

Enfoque ASD-DSDM: Nuevo marco para el desarrollo de software bajo filosofía ágil

Juan José Oliveira Maurera^a, José Esteban López Hernández^b

^a Departamento de Ingeniería de Sistemas, Unidad de Cursos Básicos, Universidad de Oriente, Av. Universidad Sector los Guaritos, Monagas, Venezuela
joliveira@udo.edu.ve, juanoseoli@hotmail.com

^b Departamento de Ingeniería de Sistemas, Unidad de Cursos Básicos, Universidad de Oriente, Av. Universidad Sector los Guaritos, Monagas, Venezuela
joseesteban15@hotmail.com

Resumen. El objetivo principal de esta investigación se focalizó en el desarrollo de una propuesta metodológica innovadora combinando los preceptos de los marcos metodológicos: desarrollo de software adaptativo (ASD) y el método de desarrollo de sistemas dinámicos (DSDM) cumpliendo los principios del manifiesto ágil, a fin de canalizar la construcción de un software para la gestión de equipos, es decir el manejo de reportes de fallas, inventario e indicadores de disponibilidad en la Superintendencia de Automatización y Control de la Empresa Mixta PDVSA Petrodelta S.A., aportando una significativa trascendencia tanto el ámbito científico por proponerse un nuevo esquema de abordaje en el proceso de elaboración de un software a la medida y para cualquier área empresarial, en otra palabras, se obtiene un marco metodológico concentrado en las necesidades organizacionales pero sin olvidar especificaciones de calidad en términos académicos y técnicos en el abordaje del proceso de desarrollo de un software; y a nivel empresarial resulta trascendente en este caso de estudio la aplicabilidad del proceso metodológico propuesto trayendo como el resultado axiomático de dicho proceso la entrega de un software operativo y de calidad que satisfaga las necesidades de la Superintendencia de Automatización y Control.

Palabras Clave: ASD, DSDM, desarrollo de software, metodologías ágiles, tecnología.

1 Introducción

Una organización, empresa, institución, o cualquiera sea el vocablo con el que se les denomine, son entes sociales que agrupan personas con el propósito de alcanzar objetivos determinados, al mismo tiempo que interactúan con el ambiente en el que se ven inmersas, razón que le acredita el término de sistemas abiertos. Existen organizaciones de diversas índoles, pero todas siguen una misma premisa: generar beneficios para sí y para los individuos que la conforman en un marco de respeto y responsabilidad para con las reglas que la supeditan en los ambientes donde se encuentren.

En el ámbito de la ingeniería del software, un sistema de información constituye una herramienta tecnológica de gran valor con funcionalidad y aplicabilidad a

variadas situaciones y entornos. En los inicios de la informática este tipo de sistemas ayudaron a llevar un mejor control de tareas simples; a medida que los equipos de cómputo evolucionaron y la computación se comenzó a transformar en ciencia, los sistemas de información comenzaron a volverse más sofisticados dando soporte, facilitando y automatizando un sin número de tareas que anteriormente eran llevadas a cabo de manera manual por las personas.

Con la aparición del internet y los conceptos de computación distribuida, comenzó el auge a nivel mundial de una nueva tendencia, como es el uso de arquitecturas cliente-servidor para las aplicaciones de cómputo. Bajo este enfoque también nacen las denominadas aplicaciones web (webapps). Ahora bien, una aplicación bajo dicho entorno, consiste en la ejecución de dos programas uno en un equipo llamado cliente con el que interactúa el usuario y que se encarga de hacer solicitudes a otro equipo conocido como el servidor, en cual se encuentra el corazón de la aplicación; encargado este último de generar la información que es devuelta al cliente para ser recibida por el usuario que la requiere.

Dado esto por sentado, en entornos como el ya citado donde la calidad del software es una premisa, pero a la vez los tiempos de desarrollo deben ajustarse para dar cumplimiento a funcionalidades a usuarios en cortos intervalos de tiempo; parafraseando al autor de reconocidas publicaciones en el área de la ingeniería del software, Pressman, [1], se puede expresar que en la economía moderna, a menudo resulta difícil o imposible predecir cómo evolucionarán con el tiempo los sistemas basados en computadoras y los ingenieros de software deben ser tan ágiles como para responder a un ambiente de negocios fluido que da origen a los cambios que experimentan dichos sistemas, resulta apropiado entonces tomar en consideración la filosofía propuesta por el manifiesto ágil cuando dice: “Individuos e interacciones sobre procesos y herramientas, software funcionando sobre documentación extensiva, colaboración con el cliente sobre negociación contractual y respuesta ante el cambio sobre seguir un plan”.

Es por ello, que se propone a manera innovadora una combinación de metodológicas en el área de la tecnología de la información como los son Adaptive Software Development (ASD) y Dynamic Systems Development Method (DSDM) traducidas al español como Desarrollo de Software Adaptativo y Método de Desarrollo de Sistemas Dinámicos respectivamente, las cuales son consideradas por Amaro y Valverde [2] y Carvajal [3], como dos metodologías ágiles con amplio reconocimiento las cuales forman parte de la mundialmente reconocida “Agil Alliance”, organización encargada de promover los principios y prácticas del desarrollo ágil y por tanto van acordes a la filosofía del manifiesto, considerado por este proyecto como una forma de pensamiento que promueve la calidad y la rapidez al mismo tiempo apoyando un desarrollo mancomunado por todos los interesados en el software, mas allá del solo equipo de desarrollo.

La ASD se selecciona por ofrecer un marco de tres (3) etapas en las cuales se funden las cinco (5) fases de DSDM que guiarán las diferentes actividades del proceso de desarrollo optándose por esta combinación donde DSDM guía las actividades internas de las etapas, ya que tal como lo cita DSDM Consortium [4], DSDM plantea un ciclo de vida para el desarrollo de proyectos en cinco fases: estudio de factibilidad; estudio del negocio; iteración del modelo funcional; iteración de diseño y construcción, y finalmente implementación, con un notable grado de clarificación en objetivos a cumplir en cada una, roles del proyecto y productos que se deben generar; además de que DSDM constituye un framework de desarrollo en contraposición a ASD que se presenta más como una concepción epistemológica.

Ahora bien al abordar la problemática empresarial, se observa que la Superintendencia de Automatización y Control de la empresa mixta PDVSA Petrodelta S.A., dedicada esta a la explotación de hidrocarburos, realiza en sus funciones diarias ciertos procesos y actividades, dentro de los que se encuentran el llenado de reportes de las fallas que pueda presentar la plataforma a su cargo, el seguimiento y administración de sus equipos y sistemas y el cálculo del indicador de disponibilidad porcentual de su plataforma. Estos procedimientos son llevados de manera semi-manual mediante el uso de formatos del programa Microsoft Word para el reporte de las fallas, y los más complejos, que exigen operaciones de cálculo, con libros de Microsoft Excel.

