



UNIVERSIDAD HISPANOAMERICANA

**PROPUESTA DE DESARROLLO DE SISTEMA ACADÉMICO
EN KAMUK SCHOOL BASADO EN SCRUM**

**Sustentante:
Mario Gerardo Angulo Cordero**

**Tesis para optar por el grado de
Licenciatura en Ingeniería Informática**

**Tutora:
Yusselin Murcia Céspedes**

**Universidad Hispanoamericana
Facultad de Ingeniería Informática
San José, Tibás**

Julio, 2019

Carta de aprobación del tutor

Heredia, 31 de julio de 2019

Sra. María Isabel Losilla
Ingeniería Informática
Universidad Hispanoamericana

Estimada señora:

El estudiante Mario Gerardo Angulo Cordero, cédula de identidad número 1-1527-0667, me ha presentado, para efectos de revisión y aprobación, el trabajo de investigación denominado **PROPUESTA DE DESARROLLO DE SISTEMA ACADÉMICO EN KAMUK SCHOOL BASADO EN SCRUM**, el cual ha elaborado para optar por el grado académico de Licenciatura.

En mi calidad de tutor, he verificado que se han hecho las correcciones indicadas durante el proceso de tutoría y he evaluado los aspectos relativos a la elaboración del problema, objetivos, justificación; antecedentes, marco teórico, marco metodológico, tabulación, análisis de datos; conclusiones y recomendaciones.

De los resultados obtenidos por el postulante, se obtiene la siguiente calificación:

a)	ORIGINAL DEL TEMA	10%	10
b)	CUMPLIMIENTO DE ENTREGA DE AVANCES	20%	19
c)	COHERENCIA ENTRE LOS OBJETIVOS, LOS INSTRUMENTOS APLICADOS Y LOS RESULTADOS DE LA INVESTIGACION	30%	29
d)	RELEVANCIA DE LAS CONCLUSIONES Y RECOMENDACIONES	20%	19
e)	CALIDAD, DETALLE DEL MARCO TEORICO	20%	20
	TOTAL		97

En virtud de la calificación obtenida, se avala el traslado al proceso de lectura.

Atentamente,

YUSSELIN
 TATIANA
 MURCIA
 CESPEDES
YUSSELIN MURCIA CÉSPEDES
Cédula identidad N 205780828
Carné Colegio Profesional N 9020

Firmado digitalmente
 por YUSSELIN
 TATIANA MURCIA
 CESPEDES
 Fecha: 2019.07.31
 19:28:06 -05'00'

Declaración jurada

Yo Mario Gerardo Angulo Cordero, mayor de edad, portador de la cédula de identidad número 1-1527-0667 egresado de la carrera de Ingeniería Informática de la Universidad Hispanoamericana, hago constar por medio de este acto y debidamente apercibido y entendido de las penas y consecuencias con las que se castiga en el Código Penal el delito de perjurio, ante quienes se constituyen en el Tribunal Examinador de mi trabajo de tesis para optar por el título de Licenciatura en Sistemas de Información, juro solemnemente que mi trabajo de investigación titulado: Propuesta de desarrollo de sistema académico en Kamuk School basado en Scrum, es una obra original que ha respetado todo lo preceptuado por las Leyes Penales, así como la Ley de Derecho de Autor y Derecho Conexos número 6683 del 14 de octubre de 1982 y sus reformas, publicada en la Gaceta número 226 del 25 de noviembre de 1982; incluyendo el numeral 70 de dicha ley que advierte; artículo 70. Es permitido citar a un autor, transcribiendo los pasajes pertinentes siempre que éstos no sean tantos y seguidos, que puedan considerarse como una producción simulada y sustancial, que redunde en perjuicio del autor de la obra original. Asimismo, quedo advertido que la Universidad se reserva el derecho de protocolizar este documento ante Notario Público. en fe de lo anterior, firmo en la ciudad de San José, a los 31 días del mes de julio del año dos mil diecinueve.



Firma del estudiante
Cédula: 1-1527-0667

Carta de aprobación del lector

Universidad Hispanoamericana
Sede Llorente
Escuela de Ingeniería Informática

Estimados señores

El estudiante Mario Gerardo Angulo Cordero, cédula de identidad 1-1527-0667, me ha presentado para efectos de revisión y aprobación, el trabajo de investigación denominado **"PROPUESTA DE DESARROLLO DE SISTEMA ACADÉMICO EN KAMUK SCHOOL BASADO EN SCRUM"**, el cual ha elaborado para obtener su grado de Licenciatura en Ingeniería Informática.

He revisado y he hecho las observaciones relativas al contenido analizado, particularmente lo relativo a la coherencia entre el marco teórico y análisis de datos, la consistencia de los datos recopilados y la coherencia entre éstos y las conclusiones; asimismo, la aplicabilidad y originalidad de las recomendaciones, en términos de aporte de la investigación.

Por consiguiente, este trabajo cuenta con mi aval para ser presentado en la defensa pública.

Atte.

ALEJANDRO
BOGANTES
SALAZAR
(FIRMA)

Digitally signed
by ALEJANDRO
BOGANTES
SALAZAR
(FIRMA)
Date: 2019.09.11
11:05:46 -06'00'

Firma:

MSc. Alejandro Bogantes Salazar
Cédula: 303940389

Carta de revisión de filóloga



COMUNICACIÓN
COMPLETA

San José, 26 de setiembre de 2019

Señores
Universidad Hispanoamericana
Facultad de Ingeniería
Escuela de Ingeniería
S.O.

NUESTROS SERVICIOS

Análisis de contenido, corrección de estilo, revisión crítica discursiva, textos políticos, publicitarios, gráficos, académicos y otros, diseño gráfico, asesoría periodística y manejo de redes sociales.

DIRECCIÓN

San Francisco de Dos Ríos,
San José, Costa Rica.

TÉLFONOS

(506) 2226-2213
(506) 8828-3285

CORREO

prozo57.nela@gmail.com

APDO

719-2350, San José
Costa Rica.

Estimados señores:

A quien corresponda:

Me sirvo de la misiva para hacer constar que el Trabajo Final de Graduación (TFG), titulado *Propuesta de Desarrollo de Sistema Académico en Kamuk School Basado en Scrum*, cuyo postulante es el Sr. Mario Angulo, cédula 1-1527-0667, ha sido objeto de la revisión y corrección filológica que corresponde. La valoración se orientó en tres direcciones principales, a saber:

- Revisión general.
- Verificación del ordenamiento de los acápites.
- Valoración de los aspectos generales de formato.

En este sentido, se ha considerado la conformidad de la escritura del texto con la sexta edición de la normativa APA, para lo cual fueron utilizados dos manuales de estilo complementarios entre sí: el *Estilo de la APA*, Universidad de Costa Rica, 2016; y el *Manual de estándares de publicación. Módulo 2: Tipografía y Ortotipografía*, Universidad Estatal a Distancia de Costa Rica (UNED) 2012.

La revisión y corrección se realizó en dos categorías principales e incluyó los siguientes aspectos:

Lingüística textual:

- Cohesión y coherencia.
- Precisión léxica
- Puntuación
- Ortografía
- Cacofonías
- Ordenamiento de los constituyentes textuales: oración y párrafo.

Aspectos de formato:

- Uniformidad de márgenes (2,5 simétricos / todo el documento).
- Uniformidad de letra (arial 12) para el cuerpo del texto.
- Uniformidad del espaciado entre párrafos.
- Ordenamiento de los constituyentes estructurales: acápites y subacápites.
- Edición de tablas y figuras.
- Edición de anexos.

Sin otro particular, suscribe



Marianela Orozco Orozco
Filóloga
Universidad de Costa Rica
Cód. 281

Carta de autorización de autores para la consulta

ANEXO 1 (Versión en línea dentro del Repositorio) LICENCIA Y AUTORIZACIÓN DE LOS AUTORES PARA PUBLICAR Y PERMITIR LA CONSULTA Y USO

Parte 1. Términos de la licencia general para publicación de obras en el repositorio institucional

Como titular del derecho de autor, confiero al Centro de Información Tecnológico (CENIT) una licencia no exclusiva, limitada y gratuita sobre la obra que se integrará en el Repositorio Institucional, que se ajusta a las siguientes características:

- a) Estará vigente a partir de la fecha de inclusión en el repositorio, el autor podrá dar por terminada la licencia solicitándolo a la Universidad por escrito.
- b) Autoriza al Centro de Información Tecnológico (CENIT) a publicar la obra en digital, los usuarios puedan consultar el contenido de su Trabajo Final de Graduación en la página Web de la Biblioteca Digital de la Universidad Hispanoamericana
- c) Los autores aceptan que la autorización se hace a título gratuito, por lo tanto, renuncian a recibir beneficio alguno por la publicación, distribución, comunicación pública y cualquier otro uso que se haga en los términos de la presente licencia y de la licencia de uso con que se publica.
- d) Los autores manifiestan que se trata de una obra original sobre la que tienen los derechos que autorizan y que son ellos quienes asumen total responsabilidad por el contenido de su obra ante el Centro de Información Tecnológico (CENIT) y ante terceros. En todo caso el Centro de Información Tecnológico (CENIT) se compromete a indicar siempre la autoría incluyendo el nombre del autor y la fecha de publicación.
- e) Autorizo al Centro de Información Tecnológica (CENIT) para incluir la obra en los índices y buscadores que estimen necesarios para promover su difusión.
- f) Acepto que el Centro de Información Tecnológico (CENIT) pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.
- g) Autorizo que la obra sea puesta a disposición de la comunidad universitaria en los términos autorizados en los literales anteriores bajo los límites definidos por la universidad en las "Condiciones de uso de estricto cumplimiento" de los recursos publicados en Repositorio Institucional.

SI EL DOCUMENTO SE BASA EN UN TRABAJO QUE HA SIDO PATROCINADO O APOYADO POR UNA AGENCIA O UNA ORGANIZACIÓN, CON EXCEPCIÓN DEL CENTRO DE INFORMACIÓN TECNOLÓGICO (CENIT), EL AUTOR GARANTIZA QUE SE HA CUMPLIDO CON LOS DERECHOS Y OBLIGACIONES REQUERIDOS POR EL RESPECTIVO CONTRATO O ACUERDO.

**UNIVERSIDAD HISPANOAMERICANA
CENTRO DE INFORMACION TECNOLÓGICO (CENIT)
CARTA DE AUTORIZACIÓN DE LOS AUTORES PARA LA CONSULTA, LA
REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA
DE LOS TRABAJOS FINALES DE GRADUACIÓN**

San José, 28 de setiembre de 2019

Señores:
Universidad Hispanoamericana
Centro de Información Tecnológico (CENIT)

Estimados Señores:

El suscrito (a) Mario Gerardo Aguila Cordero con número de
identificación 1-1527-0667 autor (a) del trabajo de graduación titulado
Propuesta de Desarrollo de Sistema Académico en Moodle School
Basado en Scrum

presentado y aprobado en el año 2019 como requisito para optar por el título
de Licenciatura en Ingeniería Informática; (SI / NO)
autorizo al Centro de Información Tecnológico (CENIT) para que con fines
académicos, muestre a la comunidad universitaria la producción intelectual
contenida en este documento.

De conformidad con lo establecido en la Ley sobre Derechos de Autor y Derechos
Conexos N° 6683, Asamblea Legislativa de la República de Costa Rica.

Cordialmente,


1-1527-0667
Firma y Documento de Identidad

Agradecimiento

Agradezco, inicialmente, a Dios por la oportunidad de desarrollar un proyecto que me permitió aplicar los conocimientos adquiridos en la universidad y lugares de trabajo. Segundo, agradezco a mis compañeros José Alejandro Chavarría Benavides, Diego Saravia Valverde, Daniel Valle Jiménez, Álvaro Roberth Herrera Castro y Luis Iván Duarte Solís por sus valiosos aportes, sin los cuales el éxito de este proyecto no hubiese sido posible.

También agradezco en gran manera a Kamuk School y, especialmente, a su director Rommel Porras por darme la oportunidad de desarrollar el proyecto en la misma institución donde curse la primaria y secundaria, tiempo durante el cual él me apoyo académicamente de manera incondicional. También, agradezco a mi madre, Irmína Cordero Villagra; a mi padre Mario Angulo Gutiérrez y a mi madrina Amalia Cordero Villagra, quienes han sido mi motor y la razón de mi esfuerzo en todo momento para cumplir esta meta.

Finalmente agradezco a mi tutora Yusselin Murcia Céspedes por su apoyo durante este proceso; y a la Universidad Hispanoamericana por brindarme las herramientas necesarias para mi desarrollo profesional. En síntesis, deseo expresar mi más sincero agradecimiento a todos por colaborar conmigo durante el proceso de adquisición del conocimiento que hoy me convierte en un profesional capaz de aportar a la sociedad.

Índice de contenidos

Carta de aprobación del tutor	ii
Declaración jurada	iii
Carta de aprobación del lector	iv
Carta de revisión de filóloga.....	v
Carta de autorización de autores para la consulta	vii
Agradecimiento	ix
Índice de contenidos	x
Índice de tablas	xvi
Índice de ilustraciones.....	xvii
Abreviaturas	xx
Resumen.....	xxi
Capítulo I.....	22
Problema del proyecto	22
1.1 Antecedentes y justificación	23
1.1.1 Antecedentes del contexto de la empresa.	23
1.2 Definición del problema.....	28
1.2.1 Problemática.	28
1.2.2 Problema general.	32
1.2.3 Problemas específicos.	32

1.3	Objetivos del proyecto	33
1.3.1	Objetivo general.	33
1.3.2	Objetivos específicos.	33
1.4	Alcances y limitaciones	34
1.4.1	Alcances.....	34
1.4.2	Limitaciones.	34
1.5	Cronograma de actividades.....	35
Capítulo II.....		39
Marco Teórico		39
2.1	Sistema	40
2.2	Sistema de información.....	40
2.2.1	Datos.....	40
2.2.2	Información.....	40
2.2.3	Conocimiento.	41
2.2.4	Sabiduría.	41
2.3	Software.....	42
2.3.1	Software de sistemas.	42
2.3.2	Software de aplicación.	42
2.3.3	Software de ingenierías y ciencias.	43
2.3.4	Software incrustado.....	43

2.3.5	Software de línea de productos.....	43
2.3.6	Aplicaciones web.....	43
2.3.7	Software de inteligencia artificial	44
2.4	Desarrollo de software.....	44
2.5	Metodologías de desarrollo	45
2.5.1	Cascada.	45
2.5.2	Proceso incremental.....	46
2.5.3	Proceso evolutivo.	47
2.5.4	Concurrentes.....	49
2.6	Metodologías ágiles de desarrollo.....	49
2.6.1	Extreme Programming (XP).	52
2.6.2	DevOps.	55
2.6.3	Kanban.....	57
2.6.4	Lean.	58
2.6.5	Scrum.....	60
2.7	Herramientas para la gestión de proyectos	66
2.7.1	JIRA	66
2.7.2	Microsoft Project.....	68
2.7.3	Azure DevOps.	70
Capítulo III		72

Marco Metodológico	72
3.1 Tipo de investigación	73
3.1.1 Tipo de investigación.....	73
3.1.2 Enfoque de la investigación.	73
3.2 Fuentes de información.....	74
3.2.1 Fuentes de información primaria	74
3.2.2 Fuentes secundarias	74
3.2.3 Sujetos de información.....	74
3.3 Técnicas y herramientas de recolección de datos.....	75
3.3.1 Entrevista.	76
3.3.2 Cuestionario	76
3.3.3 Observación	77
3.3.4 Documentos existentes	77
3.4 Variables.....	77
3.5 Diseño de la investigación	79
3.5.1 Análisis.....	79
3.5.2 Diseño.....	80
3.5.3 Desarrollo.....	80
3.5.4 Entrega.....	80
Capítulo IV	83

Diagnóstico	83
4.1 Diagnóstico administrativo.....	84
4.1.1 Resultados del instrumento.....	84
4.1.2 Estado actual.....	86
4.1.3 Estado óptimo.	88
4.1.4 Brecha.....	88
4.2 Diagnóstico técnico.....	88
4.2.1 Resultados de Instrumento	88
4.2.2 Estado Actual.	90
4.2.3 Estado Óptimo.....	91
4.2.4 Brecha.....	91
4.3 Diagnóstico de percepción.....	92
4.3.1 Resultados de Instrumento.	92
4.3.2 Estado Actual.	99
4.3.3 Estado Óptimo.....	99
4.3.4 Brecha.....	100
4.4 Diagnóstico general	100
Capítulo V	102
Diseño y desarrollo del proyecto	102
5.1 Desarrollo de propuesta.....	103

5.1.1	Diseño.....	103
5.1.2	Desarrollo.....	120
5.1.3	Entrega.....	141
Capítulo VI		147
Conclusiones y recomendaciones.....		147
6.1	Conclusiones.....	148
6.2	Recomendaciones.....	150
Bibliografía		153
Anexos		157
Anexo 1.....		158
Certificación Professional Scrum Master I		158
Anexo 2.....		159
Cuestionario 1: Formulario de Nivel Académico		159
Anexo 3.....		162
Cuestionario 2: Formulario de Nivel Tecnológico		162
Anexo 4.....		165
Cuestionario 3: Formulario de Percepción.....		165
Anexo 5.....		169
Carta de aceptación de la propuesta de Kamuk School.....		169

Índice de tablas

Tabla 1	32
Tabla 2	70
Tabla 1.	77
Tabla 2.	81
Tabla 3.	114
Tabla 4:	116
Tabla 5	120
Tabla 6:	122

Índice de ilustraciones

<i>Figura 1: Organigrama Kamuk School Fuente: Diego Saravia Valverde</i>	25
<i>Figura 2: Árbol de causas y efectos Fuente: Creación propia</i>	30
Figura 3: Proceso evolutivo de prototipos	47
Figura 4: Modelo de espiral.....	48
Figura 5: Scrum y XP	52
Figura 6: Prácticas de XP.....	55
Figura 7: Ciclo de vida de DevOps.....	57
Figura 8 Scrum Framework.....	61
Figura 9: JIRA	68
Figura 10: Microsoft Project	69
Figura 11: Azure DevOps.....	71
Figura 12: ¿Qué función desempeña usted en la institución.....	93
Figura 13: ¿Trabajó con el Sistema SIRA?	93
Figura 14: En una escala del uno al cinco, ¿cómo calificaría SIRA?	94
Figura 15: En una escala del uno al cinco, ¿qué tanto se podría mejorar la funcionalidad del sistema SIRA?.....	94
Figura 16: ¿Qué sistema académico considera de mayor utilidad y eficiencia, SIRA o Class?	95
Figura 17: En una escala del uno al cinco, ¿cómo calificaría el sistema Class?.....	95
Figura 18: En una escala del uno al cinco, ¿qué tanto considera usted que podría mejorar la funcionalidad del sistema Class?	96
Figura 19:¿Utilizaría un nuevo sistema académico que combine las mejores funcionalidades de SIRA y Class?	96

Figura 20: ¿Cuáles serían las tres principales características con las que debería contar un sistema académico?.....	97
<i>Figura 21: ¿Cuáles serían las tres principales características con las que debería contar un sistema académico?.....</i>	<i>98</i>
Figura 22: ¿Colaboraría con la prueba de un nuevo sistema académico para beneficio del centro educativo y los estudiantes?.....	98
Figura 23: Usuarios de KAMEDCR en Azure DevOps	115
<i>Figura 24: Menú de Azure DevOps</i>	<i>117</i>
Figura 25: Burndown Sprint 1.....	128
Figura 26: Burndown Sprint 2.....	129
Figura 27: Burndown Sprint 3.....	130
Figura 28: Burndown Sprint 4.....	131
Figura 29: Burndown Sprint 5.....	132
Figura 30: Burndown Sprint 6.....	133
Figura 31: Burndown Sprint 7.....	134
Figura 32: Burndown Sprint 8.....	135
Figura 33: Burndown Sprint 9.....	136
Figura 34: Burndown Sprint 10.....	137
Figura 35: Burndown General	138
Figura 36: Login de KAMEDCR	141
Figura 37: Página principal con súper usuario	142
Figura 38: Módulo de seguridad.....	142
Figura 39: Módulo administrativo	143

Figura 40: Módulo de profesores como administrador	143
Figura 41: Página principal con usuario profesor	144
Figura 42: Módulo de profesores con usuario profesor	144
Figura 43: Login desde dispositivo móvil.....	145
Figura 44: Página principal desde dispositivo móvil	146

Abreviaturas

SIRA: Sistema de Información y Rendimiento académico

FE: Formación Espiritual

SQL: Structured Query Language

MVC: Model, View and Controller

TI: Tecnología de Información

HP: Hewlett Packard

PC: Personal Computer

Resumen

En este documento se explica el proceso de desarrollo de una propuesta de sistema informático académico para Kamuk School. Dicha institución carecía de un sistema académico apto para satisfacer las necesidades educativas, administrativas y de seguridad. Con el fin de responder a dichas exigencias se conformó un grupo de trabajo, en calidad de estudiante de la Universidad Hispanoamericana, destinado a desarrollar un proyecto destinado a suplir los requerimientos informáticos de la institución. El grupo estuvo integrado por Mario Angulo Cordero, José Alejandro Chavarría Benavides, Diego Saravia Valverde, Daniel Valle Jiménez, Álvaro Roberth Herrera Castro y Luis Iván Duarte Solís; cada estudiante contribuye con el proyecto desde su propia área de formación, las cuales se corresponden con: Scrum Master y Product Owner, arquitectura, administración de base de datos, aseguramiento de la calidad y desarrollador. Debido a esto, las explicaciones y los resultados de análisis pueden ser coincidentes o similares, ya que están enlazados a un mismo entregable: el presentado a la institución.

Por ello, este documento se enfocará en el trabajo como Scrum Master y Product Owner, con Scrum como marco de trabajo. Este proyecto presentará a una misma persona en el ejercicio de ambos roles, según las mejores prácticas del marco de trabajo, con miras al cumplimiento de los procesos requeridos. Asimismo, la institución se comunicará directamente con el Product Owner para establecer un orden de trabajo que asegure un entregable final estable, funcional y, ante todo, entregado oportunamente.

Capítulo I

Problema del proyecto

1.1 Antecedentes y justificación

1.1.1 Antecedentes del contexto de la empresa.

Kamuk School es un centro educativo fundado en 1986 por Rommel Porras y Alexandra Brenes. El nombre de la institución proviene del segundo cerro más alto de Costa Rica, ubicado en el Parque Internacional La Amistad, en la cordillera de Talamanca. Kamuk School inició sus funciones en 1986 con el nivel de preescolar, bajo el nombre Pequitas. A lo largo de los años fueron implementándose nuevos niveles educativos hasta que el crecimiento y fortalecimiento de la institución llevó a la apertura de una segunda sede en Llorente de Tibás, donde se concentran los niveles de primaria. Finalmente, en el 2002 se inicia con secundaria, de modo que sus primeros graduados de secundaria egresan en el año 2006. Actualmente, tanto preescolar como primaria y secundaria se ubican en las mismas instalaciones, ubicadas en Llorente de Tibás.

Misión:

“Educar integralmente al estudiante, potencializando sus áreas: cognitiva, socio-afectiva y espiritual; a través de actividades curriculares y extracurriculares que desarrollen en ellos una actitud crítica, autónoma e independiente, coadyuvando en la formación de su carácter como un ser humano realizado y un ciudadano responsable para Costa Rica y el mundo.” (Kamuk School, 2016)

Visión:

“Coadyuvar en la formación de una persona con habilidades intelectuales, socio – afectivas, y espirituales, sustentados en los principios bíblicos y en las normas y valores de la sociedad costarricense.” (Kamuk School, 2016)

La necesidad de ejercer un mayor control sobre el registro académico, que permita conocer el avance de notas de cada estudiante conduce, en 2011, a la búsqueda de una opción digital que facilite la tarea, por lo que se recurre a la búsqueda de un desarrollador de software , cuya labor se oriente a la satisfacción de dichos requerimientos. En 2012 nace el sistema *SIRA*, cuyo *modus operandi* se basa en la prueba y error. Durante el camino, se fue trabajando para la satisfacción de las necesidades y solución de los errores conforme estos fueron presentándose, lo cual para el 2015 y 2016 da como resultado un software de sistema académico básico a la medida. A pesar de ello, en 2016 surge la necesidad de fusionar el sistema financiero con *SIRA*.

Sin embargo, para ese momento la institución cuenta con un único desarrollador, por lo cual se recurre a la búsqueda y análisis de un sistema ya hecho, susceptible de ser adaptado al área financiera. Así, en el 2017 se implementa *Class*, un sistema académico robusto, el cual abarca el área financiera y la académica. *Class* está siendo utilizado actualmente en la institución por administradores, profesores, estudiantes y padres de familia. En la figura 1 se presenta el organigrama de Kamuk School.

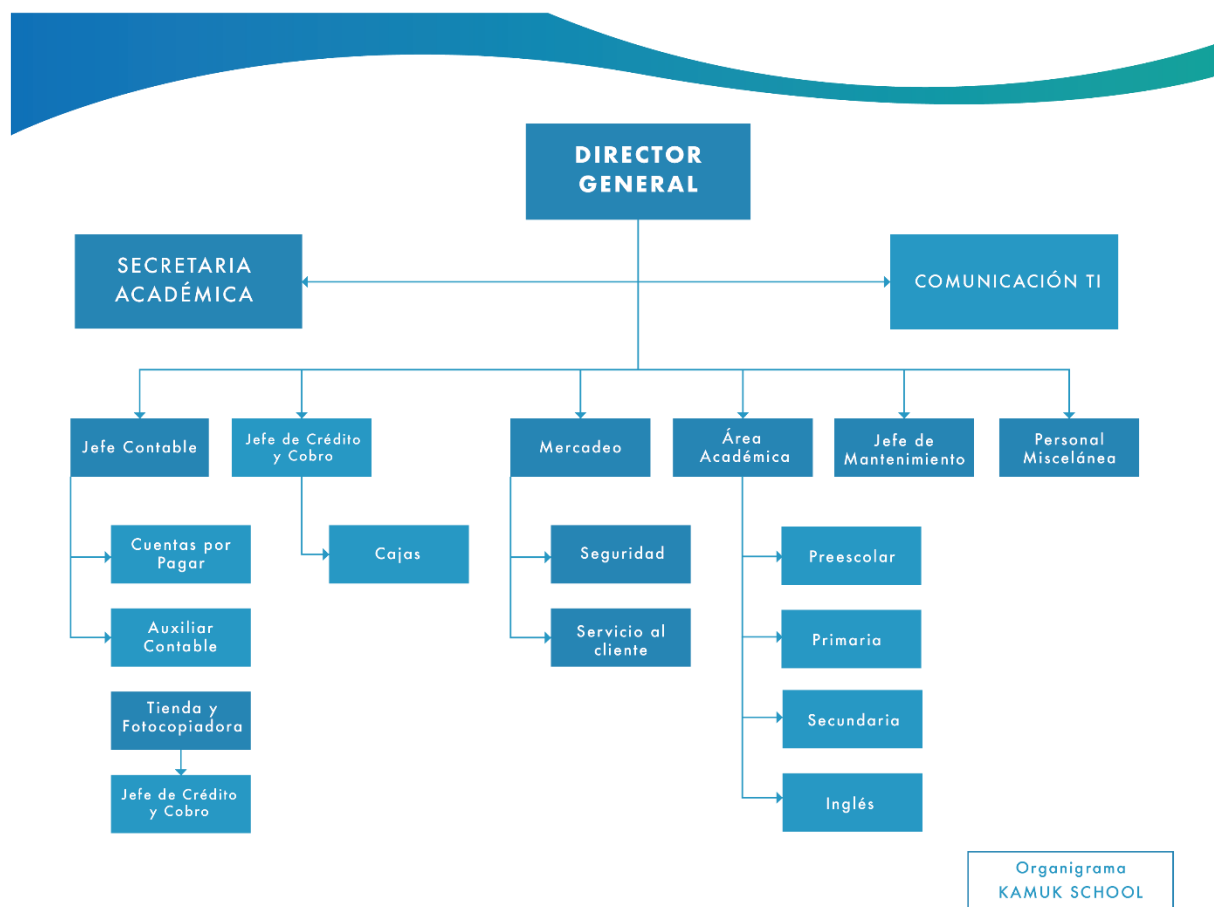


Figura 1: Organigrama Kamuk School

Fuente: Diego Saravia Valverde

Justificación del Proyecto.

Frente a la realidad de la globalización y el crecimiento de la digitalización, las exigencias del control del sistema académico institucional superan por mucho los beneficios de los procesos manuales, por lo que las empresas ya no deben depender de ellos, sobre todo cuando la institución ya ha iniciado su proceso de digitalización. “...en la sociedad, los desarrolladores en TI están cambiando los modos en que las empresas y las personas realizan sus tareas diarias.” (Joyanes Aguilar, Luis, 2015, p.

28). Tomando en consideración este supuesto, Kamuk School requiere de un software hecho a la medida, que permita y agilice la realización de sus actividades diarias y que sea adaptable a los constantes cambios sufridos en los procesos educativos, cambios que exigen a la educación un constante reinventarse y mejorarse. Ahora bien, a pesar de la velocidad con que dichas mejoras ameritan ser implementadas, el desarrollo de un software no debe tomarse a la ligera, pues se ha de analizar la metodología institucional de trabajo para cumplir adecuadamente con los requerimientos del cliente, en este caso Kamuk School.

Ante la creciente demanda de desarrollos en los últimos años, poco a poco se han descubierto metodologías de trabajo más flexibles, adaptables y enfocadas en satisfacer las necesidades del cliente final. La agilidad de estas metodologías cambia la tradicional manera de realizar el trabajo de desarrollo.

Las metodologías ágiles presentan como principal particularidad la flexibilidad, los proyectos en desarrollo son subdivididos en proyectos más pequeños, incluye una comunicación constante con el usuario, son altamente colaborativos y es mucho más adaptable a los cambios. (Montero, Cevallos, & Cuesta, 2018, p. 116)

Dentro de las muchas metodologías de desarrollo ágil conocidas, hay una en particular, Scrum.

...es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas. Scrum muestra la eficacia relativa de las técnicas de gestión de producto y las técnicas de trabajo de modo que podamos mejorar continuamente

el producto, el equipo y el entorno de trabajo. (Schwaber & Sutherland, 2017, p. 3)

Lo anterior se adapta a la realidad de desarrollo presentada por la propuesta de los módulos del sistema académico y se respalda con resultados globales. Estudios realizados en distintos lugares del mundo muestran que países como Colombia se inclinan hacia el uso de metodologías ágiles por su funcionalidad.

