

UNIVERSIDAD NACIONAL DE CHIMBORAZO



FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

Proyecto de Investigación previo a la obtención del título de Ingeniero en Sistemas y Computación

TRABAJO DE TITULACIÓN

APLICACIÓN DE LA METODOLOGÍA KANBAN EN EL DESARROLLO DEL SOFTWARE PARA GENERACIÓN, VALIDACIÓN Y ACTUALIZACIÓN DE REACTIVOS, INTEGRADO AL SISTEMA INFORMÁTICO DE CONTROL ACADÉMICO UNACH.

Autor(es):

Erika Dayana Yépez Llerena

Kleber Fabián Armijos Guillen

Tutor:

Ing. Pamela Buñay MsC.

Riobamba - Ecuador

2020

PÁGINA DE ACEPTACIÓN

Los miembros del tribunal de Graduación del proyecto de investigación de título: "APLICACIÓN DE LA METODOLOGÍA KANBAN EN EL DESARROLLO DEL SOFTWARE PARA GENERACIÓN, VALIDACIÓN Y ACTUALIZACIÓN DE REACTIVOS, INTEGRADO AL SISTEMA INFORMÁTICO DE CONTROL ACADÉMICO UNACH", presentado por los estudiantes Srta. Erika Dayana Yépez Llerena y el Sr. Kleber Fabián Armijos Guillén, dirigido por la MsC. Pamela Alexandra Buñay Guisñan. Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación escrito, con fines de graduación en el cual se ha constado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

MsC. Pamela Buñay.

Tutora del Proyecto

Firma

MsC. Diego Reina.

Miembro del Tribunal

Firma

MsC. Marlon Silva.

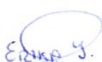
Miembro del Tribunal

Firma

Derecho de autoría

La responsabilidad del contenido de este proyecto de graduación corresponde exclusivamente a: Erika Dayana Yépez Llerena y Kleber Fabián Armijos Guillén autores, y la MsC. Pamela Alexandra Buñay Guisñan, Directora de Tesis, y al patrimonio intelectual de la Universidad Nacional de Chimborazo

Autores



.....

Erika Dayana Yépez Llerena

060513687-8



.....

Kleber Fabián Armijos Guillén

230020391-2

Directora del proyecto:



.....

MsC. Pamela Alexandra Buñay Guisñan

060424673-6

DEDICATORIA

Dedicamos este proyecto de investigación a nuestras familias quienes son pilares fundamentales en cada uno de nuestros logros, a nuestros seres queridos por su apoyo incondicional a nuestros amigos de vida, compañeros de clases, a los docentes de la Carrera de Ingeniería en Sistemas y Computación , quienes impartieron sus conocimientos e impulsaron a ser cada día mejor y tener un profesionalismo de calidad, a nuestra tutora de tesis, Ing. Pamela Buñay por apoyarnos con sus conocimientos obtenidos en su vida profesional para el desarrollo de este proyecto, al personal del departamento de desarrollo de la Universidad Nacional de Chimborazo, Ing. Cristian Quishpi, Ing. Henry Paca e Ing. Vicente Anilema por guiarnos durante todo el proceso de desarrollo y culminación de este proyecto.

Erika Yépez, Kleber Armijos

AGRADECIMIENTO

Agradecemos a Dios por acompañarnos y protegernos siempre, permitiéndonos culminar satisfactoriamente una de las etapas más importantes de nuestras vidas, agradecemos también a nuestros padres por su amor, paciencia, palabras de aliento y apoyo incondicional durante toda nuestra formación académica.

Agradecemos también a la Universidad Nacional de Chimborazo por ser la institución que cristalizó nuestro sueño, a sus Autoridades y Docentes, de manera especial al Ing. Danny Velazco, Director del Departamento Académico de la Universidad, por su apoyo y respaldo recibido al confiar en nuestro conocimiento, permitiéndonos realizar este proyecto de investigación en las instalaciones de la institución.

De la misma manera agradecemos a los docentes de la Escuela de Sistemas y Computación por haber compartido sus conocimientos a lo largo de nuestra carrera, formándonos académica y moralmente, en especial a nuestra tutora Ing. Pamela Buñay y docentes colaboradores Ing. Diego Reina e Ing. Marlon Silva quienes con su apoyo y guía nos permitieron culminar con el desarrollo del proyecto de investigación.

Agradecemos inmensamente al personal del departamento de desarrollo de la universidad, Ing. Cristian Quishpi, Ing. Henry Paca e Ing. Vicente Anilema por impartirnos su conocimiento adquirido durante su etapa profesional y brindarnos todo el apoyo necesario durante todo el proceso de desarrollo y culminación de este proyecto de investigación.

Erika Yépez, Kleber Armijos

ÍNDICE GENERAL

PÁGINA DE ACEPTACIÓN	2
DEDICATORIA	4
AGRADECIMIENTO	5
ÍNDICE GENERAL	6
LISTA DE TABLAS	8
LISTA DE FIGURAS	9
RESUMEN	10
ABSTRACT	11
INTRODUCCIÓN	12
CAPÍTULO I	13
PLANTEAMIENTO DEL PROBLEMA	13
1.1. Problema y Justificación	13
1.2. Objetivos	14
1.2.1. Objetivo General	14
1.2.2. Objetivos Específicos	14
CAPÍTULO II	15
MARCO TEÓRICO	15
2.1. Reactivos	15
2.1. Metodologías Ágiles	15
Manifiesto Ágil	15
2.1.1. Principales Metodologías Ágiles	16
2.1.2. Extreme Programming XP	16
2.1.3. AUP (AGIL UNIFIED PROCESS)	17
2.1.4. SCRUM	18
2.1.5. Iconix	19
2.1.6. Lean Software Development,	19
2.2. KANBAN	20
2.2.1. Kanban como Sistemas de Producción	20
Funcionamiento	20
Principios	21
Reglas	21
Tablero Kanban	23

2.2.2. Kanban aplicado al Desarrollo de Software	24
Reglas	24
Tablero Kanban	25
Herramientas Kanban.....	25
Roles.....	25
Beneficios	26
Kanban combinado con otras metodologías.....	26
Kanban, Scrum y Scrumban.....	26
CAPÍTULO III	28
METODOLOGIA	28
3.1. Tipo y Diseño de Investigación	28
3.2. Técnicas de recolección de Datos.....	28
3.3. Técnicas de Análisis e interpretación de la información	28
CAPÍTULO IV	41
RESULTADOS Y DISCUSIÓN	41
4.1. Resultados.....	41
4.2. Discusión	47
CONCLUSIONES	48
RECOMENDACIONES	49
REFERENCIAS BIBLIOGRÁFICAS	50
ANEXOS	53

LISTA DE TABLAS

Tabla 1: Scrum, Kanban y Scrumban.....	26
Tabla 2: Requisitos Funcionales	29
Tabla 3: Requisitos No funcionales	30
Tabla 4: Historias de Usuarios.....	36
Tabla 5: Análisis Pre-Post	47

LISTA DE FIGURAS

Figura 1: Proceso Kanban	21
Figura 2: Reglas Kanban.....	22
Figura 3: Tablero Kanban Básico	23
Figura 4: Flujo del procedimiento de Reactivos	29
Figura 5: Caso de Uso Sistema Reactivos.....	31
Figura 6: Caso de Uso Docente.....	31
Figura 7: Caso de Uso Docente-Validador.....	32
Figura 8: Vista Conceptual.....	32
Figura 9: Vista Lógica	33
Figura 10: Vista Física.....	33
Figura 11: Diagrama de Clases	34
Figura 12: Diagrama de Base de Datos	35
Figura 13: Actividades / Fuente: Autores	37
Figura 14: Flujo de trabajo / Fuente: Autores	37
Figura 15: Actividades Diarias / Fuente: Autores	38
Figura 16: Actividades en Progreso / Fuente: Autores.....	38
Figura 17: Actividades Finalizada / Fuente: Autores	39
Figura 18: Iteración en el Progreso	39
Figura 19: Control Entre Actividades / Fuente: Autores.....	40
Figura 20: Pregunta 1	41
Figura 21: Pregunta 2	42
Figura 22: Pregunta 3	42
Figura 23: Pregunta 4	43
Figura 24: Pregunta 5	43
Figura 25: Pregunta 6	44
Figura 26: Pregunta 7	44
Figura 27: Pregunta 8	45
Figura 28: Pregunta 9	45
Figura 29: Pregunta 10.....	46
Figura 30: Pregunta 11	46
Figura 31: Resultado Pre-Post.....	48

RESUMEN

Los desarrolladores de software se encuentran en constante innovación, aplicando nuevas metodologías en el proceso de desarrollo. Para seleccionar una metodología adecuada dependerá de los requisitos funcionales y no funcionales del software a desarrollar. KANBAN es una metodología ágil de desarrollo de software que reduce el tiempo de iteración entre actividades y mejora la calidad de desarrollo de cada proceso evitando cuellos de botella.

Actualmente la Universidad Nacional de Chimborazo cuenta con su repositorio de reactivos desactualizados y duplicados, cumpliendo parcialmente con las especificaciones técnicas requeridas tales como: los verbos del resultado de aprendizaje no se encuentran clasificados según la taxonomía de Bloom, la operación cognitiva no se relaciona adecuadamente con el verbo elegido, la estrategia no es la adecuada respecto a la operación cognitiva seleccionada, lo cual retrasa el proceso de aprobación del reactivo. Por tal motivo se desarrolló un software que permite la generación, validación y actualización de reactivos, con la finalidad que la Universidad Nacional de Chimborazo cuente con una herramienta propia para la gestión de reactivos. El software desarrollado forma parte del SICOA de la Universidad Nacional de Chimborazo, el mismo que se desarrolló aplicando la metodología ágil KANBAN como metodología ágil de desarrollo, utilizando la arquitectura orientada a dominio y la tecnología ASP.Net Core.

Finalmente se midió la usabilidad del sistema por medio de encuestas expertos, docentes de la Universidad Nacional de Chimborazo, donde se determinó que los usuarios recomiendan su uso en un 80%.

Palabras clave: Metodologías Ágiles, desarrollo de software, KANBAN, reactivo, usabilidad.

ABSTRACT

Software developers are constantly innovating and applying new methodologies in the update process. To select an appropriate methodology it will depend on the functional and non-functional requirements of the software to be developed. KANBAN is an agile software development methodology that reduces the iteration time among activities and improves the quality development of each process by avoiding delays.

Nowadays, at the National University of Chimborazo there is a repository of outdated and duplicated exams, and those partially fulfill the technical specifications such as: the verbs at the learning result parameters are not classified according to Bloom's taxonomy, the cognitive operation is not adequately related with the chosen verb, the strategy is not adequate, respect to the selected cognitive operation, it delays the exam approval process. For this reason, a software that allows the generation, validation and update of exams was developed, this way the National University of Chimborazo now has its own tool for exams management. The software is part of the SICOA of the National University of Chimborazo, which was developed by applying the KANBAN methodology as an agile development methodology, using a domain-oriented architecture and ASP.Net Core technology.

Finally, the usage of the system was measured through surveys done by expert professors at the National University of Chimborazo, and it was determined that 80% of users recommend its use.

Keywords: Agile Methodologies, software development, KANBAN, exam, usage.

Translation reviewed by:



MsC. Edison Damián



INTRODUCCIÓN

En los últimos tiempos los reactivos se han convertido en un instrumento primordial de evaluación para las diferentes Instituciones de Educación Superior (IES), permitiéndoles mejorar tanto su nivel académico como la calidad de educación que brindan a los estudiantes del país. Específicamente en la provincia de Chimborazo la Universidad Nacional de Chimborazo realiza sus repositorios de reactivos apoyándose en el software EXAMVIEW, el mismo que permite la creación y exportación de exámenes.

Por tal motivo se desarrolla un software que genera, valida y actualiza repositorios de reactivos, el cual está incluido en el Sistema Informático de Control Académico Institucional (SICOA), mismo que se convierte en una herramienta de control enfocada a los resultados de aprendizaje general (RAG) y resultados de aprendizaje específico (RAE) a estudiantes. Además, permite a los docentes analizar los resultados obtenidos e identificar las temáticas que deben ser retroalimentadas pedagógicamente.

