UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

ESTUDIO COMPARATIVO DE METODOLOGÍAS ÁGILES PARA EL DESARROLLO DE APLICACIONES SAAS

CHAMBA LOPEZ JHONNY FABRICIO
INGENIERO DE SISTEMAS

MACHALA
2020

UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

ESTUDIO COMPARATIVO DE METODOLOGÍAS ÁGILES PARA EL DESARROLLO DE APLICACIONES SAAS

CHAMBA LOPEZ JHONNY FABRICIO
INGENIERO DE SISTEMAS

MACHALA
2020

UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

EXAMEN COMPLEXIVO

ESTUDIO COMPARATIVO DE METODOLOGÍAS ÁGILES PARA EL DESARROLLO DE APLICACIONES SAAS

CHAMBA LOPEZ JHONNY FABRICIO
INGENIERO DE SISTEMAS

REDROVAN CASTILLO FAUSTO FABIAN

MACHALA, 19 DE FEBRERO DE 2020

MACHALA
19 de febrero de 2020

Nota de aceptación:

Quienes suscriben, en nuestra condición de evaluadores del trabajo de titulación denominado Estudio comparativo de metodologías ágiles para el desarrollo de aplicaciones SaaS, hacemos constar que luego de haber revisado el manuscrito del precitado trabajo, consideramos que reúne las condiciones académicas para continuar con la fase de evaluación correspondiente.

REDROVAN CASTILLO FAUSTO FABIAN
0702739228
TUTOR - ESPECIALISTA 1

CÁRDENAS VILLAVICENCIO OSCAR EFRÉN
0703935312
ESPECIALISTA 2

JUMBO CASTILLO FREDDY ANIBAL
0704167949
ESPECIALISTA 3

Fecha de impresión: jueves 20 de febrero de 2020 - 10:00

Parte Escrita Examen Complexivo

por Fabricio Chamba

Fecha de entrega: 10-feb-2020 11:21a.m. (UTC-0500)
Identificador de la entrega: 1254814353
Nombre del archivo: CHAMBA-PT-011119.docx (389.82K)
Total de palabras: 4145
Total de caracteres: 21942

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, CHAMBA LOPEZ JHONNY FABRICIO, en calidad de autor del siguiente trabajo escrito titulado Estudio comparativo de metodologías ágiles para el desarrollo de aplicaciones SaaS, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las dispociones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 19 de febrero de 2020

CHAMBA LOPEZ JHONNY FABRICIO
1723373765

Parte Escrita Examen Complexivo

INFORME DE ORIGINALIDAD

2%	2%	0%	3%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	Submitted to Universidad de Lima Trabajo del estudiante	2%

Excluir citas	Activo	Excluir coincidencias	< 50 words
Excluir bibliografía	Activo		

DEDICATORIA

Dedico mi trabajo en primer lugar a Dios quien durante todo este tiempo me estuvo acompañándome, dándome fortaleza en seguir avanzando para cumplir mis metas. A mis padres que siempre estuvieron incentivándome a seguir adelante y que son la razón de mis logros.

Jhonny Fabricio Chamba López

AGRADECIMIENTO

Agradezco a mis padres, quienes con su apoyo incondicional me permitieron llegar hasta el final. A mis amistades, por el apoyo incondicional que me han permitiéndome crecer. Finalmente, a mi tutor el ingeniero Fausto Redrován, quién supo apoyarme antes y durante el proceso de titulación.

Jhonny Fabricio Chamba López

RESUMEN

En la actualidad la utilización de servicios a través de Internet, conocida también como SaaS (Software as a Service) propone una solución óptima tanto para empresas como usuarios, debido al ahorro en equipos físicos e infraestructura. Este modelo brinda al usuario el fácil acceso al software desde cualquier navegador, manejando una instancia del programa para varios usuarios, pero con una capa de datos independiente del resto. Sin embargo, para el equipo de programación conlleva un gran trabajo el agregar características, arreglar bugs entre otros problemas que se presentan en el desarrollo, por ello se presenta la necesidad de gestionar estas tareas mediante una metodología ágil que dinamice el proceso de desarrollo dirigido al modelo SaaS como tal. En el presente trabajo se elaboró un prototipo SaaS, el cual fue gestionado con la metodología ágil Kanban, como resultado de la comparación de varios criterios (con énfasis en las fases de verificación y validación) de las más conocidas metodologías ágiles que se adaptan mejor al modelo de desarrollo de aplicaciones SaaS.

Palabras Claves: Metodología Ágil, Kanban, SaaS, comparativa, desarrollo.

