

Tipo de artículo: Artículo original

Aplicación de la metodología scrum-programación extrema al sistema para el control de trazas de un servidor proxy

Application of the scrum-extreme programming methodology to the proxy server's trailer control system

Yoel Benedito Aguilera^{1*}  <https://orcid.org/0000-0003-1383-690X>

Lisuan Yanquiel Miguet Zuazo²  <https://orcid.org/0000-0002-6507-4516>

Yanquier Crespo González³  <https://orcid.org/0000-0001-9257-8106>

Maily Torres Vivanco⁴  <https://orcid.org/0000-0002-8185-1004>

¹ Departamento de Desarrollo de Software. Universidad de Ciego de Ávila Máximo Gómez Báez. ybaquilera@unica.cu

² Departamento de Informática. Facultad de Informática y Ciencias Exactas. Universidad de Ciego de Ávila Máximo Gómez Báez. lyminguet@unica.cu

³ Dirección de Informatización. Universidad de Ciego de Ávila Máximo Gómez Báez. yankier@unica.cu

⁴ Departamento de Informática. Facultad de Informática y Ciencias Exactas. Universidad de Ciego de Ávila Máximo Gómez Báez. mtorres@unica.cu

* Autor para correspondencia: ybaquilera@unica.cu

Resumen

El proceso de aseguramiento de la calidad es uno de los factores fundamentales en el ciclo de vida de un software, si se realiza correctamente garantiza la completa satisfacción del cliente y un alto índice de calidad del producto desarrollado. El presente trabajo tiene como objetivo presentar la aplicación de la metodología SCRUM-Extreme Programming (SXP) en el diseño de un Sistema para el Control de Trazas de un servidor proxy. Se describe las etapas y los principales artefactos generados durante el desarrollo final del software informático al utilizar esta metodología.

Palabras clave: control; metodología; proxy; SCRUM-Extreme Programming; trazas

Abstract

The quality assurance process is one of the fundamental factors in the life cycle of a software, if done correctly it guarantees complete customer satisfaction and a high quality index of the developed product. The present work aims to present the application of the SCRUM-Extreme Programming (SXP) methodology in the design of a Trace Control System of a proxy server. It describes the stages and the main artifacts generated during the final development of computer software when using this methodology.

Keywords: Control; methodology; proxy; SCRUM-Extreme Programming; traces

Recibido: 3/06/2021

Aceptado: 30/09/2021



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

Introducción

Las metodologías de desarrollo de software surgen como una alternativa y marco de trabajo a partir de la complejidad que conlleva realizar un software y como respuesta ante los problemas que se presentaban en cada etapa de desarrollo debido a que no se lleva un control adecuado lo que generaba problemas y desarrollo de software deficientes que no cumplían los requisitos del cliente.

Una de las cuestiones más importantes cuando se va a desarrollar o producir un producto de software es sin duda la elección de la metodología de desarrollo de software a seguir para llevar a cabo este proceso, este es un paso muy importante debido a que su uso final está orientado a la estructuración, planificación y control del proceso de desarrollo. Se clasifican en dos tipos principales, metodologías ágiles y metodologías robustas o pesadas. En las metodologías tradicionales el proceso es rígido y no cambia, los requerimientos son acordados de una vez y para todo el proyecto, demandando grandes plazos de planeación y poca comunicación con el cliente (Molina Montero, Vite Cevallos, et al., 2018). Las metodologías ágiles presentan como principal particularidad la flexibilidad, incluye una comunicación constante con el usuario. El cambio de requerimientos por parte del cliente es una característica especial, así como también las entregas, revisión y retroalimentación constante (Cadavid, Fernández Martínez, et al., 2013). Dentro de las metodologías ágiles se encuentran: XP (Extreme Programming), Crystal Methodologies, DSDM (Dynamic Systems Development Method), FDD (Feature-Driven Development) (Tinoco Gómez, Rosales López, et al., 2010).