En una primera instancia esa forma de operación genera improductividad por el tiempo que se pierde al ejecutar dichas tareas de forma tan rudimentaria para una empresa en crecimiento, en segundo lugar reducen la eficiencia de la Superintendencia, y finalmente se corre el riesgo de la pérdida de la información crítica, pues al estar todo dividido entre diversas hojas y libros es difícil saber cuál es la versión más reciente, además se dificulta el aseguramiento de la seguridad lógica de las mismas.

Todo lo expuesto degeneraría en consecuencias como la pérdida de horas hombres y aumentos considerables de trabajo, bien sea por el tiempo desperdiciado en dichos procesos en si mismos, o en la nuevas tareas que se deben llevar a cabo para solucionar los errores o problemas derivados de estos.

La utilidad que proporcionaría una webapp construida bajo los enfoque de sistemas de información transaccional daría origen a una aplicación robusta y útil que apoye procesos fundamentales de una empresa; es por ello que se plantea para la Superintendencia de Automatización y Control de la empresa mixta PDVSA Petrodelta S.A. poner en marcha un plan de desarrollo único e innovador, pero suficientemente adaptativo para ser utilizado posteriormente en otro proceso de software independientemente del negocio de la organización, buscándose en este caso la construcción de una aplicación que llene el vacío originado por las necesidades de la Superintendencia en mejora de eficiencia y productividad en sus procesos de

gestión de equipos, bien sea para manejar los reportes de falla, la administración del inventario, y elaborar los indicadores de disponibilidad.

2 Metodología

Esta investigación está bajo la modalidad de proyecto factible el cual sirve como solución a un determinado problema; se dice que es factible puesto que asigna beneficios en algunas áreas específicas. En relación a esto, el manual de normas y procedimientos para la elaboración de trabajos de investigación de la UPEL [5], define la modalidad de investigación como la elaboración y desarrollo de una propuesta de modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales, “puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos” (p.28).

2.1 Tipo y nivel de la investigación

En atención a la taxonomía presentada por Hurtado [6] se clasifica este proyecto de investigación como un tipo de investigación conocida con el nombre de proyectiva bajo un nivel comprensivo considerado por la misma autora como aquel cuyos objetivos buscan explicar situaciones, predecir fenómenos y proponer alternativas de cambio. Al ser de este nivel se cumple con la definición de investigación proyectiva de Hurtado [6], “Este tipo de investigación propone soluciones a una situación determinada a partir de un proceso de indagación. Implica explorar describir, explicar y proponer alternativas de cambio, mas no necesariamente ejecutar la propuesta” (p. 114).

2.2. Diseño de la investigación

En este orden de ideas este proyecto plantea un diseño de investigación de tipo mixto, con una perspectiva temporal contemporánea con un corte transeccional y una amplitud que lo convierte en un diseño de caso. Para entender mejor el postulado anterior se hace referencia nuevamente a Hurtado [6], la cual en una de sus mas recientes publicaciones plantea un diseño de tipo mixto, el cual permite el uso tanto de fuentes vivas como documentales, considerando que los diseños son contemporáneos cuando la información obtenida corresponde a eventos actuales y a la vez son transeccionales si el investigador estudia el evento en un único momento del tiempo; y finalmente expresa que los diseños de caso son diseños multivariantes o multieventuales que centran la organización y análisis de información tratando el objeto bajo estudio como un todo.

2.3 Población y muestra

La población de estudio fue la Superintendencia de Automatización y Control de PDVSA Petrodelta S.A. y la misma estuvo conformada por los cuatro (4) empleados adscritos a dicha superintendencia. En cuanto a la selección de la muestra, haciendo referencia a Balestrini, [7] en donde plantea que la muestra solo se necesita en casos donde la población de estudio es demasiado compleja o el universo de estudio es relativamente grande, se resolvió para esta investigación conformar la muestra con la totalidad de la población, obviándose la necesidad de realizar un muestreo por método alguno.

2.4 Diseño operativo

A fin de alcanzar el objetivo general planteado en el proyecto, se utilizaron dos (2) metodologías: Adaptive Software Development (ASD) y Dynamic Systems Development Method (DSDM). La primera se maneja como una epistemología dentro de la cual se contempla el uso de la segunda (DSDM). ASD aporta tres etapas a nivel macro para el desarrollo del proyecto; en dichas etapas se incorporan las distintas fases de la metodología DSDM.

Se plantea este diseño metodológico de forma tal, que sea la metodología ASD la que regule los puntos de retroalimentación del proceso de desarrollo, puesto que DSDM plantea una retroalimentación en casi cualquier punto de la conducción del proyecto, lo que pudiera traer como consecuencia una anarquía que empaña el proceso de desarrollo.

En el esquema de la Figura 1, se aprecia que DSDM posee cinco (5) fases y dos (2) de éstas se dividen en sub-fases, que se distribuyen a lo largo de la filosofía ASD.

Por su parte, el enfoque ASD permitió que la retroalimentación se pueda dar desde cualquier sub-fase, pero sólo si éstas forman parte de la etapa de aprendizaje. También se puede apreciar en la Figura 1 el hecho de que no existe una fase o sub-fase que se llame implementación propiamente, a pesar que la versión de DSDM si la contempla.

La afirmación anteriormente expresada se debe a que la naturaleza de este caso de estudio consistió en entregar un sistema listo para su puesta en ejecución, más no se contempló las actividades inherentes a la implementación; es igualmente este motivo lo que fundamenta el hecho de que el nombre de implementación de la fase DSDM haya cambiado utilizando el descriptor “evaluación”, una vez mas connotándose que solo para este caso de estudio.

Estas modificaciones son válidas, pues tal como lo plantea ASD cada organización tiene necesidades y propiedades emergente diferentes, por lo tanto, es permisible que existan variantes en diferentes ciclos de desarrollo en diferentes proyectos donde se aplique esta nueva combinación metodológica; ninguna organización es exactamente igual a otra y mucho menos sus contextos de negocios son perfectamente análogos, por lo que no es recomendable plantear métodos rígidos para el análisis, diseño y desarrollo de sistemas, tal como si se tratara de una receta de cocina con ingredientes y pasos bien definidos e imprescindibles.

Lo que evoca este nuevo planteamiento es un dinamismo extremo para el diseño y desarrollo de software ágil caracterizado principalmente por el dinamismo operativo de sus etapas para lograr sus objetivo, un marco efectivo de desarrollo de sistemas de información.