Abstract

Currently, the use of services over the Internet, also known as SaaS (Software as a Service) proposes an optimal solution for both companies and users, due to savings in physical equipment and infrastructure. This model gives the user easy access to the software from any browser, managing an instance of the program for multiple users, but with a separate data layer from the rest. However, for the programming team it takes a lot of work to add features, fix bugs among other problems that arise in the development, so the need to manage these tasks is presented through an agile methodology that energizes the development process aimed at SaaS model as such. In the present work, a SaaS prototype was developed, which was managed with the Kanban agile methodology, as a result of the comparison of several criteria (with emphasis on the verification and validation phases) of the best known agile methodologies that are best adapted to the SaaS application development model.

Keywords: Agile Methodology, Kanban, SaaS, comparative, development.

CONTENIDO

DEDICATORIA 4

AGRADECIMIENTO 5

RESUMEN 6

CONTENIDO 8

LISTA DE FIGURAS 9

LISTA DE TABLAS 10

1. INTRODUCCIÓN 11

1.1. Marco Contextual 12

1.2. Problema 12

1.3. Objetivo General 12

2. DESARROLLO 13

2.1. Marco Teórico 13

Solución del Problema 16

2.2. Resultados 18

3. CONCLUSIONES 21

BIBLIOGRAFIA 22

ANEXOS 24

LISTA DE FIGURAS

Figura 1. Burn down Chart Scrum 16
Figura 2. Burn down chart XP 17
Figura 3. Diagrama de flujo acumulado Kanban 18
Figura 4. Inicio de Sesión de Aplicación SaaS 24
Figura 5. Página de Inicio de la aplicación SaaS 24

LISTA DE TABLAS

Tabla 1. Comparativa de metodologías y Principios del Manifiesto ágil 14
Tabla 2. Criterios cuantitativos de selección para proyectos pequeños 20

1. INTRODUCCIÓN

Hoy en día el Cloud Computing está cambiando el modelo tradicional basado en productos a un modelo basado en servicios el cual proporciona a diferentes usuarios un espacio de recursos informáticos escalables, asegurando la flexibilidad en el acceso a los datos desde cualquier lugar con la ventaja de no preocuparse por el mantenimiento, ni por la seguridad. Por ende, este paradigma propone un modelo de negocio para las diferentes empresas que no se encuentran en posición de adquirir infraestructura en tecnología, racionalizando los gastos al objetivo principal del negocio [1]. Sin embargo, el adoptar estas tecnologías según

Mariano [2], conlleva un análisis previo de los riesgos que estos pueden presentar en cuestión de adaptabilidad y de seguridad a nivel de negocio, por tal motivo se recomienda tener buenas prácticas y contar con un departamento IT que solvente estos riesgos.

En este trabajo veremos unas de las capas principales que el Cloud Computing ofrece, el modelo SaaS (Software as a Service), este modelo ofrece a múltiples clientes el acceso a un software por demanda y solo es necesario una conexión a Internet, con esto el cliente no tendrá que preocuparse por la gestión de TI, ya que el proveedor se encargará de las gestiones necesarias para que el cliente se centre en sus tareas principales del negocio [3].

El presente trabajo tiene como objetivo el comparar metodologías ágiles orientadas al desarrollo de aplicaciones SaaS, teniendo en cuenta las fases de validación y verificación del software, debido a la necesidad de gestión de tareas dentro de un equipo de desarrollo que presentan este tipo de proyectos.

1.1. Marco Contextual

El presente trabajo es el resultado a un caso de estudio planteado como examen complexivo práctico el cual presenta un problema en cuanto al desarrollo de aplicaciones SaaS, principalmente en la gestión del staff de trabajo y las tareas que involucra el desarrollo de estas aplicaciones en la nube, así como las fases metodológicas por las cuales debe pasar este modelo de distribución y en el menor tiempo posible.

1.2. Problema

En el desarrollo de aplicaciones SaaS (Software as a Service), como modelo de distribución de software siempre está abierto a mejoras y por ende se presenta una necesidad de elegir una metodología que permita al equipo de proyecto organizarse de tal manera que puedan cumplir con los requisitos planteados en el menor tiempo posible y con adaptabilidad a cambios que pueden ir presentándose en el transcurso del desarrollo, respetando los principios metodológicos de verificación y validación del software.

1.3. Objetivo General

Elaborar un estudio comparativo de metodologías ágiles orientadas al desarrollo de aplicaciones SaaS mediante un análisis de sus características y técnicas, con énfasis en los procesos de verificación y validación del software para la selección de la más efectiva.

1.3.1. Objetivos Específicos

- Realizar una investigación bibliográfica sobre metodologías ágiles para el desarrollo de aplicaciones SaaS.
- Elaborar un cuadro comparativo entre metodologías ágiles para el desarrollo de aplicaciones SaaS mediante los principios del manifiesto ágil que incluyen los procesos de verificación y validación.
- Desarrollar un prototipo SaaS con la aplicación de la metodología ágil seleccionada.