Las ventajas de aplicar metodologías ágiles en proyectos de software son: la facilidad para herencias en los desarrollos, mayor control sobre los proyectos y de los requisitos, la flexibilidad en el alcance, la posibilidad de tener cambios en los requisitos, las prioridades que permiten ajustarse a las necesidades del cliente, el tiempo y la documentación entre otros. (García González, Sepúlveda Castaño, & Montoya Suárez, 2017, p. 16)

Estos estudios revelan una preferencia por el marco de trabajo Scrum, por tratarse de un marco sencillo, aplicable a distintos desarrollos.

...la metodología ágil de desarrollo con mayor grado de aplicación de las empresas encuestadas de la ciudad de Medellín en el año 2016 y 2017 es “SCRUM”, debido a sus principios y características donde marca en su entorno de trabajo la simplicidad y escalabilidad, su objetivo principal es crear desarrollo ágil donde el cliente hace parte estratégico del equipo de trabajo y así garantizar el grado de satisfacción. (García González, Sepúlveda Castaño, & Montoya Suárez, 2017, p. 62)

A partir del respaldo del uso de tecnologías ágiles como Scrum y del requerimiento de un sistema académico por parte de la institución, se plantea el desarrollo de dos módulos del sistema, uno denominado *Administración general* y el otro *Profesores*. El nuevo sistema está destinado a cumplir con los requerimientos de Kamuk School y a ser parte de un exitoso marco de trabajo a nivel global, tal como Scrum.

Luego de varios entregables, se pretende ofrecer un producto final eficaz, apto para satisfacer los requerimientos no suplidos por los sistemas anteriores, los cuales condujeron a la administración a la erogación de importantes cantidades de dinero, con lo cual otras necesidades diarias del centro educativo quedan pendientes.

1.2 Definición del problema

1.2.1 Problemática.

Para Kamuk School resulta de vital importancia el contar con un sistema académico eficaz, pues este constituye la principal herramienta para constatar el crecimiento académico de la población estudiantil, fuente principal de los ingresos institucionales. Por tanto, desde el 2011 se pretende adquirir una herramienta tecnológica adecuada para satisfacer cada una de las necesidades de Kamuk School y que, además, se adapte fácilmente a los requerimientos surgidos durante los cursos lectivos.

La creación de la primera herramienta, SIRA, contribuyó con el aprendizaje del trabajo con herramientas digitales por parte del personal; sin embargo, la adaptación a

los cambios topa con un significativo inconveniente: la institución tiene a su disposición un único desarrollador. Además, el desarrollo de software con el que cuenta la institución en ese momento es anticuado, pues emplea tecnologías viejas y carece de la posibilidad de adaptación del sistema financiero, prioritario para el buen funcionamiento de la institución. Aunque la herramienta ofrecía facilidad de acceso a los profesores (usuarios más frecuentes), quienes la utilizan para sus distintas actividades diarias, el peso de una tecnología anticuada, desprovista de adaptabilidad a un sistema financiero, concientiza a la administración educativa en relación con la importancia de la adquisición de un software finalizado y funcional.

Class trae consigo el uso de un software un poco más moderno, con facilidad de empleo en dispositivos móviles y computadoras, y, primordialmente, con la capacidad de adaptación al sistema financiero, con el cual se requiere mantener comunicación. Class, al ser un software maduro y robusto, exige cuidado en cuanto al costo se refiere. Actualmente Kamuk School sostiene un contrato de uso por miles de dólares, lo cual representa para la institución una significativa inversión económica anualmente; además requiere el pago de horas de programación cada vez que se necesitan cambios personalizados. Se trata de una situación recurrente, pues la herramienta no cuenta con facilidad de personalización.

Poco a poco la institución fue encontrando errores de software no resueltos por el proveedor, por lo cual la institución se ve obligada a incurrir en gastos por concepto de horas de programación para solucionar tales fallas. Algunos ejemplos de estos errores son: incompatibilidad con buscadores, tecnologías de desarrollo que abren

brechas en tema de seguridad, módulos carentes de principios de usabilidad, necesidad de nuevas bases de datos cada año, entre otras.

Ante esta situación, Kamuk School, buscando lo mejor para sus estudiantes y profesores, se encuentra en búsqueda de un desarrollo de software a la medida, con tecnologías actualizadas y facilidad de adaptación de cambios.

1.2.1.1 Diagrama Causa Efecto.

ÁRBOL DE CAUSAS Y EFECTOS

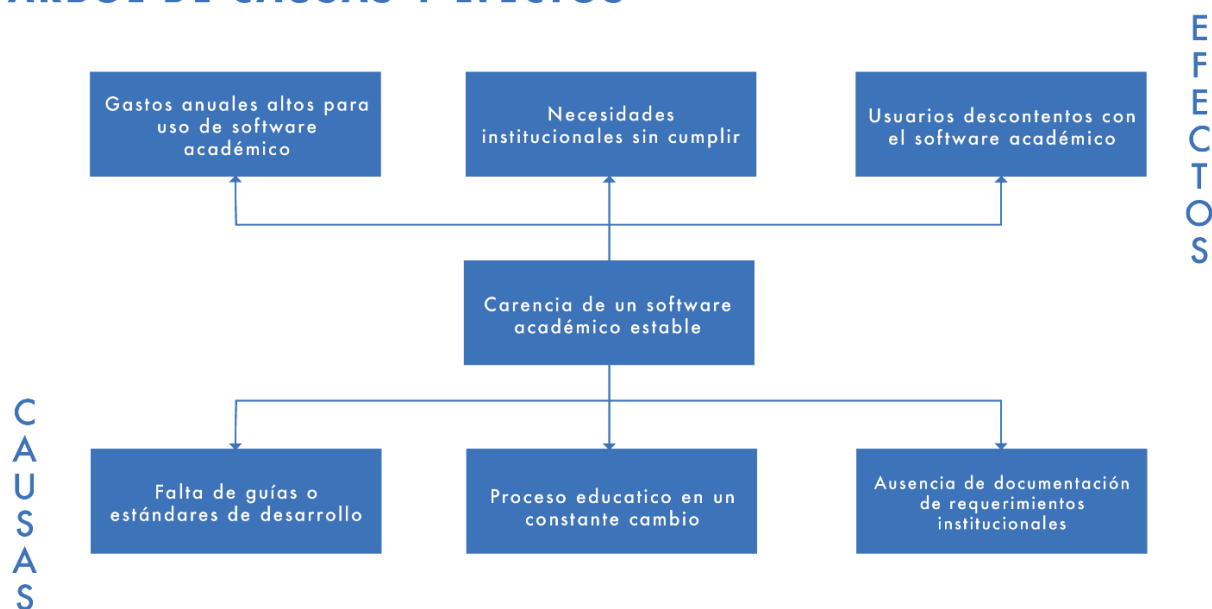


Figura 2: Árbol de causas y efectos

Fuente: Creación propia

Actualmente, Kamuk School carece de un software académico estable debido a que los sistemas previos no han logrado solventar las necesidades institucionales; esto se observa en la figura 2. Se identifican las siguientes causas:

Falta de guías o estándares de desarrollo: Kamuk School no posee lineamientos para el desarrollo de proyectos.

Proceso educativo en un constante cambio: al ser una institución actualizada, los cambios en los procesos de evaluación son constantes y los sistemas carecen de capacidad de adaptación necesaria para ajustarse a dichos cambios.

Ausencia de documentación de requerimientos institucionales: a pesar de tener un reglamento interno que proporciona los requerimientos institucionales básicos, no existe una documentación detallada donde se consignen las necesidades para un desarrollo; es decir, la documentación informativa para el desarrollo es pobre.

Lo anterior trae consigo lo siguiente:

Elevados gastos anuales en el uso de software académico: la necesidad de un sistema académico lleva a considerables gastos anuales por concepto de licencias, cantidad de usuarios y mantenimiento del sistema.

Necesidades institucionales sin cumplir: al no tener un registro detallado de las necesidades institucionales, la mayoría de estas no se satisfacen, debido a que los proveedores no conocen con detalle los procesos internos.

Usuarios descontentos con el software académico: a pesar de estar conformes con el software utilizado, los funcionarios identifican diversos aspectos por mejorar en los sistemas académicos; ellos esperan un software que realmente facilite la labor diaria por lo que se genera una importante insatisfacción y falta de aceptación del sistema académico.

1.2.2 Problema general.

¿Cómo gestionar durante el 2019 una propuesta de desarrollo del software de un sistema informático académico hecho a la medida de Kamuk School para los módulos de administración y profesores utilizando el marco de trabajo de desarrollo ágil Scrum?

1.2.3 Problemas específicos.

1. ¿Cómo presentar un entregable de la propuesta final satisfactorio para el cliente de un proyecto de desarrollo de los módulos de administración y profesores de un sistema académico hecho a la medida en Kamuk School?
2. ¿De qué manera deben administrarse los tiempos de entrega del desarrollo de los módulos de administración y profesores de un sistema académico hecho a la medida en Kamuk School, utilizando las mejores prácticas de metodologías ágiles?
3. ¿Cómo mantener la comunicación con el cliente durante el desarrollo de los módulos de administración general y profesores de un sistema académico hecho a la medida de Kamuk School, utilizando las mejores prácticas de metodologías ágiles?

1.3 Objetivos del proyecto

1.3.1 Objetivo general.

Gestionar la propuesta de desarrollo de los módulos de administración y profesores de un sistema académico hecho a la medida para Kamuk School durante el año 2019, utilizando el marco de desarrollo ágil Scrum.

1.3.2 Objetivos específicos.

1. Analizar la situación actual de Kamuk School en relación con la utilización de sistemas académicos y gestión académica durante el año 2019, para desarrollar una propuesta de sistema académico apto para la satisfacción de las necesidades institucionales.
2. Presentar un entregable final de la propuesta de desarrollo de los módulos de administración y profesores de un sistema académico hecho a la medida en Kamuk School durante el año 2019.
3. Administrar los tiempos de entrega durante el desarrollo de los módulos de administración y profesores de un sistema académico hecho a la medida en Kamuk School durante el año 2019 utilizando las mejores prácticas de metodologías ágiles.
4. Mantener la comunicación y el trabajo con el cliente durante el desarrollo de los módulos de administración y profesores de un sistema académico hecho a la medida en Kamuk School durante el año 2019 utilizando las mejores prácticas de metodologías ágiles.

1.4 Alcances y limitaciones

1.4.1 Alcances.

1. El primer entregable del proyecto corresponde al análisis de la situación actual de la empresa en relación con el uso de los sistemas académicos, tanto de los antiguos como del actual.
2. El segundo entregable del proyecto será el documento explicativo del proceso académico de la institución; es decir, de su proceso de planificación y evaluación.
3. En el tercer entregable se presentará una propuesta final en la cual se consignan los avances parciales del proyecto para el debido análisis y aprobación por parte de Kamuk School.
4. Para la administración de los tiempos del proyecto se utiliza una herramienta de gestión de proyectos para mostrar el avance de los entregables a lo largo el desarrollo de la propuesta.
5. Se mantendrá un historial de los entregables del proyecto en conjunto con la herramienta para administración de tiempos del proyecto.

1.4.2 Limitaciones.

1. Se debe respetar la privacidad del código y de los lineamientos de los derechos de autor de los sistemas académicos empleados.
2. Deben seguirse las rúbricas y el proceso de trabajo y evaluación de Kamuk School en su guía o informe de procesos estudiantiles.

3. Debe seguirse el procedimiento y los procesos de la metodología de desarrollo seleccionada para el proyecto, con el fin de asegurar su calidad y funcionamiento.
4. Deben respetarse las capacidades tecnológicas de la institución para desarrollar la propuesta de sistema académico. Esto incluye hardware, software y licenciamiento.
5. Se han de respetar la disponibilidad de Kamuk School para la presentación de los avances del proyecto, tanto de los parciales como del final.
6. En relación con el tema económico, no existe limitación para el uso de herramientas ya que Azure DevOps es de acceso gratuito y para el desarrollo se utilizarán las licencias de Microsoft que, como estudiantes, están disponibles. A pesar de ello, la implementación del proyecto dependerá de la institución y su capacidad económica para el manejo de proyectos. Dicha información es de carácter privado y no se abordará este tema dentro del este documento.

1.5 Cronograma de actividades

Tabla 1.
Tabla de Cronograma de actividades

<i>Fecha</i>	<i>Actividad</i>
<i>Sprint I</i> <i>(18 de febrero – 4 de marzo)</i>	<i>Definición de requerimientos (como Product Owner)</i>
	<i>Explicación de los requerimientos al equipo de trabajo (como Product Owner)</i>
	<i>Explicación de Scrum al equipo de desarrollo y al cliente (como Scrum Master)</i>

Fecha	Actividad
	Monitoreo de artefactos y eventos de Scrum (como Scrum Master)
	Reuniones con el personal de la institución para la presentación de los avances (como Product Owner)
<i>Sprint II</i> (5 de marzo – 19 de marzo)	Monitoreo de artefactos y eventos de Scrum (como Scrum Master)
	Reuniones con el personal de la institución para para la presentación de los avances (como Product Owner)
	Redefinición de requerimientos (como Product Owner)
<i>Sprint III</i> (20 de marzo – 3 de abril)	Monitoreo de artefactos y eventos de Scrum (como Scrum Master)
	Reuniones con el personal de la institución para la presentación de los avances (como Product Owner)
	Monitoreo de artefactos y eventos de Scrum (como Scrum Master)
<i>Sprint IV</i> (4 de abril – 18 de abril)	Reuniones con el personal de la institución para la presentación de los avances (como Product Owner)
	Monitoreo de artefactos y eventos de Scrum (como Scrum Master)
<i>Sprint V</i> (19 de abril – 3 de mayo)	Reuniones con el personal de la institución para la presentación de los avances (como Product Owner)
	Monitoreo de artefactos y eventos de Scrum (como Scrum Master)
<i>Sprint VI</i> (4 de mayo – 18 de mayo)	Monitoreo de artefactos y eventos de Scrum (como Scrum Master)

Fecha	Actividad
	Reuniones con el personal de la institución para la presentación de los avances (como Product Owner)
	Recolección de datos mediante cuestionarios en Kamuk School
<i>Sprint VII</i> (18 de mayo – 1 de junio)	Monitoreo de artefactos y eventos de Scrum (como Scrum Master)
	Reuniones con el personal de la institución para la presentación de los avances (como Product Owner)
	Monitoreo de artefactos y eventos de Scrum (como Scrum Master)
<i>Sprint VIII</i> (2 de junio – 15 de junio)	Reuniones con el personal de la institución para la presentación de los avances (como Product Owner)
	Monitoreo de artefactos y eventos de Scrum (como Scrum Master)
<i>Sprint IX</i> (16 de junio – 30 de junio)	Reuniones con el personal de la institución para la presentación de los avances (como Product Owner)
	Monitoreo de artefactos y eventos de Scrum (como Scrum Master)
<i>Sprint X</i> (1 de julio – 15 de julio)	Reuniones con el personal de la institución para la presentación de los avances (como Product Owner)
	Monitoreo de artefactos y eventos de Scrum (como Scrum Master)
<i>Sprint XI</i> 16 de julio – 31 de julio)	Reuniones con el personal de la institución para la presentación de los avances (como Product Owner)

Fuente: Equipo de proyecto de KAMEDCR

Capítulo II

Marco Teórico

2.1 Sistema

Un sistema de información se define como el “Conjunto de cosas que relacionadas entre sí ordenadamente contribuyen a determinado objeto.” (RAE, 2019). Una definición más precisa lo explica como un conjunto de elementos que trabajan para alcanzar una meta en común. De acuerdo con su objetivo, existen diferentes sistemas tales como: el solar, digestivo, nervioso, térmico, etc. Cada uno de ellos se enfoca en sus elementos para cumplir un fin específico según las necesidades de su entorno.

2.2 Sistema de información

El sistema de información es un tipo de sistema ligado a su entorno de trabajo mediante la tecnología. Joyanes (2015), en su texto *Sistemas de Información en la Empresa*, sostiene que para definir adecuadamente un sistema de información, debe haber una justa comprensión de sus componentes: datos, información, conocimiento, sabiduría y jerarquía de la información.

2.2.1 Datos.

Los datos pueden definirse como los componentes de un cuerpo de conocimiento, los cuales por sí mismos carecen de significado. Se trata de elementos que simplemente se muestran pero sin utilidad alguna por sí solos.

2.2.2 Información.

La información se puede definir como un conjunto de datos acomodados según un orden lógico para generar una idea o finalidad precisa. A diferencia de los datos, la

información se puede procesar con el fin de ser interpretada o compartida según la necesidad del caso.

2.2.3 Conocimiento.

El conocimiento se entiende como la aplicación de la información, es decir, los datos se acomodan intencionalmente de manera que luego sea posible procesarlos y aplicarlos a un problema u objetivo. La información, una vez procesada, se considera como conocimiento, ya que se puede ver una constante en su utilidad para dicho problema u objetivo definido.

2.2.4 Sabiduría.

Se puede definir la sabiduría como la aplicación del conocimiento junto con la intuición y el juicio profesional de las personas. De igual manera puede definirse como el culmen de los niveles de acomodo de datos, pues con tal acomodo el conocimiento aplicado se perfecciona mediante el uso constante, de modo que pueda ser utilizarlo en la toma de decisiones respaldadas por su constante comportamiento.

A partir de lo anterior, un sistema de información puede definirse como el conjunto de elementos de información, recopilados y organizados de manera intencional para un fin específico. De esta manera, el sistema de información se constituye en una herramienta útil para la toma de decisiones y, finalmente, a nivel empresarial reporta importantes beneficios, pues la experiencia en recopilación de datos evidencia la presencia de constantes y posibles variables en los procesos y situaciones.

2.3 Software

Se define software como el: “Conjunto de programas o aplicaciones (apps) de escritorio o de dispositivos móviles que instruyen a los dispositivos de hardware a procesar datos u otras entradas tales como órdenes de voz, reconocedores de texto, escáneres, etc.” (Joyanes Aguilar, Luis, 2015, p. 8). Dicha definición incluye la utilización de la información previamente explicada según la finalidad del software. La definición de software abarca siete tipos: de sistemas, de aplicación, de ingeniería y ciencias, incrustado, de línea de productos, aplicaciones web y de inteligencia artificial. (Pressman, Roger S., 2010, p. 6).

2.3.1 Software de sistemas.

El software de sistemas es aquel creado con la finalidad de servir a otros tipos de software para que estos cumplan sus funciones. También se le describe como el software creado para complementar el funcionamiento de otros. Normalmente se utiliza para procesar datos acomodados y desordenados, y con éstos alimentar el funcionamiento del software que complementan.

2.3.2 Software de aplicación.

Este otro tipo de software se enfoca en servir a las necesidades de negocio al que sirve. Generalmente éstos procesan información de utilidad para la toma de decisiones de los altos mandos del negocio trabajado.

2.3.3 Software de ingenierías y ciencias.

El software de ingenierías y ciencias se fundamenta en el trabajo y solución de algoritmos complejos en las distintas ramas de ingeniería o ciencias. Actualmente han modificado su funcionamiento original y algunos han pasado a ser software de sistemas para asistir a otros tipos de software, cuyas funciones son más específicas; de esta manera se distribuyen las cargas de trabajo y se maximiza su eficiencia.

2.3.4 Software incrustado.

El software incrustado es aquel que forma parte de un sistema o producto y realiza una operación definida para el sistema como tal y para el usuario final. Normalmente sus capacidades o funciones son limitadas, ya que en realidad es solo una pieza del engranaje total propio de un proceso.

2.3.5 Software de línea de productos.

Este tipo de software se crea para responder a un fin específico y puede utilizarse por una amplia variedad de consumidores. Aunque por lo general, posee una función muy específica, puede utilizarse sin importar el tipo de mercado en que se encuentre, tal como un software de venta.

2.3.6 Aplicaciones web.

La aplicación web corresponde a un software destinado a la utilización de la web abierta. Actualmente, no se reduce al simple uso de texto y gráficos, sino que busca

ofrecer soluciones a los distintos problemas de los usuarios, pues disponible en cualquier lugar del mundo.

2.3.7 Software de inteligencia artificial

Este último tipo de software utiliza algoritmos no convencionales para resolver tipos nuevos de problemas que, de hecho, no son fáciles de resolver por medios tradicionales. Éste software se utiliza en robótica, reconocimiento de patrones, redes inteligentes, asistentes de uso como Bixby de Samsung o Alexa de Amazon, entre otros.

2.4 Desarrollo de software

La creación del software requiere de una serie de pasos ordenados y definidos, cuyo resultado final permite la funcionalidad esperada. A pesar de ello, no todos los desarrollos se llevan a cabo de la misma manera. Cada uno de ellos varía según la finalidad, equipo de trabajo, ambiente de trabajo, gestión de éste, e incluso, del lapso en el que se debe presentar. Cada desarrollo exige pasos o etapas muy propios, por lo cual varían de uno a otro, y para todos ellos existen al menos tres pasos que les son comunes:

Diseño: Se refiere al proceso de planeación del software por desarrollar. En este punto se analizan sus funciones (lo qué hará el software), cómo lo hará y también cómo se creará o desarrollará.

Desarrollo: Esta es la etapa en la que se crea el software; se trata de la fase donde se ejecuta lo planteado en la etapa de diseño.

Mantenimiento: Se trata de la etapa en la cual se observa y se analiza el funcionamiento del software creado, con el fin de detectar posibles anomalías o potenciales mejoras. Cabe destacar que en cuanto a requerimiento de tiempo se refiere, el mantenimiento es una de las etapas que exige un lapso mayor.

2.5 Metodologías de desarrollo

Como se mencionó en las páginas anteriores, existen muchos modelos de desarrollo de software, dentro de los cuales se observan los siguientes: cascada, proceso incremental, proceso evolutivo y concurrentes. (Pressman, Roger S., 2010, p. 33). En las líneas siguientes se explican con mayor detalle.

2.5.1 Cascada.

Se trata de un modelo de desarrollo cuyo paradigma más antiguo de la ingeniería del software. (Pressman, Roger S., 2010, p. 34). Este modelo indica un proceso secuencial de trabajo que va desde el levantamiento de requerimientos hasta la entrega, una vez finalizado. En la figura 3 se muestra el ciclo de funcionamiento de este modelo.

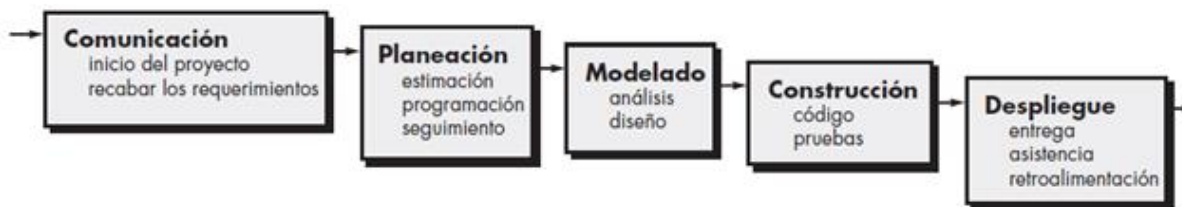


Figura 3: Modelo Cascada

Fuente: Ingeniería de del software, Un enfoque práctico

De acuerdo con la propuesta de este modelo cada fase se realiza en su totalidad y se entrega el desarrollo ya acabado; es decir, el producto definitivo, lo que quiere decir que no se realizan entregas previas (adelantos). Ahora bien, a pesar de tratarse del modelo más antiguo, este presenta ciertos problemas:

- 1) Difícilmente se cumple con el flujo del proceso informado. Los cambios en el desarrollo generan gran incertidumbre a lo largo del proceso de desarrollo del proyecto y esto hace difícil su entrega.
- 2) Es difícil para el cliente informar al desarrollador acerca de todo cuanto necesita el software para funcionar adecuadamente. A menudo, los requerimientos varían de acuerdo con los resultados esperados.
- 3) El cliente debe esperar hasta que el entregable final para constatar el resultado del proyecto, lo cual implica que cualquier disconformidad del cliente se ve hasta después de la entrega del producto final, es decir, del proyecto final, y esto dificulta la realización de los posibles cambios necesarios para cumplir con los requerimientos.

2.5.2 Proceso incremental.

El proceso incremental consiste en presentar pequeños avances a lo largo del desarrollo del proyecto. Conforme progresa el entregable va aumentando la funcionalidad del sistema, ya que cada nuevo incremento trae consigo nuevas funciones o soluciones para el cliente. Este método es utilizado cuando no se cuenta con suficiente personal como para terminar el proyecto fácilmente y con prontitud. De hecho, se avanza en el desarrollo a través de pequeñas entregas funcionales.

2.5.3 Proceso evolutivo.

Se trata de un tipo de software en el que se realizan cambios constantes durante el proceso de su elaboración. El software se inicia a partir de los requerimientos generales sin profundizar en detalles. El proceso del desarrollo adquiere un carácter iterativo mediante el cual el software evoluciona hasta alcanzar a su versión final. Este tipo de proceso se divide en dos: por prototipos o de espiral. (Referencia, pág. 37).

2.5.3.1 Prototipos.

Los prototipos se basan en ideas u opciones de software planteadas para cumplir los requerimientos generales desarrollados. Presenta un ciclo que se puede observar en la figura 4:

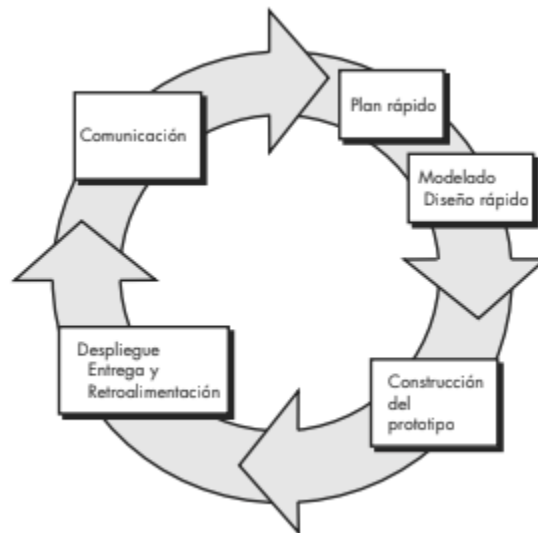


Figura 3: Proceso evolutivo de prototipos

Fuente: Ingeniería de del software, Un enfoque práctico

Este tipo de software ofrece mayor tranquilidad a los involucrados, ya que les facilita un acercamiento al sistema y les permite constatar que éste es real y alcanzable. Se debe tener claro que este modelo de trabajo constituye una manera de

utilizar los requerimientos generales como insumo para plantear una idea, a partir de la cual formular los requerimientos específicos.

2.5.3.2 *Espiral.*

El modelo de espiral es incremental y se va mejorando o perfeccionando el software a lo largo del proyecto. Cada vuelta completa de la espiral se considera como una parte de la evolución del software. A diferencia del prototipo, este modelo puede aplicarse durante todo el ciclo de vida del software desarrollado. También este modelo aplica los prototipos como medios para reducir los riesgos y, sobre todo, permite utilizar los prototipos en cualquier momento de la evolución del software. Su ciclo se explica con la figura 5:



Figura 4: Modelo de espiral

Fuente: Ingeniería de del software, Un enfoque práctico

2.5.4 Concurrentes.

“El modelo de desarrollo concurrente, en ocasiones llamado ingeniería concurrente, permite que un equipo de software represente elementos iterativos y concurrentes de cualquiera de los modelos de proceso...” (Pressman, 40). Este modelo define o explica cada actividad del proceso de desarrollo existente en el proceso; sin embargo, cada una en un estado distinto y no necesariamente en forma lineal.

2.6 Metodologías ágiles de desarrollo

A pesar de la funcionalidad de las metodologías de desarrollo tradicionales, la evolución del desarrollo de software demanda la búsqueda de nuevas metodologías de trabajo. Necesidades como la adaptación a cambios constantes, la reducción de los tiempos de entrega y el mantenimiento del nivel de calidad conducen a la búsqueda de nuevas formas de desarrollo de software para cumplir al cliente y a su vez facilitar el proceso a quienes lo siguen. Lo anterior motiva una reunión expertos, específicamente de diecisiete especialistas en la industria del desarrollo, los cuales se enfocan en establecer valores y principios para el rápido desarrollo de software con capacidad de adaptación a los cambios. También, buscan realizar proyectos incrementales e iterativos para brindar al cliente, parciales funcionales.

Como resultado de dicha reunión, nace *el Manifiesto Ágil* que valora:

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente conforma el principal factor de éxito de un proyecto de software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que

el equipo se adapte a él automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo con base en sus necesidades.

- Desarrollar un software que funcione más que conseguir una buena documentación. La regla consiste en no producir documentos a menos de que se requieran de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos y centrarse en lo fundamental.
- La colaboración con el cliente más que la negociación de un contrato. Se propone la existencia de una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos ha de ser la que determine el ritmo de la marcha del proyecto y asegure su éxito.
- Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta. (Canós & Letelier, 2012, pp. 2-3)

El *Manifiesto ágil* establece los siguientes principios:

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
 - Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo.
- Los Procesos Ágiles* aprovechan el cambio para proporcionar ventaja competitiva al cliente.

- Entregamos software funcional frecuentemente, dentro de un lapso que oscila entre dos semanas y dos meses, preferentemente en un periodo lo más corto posible.
 - Los responsables del negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
 - Los proyectos se desarrollan en torno a individuos motivados. Es necesario
 - El método más eficiente y efectivo de proporcionar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
 - El software funcionando es la medida principal de progreso.
 - *Los Procesos Ágiles* promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo de trabajo constante indefinidamente.
 - La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
 - La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
 - Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
 - Al seguir intervalos regulares el equipo reflexiona acerca de cómo mejorar su efectividad para, a continuación, ajustar y perfeccionar su comportamiento.
- (Beck, Kent et al., 2001)

De estas metodologías sobresalen: *Extreme Programming*, *DevOps*, *Kanban*, *Lean* y *Scrum*.

2.6.1 Extreme Programming (XP).

Esta se puede definir como:

“Concebida para proyectos de pequeño y mediano tamaño, prevé participación activa y frecuente con el cliente. Describe detalladamente las prácticas de desarrollo que se van a utilizar...” (Fuentes, 2015, p. 9)

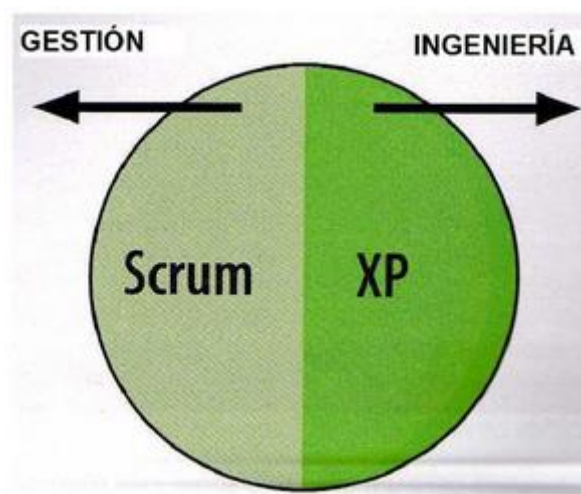


Figura 5: Scrum y XP

Fuente: Desarrollo de Software ÁGIL: Extreme Programming y Scrum

A diferencia de otras metodologías, XP se basa en la ingeniería por utilizar tal y como se ve en la figura 6, siempre en procura de seguir la filosofía del *Manifiesto Ágil*.