La metodología ágil SXP es la unión de XP y SCRUM. XP (eXtreme Programming - Programación Extrema) es una metodología de desarrollo formulada por Kent Beck, es el más destacado de los procesos ágiles de desarrollo de software, su particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto (Letelier, 2006). Por su parte SCRUM, marco de trabajo también ágil para la gestión y desarrollo de software está basado en un proceso iterativo e incremental, fue desarrollada por Hirotaka Takeuchi e Ikujiro Nonaka, es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que sepamos por donde andamos (Kniberg, 2007). SXP por su parte, desarrollada en el 2007 en la Universidad de las Ciencias Informáticas está especialmente indicada para proyectos pequeños, con rápidos cambio de requisitos donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad (Rodríguez Villar, 2007).

El propósito de este documento es presentar la aplicación de la metodología SXP en el diseño de un Sistema para el Control de Trazas de un servidor proxy.

Materiales y métodos



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

Descripción de la metodología SXP.

SXP, es un híbrido cubano de metodologías ágiles, que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Basada completamente en los valores y principios de las metodologías ágiles expuestos en el Manifiesto Ágil. Como método de estimación se utiliza la opinión de expertos y constan con métricas o indicadores para lograr una eficiente calidad.

Consta de 4 fases principales: Planificación-Definición donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto; Desarrollo, es donde se realiza la implementación del sistema hasta que esté listo para ser entregado; Entrega, puesta en marcha; y por último Mantenimiento, donde se realiza el soporte para el cliente (Orozco Vaillant, 2013). De cada una de ellas se despliegan 7 flujos de trabajo: concepción inicial, captura de requisitos, diseño con metáforas, implantación, prueba, entrega de la documentación, soporte e investigación, el cual se utiliza por el equipo de desarrollo cuando sea necesario, es decir, es un flujo que se puede mover y utilizarlo en cualquier parte del ciclo de vida del proyecto.

De estos flujos se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, planificación de las iteraciones y las actividades que se van a realizar para lograr el producto, pruebas, además de las tareas necesarias para realizar las investigaciones para documentar todo el proceso.

Flujo actual del proceso de control de trazas de un servidor proxy

La Universidad de Ciego de Ávila Máximo Gómez Báez (UNICA) está constituida por un conjunto de departamentos y facultades, donde resalta la labor desempeñada por el Departamento de Redes e Informatización, en el cual se brindan un conjunto de servicios informáticos, tales como servicios especializados del nodo, generales de la red UNICA y del Departamento de Gestión de la Información Científico-Técnico (DGIC): Dentro de los especializados del nodo está el acceso a Internet, controlar este acceso es de suma importancia, debido a la amplia documentación dañina que existe en la red, para ello cada usuario tiene una cuenta con un privilegio de acceso, según su categoría docente, año que cursa o si pertenece a un grupo de científico. Cada vez que un usuario utiliza este servicio se genera una traza, que posee la fecha, sitio al que accedió, de donde realizó el acceso y el tamaño. Analizar esta traza resulta eficiente para la universidad para tomar decisiones en cuanto al uso de la navegación internacional, pero también es



importante para los usuarios poder controlar su navegación sin tener que solicitar al Especialista de Seguridad Informática el listado de sus accesos a la navegación nacional o internacional.

Para realizar el control de las trazas del proxy en la universidad se usa la herramienta Sawmill. Esta herramienta es fácil de usar, tiene una documentación completa, genera reportes y gráficos en vivo y utiliza diversos gestores de base de datos. Pero es una herramienta privativa, solo se puede utilizar en una computadora, se demora demasiado tiempo en el análisis de ficheros grandes y muchas veces solo analiza un por ciento del total de la información, presenta problemas con los filtrados de las trazas una vez analizadas.

Para el usuario poder llevar un control de su navegación, debe acercarse al Especialista de la Seguridad Informática y hacerle una solicitud de sus trazas. Luego el especialista le entregara un listado con todas las direcciones a las que accedió la semana pasada o las anteriores a estas. Si el usuario sospecha que existe algún uso descontrolado de su cuota o piensa que existe un robo de su contraseña debe informárselo al especialista, entonces el procederá a investigar el incidente.

A partir de ese momento surge una nueva propuesta que le facilitará el trabajo al especialista, permitiéndole obtener con un día de diferencia cada una de las trazas que se conciban y generar cada uno de los reportes de forma automática y mostrar las estadísticas de forma web. Además de que los usuarios podrán ver donde acceden, la hora y la fecha que se realizó dicha vista, por lo que podrán controlar su cuota y si ocurriera algún uso no autorizado de su cuota poder indicárselo al especialista y él tomará las medidas pertinentes.

