



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN SISTEMAS E INFORMÁTICA**

**TEMA: ESTUDIO, SELECCIÓN Y APLICACIÓN DE UN MODELO  
DE DESARROLLO DE SOFTWARE Y SU ACOPLAMIENTO EN  
LA COOPERATIVA DE AHORRO Y CRÉDITO POLICÍA  
NACIONAL LTDA., DEMOSTRACIÓN DE SU VALIDEZ EN UN  
MÓDULO BÁSICO.**

**AUTORES: TIPÁN COYAGO, DARWIN MAURICIO.  
JARRÍN GUANOLUISA, MARCO VINICIO.**

**DIRECTOR: ING. GÓMEZ SALAZAR, MÓNICA ELIZABETH**

**SANGOLQUÍ**

**2017**



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, "**ESTUDIO, SELECCIÓN Y APLICACIÓN DE UN MODELO DE DESARROLLO DE SOFTWARE Y SU ACOPLAMIENTO EN LA COOPERATIVA DE AHORRO Y CRÉDITO POLICÍA NACIONAL LTDA., DEMOSTRACIÓN DE SU VALIDEZ EN UN MÓDULO BÁSICO.**" realizado por los señores **DARWIN MAURICIO TIPÁN COYAGO** y **MARCO VINICIO JARRÍN GUANOLUISA**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a los señores **DARWIN MAURICIO TIPÁN COYAGO** y **MARCO VINICIO JARRÍN GUANOLUISA** para que lo sustente públicamente.

**Sangolquí, 14 de noviembre del 2016**

Atentamente,

**ING. MÓNICA ELIZABETH GÓMEZ SALAZAR  
DIRECTORA**



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**AUTORÍA DE RESPONSABILIDAD**

Nosotros, **DARWIN MAURICIO TIPÁN COYAGO**, con cédula de identidad N° 1718093170, y **MARCO VINICIO JARRÍN GUANOLUISA**, con cédula de identidad N° 1718088477, declaramos que este trabajo de titulación "**ESTUDIO, SELECCIÓN Y APLICACIÓN DE UN MODELO DE DESARROLLO DE SOFTWARE Y SU ACOPLAMIENTO EN LA COOPERATIVA DE AHORRO Y CRÉDITO POLICÍA NACIONAL LTDA., DEMOSTRACIÓN DE SU VALIDEZ EN UN MÓDULO BÁSICO.**" ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaramos que este trabajo es de nuestra autoría, en virtud de ello nos declaramos responsables del contenido, veracidad y alcance de la investigación mencionada.

**Sangolquí, 14 de noviembre del 2016**

DARWIN MAURICIO  
TIPÁN COYAGO  
C.C. 1718093170

MARCO VINICIO  
JARRÍN GUANOLUISA  
C.C. 1718088477



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Nosotros, **DARWIN MAURICIO TIPÁN COYAGO, MARCO VINICIO JARRÍN GUANOLUISA**, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución la presente trabajo de titulación "ESTUDIO, SELECCIÓN Y APLICACIÓN DE UN MODELO DE DESARROLLO DE SOFTWARE Y SU ACOPLAMIENTO EN LA COOPERATIVA DE AHORRO Y CRÉDITO POLICÍA NACIONAL LTDA., DEMOSTRACIÓN DE SU VALIDEZ EN UN MÓDULO BÁSICO." cuyo contenido, ideas y criterios son de nuestra autoría y responsabilidad.

Sangolquí, 14 de noviembre del 2016

DARWIN MAURICIO  
TIPÁN COYAGO  
C.C. 1718093170

MARCO VINICIO  
JARRÍN GUANOLUISA  
C.C. 1718088477

## DEDICATORIA

A Dios.

Esta tesis se la dedico a mi Dios quién supo guiarme por el buen camino, por darme fuerzas para seguir adelante y no desmayar en los problemas que se presentaban, enseñándome a luchar en todo momento.

A mis padres Rosario y Luis.

Por su apoyo, consejos, comprensión, amor, ayuda en los momentos difíciles, y por trabajar duro para darme los recursos necesarios para estudiar. Me han dado todo lo que soy como persona, mis valores, principios, carácter, empeño, y el coraje para conseguir mis objetivos.

A mis hermanos Amy y Jefferson.

Por estar siempre presentes, brindándome su apoyo y acompañándome en todo momento para poderme realizar.

A mi esposa Paula.

Porque has sido, eres y serás mi compañera de toda la vida quien me ha dado todo su amor, respeto y por haberme apoyado en todos estos años que hemos estado juntos.

A mi hija Aidée.

Posiblemente en estos momentos no entiendas mis palabras, pero para cuando seas capaz de hacerlo, quiero que sepas de lo que significas para mí. Eres mi principal motivación en todo momento y como en todos mis logros en este has estado presente. Te amo hija mía.

**Darwin M. Tipán Coyago**

## **DEDICATORIA**

Dedico esta tesis a las dos personas más importantes en mi vida, mis amados padres Carlos Jarrín y Mariana Guanoluisa quienes con su apoyo y dedicación en mi formación han sido la fuerza que me impulsa a seguir adelante cumpliendo mis metas.

**Marco V. Jarrín G.**

## AGRADECIMIENTOS

El presente trabajo de tesis primeramente me gustaría agradecerle a ti Dios por bendecirme para llegar hasta donde he llegado, porque hiciste realidad este sueño anhelado.

A la UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE” por darme la oportunidad de estudiar y ser un profesional.

