

Implementación de un Sistema para el Control de Activos ISOPTEC, bajo el estándar ITIL y metodología ágil SCRUM

Mario May ,Yeni Morales ,Jorge Marrufo ,Mario Martín

M.May ,Y. Morales ,J.Marrufo y M.Martín

Universidad Tecnológica Metropolitana, Calle 115 (Circuito Colonias Sur) No. 404 x 50 en la Colonia Santa Rosa,Mérida, Yucatán

mario.may@utmetropolitana.edu.mx

M. Ramos.,V.Aguilera.,(eds.). Ciencias de la Ingeniería y Tecnología, Handbook -©ECORFAN- Valle de Santiago, Guanajuato, 2013.

Abstract

This paper aims to show the process of software development using an agile development method called Scrum to a fixed assets control system called ISOPTEC. The system is willing to become a standard for the first control of the institution's computer equipment for which it was made, and in the future intends to have more functions that will allow it to be used by higher education institutions, similar to our University and particularly to the ones that belong to the Coordination of Technological Universities.

17 Introducción

Siendo uno de los grandes retos de la administración pública y establecido en el plan nacional de desarrollo 2007 – 2012 del gobierno federal la realización de estrategias tendientes a simplificar la regulación que rigen a las instituciones, así como mejorar las políticas, normas y disposiciones de carácter general que emiten las instancias o instituciones dependientes de tres órdenes de gobierno.[1] Que las instituciones de gobierno cuentan con las atribuciones para organizar y coordinar el desarrollo administrativo integral de las dependencias y entidades de la administración pública, con el objetivo de que los recursos humanos, patrimoniales y los procedimientos y tecnologías aplicadas a su cuidado, sean aprovechados y aplicados con criterios de eficiencia[1], resulta importante la implementación de sistemas de software que mitiguen la desorganización y guíen el cuidado de los recursos patrimoniales de las instituciones públicas. Por lo anterior, resulta un reto interesante para las instituciones públicas, y en el caso que nos ocupa de las Universidades Tecnológicas, el contar con sistemas de información que coadyuven al cuidado de los recursos patrimoniales que con el fin de cumplir el objetivo académico se entrega en resguardo al personal administrativo y académico adscrito, entendiéndose por este, todos los equipos materiales e inmateriales (como software de computadora) que se utilizan cotidianamente y han sido adquiridos mediante recursos públicos destinados a la dependencia y que por ende forman parte del cuerpo patrimonial de la misma. El objetivo general de este artículo, es mostrar la importancia de aplicar controles a través de un sistema de software para administrar los recursos patrimoniales en las Universidades Tecnológicas, siendo el objeto de estudio inicial la Universidad Tecnológica Metropolitana, mostrando en especial la metodología de software utilizada, así como en el particular del implementado para controlar la maquila del software, mostrando los resultados obtenidos tanto desde un punto de vista tecnológico como de mejora en el proceso de general del control patrimonial.

17.1 Método

Al tratarse de un proceso de desarrollo de software, se ha optado por el seguir una estrategia metodológica afín a este tipo de implementaciones, por lo que el uso del SCRUM ha sido la elección; en cuanto a la funcionalidad de la aplicación, se ha decidido seguir el estándar que para activos fijos marca ITIL.

17.2 SCRUM

De acuerdo al SCRUM Primer[2] el modelo de desarrollo consiste en un marco de trabajo iterativo para el desarrollo de productos y proyectos. El trabajo se organiza en ciclos de trabajo denominados Sprints los cuales tienen las siguientes características:

1. Son iteraciones de 1 a 4 semanas y van sucediendo una detrás de otra.
2. Los sprints son de duración fija aunque en algunas ocasiones el trabajo no se haya completado.

Al comienzo de cada sprint el equipo de trabajo (Scrum Team) selecciona los elementos (historias) de una lista priorizada (Product Backlog) ,definida por el cliente (Product Owner), comprometiéndose a terminarlos al finalizar el Sprint. La regla primordial consiste en que no es posible cambiar los elementos elegidos una vez iniciado el Sprint.

