

APLICACIÓN DE LA METODOLOGÍA SCRUM-PROGRAMACIÓN EXTREMA AL SISTEMA DE CONTROL DE CONSUMO DE COMBUSTIBLES

APLICACIÓN DE LA METODOLOGÍA SCRUM-PROGRAMACIÓN EXTREMA AL SISTEMA DE CONTROL DE CONSUMO DE COMBUSTIBLES

AUTORES: Mailyn Torres Vivanco¹

Yanai Fernández Rigondeaux²

Yankiel Crespo González³

DIRECCIÓN PARA CORRESPONDENCIA: mtorres@unica.cu

Fecha de recepción: 17-06-2021

Fecha de aceptación: 29-10-2021

RESUMEN

Las metodologías para el desarrollo del software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto. El desarrollo de un software con calidad se vio afectado por la planificación irreal de los directivos para estimar tiempo, esfuerzo y costos del sistema. Estos problemas conllevan a errores costosos por fallas en el software. El presente trabajo tiene como objetivo aplicar la metodología híbrida SCRUM-Extreme Programming en el diseño del Sistema para el Control del Consumo de Combustibles. Se describieron las fases y flujos de trabajo de la metodología, así como las iteraciones para la entrega de los requisitos funcionales y los principales artefactos generados como son las Historias de Usuario, Tareas de Ingeniería y Pruebas de Aceptación durante el ciclo de vida del software.

PALABRAS CLAVE: control; consumo; combustible; metodología; Scrum-Extreme Programming

APPLICATION OF THE SCRUM-EXTREME PROGRAMMING METHODOLOGY TO THE FUEL CONSUMPTION CONTROL SYSTEM

ABSTRACT

¹ Mailyn Torres Vivanco, Ingeniera en Ciencias Informáticas, profesora asistente, Universidad de Ciego de Ávila Máximo Gómez Báez, Facultad de Informática y Ciencias Exactas Cuba. E-mail: mtorres@unica.cu. ORCID: <http://orcid.org/0000-0002-8185-1004>

² Yanai Fernández Rigondeaux, Máster en Nuevas Tecnologías para la Educación Superior, profesora auxiliar, Universidad de Ciego de Ávila Máximo Gómez Báez, Facultad de Informática y Ciencias Exactas Cuba. E-mail: yanai@unica.cu. ORCID: <http://orcid.org/0000-0003-4836-2556>

³ Yankiel Crespo González, Máster en Nuevas Tecnologías para la Educación Superior, profesor asistente, Universidad de Ciego de Ávila Máximo Gómez Báez, Facultad de Informática y Ciencias Exactas Cuba. E-mail: yankiel@unica.cu. ORCID: <http://orcid.org/0000-0001-9257-8106>

The methodologies for software development arise when there is a need to use a series of procedures, techniques, tools and documentary support when developing a product. The development of quality software was affected by the unreal planning of managers to estimate time, effort and system costs. The present work aims to present the application of the SXP hybrid methodology in the design of a Fuel Consumption Control System. The phases and workflows of the methodology were described, as well as the iterations for the delivery of the functional requirements and the main artifacts generated user stories, engineering tasks and acceptance tests during the software life cycle.

KEYWORDS: control; consumption; fuel, methodology, SCRUM- Extreme Programming

INTRODUCCIÓN

Las metodologías para el desarrollo del software surgen con el propósito de organizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto. Con la aparición de las metodologías se les da solución a la crisis de la industria del software generado por la planificación irreal de los directivos para estimar tiempo, esfuerzo y costos del software. Estos problemas ocasionan errores costosos por fallas en el software, productos con muchos defectos y demora en la aceptación de los usuarios.

Pérez (2012) expresa: "Las metodologías de desarrollo de software son decisivas en el éxito o fracaso de un software" (p.11). Imponen un proceso disciplinado con el fin de aumentar la calidad que se produce en cada fase de desarrollo y lograr la satisfacción del cliente. Se clasifican en dos tipos de grupos: metodologías tradicionales y ágiles. "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 una vez ha terminado ésta" (Molina, Vite y Dávila, 2018, p.115). 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 y Morales, 2013, p.31).