Para el desarrollo del software se implementó la metodología ágil KANBAN la misma que tiene como finalidad la reducción del tiempo en iteración entre actividades, centrándose en el progreso del trabajo. Además KANBAN evita las tareas acopiadas o comúnmente conocidas como cuello de botellas.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1. Problema y Justificación

La gestión de reactivos en las instituciones de educación superior públicas y privadas del Ecuador responden de forma poco adecuada al momento de validar los reactivos. Siendo llevado este proceso de forma manual en algunas instituciones de la ciudad de Riobamba, es por ello que se considera importante plantear una visión que permitan crear software que automatice el proceso de validación, generación y actualización de reactivos tanto de fondo y forma para posteriormente tener una base de reactivos de buena calidad que permita identificar el grado de aprendizaje de manera general y específica por parte de las autoridades hacia los estudiantes. La validación de los ítems asegura la eficiencia y el acceso a contar con una base de reactivos de excelente calidad, permitiendo de este modo un constante seguimiento al proceso.

Actualmente la Universidad Nacional de Chimborazo cuenta con su repositorio de reactivos desactualizados y duplicados, cumpliendo parcialmente con las especificaciones técnicas requeridas tales como: los verbos del resultado de aprendizaje no se encuentren clasificados en la taxonomía de Bloom, la operación cognitiva no se relaciona adecuadamente con el verbo elegido, la estrategia no es la adecuada respecto a la operación cognitiva seleccionada, lo cual retrasa el proceso de aprobación del reactivo mismo que se lo realiza de forma manual y denota malestar en la planta docente al momento de generar mejoras a los reactivos; dificultando así que se refleje un resultado óptimo en la evaluación del proceso enseñanza-aprendizaje.

La UNACH no dispone de un aplicativo web que permita automatizar el proceso de gestión de reactivos, por lo que la implementación del Sistema de Gestión de Reactivos UNACH brinda respuestas eficientes en cuanto al tiempo tanto de creación de un reactivo como la validación de forma y fondo de los mismos. Se desarrolló un software que permite la generación, validación y actualización de reactivos, con la finalidad que, la Universidad Nacional de Chimborazo cuente con una herramienta propia para la creación de reactivos, la misma que aportará al control del proceso de evaluación. El software desarrollado forma parte del SICOA de la Universidad Nacional de Chimborazo, el mismo que se desarrolló aplicando la metodología ágil KANBAN, siendo esta una metodología basada en la gestión de las actividades de acuerdo con el tiempo establecido.

1.2. Objetivos

1.2.1. Objetivo General

Aplicar la metodología KANBAN en el desarrollo del software para generación, validación y actualización de reactivos integrado al Sistema Informático de Control Académico de la Universidad Nacional de Chimborazo.

1.2.2. Objetivos Específicos

- Investigar la metodología KANBAN y sus etapas correspondientes para su aplicación en el sistema de reactivos.
- Especificar los requerimientos del software para la gestión de reactivos.
- Desarrollar un software aplicando la metodología KANBAN, para la generación, validación y actualización de reactivos.

CAPÍTULO II

MARCO TEÓRICO

2.1. Reactivos

Para (Kleen, 2003) es una pregunta a contestar, afirmación a valorar, problema a resolver, característica a cubrir o acción a realizar; están siempre contenidos en un instrumento de evaluación específico; tienen la intención de provocar o identificar la manifestación de algún comportamiento, respuesta o cualidad. Los reactivos seleccionan la información que es relevante para la evaluación. En la Universidad Nacional de Chimborazo los reactivos de una asignatura se crean en base al verbo del resultado de aprendizaje de cada unidad, dicho verbo pertenecerá a la taxonomía de BLOOM y a una operación cognitiva y estrategia determinada, estos tres parámetros deberán tener concomitancia entre sí (Figura 4).

2.1. Metodologías Ágiles

Las metodologías ágiles resuelven los problemas surgidos, tras la masificación del uso del computador, dado que las expectativas y necesidades por parte de los usuarios se hicieron más urgentes y frecuentes. Fue así como a comienzo de los 90 se crearon propuestas metodológicas para lograr resultados más rápidos en el desarrollo de software sin disminuir su calidad. Una década más tarde, en febrero de 2001 en Utah-EEUU, se reunieron 17 empresarios de la industria del software y como resultado del debate referente a las metodologías, principios y valores que deben regir el desarrollo de software de buena calidad, en tiempos cortos y flexible a los cambios, se aceptó el término ágil para hacer referencia a nuevos enfoques metodológicos en el desarrollo de software. Esta asamblea creó "The Agile Alliance", sin finalidad de lucro promoviendo aspectos relacionados al desarrollo ágil en el software, brindando asesoramiento a organizaciones. Como punto de partida o base fundamental de las metodologías ágiles se redactó y proclamó el manifiesto ágil (Uribe, 2007).

Manifiesto Ágil

Según (Canós, Letelier, & Penadés, 2012) el Manifiesto evalúa los siguientes aspectos:

- **Al desarrollador y las actividades del equipo de desarrollo sobre procesos y herramientas.** El recurso humano es el principal factor de éxito de un proyecto software. Es de suma importancia construir un buen equipo que los procesos y herramientas a utilizar, cometiendo errores al construir el entorno antes de que el

equipo de adapte al mismo. Es recomendable que el equipo establezca su propio entorno de desarrollo en base a sus necesidades.

- **Desarrollar software que funciona más que conseguir una buena documentación.** Las metodologías ágiles reconocen y respetan la importancia, tiempo y costo de mantener una documentación completa y actualizada, sin embargo, hace énfasis en “no producir documentos a menos que sean estrictamente necesarios”. Estos documentos deben ser cortos y centrarse en lo fundamental.
- **Colaboración entre el cliente y el contrato.** El cliente establece una relación con los desarrolladores, desde el comienzo hasta la culminación del proyecto. El cliente es quien sabe qué necesita o desea, el más indicado para corregir o hacer recomendaciones en cualquier momento del proyecto. Esta colaboración será la que asegure el éxito del proyecto.
- **Responder a cambios, no seguir un plan estrictamente.** Teniendo en cuenta que los cambios pueden surgir en el transcurso del desarrollo (nuevos requisitos, tecnología, equipo, etc.) determina también el éxito o fracaso del mismo. La planificación debe ser adaptable. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para los siguientes meses.

2.1.1. Principales Metodologías Ágiles

Entre los principales métodos ágiles tenemos el XP (eXtreme Programming), Scrum, Iconix, AUP entre otras. Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. Nos lo proponen para muchos clientes esta flexibilidad será una ventaja competitiva y porque estar preparados para el cambio significa reducir su coste (Figueroa, Solís, & Cabrera, 2008).

2.1.2. Extreme Programming XP

Metodología ágil que potencia al equipo de desarrollo con el cliente para alcanzar un buen ambiente laboral, basándose en retroalimentaciones entre los desarrolladores y el cliente, afrontando los cambios que se presenten a lo largo de la etapa de desarrollo del sistema XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Utilizada para proyectos cambiantes con un alto grado de adaptabilidad al equipo técnico. Los principios y prácticas son de

sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP. (Letelier & Penadés, 2012).

Características Esenciales

- **Historias de usuario:** Técnica utilizada para especificar los requisitos del software. Las historias de usuario son flexibles, cada historia de usuario debe ser delimitada y entendible y medible por un número de semanas específicas.
- **Roles:** Programador, Cliente, Encargado de pruebas (Tester), Encargado de seguimiento (Tracker), Entrenador (Coach), Consultor, Gestor (Big boss).
- **Proceso y prácticas:**
 - 1) El valor de negocio a implementar es definido por el cliente.
 - 2) Esfuerzo estimado por el desarrollador.
 - 3) La construcción es elegida por el cliente.
 - 4) Valor de negocio construido por el desarrollador.
 - 5) Repetir el proceso.

XP tiene seis fases en su ciclo de vida:

- Explorar
- Planificar Entrega
- Iterar actividades
- Poner en producción
- Brindar Mantenimiento
- Fin del ciclo de vida.

2.1.3. AUP (AGIL UNIFIED PROCESS)

Metodología utilizada comúnmente para el desarrollo de software en el área de los negocios que utiliza técnicas y conceptos afines a la metodología RUP, lo que la convierte en una versión simplificada de esta. AUP, es una metodología que adopta muchas técnicas ágiles utilizadas en XP y las formalidades de RUP, y su objetivo es adaptarse a las necesidades del proyecto, contrario a la filosofía de las metodologías tradicionales (Alvear & Quintero, 2015).

Fases:

- 1) **Concepción**, comprensión común entre el equipo de desarrollo del alcance del nuevo sistema y el cliente para definir una o varias arquitecturas candidatas para el mismo.

- 2) **Elaboración**, los desarrolladores profundizan los requisitos del sistema de acuerdo a los requerimientos establecidos.
- 3) **Construcción**, el sistema es desarrollado y probado completamente dentro del ambiente de desarrollo.
- 4) **Transición**, el sistema es llevado a entornos de preproducción y se somete a pruebas de validación y aceptación, finalmente se despliega en los sistemas de producción.

2.1.4. SCRUM

Es un marco de referencia para crear software complejo y entregarlo a tiempo de forma sencilla, mediante la aplicación de un conjunto de directrices a seguir por los equipos de trabajo y el uso de roles determinados. Scrum utiliza el concepto de Equipos Scrum, que son grupos de trabajo donde los miembros juegan roles específicos. Scrum considera que los desarrolladores son seres humanos que comenten errores, piensan en nuevas ideas en el camino y muchas características más, por dichas características se pensaría que se puede incumplir los plazos de entrega y tener muchos errores en el producto, pero, al contrario, ayuda a evitar estos problemas (Dimes, 2015).

Principios

1. **Product Backlog**: lista priorizada de características o funcionalidades que deberá tener el producto, estas funcionalidades se las obtiene de los usuarios potenciales, colegas y personas relevantes responsables del producto. Al ser una lista prioritaria se trabajará primero en los elementos más urgentes, evitando dedicar incontables horas o incluso días a aquellas características menos indispensables dejando de lado las más importantes.
2. **Iteraciones (mini hitos)**: realizar el trabajo en cortos ciclos iterativos que van desde una semana hasta un mes, este periodo es generalmente denominado iteración o sprint. El resultado de cada iteración deberá ser un producto listo para entregar, si alguna funcionalidad surge tardíamente durante la iteración es una señal de posibles cuellos de botella, por ende, deberá ser solucionado rápidamente.
3. **Retroalimentación**: el equipo revisa el producto final y lo presenta al grupo de interés, en base a los comentarios obtenidos el equipo actualiza la pila del producto e iteraciones.

2.1.5. Iconix

El proceso de ICONIX maneja casos de uso, como el RUP, es relativamente pequeño y firme, como XP, pero no desecha el análisis y diseño que hace XP. Este proceso también hace uso aerodinámico del UML mientras guarda un enfoque afilado en el seguimiento de requisitos. El proceso se queda igual a la visión original de Jacobson del manejo de casos de uso, esto produce un resultado concreto, específico y casos de uso fácilmente entendible, que un equipo de un proyecto puede usar para conducir el esfuerzo hacia un desarrollo real. El enfoque de esta metodología es flexible y abierto; siempre se puede seleccionar de los otros aspectos del UML para complementar los materiales básicos (Scott, 2001).

Posee cuatro fases y cuatro hitos de fácil comprensión y aplicación. Cada paso está establecido por requisitos trazando la capacidades de interactuar entre las diferentes actividades que se realiza en el desarrollo del software (Navarro, y otros, 2017).

2.1.6. Lean Software Development,

Toyota, una empresa que en la década de los 80 revoluciona la industria automotriz con su sistema de producción que promueve la eliminación de desperdicios, resalta la cadena de valor del producto, manufactura bajo demanda y se enfoca en el personal que agrega valor. La prioridad de este sistema es la velocidad: capacidad de satisfacer la demanda del mercado y la perfección: solo siendo perfectos de puede ser rápidos (Omaña & Cardenas, 2011).

Los autores (Poppendieck & Poppendieck, 20003) recomiendan la implementación de seis disciplinas:

- Organizar el área de trabajo
- Establecer estándares (denominación, codificación, interfaz gráfico)
- Control de versiones
- Procesos de construcción
- Integración frecuente
- Fijar políticas de pruebas (alcance, niveles de defectos, automatización y frecuencia)

2.2. KANBAN

2.2.1. Kanban como Sistemas de Producción

En la década de 1940 Toyota crea el termino Just in Time (JIT) por la necesidad de mantener las mejoras continuas, estimulando el máximo rendimiento en sus procesos de fabricación. Kanban se ha considerado un sinónimo de este término que fue diseñado para el control de inventario y reducción de desperdicios. Toyota originalmente utilizo tarjetas conectadas a diversos contenedores, comunicando que materiales deben reponer y que materiales se deben recargar. Actualmente existe variaciones de las tarjetas Kanban incluyendo tableros y sistemas electrónicos (ORDYSINSKI, 2013).