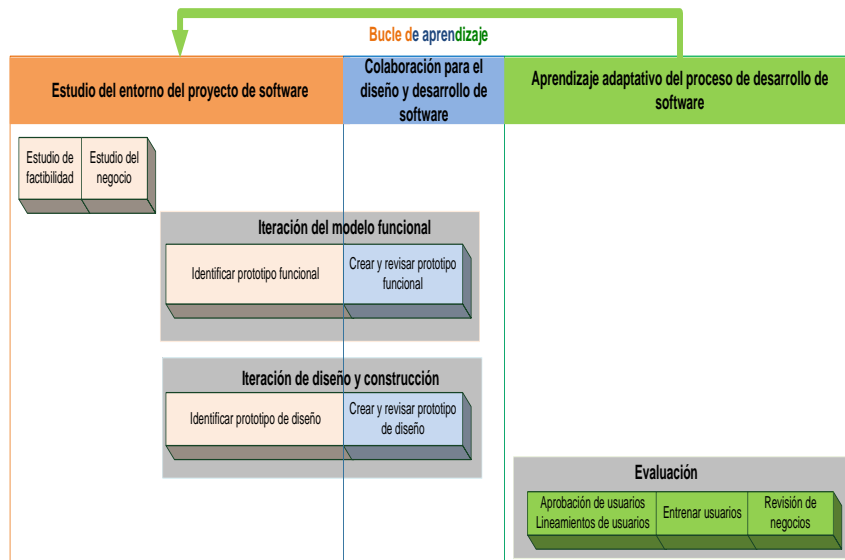


Fig. 1. Proceso metodológico combinado ASD-DSDM

Etapa I: Estudio del entorno del proyecto de software. Primera etapa del proceso de desarrollo, se fundamenta en la primera fase de la metodología ASD que es la especulación, como para ASD la especulación marca una planeación adaptativa del ciclo de desarrollo del proyecto, en esta etapa se llevaron a cabo las fases de estudio técnico, estudio del negocio, las sub-fases: identificar prototipo funcional y acordar plan de la fase iteración del modelo funcional, e identificar prototipo diseño y acordar plan de la fase iteración de diseño y construcción; siendo estas fases mencionadas pertenecientes a la metodología DSDM.

En una organización como PDVSA Petrodelta S.A. y más puntualmente la Superintendencia de Automatización y Control de la gerencia de AIT de dicha empresa, el utilizar un enfoque adaptativo que regule los estudio de negocio, levantamiento de requisitos y la planificación del proyecto, conlleva a que los planes de desarrollo tomen siempre en cuenta las necesidades de los usuarios del sistema, pues la dinámica de trabajo de la Superintendencia así lo requiere. Bajo el concepto de estudio de esta etapa se enmarcaron actividades como el conocimiento del entorno del negocio y la expectativas de estos para el software pues son ellos los que definen los objetivos del proyecto desarrollado y el posterior logro de dichos objetivos; así como también, aparecen en esta etapa el establecimiento de los roles de los diferentes participantes del proyecto, y finalmente la concertación de los requerimiento para con el nuevo software.

Etapa II: Colaboración para el diseño y desarrollo de software. Esta etapa recibe su nombre de la visión cultural que plantea ASD para el desarrollo de software donde la colaboración entre los diferentes roles involucrados en el proyecto conduce a la obtención de productos de software funcionales. Es por ello, que en esta etapa se encuentran reunidas las sub-fases de: Crear prototipo funcional, revisar prototipo y crear prototipo de diseño, revisar prototipo de diseño; correspondientes a las fases iteración del modelo funcional e iteración de diseño y construcción respectivamente; pertenecientes ambas fases la metodología DSDM.

Las actividades que se enmarcaron bajo este enfoque colaborativo fueron las propiamente orientadas a la obtención del producto de software, así como aquellas relacionadas a las pruebas y test que se llevan a cabo sobre los mismo para verificar su calidad, ejercicios propios de esta etapa consistieron en: Elaborar diagramas de funcionamiento del prototipo, modelar la arquitectura del prototipo, construir prototipo y probar el prototipo.

Es de destacar, que el modelado tanto funcional como arquitectónico del software se llevó a cabo en paralelo, y la creación y revisión de los prototipos a diferencia de DSDM que lo separa en dos sub-fases para cada fase macro, se planteó en una sola sub-fase para cada macro fase respectivamente; el objeto de este planteamiento es que por ser la Superintendencia un entorno de negocio dinámico, se aproveche el marco colaborativo que ofrece ASD y se realicen las pruebas del producto tan pronto sea posible, agilizándose los tiempo de desarrollo, sin necesidad de dividir este procedimiento en nuevas sub-fases que alarguen los tiempos de entrega del producto final.

El hecho de haber realizado estas actividades bajo una perspectiva colaborativa permitió la involucración de la mayor cantidad de interesados posible en el proceso de construcción, lo que para el caso de la Superintendencia bajo estudio fue altamente beneficioso, pues se logró que el producto final agrade a la totalidad de la plantilla de trabajadores de la superintendencia, pues todos ellos en el ejercicios de sus funciones necesitarán interactuar en algún momento con la herramienta.

Etapa III: Aprendizaje adaptativo del proceso de desarrollo del software. Para ASD el aprendizaje constituye la forma en que a medida que progresa un proyecto de software, los desarrolladores puedan aprender como grupo e individualmente, acerca de las necesidades de los usuarios del sistema, así como las expectativas que estos tienen del mismo, lográndose con ello revisiones de software en función de los requerimientos de los usuarios; además de ampliaciones y mejoras del software que se desarrolla. En esta etapa DSDM figura con la fase implementación renombrada como evaluación con tres sub-fases de la original: Aprobación de usuarios lineamientos de usuarios, entrenar usuarios, y finalmente revisión de negocios.

Como su nombre lo indica el aprendizaje es lo primordial. DSDM hace uso de este enfoque ordenado de aprendizaje que plantea ASD para así poder controlar de manera más coherente las iteraciones de desarrollos del proyecto y la subsecuente elaboración de los diferentes prototipos funcionales. Por lo que las actividades cubiertas por esta etapa se resumieron a: validar el prototipo con los usuarios en función de los requerimientos planteados, elaborar manual de uso del prototipo e identificar posibles cambios en la estructura del negocio.

Resulta indispensable un momento de aprendizaje en el desarrollo del proyecto. En las etapas anteriores se apreció un total involucramiento de los usuarios en los procesos de planeación y construcción, pero en el aprendizaje es donde el talento humano de la Superintendencia tuvo la oportunidad de evaluar el funcionamiento del software y verificar que éste se adapte a sus criterios de calidad, aprender su correcta operación, y lo más importante, determinar mejoras para iteraciones posteriores del proceso de desarrollo del software.

3 Resultados

A través de la aplicación de las diversas fases descritas por la operacionalización presentada en el marco metodológico, se obtuvo como resultado una herramienta de software luego de un proceso de dos iteraciones, la primera donde se obtenía una herramienta plenamente funcional y una segunda donde se hacían ajustes a la versión obtenida en la primera y se añadían nuevas funcionalidades a la vez. Los resultados mostrados en las líneas siguientes se desglosan en las tres etapas donde se resumen y compactan el desarrollo de las dos iteraciones, destacándose las modificaciones realizadas en la segunda aplicación del marco de desarrollo mediante alguna acotación para mejorar la presentación de los resultados.

