



Universidad
Técnica de
Cotopaxi

UNIVERSIDAD TÉCNICA DE COTOPAXI

UNIDAD ACADÉMICA DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

CARRERA DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES

PROYECTO DE INVESTIGACIÓN

TEMA:

“ADAPTACIÓN DE LA METODOLOGÍA DE DESARROLLO DE SOFTWARE SCRUM A EQUIPOS DE UN SOLO PROGRAMADOR. CASO: DEPARTAMENTO DE SISTEMAS DE LA COOPERATIVA DE AHORRO Y CRÉDITO ANDINA LTDA.”

Proyecto de Investigación presentado previo a la obtención del Título de Ingeniería en Informática y Sistemas Computacionales

Autor:

Jairo Fernando Arias Herrera

Directora:

Ing. MSc. Verónica Consuelo Tapia Cerda

Latacunga – Ecuador

2016



APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la Unidad Académica de Ciencias de la Ingeniería y Aplicadas.; por cuanto, el postulante **Arias Herrera Jairo Fernando** con el título de Proyecto de Investigación: “**ADAPTACIÓN DE LA METODOLOGÍA DE DESARROLLO DE SOFTWARE SCRUM A EQUIPOS DE UN SOLO PROGRAMADOR. CASO: DEPARTAMENTO DE SISTEMAS DE LA COOPERATIVA DE AHORRO Y CRÉDITO ANDINA LTDA.**” han considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación de Proyecto.

Por lo antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

Latacunga, Julio del 2016

Para constancia firman:

Lector 1 (Presidente)
Nombre: PhD. Gustavo Rodríguez
CC: 175700135-7

Lector 2
Nombre: Ing. MSc. Silvia Bravo
CC: 050243712-2

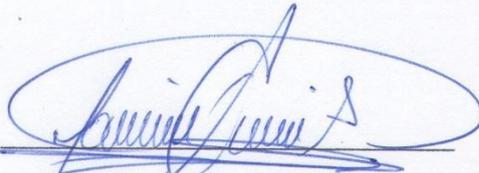
Lector 3
Nombre: Ing. MSc. José Cadena
CC: 050155279-8



DECLARACIÓN DE AUTORÍA

Yo **Jairo Fernando Arias Herrera**, declaro ser autor del presente proyecto de investigación: **“ADAPTACIÓN DE LA METODOLOGÍA DE DESARROLLO DE SOFTWARE SCRUM A EQUIPOS DE UN SOLO PROGRAMADOR. CASO: DEPARTAMENTO DE SISTEMAS DE LA COOPERATIVA DE AHORRO Y CRÉDITO ANDINA LTDA.”**, Siendo la **Ing. MSc. Verónica Consuelo Tapia Cerda** tutora del presente trabajo; y eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.



Jairo Fernando Arias Herrera
C.I.: 050315376-9



AVAL DEL TUTOR DE PROYECTO DE INVESTIGACIÓN

En calidad de Tutor del Trabajo de Investigación sobre el título:

“ADAPTACIÓN DE LA METODOLOGÍA DE DESARROLLO DE SOFTWARE SCRUM A EQUIPOS DE UN SOLO PROGRAMADOR. CASO: DEPARTAMENTO DE SISTEMAS DE LA COOPERATIVA DE AHORRO Y CRÉDITO ANDINA LTDA.”, de **Arias Herrera Jairo Fernando**, de la carrera de Ingeniería en Informática y Sistemas Computacionales, considero que dicho Informe Investigativo cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Honorable Consejo Académico de la Unidad Académica de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

Latacunga, Julio del 2016

Ing. MSc. Verónica Consuelo Tapia Cerda

AGRADECIMIENTO

A mi querida Universidad Técnica de Cotopaxi que me abrió sus puertas, brindándome la oportunidad de realizarme como profesional, del mismo modo a los docentes quienes me han compartido sus amplios conocimientos los mismos que me serán útiles en el transcurso de mi profesión.

De manera especial extendiendo mi agradecimiento a la Cooperativa de Ahorro y Crédito Andina Ltda. Por su colaboración y apertura para la aplicación de mis conocimientos en el presente proyecto y por la confianza brindada en el trabajo que día a día lo vengo desempeñando de manera responsable y eficiente.

Fernando

DEDICATORIA

Por sus incalculables esfuerzos y su amor infinito, por haber hecho todo en la vida por impulsarme a lograr mis sueños; dedico este proyecto a mi madre YOLANDA ARIAS, quien ha sido la guía y el pilar fundamental en mi vida inculcándome valores y principios que han hecho de mí una persona de bien.

Gracias a sus enseñanzas, palabras de aliento y su excelente manera de instruirme he sabido afrontar los obstáculos de la vida y he alcanzado mis metas.

A ti madre mi amor y gratitud por siempre.

Fernando

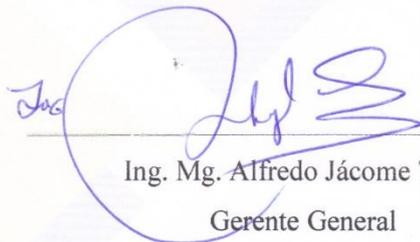
CERTIFICADO

Latacunga, 13 de Julio del 2016

A petición verbal del Sr. **ARIAS HERRERA JAIRO FERNANDO**, portador de la cédula de ciudadanía 050315376-9, tengo a bien **CERTIFICAR** que desarrolló su Proyecto de Investigación con el tema: “ADAPTACIÓN DE LA METODOLOGÍA DE DESARROLLO DE SOFTWARE SCRUM A EQUIPOS DE UN SOLO PROGRAMADOR” en el Departamento de Sistemas de la Cooperativa de Ahorro y Crédito ANDINA Ltda., proyecto realizado desde el 02 de Mayo del 2016 hasta el 13 de Julio del 2016.

Es todo en cuanto puedo certificar en honor a la verdad, autorizando al interesado hacer uso del presente como creyere conveniente.

Atentamente,



Ing. Mg. Alfredo Jácome Tapia

Gerente General

Cooperativa de Ahorro y Crédito ANDINA Ltda.



ANDINA Ltda.
COOPERATIVA DE AHORRO Y CRÉDITO
GERENCIA

Somos más... más apoyo, más impulso!

Matriz (Sector El Carmen): Av. Benjamín Terán 319 y Antonia Vela Telf.: (03) 2802 850 / 2802 754

Agencia (Sector La Merced): Quijano y Ordóñez y Félix Valencia Tel.: (03) 2802 150 / 2800 544

Latacunga - Ecuador

ÍNDICE

APROBACIÓN DEL TRIBUNAL DE TITULACIÓN	¡ERROR! MARCADOR NO DEFINIDO.
DECLARACIÓN DE AUTORÍA.....	¡ERROR! MARCADOR NO DEFINIDO.
AVAL DEL TUTOR DE PROYECTO DE INVESTIGACIÓN....	¡ERROR! MARCADOR NO DEFINIDO.
AGRADECIMIENTO.....	IV
DEDICATORIA	V
CERTIFICADO DE IMPLEMENTACIÓN.....	VI
RESUMEN	X
ABSTRACT	XI
AVAL DE TRADUCCIÓN.....	¡ERROR! MARCADOR NO DEFINIDO.
1. INFORMACIÓN GENERAL	1
2. RESUMEN DEL PROYECTO	2
3. JUSTIFICACIÓN DEL PROYECTO.....	3
4. BENEFICIARIOS DEL PROYECTO	5
5. EL PROBLEMA DE INVESTIGACIÓN	6
6. OBJETIVOS	7
6.1. OBJETIVO GENERAL	7
6.2. OBJETIVOS ESPECÍFICOS.....	7
7. ACTIVIDADES Y SISTEMA DE TAREAS EN RELACIÓN A LOS OBJETIVOS PLANTEADOS.....	8
8. FUNDAMENTACIÓN CIENTÍFICO TÉCNICA	10
8.1. ANTECEDENTES	10
8.2. REFERENTES TEÓRICOS	17
8.3. ELEMENTOS TEÓRICOS CONCEPTUALES	20
8.3.1. <i>Metodología Ágil de desarrollo de software Scrum.....</i>	<i>20</i>
8.3.2. <i>Sistema Financiero Econx.....</i>	<i>25</i>

8.3.3.	<i>Arquitectura Cliente – Servidor</i>	26
8.3.4.	<i>Php – Html</i>	26
8.3.5.	<i>JavaScript - Ajax</i>	27
8.3.6.	<i>Informix</i>	27
8.3.7.	<i>Putty</i>	27
8.3.8.	<i>Notepad++</i>	28
8.3.9.	<i>Beyond Compare</i>	28
9.	PREGUNTAS CIENTÍFICAS O HIPÓTESIS	28
9.1.	HIPÓTESIS.....	28
9.1.1.	<i>Variable independiente</i>	29
9.1.2.	<i>Variable dependiente</i>	29
10.	METODOLOGÍAS Y DISEÑO EXPERIMENTAL	29
10.1.	TIPOS DE INVESTIGACIÓN	29
10.1.1.	<i>Investigación Documental</i>	29
10.1.2.	<i>Investigación Aplicada</i>	29
10.1.3.	<i>Investigación de Campo</i>	30
10.2.	MÉTODOS DE INVESTIGACIÓN	30
10.2.1.	<i>Método Analítico</i>	30
10.2.2.	<i>Método de la Abstracción</i>	30
10.3.	TÉCNICAS DE INVESTIGACIÓN	30
10.3.1.	<i>Técnica de la Observación Directa</i>	31
10.3.2.	<i>Técnica de la Entrevista</i>	31
10.4.	INSTRUMENTOS DE INVESTIGACIÓN.....	31
10.4.1.	<i>Ficha de Observación</i>	31
10.4.2.	<i>Cuestionario de Entrevista</i>	31
10.5.	IMPLEMENTACIÓN DE LA PROPUESTA	31
11.	ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS	44
12.	IMPACTOS (TÉCNICOS, SOCIALES, AMBIENTALES O ECONÓMICOS) .	48
13.	PRESUPUESTO PARA LA PROPUESTA DEL PROYECTO	49
14.	CONCLUSIONES Y RECOMENDACIONES	51
14.1.	CONCLUSIONES.....	51

14.2. RECOMENDACIONES	52
15. BIBLIOGRAFÍA	54
16. ANEXOS.....	57

ÍNDICE DE TABLAS

Tabla 1: Sistema de Tareas en relación a los objetivos planteados	8
Tabla 2: Rol Product Owner	33
Tabla 3: Rol Scrum Master.....	34
Tabla 4: Rol Scrum Team.....	35
Tabla 5: Herramienta Historia de Usuario.....	36
Tabla 6: Artefacto Product Backlog	36
Tabla 7: Artefacto Sprint Backlog.....	37
Tabla 8: Artefacto Product Increment	37
Tabla 9: Evento Sprint Planning.....	38
Tabla 10: Evento Daily Meeting	39
Tabla 11: Evento Sprint Planning.....	39
Tabla 12: Evento Sprint Retrospective.....	40
Tabla 13: Evento Refinamiento del Backlog.....	40
Tabla 14: Scrum Técnico vs Scrum Adaptado	41
Tabla 15: Costos directos	49
Tabla 16: Costos Indirectos	49
Tabla 17: Total detallado.....	50

ÍNDICE DE IMÁGENES

Imagen 1: Proceso de Scrum adaptado a equipos de un solo programador	41
---	----

UNIVERSIDAD TÉCNICA DE COTOPAXI

UNIDAD ACADÉMICA DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

TITULO: “ADAPTACIÓN DE LA METODOLOGÍA DE DESARROLLO DE SOFTWARE SCRUM A EQUIPOS DE UN SOLO PROGRAMADOR.”

Autor: Jairo Fernando Arias Herrera

RESUMEN

El proyecto de investigación “Adaptación de la Metodología de Desarrollo de Software Scrum a Equipos de un solo programador”, tiene como objetivo adaptar el marco de trabajo de Scrum mediante la aplicación de las mejores prácticas de su modelo de procesos, para la implementación de soluciones informáticas en el Departamento de Sistemas de la Cooperativa de Ahorro y Crédito Andina Ltda. La problemática evidenciada se centra en el proceso tradicional que se emplea para el desarrollo de software, debido a que no se sigue un proceso sistemático que permita asegurar la eficacia del software implementado, ocasionando que el software puesto en producción sea poco funcional. En el proyecto se aplicó el método de la Abstracción, con el cual se abstrajo las características esenciales de Scrum para su análisis y posterior aplicación en un caso práctico. Habiendo obtenido como resultado definir un marco de trabajo enfocado a equipos con un solo programador, dotando de un proceso sistemático fundamentado en las reglas y valores sobre las cuales esta cimentado Scrum. Resultados que fueron aplicados en el desarrollo de un aplicativo denominado “Control de Denominaciones de Dinero” para el área de cajas de la Cooperativa, producto de software con un alto nivel funcional y puesto en producción. Demostrando de esta manera que Scrum es flexible y se adapta a diferentes contextos donde se lo aplique, manteniendo siempre su esencia y agilidad. El principal aporte del proyecto recae en el factor económico, ya que la Cooperativa al contar con un proceso ágil que gestione y asegure la funcionalidad de los productos de software no incurrirá en altos costos de adquisición de software a terceras personas. Finalmente se hace mención el cumplimiento de los objetivos planteados al inicio del proyecto, ya que este se fundamentó en una base teórica que proporcionó un conocimiento profundo sobre el objeto de estudio, se conoció la situación actual del lugar donde se evidenciaba la problemática mediante la buena predisposición del Jefe de Sistemas que nos concedió una entrevista, se definió las características más relevantes del marco de trabajo de Scrum que junto con sus reglas, valores y principios se los aplicó al desarrollo de un caso práctico.

Palabras clave: Adaptar, Scrum, proceso, desarrollo de software, valores

TECHNICAL UNIVERSITY OF COTOPAXI

ACADEMIC UNIT OF ENGINEERING AND APPLIED SCIENCES

TITLE: “ADAPTATION OF THE METHODOLOGY SOFTWARE DEVELOPMENT SCRUM TEAMS A SINGLE PROGRAMMER”

Author: **Jairo Fernando Arias Herrera**

ABSTRACT

The research project "Adaptation of the Methodology Software Development Scrum teams a single programmer", aims to adapt the framework of Scrum by applying best practices in its process model for implementing solutions computer in the Department of Systems credit union Andina Ltda. the evidenced issue focuses on the traditional process used for software development, because a systematic process that would ensure the effectiveness of the software didn't follow implemented, causing the software is putting into production very functional. In the project the method of abstraction, with which the essential characteristics of Scrum for analysis and subsequent application in a case study abstracted applied. It obtained as a result defined a framework focused on computers with a single programmer, providing a systematic process based on the rules and values on which is founded Scrum. Results were applied in the development of an application called "Control of Appellations of Money" for the area of cash registers of Cooperative software product with high functional level and puts into production. Thus demonstrating that Scrum is flexible and adapts to different contexts where it is applied, while maintaining its essence and agility. The main contribution of the project lies in the economic factor, since the cooperative to have a streamlined process to manage and ensure the functionality of the software products will not incur high costs of acquiring software to third parties. Finally mention meeting the goals set at the beginning of the project is done, as this was based on a theoretical basis which provided a deep knowledge on the object of study, the current situation where the problem was evident met through good predisposition Head of Systems which granted us an interview, the most important characteristics of Scrum framework together with its rules, values and principles are applied to the development of a case was defined.

Keywords: Adapt, Scrum, process, software development, values



Universidad
Técnica de
Cotopaxi

CENTRO CULTURAL DE IDIOMAS

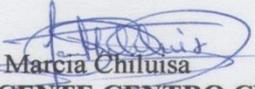
AVAL DE TRADUCCIÓN

En calidad de Docente del Idioma Inglés del Centro Cultural de Idiomas de la Universidad Técnica de Cotopaxi; en forma legal CERTIFICO que: La traducción del resumen de tesis al Idioma Inglés presentado por el señor Egresado de la Carrera de Ingeniería en Informática y Sistemas Computacionales: **ARIAS HERRERA JAIRO FERNANDO**, cuyo título versa “**ADAPTACIÓN DE LA METODOLOGÍA DE DESARROLLO DE SOFTWARE SCRUM A EQUIPOS DE UN SOLO PROGRAMADOR. CASO: DEPARTAMENTO DE SISTEMAS DE LA COOPERATIVA DE AHORRO Y CRÉDITO ANDINA LTDA.**”, lo realizó bajo mi supervisión y cumple con una correcta estructura gramatical del Idioma.

Es todo cuanto puedo certificar en honor a la verdad y autorizo al peticionario hacer uso del presente certificado de la manera ética que estimaren conveniente.

Latacunga, Julio del 2016

Atentamente,


Lic. Marcia Chiluisa
DOCENTE CENTRO CULTURAL DE IDIOMAS
C.C. 0502214307

www.utc.edu.ec

Av. Simón Rodríguez s/n Barrio El Ejido / San Felipe. Tel: (03) 2252346 - 2252307 - 2252205
VIII

UNIVERSIDAD TÉCNICA DE COTOPAXI

UNIDAD ACADÉMICA DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

CARRERA DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES

1. INFORMACIÓN GENERAL

Título del Proyecto:

Adaptación de la metodología de desarrollo de software Scrum a equipos de un solo programador. Caso: Departamento de Sistemas de la Cooperativa de Ahorro y Crédito Andina Ltda.

Fecha de inicio:

02 de Mayo del 2016

Fecha finalización:

11 de Julio del 2016

Lugar de ejecución:

El Carmen - La Matriz – Latacunga – Cotopaxi – Zona 3 – Departamento de Sistemas de la Cooperativa de Ahorro y Crédito ANDINA Ltda.

Unidad Académica que auspicia:

Unidad Académica de Ciencias de la Ingeniería y Aplicadas

Carrera que auspicia:

Ingeniería en Informática y Sistemas Computacionales

Equipo de trabajo:

Tutor de Titulación:

Ing. MSc. Verónica Consuelo Tapia Cerda

Investigador:

Jairo Fernando Arias Herrera

Área de conocimiento:

Gestión de proyectos, calidad y configuración de software

Línea de investigación

Tecnologías de la información y comunicación (TICs) y diseño gráfico

Sublíneas de Investigación

2. RESUMEN DEL PROYECTO

En la actualidad las grandes empresas disponen de equipos completos de desarrolladores de software. El entorno, en donde se lleva a cabo el proceso de desarrollo es dinámico y altamente productivo ya que se aplican metodologías que define un marco de trabajo óptimo para el equipo de desarrollo. El objeto de aplicar una metodología ágil es fortalecer el crecimiento empresarial, a medida que las empresas crecen tienden a innovar sus procesos para la producción de bienes o servicios mediante la implementación o actualización continua de sus sistemas informáticos.

