etapas o fases de la metodología Comprometidas

**EA:** Eficiencia en la Atención

**EA (Eficiencia en la Atención) = (EC/ET)\*100%**

{fiduprevisora}	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 33 DE 48
		FECHA	VERSIÓN No. 10

Caso (Desarrollo)	EC	EI	ET	EA
A	7	0	7	100%
B	6	1	7	86%
C	8	0	8	100%
D	6	2	8	55%
E	8	0	8	100%
F	7	0	7	100%

Tabla 1. Eficiencia en la atención.  
Fuente: Elaboración Propia Fiduprevisora.

**PGO:** Porcentaje General de Oportunidad en la atención  
**N:** Número total de casos o Desarrollos evaluados en el periodo.

$$PGO = \frac{\sum (EA)}{N}$$

Caso (Desarrollo)	Porcentaje
A	100%
B	86%
C	100%
D	75%
E	100%
F	100%
<b>POG</b>	<b>93%</b>

Tabla 2. Porcentaje General de Oportunidad en la atención.  
Fuente: Elaboración Propia Fiduprevisora.

#### 11.5.5. DESCUENTOS POR INCUMPLIMIENTOS

Se establecen los siguientes criterios de incumplimiento y descuentos sobre el Contrato, por la prestación de los servicios acordados:

1. Cuando se presente el incumplimiento de un Indicador durante un periodo medición, la Fábrica deberá presentar un Plan de Mejora del Servicio (SIP) el cual será revisado por FIDUPREVISORA S.A. para su aprobación, quien podrá sugerir cambios o adiciones en el plan cuando resulte procedente. El Plan de Mejora del Servicio (SIP) deberá ser aplicado a partir del mes siguiente del incumplimiento del indicador.
2. Si el grupo de soporte o proveedor de tecnología presenta incumplimiento de un indicador en el periodo y su plan de Mejora SIP permite el cumplimiento del indicador por los 6 meses subsiguientes, se dará la respectiva calificación de cumplimiento del indicador y los incumplimientos anteriormente presentados serán subsanados.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 34 DE 48
		FECHA	VERSIÓN No. 10

3. Se aplicará descuento, cuando el grupo de soporte o proveedor de tecnología incumpla durante dos (2) periodos continuos o discontinuos de un mismo indicador de servicio y se extenderá hasta cuando el grupo de soporte o proveedor de tecnología cumpla el respectivo Indicador.
4. Se aplicará descuento, cuando el grupo de soporte o proveedor no cumpla con los planes de remediación establecidos para las vulnerabilidades y/o fallas de seguridad detectadas en el desarrollo.

### 11.5.6. CÁLCULO DE DESCUENTOS Y/O SANCIONES POR INCUMPLIMIENTO

Para efectos del cálculo del descuento a efectuar a la factura de la Fábrica de incidentes, requerimientos o mantenimientos y nuevos desarrollos en caso de incumplimiento de cualquier indicador durante dos (2) meses continuos o discontinuos, se aplicarán los descuentos descritos en la siguiente tabla:

Rango porcentual de incumplimiento de un mismo indicador	Porcentaje a descontar de la factura presentada por la Fábrica
Dos meses continuos o discontinuos	5 %
Tres meses continuos o discontinuos	6 %
Cuatro meses continuos o discontinuos	7 %
Cinco meses continuos o discontinuos	8 %
Seis meses continuos o discontinuos	9 %
Siete meses continuos o discontinuos	10 %
Ocho meses continuos o discontinuos	11 %
Nueve meses continuos o discontinuos	12 %
Diez meses continuos o discontinuos	13 %

Tabla 3. Tabla relación porcentual de sanciones.

Fuente: Elaboración Propia Fiduprevisora.