A la Cooperativa de Ahorro y Crédito Policía Nacional por haberme permitido realizar esta tesis.

A mi Jefe y compañeros de trabajo que me han brindado todo el apoyo necesario para que esto se pueda realizar.

A mi compañero de tesis Marco Jarrín, por todos los esfuerzos realizados para lograr este objetivo.

Por último, a mi directora de tesis quién nos ayudó en todo momento, Msc. Mónica Gómez.

*Son muchas las personas que han formado parte de mi camino a las que me encantaría agradecerles su amistad, consejos, apoyo, ánimo y compañía en los momentos más difíciles de mi vida. Quiero darles las gracias por formar parte de mí, y por todo lo que me han ayudado.*

**Darwin M. Tipán Coyago**

## **AGRADECIMIENTOS**

Quiero agradecer a Dios por permitirme haber alcanzado una meta más y cada día guiar mi camino.

A mi Padre que con sus consejos y apoyo incondicional siempre ha permanecido a mi lado.

A mi Madre por siempre estar pendiente de mi bienestar.

A mi hermano que me ha demostrado que la distancia no es un impedimento para estar comunicado conmigo y ofrecerme sus palabras de aliento.

A mis profesores de la UNIVERSIDAD DE LAS FUERZAS ARMADAS "ESPE" por darme la formación académica.

A Darwin por confiar en mí para la elaboración de esta tesis.

A mis amigos que han estado en todo momento brindándome su apoyo incondicional.

Un profundo agradecimiento especial a la mí directora de tesis Msc. Mónica Gómez por guiarnos en la realización de este proyecto.

**Marco V. Jarrín G.**

## ÍNDICE DE CONTENIDO

### Contenido

<b>DEDICATORIA</b> .....	iv
<b>DEDICATORIA</b> .....	v
<b>AGRADECIMIENTOS</b> .....	vi
<b>AGRADECIMIENTOS</b> .....	vii
<b>ÍNDICE DE CONTENIDO</b> .....	viii
<b>RESUMEN</b> .....	xiii
<b>CAPÍTULO I. INTRODUCCIÓN</b> .....	1
1.1. INTRODUCCIÓN.....	1
1.2. ANTECEDENTES.....	2
1.3. PLANTEAMIENTO DEL PROBLEMA.....	3
1.4. JUSTIFICACIÓN E IMPORTANCIA.....	4
1.5. OBJETIVOS.....	6
1.6. ALCANCE.....	7
1.7. METODOLOGÍA.....	8
1.8. HERRAMIENTAS.....	8
<b>CAPÍTULO II. ESTUDIO DE LA EMPRESA</b> .....	9
2.1. DESCRIPCIÓN DE LA EMPRESA.....	9
2.1.1. ORIGEN.....	9
2.1.2. MISIÓN.....	10
2.1.3. VISIÓN.....	10
2.1.4. OBJETIVO SOCIAL.....	10
2.1.5. VALORES.....	10
2.1.6. PRODUCTOS Y SERVICIOS.....	11
2.1.7. ESTRUCTURA ORGANIZACIONAL.....	12
2.1.8. MAPA ESTRATÉGICO.....	12
2.2. DESCRIPCIÓN DE LA JEFATURA DE SISTEMAS.....	15
2.2.1. SITUACIÓN ACTUAL.....	15
2.2.2. OBJETIVO GENERAL.....	16
2.2.3. OBJETIVOS ESPECÍFICOS.....	17

2.2.4.	DIAGRAMA DE FLUJO DESARROLLO DE APLICACIONES .....	17
2.2.5.	ESTRUCTURA FUNCIONAL .....	18
2.2.6.	SERVICIOS .....	20
2.3.	ANÁLISIS.....	24
	<b>CAPÍTULO III. ESTADO DEL ARTE .....</b>	<b>25</b>
3.1.	METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE .....	25
3.1.1.	CONCEPTO .....	25
3.1.2.	HISTORIA .....	25
3.1.3.	MANIFIESTO ÁGIL .....	26
3.1.4.	CICLO DE VIDA.....	27
3.1.5.	TRABAJOS RELACIONADOS .....	28
3.2.	XP (PROGRAMACIÓN EXTREMA).....	31
3.2.1.	ORIGEN .....	31
3.2.2.	ENTREGABLES .....	32
3.2.3.	ROLES .....	35
3.2.4.	PROCESO XP.....	37
3.3.	SCRUM (RUGBY (MELÉ)) .....	40
3.3.1.	ORIGEN .....	40
3.3.2.	INTRODUCCIÓN AL MODELO .....	41
3.3.3.	SCRUM TÉCNICO .....	42
3.3.4.	REGLAS DE SCRUM.....	49
3.3.5.	PROCESO SCRUM.....	50
3.4.	DSDM (DYNAMIC SYSTEMS DEVELOPMENT METHOD).....	51
3.4.1.	INTRODUCCIÓN AL MODELO .....	51
3.4.2.	ROLES .....	51
3.4.3.	ENTREGABLES .....	53
3.4.4.	PROCESO DSDM.....	55
3.5.	FDD (FEATURE DRIVEN DEVELOPMENT) .....	58
3.5.1.	ORIGEN .....	58
3.5.2.	INTRODUCCIÓN AL MODELO .....	58
3.5.3.	ROLES .....	59
3.5.4.	PROCESOS .....	61
3.5.5.	REGLAS DE FDD.....	64