2.6.1.1 Historias de usuario

Las historias de usuario son el método de recolección de requerimientos funcionales y no funcionales del proyecto. La recopilación de éstas se da dentro de un proceso sumamente dinámico y flexible, y no requiere que las historias de usuario sean recogidas todas al inicio, ya que sus cambios se definen al comienzo de cada iteración.

Cabe destacar que si bien las historias de usuario tienen varias plantillas de uso, no existe una en particular definida para ello.

2.6.1.2 Roles

XP define siete roles en su proceso:

Cliente: constituido por una única persona encargada de definir las historias de usuario, su prioridad y cuáles de éstas se toman para cada iteración.

Encargado de pruebas (Tester): se trata de un colaborador, quien, junto con el cliente se encarga de definir las pruebas funcionales y de realizar las pruebas del desarrollo.

Encargado de seguimiento (Tracker): éste verifica las estimaciones y el tiempo tomado para cada proceso. Básicamente, está a cargo de asegurar el cumplimiento de los objetivos dentro de los tiempos establecidos.

Entrenador (coach): es el encargado de asegurar el cumplimiento de las mejores prácticas de XP.

Consultor: éste es ajeno al equipo de XP y sólo colabora en la resolución de problemas específicos.

Gestor (Big boss): está encargado de la coordinación del proyecto, además de constituir el punto de relación entre el cliente y los programadores.

2.6.1.3 Proceso

XP es un proceso que consta de seis fases:

Exploración: puede tomar pocas semanas o meses. En esta fase se establecen las historias de usuario y el equipo se acopla a la tecnología mientras desarrolla los prototipos.

Planificación de la entrega: durante esta fase, el cliente establece la prioridad de cada historia y el equipo estima el esfuerzo de cada una de ellas dentro de un rango de uno a tres.

Iteraciones: consiste en los ciclos repetibles proyecto antes de ser entregado. Cada iteración debe durar como máximo tres semanas.

Producción: consiste en la fase de llevar el proyecto terminado en las iteraciones a al ambiente de trabajo diario.

Mantenimiento: una vez que la iteración termina, se lleva a producción en tanto la siguiente iteración se encuentra en proceso; XP asegura el funcionamiento del proyecto ofreciéndole soporte al cliente.

Muerte del proyecto: ocurre cuando el proyecto termina con las historias de usuario.

2.6.1.4 Prácticas de XP

XP busca disminuir la curva exponencial del costo del cambio a lo largo del proyecto para que el diseño de evolución pueda seguirse. Esto se consigue gracias a las tecnologías disponibles y a las prácticas de XP: el juego de la planificación, entregas pequeñas, metáfora, diseño simple, refactorización, programación en parejas, propiedad colectiva del código, integración continua, 40 horas por semana, cliente *in situ* y estándares de programación. (Letelier & Letelier, 2006). La figura 7 muestra la organización de las prácticas del modelo XP.

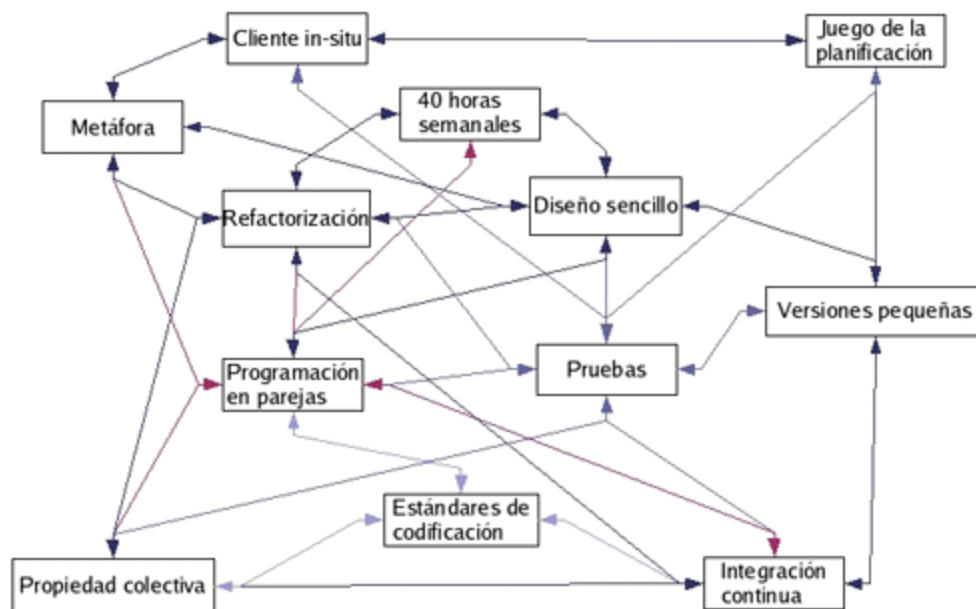


Figura 6: *Prácticas de XP*

Fuente: Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)

2.6.2 DevOps.

La palabra *DevOps* proviene de las palabras en inglés “development” y “operation”, es decir, desarrollo y operación respectivamente. (Verona, 2016, p. 2). DevOps consiste en un grupo de prácticas destinadas a reducir el tiempo transcurrido entre el momento de la confirmación de un cambio en el sistema y aquel en el que el cambio toma lugar en producción, manteniendo la alta calidad. (Bass, Weber, & Zhu, 2015)

Se presentan cinco categorías de prácticas de DevOps:

2.6.2.1 *Operaciones como ciudadanos de primera clase desde el punto de vista de requerimientos*

Este concepto asume operaciones como un punto clave al momento de plantear los requerimientos del sistema. De esta manera, se asegura el cumplimiento de lo esperado por operaciones acerca del desarrollo.

2.6.2.2 *Desarrollo es responsable del manejo de incidentes*

Esta práctica busca la reducción del tiempo entre la identificación de un error y su reparación, de tal manera que el trabajo del equipo de desarrollo se inserta en un proceso de agilización de mejoras o arreglo de errores.

2.6.2.3 *Desarrollar la implementación del desarrollo con operaciones y desarrollo*

Esta práctica busca el cumplimiento de la alta calidad de implementación en producción. Asimismo contribuye a la identificación y reparación del error en el menor tiempo posible.

2.6.2.4 *Implementación continua*

En esta parte del proceso se asegura de que las implementaciones disminuyan el tiempo de implementación del desarrollo del código, facilitando entregables al cliente.

2.6.2.5 Desarrollo de código de infraestructura

Esto busca el uso scripts de implementación con las prácticas como código de aplicación, con lo cual se asegura una implementación adecuada del software libre de errores. El ciclo de vida de DevOps se define como se observa en la figura 8:

Figura 7: *Ciclo de vida de DevOps*



Fuente: Creación propia

2.6.3 Kanban.

Kanban, representada por una tarjeta Kanban, se moverá a través de las diversas etapas de su trabajo hasta su finalización. A menudo se habla de él como un método de extracción, de forma que usted tira de sus tareas a través de su flujo de trabajo, ya que permite a los usuarios mover de sitio libremente las tareas en un entorno de trabajo basado en el equipo. (Kanban Tool, 2019)

Kanban se basa en cinco principios básicos:

Mantener visible lo que se hace: según este principio el consignar todas las tareas y elementos en un tablero permite que todas las personas del proyecto se mantengan enteradas acerca de lo que se está haciendo. Dicho tablero es la principal herramienta de Kanban, pues contiene los estados de sus tareas en “por hacer”, “en proceso” y “terminado”.

Limite la cantidad de trabajo en proceso: esto consiste en fijar metas alcanzables con miras a no sobrecargarse con trabajo. Mantener un orden en el volumen de trabajo facilita un adecuado cumplimiento de las tareas y con ello se previenen las tareas sin completar al final del proceso.

Realice un seguimiento de su tiempo: se debe monitorear y evaluar el trabajo realizado de modo que sea posible identificar las áreas de mejora a lo largo del proyecto con mayor prontitud y agilidad.

Lectura fácil de indicadores visuales: se pretende alcanzar una mayor y más fácil comprensión del trabajo realizado. El uso de tarjetas de colores contribuye a distinguir a simple vista los trabajos realizados y aquello que se encuentra pendiente.

Identificar los cuellos de botella y eliminar lo descartable: se deben aprovechar los plazos y ciclos de ejecución para detectar errores en el camino y ajustar el flujo de trabajo a beneficio con el propósito de eliminar lo que no sea necesario y así optimizar el proceso.

2.6.4 Lean.

Lean es un término cuya traducción al español es *fino* o *esbelto*. Esta metodología nace con el sistema de producción de Toyota y poco a poco se fue mejorando hasta adquirir sus características actuales. Esta metodología se basa de principios que operan como ideas o guías conducentes hacia la disciplina, mientras que las prácticas se refieren a todo cuanto es posible realizar para cumplir los principios. Lean se basa en siete principios: eliminar el desperdicio, amplificar el aprendizaje,

decidir lo más tarde posible, entregar tan rápido como sea posible, capacitar al equipo, construir integridad intrínseca y mirar el todo. (Poppendieck & Poppendieck, 2003).

2.6.4.1 *Eliminar el desperdicio*

Este principio lo que busca es que todo aquel proceso -comunicación, documentación o impedimento- sea apartado del proyecto con el fin de enfocarse únicamente en lo necesario. Esto se debe realizarse de manera iterativa hasta obtener únicamente lo necesario y útil para el cliente.

2.6.4.2 *Amplificar el aprendizaje*

Este principio asume el desarrollo como parte de un aprendizaje continuo, y cuanto mayor sea el aprendizaje, mejor será el desarrollo. A su vez, según Lean las pruebas deben realizarse tan pronto se cuente con el código, de modo que se evite la documentación o planificación detallada a futuro de manera innecesaria.

2.6.4.3 *Decidir lo más lejano posible*

Según este principio las decisiones han de tomarse de la manera lo más lejana o tarde posible con el fin de hacerlo lo más objetivamente, con suficientes hechos y con el mínimo nivel de especulación.

2.6.4.4 *Entregar tan rápido sea posible*

Este principio indica que entre más reducido sea el tiempo de entregas, mejores resultados obtendrá el proyecto. Cuando los clientes tienen poco tiempo para pensar, deben esperar a contar con mayor cantidad información y más detallada de lo que ocupan para solicitar cuanto desean de manera más precisa. A su vez, se hacen entregas reales de lo que ocupa el cliente recientemente y no de aquello que ocupaba mucho tiempo atrás, brindando así una mayor satisfacción.

2.6.4.5 *Capacitar al equipo*

El equipo toma parte de las decisiones del proyecto. Resulta importante capacitarlo de modo que sea capaz de opinar con criterio y no por presión o necesidad. Aunque existen quienes piensan que esto no es una prioridad, Lean se enfoca en este principio con miras a que el equipo de desarrollo tenga bases fuertes para trabajar y no se desvíe de la necesidad real del proyecto.

2.6.4.6 *Construir integridad intrínseca*

Éste establece el cumplimiento de lo esperado de manera integral, es decir, debe cumplirse con lo estipulado, con miras a obtener un cliente satisfecho de ver su necesidad cubierta. Además, el principio guía a que el software se mantenga útil a lo largo del tiempo.

2.6.4.7 *Mirar el todo*

Lean enfoca este último principio en no observar cada parte según su funcionalidad, sino en su funcionamiento como parte de un todo mayor, es decir, como sistema o software en conjunto. Es importante asegurar la funcionalidad entre elementos desarrollados. El principio busca que el proyecto sea vea en su totalidad de modo que no se pierdan detalles en su integración.

2.6.5 *Scrum.*

A pesar de encontrarse dentro de la categoría de metodologías ágiles, *Scrum* se define como un marco de trabajo para resolver problemas complejos adaptativos. En dicho marco de trabajo se pueden utilizar varios procesos o técnicas para llevarlo a cabo. Scrum se basa en la teoría empírica según la cual el conocimiento se adquiere

con el tiempo y la experiencia aplicada de una manera iterativa para optimizar su funcionalidad. Su ciclo de funcionamiento se ve reflejado en la figura 9.

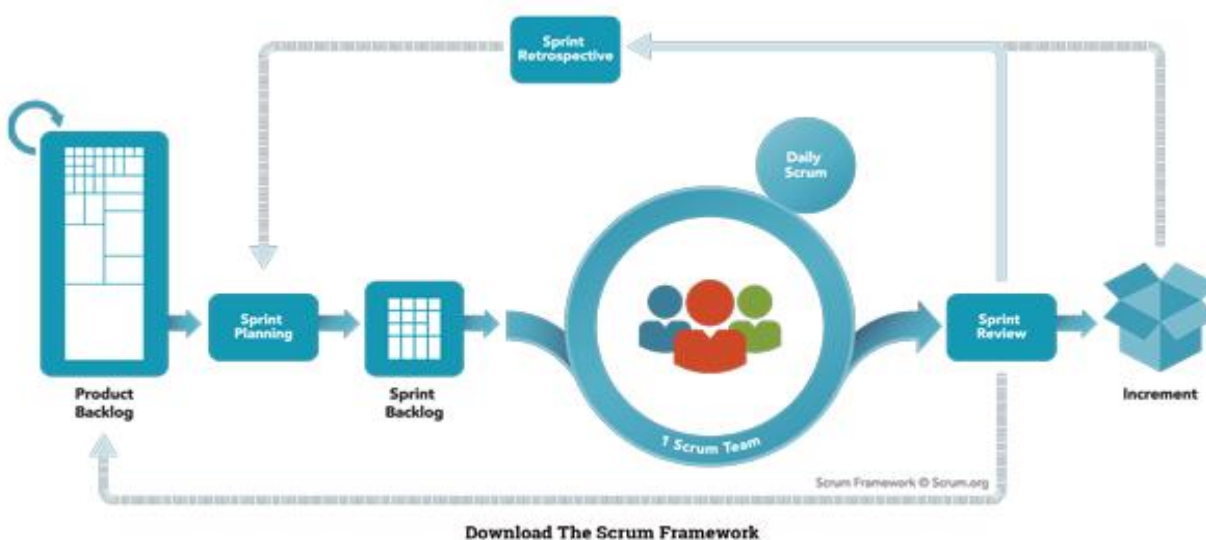


Figura 8 Scrum Framework

Fuente: <https://www.scrum.org/index.php/resources/what-is-scrum>

Este marco de trabajo se basa en tres pilares específicos:

Transparencia: todo debe ser visible para los miembros del proyecto, los cuales son sus responsables. Así, el trabajo resulta comprensible y todos conocen cuál es su definición de hecho.

Inspección: cada componente del proceso de Scrum debe ser inspeccionado para hallar cualquier posible falla a la cual darle solución de manera rápida.

Adaptación: permite que, de acuerdo con lo inspeccionado, se cambie o adapte el componente afectado, para satisfacer lo antes posible las necesidades del cliente.

Además, Scrum define los valores de compromiso, coraje, apertura y respeto.

2.6.5.1 *Roles.*

Scrum trabaja con un equipo de trabajo conformado por: Product Owner, Scrum Master y Equipo de Desarrollo.

Product Owner: es el encargado de la relación entre el cliente y el equipo de Scrum. Éste debe buscar en todo momento maximizar el valor del trabajo realizado por el equipo de Scrum. Por otra parte, esta persona está a cargo del “Product Backlog” (se explicará más adelante), el cual debe estar debidamente acomodado. Finalmente, es importante recalcar que no debe existir más de un Product Owner.

Scrum Master: es el responsable de asegurar el cumplimiento de las mejores prácticas de Scrum, por parte del equipo de trabajo y de colaborar con el Product Owner en la maximización del valor del proyecto. En un proyecto pueden existir varios Scrum Master en distintos equipos, llevando a cabo el mismo proyecto.

Equipo de desarrollo: este equipo está formado por quienes van a desarrollar el proyecto. Es importante mencionar que los procesos se atribuyen al equipo como tal y no se asigna a una única persona, pues el equipo requiere de tantas personas como sea necesario para cumplir con todos los ítems establecidos; es decir, el equipo es autosuficiente. Si alguna persona faltara, el equipo debe asegurar de cumplir con los objetivos de todas maneras.

2.6.5.2 *Eventos*

Dentro del marco de trabajo de Scrum, se dan distintos eventos para llevarse de a cabo manera adecuada.

2.6.5.2.1 *Sprint.*

El sprint es el período de tiempo dentro del cual se desarrolla un entregable parcial funcional. Aunque estos períodos de tiempo varían, normalmente requieren de una a cuatro semanas. En caso de ser necesario, el Product Owner tiene la posibilidad de cancelar el sprint; claro que la única razón válida para hacerlo sería que el objetivo del sprint ya no tenga sentido. En caso de que esto sucediera, se consideran los entregables completos para verificar si pueden ser un incremento, de lo contrario regresan al Backlog.

2.6.5.2.2 *Sprint Planning.*

El sprint planning es el proceso en el cual se seleccionan las historias de usuario que han de llevarse a cabo en el sprint venidero. En esta reunión deben estar presentes el Scum Master, Product Owner, cliente, equipo de trabajo y cualquier otra persona que con su juicio profesional colabore con el equipo durante el proceso de planeación. Este proceso exige un máximo de ocho horas. Además, el equipo de desarrollo define aquello a realizar durante el lapso definido del sprint. Es importante mencionar que no resulta necesario detallar en su totalidad todo cuanto se va a realizar; basta con mencionarlo y que tenga sentido para todos, para mantener la transparencia. A lo largo del sprint podrá detallarse la información pendiente.

2.6.5.2.3 *Daily Scrum Meeting.*

El Daily Scrum Meeting es una reunión que debe realizarse de pie durante un lapso máximo de quince minutos y debe estar presente el equipo de desarrollo y el Scrum Master, en caso necesario. En esta reunión se maneja el estado del avance del sprint empleando como guía tres preguntas: qué se hizo ayer, qué se hará hoy y si

existe algún impedimento. Todo esto con el fin de mantener la transparencia y eliminar cualquier obstáculo que retrase la entrega del incremento.

2.6.5.2.4 *Sprint Review.*

El Sprint Review es una reunión realizada al final del sprint, en la cual se revisa lo hecho durante el sprint con el fin determinar lo que se completó o no; será el Product Owner quien define si está completo o no, basándose en las historias de usuario. En esta reunión puede estar presente el cliente para verificar la funcionalidad de los entregables del sprint terminado. El tiempo límite para esta reunión son cuatro horas.

2.6.5.2.5 *Sprint Retrospective.*

Esta reunión la dirige el Scrum Master y es del interés único del Product Owner, equipo de desarrollo y el mismo Scrum Master. En ésta, se analiza lo hecho bien y cuánto se hizo mal durante el sprint con miras a mejorarlo en el sprint venidero. El límite de tiempo es de tres horas. Una vez identificadas las áreas por mejorar, se define un plan para implementarlas en el siguiente sprint y, de ser necesario, se redefine la definición de terminado. Con esto se logran cumplir dos de los tres pilares; a saber, la inspección y la adaptación.

2.6.5.2.6 *Grooming.*

Se trata de un evento fuera de los definidos anteriormente, pero que se lleva a cabo dentro del sprint. Se da con el fin de analizar y depurar los ítems dentro del Product Backlog, con el fin de que cada ítem sea preciso, transparente, coherente y dividido para, de modo que sea una tarea realizable durante un sprint.

2.6.5.3 *Artefactos.*

Scrum utiliza artefactos para mostrar la información del proyecto. Éstos son: Product Backlog, Sprint Backlog, Burn-down chart e incremento.

2.6.5.3.1 *Product Backlog.*

El Product Backlog es la lista de requerimientos, funciones y todo lo necesario del producto. Estos elementos vienen del levantamiento de requerimientos y son el uso de historias de usuario. “Los elementos de la Lista de Producto tienen como atributos la descripción, el orden, la estimación y el valor.” (Schwaber & Sutherland, 2017, p. 15). Los elementos del Product Backlog se acomodan de acuerdo con la necesidad, donde normalmente arriba se encuentran los elementos más claros y abajo los que están pendientes de más detalle.

2.6.5.3.2 *Sprint Backlog.*

El Sprint Backlog es el backlog que pertenece al sprint. En éste se listan los elementos del Product Backlog que se han de utilizar en el sprint. Esta lista se define como “...una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento “Terminado”.” (Schwaber & Sutherland, 2017, pp. 16-17). El equipo de desarrollo es el encargado de mantener este backlog y con ello definir qué incluir y qué remover del mismo.

2.6.5.3.3 *Incremento.*

El Incremento es la suma de todos los elementos de la Lista de Producto completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores. Al final de un Sprint, el nuevo Incremento debe estar “Terminado”, lo

cual significa que está en condiciones de ser utilizado y que cumple la Definición de “Terminado” del Equipo Scrum. (Schwaber & Sutherland, 2017, p. 17).

Este incremento es el proyecto como tal, lo que puede ver el cliente como objeto tangible del proceso desarrollado.

2.6.5.3.4 *Burn-down chart.*

Éste es un artefacto incluido de cierta manera dentro de los otros artefactos, ya que se utiliza para darle seguimiento al avance del sprint y determinar si la parte del esfuerzo planeado se cumple. El Product Owner lo utiliza principalmente para comparar el trabajo actual con el del sprints anteriores para asegurar el cumplimiento del proyecto

2.6.5.4 *Definición de terminado.*

Para que un elemento del sprint se defina como terminado, debe cumplir con la definición establecida por el equipo de trabajo. “Aunque esto puede variar significativamente para cada Equipo Scrum, los miembros del Equipo deben tener un entendimiento compartido de lo que significa que el trabajo esté completado para asegurar la transparencia.” (Schwaber & Sutherland, 2017, p. 18). Por ello, es esencial que esta definición concuerde desde un inicio para todos los miembros del equipo de desarrollo, Scrum Master y Product Owner.

2.7 **Herramientas para la gestión de proyectos**

2.7.1 **JIRA**

JIRA es una herramienta para la gestión de proyectos ágiles de la compañía Atlassian. Por sí misma, JIRA posee distintas funciones relacionadas con las

metodologías ágiles y en este caso aplica para el desarrollo de un sistema. Se trata de un software pago, pero que brinda pruebas gratuitas de siete días. Presenta tres opciones: “JIRA Software”, “JIRA Software + Documentation” y “JIRA Software + Help Desk”, cada uno con costos mensuales de entre diez y veinte dólares. Este software brinda las siguientes opciones:

Planificar: permite crear historias de usuario, planificar sprints y distribuir las tareas entre los miembros del equipo.

Supervisar: es posible monitorear, priorizar y analizar el trabajo del equipo con una visibilidad completa.

Lanzar: permite realizar lanzamientos, teniendo a mano la información más reciente y actualizada.

Crear informes: como parte importante de la gestión de proyectos, facilita la obtención de datos visuales para la toma de decisiones, en tiempo real.

Flujo de trabajo: permite utilizar flujos de trabajo predefinidos o crearlos personalizados con el fin de adaptarse al equipo de desarrollo.

Integración: permite la integración de herramientas ya utilizadas como gestores de conocimiento, flujos de desarrollo y verificación de estados de compilación sin salir del software.

En la figura 10 se presenta la interfaz de JIRA.

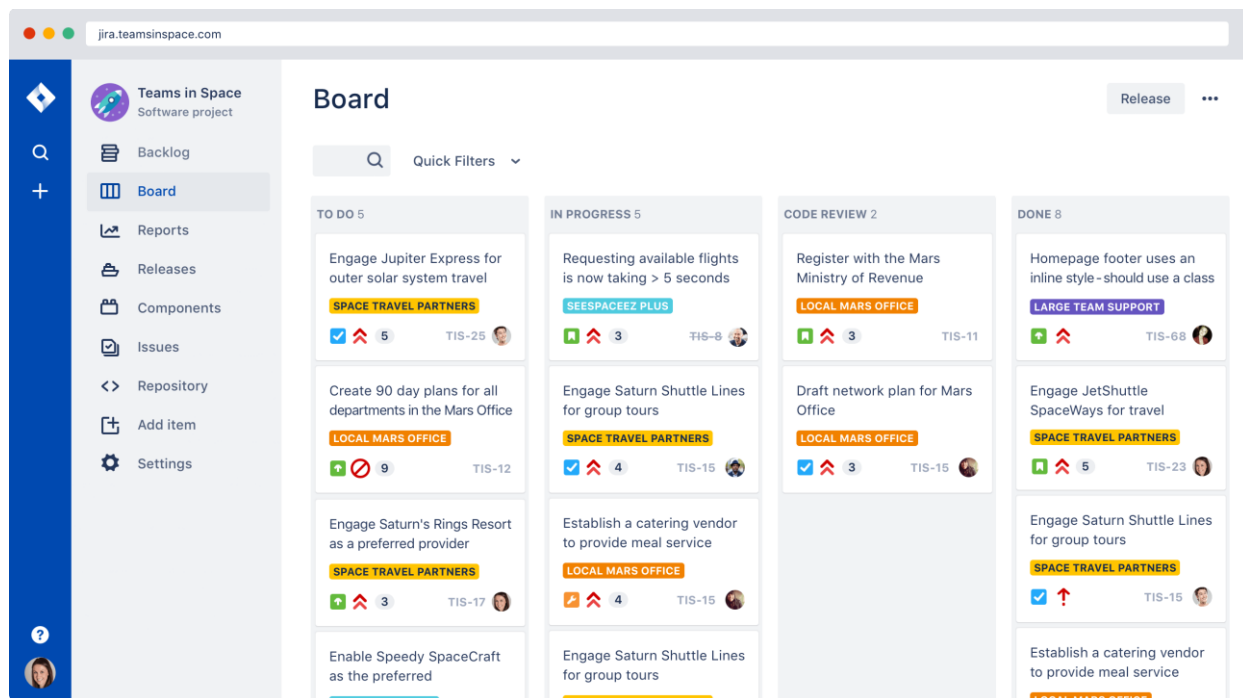


Figura 9: JIRA

Fuente: <https://es.atlassian.com/software/jira>

2.7.2 Microsoft Project.

Microsoft Project es una herramienta creada por Microsoft. Dicha herramienta cumple con la función de gestión de proyectos a nivel de paquetes de Office. La herramienta presenta opciones de manejo de plantillas de tipos de proyectos como sprints, cascada, planes de desarrollo de software, planes de negocio, etc. Además, brinda opciones de planeación de proyectos con extensiones, como gráficos de Grantt y menús previamente definidos para reducir los tiempos de entrenamiento y así simplificar el proceso de planeación de proyecto. Cuenta también con la opción para

crear reportes en tiempo real, para la toma de decisiones según la demanda. Además, ofrece opciones para la administración de carteras, recursos y otros.

No obstante, a pesar de sus múltiples beneficios, requiere de un gasto mensual de entre siete y cincuenta y cinco dólares por usuario, lo cual la convierte en una herramienta relativamente costosa a nivel de la nube. Para soluciones locales, los precios rondan entre los \$649.99 y \$1279.99 para sus versiones Estándar y Profesional. La figura 11 presenta la interfaz de usuario de Microsoft Project.

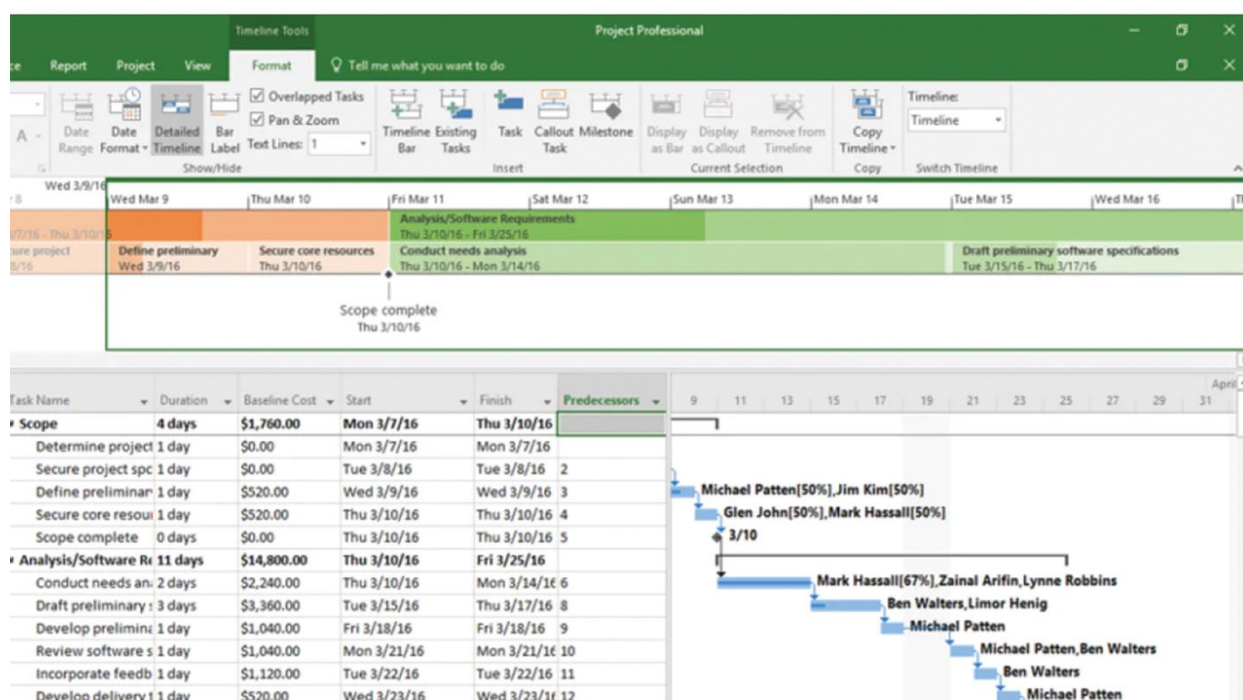


Figura 10: Microsoft Project

Fuente: <https://products.office.com/es/project/project-and-portfolio-management-software>

2.7.3 Azure DevOps.

Azure DevOps es una herramienta de Microsoft para control de proyectos. Esta herramienta permite trabajar con repositorios de proyectos programados con el uso de GitHub y Team Foundation. A su vez, Azure DevOps permite la gestión de proyectos de Capability Maturity Model Integration (CMMI), Agile, Scrum o básicos. Cual sea el tipo de proyecto seleccionado, se brindan las opciones de “boards” para controlar las historias de usuario, flujos de trabajo, repositorios, planes de pruebas, artefactos y las distintas extensiones existentes dentro de la tienda de Microsoft, donde muchas de las cuales son gratuitas.