Todos los días el Scrum Team se reúne brevemente para informar el progreso (Daily SCRUM) al jefe del equipo (SCRUM Manager) y actualizan unas gráficas sencillas (burndown charts) que les orientan sobre el trabajo restante. Al final del Sprint en presencia de todos los involucrados en el proyecto se expone todos los avances realizados los cuales consisten en componentes funcionales de la aplicación. Para ilustrar este proceso se puede ver la imagen de la figura 17.

Figura 17 Infografía de la metodología SCRUM



Se optó por emplear esta metodología debido a que el proyecto presenta las siguientes características:

- El equipo de desarrollo es pequeño (4 personas).
- Debido a que es un desarrollo interno de la institución es posible tener un contacto frecuente con el cliente el cual se ha demostrado ser clave para el éxito del proyecto.
- Por motivos operativos es necesario presentar incrementos funcionales en corto tiempo ya que el área necesita del apoyo de las herramientas que el sistema brindará.

17.2.1 Implementación de la metodología SCRUM

El proyecto está conformado por 6 ciclos de desarrollo (sprints) de duración de 3 semanas. La configuración del release quedó de la siguiente manera:

Sprint 1. El sistema deberá de contar diferentes catálogos de ítems de la base de datos, debe de tener la capacidad de contar con un esquema de base de datos normalizado y módulos de administración de esta información (altas, bajas y cambios).

Sprint 2. El sistema deberá de contar con un mecanismo de seguridad basado en roles, implicando la creación de usuario y roles del sistema.

- Registro de usuarios del sistema.
- Registro del personal relacionado con la Universidad Tecnológica Metropolitana.

Sprint 3. El sistema deberá de contar con un módulo para el registro de activos, así como la capacidad para asociar éstos al personal de la institución.

- Control de los activos en relación con las áreas controladas y no controladas por el departamento de activos fijos.
- Grupo y Subgrupo de activos.

Sprint 4. El sistema deberá de tener un módulo para el registro de préstamos de activos entre el personal.

- Préstamos de Activos entre áreas de trabajo.

Sprint 5. El sistema deberá de tener un módulo para el registro de traspasos ,devoluciones para complementar el proceso de movimientos de activos de la institución.

- Traspasos, devoluciones de activos.

Sprint 6. El sistema deberá de contar con reportes para obtener indicadores del proceso de control de activos fijos, además los reportes en algunas actividades del sistema fungirán como documentos oficiales de la institución.

- Reportes de los procesos del sistema.

Dentro de las historias descritas en el product backlog, de manera representativa se presenta en la tabla 17 un ejemplo.

Tabla 17 Ejemplo del uso de la metodología SCRUM a partir de una historia del Product Backlog

Prioridad	Descripción	Estimación
1	Yo como supervisor deseo registrar los préstamos de activos de la institución para saber en todo momento que activo ha presentado movimientos y a quien se le dio.	50
2	Yo como administrador quiero crear usuarios en la aplicación para establecer los controles de seguridad en las actividades del sistema.	10
3	Yo como supervisor deseo generar reportes de los activos fijos asignados al personal para que en caso de auditoría se le proporcione al personal su hoja de resguardo de activo fijo.	30

Uno de los artefactos claves de la metodología Scrum es la gráfica de progreso o de quemado (burning chart) ,en ésta se ilustra el progreso del equipo de desarrollo expresado en horas. El eje de las X corresponde a los días de desarrollo y en el eje de las Y corresponde el número de horas faltantes por concluir con el objetivo del sprint, lo ideal, es aproximarse a una relación equivalente entre el aumento de los días trabajados y las horas faltantes En la imagen de la figura 2 se presenta la gráfica de quemado para el Sprint 1, la cual orienta el cumplimiento del programa original de desarrollo.

Figura 17.1 Ejemplo del Burning Chart del Sprint 1



17.2.2 ITIL

Como parte de la estrategia de registro de activos fijos se tomó como referencia la librería ITIL[3] en específico el proceso de Configuration Management que tiene como finalidad establecer los mecanismos de control de los activos fijos que forman parte de la infraestructura de los servicios de TI, en este caso para adaptarlo al dominio de aplicación del sistema ISOPTEC se amplió el alcance del proceso para tomar en cuenta no solamente los activos propios de la infraestructura tecnológica sino también para registrar todo activo fijo de la institución.