***En el momento que se presente incumplimiento durante seis meses continuos o discontinuos en un mismo indicador y que a su vez no cumpla con el respectivo Plan de Mejora de Servicio, se entenderá como causal de terminación del contrato sin excluir la ejecución de lo contemplado en IN-GAD-01-006 Aplicación Clausula Penal Pecuniaria, y se iniciará el proceso de transición para la contratación de una nueva fábrica.***

### 11.6. EXCEPCIONES

Tiempo en horas y fracción asociadas a retrasos por eventos ajenos a la responsabilidad de la fábrica de Desarrollo (Decisiones de FIDUPREVISORA S.A., demora en la expedición de permisos de acceso, hechos de fuerza mayor, Ausencia del usuario, Ventanas de mantenimiento). El tiempo de excepción para cada **fase o etapa** afectada durante el período medido, debe ser documentado comple-

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 35 DE 48
		FECHA	VERSIÓN No. 10

tamente, justificado, validado entre las partes (Fabrica de Desarrollo y Fiduprevisora S.A.) y aprobado por el Director de Software de Fiduprevisora S.A., sin embargo, durante el transcurso del presente contrato se podrán agregar nuevos tipos de excepciones, previa aprobación de Fiduprevisora S.A. Se utilizarán inicialmente los siguientes tiempos:

- Ausencia del Usuario Funcional y/o usuario en general, en el momento de la atención del servicio.
- Ventanas de mantenimientos programados o autorizaciones de suspensión del servicio.
- Las remediaciones de las vulnerabilidades en las aplicaciones no son del alcance de la fábrica de software.

Nota: La aplicación del tiempo de excepción durante el plazo del contrato, la autoriza exclusivamente el Director de Software de Fiduprevisora S.A.

#### **11.7. EVALUACIÓN DE PROVEEDORES**

La evaluación del proveedor se hará conforme a lo establecido en el numeral 6.2.3, documento MP-GAD-01-015 que corresponde a la Supervisión de Contratos y Órdenes de Adquisición de Bienes y/o Servicios.

En la evaluación de proveedores se debe tener el concepto claro de los criterios a evaluar según el tipo de proveedor con el fin de garantizar la objetividad, de acuerdo con lo establecido en el formato FR-GAD-01-013 - Evaluación de Proveedores.

El supervisor del contrato a través del formato FR-GAD-01-059 - Reevaluación de Proveedores realizará la reevaluación a los contratistas que hayan obtenido una calificación "DEFICIENTE" a los dos meses de haber sido evaluados y a partir de los acuerdos de servicio pactados o el plan de acción generado con el fin de verificar el cumplimiento de estos.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 36 DE 48
		FECHA	VERSIÓN No. 10

## 12. ESTÁNDARES DE PROGRAMACIÓN

### 12.1. OBJETIVO

Llevar un orden en el estilo de programación, que sea entendible por todos los programadores y estandarice la nomenclatura del código fuente, disminuyendo la complejidad a la hora de la depuración y el mantenimiento.

### 12.2. REGLAS

Regla General:

El nombramiento de cualquier objeto debe ser descriptivo.

Regla 1: Nombres para los Archivos externos

Archivos de Scripts:

Los archivos donde reposan los scripts de creación de objetos de la base de datos se nombran de la siguiente forma, ejemplos:

TRG\_PP\_MOVIMIENTOS.sql  
VIW\_TE\_PAGOS.sql  
PKG\_PP\_DOCUMENTOS.sql

TRG → Prefijo del tipo de objeto, así: TRG-Trigger, SEQ-Secuencia, VIW-Vista, TAB-Tabla, IND-Índice, PKG-Paquete, FCN-Función, PRC-Procedimiento.

PP → Abreviación del módulo.

MOVIMIENTOS → Descripción del uso.

.sql → Extensión del archivo.

Cuando se trate de scripts que contienen modificaciones, se debe anteponer el prefijo MOD, ejemplo: MOD\_PKG\_PP\_DOCUMENTOS.sql.

Formas

Nombre forma: CPMOORPA

CP → Módulo al que pertenece

MO → Indicativo del submenú, en este caso modulo operativo.

ORPA → Abreviación de la descripción de uso, para este caso órdenes de pago.

Reporte

Nombre Reporte: CPMRORPA

CP → Módulo al que pertenece

MR → Indicativo del submenú, en este caso modulo operativo.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 37 DE 48
		FECHA	VERSIÓN No. 10

ORPA → Abreviación de la descripción de uso, para este caso órdenes de pago.