3.1 Etapa I: Estudio del entorno del proyecto de software

El proyecto de software inicia en esta etapa con lo que se denominaría la mejor de todas las prácticas a la hora de desarrollar un software: estudiar de manera integral en entorno empresarial/institucional bajo el cual debe operar. En esta etapa se evalúa la viabilidad del proyecto, se define desde el sistema de negocios en cuestión hasta la identificación del prototipo a desarrollar junto con sus respectivos requisitos, pasando por establecimiento de objetivos y planes que conduzcan al posterior desarrollo del mismo de la mano con la cadena de valor asociada a la organización (ver Figura 1).



Fig. 1. Cadena de valor de los procesos fundamentales y de apoyo de la Superintendencia de Automatización y Control.

Estudio de factibilidad. En esta fase se inicia la primera etapa de desarrollo; con la aplicación de técnicas de recolección de datos a fin de lograr un panorama de las expectativas que tiene el negocio acerca del proyecto, y se elabora un análisis de factibilidad para su construcción, continuándose con el establecimiento de objetivos para el mismo, la elaboración del plan de trabajo para la iteración de desarrollo y la asignación de los roles.

En un primer abordaje las reuniones, encuentros y entrevistas permitieron conocer que como principal expectativa de la herramienta a elaborar, el personal esperaba la reducción del tiempo invertido en operaciones rutinarias y por consecuente un aumento de su productividad general.

Igualmente, la primera vista mostró un panorama de costos muy reducidos y numerosos beneficios, situación verificada al elaborarse el análisis factible correspondiente.

Con la factibilidad apoyando la realización del proyecto se establecen los objetivos y definen los roles de trabajo que guiaron las dos (2) iteraciones de desarrollo; por su parte el plan de trabajo fue adaptado en cada iteración de acuerdo a las actividades y necesidades de las mismas.

Estudio del negocio. El estudio del negocio se realizó a fin de conocer los procesos vitales de la Superintendencia, es decir, delimitar y conocer el entorno al que apoyará el software a desarrollar, para ello, se revisaron cadenas de valor y los diagramas de los procesos fundamentales y de apoyo de la misma, se definieron las funciones de los actores del negocio dentro de los procesos de interés, igualmente, se llevó a cabo el modelado de los casos de uso y diagramas de actividad para las tareas vinculadas a los procesos sobre los que incidiría la automatización propuesta, y esto pudiera servir a posteriori en el establecimiento de funcionalidades requeridas para con el software.

Reconocimiento de requerimientos. En función de los estudios del negocio previo y en conjunto con los futuros usuario del sistema o sus representantes, tal como lo establecen los roles prescritos por el marco metodológico DSDM, se determinaron los requerimientos que debía satisfacer el software y se les asignaron prioridades de acuerdo a la escala MoSCoW, la cual contempla cuatro (4) status: M, S, C y W; donde los que reciban prioridades M y S deben ser satisfechos en la iteración en curso, los C se abordaran siempre y cuando no conlleven a excesos de tiempo en el desarrollo y los W por su parte pasan a una lista de espera dejándose de lado, pero teniéndoles en cuenta para ser revaluados en futuras iteraciones. En la primera iteración se establecieron un total de 81 requerimientos; mientras que en la segunda fueron añadidos cuatro (4), para totalizar durante todo el proceso de desarrollo los 85 requerimientos.

3.2 Etapa II: Colaboración para el diseño y desarrollo de software

Es en esta etapa en donde se realiza propiamente la construcción del software, primero modelando paralelamente su funcionalidad y arquitectura, para continuar con su construcción y finalizar con una serie de pruebas que aseguren la calidad en la herramienta obtenida.

Modelado funcional. El modelado funcional del prototipo de software inició con establecimiento de los roles del sistema y los casos de usos del mismo además se representaba la actividad de los caso de uso mediante el uso de diagramas de actividad; y la interacción de estos mediante el uso de diagramas de secuencia, también se describe el esquema de navegación dentro del software pues por tratarse de una aplicación al estilo webapp resulta indispensable estructurar el mapa de navegación entre los diferentes despliegues por los que navega el usuario. Un ejemplo de parte del modelado funcional se muestra en la Figura 2 que corresponde al diagrama de casos de uso general del sistema, posteriormente en la Figura 3 se muestra el caso de uso validar usuario con su respectiva descripción (Cuadro 1) y diagramas de actividades (Figura 4) y de secuencias (Figura 5).

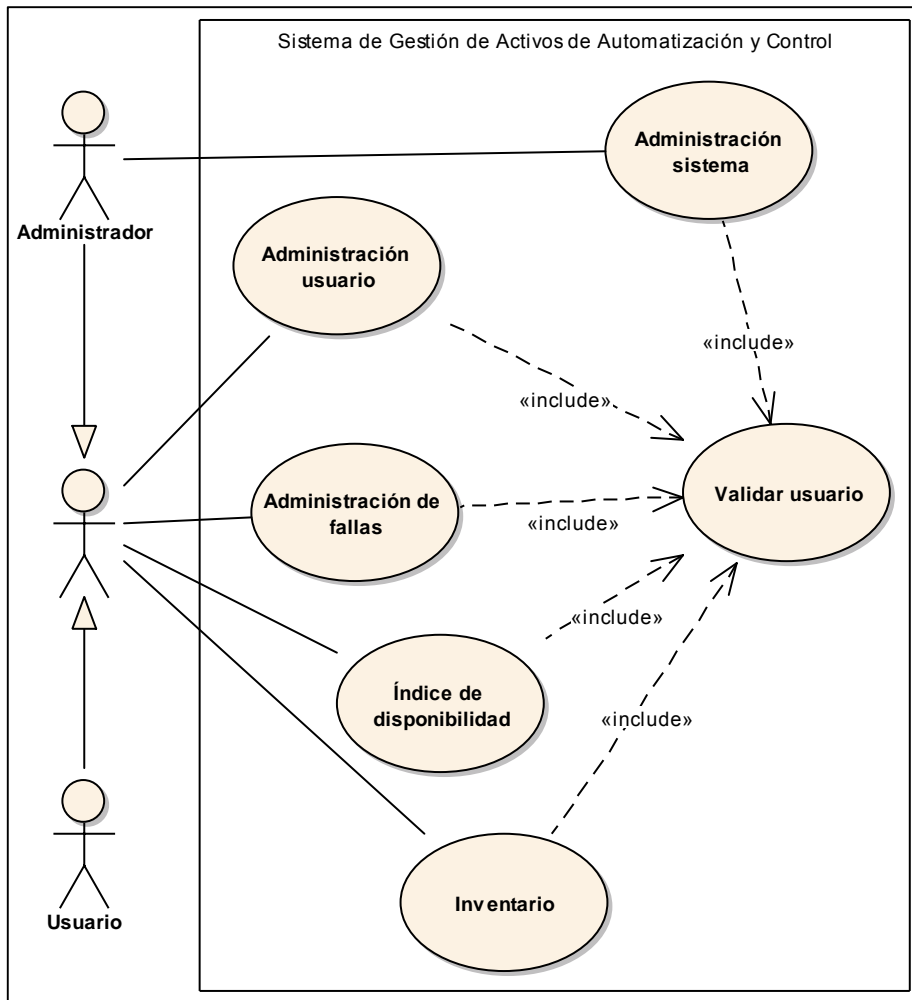


Fig. 2. Casos de uso del sistema: Diagrama de caso de uso general del sistema.