Utilizar este proceso nos va a permitir:

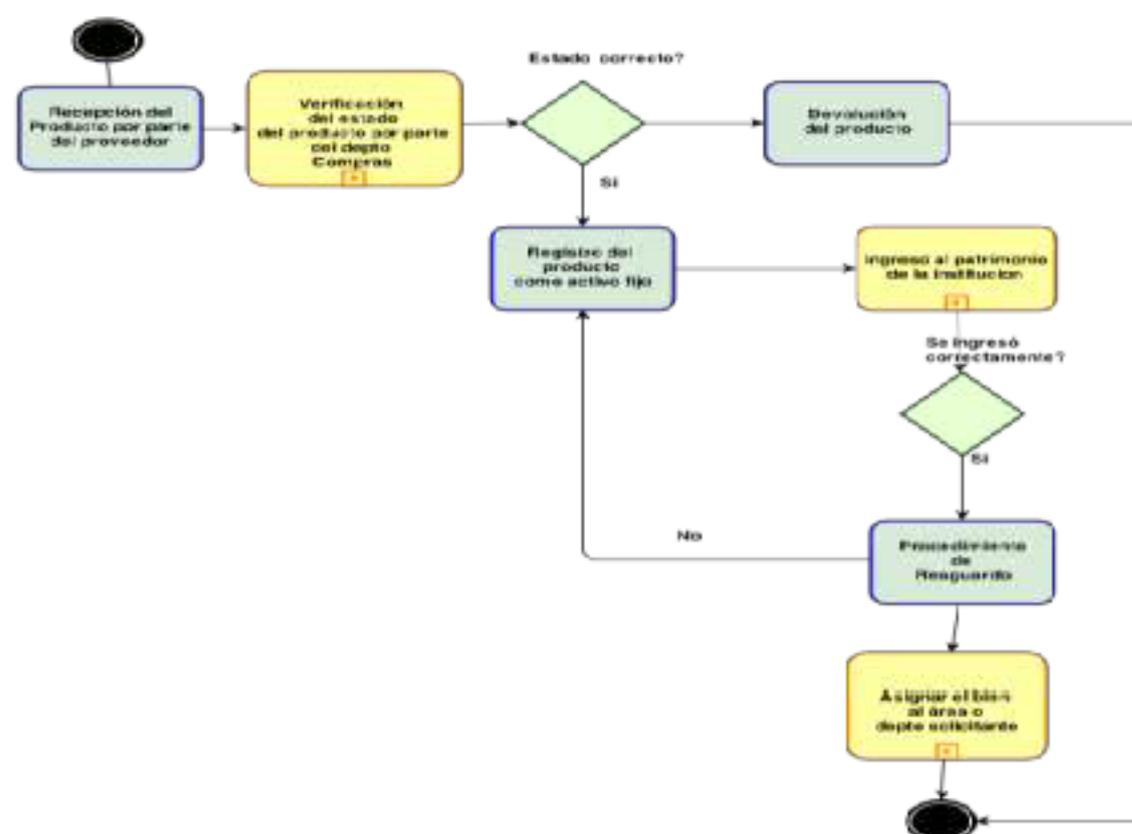
- Administrar la integridad de los activos fijos registrando aquellos activos fijos que sean pertinentes para el control administrativo.
- Administrar la información histórica sobre el estado actual así como la configuración, para saber en todo momento el estado actual de todos el inventario de la institución.
- Proveer la información necesaria que apoye a la toma de decisiones referentes al estado de la infraestructura, para saber por ejemplo si es suficiente ésta para garantizar la calidad de servicio que proporciona la Universidad.

17.3 Desarrollo

17.3.1 Análisis de requerimientos

Definida la metodología de desarrollo a utilizar, inició el proceso formal de maquila del software, siendo la primera actividad el levantamiento de requerimientos, para lo cual se analizaron los procesos manuales mediante el uso diagramas de proceso de la notación para el Modelado de Procesos de Negocio (BPMN), cuyo fin es proporcionar una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio. Entre estos interesados están los analistas de negocio (quienes definen y redefinen los procesos), los desarrolladores técnicos (responsables de implementar los procesos) y los gerentes y administradores del negocio (quienes monitorizan y gestionan los procesos). En síntesis BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación.[4]

Figura 17.2 Diagrama de proceso que describe el alta de un bien en el inventario



Al obtener la totalidad de los diagramas de proceso, se pudo tener una vista general del funcionamiento y con ello llevar a cabo la lista de requerimientos, los cuales son los mostrados en la tabla 17.1.