3.5.6. PROCESO FDD.....	66
3.6. ASD (Adaptive Software Development).....	71
3.6.1. ORIGEN .....	71
3.6.2. DEFINICIÓN.....	71
3.6.3. ENTREGABLES .....	72
3.6.4. ROLES .....	73
3.6.5. FASES.....	74
3.6.6. PROCESO DE ASD .....	78
3.7. DEFINICIÓN DE CARACTERÍSTICAS DE EVALUACIÓN.....	80
3.7.1. ANÁLISIS .....	84
3.7.2. CUADRO COMPARATIVO .....	84
3.7.3. VENTAJAS Y DESVENTAJAS.....	95
3.7.4. SELECCIÓN DE LA METODOLOGÍA QUE SE ACOPLA A LA CPN. ..	97
<b>CAPÍTULO IV. EJECUCIÓN DEL ESTUDIO.....</b>	<b>100</b>
4.1. DISEÑO DEL PROTOTIPO.....	100
4.2. DESARROLLO DEL PROTOTIPO.....	102
4.2.1. REQUISITOS DE SOFTWARE .....	102
4.2.2. DISEÑO.....	102
4.2.3. CONSTRUCCIÓN .....	106
4.2.4. VERIFICACIÓN Y VALIDACIÓN .....	114
4.2.5. DESPLIEGUE. ....	114
4.2.6. ANÁLISIS DE RESULTADOS.....	118
5.1. CONCLUSIONES.....	124
5.2. RECOMENDACIONES .....	124
5.3. LÍNEAS FUTURAS. ....	125
GLOSARIO DE TÉRMINOS.....	126
BIBLIOGRAFÍA.....	128

## Índice de Tablas.

Tabla 1 Proceso de XP. ....	37
Tabla 2 Proceso de Scrum. ....	50
Tabla 3 Proceso DSDM.....	55
Tabla 4 Proceso FDD. ....	66
Tabla 5 Proceso de ASD.....	78
Tabla 6 Indicadores de evaluación.....	83
Tabla 7 Cuadro comparativo de las metodologías de desarrollo de estudio.....	84
Tabla 8 Ventajas y Desventajas de las metodologías de desarrollo. ....	95
Tabla 9 Indicadores de desempeño análisis de resultados. ....	118
Tabla 10 Matriz análisis de resultados. ....	119

## Índice de Figuras.

Figura 1. Organigrama Estructural CPN.....	13
Figura 2. Mapa Estratégico CPN. ....	14
Figura 3: Número de agencias CPN. ....	16
Figura 4. Proceso para el Desarrollo de Aplicaciones. ....	18
Figura 5. Estructura Funcional del área de Sistemas. ....	19
Figura 6 Número de personas en el área de sistemas. ....	19
Figura 7 Número de personas en la CPN.....	20
Figura 8 Número de agencias de la CPN.....	20
Figura 9: Diagrama General del Proceso Ágil. ....	28
Figura 10: Ciclo de vida de eXtreme Programming.....	32
Figura 11: Historia de Usuario.....	33
Figura 12: Proceso Scrum. ....	42
Figura 13: Marco Scrum Técnico.....	49
Figura 14: Diagrama de procesos DSDM. ....	51
Figura 15: Proceso de la Metodología FDD.....	62
Figura 16: Iteraciones Design and Build by Feature. ....	63
Figura 17: Reglas de FDD. ....	65
Figura 18: Ciclo de Vida FDD.....	76
Figura 19 Diseño del experimento.....	101
Figura 20 Diagrama de Base de Datos. ....	103
Figura 21 Diagrama de Secuencia Servicio Activo y Servicio Pasivo. ....	104
Figura 22 Diagrama de secuencia Familiares y Empleados Civiles. ....	105
Figura 23 Pantalla Prototipo Login MIS DÉCIMOS. ....	106
Figura 24 Pantalla Prototipo Tipo de Socio MIS DÉCIMOS. ....	106
Figura 25 Pantalla Ingreso Acumulación Décimos (Activos y Pasivos). ....	107
Figura 26 Pantalla Ingreso Acumulación Décimos (Familiares y Civiles).....	107

Figura 27 Arquitectura Módulo Mis Décimos.....	109
Figura 28 Arquitectura Módulo Mis Décimos Diagrama de Despliegue.....	109
Figura 29 Pantalla Login MIS DÉCIMOS.....	110
Figura 30 Pantalla Tipo de Socio MIS DÉCIMOS.....	110
Figura 31 Pantalla Ingreso Acumulación Décimos (Activos y Pasivos).....	111
Figura 32 Pantalla Ingreso Acumulación Décimos (Familiares y Civiles).....	111
Figura 33 Contrato de Acumulación Décimos pág. 1/2.....	112
Figura 34 Contrato de Acumulación Décimos pág. 2/2.....	112
Figura 35 Sistema Reportes (Listado Solicitudes Décimos).....	113
Figura 36 Listado Solicitudes Décimos (Exportado del Sistema de Reportes).....	113
Figura 37 Se publica la solución desde Microsoft Visual Studio.....	114
Figura 38 Se publica la solución en el servidor de aplicaciones.....	115
Figura 39 Seleccionar el IIS.....	115
Figura 40 Selección de la aplicación en el IIS.....	116
Figura 41 Direccionamiento del sitio a la ubicación del módulo.....	116
Figura 42 Selección del módulo publicado.....	117
Figura 43 Resultado entre proyectos.....	120
Figura 44 Resultados del análisis.....	121
Figura 45 Resultados requisitos.....	121
Figura 46 Resultado Entregables.....	122
Figura 47 Resultado Pruebas.....	122
Figura 48 Resultado Tiempo.....	123