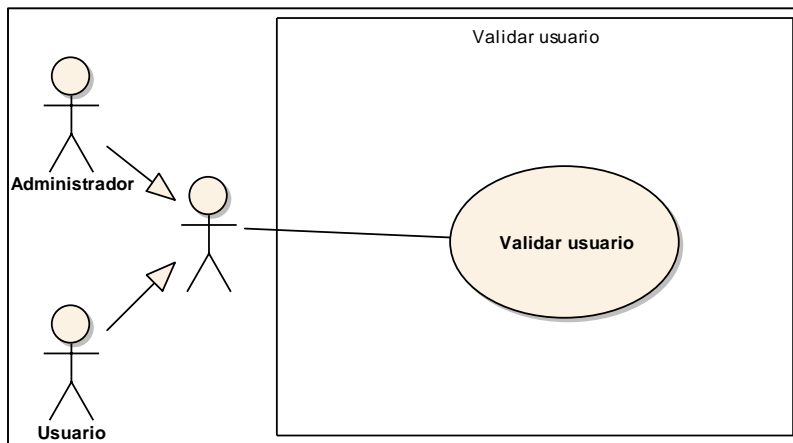


Fig. 3. Casos de uso del sistema: Validar usuario.

Cuadro 1. Descripción caso de uso del sistema: Validar usuario.

Caso de Uso	Validar usuario			CUS-001
Autor	xxxxxxx			
Actores	-Administrador -Usuario			
Tipo	Esencial			
Precondición	No aplica.			
Postcondición	-Administrador y/o Usuario con acceso al sistema concedido.			
Importancia	Vital	Urgencia	Alta	
Propósito				
Validar el usuario que desea ingresar al sistema.				
Resumen				
El Administrador y/o Usuario ingresan su identificador con su respectiva contraseña a fin de validar su registro y otorgarle permiso de acceso al sistema.				
Curso Normal (Básico)				
1	Administrador y/o Usuario: Iniciar aplicación.			
2	Administrador y/o Usuario: Ingresar identificador de usuario.			
3	Administrador y/o Usuario: Ingresar contraseña.			
4	Administrador y/o Usuario: Iniciar sesión.			
Cursos Alternos				
3a	Administrador y/o Usuario: Si identificador y/o contraseña incorrectos, regresar al paso 2 del curso normal.			
Comentarios: N/A				

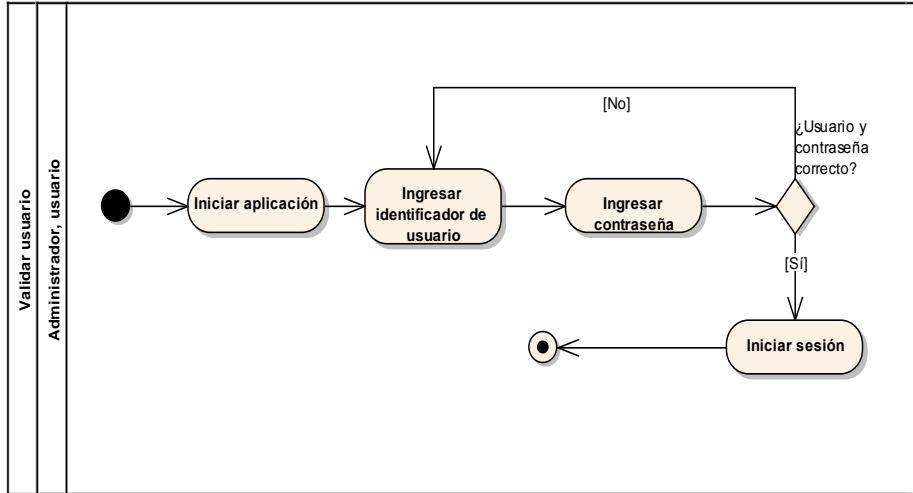


Fig. 4. Diagrama de actividad: Validar usuario.

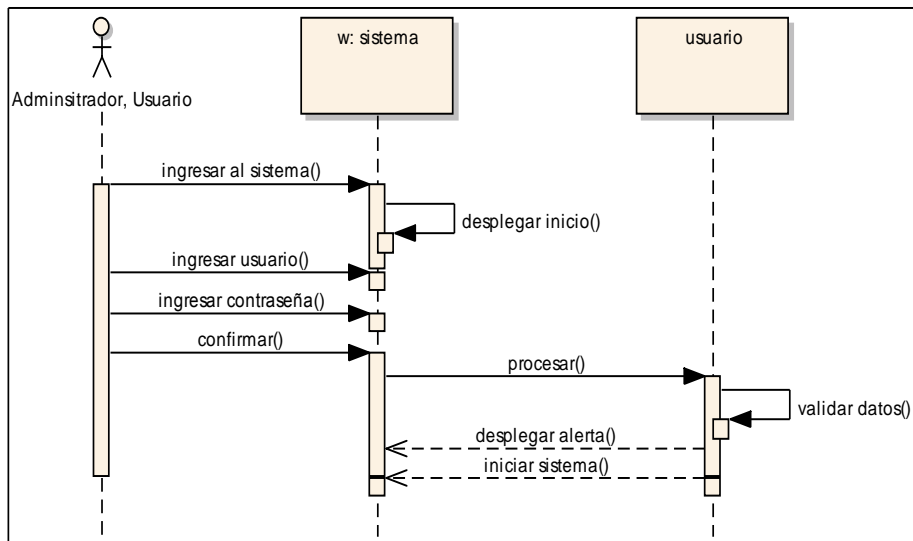


Fig. 5. Diagrama de secuencia: Validar usuario.

Modelado estructural. El modelado estructural comprendió la definición de la base de datos que soporta al sistema para esto se modelaron las tablas usando un diagrama de entidad-relación; se debe destacar que el modelado realizado en una primera iteración fue ampliado en la segunda para satisfacer nuevos requerimiento de datos realizados por los usuarios del sistema.

El modelado estructural también comprende la identificación de los diferentes subsistemas que conforman a la herramienta, así como los componentes de los mismos y por supuesto un modelo de despliegue del mismo; todo esto obedeciendo

siempre una arquitectura de tres capas: presentación, lógica del negocio y datos. Muestra de productos generados en esta fase es el diagrama de componentes para el subsistema “Índice de disponibilidad” que se aprecia en la Figura 6.