Tabla 17.1 Listado de requerimientos del sistema Isoptec

Nombre del Requerimiento	Tipo de Requerimiento	Descripción
Usuarios	No funcional (Catalogo)	Lista de usuarios permitidos a utilizar el sistema.
Activos	No funcional (Catalogo)	Lista de bienes materiales e inmateriales.
Personal	No funcional (Catalogo)	Lista dependiente de usuarios de personal con asignación de materiales.
Marcas	No funcional (Catalogo)	Lista de estados validos del bien, tomando como referencia el estado físico del bien.
Grupos de activos	No funcional (Catalogo)	División organizacional de los departamentos de la Universidad.
Subgrupos de activos	No funcional (Catalogo)	Subdivisión de los departamentos de la Universidad

Autorizar entradas	Funcional (Acción)	Validación de un nivel superior que realiza a la captura de entrada ingresado por un usuario de nivel bajo.
Movimientos	Funcional (Acción)	Asignación de un bien a un usuario del catálogo personal.
Trasposos	Funcional (Acción)	Acciones de re asignación de un bien entre diferentes usuarios del catálogo personal, ya sea por reasignación o por renuncia.
Reimpresión	Funcional (Acción)	Acción de re impresión de cualquiera de las acciones permitidas del sistema tales como asignación o trasposos.
Reportes: <ul style="list-style-type: none"> • Seguimiento de Activos • Entradas, Salidas • Existencia rápida • Ubicación actual por serie • Activos en área, personal 	Funcional (Acción)	Generación de información del estado de los resguardos, ubicaciones, reportes por área o personal de los bienes de la Universidad.
Permisos del Sistema	Funcional (Acción)	Asigna a los diferentes roles la capacidad de operación en el sistema.

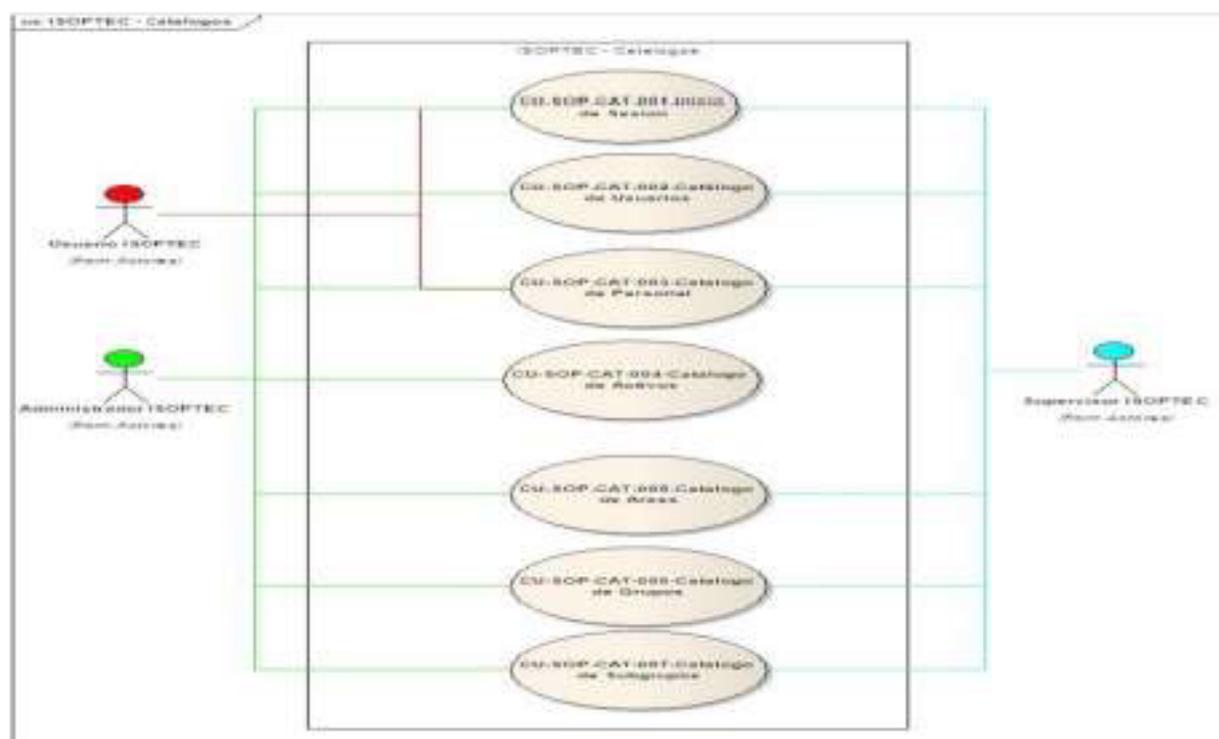
La lista de requerimientos, define la funcionalidad requerida durante el proceso de desarrollo, por lo que a partir de esta es posible la generación de los diagramas de UML que en conjunto con la metodología utilizada, es posible definir el modelo de casos de uso del sistema.

