

# GESTIÓN DE PROYECTOS DE SOFTWARE PARA INSTITUCIONES PÚBLICAS



AUTOR  
JOHANNA DEL VALLE ALVAREZ COOZ  
Ing. de Sistemas  
FUNDACITE-Mérida  
jalvarez@funmrd.gov.ve  
Venezuela

AUTOR  
BLANCA ABRAHAM ZOGHBI  
Maestría en Computación. Ing. de Sistemas  
FUNDACITE-Mérida  
blanca@funmrd.gov.ve  
Venezuela

AUTOR  
JOSE LISANDRO AGUILAR CASTRO  
Ph.D en Sistemas Distribuidos, Ing. de  
Sistemas  
CEMISID, Universidad de Los Andes  
aguilar@ula.ve  
Venezuela

AUTOR  
OSWALDO RAMON TERAN VILLEGAS  
Ph.D. en Modelado Computacional, Ing. de  
Sistemas  
FUNDACITE-Mérida, Universidad de Los Andes  
oteran@ula.ve  
Venezuela

**Fecha de Recepción del Artículo: 20 de Octubre de 2006    Fecha de Aceptación del Artículo 4 de Diciembre de 2006**  
**Artículo Tipo 2**

## RESUMEN.

*El presente artículo presenta una forma de hacer gestión de software para entes públicos bajo la noción de prestación de servicios, de acuerdo a los lineamientos del Estado Venezolano. En particular, el Estado Venezolano ha estado promoviendo el uso del software libre a fin de potenciar la soberanía tecnológica y el desarrollo endógeno. Para lograr la gestión de servicios de software colectivo o público, se hace necesario recurrir a herramientas y a metodologías que permitan una gestión inclusiva de estos aspectos, que permita la participación de las comunidades, como la planificación estratégica y la prospectiva tecnológica. Por otro lado, es necesario también incluir ciertos elementos administrativos del desarrollo orientado a la filosofía "software libre", con sentido social y comunitario, para lo cual se ha recurrido a tomar ideas de varios métodos de desarrollo de software, tales como XP, RUP y MoProSoft. La forma de hacer la gestión propuesta está siendo seguida en la Fábrica de Software Libre del Centro Nacional de Desarrollo e Investigación en Tecnologías Libres, el cual atiende a diversos entes y comunidades de la Nación Venezolana. De esta manera, la propuesta también se ha nutrido de la creatividad y del aprendizaje derivado de la praxis desarrollada en dicho centro en el cual existen tres procesos fundamentales para el manejo de proyectos de software, estos son, gestión de proyectos, administración de proyectos específicos y desarrollo de aplicaciones basadas en el uso de componentes, esto último facilita la reutilización de código fuente existente.*

## PALABRAS CLAVE

Metodología de desarrollo de software  
Software Libre  
Gestión de proyectos de software  
Gobierno software administración pública

## ABSTRACT

This paper presents an approach to manage aspect related to software projects for public administration under the notion of service providers, following Venezuela state laws and government lines. The nation is particularly interested in free

software in order to increase technologic sovereignty and endogenous development. On the one side, it is needed the use of tools and methods that allow an inclusive management for these aspects, and the participation of different communities, such as strategic planning and technologic prospective. On the other side, it is important to include some other management elements for software development, which will be brought in from a diversity of software development methods and models such as Extreme Programming, Rational Unified Process, and MoProSoft; as well as from practice and experience. This approach is being implemented by the Free Software Factory of the National Research and Development Center for Free Technologies, which gives support to many Venezuelan official entities and communities. In this manner, the proposal has also fed from the creativity and learning generated for the praxis from this center which has three main process for software projects, they are, project management, specific project administration and application development based on component use, this last one makes easier the use of already existent source code.