Regla 2: Nombres de Objetos de la Base de Datos

Campos

Los nombres de los campos estarán formados así:

Ejemplo: docu\_fecha\_documento

DOCU → Prefijo de la tabla.

FECHA\_DOCUMENTO → Descripción del campo. Sí se trata de un campo compuesto va separado por underscore (\_).

Constraints

Not Null

Los nombres de este constraint estarán formados así:

Ejemplo: nn\_cons\_auxi\_cons

NN → Indica que es not null.

CONS\_AUXI\_CONS → Nombre de la Columna.

Primary Key:

Los nombres de este constraint estarán formados así:

Ejemplo: pk\_sc\_tcons

PK → Indica que es llave primaria.

SC\_TCONS → Nombre de la tabla.

Foreign Key:

Los nombres de este constraint estarán formados así:

Ejemplo: fk\_sc\_tcons\_ge\_tauxil

FK → Indica que es llave foránea o de referencia.

SC\_TCONS → Nombre de la tabla.

GE\_TAUXIL → Nombre de la tabla con la que se referencia.

Funciones:

Los nombres de las funciones estarán formados por:

Ejemplo: FNC\_CC\_ARCHIVO

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 38 DE 48
		FECHA	VERSIÓN No. 10

FNC → Nos indica que es una función.  
 CC → Sigla del Módulo. En este caso Cartera.  
 ARCHIVO → Descripción del proceso donde aplica.

Índices:

Los nombres de los índices estarán formados así:  
 Ejemplo: ind\_sc\_tcons\_1

IND → Indica que es índice.  
 SC\_TCONS → Nombre de la tabla.  
 1 → Consecutivos de Índices, que irá de 1 a n.

Paquete base de Datos:

Los nombres de los Paquetes estarán formados por:  
 Ejemplo: PKG\_CC\_ARCHIVO

PKG → Nos indica que es un paquete.  
 CC → Sigla del Módulo. En este caso Cartera.  
 ARCHIVO → Descripción del proceso donde aplica.

Procedimientos Base de Datos:

Los nombres de los Procedimientos estarán formados por:  
 Ejemplo: PRC\_CC\_ARCHIVO

PRC → Nos indica que es un procedimiento.  
 CC → Sigla del Módulo. En este caso Cartera.  
 ARCHIVO → Descripción del proceso donde aplica.

Secuencia base de Datos:

Los nombres de las Secuencias estarán formados por:  
 Ejemplo: SEC\_CC\_archivo

SEC → Nos indica que es una secuencia.  
 CC → Sigla del Módulo. En este caso Cartera.  
 ARCHIVO → Descripción del proceso donde aplica.

Tablas:

Los nombres de las tablas estarán formados así:  
 Ejemplo: sc\_tcons

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 39 DE 48
		FECHA	VERSIÓN No. 10

SC→ Indica el módulo.

T→ Indica que se trata de una tabla.

CONS → Descripción de la tabla.

Para el caso de tablas temporales, su nombre estará formado de la siguiente manera, ejemplo:  
SC\_TMP\_CONS.

SC -> Indica el módulo.

TMP -> Indica que se trata de una tabla temporal

CONS -> Nombre de la tabla.

#### Trigger base de Datos:

El nombre de esta compuesto por el prefijo TRG, el nombre de la tabla, indicador del tiempo (BEFORE o AFTER) más una descripción de ejecución (INSERT, UPDATE o DELETE) Ej: trg\_sc\_comprobantes\_bi, trg\_cc\_formatos\_super\_au.

#### Vistas:

Los nombres de las vistas estarán formados así:

Ejemplo: sc\_vcons

SC→ Indica el módulo.

V→ Indica que se trata de una vista.

CONS → Descripción de la vista.

#### REGLA 3: Declaraciones dentro de Programas

Consiste en utilizar prefijos en minúsculas que se añaden a los nombres de las variables, o constantes que indican su tipo. El resto del nombre se debe escoger de tal forma que se acomode perfectamente a la situación, que indica, lo más claramente posible, la función que realiza la variable.