## RESUMEN

La Cooperativa de Ahorro y Crédito Policía Nacional Ltda., se encuentra ubicada en la ciudad de Quito, entidad financiera a la que se realizó la selección y aplicación de un modelo de desarrollo de software; día a día esta entidad se enfrenta al problema de realizar el desarrollo de aplicaciones, ya que no posee una metodología de desarrollo de software para la construcción de las mismas. El análisis y selección de un modelo de desarrollo software se basa en el estudio de cinco metodologías ágiles de desarrollo (XP, SCRUM, DSDM, FDD, ASD) y en indicadores de evaluación que la Cooperativa considera necesario para la construcción. Tras el análisis de diferentes metodologías de software, se seleccionó la más adecuada (SCRUM) para coadyuvar al manejo de sus aplicaciones. Se realizó un análisis cualitativo de las cinco metodologías mencionadas anteriormente con la finalidad de obtener un modelo de selección y una metodología basada en resultados. Se concluye que no existe una metodología universal para llevar a cabo con éxito todos los proyectos de desarrollo de software. Se comprobó que en la construcción de una aplicación pequeña (MIS DÉCIMOS) utilizando SCRUM se generó mejor satisfacción al cliente ahorrando tiempo, mejorando la motivación e implicación del equipo de desarrollo. Se evidenció que en el desarrollo de software con una metodología se tiene un proceso definido para poder cumplir con lo que el cliente solicita, por lo tanto, toda empresa que desarrolle aplicaciones debe tener establecida una metodología de desarrollo software.

## PALABRAS CLAVES

- **METODOLOGÍA DE DESARROLLO DE SOFTWARE.**
- **METODOLOGÍAS AGILES.**
- **CICLO DE VIDA.**
- **SCRUM.**

## ABSTRACT

Cooperativa de Ahorro y Crédito Policía Nacional Ltda., is located in Quito, is a financial institution which was the selection and application of a model of software development; Day to day this entity faces the problem of performing the application development, since it does not have a software development methodology for the construction of the same. The analysis and selection of a software development model is based on the study of five agile development methodologies (XP, SCRUM, DSDM, FDD, ASD) and on evaluation indicators that the Cooperative considers necessary for construction. After the analysis of different software methodologies, the most appropriate SCRUM was selected to assist in the management of their applications. A qualitative analysis of the five methodologies mentioned above was carried out in order to obtain a selection model and a results-based methodology. It is concluded that there is no universal methodology to successfully carry out all software development projects. It was verified that in the construction of a small application (MIS TIMES) using SCRUM generated better customer satisfaction by saving time, improving the motivation and involvement of the development team. It was evidenced that in software development with a methodology has a defined process to be able to comply with what the client requests, therefore, any company that develops applications must have established a software development methodology.

## KEYWORDS

- **SOFTWARE DEVELOPMENT METHODOLOGY.**
- **AGILE METHODOLOGIES.**
- **LIFECYCLE.**
- **SCRUM.**

## **CAPÍTULO I. INTRODUCCIÓN**

### **1.1. INTRODUCCIÓN**

El tema que se desarrollará como tesis está enfocado principalmente en cubrir una necesidad fundamental de varias empresas pequeñas que desarrollan software a la medida, que además omiten el considerar el alto nivel de importancia de una correcta documentación o alguna metodología de desarrollo apropiada para la creación del producto de software.

Este es el caso específico de la Cooperativa de Ahorro y Crédito Policía Nacional en adelante llamada CPN y de su departamento de Sistemas, la misma que dentro de su infraestructura (Help Desk, Gestión de Seguridades, Gestión de Base de Datos y Desarrollo de Software) necesita adoptar una metodología de desarrollo de software que permita:

- Crear procesos, con su respectiva documentación.
- Contar con una base para posteriores desarrollos.
- Evitar fracaso de proyectos.
- Aprovechar los recursos (tiempo y dinero).

Para cumplir con estos puntos se realizará una investigación de cinco metodologías de desarrollo de software ágiles y luego se escogerá la más apropiada y flexible acoplándola al departamento de sistemas de la CPN, específicamente al área de desarrollo y a sus requerimientos.

En particular, las metodologías ágiles mejoran la satisfacción del cliente dado que se lo involucra y compromete a lo largo de todo el proyecto, puesto que en cada etapa del desarrollo se informará de los avances, con el fin de que éste aporte con su experiencia para optimizar las características del producto final. De esta manera se pueden evitar malentendidos ya que el

cliente siempre tendrá una visión completa del estado del producto. (Martinez Fustero Eduardo, 2014).

Por lo tanto, se pretende aportar a la CPN con una metodología de desarrollo de software adaptada a la empresa en mención, pudiéndose expandir el trabajo a empresas típicas de nuestro entorno.

## **1.2. ANTECEDENTES**

La CPN, nace jurídicamente el 29 de junio de 1976. A partir del año 2002 la institución ha pasado por estrictos procesos de cambio mediante los cuales ha logrado afianzarse entre las primeras cooperativas de ahorro y crédito a nivel nacional. La cooperativa se ha basado fundamentalmente en 3 factores: personal, procesos y tecnología.

En referencia a la tecnología destacamos que es fundamental en el crecimiento de la entidad. Hoy en día la CPN está irrumpiendo en grandes proyectos que sobrelleven un alto impacto en la vida de los socios. Entre los más significativos están:

- La adquisición de la red de cajeros automáticos propios.
- La implementación del Software de Internet Banking, mediante el cual los socios tienen la facilidad de consultar su saldo, realizar transferencias bancarias, pago de servicios entre otros.