## KEYWORDS

Software Development Methodology  
Free Software  
Software Projects Management  
Government software public administration

## INTRODUCCIÓN

El Estado Venezolano ha estado planteando ciertas líneas de desarrollo, donde ocupan un lugar fundamental las ideas de desarrollo endógeno y soberanía tecnológica. En este marco se ha dado apoyo, a través de leyes y políticas, a la generación y uso de herramientas de tecnología libre [1, 2, 3, 4], por sus potencialidades para apuntalar el crecimiento endógeno y soberano de la nación venezolana. En el ámbito de las instituciones públicas se ha impulsado el desarrollo y consolidación de una serie de iniciativas a través del Ministerio de Ciencia y Tecnología (MCT). Un buen ejemplo de éstas lo constituye el trabajo que se ha adelantado desde la Fundación para el Desarrollo de la Ciencia y la Tecnología (FUNDACITE - Mérida) en temas de utilización de software libre en la elaboración de productos informáticos, que se remonta al año 2001, y que luego ha evolucionado y se ha consolidado en dos organizaciones: la Fábrica de Software Libre (FSL) y la Academia de Software Libre (ASL). La ASL fue inaugurada en Mérida en Noviembre del 2004, y su actividad se ha ido replicando en todo el territorio nacional (a través de varias ASL), constituyéndose en un elemento esencial para el apuntalamiento de la apropiación, uso y difusión del conocimiento sobre herramientas de software libre a nivel Nacional. Por otra parte, la FSL ha estado desarrollando software libre para instituciones públicas venezolanas desde 2003.

Tanto la ASL como la FSL han pasado a ser proyectos adjuntos al Centro Nacional de Desarrollo e Investigación en Tecnologías Libres (CENDITEL). CENDITEL tiene como objetivo orientar la investigación en tecnologías libres al servicio de la sociedad venezolana, generando apropiación del conocimiento. Este ente aparece como el articulador de las

iniciativas antes nombradas y ya consolidadas, así como de otras en fase de conformación, en temas como hardware y telecomunicaciones, en un Centro de carácter Nacional. La idea es potenciar y consolidar las investigaciones y desarrollos en todas estas áreas bajo una óptica de estándares abiertos libres.

Dentro de la FSL se ha estado desarrollando e implementando una metodología de desarrollo de software [10], la cual, además de adaptarse al desarrollo de software libre, debe cumplir con otras prerrogativas. Entre estas, debe considerar las características específicas del tipo de organización a la que sirve: entidades públicas o sectores sociales.

Entre las consecuencias de esta consideración se tiene: 1) el objetivo fundamental es prestar un servicio, más que satisfacer una demanda; 2) la gestión debe ir hacia afuera y no sólo hacia adentro, en el sentido de que debe descubrir las necesidades del cliente, muchas de las cuáles irán más allá de la simple implementación de una aplicación de software. Esta segunda consecuencia indica que para cumplir con el servicio en cuestión, no son suficientes las nociones tradicionales de gestión de desarrollo de software. Por tanto, se hace necesaria la inclusión de herramientas como la planificación estratégica [5, 12] y la prospectiva tecnológica [6], así como cualquier otra herramienta útil para descubrir las necesidades de la organización-pública, ente o sector social receptor del servicio.

De acuerdo a lo dicho arriba, este descubrimiento de necesidades, por un lado, va enmarcado dentro de la tarea de ofrecer un servicio social por parte de la FSL, y por la otra, debe tomar en cuenta el contexto del sector receptor del servicio, y las políticas de Estado en torno a este sector. En este sentido, la gran debilidad actual del sector público, a saber: 'la poca experiencia en el uso de herramientas libres, y la alta dependencia (por la usanza, a falta de alternativas) del software propietario', es afrontada y atenuada a través de la ASL y FSL.

En cuanto a la propuesta de una metodología adaptada al desarrollo de software libre, se han considerado varios métodos, tales como Programación Extrema (XP) [7], Racional Unified Process [8], y el Modelo de Procesos para Desarrollo y Mantenimiento de Software (MoProSoft) [9]. Generalmente, estas herramientas teóricas han sido desarrolladas pensando en el desarrollo de software propietario, excepto XP.

Muchas personas opinan que los modelos de desarrollo de software propietario y libre son muy parecidos, en tanto, que la única diferencia que perciben entre ambos, es la licencia. Sin embargo, esto no es del todo cierto, ni a nivel de gestión ni a nivel operativo. En este segundo nivel, a pesar de que se pueden mantener las mismas fases de desarrollo, hay algunas diferencias significativas en la manera en que se entiende y en la que se genera el código. Algo similar aunque de mayor relevancia ocurre cuando se trata del nivel de gestión, y más aún si se trata de software para instituciones públicas. En este trabajo se presenta una forma de realizar gestión de software conceptualizada para y utilizada en la FSL de CENDITEL.