Por su parte (Bermejo, 2011) alega que Kanban se deriva de dos palabras japonesas, kan, que significa “visual”, y ban, “tarjeta”, Kanban se denomina como una metodología de producción u organización del trabajo basada en señales visuales para la gestión del esfuerzo y dedicación del equipo de producción. Por su parte (Palao Villasante, 2014) menciona que Kanban permite identificar atascos en la producción, mejorar el tiempo de servicio de tareas y mejorar la calidad en el proceso de producción. Además, El sistema Kanban controla el flujo de recursos en procesos de producción mediante el uso de tarjetas, las cuales indican el abastecimiento de material o producción de piezas, de acuerdo a la demanda y consumo del cliente. Kanban también puede entenderse como un sistema de producción, que se enfoca en el flujo de materiales a través de señales que indican cuando debe producirse el producto y cuando debe reabastecerse la materia prima.

Funcionamiento

Kanban inicia la línea de fabricación una vez que los productos son retirados por el cliente, siendo la demanda guiada por el cliente emitiendo una señal Kanban para la fabricación de un nuevo producto o rellene un punto de stock. Como primer paso se debe definir la cantidad ideal de productos a entregar, siendo la grande la cadena de producción, evitando reducir las existencias (Córdova, 2012).

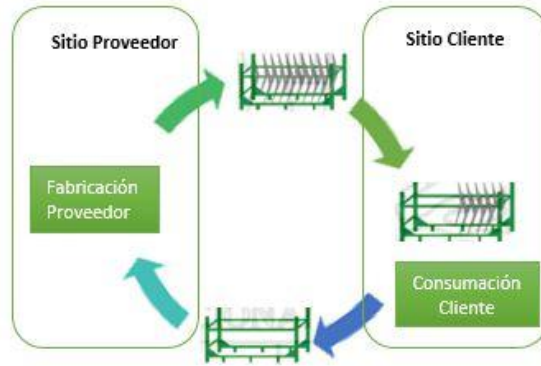


Figura 1: Proceso Kanban

Fuente: (Palao Villasante, 2014)

Principios

Según (Arango, Campuzano, & Zapata, 2015) Kanban tiene los siguientes principios:

- **Calidad:** Intenta hacer bien todo lo que se pretende realizar y mejorarlo si es posible.
- **Minimización:** Hacer justo lo necesario, sin enfocarse en actividades secundarias o innecesarias (principio YAGNI).
- **Mejora continua:** Ir mejorando continuamente los procesos, de acuerdo a los objetivos a lograr y alcanzar.
- **Flexibilidad:** Priorizar las tareas entrantes según se las necesite, decidir las tareas a realizar acorde a los faltantes o pendientes.
- **Construcción y mantenimiento:** mantener una relación a largo plazo con proveedores.

Reglas

Para (Estrada, 2006) Kanban se basa en 6 reglas importantes que se muestran en la figura 1, las mismas se detallan más adelante:

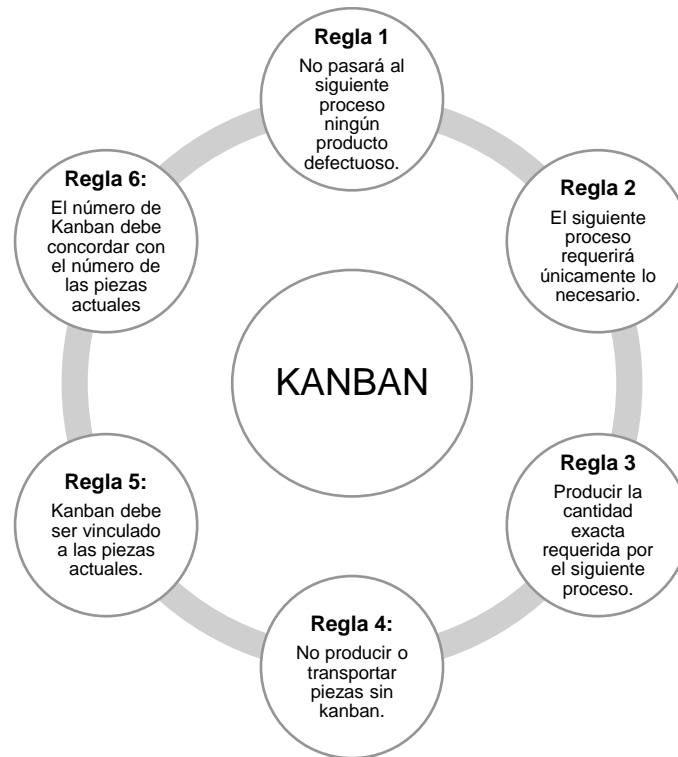


Figura 2: Reglas Kanban

Fuente: Los autores

- **Regla 1**
 Producir partes defectuosas significa que el material, equipo y mano de obra han hecho piezas que no pueden venderse, esto impide lograr la meta comparativa de reducir costos. Cuando se identifica las piezas defectuosas, se debe tomar medidas preventivas para evitar que vuelva a ocurrir.
- **Regla 2**
 Pueden surgir diversos problemas (trabajo innecesario, inventario excesivo, se producen piezas innecesarias en lugar de las se requiere en el momento) al producir demasiadas piezas o muy rápido.
- **Regla 3**
 Se debe producir las piezas en orden y mantener un inventario mínimo de piezas, para lograrlo no se debe producir más piezas de las que se indica en Kanban o en un orden diferente. Al cumplir esta regla se puede estandarizar el trabajo y reducir los costos.
- **Regla 4**

No se debe manufacturar las piezas sin las instrucciones establecidas en Kanban ni las innecesarias, se debe mantener un balancear en la producción.

- **Regla 5**

Kanban debe mantenerse siempre actualizado con las piezas que se hayan produciendo, de este modo se podrá localizarlas y monitorearlas con mayor rapidez.

- **Regla 6**

Controlar que el número registrado en Kanban concuerde con el número de piezas producidas, para mantener identificado el tiempo y a la cantidad restante, en caso que el proceso se retrase se requerirá personal y material en exceso para cumplir con el objetivo satisfactoriamente.

Tablero Kanban

Sistema de información físico asociado a la metodología lean, en el tablero se refleja las actividades establecidas en proceso determinado y el estado en el que se encuentra cada uno de ellas. Con esta herramienta las personas asumen ciertas responsabilidades, visualizando los logros y dificultades en el flujo del proceso rápidamente. Además, el equipo se enfoca en terminar las tareas que se tienen asignadas y no acumular tareas iniciadas (Torres, 2011).

En la figura 3 se observa un ejemplo básico de un tablero Kanban, consta de 3 columnas que indican el estado de cada actividad, en este caso se nombra a los estados como: ToDo, Doing y Done, se los puede denominar también En Cola, En Ejecución y Finalizado o Por Hacer, Haciendo y Listo. El número de columnas y la denominación de las mismas las establece el equipo de trabajo.



Figura 3: Tablero Kanban Básico

2.2.2. Kanban aplicado al Desarrollo de Software

David Anderson fue pionero en implementar esta metodología para el desarrollo de software, en 2004 bajo la guía de Don Reinertsen utilizó Kanban en un proyecto de TI de Microsoft. Kanban tiene como objetivo primordial determinar tareas por hacer y cambiar su prioridad en función de los acontecimientos, además con Kanban la cadena de trabajo está visible para todos, y en el caso de existir atascos es mucho más fácil conocer lo que se está produciendo, estos principios fueron considerados por Toyota y actualmente son muy útiles en el desarrollo de software (BBVAOPEN4U, 2018).

Reglas

Las tres principales reglas de Kanban según (Pérez, 2012) son las siguientes:

1. Visualizar el flujo de trabajo.

Dividir el trabajo en partes o tareas, escribirlas en una tarjeta y colocarla en la columna inicial, las columnas o estados pueden ser tantas como el equipo considere necesario que pase cada tarea. El objetivo primordial de esta regla es que el trabajo a realizar quede claro, visualizar en que está trabajando cada miembro del equipo y que todos tengan algo que hacer, siempre teniendo en cuenta las prioridades de cada tarea.

2. Determinar el límite del WIP (Work In Progress, Trabajo en Curso).

Limitar el número de tareas que se pueda realizar en cada estado del flujo de trabajo, independientemente de la magnitud del proyecto siempre hay una cantidad óptima. El principal objetivo de esta regla es detectar cuellos de botella fácilmente para buscar soluciones, muchas de las veces la solución más eficiente sería la colaboración del equipo que tenga procesos libres y pueda aceptar nuevos ítems.

3. Controlar el tiempo en completar una actividad (Lead time).

Inicia desde su petición hasta su entrega, mientras que cycle-time inicia desde que una actividad comienza hasta su finalización, es decir mide el rendimiento del proceso. Es indispensable optimizar estas métricas para el control y una mejora continua. Si se desea calcular el Throughput (rendimiento de trabajo) o cantidad de ítems que un equipo puede terminar en un periodo de tiempo determinado se debe dividir el CycleTime por el WIP.

$$\text{Throughput} = \text{CycleTime} / \text{WIP}$$

Tablero Kanban

Proporciona visibilidad de todo el proceso de software, mostrando el trabajo asignado a cada miembro del equipo, además indica las prioridades de cada tarea y resalta los cuellos de cuello de botella existentes, de este modo el equipo se concentra en los problemas que bloquean o impiden el proceso buscando una solución rápida, así se minimizará los defectos y se mantendrá un flujo estable. El uso de un tablero Kanban permitirá también equilibrar la demanda con el rendimiento del trabajo liberado por el equipo, consiguiendo un ritmo de desarrollo sostenible, un mayor rendimiento del equipo y elevar la calidad del producto final. De esta forma se reduce el tiempo del ciclo (leas time) y se incrementa la confianza del cliente al entregar avances de su producto regularmente (Guzmán, Islas, Corona, & Méndez, 2014).

Herramientas Kanban

Un tablero Kanban hecho de manera artesanal con post-it, cinta aislante y que este visible en la pared es el más recomendable, sin embargo, en muchas circunstancias es necesario usar un tablero integrado en una herramienta de software ya sea porque el equipo de trabajo esta geográficamente distribuido o por distintas razones. Actualmente existen una gran variedad de herramientas Kanban, (Garzas, 2012) considera que las siguiente las herramientas son las más populares:

- LeanKit Kanban
- AgileZen
- Flow
- Kanban Board
- SmartQ
- Simple-Kanban
- Swift-Kanban
- Target Process
- Upstart
- Kanbanize (gratis)

Roles

Según (Kniberg, Skarin, Poppendieck, & Anderson, 2010) Kanban no estable ningún rol en absoluto. Esto no quiere decir que no se pueda tener un papel de dueño del producto, Kanban es libre de añadir roles adicionales en caso de ser necesarios, siempre y cuando

se esté seguro que los roles van a añadir valor y no generar conflictos con otros elementos del proyecto. Establecer roles puede ser una gran idea al tratarse de proyectos muy grandes, permitiendo que el “jefe” sincronice los múltiples equipos de trabajo con el o los dueños del producto. Sin embargo, es importante guiarnos en la mentalidad general de Kanban (“menos es más”) e iniciar con menos.

Beneficios

Los principales beneficios para (Shore, 2010) y (Crisp, 2010) son:

- Es muy flexible y permite detectar cualquier problema existente y ajustar el flujo de trabajo para obtener mejores resultados.
- Beneficiando el flujo visual mediante tarjetas de colores distribuidas en el mismo tablero. La digitalización del tablero Kanban tiene la facilidad de acceder a su flujo desde cualquier sitio para comunicarse con el equipo de desarrolladores.
- Reduce el tiempo de espera y el dedicado a la asignación de tareas mediante el flujo constante de tareas.
- Visibilidad en tiempo real de los cuellos de botella.
- Desarrollo de software ágil sin la necesidad de tener que usar iteraciones de compromiso fijo de tiempo fijo como los sprints de Scrum.

Kanban combinado con otras metodologías

Kanban suele ser combinado con otras metodologías como herramienta de apoyo, de la misma forma en la que diversas metodologías y filosofías de desarrollo son, muchas veces, compatibles entre sí, permitiendo combinar y crear métodos que cumpla con necesidades del equipo y la organización. Un ejemplo de una metodología híbrida publicada es Scrumban que combina prácticas de Scrum y Kanban. En esta metodología se plantea a Kanban como una base sobre la cual uno puede, o no, colocar prácticas de Scrum según se considere necesario (Sanhueza, 2017).