2. DESARROLLO

2.1. Marco Teórico

2.1.1. Cloud Computing

Según Wang [4], define al Cloud Computing como un conjunto de redes habilitadas para la prestación de servicios que garantizan la escalabilidad y un correcto funcionamiento de acuerdo a las demandas que requiera el usuario, servicios que pueden ser accesibles de manera simple a través de Internet y a un coste muy bajo.

Parra [5] , reforma el concepto de Cloud Computing de una manera simple, la oportunidad de pagar por la prestación de un servicio que un negocio necesite permitiéndoles ser flexibles y adaptativos a sus necesidades, esto supone un ahorro en recursos informáticos permitiendo desempeñar las labores del negocio como tal.

También este paradigma está dirigido a diferentes áreas como la salud, gobierno entre otras, teniendo el mismo objetivo, facilitar la operatividad en las distintas áreas, facilitando los recursos informáticos a un bajo costo [6].

2.1.2. SaaS (Software as a Service)

SaaS es un modelo que proporciona funcionalidades de software a través de Internet por un coste, este modelo de negocio es beneficioso ya que elimina la necesidad de instalar la aplicación haciéndolos accesibles por medio de navegadores y desde teléfono móviles, además de los recursos informáticos que una aplicación basada en producto requeriría [7]. También

Sánchez [8] lo reitera, como las aplicaciones finales las cuales son ofrecidas a los usuarios por un coste de uso, brindándoles beneficios y asegurando que sus datos están siendo protegidos, permitiendo a los usuarios finales el centrarse en uso como tal del servicio para sus necesidades.

2.1.3. Desarrollo Ágil y SaaS

El desarrollo de software ágil cuenta con herramientas que dinamicen el proceso como tal, tienen como objetivo beneficiar al equipo con una buena gestión de tareas y roles, como también aumentar la productividad. Se centra en la recolección minuciosa de los requisitos para definir el alcance y la naturaleza con la que se va a trabajar, estimando costos y tiempo aproximado de la duración del proyecto [9].

SaaS como un modelo dentro del servicio que ofrece Cloud Computing, prestando servicios con una infraestructura basada en la nube accesible desde cualquier navegador, este modelo ofrece un tras fondo de infraestructura informática transparente para el usuario final pero con el beneficio de que funcione como el cliente lo espera con excepciones en ciertos casos de funcionalidad limitadas al servicio ya contratado [10].

Para lograr un buen producto SaaS se debe alinear con una metodología ágil que acepte cambios, que priorice a todo el equipo como un único conjunto para un mejoramiento continuo y que mantenga una versión estable que sirva de prototipo para así agregar funcionalidades que sean agregadas de manera transparente al usuario. El modelo de SaaS se alinea con los objetivos del ciclo de vida del desarrollo de software ágil por su flexibilidad a cambios y entregables de cada iteración permitiendo a los interesados decidir qué aspectos dejar o agregar para cumplir con los requerimientos establecidos [11].

2.1.4. Metodología Ágil

Según Amaya [12], esta metodología surge como alternativa a las metodologías tradicionales las cuales se derivan de los principios del manifiesto ágil [13], y se basan en un desarrollo iterativo e incremental. Estas metodologías ágiles se centran en la recolección de requisitos y están abiertos a cambios durante el proceso, así como la subdivisión del proyecto en iteraciones también conocidas como “Sprint”, y cada sprint se genera un entregable en un tiempo variado dependiendo de los requisitos que se encuentren en dicho sprint. Para que una metodología sea ágil, Valencia y Herrera [14], hace énfasis en los cuatro valores del manifiesto ágil, el primero hace prioridad a los individuos y las interacciones entre ellos sobre los procesos, el segundo hace relevancia sobre el software funcional por encima de la documentación, el tercero prioriza la colaboración del cliente por encima del contrato y por último, que sean abiertos al cambio sin la necesidad de seguir un plan establecido.

2.1.4.1. Principales metodologías ágiles para SaaS

En el siguiente cuadro se presentan las características más notorias de las principales metodologías ágiles para el desarrollo SaaS:

Tabla 1. Comparativa de metodologías y Principios del Manifiesto ágil

	Principio del Manifiesto Ágil	SCRUM	XP	Kanban
1	Satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor	✓	✓	✓

Tabla 1. (Continuación)

	Principio del Manifiesto Ágil	SCRUM	XP	Kanban
2	Aceptar cambios en cualquier etapa, para que el cliente tenga una ventaja competitiva	✓	✓	✓
3	Entregar software frecuentemente funcional desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas	✓	✓	✓
4	La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.	✓	✓	✓
5	Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.	✓	✓	✓
6	El diálogo cara a cara es el método más efectivo para comunicar información dentro de un equipo de desarrollo.	✓	✓	✓
7	El software que funciona es la medida principal del progreso.	✓	✓	✓
8	Los procesos ágiles promueven un desarrollo sostenible. Los promotores, los desarrolladores y usuarios deberían ser capaces de mantener una paz constante.	✓	✓	✓
9	La atención continua a la calidad técnica y al buen diseño mejora la agilidad.	✓	✓	✓
10	La simplicidad es esencial.	✓	✓	✓
11	Las mejores arquitecturas, requisitos y diseños surgen de los equipo organizados por sí mismos.	✓	✓	✓
12	En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.	✓	✓	✓