Entre los varios aspectos importantes a considerar en una

metodología de desarrollo de software libre, podemos mencionar, a manera de ejemplo: a) la liberación del código fuente, junto a la documentación asociada al mismo; b) los estilos de desarrollo del software, a saber: Catedral y Bazar.

En el estilo Catedral el desarrollo de software está dirigido de manera centralizada y el proceso de desarrollo esta restringido a un grupo de programadores, quienes trabajan fuertemente en la depuración del código con la finalidad de que los usuarios puedan ver menos errores en cada versión liberada [11]. En contraposición, en el estilo Bazar el desarrollo de software no es dirigido de manera centralizada, la construcción de la aplicación se realiza con la participación de una comunidad de interesados abierta (entran y salen personas a esa comunidad) que libera frecuentemente cada versión desarrollada, con la finalidad de que otros puedan depurar el código [11].

Ambos estilos de desarrollo presentan características diferentes, pero estas pueden ser combinadas según la necesidad del proyecto que se desarrolle. En el caso de la metodología propuesta en [10], resulta conveniente combinar ambos estilos, pues se requiere dirección centralizada del proceso de desarrollo, pero también la participación comunitaria en la construcción de aplicaciones. En este sentido, la gestión de proyectos considera características diversas de los dos estilos de desarrollo del software libre.

El resto de este documento se organiza de la siguiente manera: primero se presenta la estructura general de la metodología elaborada en [10]; seguidamente se presentará el proceso de gestión de proyectos para entes públicos; a continuación se describe la administración del proyecto; y, finalmente se presentan algunas conclusiones.

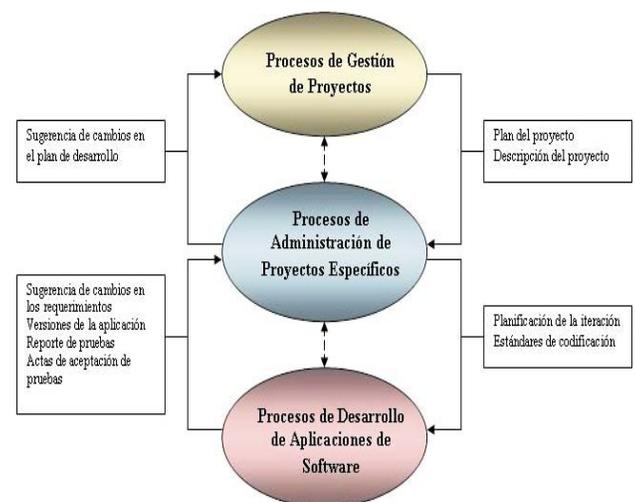
## 1. DESCRIPCIÓN GENERAL DE LA METODOLOGÍA DE DESARROLLO DE SOFTWARE LIBRE

El método al cual sirve la gestión definida en este trabajo se basa en una estructura organizacional orientada a procesos específicos. Estos procesos son: Gestión de Proyectos, Administración de Proyectos Específicos y Desarrollo de Aplicaciones de Software (ver Figura 1). En la Gestión de Proyectos y Administración de Proyectos Específicos se realizan las actividades de gerencia, planificación y control. En lo que respecta al proceso de Desarrollo de Aplicaciones se prescriben las fases de desarrollo del software que incluyen el análisis, diseño, construcción y pruebas de un sistema (aplicación). El diseño de la aplicación esta basado en una arquitectura de componentes los cuales son definidos como piezas de software que proveen un conjunto de servicios que son usados solo a través de interfaces bien definidas. Los componentes pueden ser reutilizados y adaptados a distintos sistemas, lo cual permite la reutilización de código. Este proceso está detalladamente explicado en [10].

Es importante mencionar que en los tres procesos es posible la participación de cualquier interesado externo a la organización desarrolladora, es decir, individuos o entes que no pertenecen a la organización como tal, siempre y cuando cuenten con los permisos requeridos para acceder a la plataforma de desarrollo colaborativo utilizada.