Prefijo	Significado
b	Booleano
c	Carácter (un byte)
dw	Entero largo de 32 bits sin signo (DOBLE WORD)
f	Flags empaquetados en un entero de 16 bits
h	Manipulador de 16 bits (HANDLE)
l	Entero largo de 32 bits

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 40 DE 48
		FECHA	VERSIÓN No. 10

lp	Puntero a entero largo de 32 bits
Lpfn	Puntero largo a una función que devuelve un entero
lpsz	Puntero largo a una cadena terminada con cero
n	Entero de 16 bits
p	Puntero a entero de 16 bits
pt	Coordenadas (x, y) empaquetadas en un entero de 32 bits
rgb	Valor de color RGB empaquetado en un entero de 32 bits
sz	Cadena terminada en cero
w	Entero corto de 16 bits sin signo (WORD)

#### Ejemplos:

nContador: la variable es un entero que se usará como contador.

szNombre: una cadena terminada con cero que almacena un nombre.

bRespuesta: una variable booleana que almacena una respuesta.

#### Declaración de Constantes

Para declarar una constante se debe escribir toda la palabra en letra mayúscula, tener en cuenta que el nombre de la constante sea descriptivo generalmente este nombre debe ser un sustantivo o nombre, el nombre se define con tres letras que definen el tipo del objeto seguido del nombre, si se compone de más de una palabra, la segunda debe ser separada con under score (\_) ejemplo:

```
final static int INTSIZE = 20;
```

```
final static float FLOBASE = 3.5;
```

```
final static int INTHIGH_GRAVITY = 3;
```

```
final static String STRCOMPANIA = "";
```

Si se declara más de una constante, hacer la declaración de cada una en una línea individual.

#### Declaración de Cursores:

Los nombres de los cursores estarán formados así:

Ejemplo: c\_pagos

C→ Indica que se trata de un cursor.

PAGOS → Descripción del cursor.

#### Declaración de Variables

Para declarar una variable se debe escribir toda la palabra en letra minúscula, si se compone de más de una palabra, la segunda debe comenzar con mayúscula, generalmente este nombre debe ser un sustantivo o nombre. Tener en cuenta que el nombre de la constante sea descriptivo. El nombre se define con tres letras que definen el tipo del objeto seguido del nombre

{fiduprevisora}	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 41 DE 48
		FECHA	VERSIÓN No. 10

```
int    intNumero;
String strFechaCierre;
```

Si se declara más de una variable, hacer la declaración de cada una en una línea individual.

#### REGLA 4: Nombres para los Objetos en Java

Para darle nombre a los archivos en donde estará definido nuestro código, empezamos escribiéndolo con letra mayúscula, siguiendo las demás con letra minúscula. Si el nombre se compone de más palabras, empezarlas por mayúsculas sin dejar espacio, ejemplo:

Cuenta.java , BalanceCuenta.java.

Nota: Para evitar nombres muy largos, Resumir el nombre del archivo .java en máximo dos palabras siempre y cuando el nombre sea descriptivo.

Las reglas para nombrar las variables y constantes de java son las mismas descritas anteriormente.

#### Declaración de Clases

Para declarar una clase se debe escribir la palabra comenzándola con letra mayúscula siguiendo con las demás letras minúsculas, si el nombre contiene más de una palabra empezar la siguiente con letra mayúscula sin dejar espacio, ejemplo:

```
public class Carro
{
**cuerpo de la clase**...
}
class FrenoDelantero
{
**cuerpo de la clase**...
}
```

En el desarrollo del software para Web es frecuente encontrar clases java que hacen referencia a objetos de bases de datos ejemplo: una tabla; para estos casos la clase se denominara de igual manera que el objeto de bases de datos más una palabra descriptiva de su función ejemplo:

Objeto de base de datos "Tabla" --> usuario\_sistema

```
class UsuarioSistemaExperto
{
**cuerpo de la clase**...
}
class UsuarioSistemaControlador
{
**cuerpo de la clase**...
}
```

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 42 DE 48
		FECHA	VERSIÓN No. 10