Para los diagramas de componente se debe resaltar que por tratarse de una webapp basada en una arquitectura de tres capas: presentación, lógica del negocio y datos, cada subsistema cuenta con componentes entre las diversas capas, por tal motivo en el diagrama de componente de cada subsistema. Se hace distinción de los componentes que pertenecen a distintas capas mediante el uso de colores y estereotipos; usándose: azul para la capa de presentación, beige para la de lógica del negocio y verde en la de datos.

En cuanto a los estereotipos: <<form>> denotado como un recuadro azul en la parte superior derecha representado vistas de presentación y captura de datos, <<ws client>> para vistas de la capa de presentación que usan recursos de la capa lógica, <<web service>> para procesos de la capa lógica y <<table>> para referirse a componentes que son es sí mismos tablas de la base de datos. En la Figura 7, se presenta el diagrama entidad relación correspondiente al diseño de la base de datos asociadas al proyecto de software.

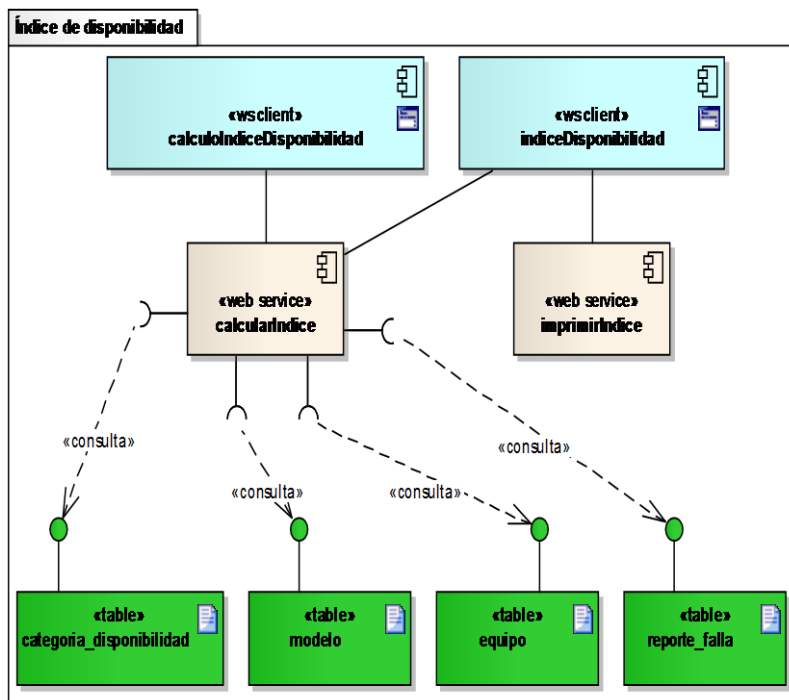


Fig. 6. Diagrama de componentes: Subsistema Índice de disponibilidad

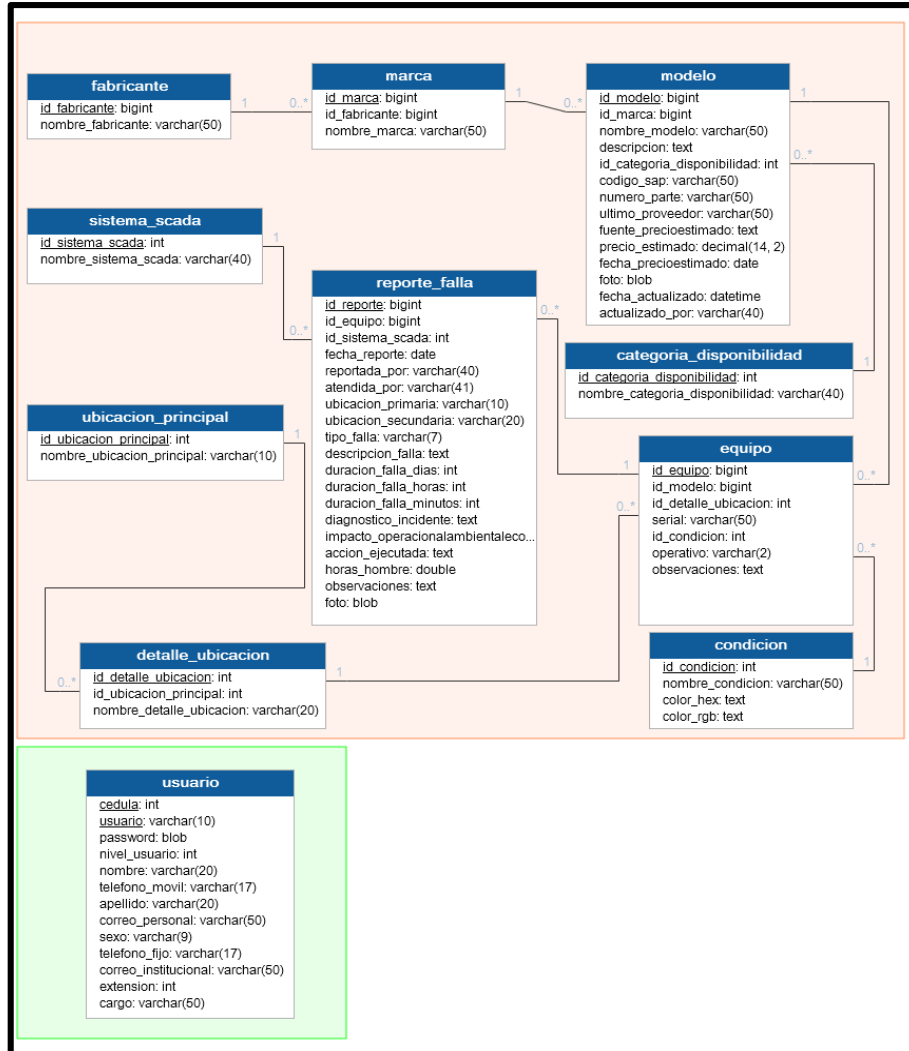


Fig. 7. Diagrama entidad-relación de la base de datos.

Construcción. En el proceso de construcción se produce el software modelado ya funcional y estructuralmente de acuerdo a los modelados realizados en cada iteración mediante las técnicas de codificación del lenguaje PHP y JavaScript, el primero presentado por Welling y Thomson, [8] como “un lenguaje de secuencia de comandos de servidor diseñado específicamente para la web” (p. 33) mientras que el segundo referido por Norton [9], como un lenguaje que se puede correr en prácticamente dentro de cualquier navegador moderno, ofreciendo por tanto una amplia versatilidad, para proporcionar ciertos comportamientos dentro de la aplicación, de la misma manera se hizo uso de SQL, y el diseño de patrones estéticos mediante hojas de estilo CSS, se debe notar que en la segunda iteración solo fueron realizadas mejoras a la

construcción realizada en la primera, que añadieron nuevas funcionalidades y la corrección de alguna falla que se haya encontrado en las pruebas.