La realidad en las empresas que no cuentan con un equipo completo para el desarrollo de software es distinta, internamente deben adaptar una metodología de desarrollo que le permita con el reducido talento humano sacar adelante proyectos de software que requiera la empresa, asegurando que este cumpla con los requerimientos solicitados y sea puesto en producción y alcance su propósito. El inconveniente de las metodologías ágiles que se emplean actualmente para el desarrollo de software, radica en que la mayor parte de ellas contemplan un número considerable de personas que integran los equipos de desarrollo, dificultando de esta manera que aquellos más reducidos puedan aplicarlas en su entorno.

El proyecto de investigación pretende establecer una alternativa asequible a los equipos que cuenten con un solo programador, facilitando la aplicación de una metodología ágil y

flexible para el desarrollo de software en su entorno laboral; es así que se ha propuesto adaptar el marco de trabajo de Scrum, para lo cual se realizará un análisis del proceso que engloba esta metodología, definiendo claramente sus roles, eventos y artefactos, las mismas que sean aplicables por el equipo.

El campo de investigación es amplio, en vista que Scrum está enmarcada en un conjunto de valores y principios para trabajar colaborativamente dentro de un equipo de desarrollo altamente productivo. Scrum aplica el modelo iterativo incremental el cual divide el proyecto en iteraciones que proporcionan un entregable parcial del producto final al cliente en un lapso corto de 1 a 2 semanas. El proceso de desarrollo de software con Scrum inicia con la definición de roles (Product Owner, Scrum Master y Scrum Team), se genera artefactos (Product Backlog, Sprint Backlog y el Product Increment), y se realizan eventos constantes (Sprint Planning, Daily Meeting, Sprint Review, Sprint Retrospective, Refinement). Todo este proceso que engloba Scrum sumado a los principios del manifiesto ágil hace que sea una de las metodologías más usadas y adaptadas por los equipos de desarrollo obteniendo resultados de alta calidad.

La visión del proyecto de investigación es crear una alternativa de la metodología de desarrollo de software Scrum, la misma que se consolide como una opción a ser adoptada y aplicada por empresas con equipos de desarrollo con un solo programador. El caso particular donde se aplicará el proyecto es al equipo del Departamento de Sistemas de la Cooperativa de Ahorro y Crédito ANDINA Ltda., que cuenta con un solo programador. Del cual se espera implementar software de calidad bajo un contexto ágil, que sea funcional y que reduzca la inversión en cuanto a la adquisición de software de terceras personas.

Palabras clave: Scrum, Equipo de Desarrollo, calidad.

3. JUSTIFICACIÓN DEL PROYECTO

La importancia de aplicar una metodología ágil para el desarrollo de software es proveer al equipo de desarrolladores un proceso sistemático para implementar software de calidad y a su vez permita sistematizar procesos agilizando favorablemente la producción de servicios o productos de una empresa, beneficiándola en distintos ámbitos como la reducción de costos, minimizar riesgos y optimizar recursos.

El proceso que involucra el desarrollo de software bajo una metodología ágil permite obtener resultados enmarcados en la calidad, funcionamiento y satisfacción de los usuarios

finales, esto se consigue mediante la participación activa del equipo de desarrollo en la cual cada uno de sus integrantes cumple un determinado rol y comparten un objetivo en común. Por tal motivo se deduce proponer una alternativa que surja a partir de una propuesta de variante de la metodología Scrum a ser adaptable a equipos con un solo programador, que mediante la definición de roles, eventos y artefactos más representativas de Scrum facilite al programador hacer uso del marco de trabajo en el proceso de nuevos productos de software.

Se ha analizado la literatura acerca del objeto de estudio y se puede determinar que existe suficiente información bibliográfica para llevar a cabo el presente proyecto de investigación, que está enfocado a equipos de desarrollo de software con un programador que pueden adoptar esta alternativa metodológica para la implementación de nuevos productos de software, pero en forma particular se lo aplicará directamente en el Departamento de Sistemas de la Cooperativa de Ahorro y Crédito ANDINA Ltda., en la cual se ha constatado que no se ha propuesto ningún estudio ni análisis acerca de la temática planteada, por lo que es de gran interés y relevancia para la Cooperativa.

La contribución que generará la adaptación de la metodología Scrum a equipos de un solo programador permitirá definir roles (Product Owner, Scrum Master, Scrum Team), eventos como (Daily Meeting, Sprint Review) y generar artefactos como (Product Backlog, Sprint Backlog, Product increment), proporcionando al programador un ambiente de desarrollo sistemático que le ayude a implementar software eficiente en beneficio de la empresa, contribuyendo también a minimizar el costo económico de la implementación.

El aplicativo para el área de cajas de la Cooperativa de Ahorro y Crédito ANDINA Ltda., que se pretende desarrollar bajo la adaptación del marco de trabajo de Scrum, reunirá características similares a la plataforma en la cual será embebido, con la diferencia que éste será implementado bajo un proceso sistemático acoplado a una planificación de actividades, levantamiento de información en formatos específicos, generando documentación técnica, aplicando casos de prueba y demás etapas de Scrum.

Para realizar el proyecto de investigación se cuenta con el respectivo respaldo de la Gerencia General de la Cooperativa y la Jefatura de Sistemas, permitiendo utilizar los recursos informáticos como servidores, sistemas, base de datos a prueba vía VPN (red privada virtual), y demás recursos que demande la ejecución del proyecto de investigación.

Los costes en los que incurrirá el proyecto de investigación serán autofinanciados, sin embargo no se escatima que se pueda recibir aportes económicos de terceras personas, los mismos que serán invertidos para aplicar el proyecto de investigación de una mejor manera.

El conocimiento que se tiene en cuanto al tema planteado se lo considera en un nivel medio, en vista que en el transcurso de la formación adquirida en la Universidad se ha trabajado con la Metodología Scrum en el desarrollo de proyectos conociendo el proceso que conlleva la aplicación de Scrum, además se cuenta con la bibliografía necesaria tanto en libros como en la web para llevar a cabo con éxito el proyecto.

Para la realización y seguimiento del desarrollo del proyecto de investigación se cuenta con la asesoría de la Ing. MgC. Verónica Tapia, profesional en el área de Ingeniería de Software, quien además tiene un amplio conocimiento de la aplicación de la metodología Scrum.

El proyecto de investigación es factible en todos sus aspectos tanto económico, talento humano necesario, infraestructura tecnológica, aval de las autoridades de la Cooperativa Andina, existe información documentada y lo más relevante se cuenta con el apoyo de los involucrados (gerente, administrador del sistema, colaboradores). Por tal razón se estima que no existirá ningún tipo de inconveniente en la realización del proyecto.

4. BENEFICIARIOS DEL PROYECTO

Los beneficiarios directos de la aplicación del proyecto de investigación son los miembros que conforman el equipo de desarrollo del departamento de sistemas de la Cooperativa de Ahorro y Crédito ANDINA Ltda., en particular el Analista – Programador, en vista que es quién aplicará el marco de trabajo de Scrum para implementar soluciones informáticas que permitan añadir nuevas funcionalidades al sistema financiero Econx acoplándolo a la realidad actual y satisfaciendo las necesidades de la Cooperativa optimizando así sus procesos. Otros beneficiarios directos son los colaboradores de las distintas áreas que conforman la institución, pues cada uno de ellos puede remitir los requisitos que necesite sean implementados en el sistema Econx, obteniendo un aplicativo funcional que ofrezca las facilidades para realizar sus actividades laborales de manera ágil y oportuna brindando un mejor servicio a sus socios y clientes. La Cooperativa actualmente cuenta con 28 colaboradores distribuidos en las distintas áreas, entre los cuales 18 son mujeres y 10 hombres.

De forma indirecta, el desarrollo del proyecto de investigación beneficiará a los socios y clientes que hagan uso de los servicios y productos financieros que ofrece la Cooperativa, quienes lo notarán en la agilidad y el corto tiempo que toman en ser atendidos. Así también de forma indirecta serán beneficiadas las 84 Cooperativas que utilizan el sistema financiero Econx a nivel nacional, las cuales pueden adaptar esta alternativa metodológica para desarrollar sus propios productos de software que satisfagan las necesidades de sus Instituciones, en vista que desde la propia iniciativa de la empresa Avmei (proveedores del sistema financiero Econx y Conexus) se propone iniciar una capacitación en cuanto a la programación y estructura del Sistema Financiero Econx, con la finalidad que cada institución tenga la facilidad de crear nuevas funcionalidades o mejorarlo acorde a su realidad, e indistintamente el proyecto puede aplicarlo cualquier otra empresa que disponga de un solo programador en su equipo de desarrollo y requiera implementar software de calidad bajo una metodología ágil.

5. EL PROBLEMA DE INVESTIGACIÓN

La Cooperativa de Ahorro y Crédito ANDINA Ltda., integró a su nómina de colaboradores un Analista – Programador para conformar con el Jefe del Departamento de Sistemas un equipo que provea del soporte técnico requerido por las distintas áreas de la institución. A más del soporte técnico y otras actividades que cumple el departamento de sistemas, el propósito principal por el cual se contrató al analista - programador fue con el objeto de que después de la entrega formal del sistema financiero por parte de los proveedores se empiece a desarrollar soluciones de software que permitan personalizar el sistema acorde a las necesidades de la Cooperativa. Los primeros desarrollos que se generaron a partir de la entrega parcial del sistema financiero por parte del programador se lo implemento de manera tradicional es decir “programar y probar” (code and fix), sin tomar en cuenta las etapas mínimas del proceso que son: análisis, diseño, implementación, pruebas, despliegue y mantenimiento, que garantizan de cierta manera, que un producto de software cumpla con los requisitos planteados por el usuario y sea funcional. El problema se origina en que el programador por motivos de tiempo, recursos y otras funciones asignadas no está en condiciones de abarcar el proceso completo que conlleva el desarrollo de software bajo una metodología, causando así que las soluciones informáticas que desarrolla tarden demasiado tiempo en ser concluidos dado que no se elabora un cronograma de actividades, no se definen todos los requisitos debido a que no se mantiene comunicación directa con los usuarios, se

realizan modificaciones de último momento en los requerimientos, no se ejecuta un plan de pruebas que valide los posibles errores de funcionalidad, no se genera ningún tipo de artefacto (especificación de requisitos, historias de usuarios, diagramas de clases, entre otros), y no cuenta con un control para la documentación del proceso. En definitiva el software que se desarrolla, no cumple con los requerimientos especificados por el usuario considerándolo un software de baja calidad y poco funcional.

La adaptación de la metodología ágil de desarrollo de software Scrum en equipos de desarrollo de un solo programador pretende generar un modelo esquemático para el desarrollo de software, por medio de su aplicación se mejorará los procesos que conlleva la implementación de soluciones informáticas, las mismas que bajo un proceso sistemático bien definido permitan subir a producción aplicativos de calidad que cumplan con los requerimientos planteados por los usuarios, reduzca tiempos, recursos, genere documentación técnica, permita validar la funcionalidad en un ámbito de producción real y sobre todo se convierta en una herramienta de ayuda al programador para futuros desarrollos.

Como caso práctico se adaptará la metodología Scrum al desarrollo del aplicativo que lleva el nombre de “Control de Denominaciones de Dinero”, para el área de cajas de la Cooperativa de Ahorro y Crédito Andina Ltda., el mismo que se lo desarrollará en un tiempo estimado de 2 meses, sin antes mencionar que la adaptación del marco de trabajo de Scrum se la aplicará a todo aplicativo que requieran los distintos departamentos que conforman la institución financiera.

6. OBJETIVOS

6.1. Objetivo General

Adaptar el marco de trabajo de Scrum a equipos de desarrollo con un solo programador, mediante la aplicación de las mejores prácticas de su modelo de procesos, para la implementación de soluciones informáticas en el Departamento de Sistemas de la Cooperativa de Ahorro y Crédito ANDINA Ltda.

6.2. Objetivos Específicos

- Analizar la información recopilada de distintas fuentes de consulta, que fundamenten la teoría necesaria para el desarrollo del proyecto.

- Conocer la situación actual del lugar dónde se lleva a cabo la investigación, para ampliar la visión en cuanto a la problemática presentada.
- Definir el proceso de Scrum para adaptarlo al desarrollo de software en equipos de un solo programador.
- Aplicar el marco de trabajo de Scrum en el desarrollo del aplicativo: “Control de Denominaciones de Dinero” para el área de cajas de la Cooperativa Andina Ltda.

7. ACTIVIDADES Y SISTEMA DE TAREAS EN RELACIÓN A LOS OBJETIVOS PLANTEADOS

Tabla 1: Sistema de Tareas en relación a los objetivos planteados

OBJETIVOS	ACTIVIDADES	RESULTADOS	MEDIOS DE VERIFICACIÓN
Analizar la información recopilada de distintas fuentes de consulta, que fundamenten la teoría necesaria para el desarrollo del proyecto	<ul style="list-style-type: none"> • Consultar diferentes fuentes bibliográficas (libros, sitios web) acorde al tema del proyecto. • Analizar y seleccionar la información que es relevante para la fundamentación teórica. • Organizar y plasmar la información en el marco teórico. 	<ul style="list-style-type: none"> • Obtener información verídica y confiable, que sustente la investigación. 	<ul style="list-style-type: none"> • Lectura comprensiva. • Análisis, Síntesis e Interpretación.
Conocer la situación actual del lugar dónde se lleva a cabo la investigación, para ampliar la visión en cuanto a la problemática presentada.	<ul style="list-style-type: none"> • Observar la problemática presenta en cuanto al desarrollo de software. • Identificar los principales problemas que se presentan por no contar con la aplicación de una 	<ul style="list-style-type: none"> • Plantear una alternativa factible para contrarrestar los problemas evidenciados durante el desarrollo de software. 	<ul style="list-style-type: none"> • Utilizar la técnica de observación y la ficha de observación para recopilar información. • Plantear las preguntas para la entrevista.

	<p>metodología.</p> <ul style="list-style-type: none"> • Aplicar una entrevista al Jefe del Departamento de Sistemas para así recabar información verídica que ayude a comprender la problemática. 	<ul style="list-style-type: none"> • Análisis e Interpretación de resultados de los instrumentos aplicados. 	
<p>Definir los roles, eventos y artefactos de Scrum a ser adaptados en el desarrollo de software en equipos de un solo programador.</p>	<ul style="list-style-type: none"> • Analizar el conjunto de buenas prácticas de Scrum. • Identificar las características de los roles, eventos y artefactos de Scrum. • Generar formatos de los artefactos, acorde a la imagen de la Cooperativa. 	<ul style="list-style-type: none"> • Aplicar las reglas, valores y principios de agilidad que plantea Scrum. • Plantear una propuesta factible para adaptar el marco de trabajo de Scrum en el equipo de un solo programador. • Socializar el marco de trabajo de Scrum e implementar los recursos necesarios que se aplicarán en el desarrollo de software. 	<ul style="list-style-type: none"> • Revisión teórica de la metodología Scrum. • Listar cada regla (roles, artefactos, eventos) de Scrum y generalizarlos acorde a las necesidades del equipo de trabajo. • Estructurar los formatos.
<p>Aplicar el marco de trabajo de Scrum en el desarrollo del aplicativo: “Control de Denominaciones de Dinero” para el área de cajas de la Cooperativa</p>	<ul style="list-style-type: none"> • Recopilar los requerimientos solicitados por el usuario. • Ejecutar los procesos de análisis, diseño, implementación y pruebas de forma iterativa. • Documentar el 	<ul style="list-style-type: none"> • Priorizar y Estimar los requerimientos en el formato correspondiente. (Product Backlog) • Aplicar la 	<ul style="list-style-type: none"> • Aplicar la propuesta del marco de trabajo de Scrum. • Redactar en los artefactos el proceso de desarrollo de software con

Andina Ltda.	proceso de desarrollo.	adaptación del marco de trabajo de Scrum. <ul style="list-style-type: none"> • Obtener la documentación correspondiente del software implementado. 	Scrum.
--------------	------------------------	--	--------

Elaborado por: Fernando Arias

8. FUNDAMENTACIÓN CIENTÍFICO TÉCNICA

En esta sección se describe el marco teórico que hace referencia a los aspectos más relevantes del tema planteado en el proyecto, previo un caso práctico en el cual se demostrará su aplicación.

8.1. Antecedentes

El origen de las metodologías ágiles de desarrollo de software (XP, Scrum, Cristal Methods, etc.), data de años atrás y su evolución ha pasado por un sinnúmero de inconvenientes hasta llegar a lo que actualmente la industria del software aplica para el desarrollo de software.

Remontarse en la historia a los orígenes del desarrollo de software a finales de los años 50 y mediados de los 60, permite recordar la lenta evolución que tuvo la industria del software a nivel mundial. La industria del software afrontó diversos problemas debido a que fue un proceso relativamente nuevo para el cual no se contaba con el personal calificado; programar fue una tarea sumamente complicada para los programadores de aquella época pues no contaban con una formación profesional adecuada sino que aprendían experimentando con los lenguajes de bajo nivel (lenguaje máquina y ensamblador). En general no se establecía ningún tipo de técnica, método ni planificación para el desarrollo de software, más bien a este se lo consideraba como un tipo de arte que debía ser creativo más no tratado como un proceso disciplinario. Estos problemas impedían que el software pueda ser entregado en el tiempo establecido, que los costos sean más elevados que el presupuestado y sobre todo que no satisfaga a plenitud los requerimientos solicitados por el cliente. Todo esto desencadenó la denominada “Crisis del Software”.

Como consecuencia de esta crisis (Weitzenfeld, 2005, pág. 4), describe brevemente una recopilación de casos donde errores en el software han causado pérdidas económicas multimillonarias, e incluso vidas humanas.

- **Accidente de un F-18 (1986).** En abril de 1986 un avión de combate F-18 se estrelló por culpa de un giro descontrolado (unrecoverable spin), atribuido a una expresión “if-then”, para lo cual no había una instrucción “else”, por considerarse innecesario, lo que originó una excepción fuera de control del programa.
- **Muertes por el Therac-25 (1985-1987).** El acelerador lineal medico Therac-25, producido por Atomic Energy of Canada Limited (AECL), fue diseñado para tratamientos de radiación (...). Entre 1985 y 1987 este sistema ocasionó la muerte de varios pacientes en diferentes hospitales de EEUU y Canadá debido a radiaciones de alto poder aplicadas sin control.

En vista de los acontecimientos que acarreaaba la crisis de software se debía tomar medidas correctivas para solventarlo de manera urgente, es así que en octubre de 1968 en Garmisch Alemania se realiza la primera conferencia de la OTAN (Organización del Tratado del Atlántico Norte) acerca del desarrollo de software, teniendo como sus mayores exponentes a Friedrich Bauer y Edsger Dijkstra, en la cual se estableció la necesidad de implementar buenas prácticas que permita garantizar la calidad del software desarrollado en un ámbito de estabilidad y funcionalidad, bajo esos argumentos se propone formalmente el término de Ingeniería de Software como solución a la crisis de la industria del software.