En el departamento de Sistemas de la CPN se desarrollan proyectos de software los cuáles son realizados de manera informal.

Para establecer la presente propuesta de tesis, se acordó entre el Gerente de la empresa en cuestión y el grupo de proyecto de tesis realizar una solución informática que ayude al adecuado análisis, procesamiento, implementación y

pruebas de cada software que se desarrolle y será medible mediante el estudio de los KPI<sup>1</sup>, para la CPN.

Con respecto a la presente investigación, el Departamento de Sistemas de la CPN documentará y controlará cada fase del proyecto, interviniendo y extendiendo el soporte en todos los ámbitos de acción y a todos los participantes.

### **1.3. PLANTEAMIENTO DEL PROBLEMA**

Debido a que anteriormente la CPN no estaba regulada bajo ningún ente de control, el Departamento de Sistemas específicamente el área de desarrollo se pasaba solo apagando incendios, se llegó al punto de tener no menos de 15 proyectos por año y esto unido al crecimiento acelerado de la Cooperativa ocasionaba grandes problemas en la creación de programas software.

Actualmente se realiza reuniones con el usuario, se toma apuntes del programa o reporte que se necesita que sea desarrollado, se elabora un documento RFD<sup>2</sup> y se comienza con la construcción del programa. Esto ocasiona problemas en el desarrollo ya que no existe un debido proceso en el cual los técnicos se puedan basar para la elaboración del mismo.

Entre los problemas más comunes tenemos:

- Ideas mal planteadas.
- Requisitos innecesarios.
- Cambios de un día para otro en la estructura de los programas o reportes.

---

<sup>1</sup> KPI: Indicador clave de desempeño.

<sup>2</sup> RFD: Requerimientos para el desarrollo.

- Incumplimiento en las fechas de entrega.

Por consiguiente, la CPN carece de una metodología de desarrollo de software que permita la gestión adecuada del proceso de elaboración de productos de software; este aspecto es fundamental, puesto que la carencia del mismo no permite llevar un debido control del transcurso de desarrollo de sistemas de una forma organizada y documentada; teniendo un alto riesgo en la implementación de los programas (sistemas) en la organización y en el ciclo de vida de los mismos.

#### **1.4. JUSTIFICACIÓN E IMPORTANCIA**

El tema que vamos a desarrollar es muy importante para resolver los problemas que la CPN enfrenta, puesto que ayudaremos a la mitigación de riesgos tales como la pérdida de tiempo y dinero al no manejarse bajo un mismo estándar para la adquisición y/o creación de un nuevo software dependiendo de la necesidad de la institución.

La técnica que se seleccionará será la adopción y acoplamiento de una metodología ágil, ya que ésta basada en un proceso incremental de pequeñas fases o ciclos rápidos que muestran al cliente resultados en un tiempo mínimo, además mantiene una estrecha cooperación entre los involucrados y el desarrollador por lo que el trabajo se torna ágil, sencillo y conciso puesto que si surgen dudas estas serán examinadas el momento de la creación con el fin de no generar retrasos en ningún ciclo.

La documentación también es parte fundamental y de acuerdo a la propuesta presentada, se la irá generando acorde a los ciclos de desarrollo presentados, permitiendo de esta manera hacer cambios de última hora ya que la adaptabilidad es otra ventajosa característica de las metodologías ágiles.

Habitualmente las empresas pequeñas, no han sido concientizadas de conflictos inherentes como los ya mencionados al momento de procesar información mediante ordenadores, la responsabilidad recae muchas veces en el equipo de desarrollo de software, que no han sabido explicar con la suficiente claridad la definición de artefactos de software a ser entregados en cada fase.

Por lo tanto, la metodología que se seleccionará y acoplará en la CPN evitará problemas como:

- Retrasos en la planificación: llegada la fecha de entregar el software éste no está disponible.
- Sistemas deteriorados: el software se ha creado pero después de un par de años el coste de su mantenimiento es tan complicado que definitivamente se abandona su producción.
- Tasa de defectos: el software se pone en producción, pero los defectos son tantos que nadie lo usa.
- Requisitos mal comprendidos: el software no resuelve los requisitos planificados inicialmente.
- Cambios de negocio: el problema que resolvía el software ha cambiado y el software no se ha adaptado.

### **Relevancia Social**

Ayudar a la empresa, empleados y personas relacionadas a la CPN a mejorar su desempeño, con un estímulo de eficacia y mejor gestión de la empresa.

### **Importancia Tecnológica**

El proyecto se presenta como solución tecnológica que permitirá facilitar el trabajo de creación y documentación de productos de software de la empresa teniendo como referencia una metodología ágil apropiada. De esta manera se obtiene una ayuda tecnológica de la que hoy carecen, como instrumento organizado. Además, se complementará el estudio poniendo en práctica el desarrollo de un módulo básico, demostrando así la validez del proyecto.

### **Importancia Económica**

Al analizar la información de los KPIs, se mejora considerablemente la toma de decisiones en cuanto a la creación de nuevo software y con ello se incurrirá en menores gastos, mejor administración de los recursos y generación de un mayor porcentaje de proyectos concluidos satisfactoriamente.

### **Beneficiarios**

- Directos
  - ✓ Cooperativa de Ahorro y Crédito Policía Nacional.
  - ✓ Departamento de Sistemas de la Cooperativa (Área de desarrollo).
- Indirectos
  - ✓ Pequeñas empresas con necesidades similares.
  - ✓ Sociedad en General.