Fuente 1: Manifiesto ágil [13]

Como se puede apreciar, las metodologías ágiles presentadas en el cuadro de arriba, todas cumplen con los principios del manifiesto ágil [13], sin embargo las diferencias se resaltan en la manera con la cual cada una se implementa y pequeños detalles que las hacen únicas a cada una. Por un lado, tenemos a SCRUM, conocido por ser un marco de trabajo completo, esta metodología ágil define sus procesos mediante iteraciones llamadas “Sprint”, que por lo general dura desde una a cuatro semanas según las necesidades que se presenten. Esta metodología también maneja roles que se encargan de las necesidades del usuario final, un SCRUM master que se encarga que todo vaya acorde a los principios y desarrolladores, sin embargo, puede haber más miembros si es que el proyecto lo demande [15].

En según lugar tenemos a XP (eXtreme Programming), una metodología muy utilizada para equipos pequeños y con entregas rápidas como dice en su nombre, creada para solventar

los cambios que se presentaban a lo largo del desarrollo en donde la incertidumbre en los requisitos puede poner en riesgo el proyecto, así como un desarrollo incremental donde el cliente recibirá un software funcional en cada fin del ciclo [16].

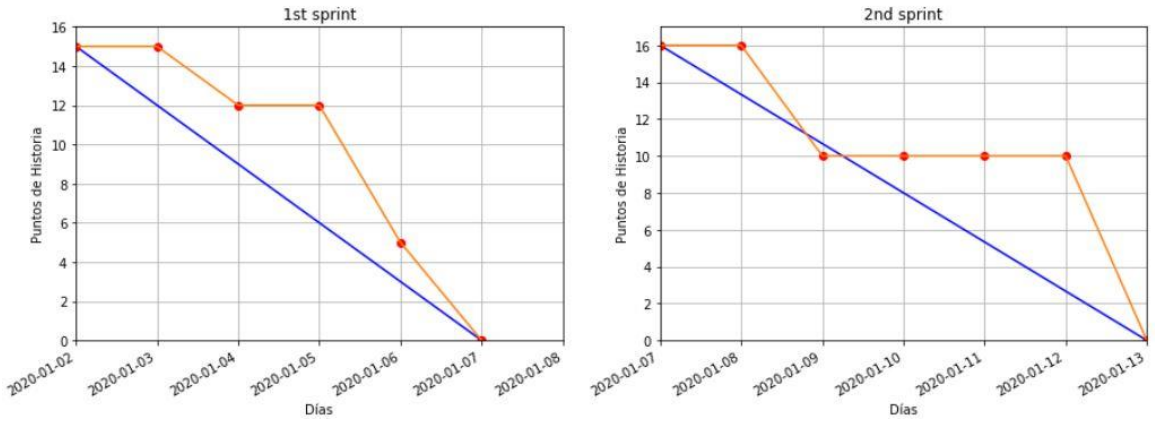
Y por último la metodología Kanban, una metodología desarrollada por Toyota para mejorar la producción, carece de iteraciones como otras metodologías ágiles, pero maximiza los procesos y prioriza la resolución de cuellos de botellas, parten de una historia de usuario que se mueve a través de las fases dibujadas en el tablero, el cual proporciona al director del proyecto y al equipo una visión clara de cómo van las tareas a lo largo del desarrollo [17].

Solución del Problema

2.1.5. Desarrollo de la metodología SCRUM

Si se utilizara SCRUM, como metodología para el desarrollo de la aplicación SaaS que se propone, tomando en cuenta que permite un amplio y ordenado marco de trabajo para proyectos, este se dividiría en dos Sprints los cuales constan de un total de diez días. Esta metodología se la subdivide en varias fases las cuales son: las actividades por realizar, las que están en proceso y las terminadas. La estimación está en puntos de historia para un mejor desempeño. Al inicio del proyecto, en las actividades estarán todas las historias de usuarios y mientras transcurre se irán trasladando a las siguientes fases hasta la culminación del proyecto.