Destacados investigadores y organizaciones han contribuido con distintas ideas y conceptos para definir el término Ingeniería de Software, sin embargo se ha analizado y considerado las siguientes definiciones que se describen a continuación:

“La Ingeniería de Software es una disciplina de ingeniería que se interesa por todos los aspectos de la producción de software, desde las primeras etapas de la especificación del sistema hasta el mantenimiento del sistema después de que se pone en operación.” (Sommerville, 2011, pág. 8).

“Es la disciplina ingenieril que proporciona y aplica los métodos y herramientas necesarios para construir software de calidad, ajustado al presupuesto, en un plazo determinado y en un contexto de constante cambio de requisitos”. (Serna, 2014, pág. 20).

De las definiciones antes señaladas se puede obtener un compendio personal de lo que es la Ingeniería de Software, definiéndola como un conjunto de técnicas formales enmarcadas en un proceso sistemático y debidamente documentado para el desarrollo de software, que cumpla a cabalidad los requerimientos establecidos por el cliente dentro del presupuesto y tiempo planificado, garantizando de esta manera la calidad del producto de software.

(Weitzenfeld, 2005), argumenta que en cuanto al proceso de desarrollo que plantea la ingeniería de software es necesario considerar su ciclo de vida, correspondiente a las diversas etapas por las cuales debe pasar un sistema, comenzando con la formulación de un problema, seguido por la especificación de requisitos, análisis, diseño, implementación o codificación, integración y pruebas del software, Después viene una fase operacional durante la cual se mantiene y extiende el sistema. (pág. 18).

Las actividades descritas por el autor mencionado forman en su conjunto el denominado “ciclo de vida del software”, las mismas que se involucran de inicio a fin en el proceso de desarrollo. Estas a su vez no son rígidas y pueden ser aplicadas en los distintos modelos de ciclo de vida que establece la ingeniería de software; la aplicación de un modelo se lo realiza acorde a la naturaleza del proyecto.

Para definir que es un modelo de ciclo de vida se hace mención al (Instituto Nacional de Tecnologías de la Comunicación, 2009), quien menciona que un modelo de ciclo de vida de software es una vista de las actividades que ocurren durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociados entre ellas. En cada una de las etapas se pueden establecer una serie de objetivos, tareas, actividades y documentación que lo caracterizan. (pág. 25). A continuación se describe los siguientes modelos:

Modelo de Ciclo de Vida en Cascada: En 1970, el investigador Winston Royce planteó lo que se puede considerar como el primer ciclo de vida del software. Dicho ciclo sugería un desarrollo secuencial y lineal de actividades. Este modelo debido su rigidez se lo suele implementar en proyectos cuyos requisitos están bien definidos y no varían durante el proceso de desarrollo. Dado su sencilla gestión es fácil de emplear, generando además entregables específicos en cada fase. Sin embargo presenta inconvenientes en proyectos complejos, ya que los requisitos no son estables y varían acorde a las necesidades que surjan por el cliente. Además su analogía de cascada restringe regresar a una etapa anterior, generando un retraso en el desarrollo. (Serna, 2014, pág. 27).

Modelo de Ciclo de Vida Incremental: Harlan Mills en 1980 propuso el enfoque incremental de desarrollo. Este modelo es de naturaleza interactiva brindando al final de cada incremento la entrega de un producto operacional. Cada incremento tiene su propio ciclo de vida y se basa en el anterior; una vez entregado un incremento no se realizan cambios sino únicamente corrección de errores; el proceso se repite hasta que se elabora el producto completo. El cliente se mantiene en constante contacto con los resultados obtenidos en cada incremento, permite además que los requerimientos varíen acorde a las necesidades que surjan sin afectar el tiempo de entrega del software. (Ramirez, 2012).

Los modelos antes mencionados representan los inicios de un proceso sistemático para el desarrollo de software dando paso a una etapa en la que su flujo de proceso fue aplicado indistintamente en las metodologías que surgieron en aquellos años.

Según la (Universidad Politécnica de Valencia, 2012), menciona que un proceso de software detallado y completo se denomina “Metodología”, la cual se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental, etc.). Adicionalmente define con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas. (pág. 13).

Menciona además que las metodologías con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales. Dentro de este grupo se hace referencia a R.U.P (Proceso Unificado Racional) Dada sus características (dirigida por casos de uso, centrada en la arquitectura, iterativo e incremental) y sus fases (inicio, elaboración, construcción, transición y producción) constituyen la metodología estándar más utilizada para el análisis, implementación y documentación de software. (Limachi, 2012, pág. 3).

Las metodologías tradicionales establecen una planificación bien definida y ordenada considerando importante la documentación para gestionar el proceso de desarrollo de software. No obstante la desventaja radica en que se requiere de una rigurosa disciplina, se genera extensa documentación y se invierte demasiado tiempo en el modelado del software, convirtiéndola en una metodología pesada.

Ante las dificultades del enfoque rígido y robusto que proponían las metodologías tradicionales, la industria del software demandaba un cambio definitivo que permita implementar software en el tiempo, costo y calidad requerida por el usuario, es así que a

inicios de los años 90 surgieron diversas alternativas de desarrollo denominadas “Metodologías Livianas” tales como Extreme Programming (XP), Crystal Methods (CM), Scrum, etc.

En febrero del 2001, tras una reunión celebrada en Utah (EEUU), nace el término “ágil” aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. (Alende, 2010).

Menciona además que tras esta reunión se creó “The Agile Alliance”, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía ágil.

“El manifiesto ágil es un documento que resume en cuatro valores y doce principios las mejores prácticas para el desarrollo de software, basados en la experiencia de 17 industriales del software, en procura de desarrollos más rápidos conservando su calidad”. (Herrera & Valencia, 2007).

La esencia del manifiesto ágil radica en sus cuatro valores:

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

Dentro del enfoque que propone el manifiesto ágil para el desarrollo de software, se puede mencionar las siguientes metodologías que aplican sus valores y principios:

Extreme Programming.- Propuesta por Kent Beck, en 1999, centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software; se preocupa por el aprendizaje de los desarrolladores; comunicación fluida entre todos los participantes; simplicidad en las soluciones implementadas y coraje para enfrentar los cambios es adecuada para proyectos donde existe un alto riesgo técnico. (Bautista, 2009).

Scrum.- Tiene su origen en un estudio de 1986 sobre los nuevos procesos de desarrollo utilizados en productos exitosos en Japón y los Estados Unidos. En 1993 Jeff Sutherland,

John Scumniotales y Jeff McKenna concibieron, ejecutaron y documentaron el primer Scrum para desarrollo ágil de software, en 1995 Ken Schwaber formalizó el proceso para la industria de desarrollo de software. (Alende, 2010, pág. 65).

Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar. (Schwaber & Sutherland, 2014, pág. 4).

Como se puede apreciar la historia nos revela un sinnúmero de sucesos desde la crisis de la industria del software, el origen de la ingeniería del software, los modelos clásicos de ciclo de vida del software, las metodologías tradicionales hasta llegar a las metodologías ágiles, las mismas que actualmente son usadas en empresas a nivel mundial que producen soluciones de software enmarcados en un contexto de calidad.

Actualmente, el creciente mercado de software en América Latina y el Caribe constituye un aspecto vital fundamentalmente por el fuerte impacto del software en el crecimiento económico y social de los países y en la calidad de vida de los ciudadanos. Sin embargo, mucho de los retos que enfrenta la industria de la región están relacionados por una parte con la necesidad de incorporar y masificar la adopción de procesos para mejoramiento de la calidad del software y su incorporación como criterio clave en procesos de adquisición del mismo, y por otra, con la capacitación y certificación del talento humano de esta industria. (Najera, 2013).

En Latinoamérica la industria del software va desarrollándose a pasos agigantados, es evidente este avance debido a que países de la región como Brasil, Colombia, Uruguay ocupan los primeros lugares en la lista de exportadores de software de calidad a nivel mundial. Para llegar a estos sitios (Martínez, Ramón, & Bertone, 2012) afirman que gran parte de las empresas dedicadas a ésta área son PYMES las cuales para desarrollar sus productos aplican metodologías ágiles que ofrecen la posibilidad de contar con procesos livianos y simples, los cuales pueden adaptarse a su estructura. Según los resultados de una encuesta realizada por State of Agile Survey en 2015, con el 52% Scrum es la metodología más usada y practicada aunque no en forma completa sino adaptando las técnicas que consideran más apropiadas a las necesidades del proyecto a desarrollar.

La aplicación de metodologías ágiles en la industria de software en Ecuador, según (Computerworld, 2016) se presenta en la década de los noventa. Su uso y utilidad se difunde rápidamente dentro de la comunidad de desarrolladores.

Agrega además que en el 2011 se crea formalmente la Comunidad Ágil Ecuador, su propósito es fomentar el uso de métodos de desarrollo ágil. Según cifras de Comunidad Ágil Ecuador, se estima que el 36% de los proyectos de desarrollo de software utilizan métodos ágiles. Para Johnny Ordoñez, miembro de la Comunidad Ágil Ecuador, va más allá de un método de desarrollo de software, es la adopción de una mentalidad que puede aplicarse dentro de otras disciplinas relacionadas al ámbito de la creatividad, comunicación o colaboración. Scrum es un modelo de referencia que precisa un conjunto de destrezas y roles para el desarrollo de proyectos enfocados al agilísimo; este marco puede ser aplicado a diferentes áreas, y adaptarse a las necesidades del usuario.

En otro artículo del sitio web referenciado menciona que las metodologías ágiles o “ligeras” constituyen un nuevo enfoque en el desarrollo de software, están orientadas a todos los equipos de desarrollo y organizaciones, independientemente de su dimensión, y se caracterizan por su flexibilidad ante los cambios y su ideología de colaboración.

Según argumenta Daniel Márquez, Consultor de Atmira (Consultora IT) los beneficios que ofrece la aplicación de estas metodologías ligeras, se ven claramente reflejados en la optimización de los resultados de las organizaciones, ya que por un lado facilitan la optimización de procesos, por otro persiguen la excelencia y mantienen la flexibilidad ante los cambios, y además permiten el enfoque a cliente orientando el producto y el desarrollo a sus necesidades a través de la incorporación de la mejora continua basada en la experiencia del usuario.

La información recabada de la situación actual de la aplicación de metodologías ágiles en la industria del software en Latinoamérica y en especial en Ecuador nos da a conocer la realidad de las empresas de desarrollo de software y su enfoque, permitiendo tener una idea clara para el desarrollo del proyecto. Ya que en la información recopilada mencionan que el éxito de los proyectos que se desarrollan está gestionado bajo metodologías ágiles que favorecen a que las soluciones informáticas estén enmarcadas en los costos y planificación establecida así como también cumplir a cabalidad los requerimientos de los usuarios.

8.2. Referentes Teóricos

Según (Schwaber & Sutherland, 2014) afirman que el tamaño óptimo del Equipo de Desarrollo es lo suficientemente pequeño como para permanecer ágil y lo suficientemente grande como para completar una cantidad de trabajo significativa. Tener menos de tres miembros en el Equipo de Desarrollo reduce la interacción y resulta en ganancias de productividad más pequeñas. Los Equipos de Desarrollo más pequeños podrían encontrar limitaciones en cuanto a las habilidades necesarias durante un Sprint, haciendo que el Equipo de Desarrollo no pudiese entregar un Incremento que potencialmente se pueda poner en producción. (pág. 6).

Sin embargo investigaciones previas que podrían guardar relación con la aplicación del marco de trabajo de Scrum en equipos con un solo programador, han logrado obtener muy buenos resultados en proyectos con sprints cortos (1 a 2 semanas), asegurando la calidad del producto final. Razón por la cual se los considera como pauta para desarrollar el proyecto aplicado al equipo de desarrollo del departamento de sistemas de la Cooperativa Andina. Las investigaciones referenciadas se las describe a continuación:

(Nuñez & Sánchez, 2011), en su artículo “Marco de procesos de desarrollo de software orientado a entornos productivos pequeños especializados en soluciones de gestión” describen la elaboración de un marco de procesos de desarrollo de software orientado a entornos productivos pequeños. Debido a que la metodología que se utilizó para el desarrollo de software, dígase SCRUM, no abarca todos los elementos de la gestión de proyectos, se llevó a cabo un estudio para demostrar los puntos de encuentro entre la guía del PMBOK y SCRUM, donde se evidenció cómo pueden ser agrupadas las diferentes áreas de conocimiento de la guía con las fases de SCRUM, destacando los artefactos generados más importantes y sin perder el sentido por el cual existen las metodologías ágiles, refiriéndose así al principio del manifiesto ágil: “La respuesta al cambio, por encima del seguimiento de un plan”. Como conclusión del trabajo argumentan que se obtuvo el modelo de marco de procesos de desarrollo de software, se describieron las actividades fundamentales propuestas que componen el marco, de igual manera se definieron cada uno de sus roles, especificando las competencias que deben reunir para poder acometer cada tarea, conjuntamente se detallaron los artefactos de entrada y salida y las herramientas y técnicas que resultan necesarios para la aplicación de este modelo.

(Vargas, 2015), en su artículo “Metodología de Desarrollo de Software Dirigida a Equipos de Trabajo Reducidos para su Aplicación en los Proyectos Integradores y Tesis en UNIANDÉS Extensión Ibarra”, propone crear una nueva metodología ágil que permite adaptarse de mejor manera a equipos de trabajo reducidos para el desarrollo de software, aplicando el marco de trabajo de Scrum como metodología de desarrollo ágil, RUP y XP para la gestión de proyectos. Como conclusión menciona que MODER (Metodología Orientada a Desarrollo con Equipos Reducidos) tuvo buena valoración por parte de profesionales externos a la institución, en lo referente a su inclinación y cumplimiento de los principios del manifiesto ágil para desarrollo de software. Finalmente recomienda utilizar la metodología MODER solo en proyectos de corta duración, baja y media complejidad, de lo contrario utilizar otra propuesta metodología acorde a las necesidades.

(Armijos, Fiallos, Villavicencio, & Abad, 2015), en la revista tecnológica de la ESPOL exponen un artículo con el tema “Aplicación de Scrum en la construcción de un simulador de Redis”. Expone la experiencia de desarrollar un simulador usando el modelo ágil Scrum. El objetivo del simulador es obtener datos relacionados al rendimiento de los procesos de balanceo de carga y al reemplazamiento de la caché en un sistema de caché distribuida. Ante la problemática suscitada por las limitaciones del proyecto (2 desarrolladores, 7 semanas de tiempo), surgió la necesidad de adoptar un marco de trabajo que permita desarrollar el simulador en el menor tiempo posible, de forma organizada y con esfuerzo reducido. Por lo tanto se adaptó el marco de trabajo de Scrum de la siguiente manera:

- **Tamaño del equipo:** 3 personas.
- **Dueño del producto:** Profesor del curso (investigador en el área de sistemas distribuidos).
- **Facilitador:** Estudiante de maestría con experiencia profesional en Scrum. Adicionalmente, se desempeñó como desarrollador.
- **Desarrollador:** Estudiante de maestría con experiencia profesional usando metodologías tradicionales.
- **Reuniones diarias:** 2 veces a la semana.
- **Reuniones de revisión:** Sólo al final, cuando el simulador estuvo terminado.
- **Reuniones de retrospectiva:** participaban los 2 desarrolladores y era realizada al final de cada sprint.

Hacen mención que la adaptación de Scrum que apliquen los equipos de desarrollo, deben obedecer a las características del proyecto como, por ejemplo: el tamaño del software, los recursos disponibles, el área de investigación, la localización del equipo de trabajo, los modos de comunicación, entre otros factores.

Concluyen mencionando que, el uso del modelo Scrum adaptado a las necesidades del proyecto, permitió desarrollar un producto con flexibilidad y sin mayor afectación a los plazos de entrega. Recomiendan hacer un análisis del producto, de los recursos disponibles y del entorno para hacer adaptaciones de Scrum. Además agregan que es necesario continuar con la realización de más casos de estudios de la aplicación de Scrum acotando que, solo la experiencia hace que los marcos de trabajo pongan a prueba su madurez y flexibilidad.

(Toapanta, 2012), En su tesis titulada “Método Ágil Scrum, Aplicado a la Implantación de un Sistema Informático para el Proceso de Recolección Masiva de Información con Tecnología Móvil”, expone que su estudio se enfocó en el análisis del método ágil SCRUM para la implementación de una metodología aplicada al desarrollo de software para dispositivos móviles. Enfatiza en que la ejecución y culminación del proyecto permitió establecer una metodología basada en SCRUM, complementada con otros métodos. El resultado es un producto de software funcional, en cuyo desarrollo se pudo demostrar la validez de SCRUM aplicado a proyectos de software de mediano tamaño, en entornos cambiantes, con grupos de trabajo pequeños que involucran permanentemente al dueño del producto.

El análisis de las investigaciones que guardan relación con el tema del proyecto, dan a conocer el extenso campo de aplicación de la Metodología Scrum en distintos entornos en los que se ha adaptado. En cada una de ellas el marco de trabajo de Scrum ha sido modificado para adaptarlo según las necesidades de cada proyecto, estableciendo roles, eventos y artefactos aplicados en el proceso de desarrollo de software. Resaltan además la importancia de adaptar Scrum ya que es aplicable a cualquier tipo de proyecto, es flexible a cambios de última hora requeridos por el cliente, establece fechas fijas para las reuniones de revisión de los sprints, permite entregar al cliente avances de las funcionalidades del software en las fechas establecidas y en general aporta buenas prácticas que elevan la productividad del equipo de desarrollo.

Finalmente se hace referencia a lo mencionado por los autores de las investigaciones analizadas, en la cual coinciden que adaptar el marco de trabajo de Scrum permite alcanzar el

éxito en proyectos de corta duración; con sprints de 1 a 2 semanas; un equipo conformado por 3 a 5 personas permite una comunicación más fluida, en él se debe incluir al cliente como parte primordial del equipo; las historias de usuario son artefactos que definen la funcionalidad que debe satisfacer el sistema de acuerdo a la prioridad establecida por el cliente, descompone en tareas independientes, negociables, valorables, estimables, pequeñas y comprobables que deben ser cumplidas en un lapso de tiempo fijo; mantener reuniones diarias de manera constante fortalece el vínculo de confianza y comunicación entre el equipo y el cliente con el fin de revisar y validar los sprint generados. Un aspecto muy importante que menciona (Alaimo, 2010) es que los programadores en un equipo Scrum programan, prueban, analizan, diseñan. Hacen todo lo necesario para ayudar al equipo a completar el trabajo comprometido en un sprint, deben estar dispuestos a contribuir en cualquier número de maneras en optimizar el rendimiento del equipo en general. Uno de los cambios más notables para los programadores en un equipo Scrum es que ya no pueden sentarse en sus cubículos y esperar que se les diga exactamente qué debe programar, ellos necesitan convertirse en participantes activos en la comprensión de los requisitos del producto interactuando con los clientes y usuarios. Del mismo modo, los programadores pueden esperar pasar más tiempo interactuando con sus compañeros de trabajo y sentirse en un espacio de equipo, participar en discusiones y ayudar a otros con problemas.