17.3.2 Diagrama de casos de uso y cálculo del esfuerzo.

Con la lista de requerimientos se diseñó el modelo de casos de uso, conjunto de diagramas y tablas de descripción que orientan el desarrollo de la funcionalidad de la aplicación; el objetivo en este nivel de desarrollo es orientar correctamente los springs del modelo funcional. En la figura 17 puede observarse uno de los diagramas generados para el modelo de casos de uso, en este caso de representa la interacción de los actores con los catálogos del sistema.

El sistema completo requirió además del aquí mostrado de los que describen las funcionalidades de autorizar devoluciones, avanzadas, consultar activos, entradas, devoluciones, reportes y salidas.

Figura 17.3 Diagrama perteneciente al modelo de casos de uso



Con el modelo de casos de uso finalizado y autorizado se procede al cálculo del esfuerzo, para ello se utiliza el propuesto por Gustav Karner en colaboración de Ivar Jacobson en el año de 1993 al modificar el método tradicional de puntos de función utilizando puntos de casos de uso, técnica posteriormente adoptada por IBM e implementada en la versión del proceso unificado de racional.[5] De primera instancia se hizo el cálculo de los puntos de casos de uso sin ajustar (UUCP) mediante la relación $UUCP = UAW + UUCW$ (Véanse tablas 17 y 17.1), cuyo valor fue de 58 puntos de casos de uso, sin ajustar.

Tabla 17.2 Cálculo del factor de peso de los actores sin ajustar (UAW)

Tipo de actor	Factor de peso	Número de actores	<i>Resultado</i>	
Simple	1	0	0	0
Promedio	2	1	2	2
Complejo	3	2	6	6
Total				8

Tabla 17.3 Cálculo del factor de peso de los casos de uso sin ajustar (UUCW)

Tipo de caso de uso		Factor de peso	Número de casos de uso	Resultado
Simple	5	3	15	
Promedio	10	2	20	
Complejo	15	1	15	
Total		50		

Posteriormente se realizó el ajuste de los casos de uso mediante la relación $UCP = UUCP \times TCF \times EF$, dando como resultado que los puntos de caso de uso ajustados (UCP) es de 53.86, véase tabla 17.1 y 17.2.

Tabla 17.4 Cálculo del factor de peso de complejidad técnica ($TCF = 0.6 + 0.01 * 19.5 = 0.795$)

Número de factor	Descripción	Peso	Valor	Factor
T1	Sistema Distribuido	2	0	0
T2	Tiempo de respuesta	1	1	1
T3	Eficiencia por el usuario	1	1	1
T4	Proceso interno complejo	1	3	3
T5	Reusabilidad	1	1	1
T6	Facilidad de instalación	0.5	4	2
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	1	2
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	2	2
T11	Objetivos especiales de seguridad	1	1	1
T12	Acceso directo a terceras partes	1	1	1
T13	Facilidades especiales de entrenamiento a usuarios finales	1	0	0
			Total Factor	19.5