Tal como se muestra en la figura 1, los tres procesos se retroalimentan entre sí a través de los productos que se generan en cada proceso. Por ejemplo, el plan del proyecto generado en Gestión es suministrado a Administración, quien a su vez genera la planificación de las iteraciones, la cual es enviada al proceso de Desarrollo (el proceso de desarrollo es basado en iteraciones. Cada iteración pasa por todas las fases de desarrollo de software, y en cada una de ellas se le da mayor peso a la implementación de una de las funcionalidades del sistema bajo consideración). Este último proceso envía reportes de pruebas, versiones, entre otros productos, a los demás procesos. Finalmente, como resultado de cambios en los requerimientos, durante el proceso de Desarrollo puede aparecer la necesidad de actualizar el plan del proyecto.

Fig. 1: Interrelación de Procesos para desarrollo de proyectos de Software.



## 2. GESTIÓN DE PROYECTOS PÚBLICOS

La Gestión de Proyectos se encarga de gestionar los proyectos de software internos y externos a la organización. Específicamente, es responsable de generar la oferta de servicio correspondiente al proyecto que se va a desarrollar, en la cual se indican, entre otras cosas, la conceptualización y la descripción del proyecto, así como el plan general del proyecto a seguir y la definición de acuerdos relacionados con la liberación y uso del código fuente.

El diagrama de flujo de trabajo de la Figura 2 presenta la secuencia de ejecución de los pasos contemplados en el proceso de Gestión de Proyectos. Luego, en la siguiente sección se describen las actividades correspondientes a cada uno de los pasos identificados en el diagrama de flujo. Recordemos que una particularidad de este flujo es que está orientado al desarrollo de software libre, y a prestar servicios a entes públicos, de una manera que le ayude a descubrir las necesidades y deficiencias organizacionales. En este sentido la gestión da mucha importancia a la participación de los involucrados en el desarrollo del proyecto, tanto a nivel de la organización desarrolladora, como al nivel de la quien recibe el servicio. De esta manera, por un lado, desde los desarrolladores hasta el gestor de proyectos; y por el otro, desde el actor que actúa como cliente de la organización que

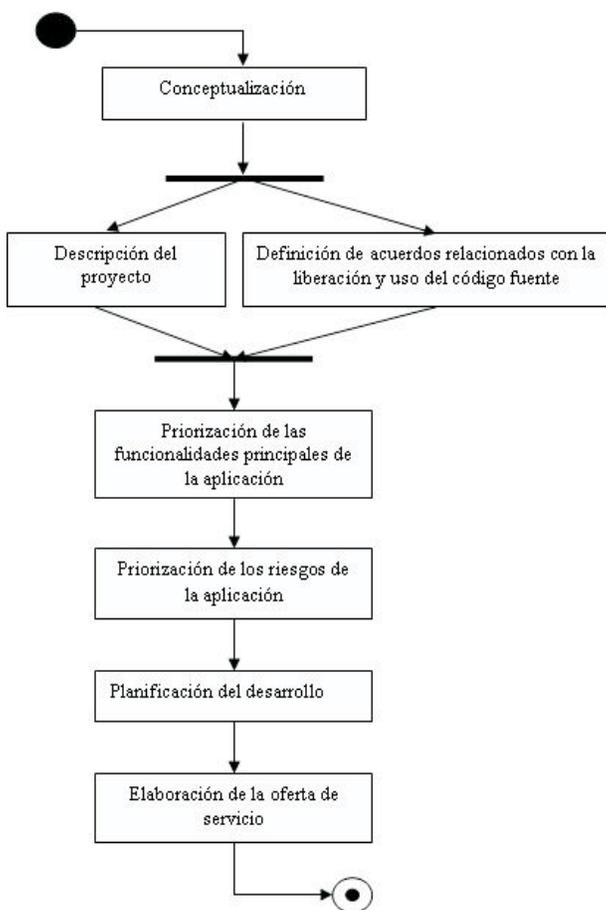
recibe el servicio, hasta aquellos usuarios finales y/o beneficiarios; se verán involucrados en el proceso de gestión del servicio (al igual que en el proceso de administración del desarrollo, que será descrito en la siguiente sección).

A continuación describiremos las actividades asociadas a cada uno de los pasos identificados en la Figura 2. Para cada actividad se indicarán los responsables de la misma, los insumos requeridos, las técnicas a utilizar y los productos que deben ser generados en cada actividad.