Resultados y discusión

En el flujo de trabajo concepción inicial se realizó el primer encuentro con los clientes, donde se analizó el proceso y los subprocesos implicados en el negocio y las expectativas de desarrollo que tenía el cliente sobre el futuro producto, herramientas a utilizar, así como el alcance del proyecto.

Una vez conocidas las necesidades del cliente, se realiza la captura de requisitos donde se hace el levantamiento de toda la información relacionada con las características y cualidades con las que va a contar el producto para definir las Historias de Usuario (HU), así como también se definen los posibles riesgos a enfrentar durante el desarrollo del sistema.

Tabla 1. Requisitos funcionales más significativos

RF 1 Obtención e inserción de trazas	RF19 Eliminar dispositivo
RF2 Autenticar	RF20 Buscar dispositivo



RF3 Insertar usuario del sistema	RF21 Insertar rango de IP
RF4 Modificar usuario del sistema	RF22 Modificar rango de IP
RF5 Eliminar usuario del sistema	RF23 Eliminar rango de IP
RF6 Buscar usuario del sistema	RF24 Buscar rango de IP
RF7 Insertar código de respuesta	RF25 Generar reporte de sitios más visitados
RF8 Modificar código de respuesta	RF26 Generar reporte de sitios más visitados por Área
RF9 Insertar dominio prohibido	RF27 Generar reporte de acceso no autorizado
RF10 Modificar dominio prohibido	RF28 Generar reporte de los dispositivos donde más se navega
RF11 Eliminar dominio prohibido	RF29 Generar reporte por usuarios que más descargan
RF12 Buscar dominio prohibido	RF30 Modificar denuncia de uso no autorizado de la cuota
RF13 Insertar dominio ignorado	RF31 Visualizar traza de navegación por IP, día y usuario
RF14 Modificar dominio ignorado	RF32 Visualizar traza de navegación por día
RF15 Eliminar dominio ignorado	RF33 Insertar denuncia
RF16 Buscar dominio ignorado	RF34 Modificar denuncia
RF17 Insertar dispositivo	RF35 Eliminar denuncia
RF18 Modificar dispositivo	RF36 Buscar denuncia

Las historias de usuario son un instrumento para el levantamiento de requerimientos para el desarrollo de un software, que ha emergido con la aparición de los nuevos marcos de trabajo de desarrollo ágil. Son descripciones cortas de una necesidad de un cliente. La gran diferencia con los casos de uso, es que son más cortas y no deben describir la interfaz con el usuario, los pasos de navegación o flujo de procesos de la aplicación (PMOinformatica.com, 2017).

Tabla 2. Historia de usuario: Autenticar Usuario

Historia de usuario	
Número: 02	Nombre de la historia de usuario: Autenticar usuario
Modificación de historia de usuario número:	
Usuario: Yoel Benedico Aguilera	Iteración asignada: 2
Prioridad en el negocio: Alta	Puntos estimados: 7 días
Riesgo en desarrollo: Baja	Puntos reales: 0,10



Descripción: El usuario debe introducir su identificador y contraseña para acceder al sistema.
Observaciones: Se hace referencia al requisito funcional número: 2

Tabla 3. Historia de usuario: Insertar usuario del sistema

Historia de usuario	
Número: 3	Nombre de la historia de usuario: Insertar usuario del sistema
Modificación de historia de usuario número:	
Usuario: Yoel Benedico Aguilera	Iteración asignada: 2
Prioridad en el negocio: Alta	Puntos estimados: 3 días
Riesgo en desarrollo: Baja	Puntos reales: 0,10
Descripción: El responsable del sistema debe introducir los datos del usuario para registrarlo en la aplicación.	
Observaciones: Se hace referencia al requisito funcional número: 18	

Luego de definidas las HU se pasó a la implementación del sistema, donde se lleva a cabo la codificación de las HU con sus respectivas tareas de ingeniería. Una historia de usuario se descompone en varias tareas de ingeniería, las cuales describen las actividades que se realizarán en cada HU, así mismo las tareas de ingeniería se vinculan más al desarrollador, ya que permite tener un acercamiento con el código (Meléndez Valladares, Gaitan, et al., 2016). Durante este flujo, a medida que se desarrollan los requisitos funcionales del sistema informático, se realizaron pruebas para de esta forma, depurar la mayor cantidad de errores en la fase de desarrollo.