Tabla 17.5 Cálculo del factor de peso de ambiente ($EF= 1.4-0.03*1.5 = 1.355$)

Número del factor	Descripción	Peso	Valor	Factor
E1	Familiaridad con el modelo del proyecto usado.	1.5	1	1.5
E2	Experiencia en la aplicación	0.5	1	0.5
E3	Experiencia OO.	1	0	0
E4	Capacidad del analista líder.	0.5	1	0.5
E5	Motivación.	1	0	0
E6	Estabilidad de los requerimientos.	2	1	2
E7	Personal media jornada.	-1	0	0
E8	Dificultad en lenguaje de programación.	-1	3	-3
			Total	1.5

Teniendo los puntos de casos de uso ajustados fue posible calcular el esfuerzo mediante la relación $E = UCP * CF$, otorgándole al factor de conversión (CF) un valor de 10 hrs, por lo que $E= 538.6$ horas/hombre, que corresponde al tiempo de programación estimado del proyecto, por lo que para calcular el esfuerzo total (Etot) se utiliza la tabla de la distribución genérica del esfuerzo, dando $Etot = 1346.5$ hrs/hombre, véase la tabla 17.5.

Tabla 17.6 Cálculo Esfuerzo Total (Etot)

Actividad	Porcentaje	Cálculo
Análisis	10.00%	134.65
Diseño	20.00%	269.3
Programación	40.00%	538.6
Pruebas	15.00%	201.975
Sobrecarga(otras actividades)	15.00%	201.975
Total		1346.5

El último paso consistió en calcular los días totales, definiendo de primera instancia el número de personas involucradas, cuatro en este caso, por lo que el tiempo estimado de desarrollo (TDesarrollo) se estima en 336.62 horas ($Etot/4$) y tomando en cuenta una dedicación de 4 horas al día TDesarrollo en días es ≈ 84.15 días hábiles, y, un costo de desarrollo de $\approx \$ 21,039.06$ M.N. tomando como referencia la hora de desarrollo tabulado en la UDS, que ya se encuentra incluido dentro de las actividades cotidianas de los profesores, por lo que el costo es representativo, ya que igualmente se cuenta con la infraestructura de implementación.

Con esta información, se ajusta al número de sprints, que por tratarse de 6, se estima el desarrollo real en 90 días hábiles (15 días hábiles por sprint).

17.4 Resultados

Al finalizar los seis sprints utilizando el lenguaje de programación C# con tecnología ASP .NET y bases de datos MySQL, se procedió a la realización de las baterías de pruebas de caja negra y caja blanca, con lo que se obtiene la liberación de sistema con la funcionalidad definida en el análisis de requerimientos, siendo que el sistema ha sido desarrollado mediante programación orientada a objetos modular pudo completarse en el tiempo estimado más un desfase de tiempo del 3%.