Kanban, Scrum y Scrumban

En la Tabla 1 se compara estas tres metodologías de varios parámetros.

Tabla 1: Scrum, Kanban y Scrumban

Parámetro	Scrum	Kanban	Scrumban
Visualización de trabajo	Parcial	Total	Total
Backlog	Sprint backlog, Product Backlog	Backlog con límites	Backlog con límites

Límite del WIP	No limitado	Limitado	Limitado
Cambios	Debe esperar hasta el siguiente sprint	Según sea necesario	Según sea necesario
Roles	Si	No	Si
Estimaciones	Si	No	No
Iteraciones	Si	No	No
ScrumBoard	Se establece después de cada sprint	Tablero persistente	Depende de las decisiones de cada iteración

Fuente: (Gamboa, 2014)

CAPÍTULO III

METODOLOGIA

3.1. Tipo y Diseño de Investigación

Esta investigación se basa en un enfoque cualitativo, permitiendo así un análisis de la metodología Kanban, empleando el método analítico mediante el cual se establecen principios, reglas, característica y funcionamiento a través de la implementación de un tablero Kanban, siendo KanbanFlow el software utilizado para el control de las actividades en el desarrollo del proyecto. Se aplica el tipo de investigación documental bibliográfica, recopilando información de diferentes fuentes como: libros, artículos científicos, tesis doctorales, entre otras.

3.2. Técnicas de recolección de Datos

Para la recopilación de información se aplicó las técnicas de entrevista y encuestas.

3.2.1. Entrevista

Utilizada para generar contacto directo con los beneficiarios, estableciendo varias entrevistas tanto con el personal de la dirección académica como del Centro de Desarrollo de Sistemas Informático (CODESI) de la UNACH con la finalidad de conocer la situación actual del proceso de gestión de reactivos y posteriormente el levantamiento de requerimientos.

3.2.2. Encuesta

Mediante esta técnica se mide el grado de usabilidad del sistema en su fase de desarrollo y la optimización del proceso de gestión de reactivos.

3.3. Técnicas de Análisis e interpretación de la información

Las encuestas se realizaron mediante la herramienta Formularios de Google que además de generarlas de manera automatizada y óptima. Permite analizar cada pregunta, al recopilar en hojas de cálculo la información ordenada y automáticamente, incluido datos y gráficos en tiempo real (google, 2020).

3.4. Procedimiento

3.4.1. Análisis de requerimientos

La figura 4 muestra el proceso de creación y validación de un reactivo y la actividad que realiza cada usuario del sistema.

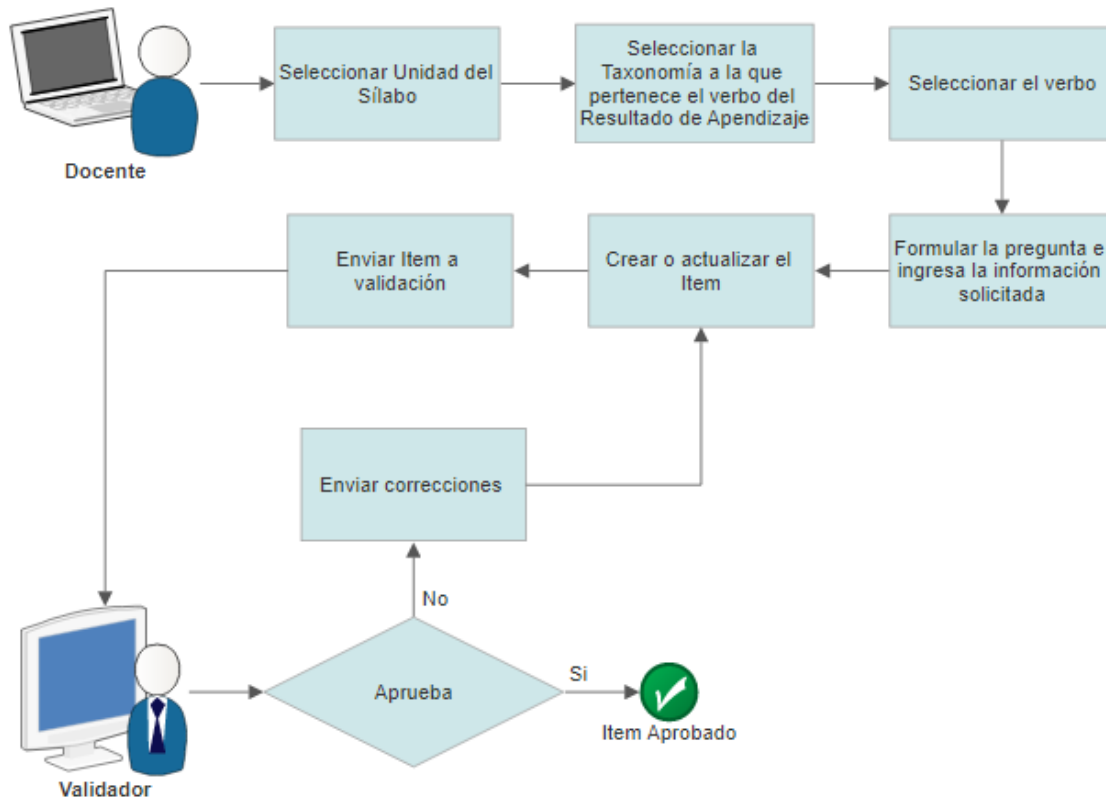


Figura 4: Flujo del procedimiento de Reactivos

- **Requisitos Funcionales**

Tabla 2: Requisitos Funcionales

Requisito	Descripción
Crear Preguntas	El sistema deberá permitir crear los ítems (preguntas): sus respectivos enunciados, respuestas de las cuales solo una debe ser correcta, bibliografía, palabra clave. Además de guardar y actualizar los datos que crea pertinente.

Enviar ítems a revisión	El sistema permitirá enviar a revisión los ítems creados a la unidad correspondiente para su respectiva validación siendo de forma y fondo
Visualizar asignaturas	El sistema permitirá visualizar al docente las asignaturas correspondientes del periodo vigente según sea el caso de acuerdo con los distributivos
Reenviar los ítems para su validación	El sistema permitirá reenviar a validación el ítem con sus respectivas correcciones
Visualización de ítems para validación	La dirección de carrera visualizará todos los ítems enviados para su respectiva validación.
Validar de fondo y forma	El sistema permitirá calificar de acuerdo con los parámetros establecidos en el reglamento institucional.
Actualizar validación forma y fondo	El sistema permitirá la actualización de la validación en un caso se haya omitido algún parámetro.

- **Requisitos No Funcionales**

Tabla 3: Requisitos No funcionales

Requisitos	Descripción
Usabilidad	El sistema debe ser fácil de aprender, utilizar, entender y sea atractivo para el usuario
Funcionalidad	El sistema deberá responder a las peticiones solicitadas según sea el caso pertinente para mostrar los datos al momento de realizar una acción
Disponibilidad	El sistema debe estar disponible todos los días del año para los usuarios

3.4.2. Diagramación

Diagramas de casos de uso

Casos de uso usando la herramienta STARUML 3.1.1

- **Actores**

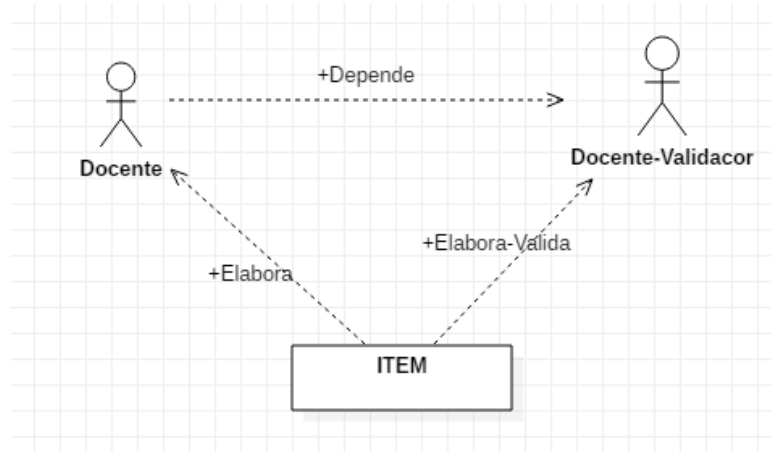


Figura 5: Caso de Uso Sistema Reactivos

Fuente: Los autores

- **Docente**

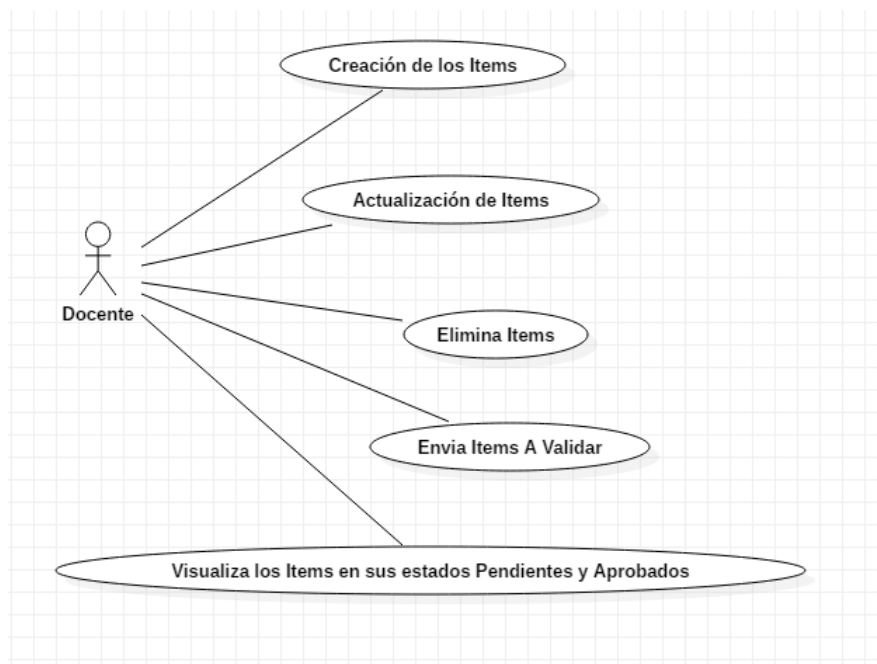


Figura 6: Caso de Uso Docente

Fuente: Los autores

- **Docente-Validador**

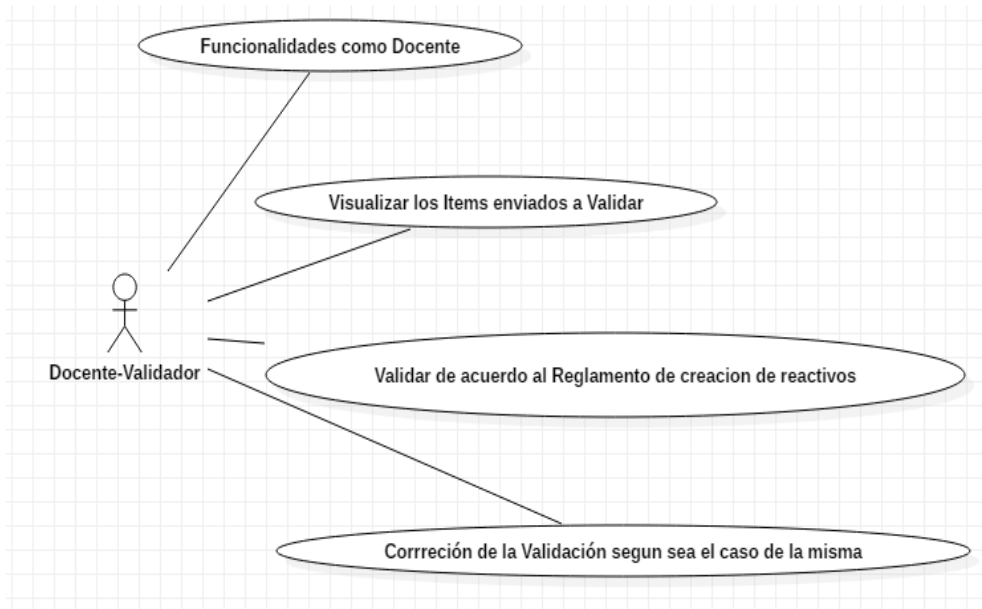


Figura 7: Caso de Uso Docente-Validador

Fuente: Los autores

3.4.3. Diseño de Arquitectura

- **Vista Conceptual**

El sistema de Reactivos desarrollado está compuesto por dos módulos Docente y Validación.