Según Tinoco, Rosales y Salas (2010) expresan: "entre las principales metodologías ágiles se encuentran: *Crystal Methodologies*, *DSDM (Dynamic Systems Development Method)*, *Lean Programming*, *FDD (Feature-Driven Development)* y *Extreme Programming*" (p.72).

La metodología ágil SXP es la fusión de Extreme Programming y SCRUM. XP (Extreme-Programming) fue introducida por Kent Beck finales de los 1990.

La metodología XP está formada por un conjunto de valores y principios, con el objetivo fundamental de un rápido desarrollo y un software de alta calidad (Mármol, 2019). Por su parte la metodología SCRUM fue desarrollada por Ken Schwaber y Jeff Sutherland en los años 1990.

Es una metodología para el desarrollo de software iterativa e incremental, se dice que es iterativa ya que se ejecuta en bloques temporales cortos y fijos (de no menos de dos semanas) que reciben el nombre de *sprints* y es incremental en tanto se obtienen funcionalidades del producto final al terminar cada iteración (Salazar, Casallas, Linares, Lozano y Valbuena, 2018). Mientras que la metodología SXP fue 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, 2007).

DESARROLLO

Descripción de la metodología SXP

SXP es la unión de las metodologías Scrum y XP siendo una metodología ágil que permite minimizar el riesgo del proyecto. La metodología se creó y utilizó por primera vez en Cuba. Una de sus ventajas es la de tener como parte del equipo al usuario final, por tanto, el cliente se encuentra vinculado directamente con el proyecto. Además, está especialmente indicada para proyectos pequeños, con rápidos cambios de requisitos y se orienta a una entrega rápida de resultados y una alta flexibilidad. Cada cierto período de tiempo realiza la entrega de pequeñas versiones del producto, aunque no cuenten con todas las funcionalidades requeridas por el cliente final. Adopta lo mejor de la combinación de ambas metodologías, del lado de XP se destacan las prácticas, valores y el ciclo de vida del software.

En lo que respecta a SCRUM es un marco de trabajo que comprende varios procesos y técnicas orientadas a la gestión del desarrollo de productos complejos. Busca generar resultados de calidad en iteraciones cortas (generalmente de 2 a 4 semanas), en las cuales el equipo de desarrollo hace uso de los eventos, artefactos y reglas asociadas (Carrasco, Ocampo, Ulloa y Azcona, 2019).

La metodología de desarrollo de software SXP consta de 4 fases y 7 flujos de trabajo:

- Planificación-Definición: Es 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 y las pruebas.
- Entrega: Entrega de la documentación y la puesta en marcha del sistema.
- Mantenimiento: Es donde se realiza el soporte al cliente.

De cada una de estas fases 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 y de las Tareas de Ingeniería, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que permite mejorar el diseño cada vez que se le añade una nueva funcionalidad.

Se utilizó la técnica opinión de expertos para encuestar a los trabajadores del Departamento de transporte de la Universidad. La población está integrada por cuatro trabajadores del departamento de transporte de la Universidad de Ciego de Ávila “Máximo Gómez Báez” encargados del proceso de control de consumo de combustibles. No se selecciona muestra puesto que se va a trabajar con la misma población. Se utiliza el Análisis Porcentual para la tabulación de la encuesta aplicada a los trabajadores del área de transporte de la Universidad de Ciego de Ávila Máximo Gómez Báez de la provincia de Ciego de Ávila.

Flujo actual del proceso de control del consumo de combustibles

El área de transporte de la Universidad de Ciego de Ávila es la encargada de mantener el control estricto del consumo de combustibles de los vehículos que dispone ya sean carros ligeros, vehículos con licencia operativa o motos. Para ello cada inicio de mes, “el técnico en uso racional de la energía” se remite a la caja de seguridad de su oficina, realiza la extracción de las tarjetas magnéticas y posteriormente las traslada a la agencia de Fincimex para recargar las mismas, logrando asignar a cada chofer determinada cantidad de combustible y en dependencia de las características del vehículo, se recarga con el tipo de combustible diésel o gasolina.

La directora de servicios generales le envía al técnico encargado de la entrega de las tarjetas un reporte con la distribución de combustible por vehículo. Sobre la base de este reporte el técnico encargado de la entrega de las tarjetas llena un modelo de entrega de combustible, autorizando a los choferes o responsables de vehículos a realizar la extracción de la tarjeta magnética. El chofer se remite al departamento de economía, se identifica en la relación de la lista de choferes de dicha oficina y se le deposita la tarjeta.