Pruebas de prototipo

Las pruebas de validación del prototipo se realizaron tanto en la primera iteración como en la segunda, pues de esta forma se garantizaba que el desarrollo inicial así como las mejoras gozarán de niveles de calidad y de detectarse fallas estas se pudieran corregir, para no afectar el rendimiento del usuario. Las pruebas realizadas sobre la aplicación contemplaron: pruebas de contenido, pruebas de mecanismo de la interfaz para los vínculos y formularios, pruebas de sintaxis de navegación para los motores de búsqueda y pruebas de seguridad para la autenticación y autorización. Ambas se realizaron en las dos iteraciones, en la primera sobre todo el sistema, mientras que en la segunda solo sobre las nuevas funcionalidades añadidas o las del sistema que fueron modificadas u objeto de revisión por errores previamente detectados; es destacable que en la aplicación de la segunda iteración no se detectó la presencia de errores y el sistema alcanzó niveles ideales de calidad y seguridad.

3.3 Etapa III: Aprendizaje adaptativo del proceso de desarrollo de software

La etapa final de proceso de desarrollo toma su relevancia del hecho en que en ella se lleva a cabo en conjunto con los usuarios la validación del sistema obtenido, se establecen los lineamientos para su uso, se elaboran los manuales correspondientes y se instruye o capacita al personal de la Superintendencia para el manejo del sistema. Sin embargo, es destacable que en esta etapa se verifica, si el negocio tienen nuevas necesidades, es decir, si los usuarios desean nuevas funcionalidades para el sistema, y se procede con base a ello a ejecutar una nueva iteración que permita mejorar el sistema y dar cumplimiento a los nuevos planteamientos.

Validación conjunta. En el proceso de validación con los usuarios se evidenció el grado de satisfacción de los mismos respecto al software; puesto que todos los requerimientos indispensables de los mismos en la primera iteración habían sido satisfechos y los errores encontrados eran menores, quedaron pendientes solo tres (3) requerimientos menores y la corrección de pequeños detalles. En la segunda iteración fueron añadidas las funcionalidades de los requerimientos pendientes, y se corrigieron los errores. Una vez terminada la segunda iteración los usuarios volvieron a encontrarse satisfechos con los resultados y se procedió a la elaboración del manual de operación del sistema; así como a un proceso de inducción a los usuarios en su uso.

Lineamientos de seguridad. Los lineamientos de seguridad estuvieron orientados en establecer recomendaciones respecto al uso y puesta en marcha de la aplicación entregada; fueron elaborados en la primera iteración y se mantuvieron iguales al cabo de la segunda.

Nuevos cambios y necesidades. Del proceso de validación conjunta de la primera iteración surgió la necesidad de algunos requerimientos nuevos para el software; los cuales junto con los pendientes de la iteración anterior dan inicio a un nuevo ciclo que se constituye en la segunda iteración de desarrollo. Para el segundo proceso iterativo

por su parte no hubo nuevos cambios o necesidades que ameritan modificaciones a nivel de software.

Una vez finalizado el proceso de construcción de esta nueva iteración se ejecutó una nueva sesión de validación y presentación del prototipo, a fin de verificar que se haya dado cumplimiento a los requisitos indispensables; la valoración arrojó como resultado que el sistema tuvo un grado absoluto de satisfacción de requerimientos y, por tanto, de las expectativas respecto al mismo, considerándose esta versión como un sistema a punto para la operación.

Finalmente, se culminó el proceso con la elaboración de un manual de operación de la herramienta y la respectiva realización de una sesión de adiestramiento con los usuarios, los cuales consideraron que la misma fue suficiente para adaptarse al manejo del sistema. Dicha sesión consistió en una explicación del uso de todas las funcionalidades del sistema, con una duración de cinco horas por persona divididas a lo largo de tres días; debido a que no se podía realizar con todos los usuarios el mismo día, sino que se debían turnar los usuarios a la hora de participar en el adiestramiento.

En un sentido más amplio es conveniente argumentar respecto al desarrollo de toda la investigación, que se propuso un esquema operativo inédito de gran valor técnico y académico, con resultados positivos para el diseño y desarrollo de software bajo la filosofía ágil, este enfoque metodológico integrado por las metodologías ASD-DSDM es utilizable para cualquier tipo de aplicación y entorno organizacional, proporcionando de esta manera, una herramienta altamente efectiva y flexible para el desarrollo moderno de software. Como referencia del producto desarrollado se muestra la pantalla de inicio (Figura 8) y una interfaz operativa de la aplicación (Figura 9).

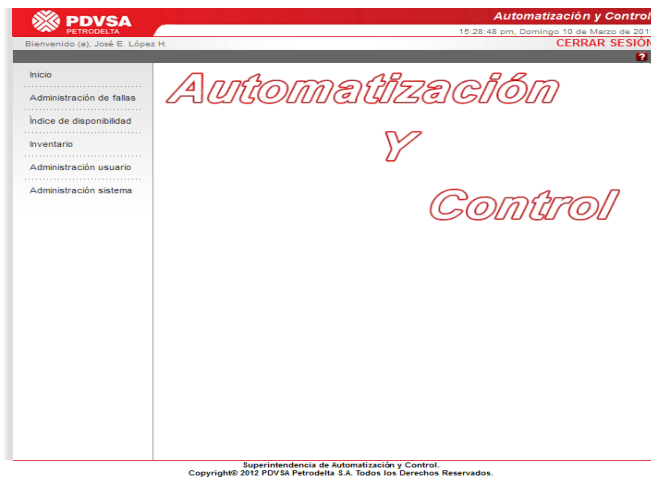


Fig. 8. Interfaz: Inicio de la aplicación

PDVSA
PETRODELTA

Bienvenido (a), José E. López H.

Automatización y Control
06:11:04 pm, Sábado 30 de Marzo de 2013
CERRAR SESIÓN

Listas PDF

Lista PDF por : fabricante - marca - ubicación

Para generar una lista PDF compacta de los equipos de un modelo en base a especificaciones de fabricante, marca y/o ubicación. Seleccione los campos necesarios a continuación y pulse el botón PDF; de no seleccionar campo alguno cuando presione el botón PDF se generara una lista compacta de todo el inventario.

Fabricante: ---Seleccione fabricante---
 Marca: ---Seleccione marca---
 Ubicación: ---Seleccione especificidad de la ubicación---

Lista PDF para cotización de procura

Para generar una lista PDF con fines de cotizaciones para procura; puede seleccionar marcas de equipos con todos sus modelos y/o modelos de equipos en específico. Cuando desee agregar todos los modelos de una marca pulse el botón "Agregar todos los modelos de la marca"; mientras que para agregar un modelo en específico pulse el botón "Agregar modelo". Una vez terminada su selección pulse el botón PDF.

Fabricante: ---Seleccione fabricante---
 Marca: ---Seleccione marca---
 Modelo: ---Seleccione modelo---

Agregar todos los modelos de la marca Agregar modelo

Lista PDF : Informe completo de inventario

Para generar una lista PDF con toda la información de inventario, pulse a continuación el botón de PDF.

Superintendencia de Automatización y Control.
Copyright © 2012 PDVSA Petrodelta S.A. Todos los Derechos Reservados.