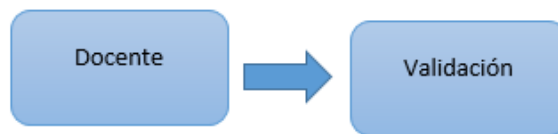


Figura 8: Vista Conceptual

- **Docente:** permite crear, actualizar y eliminar reactivos.
- **Validación:** realiza la validación de los ítems creados valorando tres relaciones con una puntuación de 1 a 5.

- **Vista Lógica**

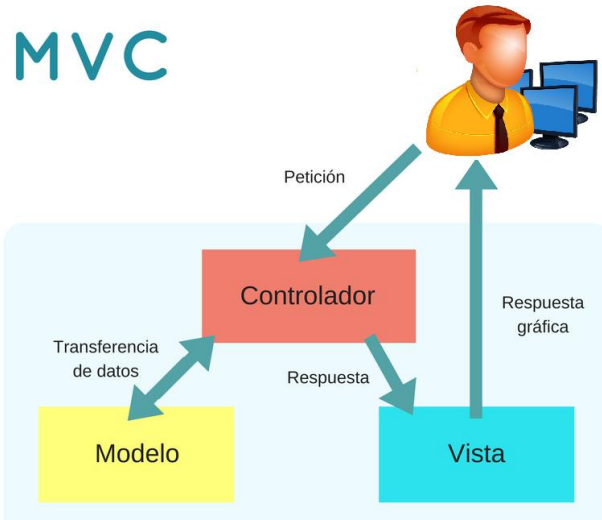


Figura 9: Vista Lógica

- **Vista Física**

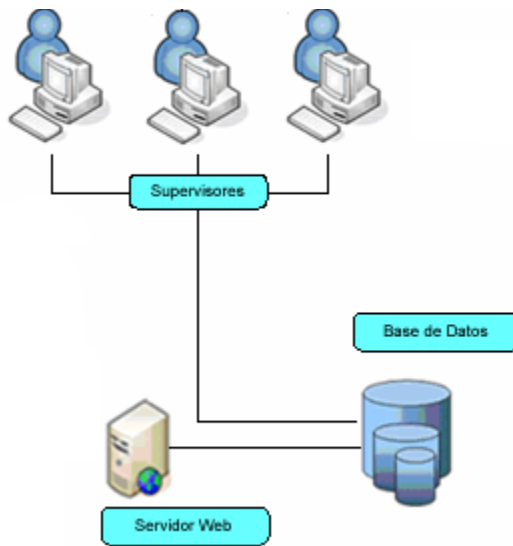


Figura 10: Vista Física

3.4.4. Modelado

- Diagrama de clases

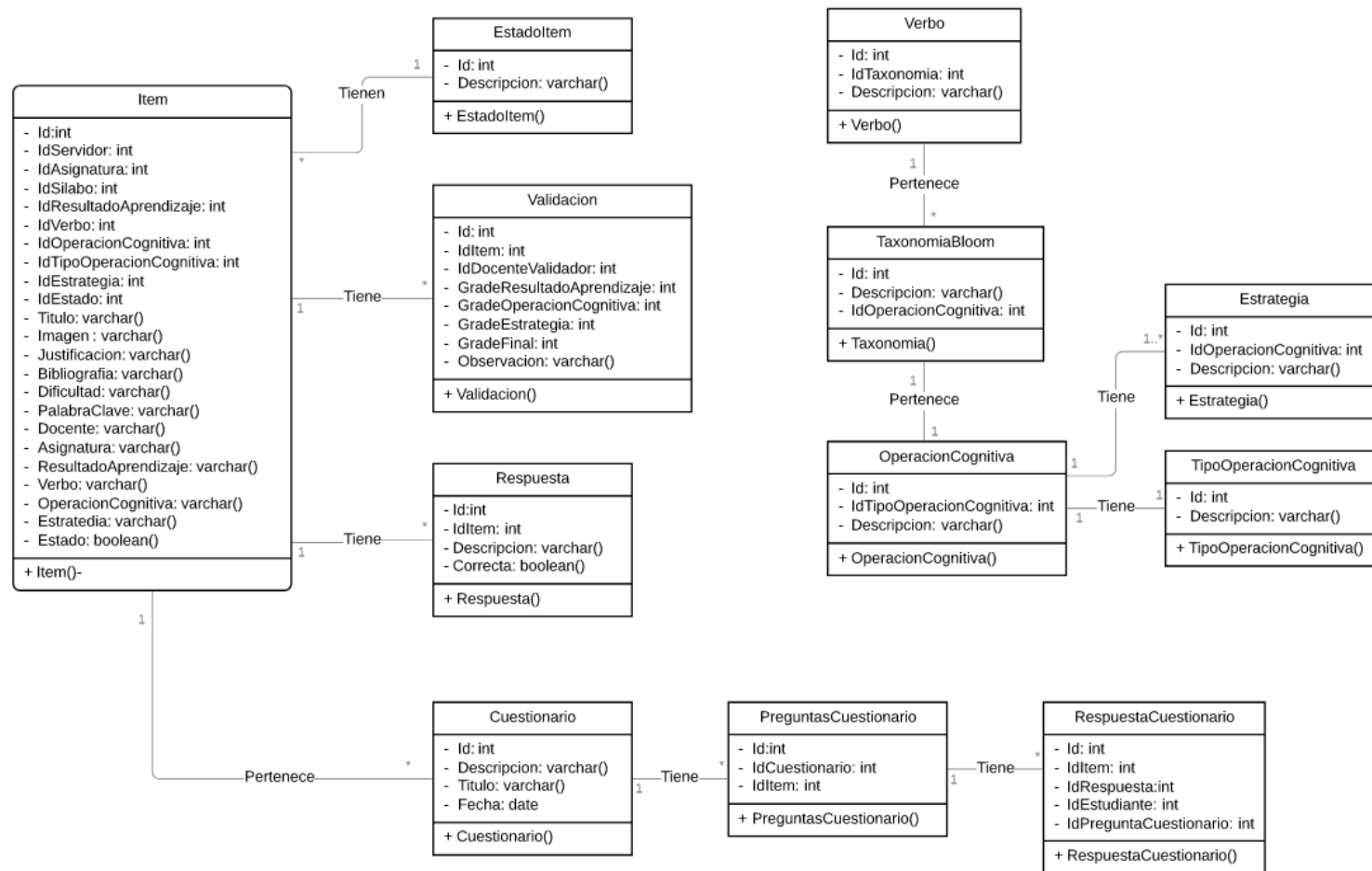


Figura 11: Diagrama de Clases

- Diagrama de Base de Datos

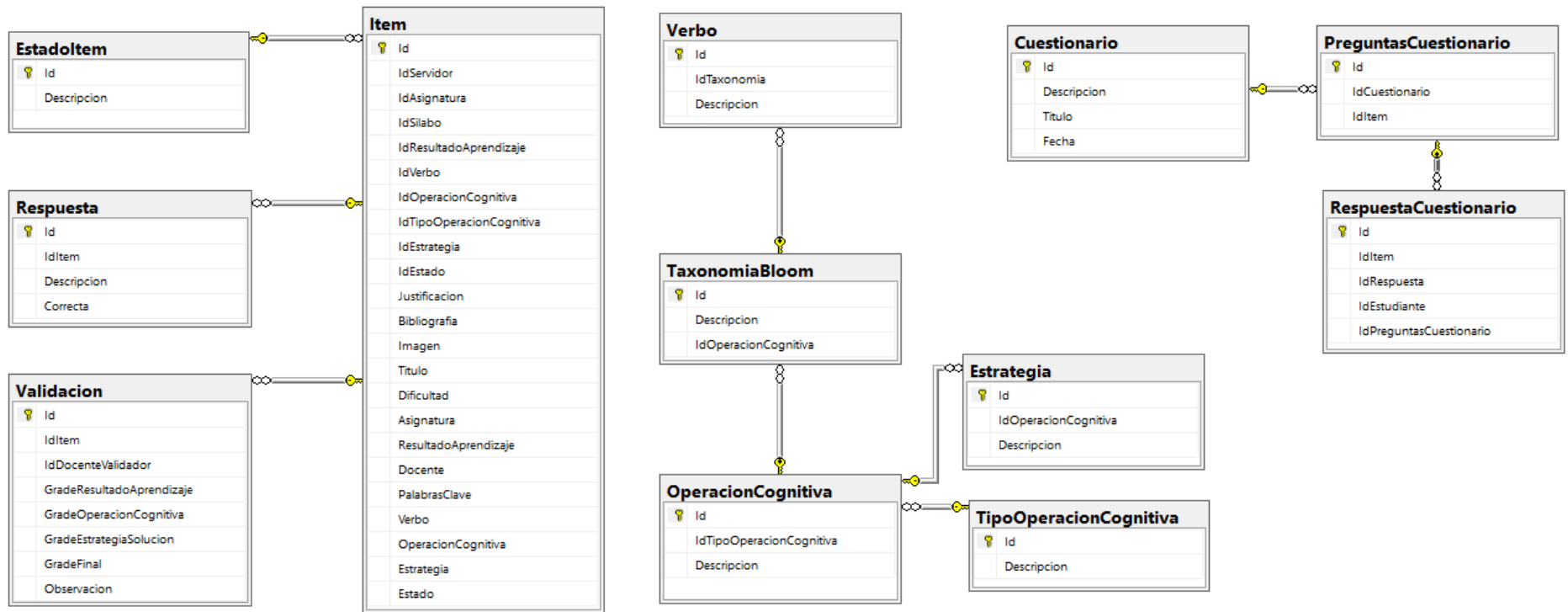


Figura 12: Diagrama de Base de Datos

3.4.5. Desarrollo del software aplicando KANBAN

- **Historias de usuarios**

Tabla 4: Historias de Usuarios

ID	Historia de Usuario	Descripción	Tamaño
HU01	Como Docente Quiero ver el listado de las Asignaturas Para crear los ítems de acuerdo con las asignaturas vigentes.	El listado de las asignaturas debe estar presente cuando el Docente se logue.	5
HU02	Como Docente Quiero ver el listado de los ítems creados para corregir y para enviar a revisión.	El listado de los ítems creados debe estar presente por asignaturas una vez que el Docente hay creado el ítem.	5
HU03	Como Docente Quiero ver el listado de los Ítems Aprobados Para tomarlos como referencia para nuevas creaciones de ítems.	El listado de los ítems aprobados debe estar presente en su vista correspondiente	3
HU04	Como Docente Quiero ver el listado del ítem que no han sido aprobados Para su posterior corrección y reenvió.	El listado de los ítems Pendientes debe estar presente en su vista correspondiente.	4
HU05	Como Docente-Validador Quiero Ver todos los ítems Enviados a Validar con sus respectivos estados para Corregirlos o Validarlos por primera vez.	El listado de Todos los ítems enviados a Validar debe estar presente en su vista correspondiente.	5
Hu06	Como Docente-Validador Quiero Ver Todo el contenido del Ítems enviado a Validar Para su correcta validación.	El Listado de los ítems enviado debe mostrar todo su contenido al momento de validar.	5

- **Descripción de las actividades (Product Backlog)**
Kanban se priorizan las actividades (sprint) por urgencia o necesidad.



Figura 13: Actividades / Fuente: Autores

Aplicación de Kanban

1. Flujo de trabajo o iteración



Figura 14: Flujo de trabajo / Fuente: Autores

2. Fases del Ciclo de desarrollo(producción)

- Backlog:** se detallan las actividades que formarán parte de los sprint las misma que serán clasificadas de acuerdo con su prioridad y necesidad siendo la de mayor necesidad la cual se inicia desarrollando.
- Actividades diarias (To Do):** el número de actividades que se muestran en esta etapa o iteración depende del tamaño del equipo (Team Developer) y de su experiencia en las actividades designadas, teniendo las misma un control no mayor al número de horas laboradas por día según sea el caso

establecido, cabe recalcar que esta fase son actividades muy puntuales que no tomen un tiempo prolongado después de hacer las respectivas pruebas y cumpliendo lo que se ha designado en la misma.