Los choferes se dirigen con sus respectivos automóviles al servicentro a cargar el combustible y reciben un comprobante. El chofer le envía ese comprobante al técnico encargado de la entrega de las tarjetas y este último llena el informe de consumo de combustible. Una vez llenado todos los datos, debe realizar todos los cálculos pertinentes para llevar un control de la entrada y salida del combustible en la universidad.

Todos estos cálculos se realizan manualmente. El surgimiento de una nueva propuesta le permite al especialista de transporte centralizar toda la información que se relaciona con el informe de consumo de combustible ya sean vehículos con licencia operativa, carros ligeros o motos, mostrando un listado de los choferes que realizaron la extracción de combustible y un listado de las tarjetas magnéticas que han sido cargadas. No obstante, el modelo de salida de combustible contiene en sus datos una fecha para asegurar la entrega de combustible al chofer. Además, brinda información referente al uso de combustible por las facultades que pertenecen a la universidad y las unidades de servicio que están vinculadas al centro.

También se adquiere información referente a la cantidad de viajes realizados, la cantidad de pasajeros, los kilómetros recorridos, el índice de consumo real y normado, el saldo de combustible a consumir el próximo mes, los datos de los choferes y de los vehículos. Una de las informaciones importantes que genera el nuevo sistema es el consumo de combustibles de todas las facultades que están integradas a la universidad, así como de las unidades de servicio que facilitan el transporte a la institución. Permite graficar el índice de consumo anual por carro con el objetivo de analizar el comportamiento del consumo de combustible por vehículo.

En el flujo de trabajo, concepción inicial se realizó el primer encuentro con el jefe de transporte y el técnico encargado de la entrega de las tarjetas donde se analizó el proceso y los subprocesos implicados en el negocio. Se definió en conjunto con el cliente las 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 funcionalidades con las que va a contar el producto para definir las Historias de Usuario (HU).

Luego de definidas las HU se pasó a la implementación del sistema, donde se llevó a cabo la codificación de las HU con sus respectivas tareas de ingeniería. Durante este flujo, a medida que se desarrolló los requisitos funcionales del sistema informático, se realizaron pruebas para de esta forma, depurar la mayor cantidad de errores en la fase de desarrollo.

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.

Siguiendo 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 liberables al final de cada iteración. Estas iteraciones están diseñadas para ser cortas y de duración fija.

Se definieron un total de cinco fragmentos de código 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. Para la primera iteración se definieron 49 requisitos para un total de 79 requerimientos de gestión, quedaron estructurados de la siguiente forma: RF1 hasta RF49.

Para la segunda iteración se completaron los 30 requisitos restantes de gestión, de la siguiente manera: RF-50 hasta RF-79. Para la tercera iteración se definieron los 14 requisitos más importantes para generar reportes para un total de 51 requerimientos, los cuales quedaron estructurados de la siguiente forma: RF-80 hasta RF-117. A continuación, se muestra una selección de los requisitos funcionales más significativos por iteraciones, ver Tabla 1:

RF-1 Autenticar usuario	RF-9 Insertar el tipo de combustible
RF-2 Insertar usuario	RF-10 Insertar choferes
RF-3 Modificar usuario	RF-11 Insertar carros
RF-4 Eliminar usuario	RF-12 Insertar la distribución de combustible
RF-5 Insertar tarjetas	RF-13 Insertar la distribución de combustible
RF-6 Modificar tarjetas	RF-14 Generar reporte de hoja de ruta
RF-7 Eliminar tarjetas	RF-15 Exportar el informe de consumo de combustible
RF-8 Mostrar tarjetas	RF-16 Generar informe sobre el índice de consumo de cada vehículo.

Tabla 1. Requisitos funcionales más significativos (elaboración propia)

El cliente jugó un papel importante en el desarrollo del Sistema de Control del Consumo de Combustibles, se desempeñó como consultor, brindando asesoría en diversas ocasiones y está familiarizado con los procesos que serán informatizados con todo el negocio de manera general.