## 2.1 CONCEPTUALIZACIÓN

**Actividades:** Se definen las necesidades y problemas del cliente relacionadas a la aplicación a desarrollar, y en función de estas se establece el alcance del proyecto. Para la definición de proyectos estratégicos institucionales se recomienda realizar estudios de prospectiva, de planificación estratégica (específicamente, un flujograma situacional para describir el problema fundamental del receptor del servicio). Estos estudios permitirán establecer las causas y variantes que determinan las necesidades o problemas que el cliente ha definido, en función de los cuales éste requiere llevar a cabo diversos proyectos y cambios organizacionales, entre los cuales puede estar el desarrollo de una o varias aplicaciones de software específicas.

Fig. 2: Flujo de trabajo de la Gestión de Proyectos de Software



**Observaciones:** Los estudios de prospectiva y de planificación estratégica se consideran importantes no solo para proyectos complejos, sino que también son de gran utilidad para conocer las verdaderas necesidades y problemas del cliente, puesto que, por lo general, los clientes/usuarios no tienen claro el origen de sus problemas, y por tanto, sus verdaderas necesidades. Esto es aun mas palpable cuando ese cliente es una institución pública.

**Responsables:** Es necesario involucrar a un grupo interdisciplinario en esta actividad, a fin de enriquecer el estudio desde diversas perspectivas sobre las cuales se determinaría el alcance del proyecto. Por tanto, se recomienda involucrar: a) cliente/usuario, b) responsable de la Gestión de Proyectos, c) administradores de proyectos, d) desarrolladores, e) personas interesadas en el proyecto f) comunidad receptora de los servicios de la institución pública para la cual se está realizando el estudio y, g) asesores.

**Insumos:** Para determinar el alcance del proyecto se requiere conocer las necesidades y los problemas del cliente relacionados con la aplicación a desarrollar.

**Técnicas:** Análisis de prospectiva, planificación estratégica (análisis situacional) y espina de pescado.

**Productos:** a) Necesidades y problemas del cliente y b) Alcance del proyecto.

## 2.2 DESCRIPCIÓN DEL PROYECTO

**Actividades:** Se describe el proyecto a desarrollar en términos del alcance establecido. Esta descripción debe ser clara y sencilla, con la finalidad de que los desarrolladores puedan tener una visión general del sistema a construir sin mayores complicaciones. Debe ser una especie de metáfora (tal cual como se define el Programación Extrema), en la cual se indica el fin del sistema y de su arquitectura general.

**Observaciones:** En la descripción del sistema se recomienda la participación de todas las personas que definieron el alcance del proyecto, esto permitirá contemplar diferentes perspectivas sobre la arquitectura o visión general del sistema, y así poder definir una descripción que abarque los puntos de vista más importantes en términos del alcance del proyecto y de las necesidades y problemas del cliente.

**Responsables:** los mismos del paso anterior.

**Insumos:** Para la descripción del sistema a desarrollar se requiere el alcance del proyecto, así como las necesidades y los problemas del cliente.

**Producto:** Descripción del proyecto.

## 2.3 DEFINICION DE ACUERDOS RELACIONADOS CON LA LIBERACION Y USO DEL CODIGO FUENTE

**Actividades:** Definir y establecer acuerdos referidos a la liberación y uso del código fuente, así como la liberación de la documentación asociada al mismo.

**Observaciones:** Los acuerdos relacionados a la liberación y

uso del código fuente son establecidos entre el cliente y la organización desarrolladora. Estos acuerdos constituyen uno de los elementos más importantes para el desarrollo de software libre, ya que el desarrollo colaborativo depende del grado de accesibilidad del código y de la documentación asociada al mismo. En el desarrollo colaborativo de software participa cualquier persona interesada que tenga acceso al código y a su documentación.

Responsables: a) Cliente/usuario y b) responsable de la Gestión de Proyectos.

Productos: Acuerdos relacionados a la liberación y uso del código fuente.

#### 2.4 PRIORIZACIÓN DE LAS PRINCIPALES FUNCIONALIDADES DE LA APLICACIÓN

Actividades: Identificar y clasificar las principales funcionalidades de la aplicación, en función de las prioridades de implementación que requiera el cliente. Cuando se clasifican las funcionalidades se debe considerar el orden de dependencia entre estas.

Responsable: a) Cliente/usuario y, b) desarrolladores.

Observaciones: El cliente/usuario debe indicar el orden de dependencia entre las funcionalidades, el cual debe ser validado por los desarrolladores.