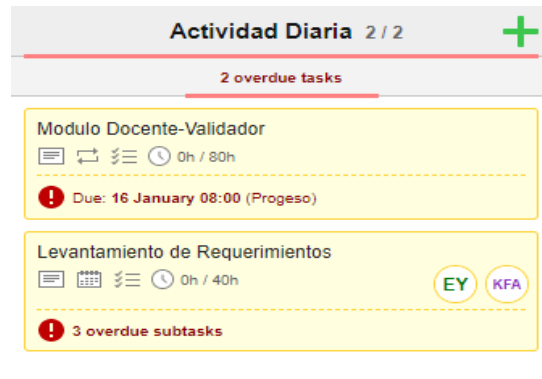


Figura 15: Actividades Diarias / Fuente: Autores

- c. **Progreso (Doing):** las actividades que de una u otra manera son terminadas en la etapa anterior (Actividades Diaria) al finalizar la jornada establecida una vez pasado por las respectivas pruebas y sin contratiempos para poder continuar su flujo al siguiente estado o etapa.

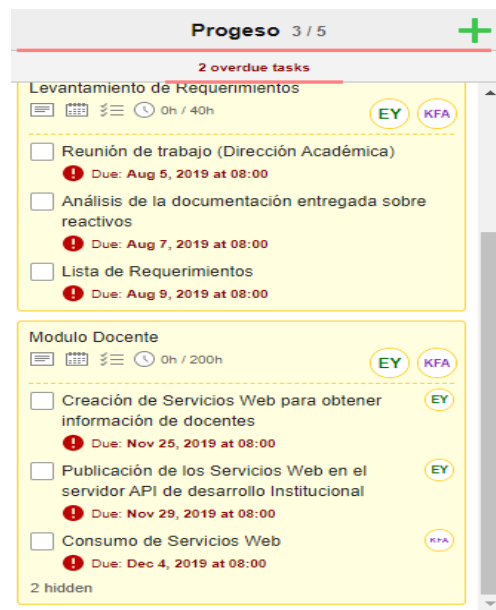


Figura 16: Actividades en Progreso / Fuente: Autores

- d. **Actividades Finalizadas (Done):** Las actividades que en sus estados anteriores cumplieron las reglas necesarias terminaran en la lista de actividades que se tomaran en cuenta para su posterior despliegue, estas

actividades cumplirán con todas las funcionalidades previstas en cada estado anterior.



Figura 17: Actividades Finalizada / Fuente: Autores

3. Deja de Empezar y Empieza a terminar (Stop Starting, start finishing)

Este principio es el que caracteriza a Kanban para mostrar un tablero actividades pendientes, no permite pasar de un estado o etapa a otro si el código no ha sido depurado correctamente y cumple la funcionalidad deseada.

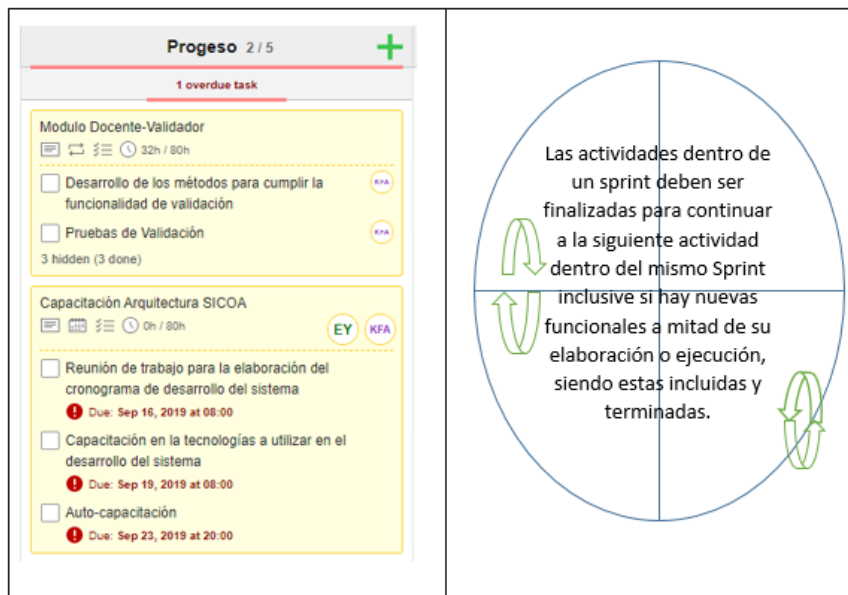


Figura 18: Iteración en el Progreso
Fuente: Autores

Tener un Control

Siendo Kanban una metodología netamente mente visual nos permite tener un control de nuestro progreso del proyecto, permitiendo así evitar la aglomeración de actividades pendientes o conocidos comúnmente como cuello de botella. Dentro del tablero Kanban de reactivos podemos observar que las actividades diarias hemos tenido actividades pendientes lo cual se dificulto en el progreso del proyecto en desarrollo.

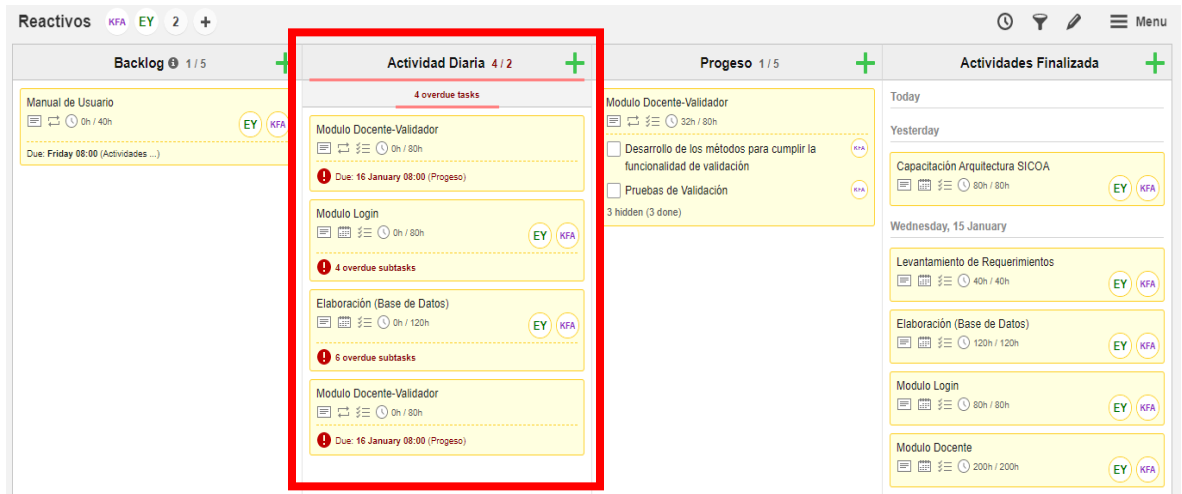


Figura 19: Control Entre Actividades / Fuente: Autores

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. Resultados

Mediante la aplicación de la metodología Kanban se obtuvo como resultado el sistema de gestión del proceso de reactivos REACSYS mismo que se encuentra en la primera fase de desarrollo (Anexo 2). Para evaluar la usabilidad del sistema en etapa de desarrollo se aplicó el método heurístico de Nielsen a 10 usuarios expertos en usabilidad, siendo representantes de la comisión revisora de reactivos institucional y de la carrera, además de los docentes colaboradores de este proyecto de investigación, obteniendo los siguientes resultados:

Pregunta 1: ¿La vista principal del Sitio refleja la identidad y pertenencia del Sistema de reactivos?

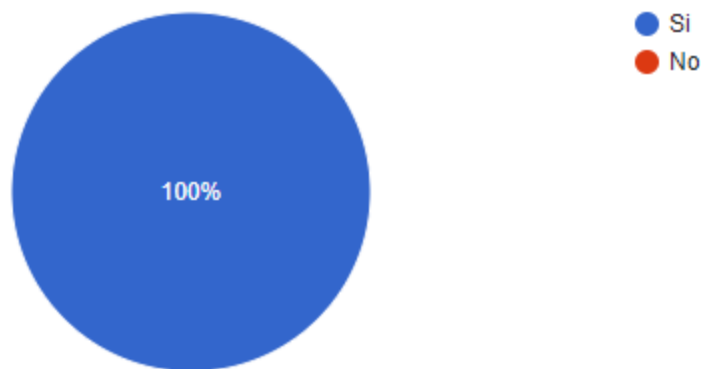


Figura 20: Pregunta 1

Interpretación: El resultado reflejado en la figura 1 nos indica que el sistema de reactivos tiene una identidad del 100% y los iconos utilizados están en relación con el sistema.

Pregunta 2: ¿El sistema le resultó fácil de usar?

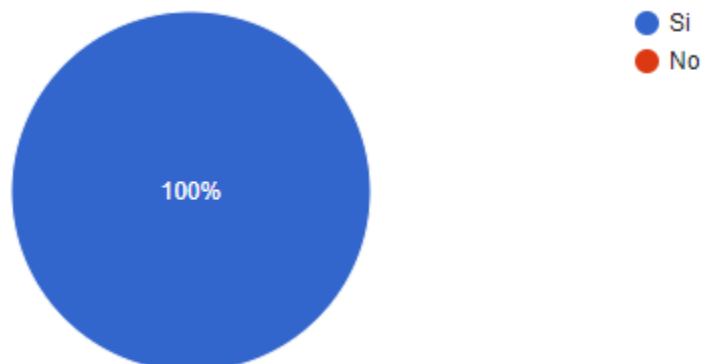


Figura 21: Pregunta 2

Interpretación: En la figura 2 refleja que al momento de interactuar con el sistema de reactivos no tuvieron complicación alguna en ninguno de los pasos previos en la creación del reactivo y en su respectiva validación.

Pregunta 3: ¿El diseño del Sitio es eficiente, rápido e intuitivo?

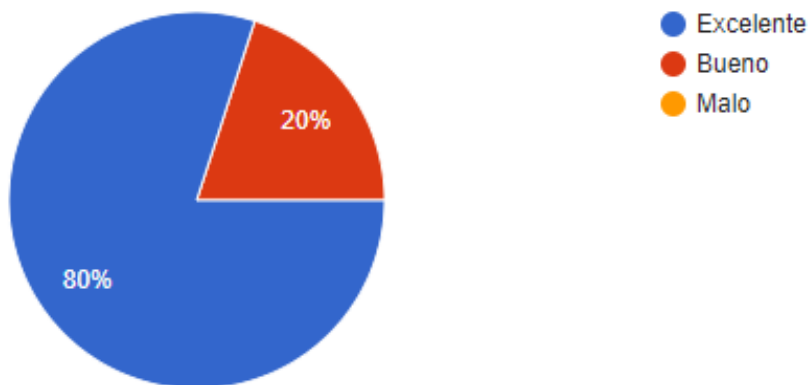


Figura 22: Pregunta 3

Interpretación: El resultado reflejado en la figura 3 nos muestra que un 80 % de los encuestados consideran que el sistema es rápido e intuitivo al momento de interactuar con el no siendo así que el 20% considera que es bueno este proceso.

Pregunta 4: ¿Cuál es el nivel de conformidad al momento de interactuar con el sistema?

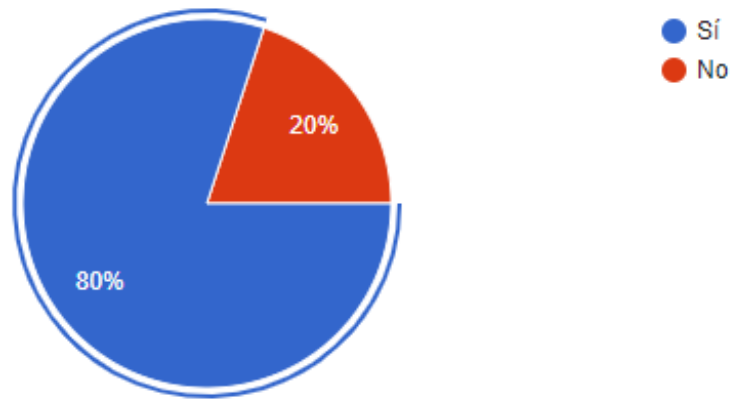


Figura 23: Pregunta 4

Interpretación: el resultado mostrado en la figura 4 nos indica que de un 100% un 80% se sintió conforme al interactuar con el sistema mientras que el otro 20% se rehusó al uso y prefiere el sistema usado actualmente denominado examview.

Pregunta 5: ¿Las funciones de este sistema están bien integradas?