Además, su activa participación en las reuniones con el desarrollador y su aporte de ideas, facilitó la planificación y ejecución de los sprints, así como la creación de un producto acorde a los objetivos de la investigación. Se implementaron los requisitos funcionales en el tiempo previsto para cada una de las iteraciones, donde el cliente aportó nuevas ideas y puntos de vista, desempeñando un papel activo. Al mismo tiempo se le realizaron pruebas al software para detectar y corregir los posibles errores.

La etapa de despliegue o implantación es el puente entre el desarrollo y la utilización habitual del sistema, que se producirá una vez que sea incorporado en la organización. Al final de esta fase, el equipo de implantación debe haber entregado todos los elementos de control del sistema. En esta etapa se divide el despliegue en varias fases para planificar el tiempo destinado a cada actividad del despliegue.

Para finalizar el ciclo de vida del software se realiza el mantenimiento al sistema. 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.

Una vez terminado el desarrollo del sistema informático, se utilizó la técnica opinión de expertos para encuestar a los trabajadores del área de transporte. En dicho departamento laboran un total de cuatro personas, se les aplicó la encuesta con el objetivo de conocer el nivel de aceptación de dicho sistema.

En el flujo de trabajo concepción inicial se definieron las Historias de Usuario para especificar los requisitos y priorizar el desarrollo. Las Historias de Usuario son definidas por el líder con la participación de los equipos de desarrollo, las cuales pueden ser modificadas en cualquier momento del ciclo de vida del proyecto, si el equipo de trabajo lo considera necesario (Ramírez, Branch y Jiménez, 2019). Su presencia asegura que los desarrollos cubren toda la funcionalidad descrita. Se implementaron las Historias de Usuarios de las funcionalidades más significativas las cuales se muestran a continuación en la Tabla 2 y Tabla 3:

Historia de Usuario	
Número: 01	Nombre Historia de Usuario: Insertar el tipo de combustible
Modificación (o extensión) de Historia de Usuario (Nro. Y Nombre):	
Usuario: Mailyn Torres Vivanco	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3 días
Riesgo en Desarrollo: Baja	Puntos Reales: 0,2
Descripción: El sistema debe insertar el tipo de combustible ya sea	

gasolina o diésel.
Observaciones: Se hace referencia al RF-27

Tabla 2. Historia de Usuario: Insertar el tipo de combustible (elaboración propia)

Historia de Usuario	
Número: 02	Nombre Historia de Usuario: Insertar tarjetas
Modificación (o extensión) de Historia de Usuario (Nro. Y Nombre):	
Usuario: Mailyn Torres Vivanco	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3 días
Riesgo en Desarrollo: Baja	Puntos Reales: 0,2
Descripción: El sistema debe insertar el número de tarjeta.	
Observaciones: Se hace referencia al RF-9	

Tabla 3. Historia de Usuario: Insertar tarjetas (elaboración propia)

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, Gaitan y Pérez, 2016). Se desarrollaron las tareas de ingeniería con sus componentes que pertenecen a las historias descritas anteriormente, ver Tabla 4 y Tabla 5.

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 01
Nombre Tarea: Insertar el tipo de combustible	

Tipo de Tarea: Desarrollo	Puntos Estimados: 2 días
Fecha Inicio: 15/11/2013	Fecha Fin: 17/11/2013
Programador Responsable: Mailyn Torres Vivanco	
<p>Descripción: Para insertar el tipo de combustible se debe crear un módulo que represente una imagen que haga referencia al tipo de combustible, una vez que se dé clic sobre la imagen se debe mostrar una ventana que tenga las opciones de insertar, filtrar, mostrar, eliminar y permita listar los elementos insertados.</p>	

Tabla 4. Tarea de ingeniería: Insertar tipo de combustible (elaboración propia)

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 02
Nombre Tarea: Insertar tarjetas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2 días
Fecha Inicio: 18/11/2013	Fecha Fin: 20/11/2013
Programador Responsable: Mailyn Torres Vivanco	
<p>Descripción: Para insertar las tarjetas se debe crear un módulo que represente una imagen que haga referencia a la tarjeta, una vez que se dé clic sobre la imagen de debe mostrar una ventana que tenga las opciones de insertar, filtrar, mostrar, eliminar y permita listar los elementos insertados.</p>	

Tabla 5. Tarea de ingeniería: Insertar tarjetas (elaboración propia)

Una vez terminada la implementación de las funcionalidades del sistema, se pasó a la etapa de pruebas de aceptación. 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, 2020). Se describieron las pruebas de aceptación correspondientes a las HU, ver la Tabla 6 y Tabla 7.