### Declaración de Métodos

Para declarar un método se escribe la palabra en minúscula, si se tienen más palabras en el nombre del método, empezar la siguiente en mayúscula sin dejar espacio. Generalmente este nombre debe ser un verbo “crear, borrar, etc.” Ejemplo:

```
public int obtenerDato( )
{
**cuerpo del método**
}
```

```
public void cerrar( )
{
**cuerpo del método**
}
```

### Indentaciones:

Para mayor organización de las líneas de código del software, se usará la indentación estándar del lenguaje JAVA, se escribirá la siguiente línea de código a 4 espacios de la línea superior, si en la línea superior se indicó el principio de una clase o un método, ejemplo:

```
public int obtenerDato( )
{ int numero; //declaración de variable
:
:
:
}
```

La llave que determina el comienzo de una clase o método irá al lado, dejando un espacio, del nombre de dicha clase o método, como se observa en el ejemplo anterior.

Se debe cerrar la llave a la misma distancia de la primera letra de la línea que especifica la clase o método. Si dentro de las clases o métodos son invocadas algunas instrucciones, se debe dejar los mismos 4 espacios, a partir del comienzo de la definición de la instrucción, y así sucesivamente si se requieren instrucciones dentro de otras instrucciones, ejemplo:

```
public int obtenerDato( )
{
int numero;
if (instrucción)
{
numero = 20;
int i = 0;
while (instrucción)
```

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 43 DE 48
		FECHA	VERSIÓN No. 10

```
{
:
:
i++;
}
}
}
```

Ejemplo de Documentación, fracción de pl/sql:

Function FNC\_FUNCION (p\_parametro1 in varchar2,  
P\_parametro2 out varchar2) return number is

```
-----
-- autor:
-- fecha:
-- version:
-- descripcion:
-----
```

```
v_dato      number(10);
c_host constant varchar2(60) := '176.16.0.50';
--
```

```
Cursor c_cursor(pc_parametro number) is
Select cias_cias
  from ge_cias
  where Cias_cias = pc_parametro;
--
```

```
Begin
--
begin
--
-- comentario
--
```

```
Select cias_cias
  into v_dato
  from ge_cias
  where cias_cias = pc_parametro
  and cias_Cias = pc_parametro;
--
```

```
exception
--
When no_data_found then
  null;
When too_many_rows then
  null;
When others then
```

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 44 DE 48
		FECHA	VERSIÓN No. 10

```

null;
--
end;
--
begin
--
-- comentario
--
Select cias_cias
into v_dato
from ge_cias,
ge_usuarios
where cias_cias = pc_parametro
and cias_Cias = pc_parametro
and cias_usuario = usuario
and (codigo_usuario) in (Select cod_user
from usuarios
where user1=user);
--
exception
--
When no_data_found then
null; -- manejo de exception
When too_many_rows then
null; -- manejo de exception
When others then
null; -- manejo de exception
--
end;

--
-- comentario
--
Update ge_tcias
set cias_stdo = 'ACT'
where cias_Cias = cias_Cias;
--
If (sql%notfound) then
?????????
end if;
--
begin
--
-- comentario
--

```

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 45 DE 48
		FECHA	VERSIÓN No. 10

```

Insert into tabla(col1, col2,col3,
                col4, col5,col6)
values
(col1, col2,col3,
 col4, col5,col6);

```

```

--
exception
--
When others then
    null; -- manejo de exception

```

```

--
end;

```

```

--
Return v_dato;

```

```

--
Exception

```

```

--
When no_data_found then
    null;

```

```

--
End;

```

#### REGLA 5: Documentación

Se refiere a la documentación de un programa en lo referente al código, con el fin de identificar aspectos generales que sirvan de guía al desarrollador.

Los comentarios generales de un programa son:

- Encabezado del programa o módulo.
- Convención utilizada para abreviaturas.
- Límites de cada sección importante.
- Variables, constantes, procedimientos y funciones.
- Funcionamiento y parámetros requeridos para cada módulo.
- Funcionamiento detallado de los trucos.