Figura 1. Burn down Chart Scrum



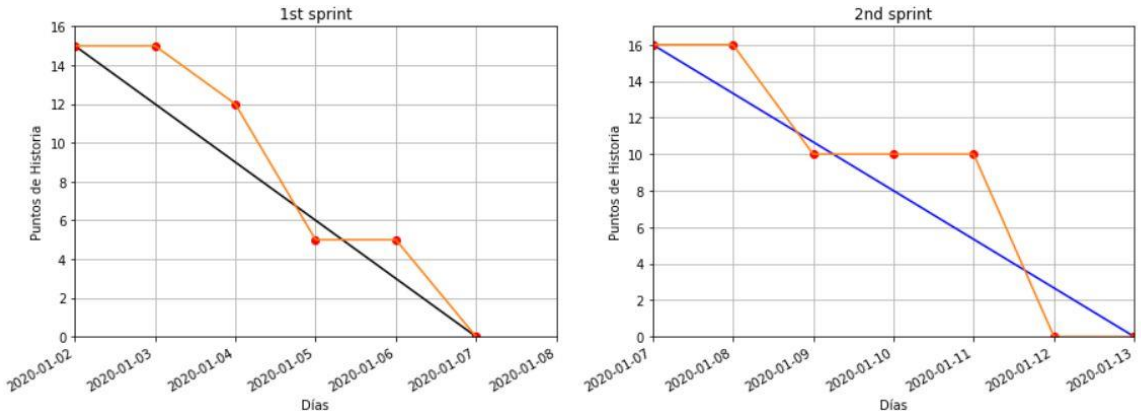
Fuente. Elaboración propia (basado en [18])

2.1.6. Desarrollo de la metodología XP (eXtreme Programming)

XP o programación extrema, como lo expresa en su nombre, prioriza la entrega rápida del producto lo más funcional posible. Manejada por iteraciones y por un flujo de trabajo donde inicia con la planeación, luego con los criterios de aceptación de los requisitos, las reuniones de equipo y la codificación en parejas, luego vuelve al inicio permitiendo una retroalimentación si diera el caso de que hubiera nuevos cambios. Se retroalimenta hasta que el ciclo de vida marque la culminación del proyecto.

En este caso se marcarían dos Sprints los cuales se manejan con puntos de historia y conforme avanza se irán disminuyendo hasta completar el número total de historias de usuario.

Figura 2. Burn down chart XP



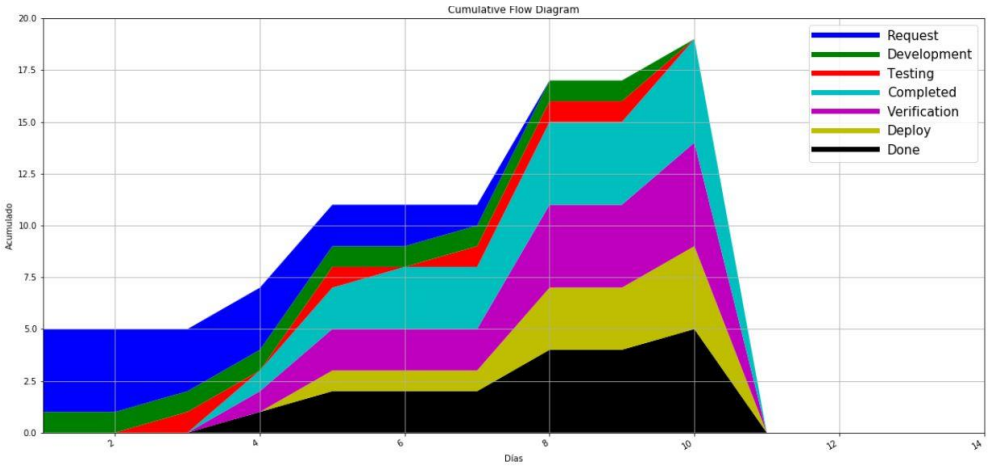
Fuente. Elaboración propia (basado en [18])

2.1.7. Desarrollo de la metodología Kanban

En la metodología Kanban, las tareas se distribuyen en varias fases graficadas a lo largo de un tablero: por realizar, en proceso y terminadas.

Al inicio del proyecto las actividades están por realizarse y conforme transcurre los días se irán moviendo a través del tablero hasta que estén todas en la fase de terminado. A continuación, se presenta el diagrama de flujo acumulado Figura 3, donde se visualiza el avance del proyecto a lo largo del tiempo establecido.

Figura 3. Diagrama de flujo acumulado Kanban



Fuente. Elaboración propia (basado en [19])

2.2. Resultados

Con la aplicación de las tres metodologías, se pudo elaborar un cuadro comparativo tomando los aspectos importantes que deben tener para ser elegida. La valoración refleja el grado de aplicación con la que la metodología aportará al criterio, siendo 5 el mejor cumplimiento y 1 implica que es adverso a cumplir con el buen desarrollo del proyecto bajo este criterio:

Presupuesto disponible

Para realizar un proyecto se debe estimar los costos que estos requerirán y dentro de estos, el costo de implementación de la metodología. A menor costo, mejor la metodología.