Es importante señalarlo en vista de la realidad al cual está orientado el proyecto, ya que en el equipo de desarrollo del departamento de sistemas de la Cooperativa Andina existe un solo programador.

Es evidente que el proyecto planteado cuenta con los suficientes referentes teóricos que afianzan las bases sobre las cuales se lo desarrollará.

8.3. Elementos teóricos conceptuales

A continuación se puntualiza los conceptos en cuanto a la metodología y herramientas aplicadas en el desarrollo del proyecto de investigación.

8.3.1. Metodología Ágil de desarrollo de software Scrum

Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar. (Schwaber & Sutherland, 2014, pág. 4).

Según (Palacio, 2015), Scrum es un modelo de desarrollo ágil caracterizado por:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto-organizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizarlas una tras otra en un ciclo secuencial o de cascada.

Scrum se puede adoptar aplicando reglas definidas, o adoptando los valores originales Scrum con reglas personalizadas. El marco de trabajo de Scrum está formado por:

- **Roles:**
 - El equipo Scrum (Scrum Team)
 - El dueño del producto (Product Owner)
 - El Scrum master
- **Artefactos:**
 - Pila del producto (Product Backlog)
 - Pila del sprint (Sprint Backlog)
 - Incremento (Product Increment)
- **Eventos**
 - Sprint
 - Reunión de planificación del sprint (Sprint Planning)
 - Scrum diario (Daily Meeting)
 - Revisión del sprint (Sprint Review)
 - Retrospectiva del sprint (Sprint Retrospective)

Artefactos:

Pila del producto: (Product Backlog), es el inventario de funcionalidades que deben incorporarse al producto a través de los sucesivos sprints. La pila está en continuo crecimiento y evolución. Se la elabora con el resultado de una reunión en donde colabora todo el equipo partiendo de la visión del propietario del producto, quien a su vez mantiene la pila ordenada por la prioridad de los elementos que confieren mayor valor al producto. Los elementos prioritarios deben tener mayor nivel de comprensión y detalle.

Pila de sprint: (sprint Backlog), es la lista que descompone las funcionalidades de la pila del producto (historias de usuario) en las tareas necesarias para construir un incremento: una parte completa y operativa del producto. Es también una herramienta para la comunicación visual directa del equipo en vista que es:

- Realizada de forma conjunta por todos los miembros del equipo.
- Cubre todas las tareas identificadas por el equipo para conseguir el objetivo del sprint.
- Sólo el equipo la puede modificar durante el sprint.
- Las tareas demasiado grandes deben descomponerse en otras más pequeñas.
- Es visible para todo el equipo. Idealmente en un tablero o pared en el mismo espacio físico donde trabaja el equipo.

Incremento: El incremento es la parte de producto producida en un sprint, y tiene como característica el estar completamente terminada y operativa, en condiciones de ser entregada al cliente. No se deben considerar como Incremento a prototipos, módulos o sub-módulos, ni partes pendientes de pruebas o integración. Idealmente en Scrum:

- Cada elemento de la pila del producto se refiere a funcionalidades entregables, no a trabajos internos del tipo “diseño de la base de datos”.
- Se produce un “incremento” en cada iteración.

Eventos:

Sprint: Se denomina sprint a cada ciclo o iteración de trabajo que produce una parte del producto terminada y funcionalmente operativa.

Reunión de planificación del sprint: reunión de trabajo previa al inicio de cada sprint en la que se determina cuál va a ser el objetivo del sprint y las tareas necesarias para conseguirlo. La reunión es conducida por el Scrum Master con la asistencia del propietario del producto y el equipo completo, y otros implicados en el proyecto. La reunión puede durar una jornada de trabajo completa, cuando se trata de planificar un sprint largo (de un mes de duración) o un tiempo proporcional para planificar un sprint más breve.

Scrum diario: Reunión diaria breve, de no más de 15 minutos, en la que el equipo sincroniza el trabajo y establece el plan para las 24 horas siguientes. Se recomienda realizarla de pie junto a un tablero con la pila del sprint y el gráfico de avance del sprint, para que todos

puedan compartir la información y anotar. En esta reunión cada miembro del equipo de desarrollo explica:

- Lo que ha logrado desde el anterior Scrum diario.
- Lo que va a hacer hasta el próximo Scrum diario.
- Si están teniendo algún problema, o si prevé que puede encontrar algún impedimento.

Al final de la reunión el equipo refresca el gráfico de avance del sprint, con las estimaciones actualizadas.

Revisión del sprint: Reunión realizada al final del sprint para comprobar el incremento. No debe durar más de 4 horas, en el caso de revisar sprints largos. Para sprints de una o dos semanas, con una o dos horas de duración debería ser suficiente. Es una reunión informativa. Su misión no es la toma de decisiones ni la crítica del incremento. Con la información obtenida, posteriormente el propietario del producto tratará las posibles modificaciones sobre la visión del producto. Al ver y probar el incremento, el propietario del producto, y el equipo en general obtienen feedback relevante para revisar la pila del producto.

Retrospectiva del sprint: Reunión que se realiza tras la revisión de cada sprint, y antes de la reunión de planificación del siguiente, con una duración recomendada de una a tres horas, según la duración del sprint terminado. El objetivo de la revisión del sprint es analizar “QUÉ” se está construyendo, mientras que una reunión retrospectiva se centra en “CÓMO” lo estamos construyendo: “CÓMO” estamos trabajando, con el objetivo de analizar problemas y aspectos mejorables. Las reuniones "retrospectivas" realizadas de forma periódica por el equipo para mejorar la forma de trabajo, se consideran cada vez más un componente del marco técnico de Scrum, si bien no es una reunión para seguimiento de la evolución del producto, sino para mejora del marco de trabajo.

Roles:

Todas las personas que intervienen, o tienen relación directa o indirecta con el proyecto, se clasifican en dos grupos: comprometidos e implicados.

Propietario del producto (Product Owner): es quien toma las decisiones del cliente. Su responsabilidad es el valor del producto. Para simplificar la comunicación y toma de decisiones es necesario que este rol recaiga en una única persona. Además decide en última instancia cómo será el resultado final, y el orden en el que se van construyendo los sucesivos incrementos: qué se pone y qué se quita de la pila del producto, y cuál es la prioridad de las

funcionalidades. Conoce el plan del producto, sus posibilidades y plan de inversión, así como del retorno esperado a la inversión realizada, y se responsabiliza sobre fechas y funcionalidades de las diferentes versiones del mismo. Finalmente es importante mencionar que la organización debe respetar sus decisiones y no modificar prioridades ni elementos de la pila del producto.

Equipo de desarrollo (Scrum Team): Lo forman el grupo de profesionales que realizan el incremento de cada sprint. Se recomienda que un equipo Scrum tenga entre 3 y 8 personas. Más allá de 8 resulta más difícil mantener la comunicación directa. Es un equipo *multifuncional*, en el que todos los miembros trabajan de forma solidaria con responsabilidad compartida. El equipo tiene espíritu de colaboración, y un propósito común: conseguir el mayor valor posible para la visión del cliente. Un equipo Scrum responde en su conjunto. Trabaja de forma cohesionada y auto-organizada. No hay un gestor para delimitar, asignar y coordinar las tareas. Son los propios miembros los que lo realizan. En el equipo:

- Todos conocen y comprenden la visión del propietario del producto.
- Aportan y colaboran con el propietario del producto en el desarrollo de la pila del producto.
- Comparten de forma conjunta el objetivo de cada sprint y la responsabilidad del logro.
- Todos los miembros participan en las decisiones.
- Se respetan las opiniones y aportes de todos.
- Todos conocen el modelo de trabajo con Scrum.

Scrum Master: Es el responsable del cumplimiento de las reglas de un marco de Scrum técnico, asegurando que se entienden en la organización, y se trabaja conforme a ellas. Proporciona la asesoría y formación necesaria al propietario del producto y al equipo. Realiza su trabajo con un modelo de liderazgo servil: al servicio y en ayuda del equipo y del propietario del producto. Proporciona:

- Asesoría y formación al equipo para trabajar de forma auto-organizada y con responsabilidad de equipo.
- Revisión y validación de la pila del producto.
- Moderación de las reuniones.
- Resolución de impedimentos que en el sprint pueden entorpecer la ejecución de las tareas.

- Gestión de las “dinámicas de grupo” en el equipo.
- Configuración, diseño y mejora continua de las prácticas de Scrum en la organización. Respeto de la organización y los implicados, con las pautas de tiempos y formas de Scrum.

Valores:

Scrum define un marco que ayuda a organizar a las personas y el flujo de trabajo. Es la “carrocería” o el interfaz visible, pero el motor de la agilidad son los valores ágiles. La agilidad no la proporciona el cumplimiento de prácticas, sino de valores.

- Foco. Como nos enfocamos en solo unas pocas cosas a la vez, trabajamos bien juntos y producimos un trabajo excelente. Entregamos elementos valiosos antes.
- Coraje. Como no estamos solos, nos sentimos apoyados y tenemos más recursos a nuestra disposición. Esto nos brinda coraje para superar grandes desafíos.
- Transparencia. Mientras trabajamos juntos, nos contamos cómo nos está yendo, y qué cosas se nos interponen en el camino. Aprendemos que es bueno expresar las preocupaciones, para que puedan resolverse.
- Compromiso. Como tenemos control sobre nuestro propio destino, nos comprometemos más a tener éxito.
- Respeto. Mientras trabajamos juntos, compartimos éxitos y fracasos, nos respetamos mutuamente, y nos ayudamos a ser merecedores del respeto.

El marco de trabajo que provee Scrum se adapta a la intención del presente proyecto, ya que su flexibilidad conjuntamente con sus artefactos, roles y eventos permite adaptarla a equipos pequeños dotándolos de una alta productividad que generará buenas prácticas conjuntamente con sus valores y los principios de agilidad para el desarrollo de software eficiente.

8.3.2. Sistema Financiero Econx

Es un Sistema Financiero para Cooperativas de Ahorro y Crédito en un ambiente WEB. Está desarrollado en base a los reglamentos establecidos por los organismos de control como la Superintendencia de Economía Popular y Solidaria, la Unidad de Análisis Financiero, etc. Es completo y brinda todas las facilidades de conectividad y los procesos requeridos por las cooperativas.

Econx está codificado bajo lenguajes de desarrollo de última generación, recalando que estos son de código abierto y no tienen costos de licenciamientos. El manejo de la información se lo realiza a través de una base de datos relacional como lo es IBM-INFORMIX versión 11.50. (Avmei Ltda., 2013).

Econx es el sistema sobre el cual Cooperativa Andina ejecuta sus procesos financieros y almacena información. En él se embeberá las nuevas funcionalidades implementadas bajo la adaptación del marco de trabajo de Scrum, mejorándolo acorde a las necesidades de la institución. Cabe recalcar que el desarrollo y pruebas de las aplicaciones se las realizará sobre Econx1 que es el ambiente de pruebas.

8.3.3. Arquitectura Cliente – Servidor

La arquitectura Cliente/Servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor, al proceso que responde a las solicitudes. Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones, la cual proporciona los mecanismos básicos de direccionamiento y transporte. (Elismen, 2014).

La arquitectura cliente – servidor, se la utiliza para acceder a los sistemas en Cooperativa Andina. Es importante mencionar que las nuevas funcionalidades desarrolladas deben responder óptimamente a la transaccionalidad fluida de datos entre las estaciones de trabajo y los servidores.

8.3.4. Php – Html

PHP (Personal Home Page/Página Principal Personal), es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web. Generalmente se ejecuta en un servidor web, tomando el código en PHP cómo su entrada y creando páginas web cómo salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. (Rojas, 2013).

HTML (HiperText Markup Language/Lenguaje de Marcas de Hipertexto) se utiliza comúnmente para establecer la estructura y contenido de un sitio web, tanto de texto, objetos e imágenes. (Definición ABC, s.f).

Es importante describir ambos lenguajes de programación orientados a la web, ya que el sistema financiero Econx está codificado en su totalidad bajo PHP y HTML, tomando en cuenta que el HTML esta embebido dentro del código PHP. Su aprendizaje y aplicación en el software a desarrollar debe contemplar esta característica para mantener su estructura.

8.3.5. JavaScript - Ajax

JavaScript es un lenguaje de scripting multiplataforma y orientado a objetos. Es un lenguaje pequeño y liviano. Dentro de un ambiente de host, JavaScript puede conectarse a los objetos de su ambiente y proporcionar control programático sobre ellos. (Neher, 2015).

AJAX (Asynchronous JavaScript And XML/JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones. (de la Cuesta, 2014).

Las funciones que ejecutan procesos en Econx están implementadas con JavaScript, que conjuntamente con Ajax son las que permiten entre otros enviar y recibir datos mediante cadenas concatenadas hacia y desde la base de datos. El programador debe comprender y aplicar la codificación para mantener el estándar establecido en el sistema.

8.3.6. Informix

Es un gestor de base de datos, incluye un RDBMS (Sistema Administrador de Base de Datos Relacionales), es un lenguaje de cuarta generación e incluye juegos de herramientas para la inclusión de SQL en programas de aplicación. (Royal, 2016).

La versión 11.50 de Informix es la que se encuentra en producción en la Cooperativa. Cabe recalcar que para la creación de nuevas tablas o la ejecución de consultas SQL durante el desarrollo de software, se usará una base de datos de prueba.

8.3.7. Putty

Putty es un cliente Telnet y SSH gratuito para plataformas Windows y Linux que le permite conectarse a un servidor remoto desde un pc conectado a Internet. Putty es altamente configurable e incluye muchas opciones de ajuste de las conexiones, sesiones y características

SSH de seguridad. Ello se debe a que el programa está particularmente dirigido a programadores y administradores de red permitiendo ejecutar comandos y procedimientos de forma remota utilizando el protocolo TCP/IP. (Tatham, 2016).

Esta herramienta se la utiliza para acceder remotamente a los servidores de aplicación y en especial al de base de datos para ejecutar consultas SQL.

8.3.8. Notepad++

Notepad++ es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. Solo para la plataforma en Microsoft Windows. Incluye opciones útiles para usuarios avanzados como desarrolladores y programadores. Se distribuye bajo los términos de la Licencia Pública General de GNU. (López, 2012).

Notepad++ es el editor de texto que provee de un ambiente óptimo para la codificación, lo que resalta de este editor es que es gratuito, minimiza el tamaño del código, provee de distintos lenguajes de programación entre ellos PHP, JavaScript., además permite su personalización para adaptarse a las necesidades del programador.

8.3.9. Beyond Compare

Es una utilidad práctica y fácil de usar que proporciona herramientas y características para comparar y sincronizar carpetas y archivos. Este programa realiza una comparación de directorios y archivos entre equipos remotos, pudiendo detectar las posibles diferencias que existan entre ambos equipos. Así se logra tener los mismos ficheros y directorios con facilidad. (Scooter Software, 2013).

Este aplicativo se lo usa para comparar los archivos y carpetas entre el sistema a prueba y el de producción. Facilitando al programador subir los cambios o nuevos archivos al servidor de aplicación.

9. PREGUNTAS CIENTÍFICAS O HIPÓTESIS

9.1. Hipótesis

Adaptar el marco de trabajo de Scrum para equipos de desarrollo con un solo programador, permitirá implementar las mejores prácticas de esta metodología para la producción de

software en el Departamento de Sistemas de la Cooperativa de Ahorro y Crédito ANDINA Ltda.

9.1.1. Variable independiente

Adaptar el marco de trabajo de Scrum para equipos de desarrollo con un solo programador.

9.1.2. Variable dependiente

Implementar las mejores prácticas de la metodología Scrum para la producción de software en el Departamento de Sistemas de la Cooperativa de Ahorro y Crédito ANDINA Ltda.

10. METODOLOGÍAS Y DISEÑO EXPERIMENTAL

10.1. Tipos de Investigación

10.1.1. Investigación Documental

Consiste en un análisis de la información escrita sobre un determinado tema, con el propósito de establecer relaciones, diferencias, etapas, posturas o estado actual del conocimiento respecto al tema objeto de estudio. (Velasquéz, 2008).

El proyecto reúne aspectos de la investigación documental, ya que se recopiló información de distintas fuentes de consulta, se analizó detalladamente y se la enfocó para estructurar una base teórica compacta con el contenido más relevante de trabajos realizados por distintos investigadores que guardan relación con el tema planteado.

10.1.2. Investigación Aplicada

Busca o perfecciona recursos de aplicación del conocimiento ya obtenido mediante la investigación pura, y por tanto no busca la verdad sino la utilidad. Se trata de investigar las maneras en que el saber científico producido por la investigación pura puede implementarse o aplicarse en la realidad para obtener un resultado práctico. (Velasquéz, 2008)

Mediante el análisis de la base teórica recopilada se obtuvo información esencial que generó un conocimiento amplio y fluido en cuanto al tema planteado, este conocimiento será aplicado a un caso práctico que devuelva un resultado que permita evidenciar la factibilidad del proyecto.

10.1.3. Investigación de Campo

Se trata de la investigación aplicada para comprender y resolver alguna situación, necesidad o problema en un contexto determinado. El investigador trabaja en el ambiente natural en que conviven las personas y las fuentes consultadas, de las que obtendrán los datos más relevantes a ser analizados. (Velasquéz, 2008).

Para tener una visión más amplia de la situación, se trabajó sobre el ambiente en donde se pudo apreciar directamente las causas que originan la problemática en cuanto al desarrollo de software.

10.2. Métodos de Investigación

10.2.1. Método Analítico

El método analítico, provee un proceso cognoscitivo que permite al investigador descomponer al objeto y estudiar sus partes de forma individual. (Campillo, 2016)

10.2.2. Método de la Abstracción

El método de la abstracción, consiste en captar las características esenciales y aislar lo que no es relevante de un objeto para su respectivo estudio y posterior aplicación. (Campillo, 2016)

El objeto de estudio del proyecto es el marco de trabajo de Scrum, para lo cual se toma los aspectos (roles, eventos y artefactos) por separado, los analiza a cada uno y abstrae lo más relevante de ellos, para ser adaptados a equipos de un solo programador, lo cual será evidenciado en un caso práctico.