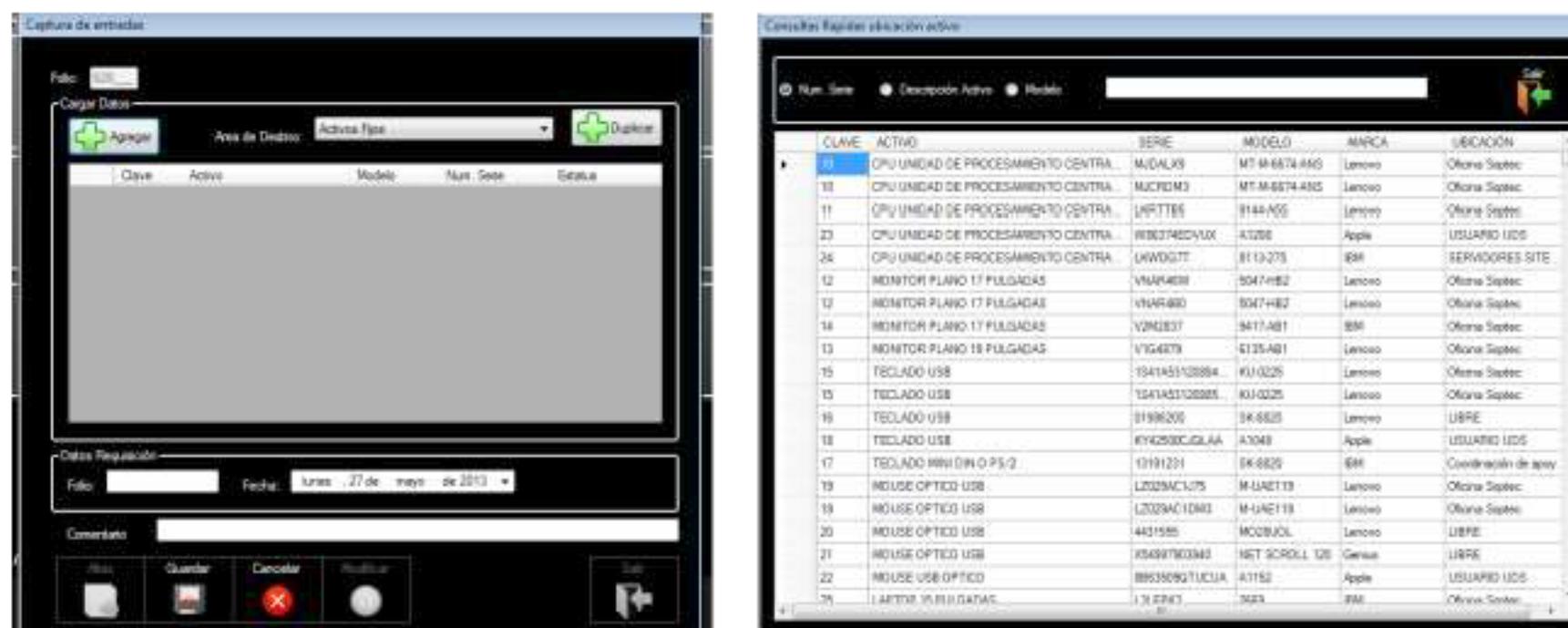
Figura 17.4 Pantalla principal del sistema ISOPTEC



En la figura 17.4 puede observarse la pantalla principal del sistema ISOPTEC, la cual divide el panel de operación en cuatro líneas de operación, siendo la primera destinada a los controles de operación de las funcionalidades principales que son la de captura de entradas, autorización de entradas, movimientos, asigna a SOPTEC y traspasos. La segunda línea de operación se encuentran los reportes más comunes, las cuales se pueden realizar por activo o por salida. En la tercera línea de operación se ubicaron los catálogos de la aplicación en el cual se administra la información de altas, bajas y cambios de los activos, personal o usuarios.

La última línea de operación tiene características avanzadas, en específico contempla la búsqueda por número de serie y los reportes especializados de existencia rápida, seguimiento y áreas controladas, véase figura 17.5.

Figura 17.5 Dos pantallas del sistema ISOPTEC, a la izquierda Captura de Entradas, a la derecha resultado de un reporte de búsqueda



17.5 Discusión

Este trabajo de investigación aplicada tuvo como propósito solucionar un añejo problema que se da al interior de la Universidad Tecnológica Metropolitana que es el control de los activos fijos y que pretende en convertirse en una propuesta de solución a la instituciones que tienen un problema similar y se encuentran en la disyuntiva de elegir la adquisición de un software comercial o la implementación de un sistema a la medida. La implementación de la aplicación ha traído un mayor control sobre el destino de los bienes de la institución, siendo mucho más sencillo al momento de que se realice una auditoría, ubicar los bienes y muy en especial el estado de las mismas, lo que garantiza siempre la certeza de contar con la información actualizada y redundante en una reducción de la erogación económica por faltantes. El hecho de que la aplicación haya seguido una metodología ITIL, da por sentado que los procesos se siguen mediante una norma de calidad y en especial validada, por lo que hizo más fácil la adaptación de los procesos manuales a los automatizados por el hecho de contar con suficiente bibliografía para sustentarlo y ejemplos de aplicación.