## **1.5. OBJETIVOS**

### **1. OBJETIVO GENERAL**

Estudiar y seleccionar una metodología ágil de desarrollo de software realizando un análisis de entre varias técnicas de creación de software mediante cuadros de comparación entre metodologías para poder acoplar la que más se ajuste a la necesidad del Departamento de Sistemas de la CPN y a su actual levantamiento de procesos.

## **2. OBJETIVOS ESPECÍFICOS**

1. Diagnosticar la situación actual del desarrollo de software en la Cooperativa con el personal involucrado mediante una matriz FODA.
2. Investigar cinco metodologías de desarrollo de software ágiles mediante el uso de libros e internet para conocer la forma en que se desarrollan y seleccionar la que de mejor manera adapte las necesidades de la empresa.
3. Implementar la metodología propuesta y aplicarla en un prototipo con fines demostrativos.
4. Realizar un análisis de resultados basado en indicadores de desempeño (ISO/IEC 9126).

### **1.6. ALCANCE**

El presente proyecto de tesis pretende la selección e implementación de una metodología de desarrollo de software ágil que se acople a la CPN. Por lo cual se realizará:

- Análisis actual de la empresa; donde se determinará como se está manejando el desarrollo de software en la empresa a través de un análisis con la matriz FODA.
- Una propuesta de una metodología ágil acoplada a la CPN.
- Implementación de la metodología seleccionada en un pequeño módulo.
- El análisis de los resultados obtenidos basados en normas de evaluación del software (ISO/IEC 9126).

## 1.7. METODOLOGÍA

El método a utilizar es el siguiente:

**Método Revisión de Literatura:** Uno de los pasos previos antes de empezar a trabajar en un proyecto de investigación consiste en analizar la literatura existente en el área sobre el objeto de estudio. De esta forma el investigador trata de situarse al mismo nivel de conocimiento que el de los investigadores que lo precedieron en el área. La revisión de la literatura se da en todas partes del proceso investigador, pero es de vital importancia en: Planteamiento, Diseño, Construcción del instrumento de recogida de datos, Muestreo, y Escritura del trabajo. (Jack Herrick, 2015)

## 1.8. HERRAMIENTAS

### 1. SOFTWARE

- Windows 7
- Microsoft Office 2010
- Microsoft Project 2010

### 2. HARDWARE

- Computador 2.00 GHz, 3.00 GB RAM
- Impresora
- Dispositivos de almacenamiento

## **CAPÍTULO II. ESTUDIO DE LA EMPRESA**

### **2.1. DESCRIPCIÓN DE LA EMPRESA**

#### **2.1.1. ORIGEN**

La Cooperativa de Ahorro y Crédito Policía Nacional Ltda., nace jurídicamente el 29 de junio de 1976. Unidos por el propósito de ayudar económicamente a sus compañeros policías a través de los beneficios brindados por una entidad cooperativista; por lo que se sustentaron en la filosofía de ayuda mutua y solidaria, encontrando así la respuesta a las necesidades de crecimiento dentro de la Policía Nacional.

Es así como la CPN nace y empieza sus actividades estableciéndose por varios años en el Primer Distrito ubicado en las calles Cuenca y Mideros en la ciudad de Quito con su primera oficina. Después de 15 años de trabajo incansable y gracias a la ayuda de los miembros que conformaban la institución y principalmente la Comandancia General de la Policía Nacional, la CPN logra tener sus propias oficinas actualmente ubicadas en Ecuador, en el cantón Quito, en las calles Av. 10 de agosto N31-218 y Mariana de Jesús, Edificio Matriz.

Debido a su gran índice de aceptación la CPN ha logrado crecer a nivel nacional con la apertura de nuevas agencias generando así mayor cobertura para sus socios.

A partir del año 2002 dicha entidad ha pasado por procesos estrictos de cambio con los cuales ha logrado consolidarse entre las primeras cooperativas de ahorro y crédito a nivel nacional. (Cooperativa Policía Nacional, 2015)

### **2.1.2. MISIÓN**

“Fomentamos el desarrollo económico y social de nuestros socios y clientes, brindando productos financieros innovadores, ágiles, seguros y oportunos para mejorar su calidad de vida.” (Cooperativa Policia Nacional, 2015)

### **2.1.3. VISIÓN**

“En el **2017** seremos una cooperativa modelo de excelencia en el servicio a sus asociados y clientes, con talento humano altamente capacitado y comprometido, con procesos estandarizados, operaciones a nivel nacional e internacional, tecnología de punta y socialmente responsables.” (Cooperativa Policia Nacional, 2015)

### **2.1.4. OBJETIVO SOCIAL**

La Cooperativa tendrá como objeto social principal **La Intermediación Financiera y de Responsabilidad Social con sus Socios.**

La cooperativa de ahorro y crédito es una entidad cerrada, lo que significa que se ceñirá a las disposiciones diferenciadas que la Superintendencia de Economía Popular y Solidaria (SEPS) emitirá para aquellas. (Cooperativa Policia Nacional, 2013)

### **2.1.5. VALORES**

Las cooperativas se basan en los valores de ayuda mutua, responsabilidad, democracia, igualdad, equidad y solidaridad, siguiendo la tradición de sus fundadores y miembros. (Cooperativa Policia Nacional, 2015)

### **2.1.6. PRODUCTOS Y SERVICIOS**

En la actualidad la CPN cuenta con los siguientes productos:

- Crédito Hipotecario
- Microcrédito
- Plan Mi Futuro
- Depósito a Plazo Fijo
- Acreditación de Sueldos
- Anticipo de Sueldos
- Crédito Ordinario
- Apertura de Cuentas
- Tarjeta Master Débit CPN
- Ayuda Mutua
- Crédito Emergente
- Transferencia de Efectivo
- Red de Pagos CPN
- Crédito Automotriz
- Planes Telefónicos Movistar
- Cuenta ChikiAhorros

La CPN cuenta entre otros con los siguientes servicios:

- Asistencia Médica
- Servicio Mortuorio
- Útiles Escolares
- Recaudaciones a
  - ✓ Corporación Nacional Telefónica CNT
  - ✓ Empresa Municipal de Agua Potable EMAAP
  - ✓ Empresa Eléctrica Quito EEQ
  - ✓ MasterCard
  - ✓ Visa Cash
  - ✓ Planes Claro

- ✓ Planes Movistar
- ✓ UNIVISA S.A
- ✓ INTERAGUA
- ✓ TV Cable Quito
- ✓ Agencia Nacional de Transito. (Cooperativa Policía Nacional, 2015)

### **2.1.7. ESTRUCTURA ORGANIZACIONAL**

En la Figura 1 podemos observar la estructura organizacional de la CPN, en la que resalta la Jefatura de Sistemas, la misma que está en el nivel operacional.

### **2.1.8. MAPA ESTRATÉGICO**

En la Figura 2 se observa el mapa estratégico de la CPN, dónde se identifica en el plan de procesos el objetivo del área de sistemas, el cuál es implementar metodologías y tecnologías para mejorar controles.

## ORGANIGRAMA ESTRUCTURAL COOPERATIVA POLICÍA NACIONAL

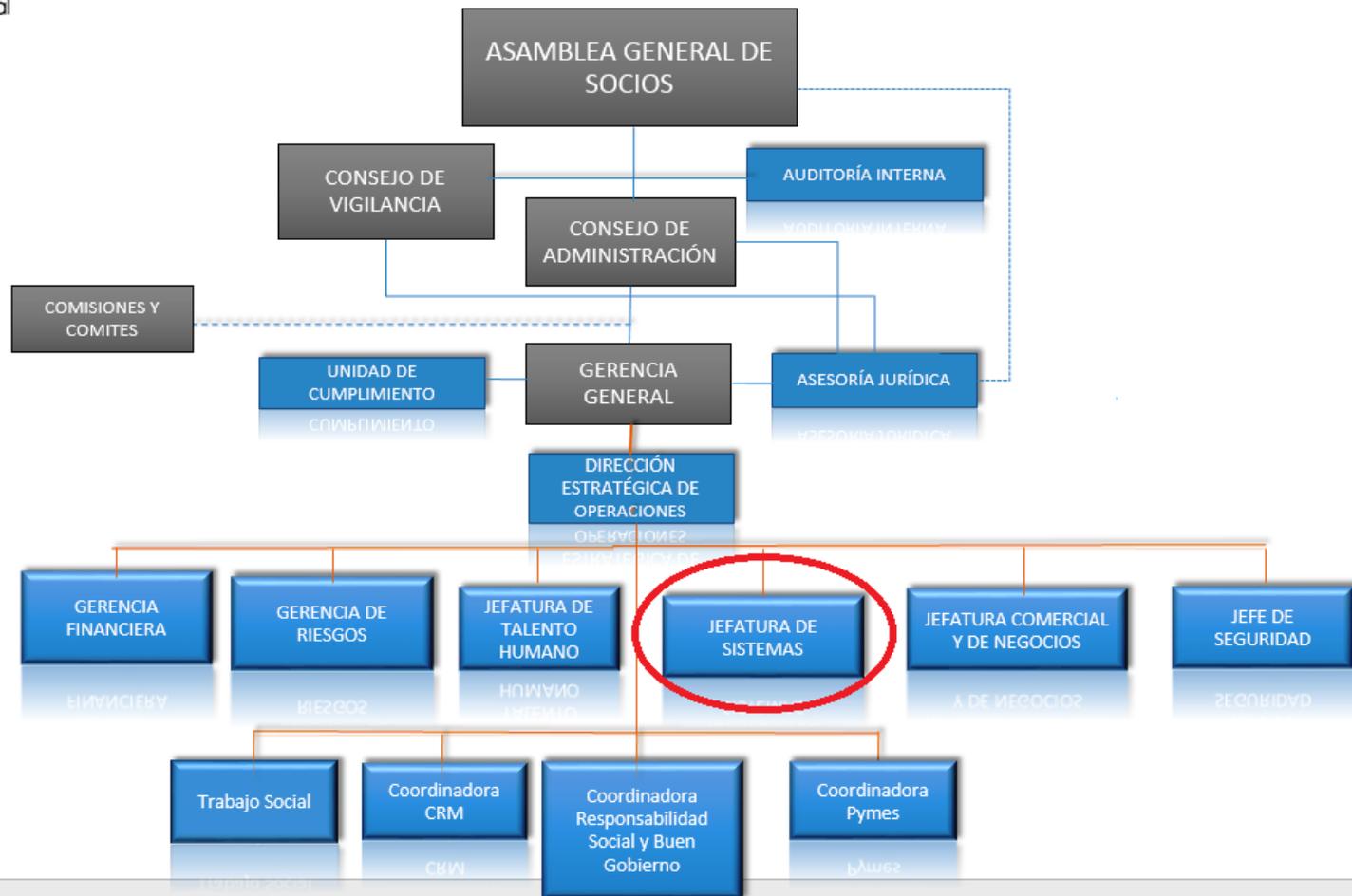


Figura 1. Organigrama Estructural CPN.