10.3. Técnicas de Investigación

Las técnicas son importantes en el proceso de la investigación, ya que integran la estructura por medio de la cual se recaba y organiza la información. Proveen además de instrumentos para la recolección de datos. Las técnicas empleadas en el proyecto son:

10.3.1. Técnica de la Observación Directa

Es una técnica que consiste en observar atentamente el fenómeno, hecho o caso, tomar información y registrarla para su posterior análisis. (Doupovec, 2010).

Bajo esta técnica se identificó la problemática que se evidencia en cuanto al proceso de desarrollado de software tradicional que se emplea en el departamento de sistemas de Cooperativa Andina.

10.3.2. Técnica de la Entrevista

La entrevista, desde el punto de vista del método, es una forma específica de interacción social que tiene por objeto recolectar datos para una indagación. (Doupovec, 2010)

La entrevista se la aplica al Jefe del Departamento de Sistemas, con la finalidad de recopilar información del proceso que emplean para el desarrollo de software.

10.4. Instrumentos de Investigación

Los instrumentos empleados para la recolección de información en el proyecto son los siguientes:

10.4.1. Ficha de Observación

Son instrumentos donde se registra la descripción detallada de lugares, personas, hechos o fenómenos que forman parte de la investigación. – Ver Anexo --

10.4.2. Cuestionario de Entrevista

Es un conjunto de preguntas diseñadas para generar los datos necesarios, con el propósito de alcanzar los objetivos del proyecto de investigación. Se trata de un plan formal para recabar información de la unidad de análisis objeto de estudio y centro del problema de investigación. – Ver Anexo --

10.5. Implementación de la propuesta

La Cooperativa de Ahorro y Crédito Andina Ltda., es una institución financiera que desde hace ocho años brinda sus servicios en la provincia de Cotopaxi, fomentando el desarrollo

local mediante la oferta de distintos servicios y productos financieros puestos a consideración de sus distinguidos socios y clientes. Cooperativa Andina cuenta con un equipo de profesionales en el Departamento de Sistemas quienes son responsables de administrar el parque tecnológico, proveer de soporte técnico, asegurar la disponibilidad y confiabilidad de la información; con la finalidad de mantener siempre operativos los procesos que se ejecutan en los sistemas.

En virtud del desarrollo institucional de Cooperativa Andina, se ha visto en la necesidad de sistematizar varios procesos e implementar nuevas funcionalidades en el sistema financiero Econx, mediante el desarrollo de software que permita agilizar la labor diaria de los colaboradores y ponerlo a servicio de socios y clientes. Sin embargo pese a los conocimientos y capacidades del equipo del departamento de sistemas se evidencia que éste no aplica una metodología que asegure la eficacia del software desarrollado.

En base a esta realidad la propuesta del proyecto de investigación se aplicará para brindar un marco de trabajo que genere buenas prácticas para el desarrollo de software bajo una metodología ágil en el equipo de un solo programador. Para demostrar la factibilidad del proyecto se lo aplicará en el desarrollo de un aplicativo para el área de cajas denominado “Control de Denominaciones”. Es importante señalar que el marco de trabajo de Scrum no se aplicará únicamente para el desarrollo del aplicativo antes mencionado, sino que se describirá de forma genérica ya que pasará a formar parte del Manual de Procesos y Procedimientos del Departamento de Sistemas. La propuesta se define de la siguiente manera:

MARCO DE TRABAJO DE SCRUM ADAPTADO A EQUIPOS DE DESARROLLO CON UN PROGRAMADOR

Para trabajar con Scrum y adaptar su marco de trabajo también se fomentará sus valores sobre los cuales esta cimentado. Siendo estos:

- *Foco*: Como nos enfocamos en solo unas pocas cosas a la vez, trabajamos bien juntos y producimos un trabajo excelente. Entregamos elementos valiosos antes.
- *Coraje*. Como no estamos solos, nos sentimos apoyados y tenemos más recursos a nuestra disposición. Esto nos brinda coraje para superar grandes desafíos.

- *Transparencia.* Mientras trabajamos juntos, nos contamos cómo nos está yendo, y qué cosas se nos interponen en el camino. Aprendemos que es bueno expresar las preocupaciones, para que puedan resolverse.
- *Compromiso.* Como tenemos control sobre nuestro propio destino, nos comprometemos más a tener éxito.
- *Respeto.* Mientras trabajamos juntos, compartimos éxitos y fracasos, nos respetamos mutuamente, y nos ayudamos a ser merecedores del respeto.

Definición de Roles

Scrum establece tres roles dentro de su marco de trabajo, los mismos que adaptándolos al proyecto se definirán de la siguiente forma:

Tabla 2: Rol Product Owner

PRODUCT OWNER (DUEÑO DEL PRODUCTO)	
	DESCRIPCIÓN
	<p>El Product Owner es la única persona autorizada para decidir sobre qué funcionalidades y características tendrá el producto. Es quien representa al cliente, usuarios del software y todas aquellas partes interesadas en el producto.</p>
PERFIL	RESPONSABILIDADES
<ul style="list-style-type: none"> • Excelente facilidad de comunicación en las relaciones interpersonales. • Excelente conocimiento del negocio. • Facilidad para análisis de relaciones costo/beneficio. • Visión de negocios 	<ul style="list-style-type: none"> ✓ Describir claramente y mediante historias de usuario las funcionalidades del producto de software. ✓ Asegurar que el Equipo de Desarrollo entiende los elementos de la Lista del Producto al nivel necesario. ✓ Ordenar los elementos en la Lista del Producto para alcanzar los objetivos y misiones de la mejor manera posible. ✓ Asegurar que la Lista del Producto es visible, transparente y clara para todos, y que muestra aquello en lo que el equipo trabajará a continuación. ✓ Participar activamente en los eventos de Scrum.

Elaborado por: Fernando Arias

El rol del Product Owner será asumido por los Jefes Departamentales de Cooperativa Andina, en vista que ellos cumplen con el perfil establecido. Sin embargo el rol puede asumirlo el colaborador que forma parte de aquel departamento que requiera el producto de software, para lo cual se lo capacitará mediante una inducción acerca del rol que debe asumir dentro del marco de trabajo de Scrum.

Tabla 3: Rol Scrum Master

SCRUM MASTER (FACILITADOR)	
	DESCRIPCIÓN
	<p>El Scrum Master es responsable de asegurarse que el trabajo del equipo vaya bien siguiendo las bases de Scrum. Además, se encarga de remover cualquier obstáculo que pueda encontrar el equipo de desarrollo.</p>
PERFIL	RESPONSABILIDADES
<ul style="list-style-type: none"> • Excelentes conocimientos de Scrum • Amplia vocación de servicio • Tendencia altruista • Amplia capacidad para la resolución de problemas • Analítico y observador • Saber incentivar y motivar • Capacidad docente e instructiva • Buen carisma para las negociaciones 	<ul style="list-style-type: none"> ✓ Garantizar la correcta aplicación de Scrum. ✓ Resolver los conflictos que entorpezcan el progreso del proyecto. ✓ Incentivar y motivar al Scrum Team, creando un clima de trabajo colaborativo. ✓ Fomentar la auto-gestión del equipo e impedir la intervención de terceros en la gestión del equipo. ✓ Interactúa con el cliente y el equipo ✓ Coordina los encuentros diarios

Elaborado por: Fernando Arias

El Rol del Scrum Master será asumido por el Jefe del Departamento de Sistemas ya que cumple con el perfil establecido, además de la experiencia que posee en cuanto a los procesos de la institución tiene un amplio conocimiento del Sistema Financiero Eonx, permitiendo identificar las necesidades y ayudar al Product Owner a especificar los requerimientos. En el caso que se designe a otro colaborador como Scrum Master, este debe cumplir con las características previstas en el perfil, de lo contrario recibirá una inducción con la finalidad que conozca el marco de trabajo de Scrum.

Tabla 4: Rol Scrum Team

SCRUM TEAM (EQUIPO DE DESARROLLO)	
	DESCRIPCIÓN
	El Scrum Team está compuesto por el diseñador, el programador y el tester que hacen el trabajo necesario para entregar el Incremento de Producto.
PERFIL	RESPONSABILIDADES
<ul style="list-style-type: none"> • Ser profesionales con conocimiento en su disciplina. • Tener "vocación" (la buena predisposición no alcanza) para trabajar en equipo • Capacidad de auto-gestión 	<ul style="list-style-type: none"> ✓ Llevar el Backlog de producto, a desarrollos potencialmente funcionales y operativos. ✓ Hacer uso de herramientas que permitan el diseño y codificación del producto de software. ✓ Participar activamente en los eventos de Scrum. ✓ Entender a detalle los requerimientos del Product Owner ✓ Seguir los valores y principios de Scrum

Elaborado por: Fernando Arias

El Scrum Team está formado por el Analista – Programador quien realizará la implementación (ciclo de vida) del producto de software y como tester el Administrador del Sistema quien cumple además con el rol de Scrum Master. Sin embargo se propone que estos roles pueden ser asumidos por colaboradores afines al perfil establecido.

Adicionalmente se hace mención a los usuarios finales que representa los grupos de interés cuyas necesidades tiene que satisfacer el proyecto. Está claro que este grupo no tiene asignado un rol específico dentro de Scrum sin embargo es importante tomarlos en cuenta.

Definición de Artefactos

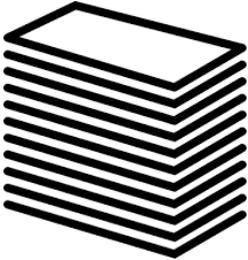
La definición de los artefactos hacer adaptados en el proyecto reúnen las características propias de Scrum, sin embargo se han modificado para acoplarlas a la imagen institucional de la Cooperativa pero siempre manteniendo su esencia.

Tabla 5: Herramienta Historia de Usuario

USER HISTORY (HISTORIAS DE USUARIO)	
	DESCRIPCIÓN
	<p>Las historias de usuario no son propias de Scrum, sin embargo su aplicación ayudará a representar los requerimientos del usuario final. Su redacción utiliza el lenguaje común del usuario.</p>
<p>Las historias de usuario se las escribe con el usuario final. Durante una reunión todas las funcionalidades que requiere el usuario final son escritas en forma de historias las mismas que deben ser estimables (tiempo que tomará implementarla), verificable (verificada en cada incremento) y valorada (grado de importancia para el usuario) para lo cual se cuenta con la presencia del Product Owner, la asesoría del Scrum Master y la presencia del Scrum Team.</p>	
-- Ver Anexo --	

Elaborado por: Fernando Arias

Tabla 6: Artefacto Product Backlog

PRODUCT BACKLOG (PILA DEL PRODUCTO)	
	DESCRIPCIÓN
	<p>El Product Backlog es la lista dinámica y evolutiva de funcionalidades ordenadas y priorizadas de los requerimientos establecidos mediante las Historias de Usuario por el Product Owner.</p>
<p>Para preparar el Product Backlog se debe empezar por priorizar y estimar las historias de usuario. El proceso inicia con el Product Owner, quien debe priorizar las historias por valor de negocio, con la finalidad de dar a conocer al Scrum Team que debe empezar a desarrollar primero. Para la estimación se aplica la técnica “Planning Poker”, la cual consiste en asignar un puntaje a cada historia de usuario, dicho puntaje no tiene relación con las horas de trabajo es más bien una proporción de esfuerzo que puede requerir una historia de usuario. Se la realiza mediante el manejo de un mazo de cartas numeradas con la serie Fibonacci (0,1,2,3,5,8,13,20,40,100, ?, ∞). Es importante también delimitar la velocidad del Scrum Team en cuanto a la cantidad de historias que pueden desarrollar en el tiempo establecido de un Sprint. En vista que es la primera ocasión que el equipo de desarrollo del departamento de sistemas de Cooperativa ANDINA desarrollará software bajo el marco de trabajo de Scrum,</p>	

se marcará un antecedente que permita delimitar la velocidad para futuros proyectos. Para lo cual se procede a tomar tres historias de usuario priorizadas y estimadas, el valor que se obtenga de la suma de la estimación de cada historia será la velocidad del Scrum Team, es decir que esas tres historias serán desarrolladas en el tiempo establecido para el Sprint.

-- Ver Anexo --

Elaborado por: Fernando Arias

Tabla 7: Artefacto Sprint Backlog

SPRINT BACKLOG (PILA DE TAREAS DE LA ITERACIÓN)	
	DESCRIPCIÓN
	El Sprint Backlog es la lista de tareas que el Scrum Team se compromete a desarrollar en un determinado tiempo (1 – 4 semanas).
<p>El proceso inicia después de haber refinado (entender claramente) el Product Backlog, seguidamente se toma una historia del Product Backlog y se la divide en varias tareas a desarrollar en el tiempo definido para completar el Sprint. Cada una de las tareas se las debe estimar en horas (2, 4, 6, 8), si la tarea sobrepasa este tiempo se la debe descomponer en sub-tareas. El programador empieza a trabajar en cada tarea del Sprint Backlog, registra el avance (Por hacer, haciendo, terminado) mediante un tablero virtual. Es importante mencionar que durante el desarrollo del Sprint la lista del Sprint Backlog no se puede modificar. Cualquier modificación se la incluirá en el siguiente Sprint. Iniciado el Sprint se incorpora el evento Daily Meeting.</p>	
-- Ver Anexo --	

Elaborado por: Fernando Arias

Tabla 8: Artefacto Product Increment

PRODUCT INCREMENT (INCREMENTO DEL PRODUCTO)	
	DESCRIPCIÓN
	Es el incremento funcional que se entrega al final de un Sprint permitiendo implantarlo en un ambiente de producción. Conjuntamente con los demás incrementos forman el producto final.
Para desarrollar el incremento de un Sprint el programador debe definir herramientas que le	

permitan realizar el respectivo proceso de ciclo de vida (análisis, diseño, implementación y pruebas). Para lo cual también aplicará otras herramientas como (diagramas UML, y casos de pruebas). El incremento debe figurar como “Terminado” con la finalidad que el Product Owner lo suba a producción. Al concluir con un Sprint únicamente se realizará la presentación del incremento funcional. Al finalizar con todos los Sprint establecidos, el Scrum Team redactará un Acta de Entrega – Recepción del Software.

-- Ver Anexo --

Elaborado por: Fernando Arias

Definición de Eventos

Los eventos que se realizarán para la adaptación del marco de trabajo de Scrum se ajustarán acorde a las necesidades del equipo de trabajo, pero siempre manteniendo las características que establece Scrum.

Tabla 9: Evento Sprint Planning

SPRINT PLANNING (PLANIFICACIÓN DE SPRINT)	
	DESCRIPCIÓN
	Esta reunión se la realiza en cada inicio de un Sprint, su objetivo es determinar.
<p>Las actividades a realizar en esta Reunión son:</p> <ul style="list-style-type: none"> • Product Owner presenta las historias de usuario priorizadas • Se aplica la técnica Planning Poker para estimar las historias • Seleccionar las historias del Product Backlog a ser implementadas • Dividir las historias de usuario en tareas pequeñas. • Estimar en horas el desarrollo de las tareas de una historia. • Pasar las tareas al formato del Sprint Backlog para su gestión. 	
Esta reunión es dinámica por ende no requiere generar ningún artefacto, se trabaja sobre las historias de usuario y el Product Backlog. El tiempo máximo de la reunión es de 4 horas.	

Elaborado por: Fernando Arias

Tabla 10: Evento Daily Meeting

DAILY MEETING (REUNIÓN DIARIA)	
	DESCRIPCIÓN
	Es una reunión diaria que permite abarcar un conocimiento más amplio del avance del Sprint que se encuentra en desarrollo.
<p>Formato de la Reunión:</p> <ul style="list-style-type: none"> • <i>Periodicidad:</i> diario • <i>Cuando:</i> a partir de las 11:00 horas • <i>Duración:</i> máximo 10 minutos • <i>Asistentes:</i> Product Owner, Scrum Master, Scrum Team • <i>Convocar:</i> Mediante el Messenger Institucional • Todos de pie en el Departamento de Sistemas <p>La dinámica de la reunión es Informar:</p> <ul style="list-style-type: none"> • Los avances del día anterior • Las tareas que se realizarán este día • Impedimentos o amenazas 	
La reunión como se la describe es Informativa, no requiere ningún artefacto ya que sobrecargaría el marco de trabajo ágil de Scrum.	

Elaborado por: Fernando Arias

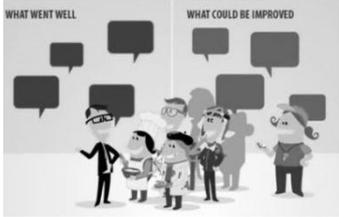
Tabla 11: Evento Sprint Planning

SPRINT REVIEW (REVISIÓN DE SPRINT)	
	DESCRIPCIÓN
	Es una reunión en la cual el Scrum Team conjuntamente con el Product Owner y el Scrum Master revisan los resultados obtenidos del Sprint.
<p>El motivo de la reunión es:</p> <ul style="list-style-type: none"> • Mostrar la funcionalidad del incremento del Sprint finalizado. 	
En esta reunión no se requiere artefacto alguno, ya que la demostración del incremento del Sprint se lo realizará en tiempo real con el Product Owner y el Scrum Master. Tiempo	

máximo para la reunión, 30 minutos.