En cuanto a su costo, puede decirse que tiene un precio económico, pues da la opción para el manejo de cinco cuentas con acceso completo a las funcionalidades, requiriendo de un pago por cuenta adicional. Su adaptación al Integrated Development Environment (IDE), Microsoft Visual Studio da un beneficio extra al facilitar el control de versionamiento para gestión del proyecto. La figura 12 muestra la interfaz de usuario de Azure DevOps.

Azure DevOps

Contoso / AdventureWorks Mobile / Pipelines / Builds

AdventureWorks Mobile

History Analytics

Commit	Build #	Branch	Date/time
Fix mdm encryption cert Robin Counts	10393	master	12:59 PM
Add mc cert - use rotatable secret for every iteration Robin Counts	10392	develop	10:32 AM
Cleanup auth handlers. Use handlers from the service Celeste Burton	10391	master	8:39 AM
Make kusto and admin apps optional Robin Counts	10390	feature/2-factor-auth	Yesterday, 2:45 PM
[Streaming][Bug] Closing stream before injecting Colin Ballinger	10389	master	Yesterday, 11:13 AM
Fix deployment arm param Colin Ballinger	10388	master	Tuesday, 1:49 PM
[Streaming] Handling oversized payload by finder Celeste Burton	10387	feature/some-feature	Monday, 2:13 PM
Fixing RDP bug Kat Larsson	10386	master	Jul 6, 4:23 PM
Fix KV URL resolution Robin Counts	10385	hotfix/view-perf	Jul 5, 1:43 PM
Fix debug console output when running unit tests Robin Counts	10384	master	Jul 3, 9:48 AM
Fix mdm encryption cert Robin Counts	10383	master	Jul 1, 11:31 AM

Project settings

Figura 11: Azure DevOps

Fuente: <https://azure.microsoft.com/en-us/services/devops/>

Capítulo III

Marco Metodológico

3.1 Tipo de investigación

3.1.1 Tipo de investigación.

Esta investigación se basa en el tipo aplicado, el cual "...se caracteriza porque busca la aplicación o utilización de los conocimientos adquiridos, a la vez que se adquieren otros, después de implementar y sistematizar la práctica basada en investigación." (Cordero, 2009, p. 159). Así las cosas, la investigación aplicada se emplea aquí con el propósito de poner en práctica los conceptos adquiridos relativos a la gestión de proyectos y al uso de metodologías ágiles, óptimas para mejorar el estado de la situación tecnológica de un centro educativo, específicamente en cuanto al empleo de sistemas de información.

3.1.2 Enfoque de la investigación.

El enfoque de la investigación será cualitativo, lo cual quiere decir que: "se selecciona cuando el propósito es examinar la forma en que los individuos perciben y experimentan los fenómenos que los rodean, profundizando en sus puntos de vista, interpretaciones y significados" (Hernández Sampieri, Roberto, Fernández Collado, Carlos, & Baptista Lucio, María del Pilar, 2014, p. 358).

Por otra parte, este enfoque facilita la construcción de nuevos conocimientos, pues posibilita la observancia del comportamiento de los participantes y la forma en que se desarrolla la conducta. Finalmente, son las acciones propias de la investigación - comprender, examinar y desarrollar- las que permiten la utilización de este enfoque de investigación (Hernández Sampieri, Roberto et al., 2014, p. 358).

3.2 Fuentes de información

3.2.1 Fuentes de información primaria

Las fuentes de información primaria para este proyecto serán los resultados del uso de herramientas de recolección de datos; todo aquel material brindado por entes oficiales de certificación del proceso de Scrum Scrum.org para llevar la ejecución con las mejores prácticas de dicho marco de trabajo y del conocimiento como Professional Scrum Master I (ver Anexo 1).

3.2.2 Fuentes secundarias

Las fuentes secundarias de información son todas aquellas publicaciones de revistas tecnológicas, tesis, libros, revistas académicas, reglamentos y procedimientos académicos de Kamuk School y todo aquel material respaldado por un autor definido, que colabore con el proceso de la realización del proyecto.

3.2.3 Sujetos de información.

Como sujetos de información, se tendrán tres personas de la institución:

Tabla 2.
Sujetos de información

Puesto Laboral	Profesión	Experiencia	Relación con el tema
Gerente de operaciones	Encargado de las operaciones institucionales y	2 años	Fuente de información de la realidad institucional y

Puesto Laboral	Profesión	Experiencia	Relación con el tema
	voz en la toma de decisiones		elaboración de propuestas de mejoras institucionales
Coordinadora de F.E. y Comunicación	Encargada del software académico y procesos académicos de la institución	8 años	Fuente de información del software académico institucional y los procesos internos de planeación y ejecución de curso lectivo
Encargado TI	Encargado de la tecnología de la institución, tanto a nivel de hardware como software	2 años y medio	Fuente de datos técnicos a utilizar durante la propuesta, así como punto de contacto para las reuniones con el equipo de trabajo

Fuente: Equipo de proyecto de KAMEDCR

3.3 Técnicas y herramientas de recolección de datos

Según Sampieri, la recolección de datos consiste en crear un plan detallado de cuanto se va a hacer para recolectar datos con un propósito específico. (Hernández

Sampieri, Roberto et al., 2014). Para esta investigación se utilizarán cuatro tipos de herramientas: entrevistas, encuestas de tipo cuestionario, observación y documentos existentes.

3.3.1 Entrevista.

La entrevista "...consiste en una especie de conversación con el participante o grupo de participantes..."(Pazos Jiménez, Ethel & Gutiérrez M., Federico, 2012, p. 81).

Esta herramienta puede darse de tres maneras:

Estructurada: se realizan preguntas concretas.

Semiestructurada: se formulan preguntas fuera de las previstas.

No estructurada: se limitan las preguntas de acuerdo con la forma en que se desarrolle la entrevista como tal.

En esta investigación se utilizará la entrevista semiestructurada.

3.3.2 Cuestionario

El cuestionario se define como el "conjunto de preguntas respecto de una o más variables que se van a medir."(Hernández Sampieri, Roberto et al., 2014, p. 217). Los cuestionarios pueden construir con dos tipos de preguntas: abiertas y cerradas. Las preguntas abiertas consisten en preguntas que "no delimitan de antemano las alternativas de respuesta, por lo cual el número de categorías de respuesta es muy elevado..." (Hernández Sampieri, Roberto et al., 2014, p. 220). Por otra parte, las preguntas cerradas son aquellas que delimitan las posibles respuestas de la persona

consultada. En esta investigación se utilizarán ambos tipos de preguntas de acuerdo con el área por tratar.

3.3.3 Observación

Este instrumento consiste en “...el registro sistemático, válido y confiable de comportamientos y situaciones observables, a través de un conjunto de categorías y subcategorías.” (Hernández Sampieri, Roberto et al., 2014, p. 252). Este tipo de recolección de datos se utiliza con el fin de comprender realmente las necesidades académicas para realizar una propuesta acorde con el comportamiento del centro educativo.

3.3.4 Documentos existentes

El último instrumento por utilizar es la información existente en el centro educativo. La institución cuenta con documentación definida, tal como su manual de marca, reglamento de la institución y otros recursos igualmente valiosos.

3.4 Variables

Tabla 1.
Variables de la investigación

Objetivo específico	Variables asociadas	Descripción
Analizar la situación actual de Kamuk School con respecto al uso de sistemas académicos y gestión académica durante el año 2019, para	Aceptación de sistemas de información, capacidades tecnológicas de la institución, cambio de gestión académica y las	Consiste en una observación detallada del uso de sistemas académicos (en su funcionalidad como sistema de información),

Objetivo específico	Variables asociadas	Descripción
desarrollar una propuesta de sistema académico afín a la institución.	políticas de trabajo institucionales.	del proceso y de la gestión académica de la institución para comprender sus necesidades.
Presentar un entregable final de la propuesta de desarrollo de los módulos de administración general y profesores de un sistema académico hecho a la medida para Kamuk School durante el año 2019.	Requerimientos institucionales y aceptación por parte de la institución	De acuerdo con las necesidades observadas en los procesos anteriores, se generará una propuesta de sistema académico para la institución
Administrar los tiempos de entrega durante el desarrollo de los módulos de administración general y profesores de un sistema académico hecho a la medida para Kamuk School durante el 2019, utilizando las mejores prácticas de metodologías ágiles.	Herramienta de gestión de proyectos y disponibilidad del equipo de trabajo	Consiste en la ejecución del proyecto siguiendo las prácticas de la metodología utilizada en la propuesta y así controlar los tiempos de muestra de entregables parciales
Mantener la comunicación y trabajo con el cliente durante el desarrollo de los módulos de administración general y profesores de un sistema académico hecho	Herramienta de gestión de proyectos y disponibilidad de la institución.	Mantener una constante comunicación con la institución durante el desarrollo de la propuesta, y de esta manera cumplir con las necesidades del

Objetivo específico	Variables asociadas	Descripción
a la medida en Kamuk School durante el año 2019, utilizando las mejores prácticas de metodologías ágiles.		cliente, manteniendo a la vista los avances del entregable final.

Fuente: Creación propia



Figura 13: Diseño de la investigación

Fuente: Creación propia

3.5 Diseño de la investigación

3.5.1 Análisis.

En esta etapa se observa y obtienen datos de la institución a nivel de uso de sistemas académicos como la serie de procesos guía del trabajo educativo. En esta etapa prevalece el empleo de instrumentos como las entrevistas, cuestionarios y observación. Una vez obtenida la información no escrita de la institución mediante los instrumentos, se procede utilizar los documentos institucionales para conocer la situación actual. Finalmente, se detallan las necesidades de la institución y el problema por trabajar durante la ejecución del proyecto.

3.5.2 Diseño.

De acuerdo con los datos obtenidos a partir del análisis y en conjunto con el equipo de trabajo, se define la metodología por utilizar para el desarrollo y la definición de tiempos de trabajo. Por último, se presenta a la institución el plan de trabajo, para obtener un acuerdo con lo que se ha de desarrollar durante la ejecución.

3.5.3 Desarrollo.

Esta es la etapa en donde se lleva a la práctica lo establecido en la etapa de diseño. Esta fase requiere del trabajo del resto de sujetos de información, a saber: arquitecto, administrador de base de datos, analista de calidad y desarrolladores. Durante la ejecución se emplea la metodología definida junto con las herramientas disponibles para abarcar todas las necesidades durante el proceso. Aquí se requiere de documentación precisa acerca de los avances realizados, así como de la presentación de entregables funcionales. Cabe mencionar que todo entregable funcional debe pasar por el proceso de pruebas de calidad para asegurar el cumplimiento de cada requerimiento establecido.

3.5.4 Entrega.

Esta última etapa consiste en la entrega del proyecto final a la institución. En este momento, se entrega el manual de usuario de la propuesta y las recomendaciones resultantes del desarrollo de esta. Finalmente, esta entrega debe ser aprobada por los representantes de la institución y su implementación depende de la decisión tomada por ellos.

Tabla 2.
Matriz de coherencia

Objetivo	Entregable	Fase	Técnicas o métodos de recolección de información	Instrumentos	Temas relacionados con el marco teórico
1 Analizar la situación actual de Kamuk School con respecto al uso de sistemas académicos y gestión académica durante el año 2019, para desarrollar una propuesta de sistema académico a fin a la institución.	Requerimientos para la propuesta	Análisis	Entrevista, cuestionario, observación y documentos institucionales	Entrevista tecnológica, académica y de percepción	Sistema y sistema de información
2 Presentar un entregable final de la propuesta de desarrollo de los módulos de administración general y profesores de un sistema académico hecho a la medida en Kamuk School durante el año 2019.	Propuesta de desarrollo funcional con acceso a través de la red	Entrega	Reunión con el cliente y el equipo de trabajo		Software, desarrollo de software, metodologías de desarrollo y metodologías ágiles de desarrollo

Objetivo	Entregable	Fase	Técnicas o métodos de recolección de información	Instrumentos	Temas relacionados con el marco teórico
3 Administrar los tiempos de entrega durante el desarrollo de los módulos de administración general y profesores de un sistema académico hecho a la medida para Kamuk School durante el año 2019, utilizando las mejores prácticas de metodologías ágiles.	Documentación de los tiempos de trabajo y entregables parciales	Diseño y desarrollo	Reuniones con el equipo del trabajo y gráficos de avance del proyecto		Herramientas para la gestión de proyectos
4 Mantener la comunicación y trabajo con el cliente durante el desarrollo de los módulos de administración general y profesores de un sistema académico hecho a la medida en Kamuk School durante el año 2019 utilizando las mejores prácticas de metodologías ágiles.	Documentación de las comunicaciones con la institución	Diseño y desarrollo	Reunión con el cliente		Software, desarrollo de software, metodologías de desarrollo y metodologías ágiles de desarrollo

Fuente: Creación propia

Capítulo IV

Diagnóstico

El diagnóstico institucional se realizó con base en la realidad de conocimiento de las fuentes primarias de información según su área. De igual manera, se realizó un diagnóstico de percepción general, en el cual los distintos usuarios del sistema académico proporcionaron su opinión en relación con el empleo de este. Los cuestionarios utilizados se encuentran en los anexos 2, 3 y 4 respectivamente.

4.1 Diagnóstico administrativo

4.1.1 Resultados del instrumento.

El diagnóstico administrativo se centra en el trabajo y necesidades académicas del público meta de la institución. Se elabora el cuestionario para la Coordinadora de F.E. y Comunicación, encargada de los sistemas académicos y del enlace de comunicación entre los padres de familia y la institución. Finalmente, como coordinadora ha estado presente en los procesos de uso de SIRA y Class.

A continuación se presentan los resultados del cuestionario:

Fecha de realización: 23/5/2019

1) ¿Cuál es la importancia del control académico de los estudiantes?

Respuesta: “un sistema de control académico virtual permite al padre tener información en tiempo real y una comunicación más expedita entre institución-hogar”

2) ¿Cuál es la importancia de la planeación institucional anual?

Respuesta: “si se refiere a la planificación académica, ésta es muy importante, porque permite dosificar adecuadamente todos los contenidos que deben ser cubiertos en un periodo específico”

3) ¿Por qué se tiene la necesidad de un software académico?

Respuesta: “para mejorar la comunicación institución-hogar y viceversa, sin necesidad de estar desplazándose al sitio. La comunicación oportuna con los padres de familia reduce las probabilidades de que un alumno pierda el año académico y se puede ofrecer mejor contención (hogar-institución) cuando el estudiante lo requiera, ya que la comunicación por medios virtuales nos permite más cercanía para con el padre de familia.”

4) ¿Cuáles son los beneficios de contar con un software académico para Kamuk School?

Respuesta: “Entrega del rendimiento académico en tiempo real, comunicación continua hogar-institución, control de la correspondencia docente-padre, docente-estudiante, así como el control de actividades académicas, mediante un calendario virtual.”

5) ¿Qué beneficios tenía el sistema académico SIRA para Kamuk School?

Respuesta: “Todo lo descrito en la pregunta anterior: Entrega del rendimiento académico en tiempo real, la comunicación continua hogar- institución, control de la correspondencia docente-padre, docente-estudiante, así como el control de actividades académicas mediante calendario virtual”

6) ¿De qué funcionalidades carecía el sistema académico SIRA en Kamuk School?

Respuesta: “sistema de control cruzado de justificaciones, o sea, que el sistema, de forma automatizada, anotara en el registro docente la ausencia justificada y de esta forma el profesor sabría que ya el padre había notificado la falta de asistencia del estudiante a la persona encargada de justificar. Además no estaba desarrollada la parte contable”

7) ¿Cuál es el estado actual de SIRA con respecto a Kamuk School?

Respuesta: “el sistema está sin usarse, ya que contratamos otro”

8) ¿Qué beneficios tiene el sistema académico Class para Kamuk School?

Respuesta: “integración académica y financiera”

9) ¿De qué funcionalidades carece el sistema académico Class?

Respuesta: “el sistema es muy complicado para administrar la plataforma y además solo funciona en internet explorer, no es la misma plataforma que usan los docentes, no se puede manipular desde dispositivos móviles”

10) ¿Qué características debe cumplir un sistema académico para Kamuk School?

Respuesta: “los descritos en la pregunta de cuáles son los beneficios de un software académico, de que estén integrados a una plataforma contable y que esté apto para navegar desde dispositivos móviles”

4.1.2 Estado actual.

Actualmente el centro educativo se encuentra en un estado de necesidad continua por la deficiencia del cumplimiento de requerimientos educativos. Un sistema

académico como SIRA le satisface a Kamuk School la mayoría de las necesidades básicas; sin embargo, traía consigo ciertos problemas de tecnología como: lenguajes de programación antiguos, uso de interfaces ya no aptas para los navegadores, además de otras brechas a nivel tecnológico. Asimismo, la falta de integración con áreas como la de finanzas crea la necesidad inmediata de un cambio. Una vez identificada la necesidad de implementar otro tipo de software, Kamuk School decide implementar Class, otro sistema académico que le permite integrar el proceso académico con el contable.

A pesar de trabajar de buena manera e integrarse como se esperaba, poco a poco el nuevo sistema empieza a mostrar deficiencias como lo son estrecha capacidad de trabajo con diferentes exploradores de internet, poca adaptación a los cambios educativos por tratarse de una institución en constante modernización para mejorar sus servicios académicos y deficiencias en usabilidad. En general, la falta de cumplimiento de requerimientos coloca a Kamuk School en una posición delicada en cuanto al uso de sistemas de información, dado que estos constituyen un tipo de herramienta de vital importancia ante un acelerado proceso de globalización, donde los dispositivos tecnológicos constituyen una herramienta para obtener información precisa en tiempo con miras a facilitar la vida de cada usuario.

Finalmente, la constante inversión en este tipo de sistemas motiva a Kamuk School a buscar la implementación de un sistema académico hecho a la medida, con el fin de reducir el costo anual por mantenimiento y poder satisfacer cada una de sus necesidades, mediante un análisis adecuado del funcionamiento institucional.

4.1.3 Estado óptimo.

El estado óptimo de un sistema académico en Kamuk School consiste en un desarrollo que cumpla con los procesos educativos definidos en el reglamento educativo de la institución. Lo anterior unido a la facilidad de acceso de padres de familia y estudiantes para mantener una comunicación a tiempo real con la institución para así lograr que los estudiantes logren alcanzar sus metas educativas y maximizar su nivel académico para beneficio personal e institucional.

Asimismo, dicho sistema de estar listo para adaptación de cambios en la evaluación estudiantil y finalmente estar totalmente enfocado a su facilidad de uso en las distintas plataformas a las cuales tendrán acceso sus usuarios: padres de familia, estudiantes y funcionarios de Kamuk School.

4.1.4 Brecha.

La institución debe contar con un sistema académico que cumpla con las necesidades establecidas en los instrumentos de información de la institución, es decir con los reglamentos y procesos institucionales.

4.2 Diagnóstico técnico

4.2.1 Resultados de Instrumento

El diagnóstico técnico se centrará en la obtención de datos técnicos a nivel institucional, así como sus prácticas de control de proyectos y guía de trabajo de distintas áreas. Se elabora el cuestionario al Encargado de TI ya que es el encargado

de la información tecnológica para la institución y encargado y medio de contacto para el desarrollo de este proyecto. A continuación, los resultados del cuestionario:

Fecha de realización: 27/5/2019

1. ¿Cuenta Kamuk School con alguna guía para contratación o compra de servicios de tecnológicos? En caso afirmativo, indique cuál.

Respuesta: “No”

2. ¿Cuáles son las características de hardware y software del servidor en el que se podría implementar la propuesta del software académico?

Respuesta: “HP ProDesk 400 G3 MT Business PC, windows 10”

3. ¿Cuenta Kamuk con licenciamiento de base de datos? De ser así, indique cuál y qué herramienta se utiliza.

Respuesta: “No”

4. ¿Cuenta Kamuk con licenciamiento para plataformas Windows (Windows Server)? De ser así, especifique con mayor detalle.

Respuesta: “No”

5. ¿Se cuenta con contrato de algún VPS o cualquier otro tipo de servicio en línea? De ser así, especifique con mayor detalle.

Respuesta: “Tenemos un server con el proveedor HosGator, 1 TB de memoria y con MySQL para gestionar bases de datos.”

6. ¿Cuenta Kamuk School con alguna guía de gestión de proyectos tecnológicos? En caso de contar con alguna, indique cuál.

Respuesta: “No”

7. ¿Cuenta Kamuk School con alguna guía de estándares de calidad? En caso de contar con alguna, indique cuál.

Respuesta: “No”

4.2.2 Estado Actual.

El estado tecnológico actual de la institución enfrenta el reto de implementar muchas mejoras. A pesar de ser un centro educativo, la planeación tecnológica adquiere gran importancia, sobre todo por el hecho de que se depende de ello en gran medida y parte de su presupuesto está dispuesto para esto. Actualmente la institución carece de planeación o guías de compras de servicios tecnológicos, licenciamiento local para sus sistemas, gestión de proyectos tecnológicos o estándares de calidad para el uso de cualquier sistema académico o de cualquier otro tipo. Asimismo, carece de una infraestructura local adecuada para la instalación de una aplicación local, ni siquiera se cuenta con un servidor con capacidades físicas adecuadas para soportar las cargas de las transacciones.

Finalmente, a nivel tecnológico se cuenta con el servicio de un servidor con Hostgator, el cual posee las características más básicas del proveedor para manejar bases de datos, lo cual no implica necesariamente que puede cumplir con las

necesidades reales, ya que no se tiene claro un proceso para la contratación de servicios.

4.2.3 Estado Óptimo.

A nivel institucional se debe contar con guías de planificación, gestión de proyectos, gestión de calidad y de otros aspectos como el conocimiento de infraestructura necesaria para satisfacer sus necesidades, de manera que sea posible la adquisición de cualquier servicio tecnológico de uso diario. De esta manera, se asegura que la inversión por realizar no afectará negativamente a la institución en el futuro al causar gastos innecesarios por cambios, deficiencias con el servicio y, principalmente, por incumplimiento en la satisfacción de las necesidades del centro educativo.

4.2.4 Brecha.

La institución debe contar con un proceso definido de gestión de proyectos, contratación de servicios y control de calidad para asegurar un eficiente empleo de tecnologías. Además, se debe contar con una infraestructura debidamente planificada para implementar desarrollos locales o, dado el caso, una documentación y contratación de servicios precisa, que defina lo que se necesita para mantener sistemas en línea.

4.3 Diagnóstico de percepción

4.3.1 Resultados de Instrumento.

El diagnóstico de percepción se centrará en los usuarios de los sistemas académicos sin importar su labor en la institución, ya que el sistema académico se emplea en todos los niveles, desde el administrativo hasta el de los profesores. A los estudiantes y padres de familia no se les incluye debido a que los módulos de la propuesta son administrativo y profesores. De igual manera, los padres y estudiantes se adaptan ajustan a las disposiciones institucionales.

Dada la realidad del trabajo institucional, la Coordinadora de F.E. y Comunicación envió el formulario virtual a los funcionarios, tomando en consideración a preescolar, primaria, secundaria y administración, para un total de diecinueve resultados. La Coordinadora ni el encargado de TI llenaron los formularios para asegurar la objetividad de resultados. A continuación, los resultados del cuestionario:

Fechas de realización: 24/5/2019 y 27/5/2019

1) ¿Qué función desempeña usted en la institución?

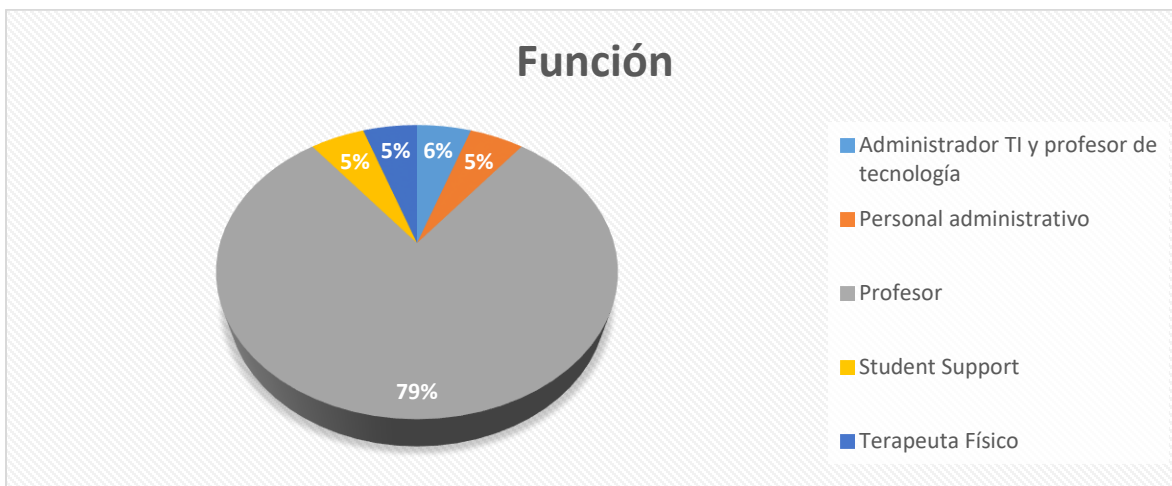


Figura 12: ¿Qué función desempeña usted en la institución

Fuente: Funcionarios Kamuk School (2019)

2) ¿Trabajó con el Sistema SIRA? Si contesta no, favor pasar a la pregunta 6.

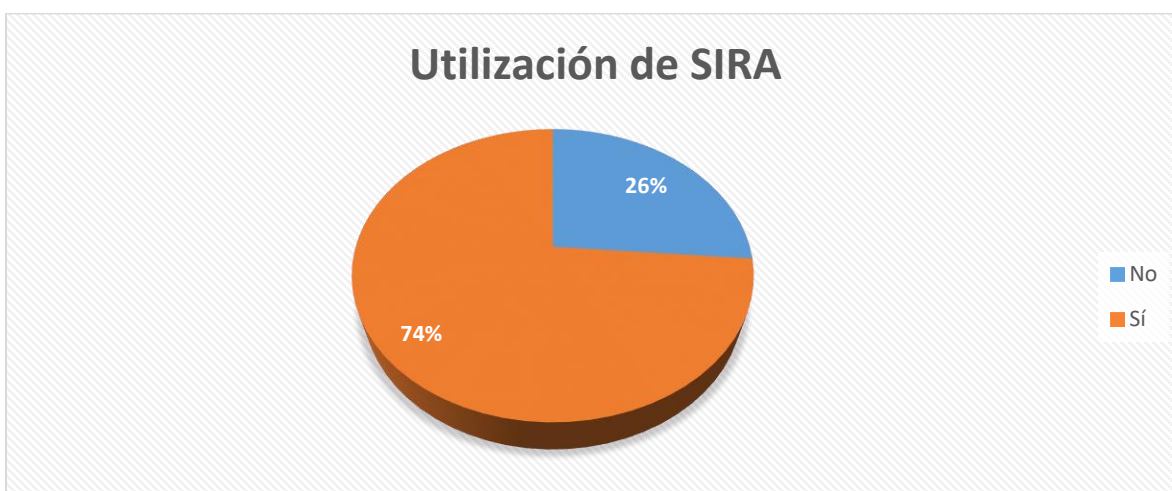


Figura 13: ¿Trabajó con el Sistema SIRA?

Fuente: Funcionarios Kamuk School (2019)

3) En una escala del uno al cinco, ¿cómo calificaría SIRA?

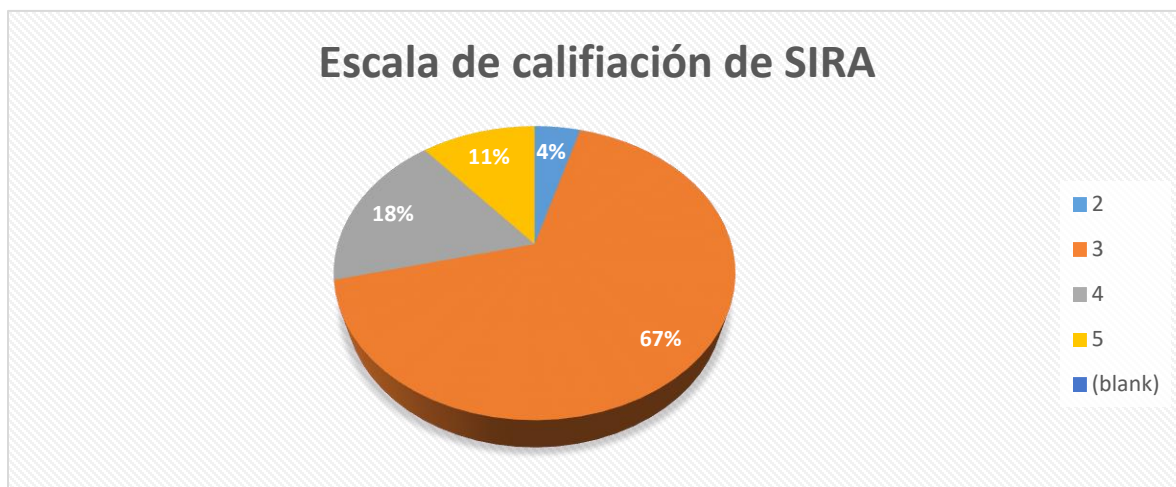


Figura 14: En una escala del uno al cinco, ¿cómo calificaría SIRA?

Fuente: Funcionarios Kamuk School (2019)

4) En una escala del uno al cinco, ¿qué tanto se podría mejorar la funcionalidad del sistema SIRA?

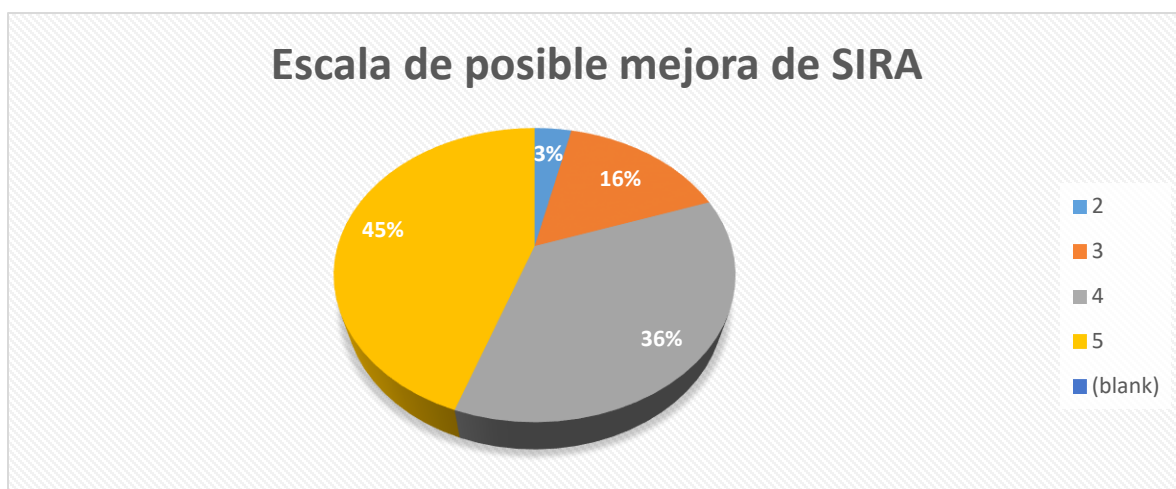


Figura 15: En una escala del uno al cinco, ¿qué tanto se podría mejorar la funcionalidad del sistema SIRA?