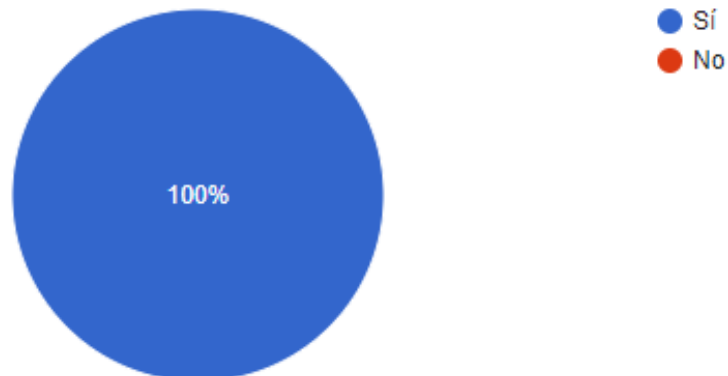


Figura 24: Pregunta 5

Interpretación: de acuerdo con los resultados que se muestran en la figura 5 el 100% considera que las funciones, con el proceso están integradas de una manera adecuada al sistema de reactivos.

Pregunta 6: ¿En que porcentaje considera usted que el uso correcto del sistema REACSYS reduce el tiempo de validación de reactivos?

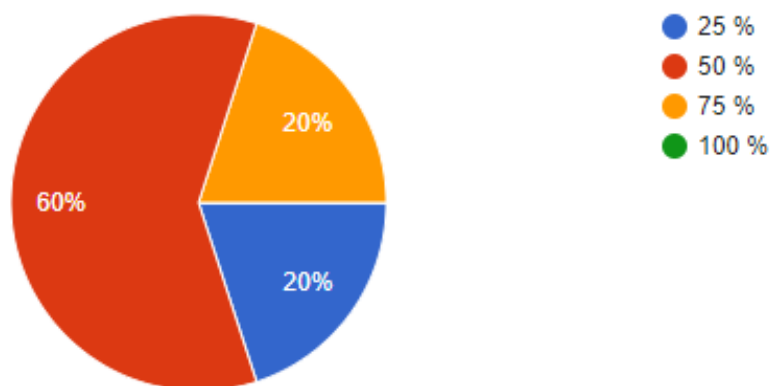


Figura 25: Pregunta 6

Interpretación: un 60% de los encuestado considera que el uso correcto del sistema reduce el tiempo de validación de un ítem en un 50% al proceso que anteriormente lo hacían mientras que un 20% considera que lo reduce en un 75% y el otro 20% Considera que únicamente lo reduce en un 25%.

Pregunta 7: ¿En que porcentaje considera usted que el uso correcto del sistema REACSYS reduce el tiempo de Creación de reactivos?

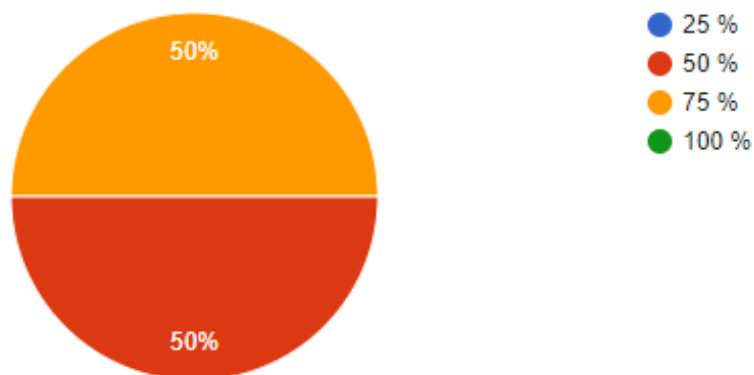


Figura 26: Pregunta 7

Interpretación: un 50% considera que el uso correcto del sistema reduce en un 75% el proceso de creación de un reactivo mientras que el otro 50% considera que solo lo hace en un 50%.

Pregunta 8: ¿Recomendaría usar este sistema web?

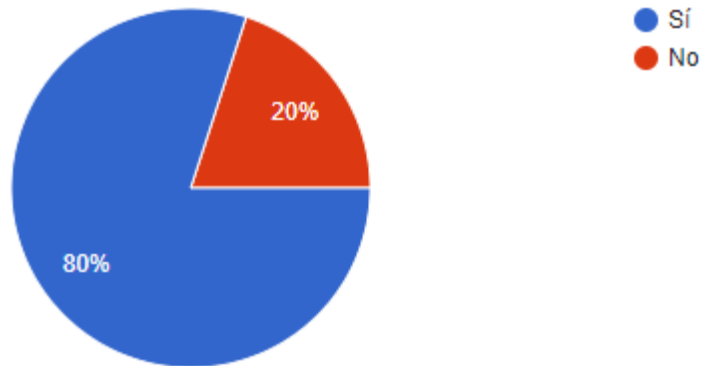


Figura 27: Pregunta 8

Interpretación: un 80% recomienda el uso de este sistema mientras que el 20% considera continuar con el sistema actual.

Pregunta 9: ¿En el sistema actual cuantos parámetros ingresa como información complementaria del reactivo?

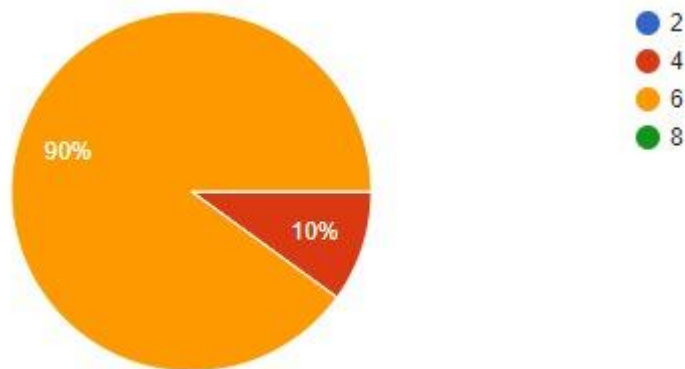


Figura 28: Pregunta 9

Interpretación: un 90% indican que los parámetros que ingresan como información complementaria del reactivo en el sistema actual son 6 mientras que el 10% indica que ingresan 4.

Pregunta 10: ¿Cuánto tiempo tarda en generar un reactivo con la utilización del sistema actual?

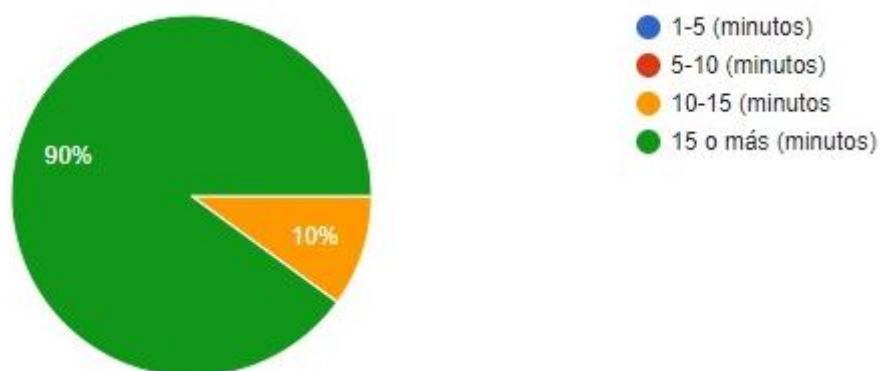


Figura 29: Pregunta 10

Interpretación: en un 90% el tiempo de creación de reactivo al utilizar el sistema actual está en un intervalo de tiempo de 15 o más minutos mientras que un 10% solo emplea un intervalo de 10 a 15 minutos al momento de elaborar un reactivo.

Pregunta 11: ¿Cuánto tiempo tarda en generar un reactivo con la utilización del sistema REACSYS?

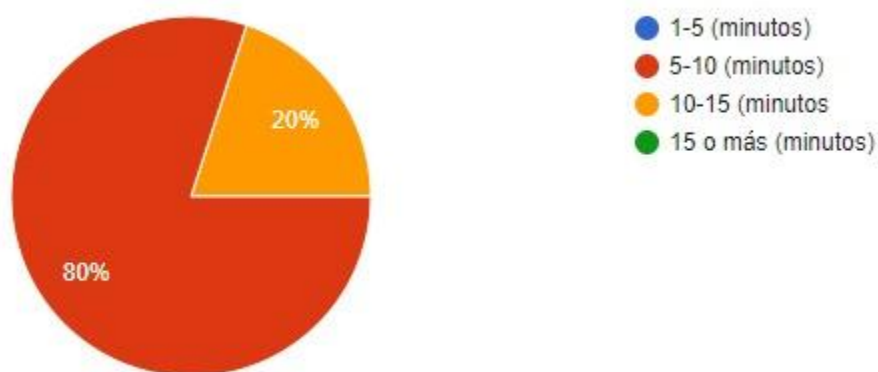


Figura 30: Pregunta 11

Interpretación: en un 80% el tiempo por creación de reactivo al utilizar el sistema REACSYS está en un intervalo de tiempo de 5-10 minutos.

Análisis final: La aplicación de encuestas a usuarios y expertos en usabilidad permitió determinar que los usuarios consideran que el sistema reduce tiempo al proceso de creación y validación en un 75 y 50 % respectivamente, además se determinó que el sistema el sistema de reactivos (REACSYS), tiene una buena acogida con los Docentes de la Universidad siendo recomendado en un 80% para implementación del mismo dentro de la Universidad Nacional de Chimborazo.

4.2. Discusión

Los resultados reflejan una total mejoría al proceso de Gestión de los reactivos que actualmente se lo realiza con otra herramienta y de manera manual. En la Tabla 5 y Figura 29 se evidencian los resultados obtenidos mediante un análisis Pre-Post utilizando la prueba estadística no paramétrica de chi cuadrado. Como resultado Pre se obtuvo que aproximadamente un docente tarda más de 15 minutos creando su reactivo en el software actual (Examview), mismo que requiere el ingreso de 6 parámetros como información completaría para crear el reactivo. Los resultados Post obtenidos fueron que al usar el sistema REACSYS desarrollado en este proyecto de investigación el tiempo aproximado de creación de un reactivo es de 5 a 10, mientras que los parámetros requeridos como información completaría son únicamente 2.

Tabla 5: Análisis Pre-Post

Parámetros	Pre	Post
Tiempo estimado para la creación de un reactivo (min)	15-20	5-10
Información complementaria del ítem (# de parámetros)	6	2

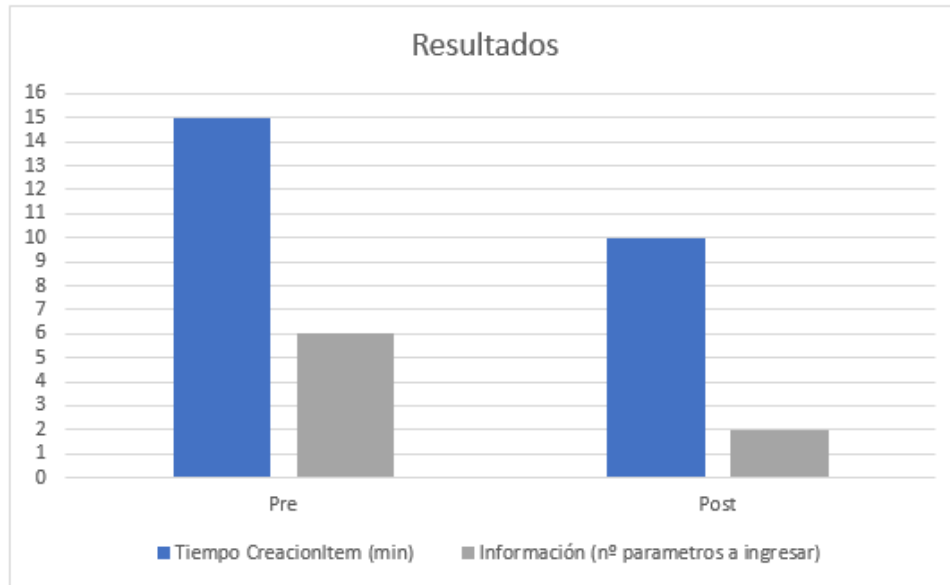


Figura 31: Resultado Pre-Post

CONCLUSIONES

- De acuerdo a la investigación realizada se concluye que la metodología ágil KANBAN permite agilizar los procesos y obtener un sistema de mejor calidad en poco tiempo, a diferencia del uso de metodologías tradicionales o inclusive ágiles. Su alto grado de flexibilidad y adaptabilidad al equipo de trabajo es lo que permite que en cada etapa que se encuentre una actividad no puede seguir su flujo si no ha sido terminada y validada sin ningún error.
- Se concluye que la metodología KANBAN se acopla a cualquier lenguaje de programación, en este caso para el desarrollo del sistema de reactivos se utilizó C# con ASP.NET CORE. Se trabajó con la arquitectura orientada al dominio definida por la Universidad Nacional de Chimborazo. Se levantaron los requisitos funcionales y no funcionales mediante reuniones de trabajos establecidas con las autoridades solicitantes del sistema.
- Se desarrolló satisfactoriamente el software para la generación, validación y actualización de reactivos, mediante la aplicación de la metodología Kanban que es una de las metodologías más ligeras o menos restrictivas y adaptable para cualquier tipo de proyecto ya sea este llevado a cabo por un gran número de personas o un grupo pequeño, considerando siempre que exista contacto con el cliente constantemente.