Elaborado por: Fernando Arias

Tabla 12: Evento Sprint Retrospective

SPRINT RETROSPECTIVE (REUNIÓN DE RETROSPECTIVA)	
	DESCRIPCIÓN
	Reunión en la cual se toma en cuenta que es lo que salió bien y que salió mal, así como aceptar sugerencias para mejorar las cosas en el futuro.
<ul style="list-style-type: none"> • La reunión de retrospectiva debe durar máximo 30 minutos. • Documentar lo que salió bien y lo que salió mal durante el Sprint • Mencionar sugerencias para mejorar el proceso 	
-- Ver Anexo --	

Elaborado por: Fernando Arias

Tabla 13: Evento Refinamiento del Backlog

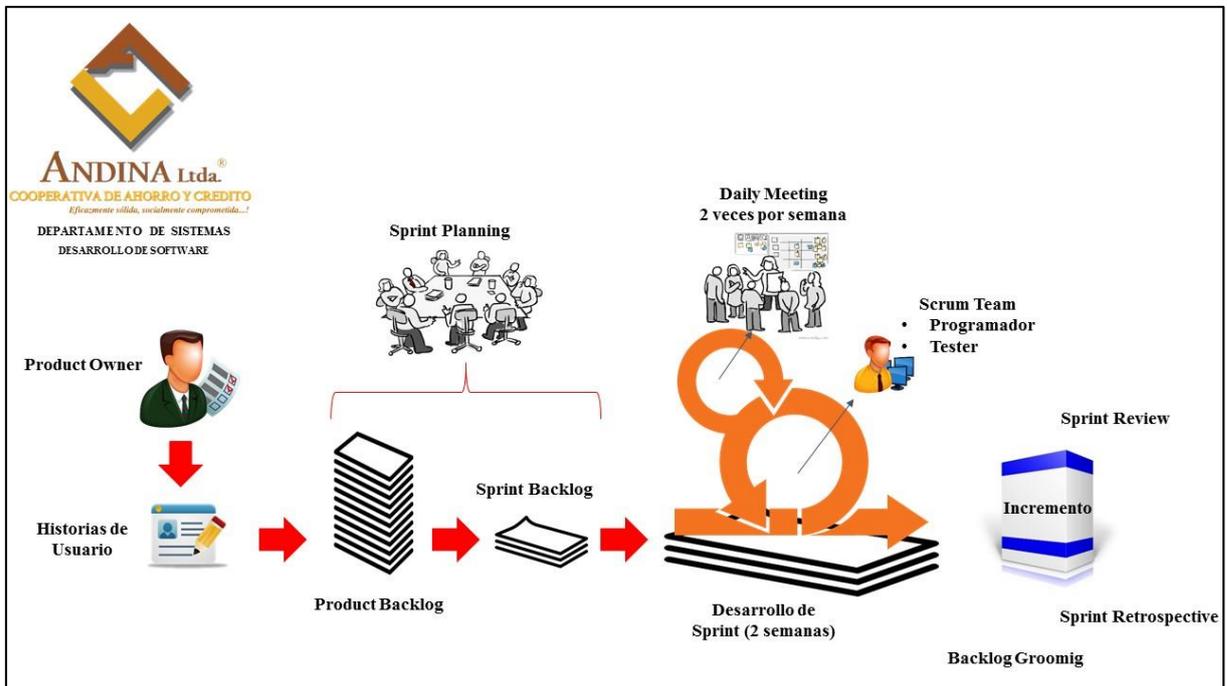
BACKLOG GROOMING (REFINAMIENTO DE BACKLOG)	
	DESCRIPCIÓN
	Es la reunión en la que el Product Owner, el Scrum Master y el Scrum Team se reúnen para revisar el Product Backlog y añadir, retirar o reestimar historias de usuario.
<p>En esta reunión el Product Owner es quien propone cambios al Product Backlog, estos cambios se los debe realizar conjuntamente en el siguiente Sprint. Importante mencionar que no se puede cambiar las tareas que se estén desarrollando durante la iteración. Las nuevas funcionalidades deben estar priorizadas y estimadas.</p> <p>Dinámica de la Reunión:</p> <ul style="list-style-type: none"> • <i>Duración:</i> 1 hora (depende de las nuevas historias de usuario) • <i>Asistentes:</i> Scrum Master – Scrum Team – Product Owner • <i>Periodicidad:</i> 1 por cada Sprint • Se debe priorizar y estimar las nuevas historias 	
En esta reunión se actualiza el Product Backlog, sin intervenir en el trabajo que se está	

desarrollando en aquel momento. Se evidencia la flexibilidad de Scrum ante los requisitos cambiantes.

Elaborado por: Fernando Arias

PROCESO DE SCRUM ADAPTADO A EQUIPOS DE UN SOLO PROGRAMADOR

Imagen 1: Proceso de Scrum adaptado a equipos de un solo programador



Elaborado por: Fernando Arias

Quizás el proceso de Scrum que se describe es idéntico a otros, sin embargo difiere en la forma en la que cada uno de sus elementos es aplicado al contexto de la investigación.

MARCO DE TRABAJO TÉCNICO DE SCRUM vs MARCO DE TRABAJO DE SCRUM ADAPTADO A EQUIPOS DE UN SOLO PROGRAMADOR

Tabla 14: Scrum Técnico vs Scrum Adaptado

Metodología Proceso	SCRUM TÉCNICO	SCRUM (PROPUESTA)
ROLES	<p>Product Owner</p> <ul style="list-style-type: none"> Es quien toma las decisiones del cliente. Su responsabilidad es el valor del producto. Para simplificar la comunicación y toma de decisiones es necesario que 	<p>Product Owner</p> <ul style="list-style-type: none"> Determina las prioridades de las historias de usuario con asesoría del Scrum Master y el Scrum Team.

	este rol recaiga en una única persona.	
	Scrum Master <ul style="list-style-type: none"> Gestiona y facilita la ejecución de las reglas de Scrum 	Scrum Master <ul style="list-style-type: none"> Gestiona y facilita la ejecución de las reglas de Scrum, colabora dinámicamente con el Product Owner y es parte del Scrum Team.
	Scrum Team <ul style="list-style-type: none"> Construye el producto. Número establecido para conformar el equipo entre 3 y 8 personas 	Scrum Team <ul style="list-style-type: none"> Construye el producto. El número de miembros establecido 2. Programador ayuda del Scrum Master como Tester. Además se toma en cuenta a los interesados ya que aportan también con ideas que favorecen al equipo.
ARTEFACTOS	Product Backlog <ul style="list-style-type: none"> Relación de requisitos del producto. No es necesario excesivo detalle. Priorizados, lista abierta a todos los roles. El Product Owner es su responsable y quien decide. 	Product Backlog <ul style="list-style-type: none"> Para su elaboración se hace uso de las historias de Usuario. Cada historia es priorizada y estimada mediante la técnica “Planning Poker”. Dinámica y evolutiva. Product Owner es su responsable. Se genera un formato personalizado.
	Sprint Backlog <ul style="list-style-type: none"> Requisitos comprometidos por el equipo para el Sprint con nivel de detalle suficiente para su ejecución. 	Sprint Backlog <ul style="list-style-type: none"> Lista de requerimientos que serán implementados por el equipo de desarrollo para su seguimiento se emplea un formato personalizado.
	Product Increment <ul style="list-style-type: none"> Parte del producto desarrollado en un sprint. En condiciones de ser usada (pruebas, codificación limpia y documentada) 	Product Increment <ul style="list-style-type: none"> Resultado funcional de un Sprint. Se aplica las etapas de (análisis, diseño, implementación y pruebas). Se definen herramientas como los diagramas UML y un formato para los casos de uso.
EVENTOS	Sprint Planning <ul style="list-style-type: none"> 1 jornada de trabajo (máx.) el propietario del producto explica las prioridades. El equipo estima el esfuerzo de los requisitos prioritarios y se elabora la pila del Sprint. 	Sprint Planning <ul style="list-style-type: none"> Se establece un tiempo de 4 horas como máximo para que se expliquen los requisitos, se estime el esfuerzo y se elabora la pila del Sprint.

	<p>Daily Meeting</p> <ul style="list-style-type: none"> • 15 minutos máximos. Responsabilidad del equipo. Cada miembro expone lo que hizo ayer, lo que va hacer hoy, si tiene o probé problemas. Se actualiza la pila del Sprint. 	<p>Daily Meeting</p> <ul style="list-style-type: none"> • 10 minutos máximos de reunión. • El programador expone lo que hizo ayer, lo que hará hoy y si tiene algún tipo de inconvenientes. Se actualiza la pila del Sprint.
	<p>Sprint Review</p> <ul style="list-style-type: none"> • Informativa máx. 4 horas, presentación del incremento, planteamiento de sugerencias y anuncio del próximo Sprint. 	<p>Sprint Review</p> <ul style="list-style-type: none"> • Máximo 2 horas. • Se realiza una demostración del incremento. • Se atiende sugerencias por parte del Product Owner • Se anuncia el próximo Sprint.
	<p>Sprint Retrospective</p> <ul style="list-style-type: none"> • El equipo automatiza la forma de trabajo identificación de fortalezas y debilidades. Refuerzo de las primeras y mejoras de las siguientes. 	<p>Sprint Retrospective</p> <ul style="list-style-type: none"> • Se mantiene el mismo proceso, adicionalmente se genera un artefacto para recopilar la información del evento.
	<p>Backlog Grooming</p> <ul style="list-style-type: none"> • No existe en el Scrum Técnico 	<p>Backlog Grooming</p> <ul style="list-style-type: none"> • Se lo toma en cuenta con la finalidad de ir refinando la pila del producto, mediante este evento se pueden añadir, modificar o eliminar funcionalidades sin afectar el desarrollo del Sprint. Dura máx.1 hora y se lo realiza una vez por Sprint.

Elaborado por: Fernando Arias

La información proporcionada con la que se elaboró la matriz se obtuvo de (Palacio, 2015), la cual se comparó con la propuesta planteada en el proyecto, el objeto de esta matriz es identificar las diferencias y similitudes entre la propuesta y Scrum Técnico. Haciendo un análisis simplificado se diferencia en el número de profesionales que integran el equipo de desarrollo (1 programador), el Scrum Master colabora como Tester pero no es tomado en cuenta dentro del equipo. El tiempo de los eventos también es un factor que influye en las diferencias con Scrum Técnico, pues el tiempo por fuerza mayor se la debe reducir para no afectar el horario laboral de los colaboradores que conforman el equipo de trabajo siendo este

tiempo de alta productividad, en esta parte hay que acotar que la propuesta precisa que todo el equipo se reúna en los eventos con la finalidad de fomentar la comunicación en el equipo como una buena práctica.

11. ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS

Con la finalidad de cumplir con los objetivos planteados en el presente proyecto de investigación, se realizó el levantamiento de información de la situación actual del lugar en donde se evidencia la problemática a través de la aplicación de técnicas e instrumentos de investigación, con el propósito de mejorar el proceso de desarrollo de software en el departamento de sistemas de la Cooperativa ANDINA y obtener datos que permitieron describir de forma clara las etapas del proceso tradicional existente en aquel momento. A continuación se presenta el análisis de la observación directa y la entrevista aplicada.

Observación Directa: Mediante esta técnica se observó que las etapas para el desarrollo de software se las aplicaba de forma ambigua. De tal manera que para la fase de recopilación de requerimientos se los hacía vía correo electrónico o mediante una plática con el personal del área de sistemas, los mismos que no contaban con un documento formal para tal efecto únicamente se los escribía en un cuaderno, el análisis era completamente nulo. Para el diseño no aplicaban ningún tipo de diagrama, flujo de procesos o relación de base de datos, prácticamente el diseño para ellos era maquetar el interfaz de usuario. En cuanto a la fase de implementación se observó que hacían uso de editores de código como el Notepad++ o el ConText, para crear las tablas en la base de datos se accedía remotamente al servidor de datos mediante el putty, la codificación no tenía inconveniente alguno; sin embargo se observó que el programador debía entregar todo el producto de software en una fecha límite, la misma que no se cumplía y se aplazaba por algunas días e incluso semanas, provocando el disgusto del usuario quien solicitó el software. Para la fase de pruebas el programador era “juez y parte”, mientras codificaba iba realizando las pruebas no contaba con una herramienta que le permita registrar y validar los casos de pruebas. Nunca se observó que el usuario quien solicitó el software acudiese a preguntar cómo iba el desarrollo. Una vez finalizado el producto de software era puesto a producción sin la revisión de quien lo solicitó, únicamente al juicio del programador y tampoco existía un documento formal para la entrega del software, se lo hacía vía correo electrónico confirmando que el requerimiento estaba ya en producción. Una vez en producción y al ser usado por el usuario se sugerían cambios y se evidenciaban errores.

Entrevista: Mediante esta técnica se corroboró la información observada, ya que la fuente de información veraz fue el Jefe de Sistemas. De forma concreta el análisis de la entrevista arrojó que no se aplicaba una metodología para el desarrollo de software debido a la falta de tiempo y por las distintas actividades que están relacionadas a sus funciones, así como también influía el hecho que era la única persona en el departamento de sistemas.

Aplicación de la propuesta en el desarrollo del aplicativo

“Control de Denominaciones de Dinero” para el Área de Cajas de Cooperativa Andina.

Los roles se definieron de la siguiente manera:

ROL	COLABORADOR	CARGO
Product Owner	Tlga. Jessica Rocha	Jefe del Departamento de Operaciones
Scrum Master	Ing. Fernando Pacheco	Jefe del Departamento de Sistemas
Scrum Team	Sr. Fernando Arias	Analista - Programador

Se convocó al equipo de trabajo (Product Owner, Scrum Master, Scrum Team) a una reunión previa, con la finalidad de escribir las historias de usuario.

En la reunión de Sprint Planning se procedió a priorizar y a estimar las historias de usuario mediante la técnica del Planning Poker que consistió en tomar las historias de usuario y asignarles un valor de la serie Fibonacci para estimar el esfuerzo (días) que tomará en implementar una historia, es importante mencionar que el programador con la experiencia adquirida tiene gran parte de la responsabilidad para establecer el esfuerzo; este proceso se lo aplicó con la finalidad de escribir el Product Backlog. -- Ver Anexo --. Además se procedió a dividir a cada historia del Product Backlog en tareas ejecutables y estimadas en horas, especificándolas en el formato creado para el efecto, las cuales se convirtieron en el compromiso del Scrum Team. -- Ver Anexo --. Finalmente se estableció que iniciado el desarrollo de las tareas del primer Sprint, al día siguiente se realizará la Reunión diaria a partir de las 11:00 am. En la cual debe estar presente todo el equipo de trabajo, se dio a conocer las actividades que se deben cumplir y la dinámica del evento. Terminada la reunión el Scrum Team empezó con la codificación.

Al siguiente día el Scrum Master procedió a recordar la asistencia a la reunión diaria a darse a las 11:00 en el Departamento de Sistemas. Asistió todo el equipo de trabajo, durante

esta primera reunión diaria, el Scrum Team dio a conocer los avances del día lunes y martes de las tareas desarrolladas del primer Sprint. Hizo una breve descripción de las tareas que desarrollará entre el miércoles y jueves, en cuanto a impedimentos o amenazas detalló que las demás funciones que realiza (soporte técnico, help desk, etc.) estaba afectando las horas estimadas para el desarrollo de las tareas. Tomó la palabra el Scrum Master y mencionó que el tiempo que dure el desarrollo del Sprint él atenderá todos los requerimientos solicitados por los demás colaboradores.

El Scrum Team empezó con el proceso de desarrollo haciendo uso de las herramientas para el efecto.

Las demás reuniones diarias se las realizó de la misma manera. Con ayuda del Scrum Master, el Scrum Team pudo avanzar sin contratiempos.

Una vez concluida con la codificación del Sprint, el Scrum Team solicita al tester (Jefe de Sistemas/Scrum Master) la colaboración para realizar los casos de pruebas. El Scrum Master para la validación de las pruebas hace uso del formato correspondiente. – Ver Anexo --

Se cumplieron los 12 días establecidos para terminar el Sprint, el Scrum Master convocó a la reunión de Revisión del Sprint al equipo de trabajo, en la cual el Scrum Team una hora antes preparó el ambiente en el sistema y base de datos a prueba para exponer el primer incremento funcional. Durante la demostración surgieron algunos cambios por parte del Product Owner los mismos que se los realizaron en el mismo momento en vista que no eran muy complejos. El mismo día se llevó a cabo la reunión de Retrospectiva, las evidencias se las redactó en el formato para el efecto. --Ver Anexo --

Se inició con el segundo Sprint siguiendo el mismo procedimiento.

Para el Tercer Sprint se procedió de la misma manera, al final de este Sprint se integró todo el aplicativo. -- Ver Anexo --

Como se planteó en la propuesta, la entrega formal del producto de software al Product Owner se realizará una vez concluida con todas las iteraciones que en este se haya establecido. Mediante un Acta de Entrega – Recepción del producto de Software. -- Ver Anexo --

Para este producto de software no se realizó un refinamiento del Backlog, ya que el Product Owner definió claramente sus requerimientos y procuró que sean entendidos por el Scrum Team.

Los resultados obtenidos después de haber adaptado el marco de trabajo de Scrum en el equipo de un solo programador del Departamento de Sistemas de la Cooperativa ANDINA Ltda., se los detalla a continuación:

Mediante la adaptación del marco de trabajo de Scrum, se conforma un equipo que trabajará de manera continua desde el inicio hasta el final del producto de software, fomentando una comunicación más fluida entre los miembros del equipo que permita entender los requerimientos y mediante los valores de Scrum llevarlos a un nivel altamente productivo. Con la finalidad de documentar el proceso de desarrollo se emplearon artefactos que guardan características específicas de Scrum, dándoles un formato acorde a la imagen institucional facilitaron la especificación de requerimientos y gestión de tareas, es importante mencionar que pese a la gran cantidad de plantillas que existen para documentar Scrum se han adaptado las más significativas dentro del marco de trabajo (Historias de usuario, Product Backlog, Sprint Backlog, Sprint Retrospective, entrega de incremento), con la finalidad de mantener el principio ágil “software funcionando sobre documentación extensiva”.

Los eventos se desarrollaron de manera que no influya con las labores de los colaboradores que forman el equipo de trabajo. Manteniendo hasta el último detalle las especificaciones que precisa Scrum. Estas reuniones permitieron mantener informados acerca del desarrollo de cada uno de los Sprints por parte del Scrum Team, especialmente al Product Owner que estuvo siempre al tanto de los avances y fortaleció la comprensión de los requisitos en cada reunión diaria, la participación del Product Owner fue dinámica y productiva para todo el equipo. La reunión del Backlog Grooming que se realizó al final de cada Sprint permitió aplicar algunas sugerencias por parte del Product Owner pero estas fueron muy superficiales ya que se trataba del texto de las alertas que despliega el aplicativo, algo que realmente llamó la atención al Scrum Team, ya que por lo general los cajeros pasan por alto las alertas que despliega el Sistema Econx, la sugerencia del Product Owner fue en el caso puntual de los excedentes de caja que la alerta que despliegue sea constante hasta que el cajero realice el respectivo proceso del registro de excedente, pues es un riesgo para la Cooperativa mantener cantidades altas de dinero en las cajas.

La propuesta que se adaptó fue de gran acogida en la cooperativa, ya que desde un comienzo se contó con la autorización del Gerente General y la predisposición de los colaboradores y el producto de software implementado está en producción y es 100% funcional.

12. IMPACTOS (TÉCNICOS, SOCIALES, AMBIENTALES O ECONÓMICOS)

La aplicación del proyecto de investigación en el Departamento de Sistemas de la Cooperativa Andina, generó un impacto Técnico en el proceso de desarrollo de software, mediante la adaptación del marco de trabajo de Scrum al equipo de un solo programador, consiguiendo con aquello un proceso sistemático que junto con los valores y principios de Scrum genere buenas prácticas para el desarrollo de software eficiente. De la misma manera afectó a los Manuales Técnicos del Departamento de Sistemas ya que la propuesta del proyecto será incluido en el Manual de Procesos y Procedimientos.