Producto: Priorización de las funcionalidades.

#### 2.5 PRIORIZACION DE LOS RIESGOS

Actividades: Identificar, priorizar y asociar los riesgos a las funcionalidades de la aplicación (sistema).

Observaciones: Los riesgos se priorizan según su impacto en el desarrollo de las funcionalidades de la aplicación. Es conveniente que los riesgos sean identificados por un grupo de personas con conocimiento en el proyecto a desarrollar. De esta manera se contemplarán gran cantidad de riesgos identificados desde diferentes perspectivas.

Responsable: a) Cliente/usuario, b) responsable de la Gestión de Proyectos, c) administradores de proyectos, d) desarrolladores, e) personas interesadas en el proyecto y, f) asesores.

Producto: Priorización de riesgos.

#### 2.6 PLANIFICACIÓN DEL DESARROLLO

Actividades: Elaborar o actualizar el plan general del proyecto. En este plan se debe indicar el número de iteraciones y la fecha de realización de las mismas, así como las funcionalidades a implementar en cada iteración, y las horas hombre requeridas para el desarrollo. Para poder indicar el orden de desarrollo de las funcionalidades, se requiere la clasificación de las mismas de acuerdo a la prioridad establecida por el cliente/usuario y la prioridad del riesgo asociado a cada funcionalidad; así como el orden de dependencia entre las mismas.

Observaciones: El plan general del proyecto puede ser actualizado o modificado en caso de que: a) queden funcionalidades de iteraciones anteriores sin implementarse, o funcionalidades que no han superado las pruebas de aceptación; b) surgen nuevas funcionalidades o cambios en las mismas y; c) aparecen nuevos riesgos o cambios en las prioridades de los riesgos existentes.

Responsable: Responsable de la Gestión de Proyectos.

Insumos: a) Priorización de funcionalidades y, b) priorización de riesgos.

Producto: Plan general del proyecto.

#### 2.7 ELABORACION DE LA OFERTA DE SERVICIO

Actividades: Elaborar y validar la oferta servicio. Se sugiere el siguiente contenido para este documento: a) propósito de la oferta, b) alcance y descripción del proyecto, c) fecha de entrega de versiones, d) equipo de trabajo y horas hombre requeridas para el desarrollo del sistema, e) costos del proyecto y, f) la plataforma de operación del ambiente real de la aplicación.

Observaciones: El cliente es el responsable de validar la oferta de servicio, mientras que el representante de la Gestión de Proyectos debe realizar las modificaciones sugeridas durante la validación de la misma.

Responsables: a) Cliente/usuario y, b) Responsable de la Gestión de Proyectos.

Insumos: a) Alcance del proyecto y, b) descripción del proyecto.

Producto: Oferta de servicio.

### 3. ADMINISTRACIÓN DEL DESARROLLO

La Administración de Proyectos Específicos se encarga de dirigir el grupo desarrollador responsable de una aplicación de software determinada (ver Figura 3). En este sentido, cada proyecto de software tiene un administrador del proyecto, quien se encarga de planificar y organizar las actividades respectivas a cada una de las iteraciones contempladas en el plan de desarrollo. De igual manera, el administrador del proyecto tiene la responsabilidad de: a) administrar la plataforma de desarrollo colaborativo (la cual permitirá apoyar las actividades propias del desarrollo del software libre), b) gestionar la configuración del sistema, c) dar seguimiento y control al proyecto, d) definir los estándares de codificación, e) asegurar la calidad de la aplicación y, f) administrar subcontratos. Esta última actividad no ha sido contemplada aún en la metodología utilizada en la FSL.

#### 3.1 ADMINISTRACIÓN DE LA PLATAFORMA DE DESARROLLO COLABORATIVO

Actividades: La plataforma de desarrollo colaborativo es seleccionada e instalada, y luego utilizada para publicar las versiones de la aplicación y los documentos asociados a las