Tamaño del proyecto

La cantidad de personal que se requerirá para en un proyecto, es un aspecto importante a tener en cuenta, cada metodología maneja roles y estos una cierta cantidad de personas para poder lograr hacer funcionar una metodología. A mayor puntaje más efectiva para proyectos pequeños.

Tiempos limitados

Cada metodología se maneja en intervalos de tiempo definidos, por ser metodologías ágiles a diferencia de las tradicionales, manejan ciclos cortos en cada iteración. Esto implica que

las iteraciones perjudican a los proyectos pequeños y benefician a los grandes. Más valorado implicaría que aprovecha mejor los tiempos cortos.

Documentación necesaria

Al ser metodologías ágiles no manejan una documentación exhaustiva solo el código resultante al final de cada iteración. Menos documentación es mejor.

Personal necesario

Dependiendo del tamaño del proyecto, se necesitará personal que coordine las disciplinas en cada etapa del desarrollo. Mientras más grande el proyecto, la metodología se vuelve más compleja y por ende necesita más personal. Mientras menos personal se necesite será mejor.

Adaptabilidad a cambios

Un buen manejo a los cambios repentinos durante el proceso de desarrollo, permitiéndoles responder a tiempo.

Si acepta y maneja nuevos cambios con facilidad es mejor.

Cliente

Aspecto fundamental dentro de un proyecto, el cliente es quien proveerá los requerimientos y por ende debe estar dentro del desarrollo lo más cercano posible para concretar las necesidades planteadas. Este criterio evalúa el tiempo que el cliente interviene dentro de la metodología. Más intervención del cliente es mejor.

Fase de verificación

En esta fase responde a la pregunta: ¿Se está construyendo el software de manera correcta?, el cual evalúa los requisitos planteados en la planificación. Cuanto mejor se maneje o se tenga procedimientos más claros para esta fase, será mejor.

Fase de validación

En esta fase por el contrario responde a la pregunta: ¿Se ha construido el software correcto?, haciendo énfasis en lo que el cliente realmente espera. Mientras más intervención del cliente, la fase de validación aumenta en efectividad.

Tabla 2. Criterios cuantitativos de selección para proyectos pequeños

	SCRUM	XP	Kanban
Presupuesto disponible	1	5	4
Tamaño del proyecto	1	5	5
Tiempos limitados	1	4	5
Documentación necesaria	1	3	2
Personal necesario	1	2	4
Adaptabilidad a cambios	2	5	4
Cliente	5	4	3
Fase de verificación	3	5	3
Fase de validación	5	3	4
TOTAL	20	36	34

Fuente: Elaboración propia

Como se puede apreciar en la Tabla 2, se presenta los criterios de selección de una metodología ágil para proyectos pequeños, podemos ver que XP proporciona mayor ventaja sobre proyectos pequeños, mientras que Kanban tiene una pequeña aproximación a XP, y, por otro lado, SCRUM no cumple las características de cumplimiento para este tipo de proyectos. Entre XP y Kanban, los criterios de tiempo limitado y el personal necesario son los que hicieron la diferencia cuando se seleccionó la metodología para el desarrollo del prototipo (ver ANEXO A), y por tal motivo se utilizó Kanban, lo que permitió terminar satisfactoriamente el proyecto SaaS, producto final de este examen complexivo.

3. CONCLUSIONES

Al investigar diferentes metodologías ágiles para desarrollo SaaS, se determinó que las más utilizadas son SCRUM, XP y Kanban según el tipo de proyecto, y dicha investigación permitió comparar esos aspectos que destacan en cada metodología que son esenciales para el desarrollo óptimo de una aplicación SaaS.

La investigación que se realizó ha contribuido en la identificación de aspectos importantes al seleccionar una metodología ágil para el desarrollo de aplicaciones SaaS, que permiten resaltar las características más convenientes para este modelo de distribución.

Para el análisis comparativo se elaboró un cuadro con diferentes criterios de evaluación que toman en cuenta principios del manifiesto ágil y procesos de verificación y validación, lo que permitió establecer conclusiones con facilidad.

La comparativa reveló que, para proyectos tipo SaaS grandes haciendo énfasis en las fases de verificación y validación del software, SCRUM proporciona un amplio marco de trabajo para un equipo el cual requiere un amplio control de sus tareas dentro de un equipo de desarrollo, por otro lado, para proyectos pequeños, XP proporciona una rápida gestión para el desarrollo con entregas rápidas y cambios inmediatos. Mientras que Kanban está orientada a proyectos pequeños y medianos por ser manejado como una línea de optimización de procesos el cual proporciona con poco personal un ágil desarrollo en tiempos cortos.