El impacto que más relevancia tuvo es en lo Económico, ya que mediante la adaptación del marco de trabajo de Scrum se puede implementar software eficiente, de esta forma no incurrirá en altos costos para la adquisición de software de terceras personas, en el caso de Cooperativa Andina el software se lo adquiere a la empresa Avmei Ltda., (proveedores del sistema financiero Econx, por alrededor de \$300.00 por aplicativo).

Al finalizar la adaptación del marco de trabajo de Scrum a un caso práctico, se evidencia claramente que se verificó la hipótesis planteada al inicio del Proyecto de Investigación. El mismo que permitió generar buenas prácticas para el desarrollo del software eficiente dentro del Departamento de Sistemas de Cooperativa Andina Ltda., en su equipo de desarrollo con un solo programador, el aplicativo desarrollado se gestionó e implemento bajo las reglas, valores y principios de Scrum, dando como resultado un software eficiente y funcional que está en producción. Adicional a ello es importante recalcar que el equipo de desarrollo del Departamento de Sistemas implementará software que requiera la Cooperativa, sin que esta invierta altos costos en la adquisición de productos de software de terceras personas.

13. PRESUPUESTO PARA LA PROPUESTA DEL PROYECTO

Tabla 15: Costos directos

DESCRIPCIÓN DE ACTIVIDADES	UNIDAD	VALOR/UNIDAD	VALOR
Copias de los libros consultados	87	0,02	1,74
Resma de Papel bond	2	3,50	7,00
Servicio de internet (mensual)	2	26,00	52,00
Trabajo Intelectual	400 h	2,02	808,00
Frascos de tinta (4 colores)	4	3,00	12,00
Disco duro	1	95,00	95,00
Suministros de oficina (varios)	1	20,00	20,00
Anillados del proyecto	4	1,00	4,00
TOTAL			999,74

Elaborado por: Fernando Arias

Tabla 16: Costos Indirectos

DETALLE O DESCRIPCIÓN	VALOR
Alimentación	80,00
Transporte	60,00
Imprevistos (10%)	113,97
TOTAL	253,97

Elaborado por: Fernando Arias

Cálculo imprevistos:

$$\mathbf{GD + GI = 999.74 + 140.00 = 1139.74}$$

$$\mathbf{10\% DE IMPREVISTOS = 1139.74 \times 10\% = 113,97}$$

$$\mathbf{GT = 1139.74 + 113,97 = 1253,71}$$

Tabla 17: Total detallado

DETALLE O DESCRIPCIÓN	VALOR
Costo Directos	999,74
Costos Indirectos	253,97
COSTO TOTAL DEL PROYECTO	1253,71

Elaborado por: Fernando Arias

14. CONCLUSIONES Y RECOMENDACIONES

Luego de haber recopilado la debida información, estructurarla y analizarla, se adquirió un conocimiento más amplio en cuanto a la adaptación de Scrum en equipos de un solo programador que paso a ser aplicado en una propuesta la misma que será demostrada mediante su aplicación en un caso real, el desarrollo de un aplicativo para el área de cajas denominado “Control de Denominaciones”, dando como conclusiones las siguientes:

14.1. Conclusiones

- Se consultó distintas fuentes bibliográficas, obtenido una base teórica suficiente que permitió estructurar la propuesta que será implementada para el desarrollo de software en el departamento de sistemas de la Cooperativa ANDINA. No solo se obtuvo teoría para fundamentar la propuesta sino que además se basó en trabajos relacionados de los cuales se abstraigo la experiencia adquirida de otros autores para fortalecer y aplicarlos en la propuesta planteada.
- En vista de la relación directa que se mantuvo en el lugar de la investigación, se recopiló información suficiente mediante la observación de la situación real de ese momento, además se aplicó una entrevista al Jefe de Sistemas en cuanto al proceso que se sigue para el desarrollo del software en el departamento de sistemas de Cooperativa ANDINA.
- Se definió la estructura del marco de trabajo de Scrum mediante el análisis de sus elementos, abstrayendo características esenciales de cada uno de ellos. En primera instancia se especificó un perfil general que permitirá definir y asignar los roles para las personas que conformarán el equipo de trabajo; los artefactos que se generan en el proceso de desarrollo guardan estricta relación a lo establecido por Scrum, sin embargo se les ha dado un formato apegado a la imagen institucional; en cuanto a los eventos que plantea Scrum hubo modificaciones significativas en los tiempos y en las roles que deben participar en ellos. La adaptación del marco de trabajo en el equipo de un solo programador cumple con las reglas, valores y principios de Scrum, se mantuvo ciertas características propias de los elementos ya que si se los modifica se perdería la esencia del marco de trabajo de Scrum.

- Definir la estructura que será adaptada a equipos de un programador en el departamento de sistemas de Cooperativa ANDINA se logró sin dificultad alguna, en vista que los proyectos de software que allí se implementan son pequeños pero con un alto desempeño funcional.
- La propuesta que se planteó fue aplicada en un caso práctico, demostrando que la adaptación del marco de trabajo de Scrum en equipos de un solo programador permite implementar un proceso sistemático para el desarrollo de software de calidad bajo las reglas, valores y principios de una de las metodologías ágiles más aplicada en la industria del software. Demostrando además que Scrum es flexible y se adapta a cualquier contexto en donde se lo aplique para la gestión de proyectos de software, como es el caso del departamento de sistemas de Cooperativa ANDINA en la cual su adaptación generó un impacto técnico ya que éste, pasó a formar parte del Manual de Procesos y Procedimientos del área. Su impacto en el factor económico de la Cooperativa se lo ve reflejado en la reducción de la inversión en cuanto a la adquisición de software de terceros. Como un aporte adicional la aplicación del marco de trabajo de Scrum reduce la utilización de recursos como el papel, carpetas e impresiones ya que Scrum no se basa en la documentación excesiva sino en la agilidad y eficiencia del equipo de trabajo.

14.2. Recomendaciones

- Para adaptar Scrum a la gestión de proyectos de software se recomienda mantener la esencia del marco de trabajo de la metodología, seguir y aplicar sus reglas, valores y principios de agilidad, caso contrario se determinaría que no se está aplicando Scrum.
- Scrum define los artefactos esenciales que permiten la gestión del proyecto de software, se puede hacer uso de artefactos externos si la finalidad es tener una documentación completa, sin embargo la excesiva documentación atenta contra el principio de agilidad “software funcionando sobre documentación extensiva”
- El ambiente en la cual se desenvuelve el equipo que adapta el marco de trabajo de Scrum puede ser la mejor; contar con recursos, tiempo, etc., pero si no se cuenta con una comunicación fluida entre los roles que integran el equipo no se está generando buenas prácticas, para lo cual se recomienda fomentar la comunicación y practicarla a diario.

- Para ambientes donde no se aplica un marco de trabajo ágil para el desarrollo de software y se sigue un proceso tradicional, se recomienda socializar la metodología para ser aplicada mediante una capacitación a los colaboradores, para que tengan el conocimiento y participen dinámicamente durante el proceso.

15. BIBLIOGRAFÍA

- Alaimo, M. (05 de Febrero de 2010). *Programadores Ágiles. Transición*. Recuperado el 02 de Mayo de 2016, de Martín Alaimo: <http://www.martinalaimo.com/es/blog/programadores-agiles>
- Alende, A. (27 de Mayo de 2010). *DOCPLAYER*. Recuperado el 02 de Mayo de 2016, de La Utilización de los Métodos Ágiles. en las Empresas de Desarrollo de Software de Argentina. Andrea: <http://docplayer.es/1449119-La-utilizacion-de-los-metodos-agiles-en-las-empresas-de-desarrollo-de-software-de-argentina-andrea-n-alende-universidad-caece-mar-del-plata.html>
- Armijos, A., Fiallos, A., Villavicencio, M., & Abad, C. (29 de Diciembre de 2015). *Revista Tecnológica ESPOL*. Recuperado el 02 de Mayo de 2016, de Aplicación de Scrum en la construcción de un simulador de Redis: <http://www.rte.espol.edu.ec/index.php/tecnologica/article/view/446/311>
- Avmei Ltda. (2013). *Anexo A. Modulos del Sistema*. Quito.
- Bautista, J. (24 de Junio de 2009). *Universidad Unión Bolivariana*. Recuperado el 02 de Mayo de 2016, de Ingeniería de Software. Programación Extrema XP: <http://ingenieriadesoftware.mex.tl/images/18149/PROGRAMACI%C3%93N%20EXTREMA.pdf>
- Calderón, A., Castillo, M., & Bercovich, N. (31 de Julio de 2015). *Vicepresidencia de la República de Ecuador*. Recuperado el 02 de Mayo de 2016, de La cadena del software en Ecuador Diagnóstico, visión estratégica y lineamientos de política: <http://www.vicepresidencia.gob.ec/wp-content/uploads/2015/07/Resumen-Cadena-Software.pdf>
- Campillo, R. (31 de 08 de 2016). *El método científico*. Obtenido de WordPress: <https://eleklektiko.files.wordpress.com/2008/09/el-metodo-cientifico.pdf>
- Computerworld*. (2016). Recuperado el 02 de Mayo de 2016, de El software ágil tiene su comunidad en Ecuador: <http://computerworld.com.ec/actualidad/tendencias/98-el-software-agil-tiene-su-comunidad-en-ecuador.html>

- de la Cuesta, O. (21 de Mayo de 2014). *Ejemplos de AJAX sencillos. Recordando conceptos*. Recuperado el 02 de Mayo de 2016, de Blog de Oscar de la Cuesta: <http://www.palentino.es/blog/ejemplos-de-ajax-sencillos/>
- Definición ABC. (s.f). *Definición de HTML*. Recuperado el 02 de Mayo de 2016, de <http://www.definicionabc.com/tecnologia/html.php>
- Doupovec, M. (Julio de 2010). *Tecnicas de la Investigacion*. Obtenido de Blogspot: <http://metodologia02.blogspot.com/p/tecnicas-de-la-investigacion.html>
- Elismen. (08 de Octubre de 2014). *Aequitectura Cliente/Servidor*. Recuperado el 02 de Mayo de 2016, de Sharebooks: <http://sharebooks.esy.es/teleprocesos/arquitectura-clienteservidor/>
- Herrera, E., & Valencia, L. (28 de Mayo de 2007). *dialnet.unirioja.es*. Recuperado el 02 de Mayo de 2016, de Del Manifiesto Ágil sus Valores y Principios: <https://dialnet.unirioja.es/descarga/articulo/4809645.pdf>
- Instituto Nacional de Tecnologías de la Comunicación. (19 de Junio de 2009). *Ingeniería del Software: Metodologías y Ciclos de vida*. Recuperado el 01 de Mayo de 2016, de Scribd: <https://es.scribd.com/doc/62931905/Guia-de-Ingenieria-Del-Software>
- Limachi, B. (14 de Septiembre de 2012). *Metodología RUP*. Recuperado el 01 de Mayo de 2016, de SlideShare: <http://es.slideshare.net/bernardolimachi/metodologia-rup-14288208>
- López, J. (06 de Julio de 2012). *Desarrollo de un Complemento para Firefox*. Recuperado el 02 de Mayo de 2016, de PepeDesign: http://www.pepedesigns.es/wp-content/uploads/2013/06/PFC-Jose_Luis_Lopez_Dieguez-2011_2012.compressed.pdf
- Martínez, N., Ramón, H., & Bertone, R. (24 de Julio de 2012). *SEDICI Repositorio Institucional de la UNLP*. Recuperado el 02 de Mayo de 2016, de Aplicabilidad de competisoft a partir de un método ágil como scrum: http://sedici.unlp.edu.ar/bitstream/handle/10915/23722/Documento_completo.pdf?sequence=1
- Najera, M. (26 de Septiembre de 2013). *Pymes Practicas Comunidad de aprendizaje*. Recuperado el 02 de Mayo de 2016, de Dinamizando el mercado de software de América Latina y el Caribe:

<http://www.fomin.org/pymespracticass/Blog/ArtMID/4214/ArticleID/985/Dinamizando-el-mercado-de-software-de-Am233rica-Latina-y-el-Caribe.aspx>

Neher, R. (26 de Septiembre de 2015). *Introducción JavaScript*. Recuperado el 02 de Mayo de 2016, de Mozilla Developer Network: <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n>

Núñez, C., & Sánchez, L. (12 de Diciembre de 2011). *Serie Científica-Universidad de las Ciencias Informáticas*. Recuperado el 02 de Mayo de 2016, de Marco de procesos de desarrollo de software orientado a entornos productivos pequeños especializados en soluciones de gestión: <http://publicaciones.uci.cu/index.php/SC/article/view/556/488>

Palacio, J. (24 de Abril de 2015). *Gestión de proyectos Scrum Manager*. Recuperado el 02 de Mayo de 2016, de Scrum Manager: http://www.scrummanager.net/files/gestion_proyectos_scrum_manager.pdf

Ramirez, D. (5 de Noviembre de 2012). *Modelo Incremental*. Recuperado el 01 de Mayo de 2016, de Blue: <http://diana09.blogspot.com/2012/11/modelo-incremental.html>

Rojas, C. (06 de Noviembre de 2013). *Descripción general de PHP*. Recuperado el 02 de Mayo de 2016, de CODEJOBS: <https://www.codejobs.biz/es/blog/2013/11/06/descripcion-general-de-php>

Royal, A. (20 de Abril de 2016). *Informix exposicion-2016*. Recuperado el 02 de Mayo de 2016, de SlideShare: <http://www.slideshare.net/anibalknightblackzero/informix-exposicion2016>

Schwaber, K., & Sutherland, J. (24 de Noviembre de 2014). *SCRUM GUIDES*. Recuperado el 02 de Mayo de 2016, de La Guía de Scrum: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>

Scooter Software. (23 de Febrero de 2013). *Beyond Compare*. Recuperado el 02 de Mayo de 2016, de CCM Benchmark group: <http://es.ccm.net/download/descargar-2321-beyond-compare>

Serna, E. (04 de Abril de 2014). *Libro Blanco de la Ingeniería de Software*. Recuperado el 01 de Mayo de 2016, de Fundación Instituto Antioqueño de Investigación - IAI: <http://fundacioniai.org/Libro3.pdf>

- Sommerville, I. (2011). *Ingeniería de Software* (Novena ed.). Mexico D.F: Pearson Educación de México, S.A.
- Tatham, S. (15 de Junio de 2016). *PuTTY*. Recuperado el 02 de Mayo de 2016, de CCM Benchmark group: <http://es.ccm.net/download/descargar-395-putty>
- Toapanta, K. (Noviembre de 2012). *Método ágil Scrum, aplicado a la implantación de un sistema informático para le proceso de recolección con tecnología móvil*. Recuperado el 02 de Mayo de 2016, de Repositorios ESPE: <http://repositorio.espe.edu.ec/bitstream/21000/5893/1/T-ESPE-034427.pdf>
- Universidad Politécnica de Valencia. (2012). *Proceso de Desarrollo de Software*. Recuperado el 01 de Mayo de 2016, de Universidad Politécnica de Valencia. Departamento de Sistemas Informáticos y Computación: www.dsic.upv.es/asignaturas/facultad/lsi/doc/IntroduccionProcesoSW.doc
- Vargas, L. (26 de Marzo de 2015). *UNIANDES*. Recuperado el 02 de Mayo de 2016, de Metodología de Desarrollo de Software Dirigida a Equipos de Trabajo Reducidos para su Aplicación en los Proyectos Integradores y Tesis en Uniandes Extensión Ibarra: <http://dspace.uniandes.edu.ec/handle/123456789/1599>
- Velasquéz, C. (06 de 09 de 2008). *Metodología de la Investigación I*. Obtenido de <http://soradelvalle.blogspot.com/>
- Weitzenfeld, A. (2005). *Ingeniería de Software Orientada a Objetos con UML, Java e Internet*. Mexico D.F: Thomson Editores S.A.

ANEXOS

HOJA DE VIDA DEL TUTOR DE TITULACIÓN

Apellidos: Tapia Cerda
Nombres: Verónica Consuelo
C.I.: 050205369-7
Teléfono celular: 099252383
Correo Electrónico: veronica.tapia@utc.edu.ec

FORMACIÓN ACADÉMICA:

- **Posgrado:**
 - Universidad de las Fuerzas Armadas
Magister en Ingeniería de Software

- **Pregrado:**
 - Universidad Regional Autónoma de los Andes
Ingeniería en Sistemas e Informática

HOJA DE VIDA INVESTIGADOR

Apellidos: Arias Herrera
Nombres: Jairo Fernando
C.I.: 050315376-9
Fecha de nacimiento: 03 de Septiembre de 1988
Teléfono: 0995642608
Correo Electrónico: farias@coopandina.fin.ec

FORMACIÓN ACADÉMICA:

- Instituto Tecnológico Superior “Ramón Barba Naranjo”
Bachiller Técnico Industrial en Mecánica Automotriz

Latacunga, 02 de Mayo del 2016

Ing. MSc.

Alfredo Rafael Jácome Tapia

**GERENTE GENERAL DE LA COOPERATIVA DE AHORRO Y CRÉDITO
ANDINA LTDA.**

Presente.-

De mi consideración:

Luego de expresarle un cordial saludo, por medio del presente me permito solicitar a Ud. muy comedidamente su AUTORIZACIÓN para aplicar en el departamento de sistemas el Proyecto de Investigación que titula “**ADAPTACIÓN DE LA METODOLOGÍA DE DESARROLLO DE SOFTWARE SCRUM A EQUIPOS DE UN SOLO PROGRAMADOR.**”, el mismo que es requisito final para la obtención del título de Ingeniero en Informática y Sistemas Computacionales y estará bajo la asesoría y supervisión de la Ing. MSc. Verónica Tapia docente de la Universidad Técnica de Cotopaxi.

El referido proyecto de investigación tiene como propósito proveer un proceso sistemático que genere buenas prácticas para el adecuado desarrollo de software, mediante la adaptación del marco de trabajo de la metodología ágil Scrum en el equipo de desarrollo del departamento de sistemas que cuenta con un solo programador.

Los beneficios que este proyecto aportará a la Cooperativa Andina son múltiples entre los más relevantes pongo a su consideración los siguientes:

- Desarrollar productos de software de calidad, que cumpla a cabalidad los requerimientos de los usuarios en el menor tiempo.

Somos más... más apoyo, más impulso!

Matriz (Sector El Carmen): Av. Benjamín Terán 319 y Antonia Vela Telf.: (03) 2802 850 / 2802 754

Agencia (Sector La Merced): Quijano y Ordóñez y Félix Valencia Tel.: (03) 2802 150 / 2800 544

Latacunga - Ecuador



ANDINA Ltda.
COOPERATIVA DE AHORRO Y CRÉDITO

- Implementar nuevas funcionalidades al sistema Financiero Econx, acorde a las necesidades que se requiera.
- Sistematizar procesos manuales y añadirlos al sistema Financiero Econx para reducir el excesivo uso de otros recursos como el papel.
- Optimizar los procesos que ejecutan los colaboradores como parte de sus actividades diarias para brindar una ágil y oportuna atención a socios y clientes.
- Reducir la inversión en la adquisición de software a terceras personas.