Caso de Prueba de Aceptación	
Código: HU1-CP1	Historia de Usuario: Insertar el tipo de combustible
Nombre: Mailyn Torres Vivanco	
Descripción: Probar la funcionalidad de insertar el tipo de combustible	
Condiciones de Ejecución: No utilizar números para el tipo de combustible. No utilizar letras, ni caracteres extraños para el precio del combustible.	
Entrada / Pasos de ejecución: Introducir el tipo de combustible (diésel o gasolina). Introducir el precio del combustible. Juego de datos válido: Tipo de combustible: gasolina. Precio del combustible: 5.98.	
Resultado Esperado: Lograr guardar la información en el sistema.	
Evaluación de la Prueba: Satisfactoria	

Tabla 6. Caso de Prueba de Aceptación: Insertar el tipo de combustible (elaboración propia)

Caso de Prueba de Aceptación	
Código: HU2-CP2	Historia de Usuario: Insertar tarjetas
Nombre: Mailyn Torres Vivanco	
Descripción: Probar la funcionalidad insertar tarjetas.	
Condiciones de Ejecución: No utilizar letras, ni caracteres extraños para la tarjeta. Solo puede tener siete dígitos.	

<p>Entrada / Pasos de ejecución: Introducir el número de tarjeta. Juego de datos válido: Número de tarjeta: 1234567</p>
<p>Resultado Esperado: Lograr guardar la información en el sistema.</p>
<p>Evaluación de la Prueba: Satisfactoria</p>

Tabla 7. Caso de Prueba de Aceptación: Insertar tarjetas (elaboración propia)

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 conformar la planificación del tiempo destinado a cada fase del despliegue. La siguiente figura representa la relación entre los nodos del modelo de despliegue, donde el nodo cliente representa cualquier máquina que interactúa con la aplicación, a su vez puede estar comunicada con una impresora mediante el puerto LPT o USB para la impresión de los reportes. Esta PC además cuenta con un navegador para realizar las peticiones al Servidor Web mediante el protocolo Http el cual interactuando con el Servidor de Base de Datos responderá a las peticiones realizadas, ver Figura 1.

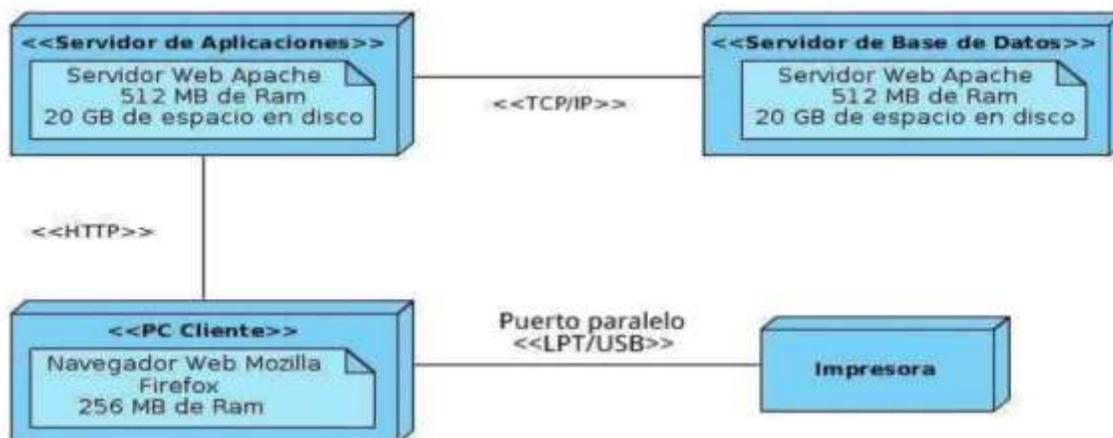


Figura 1. Diagrama de despliegue (elaboración propia)

Para medir el nivel de satisfacción de los clientes con respecto al sistema informático, se realizó una encuesta a los usuarios del departamento de transporte de la Universidad de Ciego de Ávila que

utilizan el Sistema Informático para el Control del Consumo de Combustibles.