Fig. 9. Interfaz de usuario: Listas PDF

Ventajas de la propuesta metodológica para el desarrollo de software. La investigación contempló la generación de un producto de software de calidad mediante la aplicación de metodologías innovadoras, efectivas, distintivas y certificadas para el área del desarrollo de software (ASD y DSDM), capaces de adaptarse al entorno organizacional, logrando productos de software preparados para evolucionar con el negocio (Superintendencia de Automatización y Control), permitiendo dar apoyo en la consecución de las metas así como en los objetivos del mismo: Diseñar, implementar, administrar y mantener la plataforma de automatización y control (conformada por equipos de supervisión, control, respaldo de energía, entre otros) asociada a los procesos operativos, para garantizar la continuidad operacional.

Tal adaptación metodológica, según obtenidos, permitió un ahorro de al menos un 33 % del tiempo dedicado al diseño y la construcción comparados con los procedimientos tradicionales (específicamente al utilizar otra metodología ágil como Programación Extrema por ejemplo según la experiencia de los desarrolladores internos de la organización). En este sentido, la concepción epistemológica pensada para el desarrollo de software y sistemas complejos ASD combinada efectivamente con una metodología flexible para el desarrollo de software ágil (ASDM) permite una estructura dinámica para construir sistemas mediante la realización de prototipos incrementales que cumplan restricciones ajustadas de tiempo con una amplia diversidad de requerimientos. Esta reducción considerable de la carga de trabajo y retrabajo por parte de los desarrolladores es otra gran ventaja que presenta esta nueva propuesta. En el cuadro 2, se presenta un resumen con la combinación metodológica y las ventajas obtenidas con esta metodología.

Cuadro 2. Resumen de resultados y ventajas obtenidas.

Etapa	Metodología	Actividades asociadas	Diferencia con otras Metodología
Etapa I Estudio del entorno del proyecto de software	ASD: Etapa I: Especulación DSDM: Fase I: Estudio de factibilidad Fase II: Estudio del negocio Fase III: Iteración del modelo funcional (Sub-fases: Identificar prototipo funcional, acordar plan) Fase IV: Iteración de diseño y construcción (Sub-fases: Identificar prototipo de diseño, acordar plan)	Determinar expectativas y objetivos del proyecto. Analizar la factibilidad del proyecto. Elaborar plan de trabajo general del proyecto. Asignar los roles del proyecto. Delimitar el sistema de negocios. Reconocer los procesos del negocio. Identificar los actores del negocio. Determinar requerimientos del prototipo.	Transcendencia otorgada al modelado del negocio en función a los requisitos de los usuarios del sistema
Etapa II Colaboración para el diseño y desarrollo de software	ASD: Etapa I: Colaboración DSDM: Fase III: Iteración del modelo funcional (Sub-fases: Crear y revisar prototipo funcional) Fase IV: Iteración de diseño y construcción	Elaborar diagramas de funcionamiento Modelar la arquitectura de datos asociada al prototipo. Modelar la arquitectura del prototipo. Construcción y pruebas	Participación efectiva de los usuarios en las iteraciones. Ahorro en tiempos de diseño y desarrollo.
Etapa III: Aprendizaje adaptativo del proceso de desarrollo de software	ASD: Etapa III: Aprendizaje DSDM: Fase V: Implementación (Sub-fases: Aprobación de usuarios lineamientos de usuarios, entrenar usuarios, revisión de negocios)	Validar el prototipo con los usuarios en función de los requerimientos planteados. Identificar posibles cambios en la estructura del negocio. Elaborar manual de uso del prototipo.	Validación progresiva de resultados según los requisitos exigidos por los usuarios; aprendizaje de posibles errores de diseño. Ahorro de tiempo de implantación. Reducción de resistencia al cambio.

4. Conclusiones

a) Se describió de manera clara el funcionamiento de la Superintendencia conociéndose sus procesos fundamentales; mediante el uso de reuniones y entrevistas, profundizándose en la comprensión del funcionamiento y necesidades de la misma.

b) El uso de diagramas para modelar las funciones del negocio de interés en la construcción de la herramienta, permitió establecer una clara identificación de los requerimientos que debía satisfacer la misma; a la par que estos eran priorizados.

c) Mediante un modelado funcional y uno arquitectónico, se estableció la funcionalidad que proporcionaría el software; así como la estructuración de los datos que manejaría, los subsistemas que lo conformarían y el mapa de despliegue que obedecería.

d) La ejecución de un criterio de pruebas permitió la minimización de fallas, asegurándose la calidad del producto, siendo las pruebas realizadas: pruebas de contenido, pruebas de mecanismo de la interfaz para los vínculos y formularios, pruebas de sintaxis de navegación para los motores de búsqueda y seguridad.

e) Los procesos de validación en conjunto con los usuarios aseguraron la entrega de un sistema que cumpla a cabalidad las expectativas de la Superintendencia y se constató la garantía que ofrece el proceso metodológico combinado en la generación de software funcional y de calidad.

f) El trabajo investigativo desarrollado con el caso de estudio presentado refleja la trascendencia científica y tecnológica de la propuesta, pues se obtuvieron mejoras significativas con el uso de esta combinación metodológica inédita, especialmente en lo que respecta a la reducción de tiempos de entrega e interacción usuario-desarrollador. Tal combinación representa una guía referencial para futuras investigaciones sobre desarrollo de software independientemente de su naturaleza.

Referencias

1. Pressman, R.: Ingeniería de software: Un enfoque práctico, 7ma edición, McGraw-Hill, México (2010)
2. Amaro, S. y Valverde, J.: Metodologías Ágiles, [en línea] <http://www.seccperu.org/files/Metodologias%20Agiles> [Consulta: 22 de Enero 2013]
3. Carvajal, J.: Metodologías Ágiles: Herramientas y modelo de desarrollo para aplicaciones, MSc. Tesis, Universidad Politécnica de Cataluña, España (2008)
4. DSDM Consortium, DSDM Public Version 4.2, [en línea], http://intra.iam.hva.nl/content/0708/propedeuse/ontwikkelmethoden_en_technieken/intro-en-materiaal/DSDM.pdf, [Consulta: 22 de Enero 2013]
5. UPEL: Manual para la elaboración de tesis de grado, maestría y doctorado, 5ta edición, FEDEUPEL, Venezuela (2012)
6. Hurtado, J.: El proyecto de investigación: Comprensión holística de la metodología y la investigación, 6ta edición, Quirón Ediciones & Sypal, Colombia-Venezuela (2010)
7. Balestrini, M.: Cómo se Elabora el Proyecto de Investigación, BL Consultores Asociados, Venezuela (2006)
8. Welling, L. y Thomson, L.: Desarrollo web con PHP y MySQL, 3ra edición, ANAYA Multimedia, España (2005)
9. Norton, P.: Introducción a la computación, 6ta edición, McGraw-Hill, México (2006)