Fuente: Funcionarios Kamuk School (2019)

- 5) ¿Qué sistema académico considera de mayor utilidad y eficiencia, SIRA o Class?

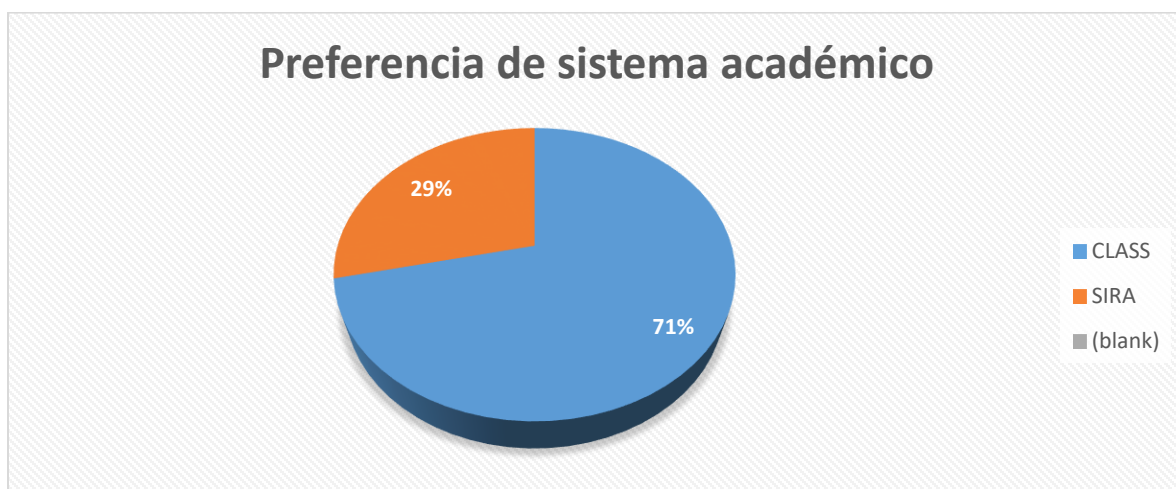


Figura 16: ¿Qué sistema académico considera de mayor utilidad y eficiencia, SIRA o Class?

Fuente: Funcionarios Kamuk School (2019)

- 6) En una escala del uno al cinco, ¿cómo calificaría el sistema Class?

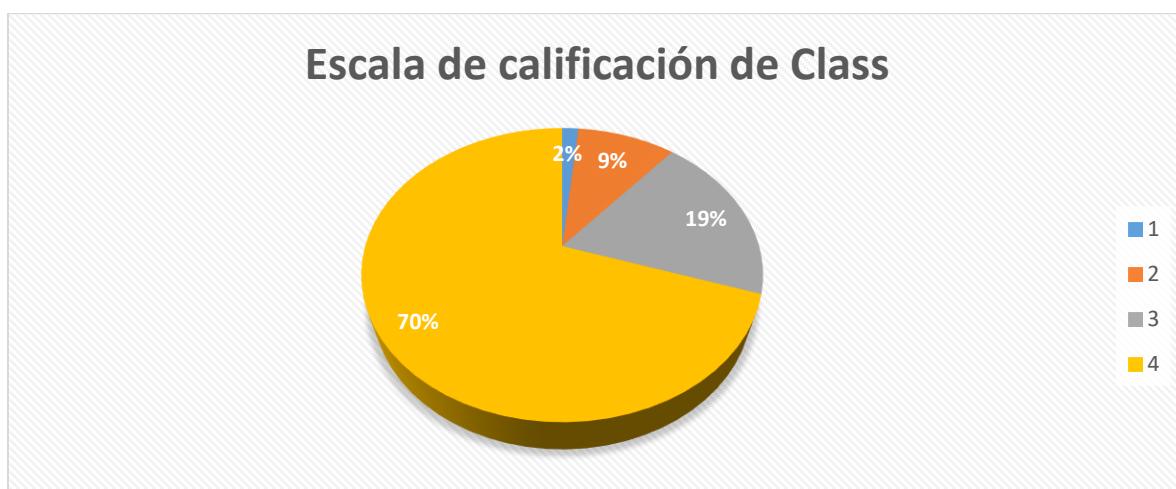


Figura 17: En una escala del uno al cinco, ¿cómo calificaría el sistema Class?

Fuente: Funcionarios Kamuk School (2019)

- 7) En una escala del uno al cinco, ¿qué tanto considera usted que podría mejorar la funcionalidad del sistema Class?

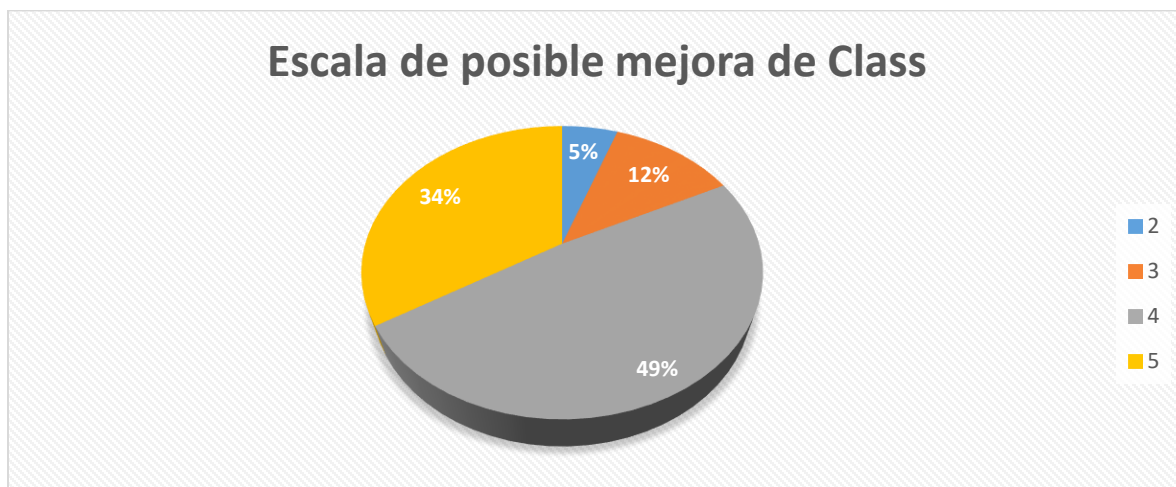


Figura 18: En una escala del uno al cinco, ¿qué tanto considera usted que podría mejorar la funcionalidad del sistema Class?

Fuente: Funcionarios Kamuk School (2019)

- 8) ¿Utilizaría un nuevo sistema académico, que combine las mejores funcionalidades de SIRA y Class? Omitir esta pregunta si nunca trabajó con el sistema académico SIRA.

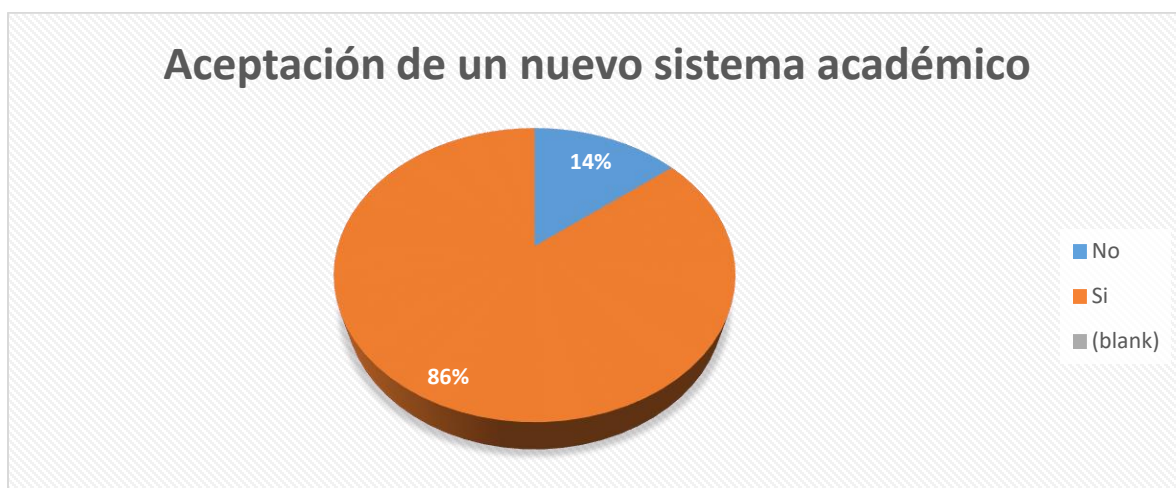


Figura 19: ¿Utilizaría un nuevo sistema académico que combine las mejores funcionalidades de SIRA y Class?

Fuente: Funcionarios Kamuk School (2019)

9) ¿Cuáles serían las tres principales características con las que debería contar un sistema académico?



Figura 20: ¿Cuáles serían las tres principales características con las que debería contar un sistema académico?

Fuente: Funcionarios Kamuk School (2019)

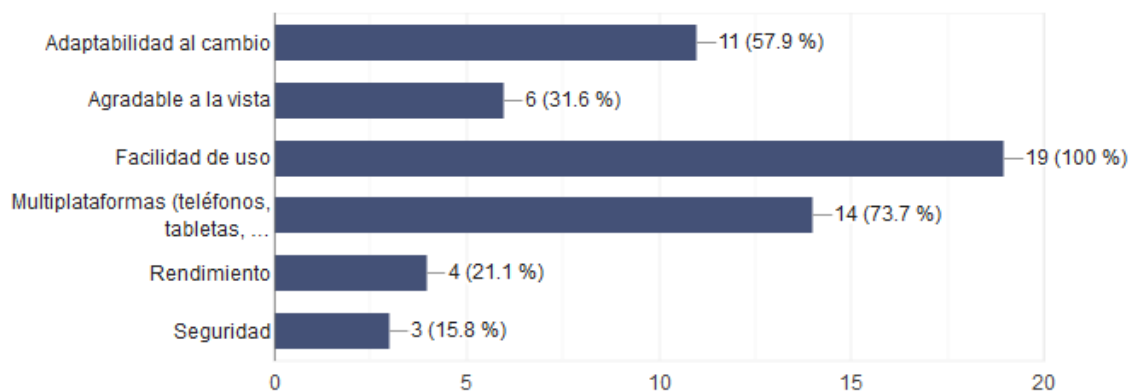


Figura 21: ¿Cuáles serían las tres principales características con las que debería contar un sistema académico?

Fuente: Funcionarios Kamuk School (2019)

10) ¿Colaboraría con la prueba de un nuevo sistema académico para beneficio del centro educativo y los estudiantes?



Figura 22: ¿Colaboraría con la prueba de un nuevo sistema académico para beneficio del centro educativo y los estudiantes?

Fuente: Funcionarios Kamuk School (2019)

4.3.2 Estado Actual.

La percepción actual de los funcionarios de la institución evidencia un alto grado de aceptación de los sistemas académicos utilizados. A pesar de ello, se muestra una clara identificación de debilidades de los sistemas, por no cumplir con las expectativas esperadas. Ambos sistemas académicos presentan fortalezas y debilidades que pueden utilizarse en el sistema académico a la medida para Kamuk School. De acuerdo con los datos suministrados, la gran mayoría de funcionarios estarían anuentes a probar un nuevo sistema académico siempre y cuando este prometa la satisfacción de las necesidades básicas, tales como: facilidad de uso, utilización en múltiples plataformas, adaptación al cambio, agradable a la vista, con buen rendimiento y seguridad.

Todo lo anterior fue alcanzado según lo planeado, a partir del análisis y la recolección de datos de las fortalezas de SIRA y Class, para maximizar sus fortalezas y sustentar sus debilidades a través de un nuevo desarrollo.

4.3.3 Estado Óptimo.

Este estado se alcanza cuando se cuenta con un sistema académico adecuado, que cumple con las necesidades de sus usuarios, tomando en consideración su opinión y la experiencia adquirida durante el proceso de implementación de sistemas académicos anteriores. El cumplimiento de las necesidades debe ser alcanzado gracias a una adecuada recolección de datos acerca de los requerimientos y de constantes pruebas del desarrollo hecho a la medida para asegurar su aceptación.

4.3.4 Brecha.

La institución debe tener un sistema académico afín a los usuarios, para asegurar que su implementación se adapte y cumpla con la satisfacción de sus necesidades diarias, sin enfocarse únicamente en el cumplimiento de las necesidades institucionales.

4.4 Diagnóstico general

La institución ya ha establecido en su reglamento institucional aquellas necesidades básicas que no se cumplen o no tomadas en consideración por los sistemas SIRA o Class. La experiencia ha llevado al personal de la institución a comprender que un desarrollo hecho a la medida o el uso de sistemas existentes no llenan sus expectativas por completo, dada la falta de procesos que permitan conocer lo que se ocupa, cómo se ocupa y cuándo se ocupa. De igual manera, las pruebas parciales de implementación son inexistentes y por ello la identificación de errores, necesidades sin cumplir o presencia de elementos innecesarios ocurre hasta el momento mismo en que un sistema se está totalmente implementado.

La carencia de procesos relacionados con las guías o documentación de desarrollo, compra o alquiler de servicios tecnológicos, definición de calidad y muchos otros elementos conduce a un proceso de adquisición de tecnologías muy subjetivo, lo cual da como resultado un éxito parcial. Dicho resultado se evidencia en el nivel de satisfacción de los usuarios de los sistemas, al estar conformes con lo existente aunque, a pesar de ello, encuentran fácilmente las debilidades de las herramientas de uso diario.

Así las cosas, resulta evidente que el proceso tecnológico institucional presenta muchas áreas de mejora, que una buena planificación puede solucionar.

La experiencia Kamuk School con sistemas no ha sido la mejor, razón por la cual se encuentra en búsqueda de un sistema académico que cumpla las necesidades de la institución, con el propósito de evitar otros gastos de implementación, de desarrollo de componentes faltantes o de procesos legales, como sucede actualmente con SIRA, por carecer de su código fuente a pesar de ser de su propiedad.

La falta de gestión de proyectos definida es crítica y por ello un desarrollo hecho a la medida debe contar con un análisis adecuado acerca de la situación actual, un levantamiento de requerimientos basado en la experiencia, documentación institucional, un desarrollo de la mano con la institución que asegure el agrado de los usuarios, una documentación precisa sin exceso que permita tener al alcance lo realizado y un proceso de calidad definido, que asegure el cumplimiento de requerimientos, de modo que sea posible contar con un sistema académico adecuado a las distintas necesidades institucionales.

Capítulo V

Diseño y desarrollo del proyecto

5.1 Desarrollo de propuesta

A partir de los resultados obtenidos en el capítulo IV, se procede a la ejecución de la propuesta de desarrollo para el proyecto. El desarrollo consiste en la elaboración de un sistema académico hecho a la medida para la institución Kamuk School. Dicho sistema constará de los dos primeros módulos, siendo éstos Administración y Profesores. La propuesta consta de tres partes: diseño, desarrollo y entrega.

5.1.1 Diseño.

5.1.1.1 Levantamiento de requerimientos Kamuk School.

Para elaborar un sistema académico a la medida, es necesario levantar los requerimientos o historias de usuario, base del desarrollo del equipo de trabajo. De ahí que se realizó un levantamiento de requerimientos con una miembro del personal de la institución y se acuerda lo siguiente:

Fecha de la reunión: 19 de marzo del 2019

Lugar: Kamuk School

Participantes:

- Evander Vallecillo Rizo (Encargado TI – Kamuk School)
- Mario Angulo Cordero (Estudiantes – Universidad Hispanoamericana)

Temas tratados:

- A. Requerimientos del sistema.
- B. Aspectos técnicos.

Acuerdos:

#	Descripción del acuerdo	Responsable	Fecha requerida
1.	Iniciar con el proyecto	Mario Angulo	-

Se informará a:

- Diego Saravia.
- Daniel Valle.
- Robert Herrera.
- Iván Duarte.
- Alejandro Chavarría.

Detalle de los temas tratados:**A. Requerimientos del sistema*****Usuarios***

Los usuarios del sistema han de contar con información básica como: tipo de identificación, número de identificación, nombre, apellidos, género (masculino, femenino, indefinido), fecha de nacimiento, números telefónicos, residencia, nacionalidad, nombre del usuario, correo, contraseña, roles y estado del usuario. Los usuarios se dividirán en: estudiante, padre de familia y funcionario de Kamuk. Cada uno de los usuarios debe tener un rol asignado, de lo contrario no podrá ser ingresado.

Estudiantes:

Los estudiantes, aparte de la información como tal de usuarios, deben aportar la siguiente información:

- Perfil de entrada (definido y establecido en el sistema SIRA) por año lectivo
 - Aspectos generales:

Aspectos Generales	
1.	Mantiene periodos de atención concentración adecuados durante su trabajo en clase
2.	Completa en el aula el trabajo asignado
3.	Sigue instrucciones correctamente
4.	Retiene y utiliza la información estudiada previamente
5.	Muestra un buen nivel de independencia en la ejecución de sus trabajos
6.	Muestra buenos hábitos de trabajo
7.	Se presenta puntualmente a las lecciones

- Destrezas y habilidades:

Destrezas y habilidades propias del nivel	
1.	Posee conocimientos adecuados de los contenidos requeridos para su nivel
2.	Logra poner en práctica conceptos y procedimientos propios de la materia
3.	Muestra interés por la materia
4.	Participa de manera regular en la clase
5.	Se esfuerza por mejorar cada día
6.	Muestra una actitud responsable hacia su trabajo dentro y fuera del aula

- Área socioafectiva:

Area Socioafectiva	
1.	Se relaciona adecuadamente con sus compañeros
2.	Muestra un adecuado manejo de sus emociones
3.	Respeto las figuras de autoridad
4.	Muestra una conducta adecuada acorde con las situaciones escolares cotidianas

- Perfil de salida (definido y establecido en el sistema SIRA) por año lectivo

- Cada perfil debe indicar si los puntos están con estado: logrado, logrado con ayuda, en proceso y no logrado todavía.
- Fecha de ingreso a la institución
- Fotografía
- Estado de matrícula
- Datos médicos (tipo de sangre, padecimientos, medicamentos autorizados en caso de emergencia, medicamentos restringidos en caso de emergencia)
- Padres del estudiante
- Autorizados para retirar al estudiante
- Adecuación curricular

Padres:

- Profesión
- Restricciones legales (de existir alguna)
- Estudiantes (hijos)

Funcionarios:

- Labor o título
- De ser profesor, nivel académico (prescolar, primaria o secundaria)

Perfiles

Los perfiles están definidos en las características de los usuarios.

Roles

Por otro lado, dicho sistema debe permitir que el usuario tenga múltiples roles, por ejemplo: padre de familia y funcionario de la institución. Se requiere de un gestor de roles que facilite su creación, modificación, el borrado o su consulta. Dicha sección debe estar habilitada para eliminar la posibilidad de la existencia de usuarios repetidos, así como para que la información suministrada cumpla con los requisitos básicos, como los

formatos de entrada. Cada rol debe definir a qué secciones del sistema permite el ingreso del usuario. Los roles se definen de acuerdo con las necesidades de acceso a las pantallas. Dichos roles serían el de profesor, dirección, coordinación, auxiliar, finanzas y tecnología de información.

Login

Para ingresar al sistema académico, la página web debe contar con un *login* general que exija la identificación mediante un nombre de usuario y contraseña. Dicho proceso, debe desplegar, al terminar, la pantalla principal mostrando las opciones ligadas con los accesos asignados al perfil de usuario: estudiante, padre de familia y funcionario de la institución. Dicho proceso de login debe contar con seguridad para restringir el acceso al sistema académico y su información.

Pantalla Principal

El software habrá de desplegar una página con un menú de opciones utilizables de acuerdo con los roles asignados a dicho usuario. Esta pantalla debe exhibir los colores de la institución para salvaguardar la imagen de Kamuk School y sus características visuales. Dicha información se encuentra en el manual de marca de la institución.

Módulo Administración

Dicho módulo debe contar con las siguientes funcionalidades:

- **Control de usuarios:** sección dentro del menú para consulta, creación, modificación y eliminación de usuarios, según las características antes citadas. Además, debe contar con formularios para editar cualquiera de los registros, con

el fin de contar con un mantenimiento preciso para los usuarios del sistema. Dicha sección, ha de asegurar la inexistencia de usuarios repetidos y que la información suministrada cumpla con los requisitos básicos como formatos de entrada.

- **Control de roles:** sección dentro del menú destinada a la consulta, creación, modificación y eliminación de roles contando con las características citadas previamente. Además, debe desplegar formularios para la edición de cualquiera de los registros, con el fin de contar con un mantenimiento preciso para los roles de sistema. Dicha sección, deberá asegurar la inexistencia de roles repetidos, y que la información suministrada cumpla con los requisitos básicos como formatos de entrada.
- **Control de estudiantes:** sección dentro del menú para consulta, creación, modificación y eliminación de estudiantes contando con las características citadas previamente. Además, deberá contar con formularios para editar cualquiera de los registros, con el fin de que haya un mantenimiento preciso para los estudiantes del sistema. Dicha sección, debe asegurar la inexistencia de estudiantes repetidos y que la información suministrada cumpla con los requisitos básicos como formatos de entrada. Se debe contar con la historia de los estudiantes que hayan formado parte de Kamuk School y citar el estado de ellos.
- **Control de materias:** sección dentro del menú consignada a la consulta, creación, modificación y eliminación de materias. Además, ha de contar con formularios para la edición de cualquiera de los registros, de modo que se cuente con un mantenimiento preciso para las materias del sistema. Dicha sección, debe asegurar la no repetición de materias, y el cumplimiento de la información

suministrada con los requisitos básicos como formatos de entrada. Dichas materias no se deben borrar sino mantener por estado de actividad. Cada una de ellas debe presentar las características básicas: descripción, estado y nivel académico al que pertenece (prescolar, primaria y secundaria).

- **Control del año lectivo:** sección que debe permitir el control del año lectivo por iniciar o en curso. Aquí se hará la planificación anual de: ----
 - Distribución de secciones: debe asignar a los estudiantes la sección, el profesor guía, las materias y sus respectivos profesores asignados.
 - Distribución de aulas, una por sección.
 - Definición de períodos (bimestre, trimestre, cuatrimestre o semestre).

Dicha sección debe permitir la creación los cursos lectivos año con año sin afectar las configuraciones de años previos, para mantener un registro de estos. El año se debe planear previo a su inicio.

- **Asignación de horarios:** sección separada del control de año lectivo. Dicha sección debe permitir la organización de los horarios de las secciones creadas en el año lectivo y asegurar la distribución de los profesores a lo largo de la semana, sin que dos o más de ellos tengan clases asignadas simultáneamente. En esta sección se deben distribuir las materias del nivel académico creadas en la sección de materias.

Módulo Profesores

Dicho módulo debe contar con las siguientes funcionalidades:

- **Control de asistencia:** Cada profesor debe tener la posibilidad de pasar la asistencia de las secciones para controlar la actividad estudiantil. La asistencia ha

de mostrarse por sección y el profesor pasará lista en el momento. Las opciones de asistencia serán: ausente, tarde o presente.

- **Control de boletas:** sección dentro del menú destinada a la consulta, creación, modificación y eliminación de boletas, según las características citadas en el reglamento de la institución. Además, debe contar con formularios para la edición de cualquiera de los registros, de modo que se cuente con un mantenimiento preciso para las boletas de los estudiantes. Dicha sección, debe asegurar que la información suministrada cumpla con los requisitos básicos como formatos de entrada. Las boletas se dividirán en colores según la gravedad. Dichas gravedades son: leves (amarillo), grave (verde) y muy grave (azul). Las boletas se enviarán a revisión de auxiliares para su aplicación o anulación. Cuando se aplica, se deberá descontar los puntos de conducta.
- **Evaluación de materias:** Cada profesor debe definir la forma de evaluar la materia; es decir, el educador podrá definir los rubros por evaluar, así como su respectivo puntaje. Es decir, el profesor ha de gozar de la posibilidad de realizar cantidad indefinida de pruebas y de definir el porcentaje que puede dársele a según los rubros de evaluación. Las evaluaciones deben ser susceptibles de edición en cualquier momento y únicamente por el profesor encargado de dicha materia. Un ejemplo de evaluación es el siguiente, pues las evaluaciones ya no son por entregables sino por objetivos, los cuales tienen sus propias rúbricas de evaluación:

Matemática	Rubro I	30%	Trabajo I	10%
			Trabajo II	15%
			Proyecto	30%

		Tareas	5%
		Evaluaciones escritas	40%
		Trabajo I	10%
		Trabajo II	10%
Rubro II	35%	Proyecto	25%
		Tareas	10%
		Evaluaciones escritas	45%
		Trabajo I	10%
		Trabajo II	10%
Rubro III	35%	Proyecto	25%
		Tareas	10%
		Evaluaciones escritas	45%

La sección debe asegurar el cálculo del porcentaje de la rúbrica de acuerdo con los resultados de los trabajos realizados, a los cuales el profesor asigna un determinado porcentaje de la nota. Los profesores deben tener la posibilidad de crear incontables números de trabajos e incontables rubros de evaluación, tales como: Comprensión de la matemática y desarrollo de la lógica matemática.

- **Registro académico de estudiantes:** El sistema debe permitir el control del registro académico de los estudiantes como se realiza en cualquier institución. Debe ser posible conocer las notas de todas las materias y cómo se fracciona dicha evaluación. Cada registro debe estar ligado con un estudiante. Al final de cada curso lectivo, las evaluaciones por trabajos se eliminarán y se tendrá únicamente la nota final del período por cada estudiante en cada materia cursada.

Este registro debe mostrar los resultados de las evaluaciones de las materias, así como de la conducta de estudiante, la cual dependerá de las tardías, ausencias injustificadas y boletas recibidas. Este registro se lleva en cada período académico durante el curso lectivo. El registro académico lo verá únicamente el profesor guía de la sección, coordinación y dirección (la sección es sólo por año).

- **Calendario de cursos:** Cada profesor deberá contar con la posibilidad de ver su calendario, especificando las clases asignadas para su orden diario. El anterior mostrará únicamente la materia a impartir, sección y aula. Este calendario es de vista únicamente, no debe admitir ningún cambio

Módulos Restantes

El sistema debe crear una base de datos que contenga información de los módulos siguientes y una prevista para ligar la información de los estudiantes al sistema financiero de la institución.

- **Historial:** se requiere de una sección destinada al manejo del historial académico de los estudiantes. La información ha de ingresarse bajo la coordinación académica y la dirección, de modo que sea posible proporcionar la información solicitada por el Ministerio de Educación y por los estudiantes o sus encargados cuando requieren del historial para otros procesos académicos, tales como el traslado a otra institución, o en caso de que se soliciten otro tipo de registros por razones legalmente permitidas.
- **Reportes:** se requiere de una sección de reportes, donde se obtenga información de la base de datos general. Esta unidad debe facilitar el acceso al registro de los estudiantes y filtrar los reportes según lo especificado: datos anuales del

estudiante, historial académico del estudiante, cantidad de estudiantes por sección o profesor, cantidad de estudiantes por nivel o año, entre otros.

- **Repositorio de trabajos:** el sistema debe contar con un repositorio que permita a los estudiantes entregar y subir trabajos para la revisión de los profesores. Dicho repositorio ha de presentar a los profesores la opción para establecer los tiempos de entrega y definir las notas una vez revisados los trabajos. Dichas notas deben ligarse a la evaluación del estudiante.
- **Mensajes:** el sistema académico deberá contar con un sistema parecido al del correo electrónico y debe ofrecer la facilidad de filtrar por conversación, así como de guardar el historial por año. Una vez terminado el curso lectivo, se limpia la información para asegurar el control de uso del disco.
- **Permisos:** el sistema exige una sección de permisos, donde los padres puedan gestionar permisos específicos para los hijos. El sistema enviará los permisos a los profesores o a la administración según sea la necesidad, lo cual facilitará el control necesario para que sólo los padres puedan solicitar los permisos para las ausencias, salidas temprano, llegadas tardías o faltas de uniforme.
- **Calendario de actividades:** el sistema debe contar con un calendario donde los padres y estudiantes puedan informarse acerca de las actividades cercanas de la institución, de modo que estén enterados acerca de a cuáles de ellas deben asistir y tengan la posibilidad planearlas.
- **Módulo de estudiantes:** permite únicamente la lectura del registro académico del estudiante, aunque se incluye un acceso extra al repositorio para subir sus trabajos asignados.

- **Módulo de padres de familia:** dicho módulo tendrá las mismas características del de estudiantes, con la diferencia de que se les facilitará a los padres el acceso a permisos, de modo que estos se manejen exclusivamente entre el funcionario de la institución y los padres. Los padres sólo podrán ver las notas si sus hijos son menores de edad, de lo contrario estará bloqueado el acceso a éstas.

B. Aspectos técnicos

Crear un sistema web accesible desde los navegadores más utilizados, tales como Google Chrome, Mozilla Firefox o Microsoft Edge. Se debe utilizar Net Framework en el entorno de desarrollo de Visual Studio 2017 y, por último, se debe crear la base de datos en SQL Server 2016.