Otra aportación de esta investigación es el observar el resultado de la aplicación de una metodología de software emergente tal como son las consideradas ágiles, en este caso la Scrum, ya que por lo general muchas instituciones eligen utilizar métodos tradicionales como el de vida clásico, de cascada, y muy en especial mediante el uso del proceso unificado de desarrollo, en nuestro caso, se utilizó la metodología Scrum sin descuidar los modelos que se generan en esas metodologías, que a la larga, conllevan uno de los principales beneficios de documentar correctamente como es el hecho de poder continuar la aplicación incluso con un equipo nuevo de desarrollo reduciendo significativamente el tiempo la curva de aprendizaje de la funcionalidad de la aplicación y por ende, redundando en beneficios en cuanto al tiempo de desarrollo, que como indican los estudios realizados que influenciaron la estandarización de los procesos de software.

En cuanto a la metodología utilizada, resulta favorable debido al éxito de implementación, que debe de tomarse en cuenta pudo ser beneficiada al tiempo y números de proyectos que ha desarrollado el mismo equipo de programadores, sin embargo, los autores estimamos que se debe más al conocimiento de los artefactos y al conocimiento de la problemática, ya que es un tema muy afín al ser parte de la institución en la que se ha implementado y dicho sea de paso es un problema que surge mitigar a partir de la observación simple de la problemática, no necesitando en este caso la interacción de un grupo o terceras personas ajenas a la situación.

Mejoras que sería interesante implementar, son la adaptación de la metodología ágil a un ambiente en el cual se complementa con el seguimiento de un estándar PSP y CMMI, ya que es un tema actual en el que los investigadores de los estándares encuentran interesante estudiar, ya que por un lado se contempla la simplicidad de los sistemas desarrollado mediante metodología ágil como Scrum, y por otro la importancia de contar con la formalidad que brindan las metodologías tales como CMMI.

17.6 Conclusiones

Con la implementación del sistema propuesto es posible mejorar aspectos de control del área de activos fijos de la institución brindado una plataforma acorde a las necesidades del departamento y preparada para futuros cambios que pudiesen generarse con base al manejo de la normatividad implementada con el valor agregado de utilizar un esquema de seguridad más robusto y confiable.

El sistema ha demostrado su eficiencia al hacer uso de las interfaces y catalogos desarrollados así como reportes que se generan de forma periodica con el fin de proporcionar monitoreo y control de los activos; de igual forma es posible implementar una interfaz web la cual permita a manera de consumo de servicio a los usuarios poder consultar y solicitar actualizaciones o verificaciones de los activos que se les tienen asignado haciendo de este un proceso que involucre a todas las partes (usuario-activos-usuario).

Por otra parte con el fin de agilizar el proceso de inventarios y revisiones periodicas se tiene planeado continuar con el desarrollo para interfaces ligeras como pueden ser aplicaciones para Android que permiten realizar las revisiones desde un dispositivo movil que cuente con estas características y prestaciones, permitiendo una completa integración y actualizaciones de los datos a traves de los mismos.

Al terminar el desarrollo completando los seis sprints se entregó al cliente un programa funcional con todos los requerimientos del sistema cumplidos esto es debido a que la metodología SCRUM establece que al final de cada ciclo de desarrollo se tiene que entregar al cliente un incremento funcional, también la metodología requiere una participación más proactiva por parte del cliente dando como resultado un sistema que en todo momento esté acorde a las necesidades y expectativas del cliente.

Al utilizar la librería ITIL como directriz para el registro de activos fijos dio como resultado un sustento metodológico en la aplicación de las políticas de control de los mismos y resultó interesante observar que es fácil trasladar dichas políticas hacia activos de ámbitos generales.

17.7 Referencias

Presidencia de la República de México “Plan Nacional de Desarrollo 2007-2012 de los Estados Unidos Mexicanos” Recuperado el 22 de Septiembre de 2012 <http://pnd.gob.mx/> [2] P. Deemer ,G. Benefield, C. Larman, B. Voode. “Scrum Primer” 2010

APM Group Ltd “ITIL official Website” Recuperado el 25 de Mayo del 2013 en <http://www.itil-officialsite.com/>

Bruce Silver, “BPMN Method and Style: A levels-based methodology for BPM process modeling and improvement using BPMN 2.0”, Cody-Cassidy Press 2009

I. Jacobson, G. Booch,J. Rumbaugh. “The Unified Software Development Process” .AddisonWesley 2000