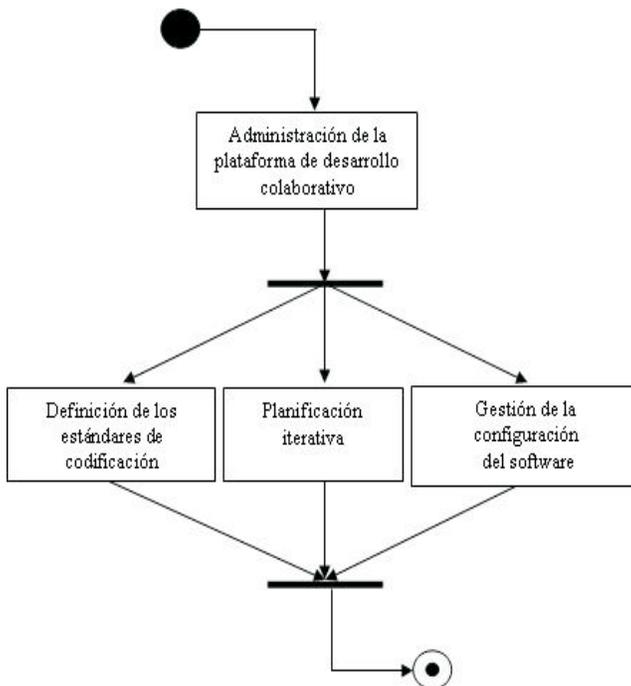
mismas obtenidos en cada iteración. La misma debe ser, preferiblemente, una herramienta computacional que facilite el desarrollo de software que involucra múltiples grupos de programación. Esta herramienta debe manejar un servidor de versiones, correo electrónico, listas, foros de discusión, planificación de actividades, presupuesto, recursos humanos involucrados en el proyecto, entre otros. Además, debe permitir la organización de la información, ayudando a colocar el software accesible al público que lo requiera y que sea autorizado por el equipo que lleva el proyecto, siguiendo los acuerdos de liberación y uso de código fuente.

Responsables: Principalmente: a) Administradores de proyectos; pero en la selección de la plataforma de desarrollo colaborativo también participan: b) Cliente/usuario, b) Responsable de la Gestión de Proyectos, d) Desarrolladores, e) Asesores, f) Demás interesados en el proyecto.

Insumo: Versiones de cada iteración.

Producto: Versión publicada.

Fig. 3: Flujo de trabajo de la Administración del desarrollo.



### 3.2 DEFINICIÓN DE LOS ESTÁNDARES DE CODIFICACIÓN

Actividades: Establecer los estándares de codificación que se utilizarán durante el desarrollo, y elaborar el documento de estándares. Esta actividad permite que cada grupo que colabora en el desarrollo entienda con mayor facilidad el código fuente, y a su vez, los aportes de los demás grupos.

Observaciones: Se recomienda establecer los estándares de codificación de manera que permitan la lectura rápida y simple del código, facilitando así el trabajo conjunto. Este documento debe contener las reglas de codificación

consideradas para el proyecto.

Responsables: Administrador del proyecto.

Insumo: Reglas de codificación consideradas

Producto: Estándares de codificación.

### 3.3 PLANIFICACIÓN ITERATIVA

Actividades: Planificar el desarrollo de las funcionalidades asociadas a cada iteración considerada en el plan de desarrollo.

Observaciones: Al culminar una iteración se planifica la siguiente iteración. Se consideran las funcionalidades asociadas a la iteración planificada y las funcionalidades de iteraciones anteriores aún no implementadas, o que aún no han pasado satisfactoriamente las pruebas de aceptación.

Responsables: Administrador del proyecto.

Insumo: Plan de desarrollo, Metodología de Desarrollo de Software.

Producto: Plan de la iteración.

### 3.4 GESTIÓN DE LA CONFIGURACIÓN DE SOFTWARE

Actividades: Incorporar a la configuración del software los productos y documentos, validados y aprobados, que conforman la aplicación desarrollada.

Observaciones: En este caso, se puede definir un "sitio" dentro de la plataforma de desarrollo colaborativo, que pueda utilizarse para agrupar los productos y documentos de software que se generen durante el desarrollo de una aplicación.

Responsables: Administrador de proyectos.

Insumo: Productos y documentos del software obtenidos en cada una de las fases de desarrollo.

Producto: Configuración del software.

### 3.5 CONTROL Y SEGUIMIENTO DEL PROYECTO

Actividades: Recolectar los reportes de pruebas aplicadas a la versión de la aplicación obtenida en cada iteración. Determinar en base a los reportes de pruebas, si la versión obtenida en la iteración respectiva puede ser publicada en el control de versiones.