5.1.1.2 Equipo de trabajo.

El equipo de trabajo del proyecto se conforma de las siguientes personas y roles:

Tabla 3.
Equipo de trabajo

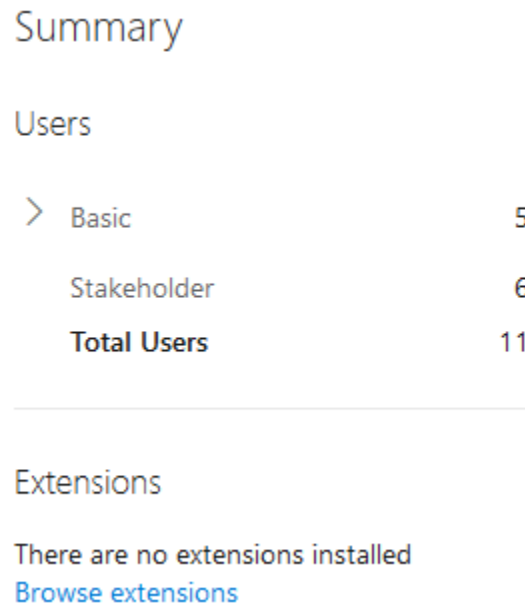
Nombre	Rol
Mario Angulo Cordero	Scrum Master y Product Owner
Alejandro Chavarría Benavides	Arquitecto
Daniel Valle Jiménez	Administrador de base de datos
Diego Saravia Valverde	Gestión de calidad
Roberth Herrera Castro	Desarrollador
Iván Duarte Solís	Desarrollador

Fuente: Elaboración propia

La descripción de los roles es únicamente informativa para la institución, su propósito consiste en respaldar el conocimiento de los integrantes; cumple la misma función para con la universidad, pues cada miembro desarrolla el tema de su proyecto enfocado en un área específica. A pesar de ello, el equipo de trabajo es autosuficiente y podrá cumplir con sus tareas como tal; en caso de la ausencia de algún miembro, el equipo debe adaptarse y completar las tareas del sprint, de modo que se termine la propuesta del proyecto para la institución Kamuk School.

5.1.1.3 Herramienta de gestión de proyectos.

Para la gestión de proyectos se utiliza la herramienta de uso gratuito Azure DevOps, ligada a una cuenta de correo kamedcr@outlook.com. Dentro de la organización creada se utiliza el proyecto KAMEDCR y se distribuyen los usuarios como se ve en la figura 25.



Summary	
Users	
> Basic	5
Stakeholder	6
Total Users	11
Extensions	
There are no extensions installed	
Browse extensions	

Figura 23: Usuarios de KAMEDCR en Azure DevOps

Fuente: Azure DevOps

Los usuarios *Basic* son aquellos que tienen acceso total a la herramienta, incluyendo a los repositorios. Los usuarios *Stakeholder* son aquellos con un acceso restringido, es decir, gozan de un acceso limitado, sin derecho a ver los repositorios. Por licenciamiento, únicamente se permite a cinco usuarios el empleo Basic sin costo. Los tipos de usuario se clasifican de la siguiente manera:

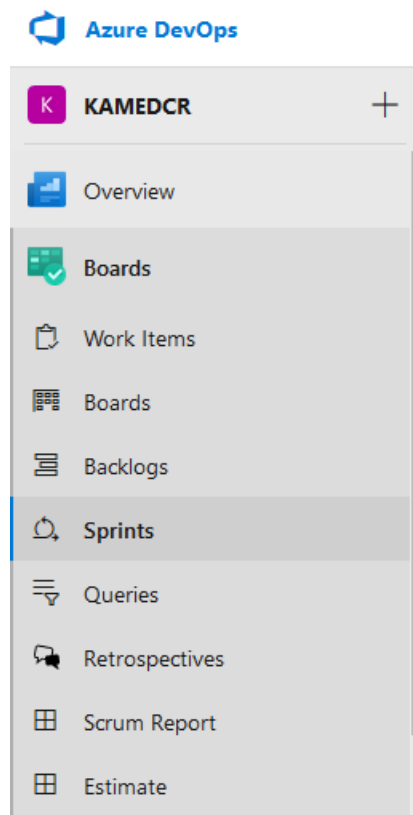
Tabla 4:
Usuarios de KAMEDCR en Azure DevOps

Nombre	Rol	Tipo de usuario
Kamed CR	Usuario dueño del proyecto	Basic
Mario Angulo	Scrum Master y Product Owner	Basic
Alejandro Chavarría	Arquitecto	Basic
Roberth Herrera	Desarrollador	Basic
Iván Duarte	Desarrollador	Basic
Diego Saravia	Gestión de calidad	Stakeholder
Daniel Valle	Administrador de bases de datos	Stakeholder
Alejandro Bogantes	Tutor	Stakeholder
Erick López	Tutor	Stakeholder
Luis Navarro	Tutor	Stakeholder
Yusselin Murcia	Tutora	Stakeholder

Fuente: Elaboración propia

Además, la herramienta permite usar el menú como se observa en la figura 26.

Figura 24: Menú de Azure DevOps



Fuente: Azure DevOps

Las herramientas se utilizan de la siguiente manera:

Work Items: Dentro de la sección pueden observarse las historias de usuario y tareas según un orden para cada actividad, ya sea como nuevas, aprobadas, guardadas y listas.

Boards: En esta sección se muestra la lista ordenada de las tareas totales del proyecto

Back Logs: Dentro de esta sección, se utilizan Work Items, las cuales corresponden con las historias de usuario, y Tasks, tratadas como tareas.

Sprints: En la sección se muestran las tareas en sus estados de: por hacer, en proceso y terminado. Así mismo, se tienen las vistas Backlog y el burn-down chart del sprint seleccionado.

Queries: Se utiliza únicamente para la búsqueda de datos pertenecientes al proyecto en caso de no hallarlos.

Retrospectives: Esta sección es de uso exclusivo para las reuniones de Restrospective, de Scrum.

Scrum Report: Esta es una sección de control destinada a los reportes según sea la necesidad del momento.

Estimate: Esta sección se utiliza para estimar el costo de los trabajos durante el Sprint Planning.

Aparte de las anteriores, se utiliza una herramienta más llamada Repositories. Dicha sección contiene el repositorio donde se almacena el proyecto a nivel de

controlador de versiones, ligado a Visual Studio (herramienta de desarrollo utilizada). El repositorio principal se llama Dev.

5.1.1.4 Eventos de Scrum

Los eventos de sprint se desarrollarán de la siguiente manera:

Sprint: Cada sprint tendrá una duración de dos semanas.

Sprint Planning: Debido a las diferencias de ubicación, las reuniones se darán de manera remota, por medio de una videollamada. En dichas reuniones se contará con la presencia del equipo del proyecto, así como de cualquier otra persona que el Scrum Master considere necesaria para ayudar al equipo en la planeación de tiempos. Por otra parte, se utilizará la herramienta Estimates de Azure DevOps para el cálculo de tiempos.

Daily Scrum Meeting: Las reuniones tendrán lugar de lunes a viernes, quedando libre los fines de semana, se darán de manera remota y tendrán una duración máxima de quince minutos.

Sprint Review: Por situaciones de tiempo, esta reunión contará únicamente con el equipo de proyecto y se dará de manera remota. Posteriormente, el Product Owner presentará el incremento a la institución de acuerdo con su disponibilidad.

Sprint Retrospective: Estas reuniones se darán de manera remota y se utilizará la herramienta Retrospectives de Azure DevOps. En estas reuniones se contará con todo el equipo del proyecto.

Groomings: Se darán de manera remota y se harán según las necesidades dadas a lo largo del avance de cada Sprint.

5.1.1.5 Artefactos de Scrum.

Product Backlog y Sprint Backlog: Se manejarán con la herramienta Backlog de Azure DevOps. El Product Backlog se elaborará de acuerdo con los requerimientos del proyecto y el Sprint Backlog variará según el Sprint en curso.

Incremento: El incremento ha de considerar únicamente las tareas marcadas como completadas y aprobadas por el Product Owner. Dicho avance deberá encontrarse en el área de Repositories y se publicará a nivel de red en la página web kamedcr.com.

Burn-down chart: Se utilizará el que brinda la herramienta Azure DevOps según cada sprint.

5.1.2 Desarrollo

5.1.2.1 Product Backlog,

A continuación, se observan los ítems trabajados con el Product Backlog del proyecto:

Tabla 5 .
Ítems del Product Backlog

Número	Ítem
1	Plantillas de requerimientos
2	Diseño de la base de datos
3	Control de boletas
4	Control de roles
5	Control de estudiantes
6	Control del año lectivo

Número	Ítem
7	Control de materias
8	Asignación de horarios
9	Evaluación de materias
10	Registro académico de estudiantes
11	Calendario de cursos
12	Control de asistencia
13	Pantalla principal
14	Control de usuarios
15	Historial
16	Reportes
17	Repositorio de trabajos
18	Mensajes
19	Permisos
20	Calendario de actividades
21	Control de aulas
22	Documento de Alejandro
23	Documento de Mario
24	Documento de Iván
25	Documento de Daniel
26	Documento de Diego
27	Documento de Roberth

Número	Ítem
28	Pruebas de calidad de programación de control de personas
29	Convocatoria
30	Pruebas finales de calidad
31	Base de datos para uso futuro

Fuente: Equipo de proyecto de KAMEDCR

5.1.2.2 Sprints.

De acuerdo con las reuniones de sprint planning, los sprints se trabajaron de la siguiente manera:

Tabla 6:
Calendario de Sprints

Fecha	Tareas	Meta
Sprint I (18 de febrero – 4 de marzo)	- Plantillas de requerimientos - Estructura de la página - Plantillas de calidad - Plan de calidad - Lineamientos de programación - Diseño de base de datos - Login de la página web - Control de versiones - Tabla de usuarios - Contratación de host - Compra de dominio	Adaptarse al uso de Scrum, reforzando los conceptos de programación para acoplar el equipo y entregar la primera parte funcional del proyecto

Fecha	Tareas	Meta
Sprint II (5 de marzo – 19 de marzo)	- Instalación de software en el host - Publicación de página web	Completar los mantenimientos de tablas restantes para las actividades dependientes de éstas.
	- Login de la página web	
Sprint III (20 de marzo – 3 de abril)	- Control de estudiantes - Control de asistencia - Diseño de base de datos - Pantalla principal - Control de usuarios - Control de roles - Control de materias - Control de boletas - Control de aulas - Control de estudiantes - Control de año lectivo - Asignación de horarios - Evaluación de materias	Terminar los controles necesarios con éxito para los procesos críticos del sistema como evaluación y planeación de año lectivo
	- Control de asistencia - Calendario de cursos - Control de usuarios - Control de roles - Control de boletas - Control de aulas	Completar el 100% de las tareas del Sprint para no heredar procesos a los siguientes sprints
Sprint IV (4 de abril – 18 de abril)		

Fecha	Tareas	Meta
Sprint V (19 de abril – 3 de mayo)	- Registro académico de estudiantes	Terminar los CRUDs para los dos mantenimientos finales
	- Control de estudiantes	
	- Asignación de horarios	
	- Control de usuarios	
	- Control de roles	
	- Control de materias	
	- Control de boletas	
	- Control de aulas	
	- BUG [2] Control de boletas	
	- BUG [1] Control de usuarios	
	- Documento Roberth	
	- Documento Diego	
	- Documento Daniel	
	- Documento Iván	
	- Documento Mario	
Sprint VI (4 de mayo – 18 de mayo)	- Documento Alejandro	Complementar el trabajo del documento y la ejecución del proyecto
	- Control de estudiantes	
	- Control del año lectivo	
	- Asignación de horarios	
	- Evaluación de materias	
	- Control de asistencia	
	- Calendario de cursos	
	- Control de boletas	
	- Control de aulas	
	- Registro académico de estudiantes	
	- Historial	

Fecha	Tareas	Meta
Sprint VII (18 de mayo – 1 de junio)	- Mensajes	Completar la documentación previa a la ejecución del proyecto para enfocarse en la propuesta
	- Documento Roberth	
	- Documento Diego	
	- Documento Daniel	
	- Documento Iván	
	- Documento Mario	
	- Documento Alejandro	
	- Documento Roberth	
	- Documento Diego	
	- Documento Daniel	
	- Documento Iván	
	- Documento Mario	
	- Documento Alejandro	
	- Control de estudiantes	
Sprint VIII (2 de junio – 15 de junio)	- Control del año lectivo	Llevar de la mano el documento con la programación y cumplir con los ítems del sprint
	- Evaluación de materias	
	- Control de asistencia	
	- Calendario de cursos	
	- Mensajes	
	- Repositorio de trabajos	
	- Pruebas de Calidad de programación del control de personas	
	- Permisos	
	- Documento Roberth	
	- Documento Diego	
Sprint IX (16 de junio – 30 de junio)	- Documento Daniel	Llevar de la mano el documento con la programación y cumplir con los ítems del sprint
	- Documento Iván	
	- Documento Mario	
	- Documento Roberth	
	- Documento Diego	

Fecha	Tareas	Meta
Sprint X (1 de julio – 15 de julio)	- Documento Alejandro	Terminar el desarrollo y bugs del sistema académico
	- Asignación de horarios	
	- Evaluación de materias	
	- Diseño de la base de datos	
	- Registro académico de estudiantes	
	- Repositorio de trabajos	
	- Permisos	
	- Calendario de actividades	
	- Reportes	
	- BUG [3] Justificación de ausencias	
	- Documento Roberth	
	- Documento Diego	
	- Documento Daniel	
	- Documento Iván	
	- Documento Mario	
	- Documento Alejandro	
	- Control del año lectivo	
	- Asignación de horarios	
	- Evaluación de materias	
	- Convocatoria	
Sprint XI (16 de julio – 31 de julio)	- Repositorio de trabajos	Terminar del documento escrito y presentar versión final a Kamuk
	- Reportes	
	- Base de datos para uso futuro	
	- Pruebas finales de calidad	

Fecha	Tareas	Meta
	- Documento Roberth	
	- Documento Diego	
	- Documento Daniel	
	- Documento Iván	
	- Documento Mario	
	- Documento Alejandro	

Fuente: Elaboración propia

5.1.2.3 Burn-down charts.

Como herramienta de medición del trabajo del equipo, se utilizan los burn-down charts para verificar el avance del desarrollo a lo largo de cada sprint. Durante cada uno de ellos, el equipo de desarrollo fue mejorando su cohesión y conocimiento acerca de la estimación de tiempos. En los primeros seis, se evidencia que los cálculos de trabajo superaban la capacidad de desarrollo del equipo. Además, durante cada uno de los primeros seis sprints, era necesario decidir aquello que podía dejarse para el sprint siguiente, ya que el cálculo realizado en los sprint planning resultó irreal.

Gracias a la maduración del equipo a partir del sprint siete se hace posible determinar la manera de completar eficientemente lo planteado, sin dejar tareas pendientes, lo cual agilizó en gran manera la velocidad de cumplimiento de labores sin terminar. El primer gráfico se muestra en blanco debido a un error del equipo en cuanto a dónde colocar las horas pendientes en el Azure DevOps. Por otra parte, los gráficos presentados se encuentran en inglés dada la configuración de la herramienta DevOps. A continuación, los burn-down charts a lo largo del desarrollo:

Burndown for: Sprint 1

How do I read this chart?

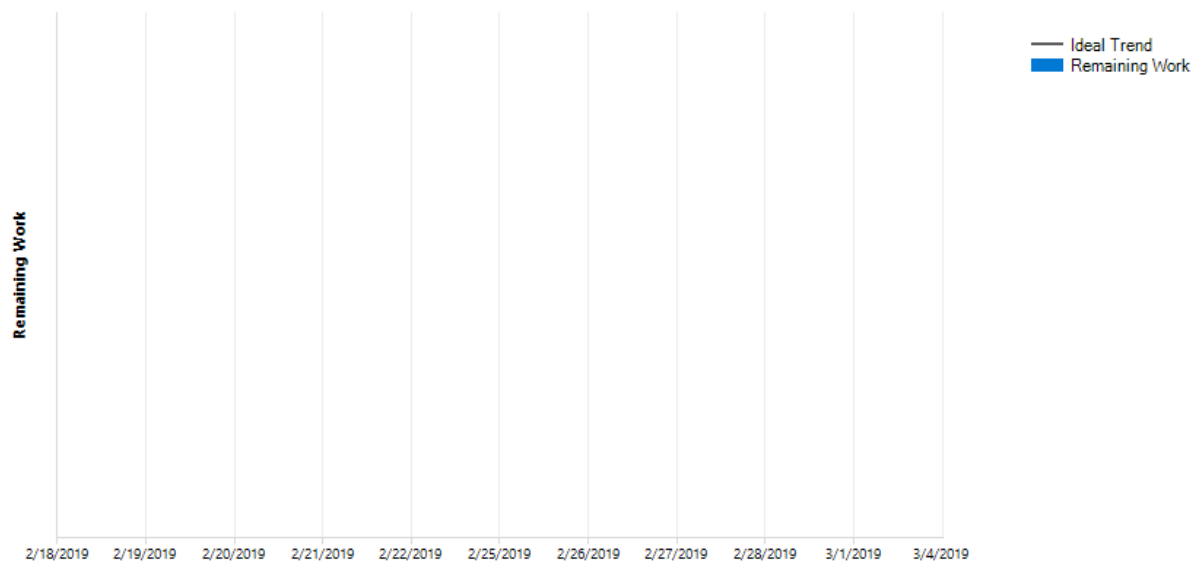


Figura 25: Burndown Sprint 1

Fuente: Azure DevOps

Como se observa, el gráfico no presenta ninguna tendencia o dato para analizar, ya que el equipo utilizó la herramienta para el control de tiempos de manera incorrecta, debido al proceso de aprendizaje de uso de la herramienta Azure DevOps.

Burndown for: Sprint 2

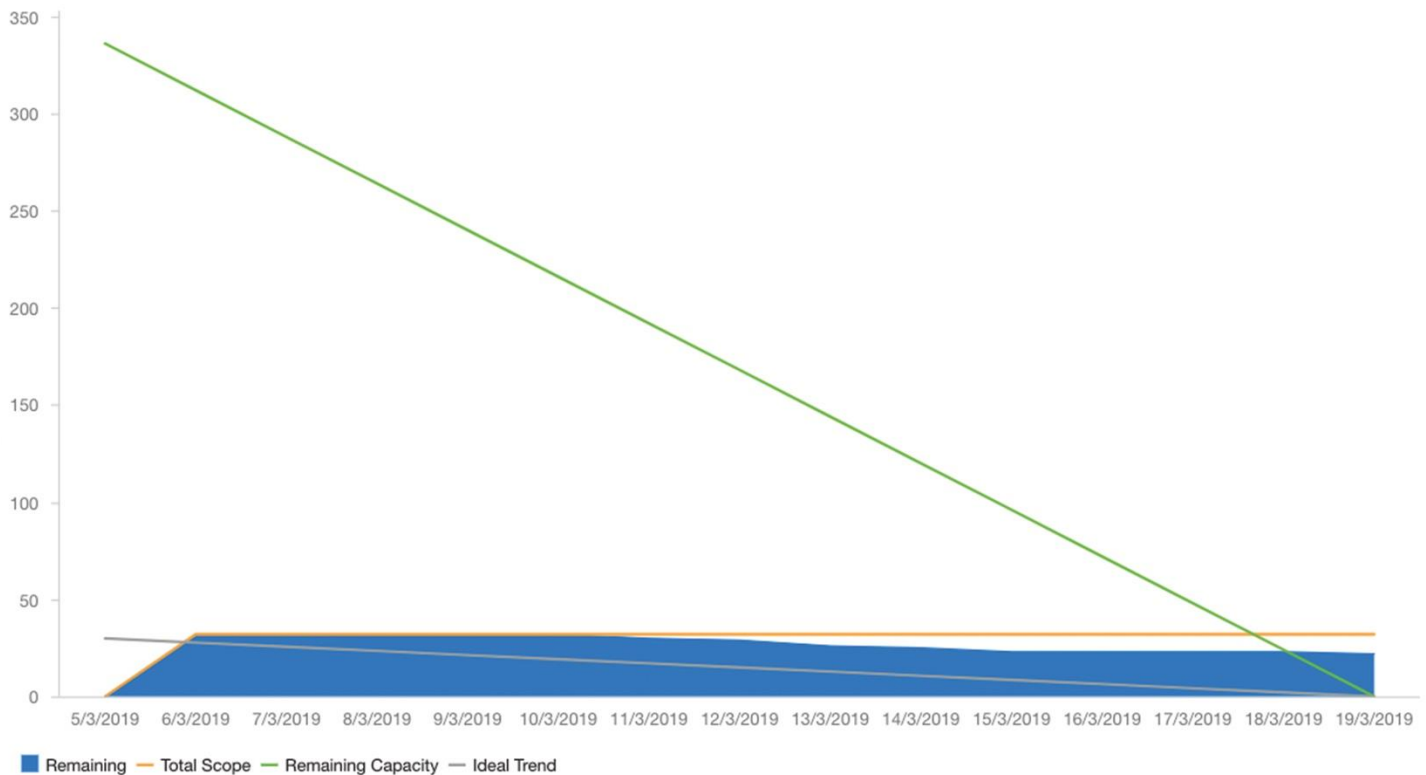


Figura 26: Burndown Sprint 2

Fuente: Azure DevOps

Durante el segundo sprint, se observa un planeamiento incorrecto de los cálculos de trabajo, pues el equipo estipuló una carga de trabajo superior a su capacidad; la información azul da cuenta de cuanto quedó pendiente de realizar. En este caso particular lo pendiente es mucho, por lo cual debió moverse para el siguiente sprint durante el sprint review.

Burndown for: Sprint 3

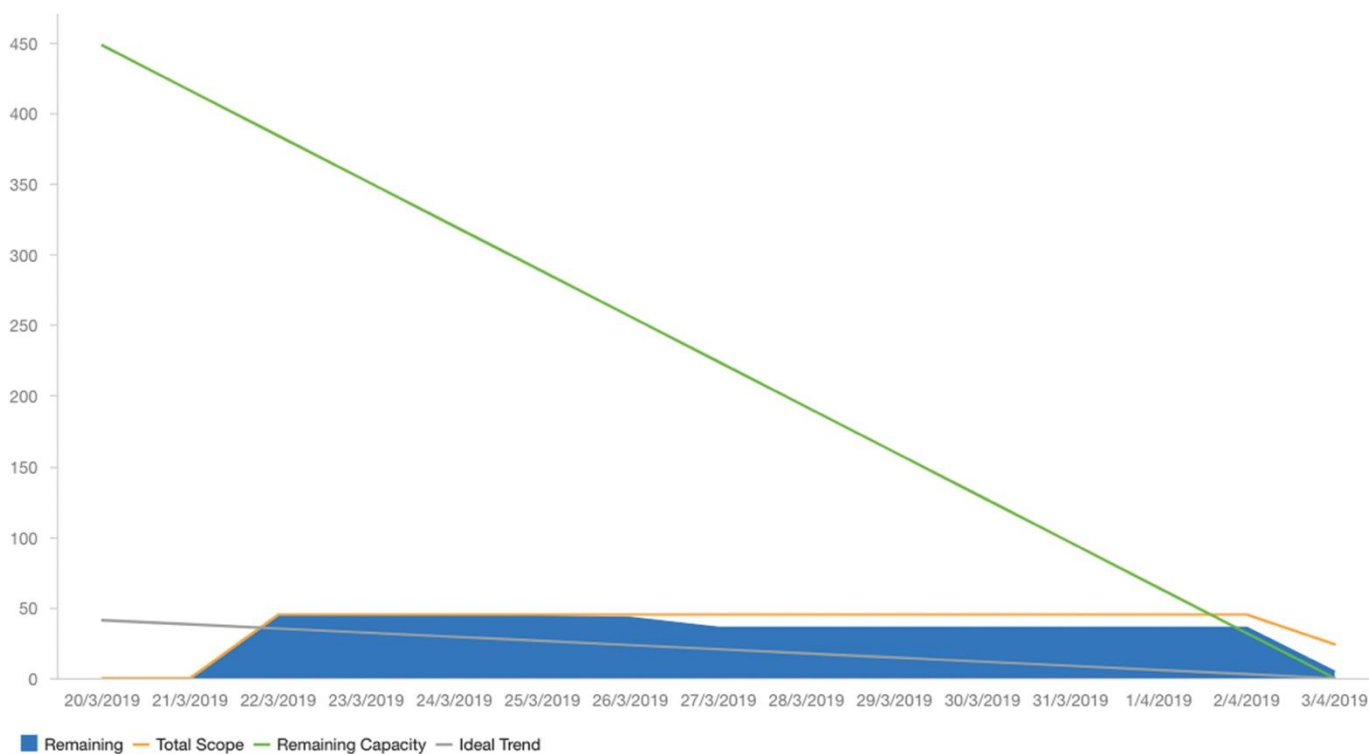


Figura 27: Burndown Sprint 3

Fuente: Azure DevOps

Dentro del sprint 3, el equipo de trabajo se desorientó, ya que se fue necesario replantear los requerimientos del sistema, por lo cual el trabajo de desarrollo se dejó para el siguiente sprint. Por esta razón se muestra una reducción drástica de trabajo pendiente en tan sólo un día.

Burndown for: Sprint 4

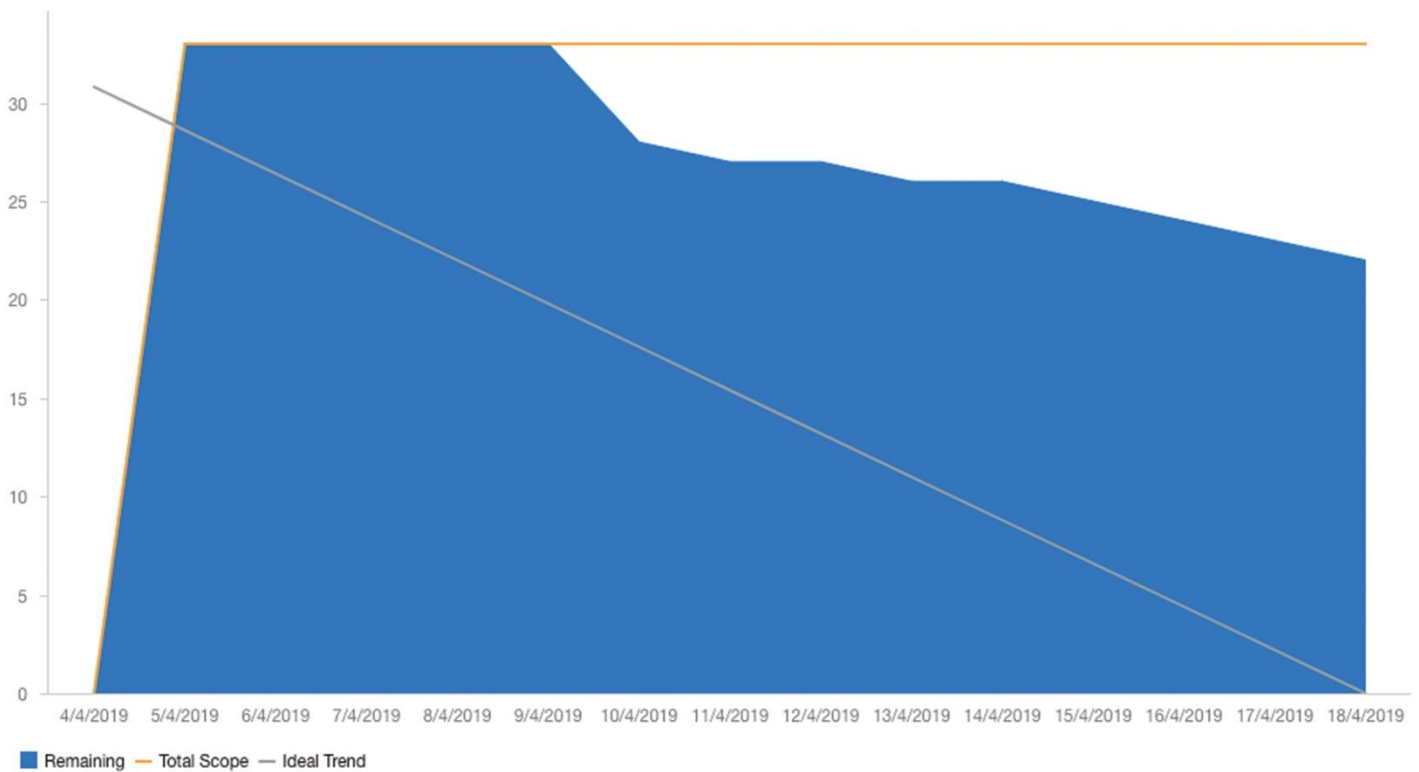


Figura 28: Burndown Sprint 4

Fuente: Azure DevOps

En el sprint 4, el equipo de desarrollo empieza a comprender cuál es su capacidad de trabajo; durante las reuniones de sprint planning, el equipo definía su trabajo de una manera más precisa, aún con algunas fallas evidentes de cálculos de tiempo.

Burndown for: Sprint 5

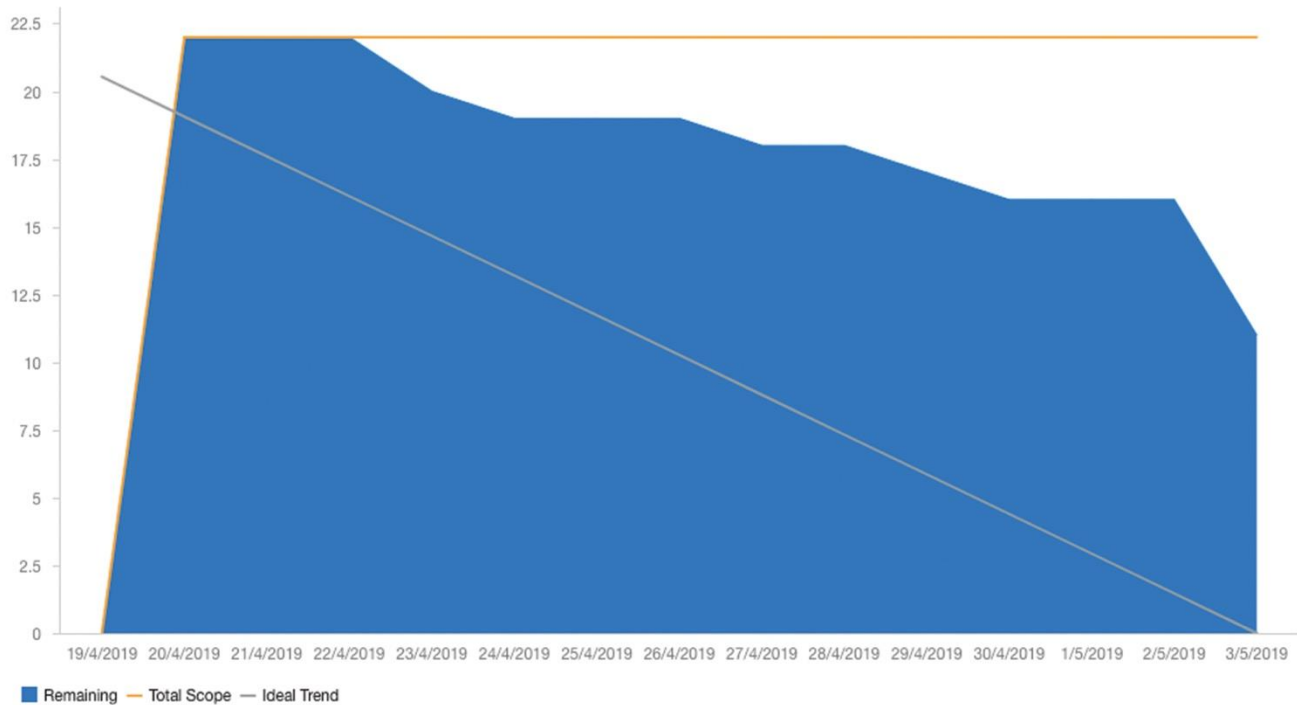


Figura 29: Burndown Sprint 5

Fuente: Azure DevOps

Debido a que en el sprint anterior el equipo cae en razón acerca de su capacidad para trabajar mejor, éste calcula nuevamente el volumen de trabajo que puede realizar, pero todavía no lo cumple. Esta situación obedeció principalmente a que no contemplaron adecuadamente las tareas por realizar, por lo que de nuevo deben moverse para el sprint siguiente, durante el sprint review.