Para la ejecución del Proyecto de Investigación solicito se me conceda los respectivos permisos para realizar las actividades establecidas en el cronograma adjunto, siendo las más relevantes las que describo a continuación:

- Recopilar información mediante la aplicación de Entrevistas y Encuestas.
- Socializar el proyecto con los Jefes Departamentales.
- Mantener reuniones con los colaboradores que intervengan en el Proyecto.
- Evidenciar las reuniones mediante fotografías y nóminas de asistencias.
- Hacer uso de los sistemas y base de datos a prueba.

Consciente de su compromiso con la superación profesional de sus colaboradores, seguro estoy de poder contar con su receptividad la cual me permitirá alcanzar un logro más en vida personal y profesional, la misma que la pondré a consideración para el engrandecimiento de tan noble institución financiera que usted representa.

En espera de una favorable respuesta, expreso mis más sinceros agradecimientos.

- Adjunto cronograma de actividades para su conocimiento.

Atentamente,


Jairo Fernando Arias Herrera
Analista - Programador
COAC - ANDINA LTDA.


ANDINA Ltda.
COOPERATIVA DE AHORRO Y CRÉDITO
GERENCIA



Somos más... más apoyo, más impulso!

Matriz (Sector El Carmen): Av. Benjamín Terán 319 y Antonia Vela Telf.: (03) 2802 850 / 2802 754

Agencia (Sector La Merced): Quijano y Ordóñez y Félix Valencia Tel.: (03) 2802 150 / 2800 544

Latacunga - Ecuador

www.coopandina.fin.ec

FICHA DE OBSERVACIÓN

FICHA N°	1
ELABORA:	Fernando Arias
LUGAR:	Departamento de Sistemas de Cooperativa ANDINA Ltda.
OBJETIVO	Recopilar información necesaria de la situación actual del proceso para el desarrollo de software
LO OBSERVADO	REGISTRO ETNOGRÁFICO
Recopilación y análisis de requisitos de usuario	<p>Cuando un colaborador requiere un producto de software, este se acerca al Departamento de Sistemas y mantiene una plática muy breve con el Administrador del Sistema. El colaborador da a conocer sus necesidades de una forma desordenada sin una lógica clara, propone y propone varias ideas que se le ocurre. Habla con una gran soltura en espera que sea clara la idea que quiere transmitir hacia el Administrador del Sistema. Por otro lado el compañero Administrador quien cumple diversas actividades dentro de sus funciones asignadas, simplemente no presta atención ya que se encuentra enfocado a otra tarea solicitada. Se observa detenidamente que el Administrador no logra entender lo que el colaborador necesita, es ahí que para “dar por terminado la plática” el Administrador es quien propone una solución planteada por él mismo y convence al colaborador. De las tantas ideas que propuso el colaborador solo anotó unas cuantas en una pequeña hoja de papel que estuvo cerca. El colaborador sale de la oficina mientras que el Administrador se dedica a la actividad que estuvo realizando. Otra forma de hacer llegar los requerimientos para desarrollar software es a través de correos electrónicos, en los cuales el colaborador expone de manera directa lo que necesita que el software realice. De la misma manera el administrador solo lo lee y no realiza ninguna acción inmediata. Dejándola para “cuando haya tiempo”.</p>
<u>Comentario:</u>	Se observó que no existe una comunicación fluida entre el colaborador y el administrador del sistema, ya que existe un distractor que bloquea la fluidez de la plática. El administrador no toma nota de manera detallada de lo que realmente requiere el colaborador. Es la primera y única vez que el usuario tiene algún tipo de contacto con el administrador.

Elaborado por: Fernando Arias

FICHA DE OBSERVACIÓN

FICHA N°	2
ELABORA:	Fernando Arias
LUGAR:	Departamento de Sistemas de Cooperativa ANDINA Ltda.
OBJETIVO	Recopilar información necesaria de la situación actual del proceso para el desarrollo de software
LO OBSERVADO	REGISTRO ETNOGRÁFICO
Proceso para el diseño de software	<p>Posterior a la “primera plática” entre el colaborador y el administrador del sistema. Y a la poca información recabada de las necesidades del colaborador, el administrador empieza a diseñar lo que podría ser la funcionalidad del producto de software solicitado. El diseño que plantea el administrador no se basado en el análisis del proceso que debe cumplir el software para obtener los resultados esperados, este diseño no es estructurado. No se contempla el uso de una herramienta para representar la arquitectura del software ni se aplica un proceso para definir los elementos que podrían involucrarse en el software. El diseño el administrador lo realiza de forma que todo conlleva a la maquetación del interfaz de usuario, invierte tanto tiempo en detallar la interfaz que pasa horas y horas cuadrando un formulario o añadiendo otros elementos que no son relevantes en el software solicitado.</p>
<u>Comentario:</u>	Se observó que el diseño se lo realiza en base a la interfaz de usuario más no a la arquitectura de los componentes que forman parte de él. De la misma manera el tiempo no lo planifica e invierte horas en detallar el interfaz.

Elaborado por: Fernando Arias

FICHA DE OBSERVACIÓN

FICHA N°	3
ELABORA:	Fernando Arias
LUGAR:	Departamento de Sistemas de Cooperativa ANDINA Ltda.
OBJETIVO	Recopilar información necesaria de la situación actual del proceso para el desarrollo de software
LO OBSERVADO	REGISTRO ETNOGRÁFICO
Pruebas del software	Las pruebas otro inconveniente para el administrador, se evidencia que mientras codifica también va haciendo las pruebas, esto es notorio cada vez que acierta en alguna funcionalidad se entusiasma. De nuevo no está presente el colaborador para validar el software.
<u>Comentario:</u>	En las pruebas no se aplica una planificación para realizarlas conjuntamente con el colaborador, desde la parte técnica no es una buena práctica codificar y ser juez de su propia creación.

Elaborado por: Fernando Arias

FICHA DE OBSERVACIÓN

FICHA N°	4
ELABORA:	Fernando Arias
LUGAR:	Departamento de Sistemas de Cooperativa ANDINA Ltda.
OBJETIVO	Recopilar información necesaria de la situación actual del proceso para el desarrollo de software
LO OBSERVADO	REGISTRO ETNOGRÁFICO
Implantación del software	Luego de haber llevado un proceso ambiguo para el desarrollo de software, viene la parte de subir a producción el software; para lo cual el administrador copia los nuevos archivos al servidor de aplicación, una vez puesto en producción éste realiza pruebas nuevamente para “validar”. La forma que el administrador valida el funcionamiento del nuevo software es cuando luego de unas horas en funcionamiento el software empieza a dar errores, y llega el colaborador con la queja al departamento de sistemas. El administrador ingresa a los archivos del sistema e identifica en donde está el error para corregirlo.
<u>Comentario:</u>	Se observó que el software implementado al subirlo a producción no es completamente funcional de tal manera que genera errores tarde o temprano y tendrá que corregirlo invirtiendo tiempo.

Elaborado por: Fernando Arias

**ENTREVISTA AL JEFE DEL DEPARTAMENTO DE SISTEMAS DE LA
COOPERATIVA DE AHORRO Y CRÉDITO ANDINA LTDA.**

El objetivo de la siguiente entrevista es recopilar la información necesaria para conocer la situación actual del proceso que se aplica para el desarrollo de software.

Nombre Entrevistado:	Ing. Fernando Pacheco
Profesión:	Ingeniero en Informática y Sistemas Computacionales
Cargo:	Administrador del Sistema
Fecha de la entrevista:	04 de Mayo del 2016
Entrevistador:	Sr. Fernando Arias

1. Que tiempo trabaja en el Departamento de Sistemas de Cooperativa Andina?

Administrador: Llevo trabajando 5 años en la Institución

2. Cuáles son sus principales funciones en el Departamento de Sistemas?

Administrador: Mis principales funciones es administrar lo concerniente al sistema financiero que manejamos, soportes, mantenimiento a nivel de hardware.

3. Dentro de sus funciones está el desarrollado Software?

Administrador: No está dentro de mis funciones tácitamente pero si lo realizó.

4. Ha aplicado alguna metodología para el desarrollo de software?

Administrador: En realidad dentro de la institución no se aplicado una metodología para el desarrollo de software, ya que el sistema ya está hecho y nosotros simplemente tenemos que tomar lo que esta y modificarlo.

5. Cómo es el proceso que aplica usted para el desarrollo de software?

Administrador:

Para el análisis?

Lamentablemente no tenemos manuales todavía del sistema nos toca revisar el código fuente, como fue desarrollado, como juegan las variables y de ahí empezamos a trabajar.

Para el diseño?

De igual manera nos debemos acoplar a lo que el sistema ya está hecho para nosotros diseñarlo en base a las ventanas, imágenes que el sistema ya lo tiene.

Para la implementación?

Casi no se ha hecho muchas implementaciones

Para las pruebas?

Son pruebas simples falla – error y validamos.

6. Cuáles son los problemas más críticos que se evidencian durante el desarrollo de software?

Administrador: El principal problema es que nosotros deberíamos tener la información completa del manejo de la base de datos, los procedimientos ya que en cuanto a front – end no se tiene mayor problema.

7. Cómo asegura que los productos de software en producción son completamente funcionales?

Administrador: lo comprobamos cuando los ponemos a trabajar, cuando damos fallo – error y vamos esos errores los vamos subsanando con el tiempo. Sabemos que esta funcional cuando los errores llegan a ser mínimos o cero.

8. De qué manera evidencia si se ha logrado cumplir con las expectativas del usuario que usa el software?

Administrador: Se evidencia cuando el usuario final está conforme, no tiene alguna sugerencia, cuando el usuario no le pone reparos al producto final.

9. Mantiene un registro documentado de los productos de software implementados?

Administrador: Actualmente se lo está registrando, pero vamos a cambiar la forma del registro, ya que lo hacemos de forma rustica.

10.Cuál es la inversión que realiza Cooperativa ANDINA para la adquisición de productos de software?

Administrador: El sistema Econx bordea los 25.000 - 30.000 es la única inversión que se ha hecho en cuanto a sistemas, es por eso que nos toca a nosotros continuar con ese trabajo.

11. Cuáles son las causas por las cuales no se adopta una metodología para el desarrollo de software en el Departamento de Sistemas?

Administrador: las causas principales es que no se ha sentido la necesidad de desarrollar cosas nuevas en la institución, en vista que somos una institución en crecimiento y básicamente la necesidad obliga, entonces en estos tiempos ya nos vemos obligados ya vemos la necesidad y vamos a empezar a trabajar.

12. Considera usted que mantener una comunicación constante y fluida con los colaboradores que solicitan un producto de software permite entender de mejor manera lo que ellos necesitan?

Administrador: Si por supuesto creo que va de la mano de la recolección de requisitos, de la ingeniería de software va de la mano el hecho que podamos conversar con la persona de que en los procesos mínimos entendamos cuales son los errores que nos van a dar o cuáles son esas particularidades que van hacer al sistema si se puede decir tambalear cuando ya esté funcionando.

13. Si se decidiera por aplicar una metodología cual sería? Y porqué

Administrador: por experiencia propia la metodología XP.

14. Según su punto de vista en qué aportaría a la Cooperativa Andina la adaptación del marco de trabajo de una metodología ágil para el desarrollo de software?

Administrador: Aportaría mucho, tenemos procesos los cuales deben ser desarrollados, los cuales a lo largo del tiempo nos van a demostrar como una metodología ágil se ajustan a estos procesos en su desarrollo.

Ing. Fernando Pacheco
Entrevistado

Sr. Fernando Arias
Entrevistado



DEPARTAMENTO DE SISTEMAS
DESARROLLO DE SOFTWARE
Historias de Usuario y Criterios de Aceptación

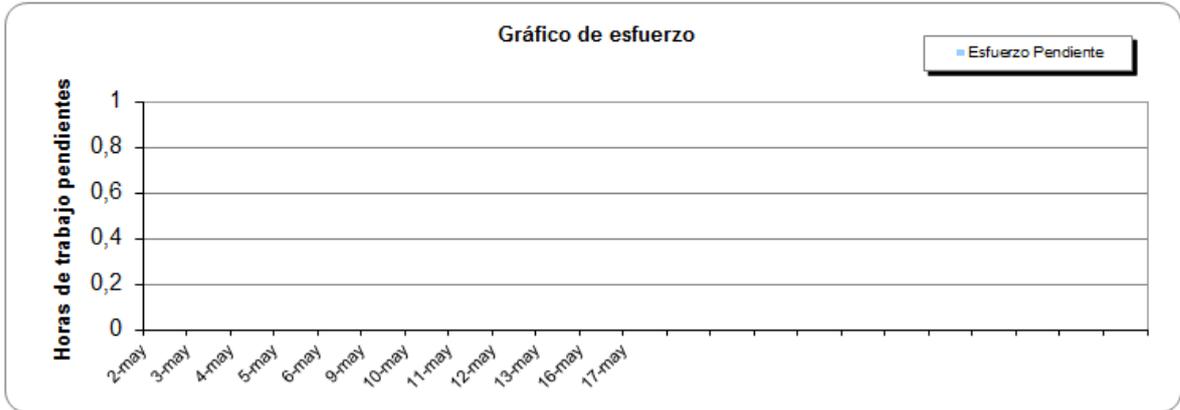
Proyecto:	
Area:	
Fecha:	
Versión:	
Elaborado por:	

Identificador (ID) de la Historia	Enunciado de la Historia				Criterios de Aceptación			
	Rol	Característica / Funcionalidad	Razón / Resultado	Número (#) de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento esperado

Elaborado por: Fernando Arias

	DEPARTAMENTO DE SISTEMAS DESARROLLO DE SOFTWARE <u>Pila de Producto (Product Backlog)</u>				N° Product Backlog		
					Proyecto:		
					Area:		
					Product Owner:		
					Fecha:		
					Versión:		
Identificador (ID) de la Historia	Enunciado de la Historia	Alias	Estado	Dimensión / Esfuerzo (días)	Iteración (Sprint)	Prioridad	Comentarios

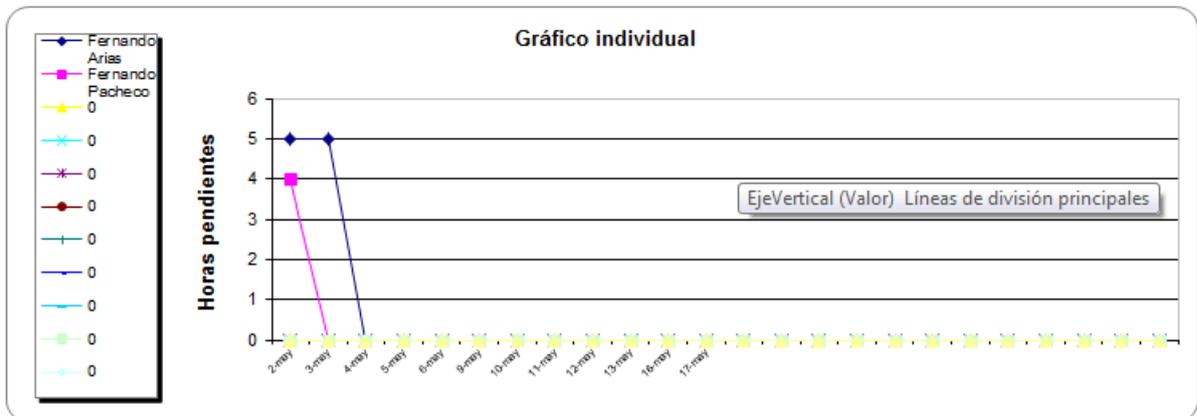
Elaborado por: Fernando Arias



Elaborado por: Fernando Arias



Elaborado por: Fernando Arias



Elaborado por: Fernando Arias

CONTROL DE VERSIONES					
<i>Versión:</i>	<i>Scrum Team por:</i>	<i>Revisada por:</i>	<i>Aprobada por:</i>	<i>Fecha:</i>	<i>Motivo:</i>

NOMBRE DEL PROYECTO	SIGLAS DEL PROYECTO
Cooperativa de Ahorro y Crédito ANDINA Ltda. Área:	

DECLARACIÓN DE LA ACEPTACIÓN FORMAL
Descripción:
Funcionalidades:
Fecha inicio – Fecha fin



ANDINA Ltda.
COOPERATIVA DE AHORRO Y CREDITO

COOPERATIVA DE AHORRO Y CRÉDITO ANDINA LTDA.

DEPARTAMENTO DE SISTEMAS

ACTA DE ENTREGA – RECEPCIÓN DE PRODUCTOS DE SOFTWARE

OBSERVACIONES ADICIONALES	
REVISADO Y VALIDADO POR (SCRUM MASTER)	
<i>NOMBRE Y CARGO DEL FUNCIONARIO</i>	<i>FECHA</i>
ACEPTADO Y APROBADO (PRODUCT OWNER)	
<i>NOMBRE Y CARGO DEL FUNCIONARIO</i>	<i>FECHA</i>

FIRMAS DE RESPONSABILIDAD		
PRODUCT OWNER		
SCRUM MASTER		
SCRUM TEAM		

Elaborado por: Fernando Arias



DEPARTAMENTO DE SISTEMAS
DESARROLLO DE SOFTWARE
Reunión de Retrospectiva

Proyecto:

Control de Denominaciones de Dinero

Identificador (ID) de la Reunión	Lugar de la Reunión	Fecha de la Reunión	Número de la Iteración / Sprint	Personas convocadas a la Reunión	Personas que asistieron a la Reunión	Duración de la Reunión	Reunión Retrospectiva		
							¿Qué salió bien en la iteración? (aciertos)	¿Qué no salió bien en la iteración? (errores)	¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)

Elaborado por: Fernando Arias



DEPARTAMENTO DE SISTEMAS
DESARROLLO DE SOFTWARE
Casos de Prueba

Proyecto:	
Area:	
Tester:	

Identificador (ID) Caso de Prueba	Fecha de Ejecución de Caso de Prueba	Nombre del Caso de Prueba	Precondición del Caso de Prueba	Descripción del Caso de Prueba	Respuesta Esperada de la Aplicación	Verificado		Resultados Esperados	Veredicto	Comentarios
						SI	NO			
						<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/> PASÓ	
						<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/> FALLÓ	
						<input type="checkbox"/>	<input type="checkbox"/>			
						<input type="checkbox"/>	<input type="checkbox"/>			

Elaborado por: Fernando Arias