Observaciones: La versión puede ser publicada si a superado las pruebas que corresponden.

Responsables: Administrador de proyectos.

Insumo: Reportes de pruebas.

Producto: Reportes de evaluación y seguimiento.

#### 4. CONCLUSIONES

En el presente trabajo se ha presentado una forma alternativa de hacer gestión en el desarrollo de software para el sector público. Estos elementos han sido ya implementados en la FSL, donde se ha desarrollado e implementado una metodología propia de desarrollo de software libre que integra particularidades, ventajas y beneficios de diferentes metodologías, métodos y modelos de desarrollo de software conocidas. Actualmente, se han automatizado un número significativo de procesos de la gestión de proyectos de la FSL, y se están haciendo desarrollos que permitirán integrar en un corto plazo las diferentes etapas de la fase de desarrollo de sistemas.

La gestión de desarrollo de software presentada está orientada hacia el servicio público visto de manera global o sistémica, participativa, y con estándares abiertos y libres. Todo ello siguiendo los lineamientos seguidos por el Estado Venezolana tendientes al beneficio colectivo, como la búsqueda del desarrollo endógeno y la soberanía tecnológica.

La gestión es sistémica en cuanto toma en cuenta la organización que recibe el servicio de manera integral, haciendo énfasis en la necesidad o en el problema general, más allá de la sola atención a los asuntos informáticos. Para ello, la gestión se vale de herramientas como la planificación estratégica y la prospectiva. Adicionalmente, el proceso es acompañado por iniciativas como la ASL que contribuye a cerrar brechas en cuanto a las dificultades para migrar hacia el uso software libre. La ASL garantiza el entrenamiento a los usuarios desde que un primer prototipo se encuentra operativo. Tal entrenamiento se aprovecha para validar algunos elementos del sistema y requerimientos que previamente han sido levantados.

La gestión es participativa dado que involucra a diferentes miembros de la organización que presta el servicio, así como de la que recibe el servicio, a diferentes niveles y con diversos roles y responsabilidades. Por otro lado, el modelo de gestión presentado considera diferentes aspectos fundamentales para el desarrollo de software libre (ambiente colaborativo de desarrollo, liberación del código, etc.).

Finalmente, es bueno acotar que la metodología donde se ha implementado el modelo de gestión presentado en este artículo [10], propone a la vez un proceso de desarrollo de software iterativo e incremental, que involucra a los usuarios desde los primeros pasos del desarrollo hasta las pruebas del sistema.

#### 5. REFERENCIAS

- [1] Decreto con rango y fuerza de Ley Orgánica de Ciencia, Tecnología e Innovación, publicada en Gaceta Oficial N° 37.291 de fecha 26 de septiembre de 2001.
- [2] Ley Orgánica de Telecomunicaciones. Publicada en Gaceta Oficial N° 36.970, de fecha 12 de junio de 2000.
- [3] Plan Nacional de Telecomunicaciones.
- [4] Decreto N° 3.390, Publicado en Gaceta Oficial N° 38.095, de fecha 28 de diciembre de 2004.
- [5] Adalberto Zambrano Barrios. Gerencia Estratégica y

Gobierno. Ediciones IESA. 2001.

- [6] Jose Aguilar, Oswaldo Terán y William Morantes, Prospectiva Tecnológica, Fundacite-Mérida (Ed.), Mérida, 2006.
- [7] Ken Beck, Extreme Programming Explained: Embrace Change, Second Edition, Addison Wesley Profession, 2004.
- [8] Philippe Kruchten, The Rational Unified Process An Introduction, Second Edition, Addison-Wesley, 2000.
- [9] Oktaba Hanna et al., Modelo de Procesos para la Industria de Software (MoProSoft, Versión 1.3), 2005 (<http://www.software.net.mx>).
- [10] Johanna Alvarez, Blanca Abraham, Jose Aguilar, Oswaldo Terán, Metodología para el Desarrollo de Software Libre: Buscando el Compromiso entre Funcionalidad y Riesgos, Reporte Técnico 001-2006, Fábrica de Software Libre, Fundacite, Merida, 2006.
- [11]<http://biblioweb.sindominio.net/telematica/catedral.htm>
- [12] Carlos Matus y Adalberto Zambrano, Gobierno y Planificación, Guía de Análisis Teórico, IESA Centro Zulia, 1997.