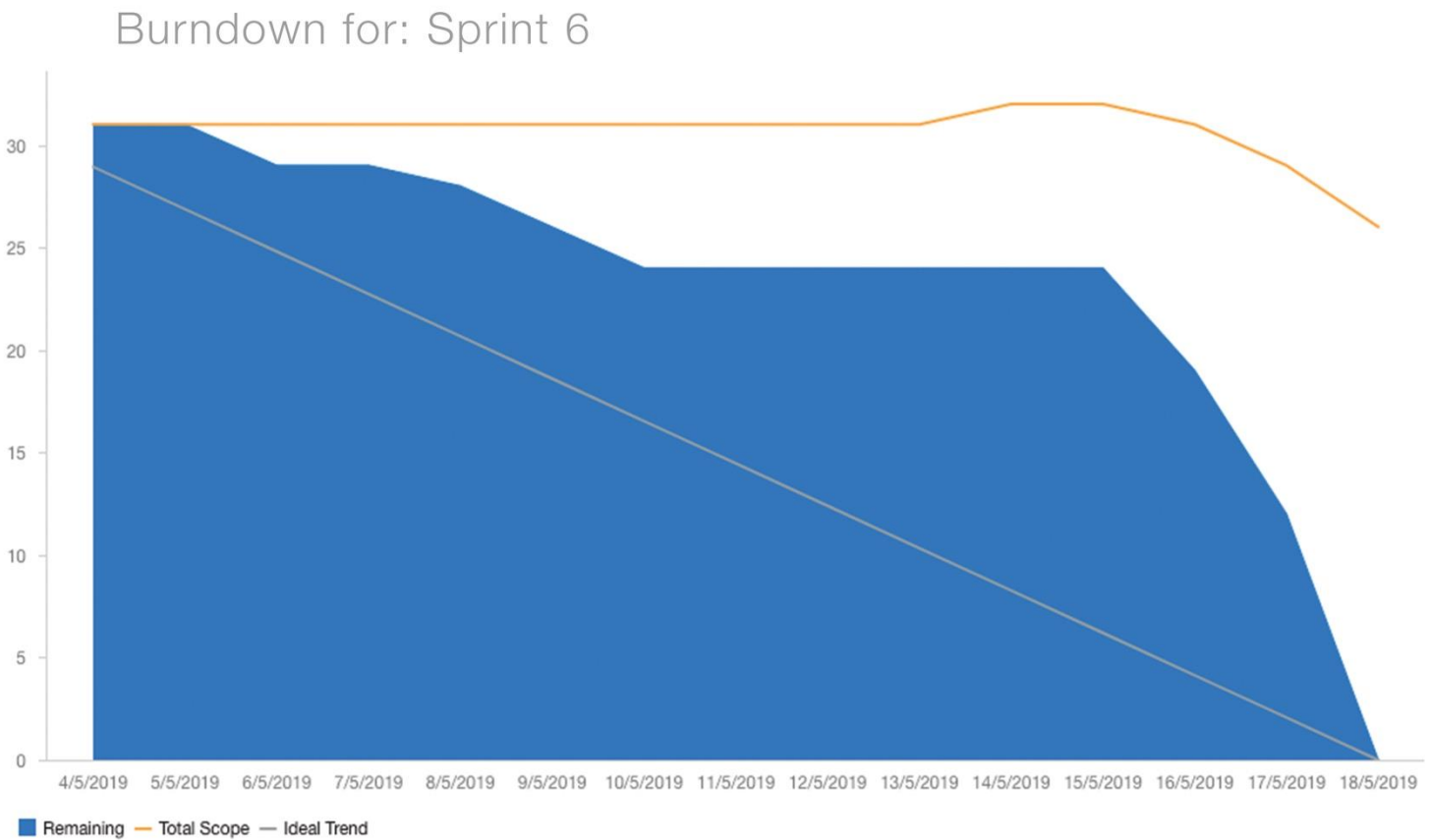


Figura 30: Burndown Sprint 6

Fuente: Azure DevOps

A partir de los errores del sprint 5, el equipo logra nuevamente una mejora en el control de tiempos y alcanza a terminar la mayoría de las tareas pendientes. A pesar de ello, la información azul evidencia que al final del sprint deben arrastrarse ciertas tareas incompletas al sprint 7.

Burndown for: Sprint 7

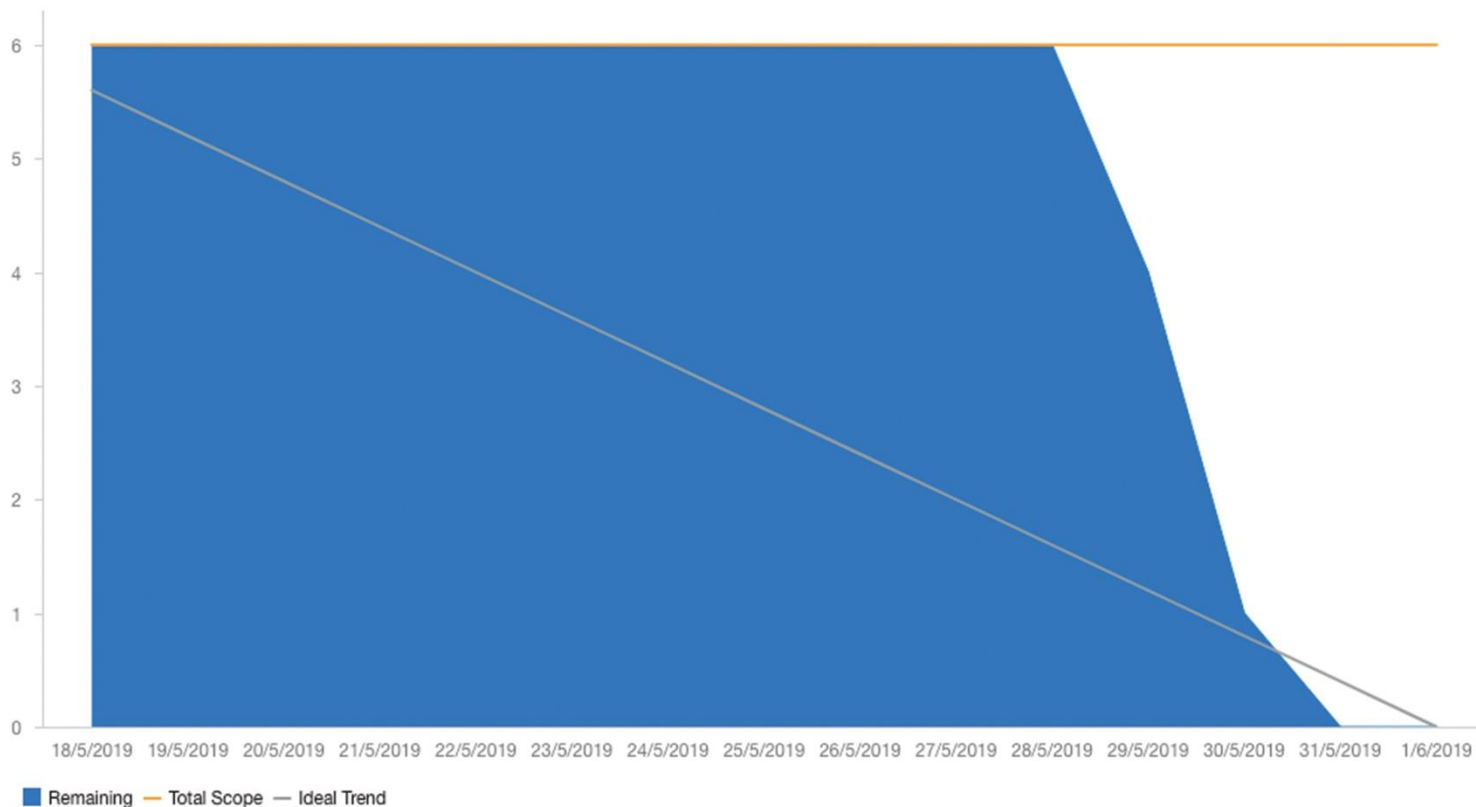


Figura 31: Burndown Sprint 7

Fuente: Azure DevOps

En el sprint 7, el equipo de trabajo encuentra la manera correcta de calcular tiempos y completa las tareas estipuladas en su totalidad por primera vez, sin necesidad de arrastrar nada al sprint 8. En este punto se constata la madurez del equipo y se logra que, a través de la experiencia, los cálculos de cargas de trabajo sean más precisos.

Burndown for: Sprint 8

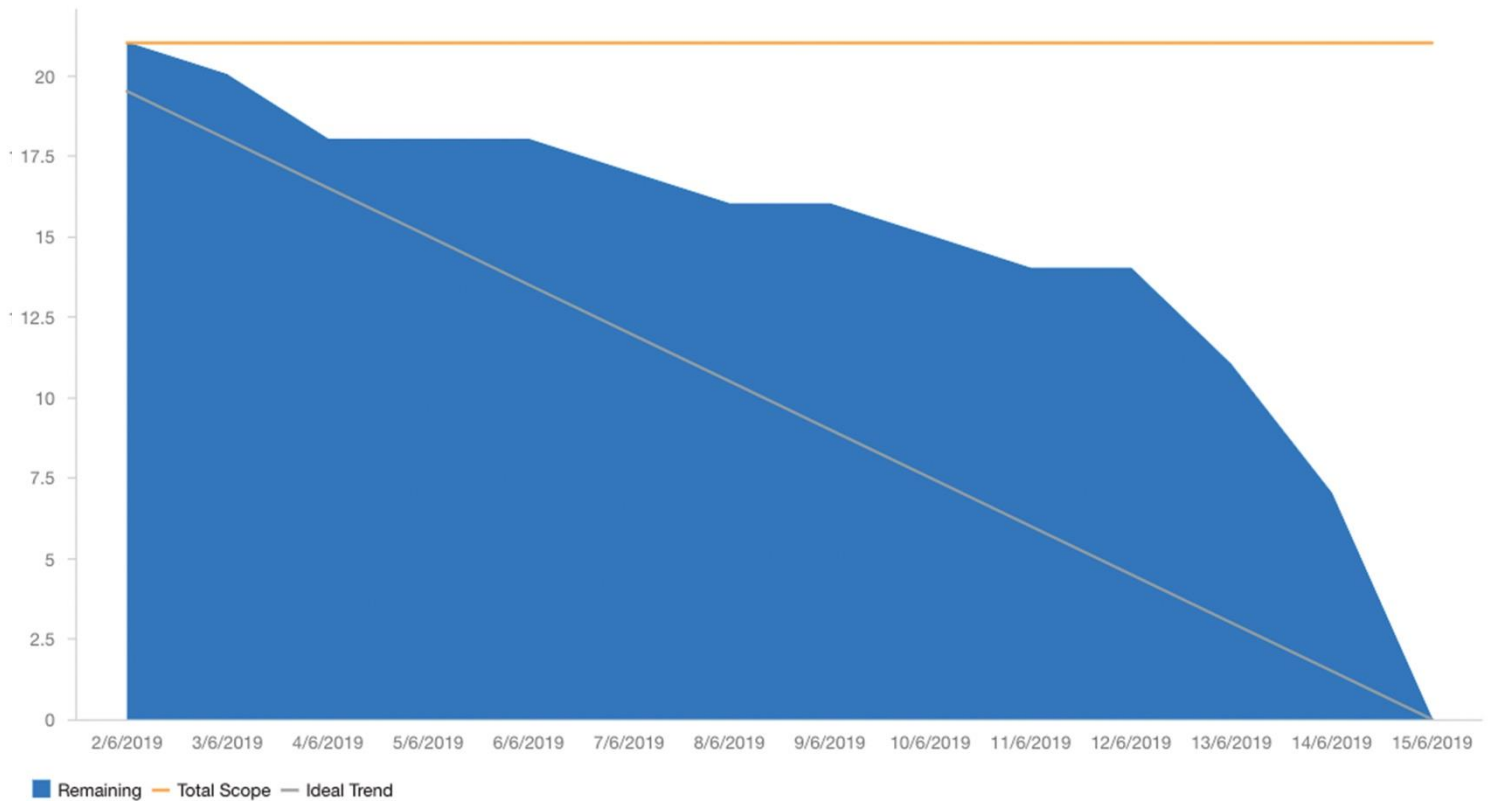


Figura 32: Burndown Sprint 8

Fuente: Azure DevOps

Al igual que el sprint 7, el equipo termina eficientemente las correspondientes tareas asignadas al 8. Además, los datos en azul dan cuenta de la mejora el rendimiento y disminuye el número de tareas de manera más balanceada. Lo anterior contribuye con el mejoramiento del trabajo del equipo, de manera que sea posible el enfocarse en lo realmente necesario.

Burndown for: Sprint 9

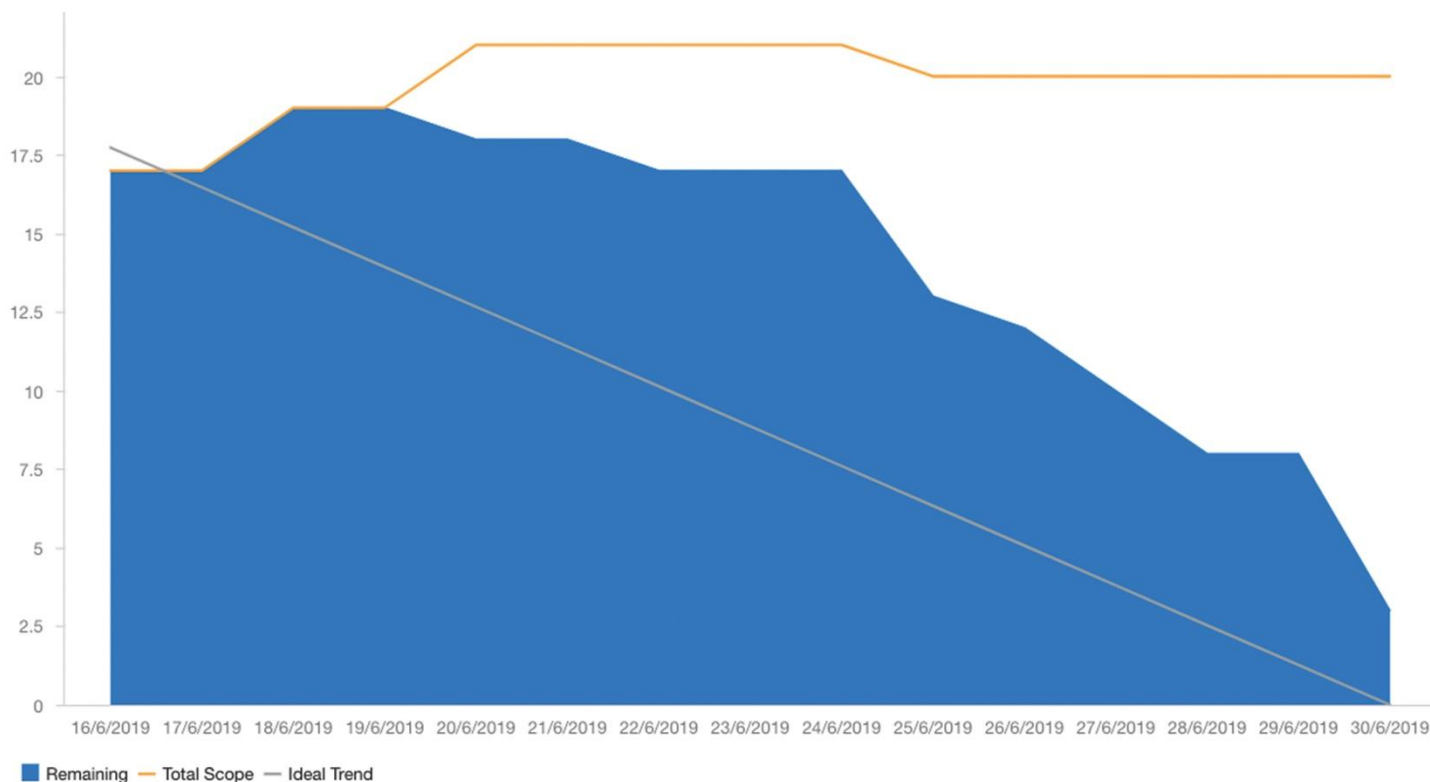


Figura 33: Burndown Sprint 9

Fuente: Azure DevOps

El sprint 9 se completó exitosamente, aunque se muestra incompleto. Sin embargo, se observa que la herramienta de Azure DevOps no refresca en su totalidad los gráficos generados durante el último día del sprint, lo cual significa que Azure DevOps debe tener los trabajos listos un día antes de terminar el sprint para tener un Burndown Chart preciso.

Burndown for: Sprint 10

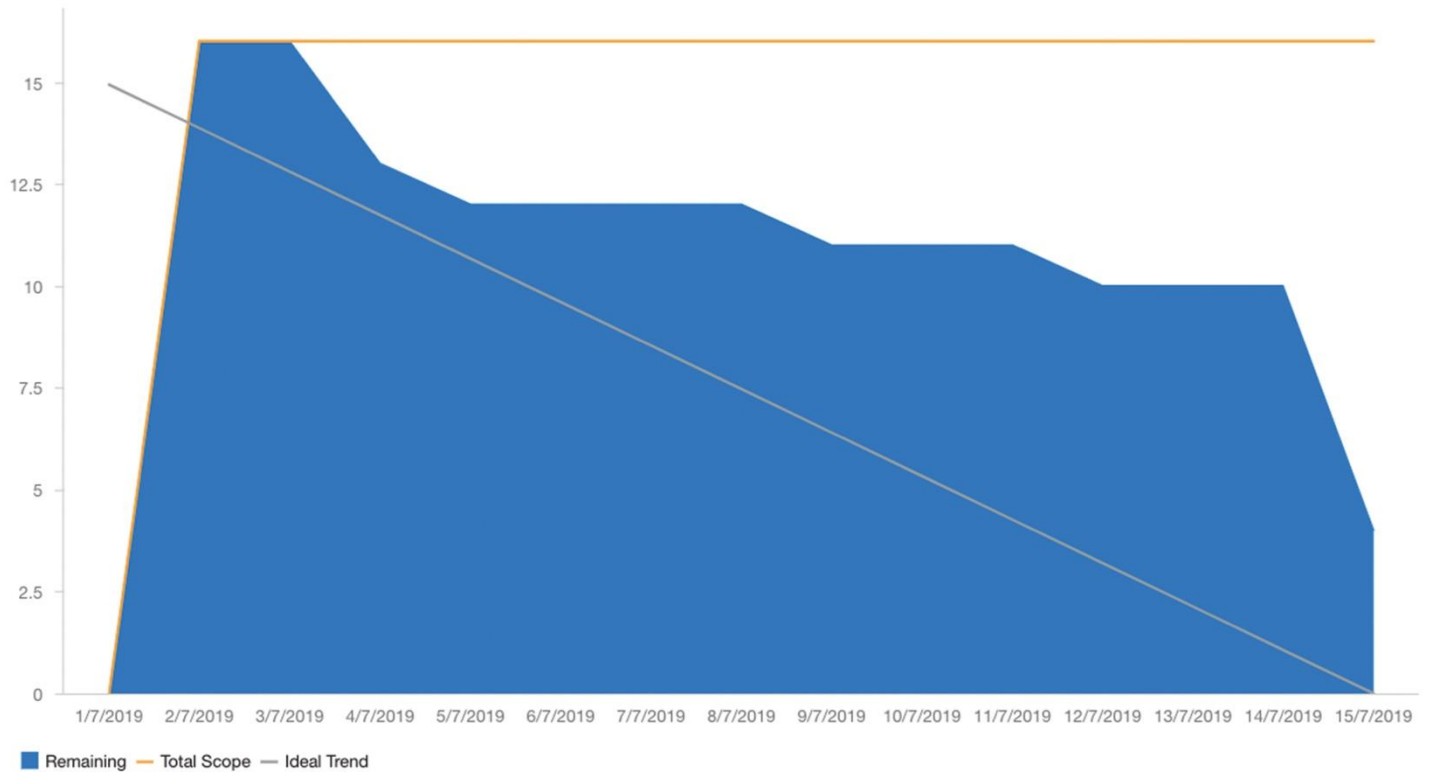


Figura 34: Burndown Sprint 10

Fuente: Azure DevOps

El penúltimo sprint del proyecto está destinado a terminar el desarrollo del proyecto. La gráfica presenta la forma en que los trabajos pendientes dejan de arrastrarse de un día a otro, por lo que la carga de trabajo se distribuye mejor durante el sprint y no se fuerza el trabajo en los últimos días de éste.

Debido a cambios surgidos en la herramienta de Azure DevOps, no se generan más los burndown charts una vez terminado el sprint. A pesar de ello, se completa el sprint sin dejar tareas pendientes y se entrega a la institución el producto final funcional, incluyendo los dos módulos estipulados: administrativo y profesores.

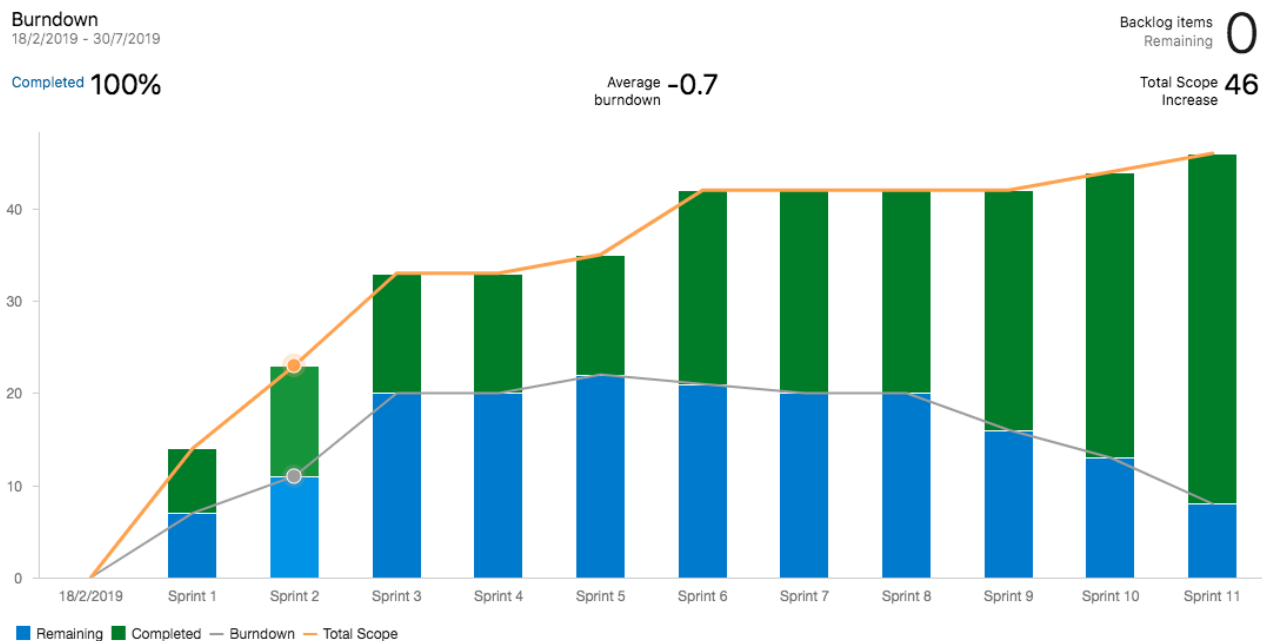


Figura 35: Burndown General

Fuente: Azure DevOps

Es posible observar cómo al final del último sprint se completan los ítems de backlog definidos para el proyecto realizado. A pesar de ello, hay módulos pendientes que no fueron tomados en cuenta para la generación del gráfico final, los cuales podrán desarrollarse utilizando la herramienta si la institución decide continuar con el proyecto.

5.1.2.4 Sprint Retrospectives.

Al final de cada sprint, habiendo terminado el sprint review –encargado de verificar las tareas de sprint completadas satisfactoriamente, se procedió con los sprint retrospectives, pues éstos constituyen la fuente de retroalimentación de mayor importancia para el equipo de desarrollo. Cada una de estas reuniones trató tres asuntos esenciales: qué se hizo bien, qué no se hizo bien y qué se va a mejorar en el siguiente sprint. Las fechas de dichas reuniones fueron:

- Sprint 1 (4 de marzo)
- Sprint 2 (20 de marzo)
- Sprint 3 (4 de abril)
- Sprint 4 (18 de abril)
- Sprint 5 (3 de mayo)
- Sprint 6 (17 de mayo)
- Sprint 7 (1 de junio)
- Sprint 8 (15 de junio)
- Sprint 9 (30 de junio)
- Sprint 10 (15 de julio)
- Sprint 11 (30 de julio)

A pesar de ser información de carácter exclusivo para el equipo de desarrollo, a continuación se presentan los resultados de una de las reuniones:

¿Qué se hizo bien?:

- Excelente labor del Scrum Master [5 votes]

- Buen trabajo en equipo [4 votes]
- La colaboración se mantuvo excelente [4 votes]
- Mejoró la comunicación [5 votes]
- Buena planeación de trabajo [5 votes]
- La comunicación ha mejorado [4 votes]
- Se terminó el sprint [4 votes]
- Coordinación del equipo [4 votes]
- Trabajo en equipo [4 votes]
- Se complementó el Sprint. [5 votes]
- Product Owner atento a las consultas con la institución [4 votes]
- Excelente lider, Product Owner [4 votes]

¿Qué no se hizo bien?:

- Mucha pérdida de tiempo [4 votes]
- Problemas con el VPS [5 votes]
- Se puede mejorar los tiempos [3 votes]
- No solucionaron rápido los proveedores los problemas del VPS [2 votes]
- Falta de comunicación por parte de la Universidad [6 votes]

5.1.3 Entrega.

A continuación, las pantallas del proyecto programado presentadas a Kamuk School el día 24 de julio de 2019, las cuales fueron del agrado de los personeros de la institución, quienes se mostraron satisfechos con en el tiempo estipulado y, por ende, las aprobaron.