Tabla 4. Tarea de ingeniería: Autenticar Usuario

Tarea de Ingeniería	
Número de tarea: 2	Número de historia de usuario: 2
Nombre de la tarea: autenticar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 10
Fecha inicio: 1/12/2015	Fecha fin: 8/12/2015
Programador responsable: Yoel Benedico Aguilera	



Descripción: Para acceder a la aplicación el usuario tiene que introducir su identificador y contraseña de ser correctos aparecerá la interfaz de bienvenida al sistema.

Tabla 5. Tarea de ingeniería: Insertar usuario

Tarea de Ingeniería	
Número de tarea: 3	Número de historia de usuario: 3
Nombre de la tarea: Insertar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 9/12/2015	Fecha fin: 12/12/2015
Programador responsable: Yoel Benedico Aguilera	
Descripción: Para insertar los datos del usuario es necesario ir a la opción Gestionar y dar clic en insertar usuario, luego mostrará una vista con el listado de todos los usuarios y dará la opción de insertar un nuevo usuario.	

Una vez terminada la implementación de las funcionalidades del sistema, se pasó a la etapa de pruebas de aceptación, donde se corrigieron errores que no fueron detectados en el proceso de implementación y se comprobó si el producto final está en concordancia con los requisitos planteados por el cliente. La metodología tiene como una de sus prácticas el llamado test first, que es una disciplina de desarrollo de programas con pruebas, por lo que las pruebas son especificadas antes de comenzar las etapas de desarrollo de los productos. Para ello, se especifican los Casos de Prueba, son pruebas tipo caja negra a nivel del sistema: cada prueba de aceptación corresponde a un resultado producido por el sistema (López de la Nuez, 2020). En la tabla 6 y tabla 7 se describieron las pruebas de aceptación correspondientes a las HU.

Tabla 6. Caso de Prueba de Aceptación: Autenticar usuario

Caso de Prueba de Aceptación	
Código de Caso de Prueba: HU2_CP2	Nombre de Historia de Usuario: Autenticar usuario
Nombre de la persona que realiza la prueba: Yoel Benedico Aguilera	
Descripción de la Prueba: Probar la funcionalidad de Autenticar usuario	



Condiciones de ejecución: No utilizar caracteres extraños en el campo usuario
Entrada/ Pasos de ejecución: <ul style="list-style-type: none"> • Introducir usuario. • Introducir contraseña. Juego de datos válidos: <ul style="list-style-type: none"> • Usuario: admin • Contraseña: admin123
Resultado esperado: El usuario queda autenticado y puede usar las funcionalidades según el usuario que sea y el nivel de acceso que posea.
Evaluación de la prueba: Satisfactoria.

Tabla 7. Caso de Prueba de Aceptación: Insertar Usuario

Caso de Prueba de Aceptación	
Código de Caso de Prueba: HU3_CP3	Nombre de Historia de Usuario: Insertar usuario del sistema
Nombre de la persona que realiza la prueba: Yoel Benedico Aguilera	
Descripción de la Prueba: Probar la funcionalidad de Insertar usuario del sistema	
Condiciones de ejecución: No utilizar caracteres extraños en los campos usuario y nombre. Introducir una dirección de correo válida.	
Entrada/ Pasos de ejecución: <ul style="list-style-type: none"> • Introducir nombre y apellidos • Introducir correo electrónico • Introducir usuario. • Introducir contraseña. Juego de datos válidos: <ul style="list-style-type: none"> • Nombre y Apellidos: Yoel Benedico Aguilera 	



<ul style="list-style-type: none">• Correo electrónico: ybaguilera@unica.cu• Usuario: ybaguilera• Contraseña: admin123
Resultado esperado: Lograr guardar la información al sistema.
Evaluación de la prueba: Satisfactoria.

Seguendo el enfoque iterativo de SXP y con el objetivo de proporcionar mejores resultados y una mayor satisfacción al cliente, se realizan diferentes sprints, que no son más que fragmentos de códigos funcionales entregados al cliente, de acuerdo al principio dinámico de la metodología, la cual requiere que los equipos completen algún tipo de producto potencialmente liberable al final de cada iteración. Estas iteraciones están diseñadas para ser cortas y de duración fija.