Por último, se desarrolló exitosamente un prototipo (ver ANEXO B) de proyecto SaaS mediante la metodología ágil Kanban, que se seleccionó como resultado de la aplicación de la evaluación mediante el cuadro comparativo elaborado y de acuerdo al tipo de proyecto y tiempos de entrega.

BIBLIOGRAFIA

[1] R. R. Coello Yagual y L. M. Pico Versoza, «Análisis de las ventajas y desventajas del sistema de gestión de la seguridad de la información y su influencia en la competitividad de las empresas que utilizan Cloud Computing y Big Data en el Ecuador,» *INNOVA,* vol. III, nº 4, pp. 181-195, 2018.

[2] M. J. Benito Gómez, «Cloud computing: algo más que una tendencia,» *Red seguridad,* pp. 16-17, 2019.

[3] J. F. Herrera Cubides, N. Y. Gelvez García y D. A. López Sarmiento, «LMS SaaS: Una alternativa para la formación virtual,» *Ingeniare,* vol. XXVII, nº 1, 2019.

[4] L. Wang, J. Tao, M. Kunze, A. Castellanos Canales, D. Kramer y K. Wolfgang, «Scientific Cloud Computing: Early Definition and Experience,» IEEE, Rochester, 2008.

[5] M. J. Parra Royón, «Servicios de minería de datos en cloud computing,» Universidad de Granada, Granada, 2019.

[6] A. Abdulwahab Ali, K. Eltahir, N. Muawya y S. Haifeng, «Gender of Effect on Cloud Computing Services Adoption by University Students: Case study of Saudi Arabia,» *International Journal of Innovation: IJI Journal,* vol. VII, nº 1, pp. 155-177, 2019.

[7] M. Zárate Salas y L. Mendoza Colombo, «Cloud computing: a review of paas, iaas, saas services and providers,» *Lámpsakos,* nº 7, pp. 47-57, 2012.

[8] P. Palos Sánchez, «Estudio organizacional del cloud computing en empresas emprendedoras,» *3c Tecnología: glosas de innovación aplicadas a la pyme,* vol. VI, nº 2, pp. 1-16, 2017.

[9] F. Gutierrez Sierra, J. C. Ramos Rivera y F. Sáenz Blanco, «Establishment of Agile Teams for Software Development,» *Dimensión empresarial,* vol. XVI, nº 2, pp. 39-54, 2018.

[10] A. Rico Ortega, «Multi-tenancy Multi-target (MT2): An extension to cloud multi-tenant architectures for multi-service support in SaaS enterprise information systems,» Universidad de Granada, Granada, 2016.

[11] S. B. Chavez, A. E. Martín, N. Rodríguez, M. A. Murazzo y A. Valenzuela, «Metodología ÁGIL para el desarrollo SaaS,» de *XIV Workshop de Investigadores en Ciencias de la Computación*, La Plata, 2012.

[12] Y. D. Amaya Balaguera, «Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual,» *Revista de Tecnología,* vol. XII, nº 2, pp. 111-124, 2013.

[13] B. Kent, M. Beedle, B. Arie van, C. Alistair, C. Ward, F. Martian, G. James, H. Jim, H. Andrew, J. Ron, K. Jon, M. Brian, M. Rober, M. Steve, S. Ken y s. Jeff, «Manifiesto por el Desarrollo Ágil de Software,» 12 Febrero 2001. [En línea]. Available: https://agilemanifesto.org/iso/es/manifesto.html. [Último acceso: 13 Enero 2020].

[14] L. E. Valencia Ayala y E. Herrera Uribe, «DEL MANIFIESTO ÁGIL SUS VALORES Y PRINCIPIOS,» *Scientia et Technica,* vol. II, nº 34, pp. 381-385, 2007.

[15] G. Sobrevilla, J. Hernández y P. Velasco Elizondo, «Aplicando Scrum y Prácticas de Ingeniería de Software para la Mejora Continua del Desarrollo de un Sistema Ciber-Físico,» *Revista electrónica de Computación, Informática, Biomédica y Electrónica,* vol. I, nº 1, pp. 1-15, 2017.

[16] J. R. Laínez Fuentes, Desarrollo de Software ÁGIL, IT Campues Academy, 2015.

[17] M. D. Arango Serna, L. F. Campuzano Zapata y J. A. Zapata Cortes, «Mejoramiento de procesos de manufactura utilizando Kanban,» *Revista de Ingenierías: Universidad de Medellín,* vol. XIV, nº 27, pp. 221-233, 2015.