Se aplicó el método estadístico Análisis Porcentual para interpretar los resultados, arrojando los siguientes criterios:

- 1- Existen dos personas que consideran que el sistema informático para el Control del Consumo de Combustibles soluciona sus necesidades, lo cual representa un 100 % de las personas encuestadas.
- 2- Existe una persona que expresa que el sistema siempre está siendo aplicado para el Control del Consumo de Combustibles, lo que representa un 50 % del total de personas encuestadas. Sin embargo, una persona expresa que solo a veces está siendo aplicado, lo que representa un 50 % del total de personas encuestadas.
- 3- Dos personas encuestadas afirman que se sienten cómodos al interactuar con el sistema, lo cual representa un 100 % de personas encuestadas.
- 4- Dos personas encuestadas afirman que los cálculos realizados arrojan los resultados esperados, lo cual representa un 100 % de personas encuestadas.
- 5- Dos de las personas encuestadas afirman que el proceso se tornaba más lento antes de ser informatizado.
- 6- Dos de las personas encuestadas expresan que el nivel de satisfacción con el sistema informático es aceptable lo cual representa un 100%.

Los resultados de la entrevista fueron los esperados, corroborando que el software cumple con las expectativas propuestas. La utilización de la técnica empleada, junto a las pruebas particulares realizadas como parte del proceso de desarrollo del sistema, corrobora la efectividad de la misma y las posibilidades que brinda a los clientes involucrados.

CONCLUSIONES

El trabajo con la metodología de desarrollo de software SXP ayudó a cumplir con los plazos establecidos en la etapa de planeación del sistema. La comunicación con el cliente fue fluida lo cual permitió requisitos adaptables y cambiantes. En cada una de las iteraciones se le entregó al usuario final pequeñas versiones funcionales del software, esto permitió una visión y valoración por parte del cliente expresando su criterio y puntos de vistas. La aplicación de la metodología generó un grupo de procedimientos a seguir con la creación de artefactos tales como las historias de usuario y tareas de ingeniería. Además, permitió trabajar de forma ordenada y rápida logrando un estrecho vínculo con el cliente y brindando un sistema de acuerdo a las necesidades requeridas.

REFERENCIAS

- Cadavid, A.N., Fernández, J. D. y Morales, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospect*, 11(2), 30-39.
- Carrasco, M. K., Ocampo, W. J., Ulloa, L.J. y Azcona, J. (2019). Metodología Híbrida de Desarrollo de Software combinando XP y SCRUM. *Mikarimin. Revista Científica Multidisciplinaria*, 5(2), 109-116.
- López, H. (2020). 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, Cuba.
- Mármol, D. (2019). Introducción a XP. [En línea]. La Habana: Editorial Ediciones Futuro Recuperado de: <https://publicaciones.uci.cu/index.php/serie/article/view/323>.
- Meléndez, S.M., Gaitan M.E. y Pérez, N.N. (2016). Metodología ágil de desarrollo de software Programación Extrema. Tesis de grado en Ciencias de la Computación. Departamento de Computación. Universidad Nacional Autónoma de Nicaragua.
- Molina, B., Vite, H. y Dávila, J. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales-Revista multidisciplinaria de investigación*, 2(17), 114-121.
- Pérez, M. J. (2012). Guía Comparativa de Metodologías Ágiles. Tesis de grado en Ingeniería Informática de Servicios y Aplicaciones. Departamento de Servicios y Aplicaciones. Universidad de Valladolid.
- Ramírez, D. L., Branch, J. W., y Jiménez, J. A. (2019). Metodología de desarrollo de software para plataformas educativas robóticas usando ROS-XP. *Revista Politécnica*, 15(30), 55-69
- Rodríguez, M. (2007). Introducción de procedimientos ágiles en la producción de software en la Facultad 7 de la Universidad de Ciencias Informáticas. Tesis de grado en Ingeniería Informática. Facultad 7 de la Universidad de Ciencias Informáticas. Universidad de Ciencias Informáticas, Cuba.
- Salazar, J. C., Casallas, Á. T., Linares, J. C., Lozano, A., y Valbuena, Y. L. (2018). Scrum versus XP: similitudes y diferencias. *Tecnología Investigación y Academia*, 6(2), 29-37.
- Tinoco, O., Rosales, P.P. y Salas, J. (2010). Criterios de selección de metodologías de desarrollo de software. *Industrial data*, 13(2), 70-74.