Las iteraciones se pueden entender como mini proyectos: en todas las iteraciones se repite un proceso de trabajo similar para proporcionar un resultado completo sobre el producto final, de manera que el cliente pueda obtener los beneficios del proyecto de forma incremental. En cada iteración evoluciona el producto a partir de los resultados completados en las iteraciones anteriores, añadiendo nuevos objetivos/requisitos o mejorando los que ya fueron completados. Un aspecto fundamental para guiar el desarrollo iterativo e incremental es la priorización de los objetivos/requisitos en función del valor que aportan al cliente (proyectosagiles, 2017-2018).

Se definieron un total de 4 sprints y en cada uno se implementaron un conjunto de requisitos siguiendo un grado de importancia acordado en las reuniones con el cliente, estos encuentros se realizan cada 30 días.

En la primera iteración se implementó el RF1 es de vital importancia porque permitirá obtener las trazas del servidor proxy, posteriormente debe leer esos registros, dividir sus líneas en cada uno de los datos que tiene y guardarlos en la base de datos.

En la segunda iteración definida, los requisitos se implementaron de la siguiente forma: RF2, RF3, RF4, RF5, RF6, RF7, RF8, RF9, RF10, RF11, RF12, RF13, RF14, RF15, RF16, RF17, RF18, RF19, RF20, RF21, RF22, RF23, RF24.

En la tercera iteración definida, los requisitos se implementaron de la siguiente forma: RF25, RF26, RF27, RF28, RF29, RF30

En la cuarta iteración definida, los requisitos se implementaron de la siguiente forma: RF31, RF32, RF33, RF34, RF35, RF36

El cliente jugó un papel fundamental en el desarrollo de la aplicación, ya que tuvo una activa participación durante todo el periodo de desarrollo del sistema. El mismo emitió su criterio y puntos de vistas en cada uno de los momentos



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

requeridos, aportando así a la creatividad del diseño e implementación de la aplicación. Además, como parte del equipo de desarrollo aumentó el nivel de preocupación y responsabilidad por parte del desarrollador. Por su parte el desarrollador jugó el papel principal en la creación de la aplicación informática, debido a que es el actor principal en cuanto a planificación, diseño, implementación, prueba y mantenimiento del producto informático. Esta integración facilitó la calidad requerida del producto final.

Una vez finalizada la etapa de Desarrollo, se hace la Entrega del producto al cliente para su despliegue. La etapa de despliegue puede ser un fracaso total si no se interactúa con el cliente adecuadamente, no se entrega el producto como este exige, del mismo modo si no se utilizan prácticas y técnicas que dejen satisfechos al mismo. Por lo que es necesario dividir el despliegue en varias fases para guiar dicha etapa y conforman la planificación del tiempo destinado a cada fase del despliegue.

Tabla 8. Cronograma de actividades de despliegue

No	Actividad	Responsable	Ejecutor	Fecha	Observación
1	Creación de las condiciones tecnológicas para la instalación y puesta en marcha del sistema	Universidad de Ciego de Ávila Departamento de Informatización	Jefe de Seguridad Informática	23/4/2015	Definir local y PC donde se realizará el despliegue del sistema
2	Instalación y configuración del sistema	Universidad de Ciego de Ávila Departamento de Informatización	Yoel Benedico Aguilera	27/4/2015 hasta 3/5/2015	Llevar manual de instalación del sistema
3	Formación y capacitación para los usuarios	Universidad de Ciego de Ávila Departamento de Informatización	Jefe de Seguridad Informática	4/5/2015 hasta 24/5/2015	Llevar manual de usuario Llevar ayuda del sistema
4	Creación del Informe de las No Conformidades Detectadas	Universidad de Ciego de Ávila Departamento de Informatización	Yoel Benedico Aguilera	4/5/2015 hasta 24/5/2015	Utilizar Informe de las No Conformidades Detectadas



5	Resolución de las No Conformidades Detectadas	Facultad de Ciencias Informáticas	Yoel Benedico Aguilera	25/5/2015 hasta 17/6/2015	Actualización del sistema con las No Conformidades
6	Entrega del Acta de Validación de la Fase de Despliegue	Universidad de Ciego de Ávila Departamento de Informatización	Jefe de Seguridad Informática Yoel Benedico Aguilera	18/6/2015	Concluye etapa de despliegue del software

Para finalizar esta la etapa de Mantenimiento. En el transcurso de la etapa se le brindará soporte al sistema informático anteriormente desplegado. Esto se llevará a cabo mediante la puesta en marcha de diferentes actividades ligadas a este servicio, que contribuyen a organizar el proceso, garantizando el correcto funcionamiento del sistema. El objetivo principal de esta fase es asegurar que el producto desarrollado trabaje con la calidad requerida, obteniéndose una completa satisfacción por parte de los clientes.