La documentación del código de un programa se puede dividir en varios niveles:

#### Documentación de Procedimientos:

En la cabecera del fichero se escribe un recuadro con los siguientes datos:

Para nuevos procedimientos, paquetes y funciones

```

--
-- CREACION
--
-- Autor: NNNN
-- Fecha: 11-11-1111
-- Versión: 1.1.0
-- Descripción: Procedimiento que genera valor presente.

```

```

--
--

```

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 46 DE 48
		FECHA	VERSIÓN No. 10

Para ajustes a los procedimientos, paquetes y funciones.

Un ejemplo de este tipo de recuadro podría ser:

```
-- MODIFICACIONES
--
-- (Caso ###)
-- Autor: NNNN
-- Fecha: 11-11-1111
-- Versión: 1.1.0
-- Descripción: Se ajusta llamado a procedimiento, y se crean nuevas
-- variables.
--
--
```

#### Documentación de Clases:

Para hacer documentación de clases se debe seguir el siguiente modelo:

```
/**
```

```
* Nombre de la Clase:
* Propósito:
* Autor(es):
* Fecha de creación
*/
```

#### Documentación de Métodos:

Para la documentación de métodos se debe seguir el siguiente modelo:

```
/**
```

```
* Nombre del Método:
* Propósito:
* Autor
*/
```

### **12.3. MANEJO DE DOCUMENTACIÓN.**

La documentación se clasifica de acuerdo con su uso. Para la elaboración de la documentación se debe tener en cuenta a quién va dirigido, es decir, el manual puede ser manejado desde el director de la empresa hasta por quien captura datos. Por consiguiente, debe redactarse de forma clara y sencilla para que lo entienda cualquier tipo de usuario.

Contenido

Instalación: indicar detalles completos sobre la forma de instalar el sistema en un ambiente particular.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 47 DE 48
		FECHA	VERSIÓN No. 10

Iniciación al uso del sistema: en este punto se explica cómo iniciarse en el sistema y cómo se pueden utilizar sus cualidades comunes. Esta documentación debe decir al usuario cómo salir de un problema cuando las cosas funcionan mal.

Diagrama general del sistema:

Muestra en forma condensada el flujo general de la información y de las actividades que se realizan en el sistema.

Proporciona una visión general del sistema. Representar los diagramas utilizando para ello diagramas de bloques.

Explicación Genérica De Las Fases Del Sistema: en este punto se explica en forma específica y detallada todas las operaciones que aparecen representadas en forma gráfica en el diagrama general. Se analizan cada una de las fases señalando:

El proceso principal que se desarrolla.

La entrada de la información.

La obtención de un resultado parcial.

El envío de información a otra dependencia.

Presentar en este punto el flujo del sistema en las pantallas utilizadas por cada módulo. Puntos a documentar en una pantalla:

Explicación del recorrido para llegar a la pantalla.

Formato de los datos a capturar.

Formato en que son capturados los datos.

Situaciones Anormales: se presentan mensajes que se emiten al momento de la captura de los datos o cualquier condición fuera de lo normal. Ejemplo:

Situación anormal	Mensaje	Causas	Soluciones
-------------------	---------	--------	------------

Mantenimiento de Manuales: Cada vez que se desarrolle algún cambio en el sistema se debe actualizar el manual, antes de que la nueva versión se ponga en producción.

{fiduprevisora)	MODELO OPERATIVO DE MANTENIMIENTO Y DESARROLLO DE SOFTWARE	ML-GTE-02-003 CÓDIGO	HOJA 48 DE 48
		FECHA	VERSIÓN No. 10

FIRMA _____ NOMBRE JONNATTAN ARLEY GONZÁLEZ RUÍZ	FIRMA _____ NOMBRE IVÁN A. MARTÍNEZ V.
ELABORÓ Profesional Proyectos e Integraciones	ELABORÓ Consultor de Gerencia Nacional de Planeación
FIRMA _____ NOMBRE CÉSAR AUGUSTO PORTILLA MOYA	FIRMA _____ NOMBRE LEIDY POLET ROJAS PÉREZ
REVISÓ Oficial de seguridad de la Información	REVISÓ Directora de Software
FIRMA _____ NOMBRE CÉSAR BELTRÁN SOTELO	
APROBÓ Gerente de Tecnología de Información	