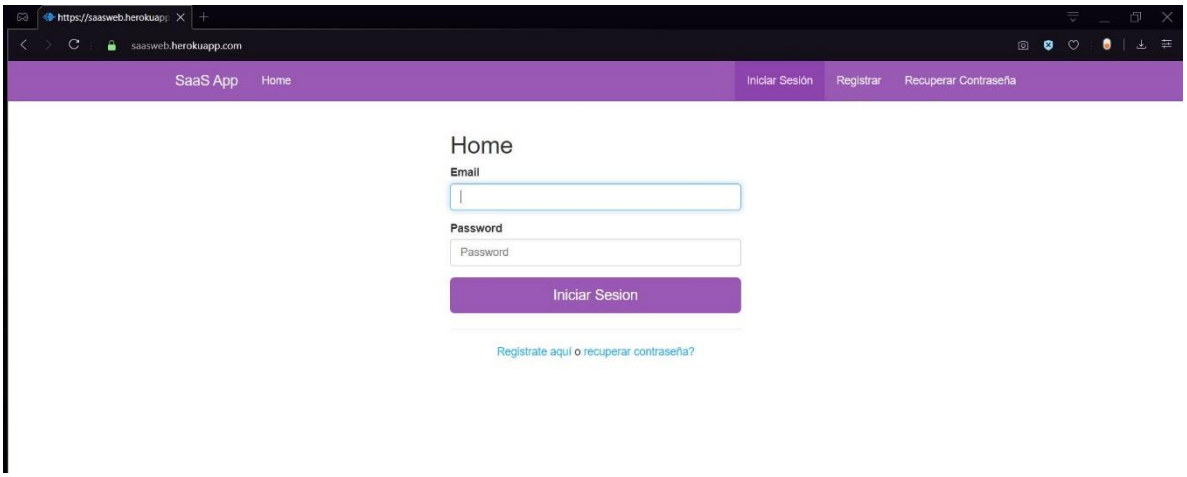
[18] A. Junaid y B. Saugata, «Improving Software Development Using Scrum Model by Analyzing Up and Down Movements on The Sprint Burn Down Chart: Proposition for Better Alternatives,» *International Journal of Digital Content Technology and its Applications,* vol. III, nº 3, pp. 109-115, 2009.

[19] P. Torres Letelier, «Una actividad para enseñar el uso de tableros kanban y diagramas de flujo acumulado,» *Actas de las XXI Jornadas de la Enseñanza,* pp. 288-295, 2015.

ANEXOS

ANEXO A: Inicio de Sesión de Aplicación SaaS

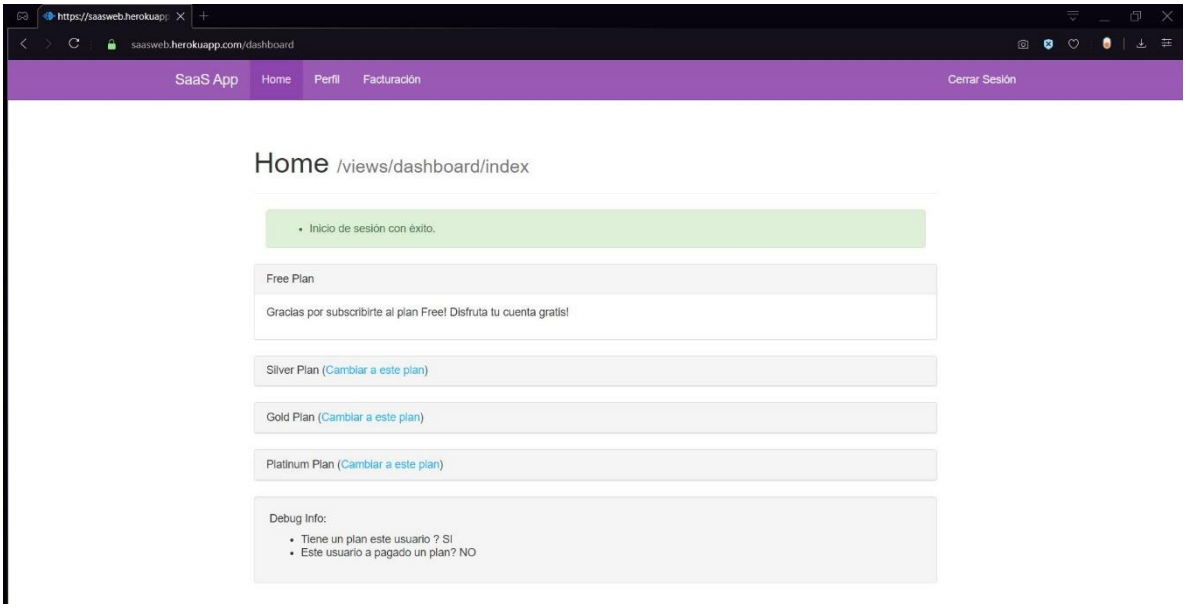
Figura 4. Inicio de Sesión de Aplicación SaaS



Fuente 2. Elaboración propia

ANEXO B: Página de Inicio de la aplicación SaaS

Figura 5. Página de Inicio de la aplicación SaaS



Fuente 3. Elaboración propia