Conclusiones

El presente trabajo se centró en la utilización de la metodología SXP para la implementación de un sistema capaz de controlar las trazas generadas por el servidor proxy Squid. Se demostró que la metodología SXP contiene la organización de los procedimientos a seguir paso a paso, con la generación de cada uno de los artefactos, antes mencionados, son necesarios para lograr una documentación con el éxito y la eficiencia necesaria que requiere un proceso de software.

Conflictos de intereses

Ninguno de los autores manifestó la existencia de posibles conflictos de intereses que debieran ser declarados en relación con este artículo.

Contribución de los autores

Yoel Benedico Aguilera: contribución a la idea y formulación de la investigación, su análisis e interpretación. Aplicación de técnicas formales para sintetizar los datos del estudio. Desarrollo de la investigación y del software. Redacción de la versión final del artículo.

Yanquier Crespo González: Responsable de gestión y coordinación de la planificación y ejecución del trabajo. Provisión de recursos informáticos para el análisis de la investigación.



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

Lisuan Yankiel Minguet Zuazo: Responsable de la actividad de administración para la interpretación de datos del código de software.

Mailyen Torres Vivanco: Contribución al diseño del artículo y responsable de la validación y verificación del software.

Financiamiento

El trabajo no requirió financiación. Formó parte de una de las líneas de investigación que se desarrollan en la Facultad de Informática y Ciencias Exactas de la Universidad de Ciego de Ávila Máximo Gómez Báez.

Referencias Bibliográficas

- CADAVID, A. N.; FERNÁNDEZ MARTÍNEZ, J. D., et al. Revisión de metodologías ágiles para el desarrollo de software. *Prospect*, 2013, 11(2): 30-39.
- MOLINA MONTERO, B.; Vite Cevallos, H., et al. Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales-Revista multidisciplinaria de investigación*, 2018, 114-121.
- Kniberg, H. (2007). *Scrum and xp from the trenches*. Lulu. com.
- Letelier, P. (2006). *Metodologías Ágiles para el desarrollo de software: extreme programming (xp)*.
- MELÉNDEZ BALLADARES, S.M.; GAITAN M.E., et al. Metodología ágil de desarrollo de software Programación Extrema. Tesis de grado en Ciencias de la Computación. Departamento de Computación. Universidad Nacional Autónoma de Nicaragua, 2016.
- Orozco Vaillant, M. E. (2013). *Informe de investigación de la Metodología SXP*. Bayamo: Universidad de Gramma.
- PMOinformatica.com. (2017). *La oficina de proyectos de informática*. (2017), [Consultado el: 7 de junio de 2018.] Disponible en: <http://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>
- proyectosagiles. (2017-2018). *proyectos agiles.org*. [En línea]. (2017), [Consultado el: 7 de julio de 2018.] Disponible en: <https://proyectosagiles.org/desarrollo-iterativo-incremental/>
- Rodríguez Villar, M. (2007). *Introducción de procedimientos ágiles en la producción de software en la Facultad 7 de la Universidad de Ciencias Informáticas*. Universidad de Ciencias Informáticas, 2007: PhD thesis.
- TINOCO GÓMEZ, O.; Rosales López, P.P., et al. Criterios de selección de metodologías de desarrollo de software. *Industrial data*, 2010, 13(2): 70-74.
- LÓPEZ DE LA NUEZ, H. Sistema Informático de Gestión de Solicitudes de Servicio para la Suit de Gestión Técnica Informática de la Empresa “EMPIFAR Habana”. Tesis de grado en Ingeniería Informática. Departamento de Informática. Universidad de Ciego de Ávila Máximo Gómez Báez, 2020.