- Mediante las encuestas realizadas sobre el uso del sistema REACSYS a usuario y expertos en usabilidad se concluye que la integración de las funcionalidades es adecuada, que el sistema es fácil de usar y que reduciendo el tiempo tanto de creación como de validación de un reactivo se logra la completa satisfacción del usuario, mismos que recomiendan su uso y posterior implementación e integración con los diferentes sistemas de la institución.

RECOMENDACIONES

- Recomendamos que una vez que el sistema se encuentre en producción, éste debe actualizarse constantemente aplicando la misma metodología Kanban a los nuevos requerimientos del proceso, para mantener el sistema usable, funcional y en un corto tiempo para que así puedan los docentes de la institución utilizar de una adecuada manera.
- Recomendamos que antes de empezar a utilizar una metodología para el control del flujo de trabajo o híbrida se debe considerar si se tiene los conocimientos necesarios y los insumos correctos.
- Recomendamos también que este proyecto continúe su desarrollo hasta su implementación para lograr los objetivos para lo que fue creado, beneficiando considerablemente a estudiante, docente y autoridades.

REFERENCIAS BIBLIOGRÁFICAS

- Alvear, A., & Quintero, G. (2015). Integrating software development techniques, usability, and agile methodologies. *Desarrollo de software integrando técnicas de usabilidad y metodologías ágiles. Actas de Ingeniería, Universidad del Turabo – Puerto Rico*, 1(94-103), 10.
- Arango, S. M., Campuzano, Z. L., & Zapata, C. J. (2015). Mejoramiento de procesos de manufactura utilizando Kanban. *Ingenierías Universidad de Medellín*, 221-233.
- BBVAOPEN4U. (13 de 04 de 2018). *Qué es el método Kanban y por qué funciona en la programación de software*. Obtenido de <https://bbvaopen4u.com/es/actualidad/que-es-el-metodo-kanban-y-por-que-funciona-en-la-programacion-de-software>
- Bermejo, M. (2011). Kanban. *Universitat Oberta de Catalunya*, 36.
- Canós, J. H., Letelier, P., & Penadés, M. C. (2012). Metodologías Ágiles en el Desarrollo de Software. *DSIC -Universidad Politécnica de Valencia*, 8.
- Córdova, P. A. (2012). *Aplicación de kanban en el desarrollo de un sistema web para administración del consultorio odontológico fitoc dental*. Quito: Pontificia Universidad Católica de Ecuador.
- crisp, C. (2010). *KANBAN*. Obtenido de <https://www.crisp.se/gratis-material-och-guider/kanban>
- Dimes, T. (2015). *Conceptos Básicos de Scrum: Desarrollo de software Agile y manejo de proyectos Agile*. Babelcube Inc.
- Estrada, J. A. (2006). *Sistema KANBAN, como una ventaja competitiva en la micro, pequeña y mediana empresa*. Obtenido de Universidad Autónoma del Estado de Hidalgo : https://s3.amazonaws.com/academia.edu.documents/47390290/Sistema_KANBAN.desbloqueado.pdf?response-content-disposition=inline%3B%20filename%3DKanban.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20200131%2Fus-east-1%2Fs3%2Faw
- Figueroa, R. G., Solís, C. J., & Cabrera, A. A. (2008). Metodologías tradicionales vs. metodologías ágiles. *Universidad Técnica Particular de Loja, Escuela de Ciencias de la Computación.*, 9.
- Gamboa, M. J. (2014). Aumento de la productividad en la gestión de proyectos, utilizando una metodología ágil aplicada en una fábrica de software en la ciudad de Guayaquil. *Tecnológica-ESPOL*, 1-36.

- Garzas, J. (2012). *javiergarzas.com*. Obtenido de Diez herramientas Kanban: <http://www.javiergarzas.com/2012/01/herramientas-kanban.html>
- google. (2020). *google*. Obtenido de <https://www.google.com/intl/es-419/forms/about/>
- Guzmán, D. I., Islas, U. C., Corona, C. P., & Méndez, B. E. (2014). Metodología ágil Scrumban en el proceso de desarrollo y mantenimiento de software de la norma MoProSoft. *Research in Computing Science*, 97-107.
- Kleen, B. L. (2003). *Evaluacion del aprendizaje, alternativas y desarrollo*. Obtenido de [https://moodle.asignaturas.usb.ve/pluginfile.php/24391/mod_resource/content/1/Evaluacion%20del%20aprendizaje.%20Alternativas%20y%20nuevos%20desarrollos\(1\).pdf](https://moodle.asignaturas.usb.ve/pluginfile.php/24391/mod_resource/content/1/Evaluacion%20del%20aprendizaje.%20Alternativas%20y%20nuevos%20desarrollos(1).pdf)
- Kniberg, H., Skarin, M., Poppendieck, M., & Anderson, D. (2010). Kanban y Scrum—obteniendo lo mejor de ambos. *C4Media Inc*.
- Letelier, P., & Penadés, M. C. (2012). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *Universidad Politécnica de Valencia* , 17.
- Navarro, M. M., Moreno, M. M., Aranda, L. J., Parra, L. L., Rueda, L. J., & Pantano, J. C. (2017). Selección de Metodologías Ágiles e Integración de Arquitecturas de Software en el Desarrollo de Sistemas de Información. *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires)*., 5.
- Omaña, M., & Cardenas, J. T. (2011). Herramientas de manufactura esbelta aplicadas al desarrollo de software con calidad. *Avances en Sistemas e Informática*, 8.
- ORDYSINSKI, T. (2013). Kanban based information management in organization. *Polish Association for knowledge Management, Series: Studies and Proceedings*, 76-85.
- Palao Villasante, A. S. (2014). Modelo de Sistema de Información de registro y monitoreo socio ambiental participativo del proyecto de exploracion minero Chucapaca comparando las Metodologias Ágiles Scrum y Kanban. *Universidad Nacional del Altiplano*.
- Pérez, P. M. (2012). *Guía comparativa de Metodologias Agiles*. Universidad de Valladolid.
- Poppendieck, M., & Poppendieck, T. (20003). *Lean Software Development: An Agile Toolkit*. Addison-Wesley.
- Sanhueza, G. J. (2017). Metodología Lean-PMI para desarrollo de proyectos de software. *Doctoral dissertation, Universidad de Concepción. Facultad de Ingeniería. Departamento de Ingeniería Informática y Ciencias de la Computación*.
- Scott, R. (11 de Octubre de 2001). *InformIT*. Obtenido de <http://www.informit.com/articles/article.aspx?p=167902>

- Shore, L. (2010). *Kanban Tool*. Obtenido de ¿Por qué utilizar la metodología Kanban?: <https://kanbantool.com/es/metodologia-kanban>
- Torres, R. R. (2011). *UNA HERRAMIENTA EXPRESIVA PARA IMPLEMENTACIÓN DE TABLEROS*. Chile: UNIVERSIDAD DE CHILE.
- Uribe, E. H. (2007). *DEL MANIFIESTO ÁGIL SUS VALORES Y PRINCIPIOS* . Obtenido de <https://www.redalyc.org/pdf/849/84934064.pdf>

ANEXOS

- Anexo 1

Encuesta

Cuestionario para medir la Usabilidad en
etapa de desarrollo del Sistema Web
para Generación de Reactivos
Unach(REACSYS)

*Obligatorio

¿La vista principal del Sitio refleja la identidad y pertenencia del Sistema de reactivos? *

- Sí
- No

¿El sistema le resultó fácil de usar? *

- Sí
- No

¿El diseño del Sitio es eficiente, rápido e intuitivo? *

- Sí
- No

¿Recomendaría usar este sistema web? *

- Sí
- No

¿En el sistema actual cuantos parámetros ingresa como información complementaria del reactivo? *

- 2
- 4
- 6
- 8

¿Cuanto tiempo tarda en generar un reactivos con la utilización del sistema actual? *

- 1-5 (minutos)
- 5-10 (minutos)
- 10-15 (minutos)
- 15 o más (minutos)

¿Cuanto tiempo tarda en generar un reactivos con la utilización del sistema Reacsys? *

- 1-5 (minutos)
- 5-10 (minutos)
- 10-15 (minutos)
- 15 o más (minutos)

¿Cuál es el nivel de conformidad al momento de interactuar con el sistema? *

- Excelente
- Bueno
- Malo

¿Las funciones de este sistema están bien integradas? *

- Sí
- No

¿En que porcentaje considera usted que el uso correcto del sistema Reacsys reduce el tiempo de validación de reactivos? *

- 25 %
- 50 %
- 75 %
- 100 %

¿En que porcentaje considera usted que el uso correcto del sistema Reacsys reduce el tiempo de Creación de reactivos? *

- 25 %
- 50 %
- 75 %
- 100 %

- **Anexo 2**

Certificado de desarrollo



Dirección Académica
VICERRECTORADO ACADÉMICO

en movimiento

CERTIFICACIÓN

A petición de la parte interesada; yo, Ing. Danny Velasco Silva, **DIRECTOR ACADÉMICO**, certifico:

Que, el(la) **Sr(a). KLÉBER FABIÁN ARMIJOS GUILLÉN**, portador(a) de la cédula de ciudadanía No. 2300203912, egresado de la Universidad Nacional de Chimborazo de la Facultad de Ingeniería de la Carrera de Sistemas y Computación, **entregó el Sistema de Gestión de Reactivos**, cumpliendo con la primera fase, como parte de su Tesis de Grado "APLICACIÓN DE LA METODOLOGÍA KANBAN EN EL DESARROLLO DEL SOFTWARE PARA GENERACIÓN, VALIDACIÓN Y ACTUALIZACIÓN DE REACTIVOS, INTEGRADO AL SISTEMA INFORMÁTICO DE GESTIÓN ACADÉMICA UNACH"

Es todo cuanto puedo certificar en honor a la verdad, facultando al interesado(a) hacer uso del presente, para los fines que crea conveniente.

Riobamba, 30 de enero de 2020


Ing. Danny Velasco Silva
DIRECTOR ACADÉMICO



jcrd



Dirección Académica
VICERRECTORADO ACADÉMICO

en movimiento

CERTIFICACIÓN

A petición de la parte interesada; yo, Ing. Danny Velasco Silva, **DIRECTOR ACADÉMICO**, certifico:

Que, el(la) **Sr(a). ERIKA DAYANA YÉPEZ LLERENA**, portador(a) de la cédula de ciudadanía No. 0605136878, egresada de la Universidad Nacional de Chimborazo de la Facultad de Ingeniería de la Carrera de Sistemas y Computación, **entregó el Sistema de Gestión de Reactivos**, cumpliendo con la primera fase, como parte de su Tesis de Grado "APLICACIÓN DE LA METODOLOGÍA KANBAN EN EL DESARROLLO DEL SOFTWARE PARA GENERACIÓN, VALIDACIÓN Y ACTUALIZACIÓN DE REACTIVOS, INTEGRADO AL SISTEMA INFORMÁTICO DE GESTIÓN ACADÉMICA UNACH"

Es todo cuanto puedo certificar en honor a la verdad, facultando al interesado(a) hacer uso del presente, para los fines que crea conveniente.

Riobamba, 30 de enero de 2020


Ing. Danny Velasco Silva
DIRECTOR ACADÉMICO



jcrd

Campus Norte

Av. Antonio José de Sucre, Km 1 1/2 vía a Guano

Teléfonos (593-3) 3730880 - Ext: 1255

- **Anexo 3**

Manual de Usuario

SISTEMA DE REACTIVOS

Elaborado por: Erika Dayana Yepéz Lleran
Kleber fabian Armijos Guillen

Febrero 2020