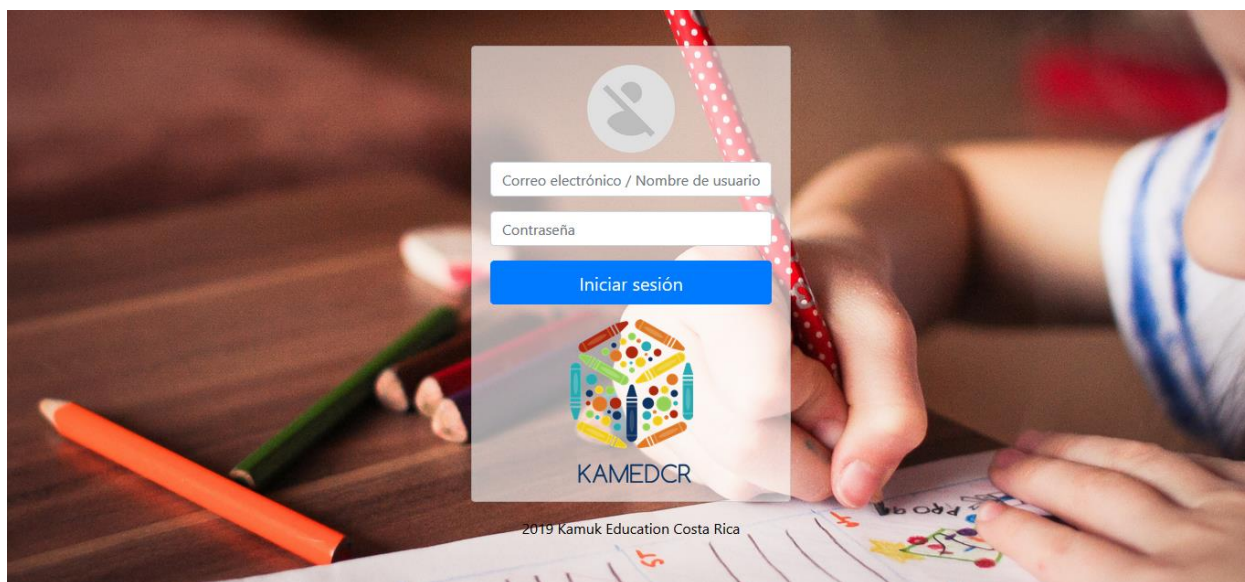


Figura 36: Login de KAMEDCR

Fuente: kamedcr.com

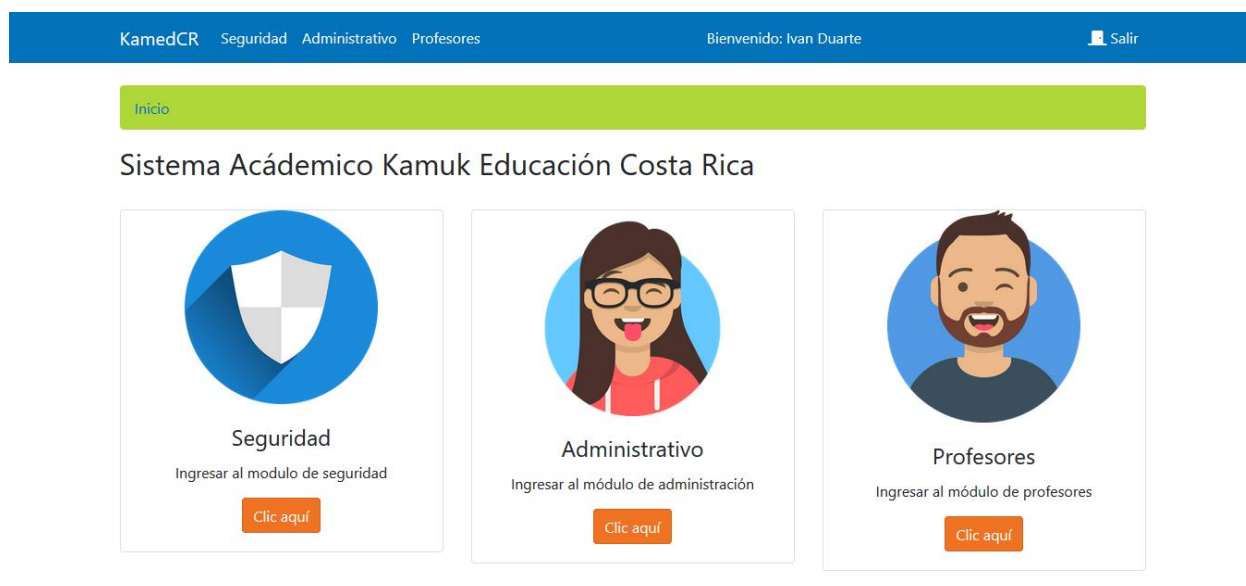


Figura 37: Página principal con súper usuario

Fuente: kamedcr.com

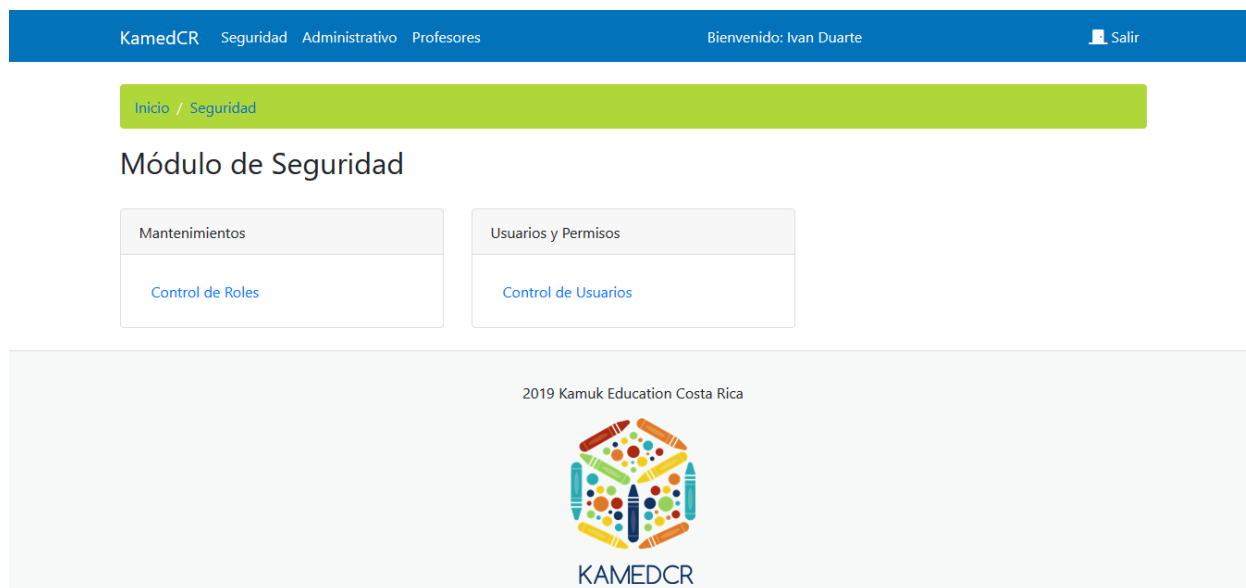


Figura 38: Módulo de seguridad

Fuente: kamedcr.com

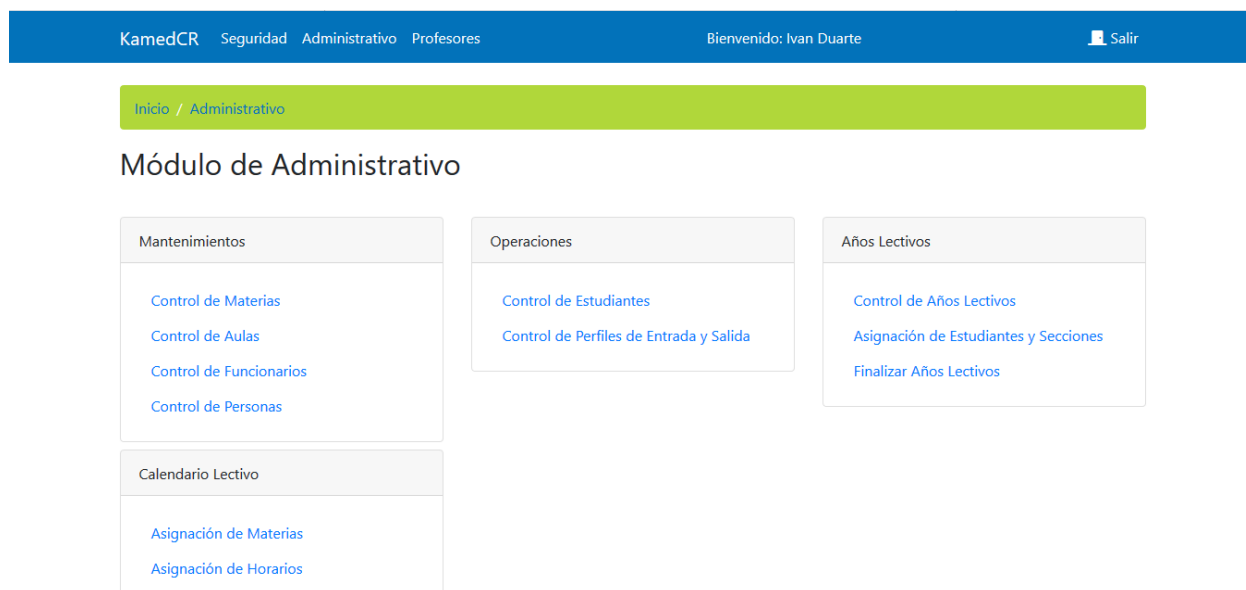


Figura 39: Módulo administrativo

Fuente: kamedcr.com

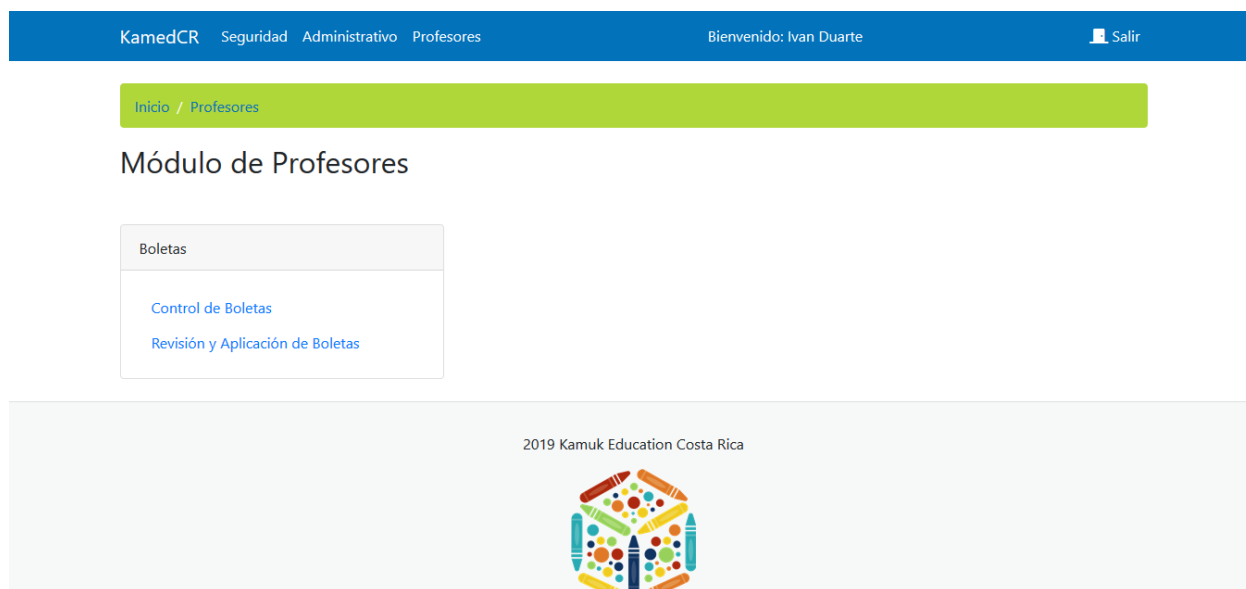


Figura 40: Módulo de profesores como administrador

Fuente: kamedcr.com

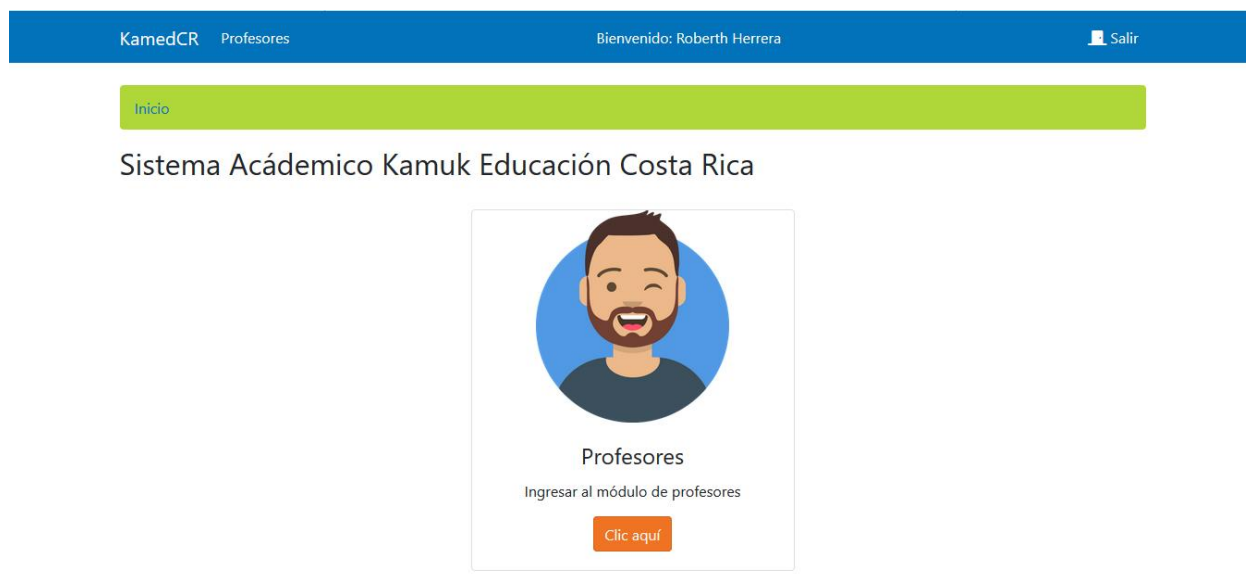


Figura 41: Página principal con usuario profesor

Fuente: kamedcr.com

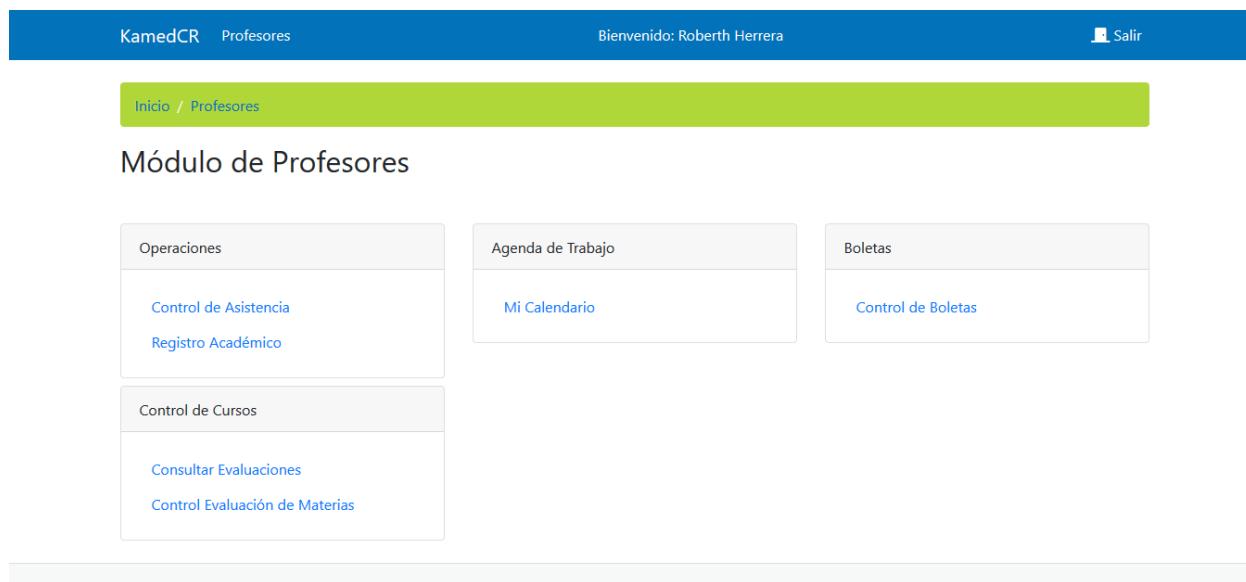


Figura 42: Módulo de profesores con usuario profesor

Fuente: kamedcr.com



Figura 43: Login desde dispositivo móvil

Fuente: kamedcr.com



Figura 44: Página principal desde dispositivo móvil

Fuente: kamedcr.com

Capítulo VI

Conclusiones y recomendaciones

6.1 Conclusiones

La implementación de un sistema académico eficaz, capaz de cubrir las necesidades de Kamuk School, ha sido un problema de gran impacto para la entidad. A través del proyecto realizado junto con el equipo de trabajo, se presenta la propuesta de un sistema académico hecho a la medida, de acuerdo con las necesidades actuales de la institución. Se parte de las características positivas de los sistemas SIRA y Class, con miras a mejorar las deficiencias de ambos sistemas académicos. Dicha propuesta se presenta como proyecto gestionado a lo largo del tiempo con un marco de trabajo como Scrum.

El estudio de la problemática institucional y la presentación de la propuesta han conducido a las siguientes conclusiones, en respuesta a los objetivos del proyecto según el orden respectivo:

- 1) Un estudio previo de la utilización de sistemas académicos, junto al estudio de la historia de uso de sistemas académicos en Kamuk School, permite crear una idea clara del porqué la institución los utiliza. A partir de dicha información se conoce con detalle las necesidades existentes desde su perspectiva administrativa y su realidad tecnológica, facilitando el proceso de plantear una propuesta adaptada a su realidad. Además, resulta importante el análisis del modo de operar de una institución en su gestión académica, pues ésta puede variar drásticamente entre una y otra. Los documentos informativos tales como el manual de marca, misión, visión y el reglamento facilitan el conocimiento del trabajo de la institución. Dicho conocimiento sumado al saber tecnológico contribuye la elaboración de una propuesta afín a la gestión anual de Kamuk

School, por lo tanto, es posible evitar que la institución enfrente los problemas presentados por los sistemas académicos SIRA y Class, los cuales no contemplan las necesidades específicas de Kamuk School en cuanto a la gestión académica. Por lo tanto, el cumplimiento del objetivo consigue al cien por ciento.

- 2) Dentro de un sistema académico existen varios módulos, pero los de mayor importancia son el módulo de profesores y el módulo administrativo, pues son los que gestionan la mayor parte de las labores durante un año lectivo. A través del estudio de gestión académica, administrativa y de opiniones de usuarios se generó la propuesta de un sistema académico con los módulos antes citados y cumplió con los requerimientos de la institución en un entregable final. En suma, se logra el cien por ciento del cumplimiento del objetivo.
- 3) El uso de metodologías ágiles contribuye a que el control de tiempos de entrega de un proyecto sea realmente preciso. Situaciones como insatisfacción del cliente con la entrega, equipos sin organización y una pobre planificación conllevan al fracaso de proyectos. El empleo de Scrum evitó lo anterior de manera satisfactoria. Reuniones diarias del equipo de trabajo, reuniones específicas dentro del sprint y planeaciones cada dos semanas llevaron a que entregables pequeños cumplieran con las expectativas de Kamuk School y a que al final del proceso se respetaran los tiempos establecidos para la entrega de la propuesta de desarrollo. Según lo anterior y con el respaldo de los gráficos mostrados en la sección 5.1.2.3 del proyecto, se cumple el objetivo en un cien por ciento.

- 4) La utilización de herramientas de gestión de proyectos y una metodología ágil consignan la atención del cliente, de modo que esté siempre atento a cuanto sucede, de modo que informe de manera oportuna acerca posibles fallas. La utilización de la herramienta Azure DevOps permite llevar un control con los interesados del proyecto, para que ellos puedan observar lo qué sucede. A pesar de que la institución no formó parte de los usuarios de la herramienta, los tutores relacionados con el proyecto siempre contaron con la posibilidad acceso, lo cual a su vez les proporcionó la opción para retroalimentar oportunamente. El marco de trabajo Scrum brindó al cliente la oportunidad de estar presente en distintos eventos (tal como el Sprint Review) y así pudiese verificar que lo desarrollado cumpliera con sus expectativas y los requerimientos. Además, la presencia de un Product Owner agilizó el proceso de comunicación entre el equipo de trabajo y el cliente, con miras a satisfacer las necesidades surgidas durante el desarrollo de la propuesta. Con lo anterior confirma el cumplimiento del objetivo en un cien por ciento.

6.2 Recomendaciones

Luego de desarrollar la propuesta del sistema académico y obtener las conclusiones del proceso, es posible ofrecer una serie de recomendaciones. A pesar de haberse gestionado la propuesta de dos módulos principales, se considera necesario llevar a cabo el proceso de desarrollo de los módulos restantes, tales como mensajería, repositorios y reportes, pues son parte fundamental del funcionamiento completo de un sistema académico. Además, lo ideal es hacerlo siguiendo los lineamientos llevados en el proceso anterior, de manera que sea posible mantener un

equilibrio entre el proceso de desarrollo y la aceptación de la propuesta, incluyendo dichos módulos. Así las cosas, surgen las siguientes recomendaciones:

- 1) Kamuk School debe mantener una documentación formal, donde establezca los requerimientos necesarios para el buen funcionamiento de un sistema académico, lo cual evitaría incurrir en gastos innecesarios en el futuro, hecho ocurrido con los sistemas SIRA y Class. Aplica la misma recomendación para cualquier otro tipo de sistema que deseen implementar, pues el costo de los cambios en producción es mucho mayor, sin importar si es un software hecho a la medida o la compra de un servicio. Tal documentación puede basarse en aquella generada durante el proceso de desarrollo del este proyecto.
- 2) A pesar de contar con un reglamento, algunos de los procesos académicos deben ser adaptados a la herramienta utilizada, ya sea el empleo de un software o del uso de papel para gestión manual. La institución debe definir con claridad la forma de gestionarse académicamente y utilizar tecnologías realmente funcionales para una gestión definida, con lo cual se modificaría lo menos posible su modalidad de trabajo.
- 3) Es deseable seguir el modelo de trabajo con Scrum y la herramienta Azure DevOps para el desarrollo de los módulos restantes y así producir un entregable completo, con un estándar definido. Incurrir en el uso de otros métodos o herramientas podría generar situaciones similares a las dadas con SIRA, tales como necesidades sin cumplir. Por lo tanto, se brinda la documentación del éxito del proyecto y queda a disposición de la institución para su empleo en los módulos restantes.

- 4) Kamuk School debe contar con un encargado de monitorear los proyectos institucionales. La coordinación entre dicha persona y el equipo de proyecto resulta esencial para identificar a tiempo las áreas de mejora. Esto evitaría los retrasos por cambios a realizar durante las etapas finales de entrega de un proyecto.
- 5) Finalmente, el uso de la herramienta DevOps, o de cualquier otra herramienta de gestión que incluya a la institución, es de gran importancia. La única manera de monitorear actualmente un proyecto es depender de las citas generadas por el equipo de trabajo, por lo tanto, las revisiones del progreso no son accesibles en tiempo real. El uso de una herramienta como DevOps permitirá a la institución la revisión del progreso, informar los cambios requeridos y conocer en todo momento el progreso del desarrollo de cualquier proyecto.

Bibliografía

Atlassian. (2019). Jira | Software de seguimiento de proyectos e incidencias.

Recuperado 19 de mayo de 2019, de Atlassian website:

<https://es.atlassian.com/software/jira>

Azure DevOps Services | Microsoft Azure. (2019). Recuperado 19 de mayo de 2019, de

<https://azure.microsoft.com/en-us/services/devops/>

Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A Software Architect's Perspective*.

Addison-Wesley Professional.

Beck, Kent, Beedle, Mike, van Bennekum, Arie, Cockburn, Alistair, Cunningham, Ward,

Fowler, Martin, ... Thomas, Dave. (2001). Principios del Manifiesto Ágil.

Recuperado 14 de mayo de 2019, de

<https://agilemanifesto.org/iso/es/principles.html>

Cordero, Z. R. V. (2009). La Investigación aplicada: Una forma de conocer las

realidades con evidencia científica. *Revista Educación*, 33(1), 155-165.

<https://doi.org/10.15517/revedu.v33i1.538>

Dimes, T. (2015). *Conceptos Básicos De Scrum: Desarrollo De Software Agile Y*

Manejo De Proyectos Agile. Babelcube Inc.

Fuentes, J. R. L. (2015). *Desarrollo de Software ÁGIL: Extreme Programming y Scrum*.

IT Campus Academy.

García González, M. O., Sepúlveda Castaño, J. M., & Montoya Suárez, L. M. (2017).

Metodologías Ágiles de Desarrollo, un Caso de Aplicación en Medellín, Colombia

2016 - 2017. *Agile Development Methodology, an Application Case in Medellin*,

Colombia 2016 - 2017., (18), 58-63. <https://doi.org/10.21501/21454086.2521>

- Hernández Sampieri, Roberto, Fernández Collado, Carlos, & Baptista Lucio, María del Pilar. (2014). *Metodología de la Investigación* (Sexta). México D.F.: McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.
- Joyanes Aguilar, Luis. (2015). *Sistemas de Información en la Empresa. El impacto de la nube, la movilidad y los medios sociales*. (Primera). Alfaomega Grupo Editor, S.A. de C.V., México.
- Kamuk School. (2016). Nosotros | Kamuk School. Recuperado 27 de febrero de 2019, de Nosotros website: <http://www.kamukschool.ed.cr/nosotros/>
- Kanban Tool. (2019). Metodología Kanban | Kanban Tool. Recuperado 14 de mayo de 2019, de <https://kanbantool.com/es/metodologia-kanban>
- Letelier, P., & Letelier, P. (2006, abril 15). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP) [Artículo]. Recuperado 14 de mayo de 2019, de www.cyta.com.ar/ta0502/v5n2a1.htm website: http://www.cyta.com.ar/ta0502/b_v5n2a1.htm
- Montero, B. M., Cevallos, H. V., & Cuesta, J. D. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espiraes revista multidisciplinaria de investigación*, 2(17). <https://doi.org/10.31876/re.v2i17.269>
- Pantaleo, Guillermo, R., Ludmila. (2016). *Ingeniería de Software*. Alfaomega Grupo Editor, S.A. de C.V., México.
- Pazos Jiménez, Ethel, & Gutiérrez M., Federico. (2012). *Manual para el curso de Métodos de Investigación* (Primera). San José Costa Rica: SECADE.
- Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit: An Agile Toolkit*. Addison-Wesley.

Pressman, Roger S. (2010). *Ingeniería del software. Un enfoque práctico* (Séptima).

México: MCGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.

RAE. (2019). «Diccionario de la lengua española» - Edición del Tricentenario.

Recuperado 1 de mayo de 2019, de «Diccionario de la lengua española» -

Edición del Tricentenario website: <http://dle.rae.es/>

Schwaber, K., & Sutherland, J. (2017). *La Guía de Scrum*. Recuperado de

<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf#zoom=100>

Smith, Hein. (2018). *SCRUM. The Ultimate Beginner's Guide to Learn and Master Scrum Agile Framework*.

Software de administración de proyectos | Microsoft Project. (2019). Recuperado 19 de mayo de 2019, de <https://products.office.com/es-ww/project/project-and-portfolio-management-software>

Verona, J. (2016). *Practical DevOps*. Recuperado de

<https://search.ebscohost.com.uh.remotexs.xyz/login.aspx?direct=true&db=e000xww&AN=1172045&lang=es&site=eds-live&scope=site>

What is Scrum? (s. f.). Recuperado 19 de mayo de 2019, de Scrum.org website:

<https://www.scrum.org/index.php/resources/what-is-scrum>

Anexos

Anexo 1**Certificación Professional Scrum Master I**

Anexo 2

Cuestionario 1: Formulario de Nivel Académico

Le damos la más cordial bienvenida al formulario del nivel académico. Este tiene como objetivo dar a conocer aspectos importantes acerca del empleo del sistema de gestión académica para el centro educativo Kamuk School.

Este formulario es parte de un proyecto de graduación de seis estudiantes de la carrera de Ingeniería Informática de la Universidad Hispanoamericana en el 2019.

Agradecemos toda su colaboración para completar este formulario con un total de diez preguntas.



*



Por favor, le solicitamos responder las siguientes preguntas con el mayor detalle posible.

1. ¿Cuál es la importancia del control académico de los estudiantes?

2. ¿Cuál es la importancia de la planeación institucional anual? *

3. ¿Por qué se necesita un software académico? *

4. ¿Cuáles son los beneficios para Kamuk School al contar con un software académico? *

5. ¿Qué beneficios reporta el sistema académico SIRA para Kamuk School? *

6. ¿De qué funcionalidades carece el sistema académico SIRA en Kamuk School?*

7. ¿Cuál es el estado actual de SIRA con respecto a Kamuk School? *

8. ¿Qué beneficios reporta el sistema académico Class para Kamuk School? *

9. ¿De qué funcionalidades carece el sistema académico Class? *

10. ¿Qué características debe cumplir un sistema académico para Kamuk School?*

Con la tecnología de



Anexo 3

Cuestionario 2: Formulario de Nivel Tecnológico

Le damos la más cordial bienvenida al formulario del nivel tecnológico, cuyo objetivo consiste en dar a conocer aspectos importantes sobre el empleo de los sistemas e infraestructura tecnológica en el centro educativo Kamuk School.

Este formulario forma parte de un proyecto de graduación de seis estudiantes de la carrera de Ingeniería Informática de la Universidad Hispanoamericana en el año 2019.

Agradecemos toda su colaboración en completar este formulario, el cual consta de un total de 7 preguntas.



Por favor, le solicitamos responder las siguientes preguntas con el mayor detalle posible.

1. ¿Cuenta Kamuk School con alguna guía para contratación o compra de servicios tecnológicos? En caso afirmativo, indique qué se señala en ella*

2. ¿Cuáles son las características de hardware y software del servidor en el que se podría implementar la propuesta del software académico? *

3. ¿Cuenta Kamuk con licenciamiento de base de datos? De ser así, indique cuál licenciamiento y qué herramienta se utiliza. *

4. ¿Cuenta Kamuk con licenciamiento para plataformas Windows (Windows Server)?

De ser así, especifique con mayor detalle. *

5. ¿Se cuenta con contrato de algún VPS o cualquier otro tipo de servicio en línea? De ser así, especifique con mayor detalle.

6. ¿Cuenta Kamuk School con alguna guía de gestión de proyectos tecnológicos? En caso de contar con alguna, qué indica la misma. *

7. ¿Cuenta Kamuk School con alguna guía de estándares de calidad? En caso de contar con alguna, qué indique cuál. *

Con la tecnología de



Anexo 4

Cuestionario 3: Formulario de Percepción

Le damos la más cordial bienvenida al formulario del nivel de percepción. Este tiene como objetivo dar a conocer el grado de satisfacción y percepción acerca del sistema de gestión académica en el centro educativo Kamuk School.

Este formulario forma parte de un proyecto de investigación de seis estudiantes de la carrera de Ingeniería Informática de la Universidad Hispanoamericana en el año 2019.

Agradecemos toda su colaboración para completar este formulario, el cual cuenta con un total de diez preguntas.



Por favor, le solicitamos responder a las siguientes preguntas con el mayor posible.

1. ¿Qué función desempeña usted en la institución? *

Por favor marque solo un óvalo.

☐

Profesor

☐

Personal administrativo

☐

Gerencia

☐

Otros: _____

2. ¿Trabajó con el Sistema SIRA? Si contesta no, favor ir a la pregunta 6 *

Por favor marque solo un óvalo

☐ Si

☐ No

3. En una escala del uno al cinco, ¿cómo calificaría SIRA?

Por favor marque solo un óvalo

1

2

3

4

5

☐

☐

☐

☐

☐

Muy malo

Muy bueno

4. En una escala del uno al cinco, ¿qué tanto se podría mejorar la funcionalidad del sistema SIRA?

Por favor marque solo un óvalo

1

2

3

4

5

☐

☐

☐

☐

☐

Muy poco por mejorar

Mucho por mejorar

5. ¿Qué sistema académico considera de mayor utilidad y eficiencia, SIRA o Class?

Marque solo un óvalo.

☐ SIRA

☐ CLASS

6. En una escala del uno al cinco, ¿cómo calificaría el sistema Class? *

Por favor marque solo un óvalo

1	2	3	4	5
☐	☐	☐	☐	☐
Muy malo			Muy bueno	

7. En una escala del uno al cinco, ¿qué tanto considera usted que podría mejorar la funcionalidad del sistema Class? *

Por favor marque solo un óvalo

1	2	3	4	5
☐	☐	☐	☐	☐
Muy poco			Mucho	

8. ¿Utilizaría un nuevo sistema académico que combine las mejores funcionalidades de SIRA y Class? Omitir esta pregunta si nunca trabajó con el sistema académico SIRA

Por favor marque solo un óvalo

☐ Si

☐ No

9. ¿Cuáles serían las tres principales características con las que debería contar un sistema académico? *

Selecciona todas las opciones que correspondan.

- ☐ Adaptabilidad al cambio
- ☐ Agradable a la vista
- ☐ Facilidad de uso
- ☐ Multiplataformas (teléfonos, tabletas, computadoras, entre otros)
- ☐ Rendimiento
- ☐ Seguridad

10. ¿Cuáles serían las tres principales características con las que debería contar un sistema académico? *Marca solo un óvalo.

- ☐ Si
- ☐ No

Con la tecnología de



Anexo 5**Carta de aceptación de la propuesta de Kamuk School**

24 de julio del 2019, San José

Señores de la Universidad Hispanoamericana:

Por este medio, yo, Evander Vallecillo Rizo, cédula de identidad 7-0216-0484, informo que el estudiante Mario Gerardo Angulo Cordero, cédula de identidad 1-1527-0667, completó con satisfacción de la institución la propuesta de desarrollo de sistema académico en Kamuk School. Confirmando que dicha persona tuvo constante comunicación con nosotros y culminó exitosamente la propuesta y queda a decisión de la institución la implementación de ésta. Dicho proceso toma un tiempo considerable de análisis con el equipo institucional y se informará al estudiante dicha decisión en caso de adoptarse la propuesta.

Respetuosamente,



Evander Vallecillo Rizo

7-0216-0484