Fuente: (Cooperativa Policia Nacional, 2015)

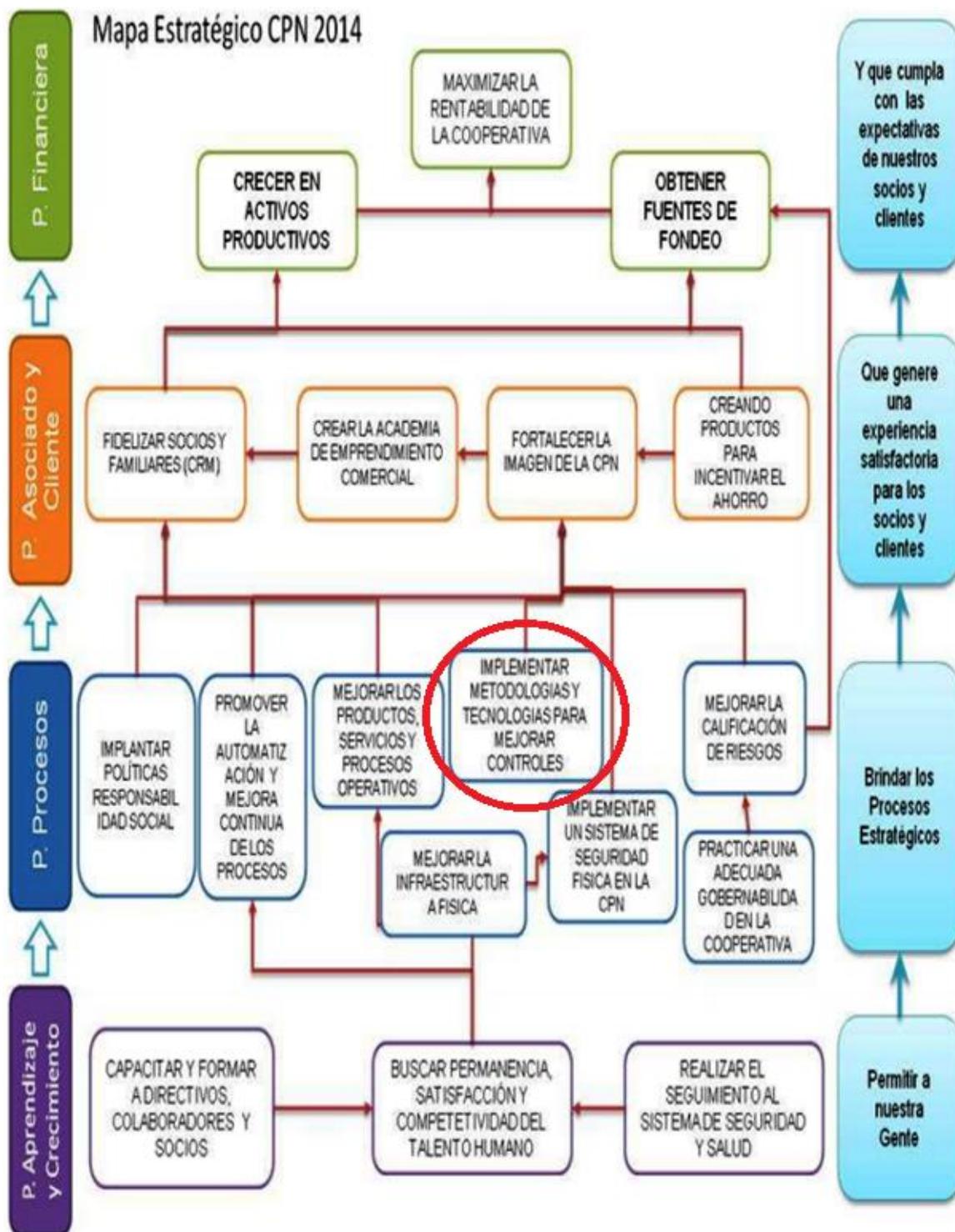


Figura 2. Mapa Estratégico CPN.

Fuente: (Cooperativa Policía Nacional, 2015)

## **2.2. DESCRIPCIÓN DE LA JEFATURA DE SISTEMAS**

### **2.2.1. SITUACIÓN ACTUAL**

La Jefatura de Sistemas es un área en donde se prestan servicios relacionados al soporte y mantenimiento de Sistemas de Información adquiridos y al desarrollo, mantenimiento y soporte de aplicaciones para la CPN; así como también gestionar las seguridades relacionadas al manejo y uso de la información.

El área de Sistemas da soporte y mantenimiento a los siguientes sistemas: Cobis, Nómina, Inventarios, Activos Fijaos, Sistema de Pagos Interbancarios, Cooperativa Virtual, SoftExpert, SMS, Huella Digital, Puntomático, Image Folder, CRM, Extreme Web, Anexo Transaccional en la matriz y sus veinte y ocho agencias como se evidencia en la Figura 3.

En la matriz se ha desarrollado algunas aplicaciones como: Inventarios, Activos Fijos, Modulo de Reportes, Sistema de Reportes Web replicando estas a las veinte y seis agencias mencionadas.

La CPN considera al área de sistemas dentro de la planificación estratégica del 2014, en procesos como el ente que podrá implementar metodologías y tecnologías para mejorar controles. (Figura 2).

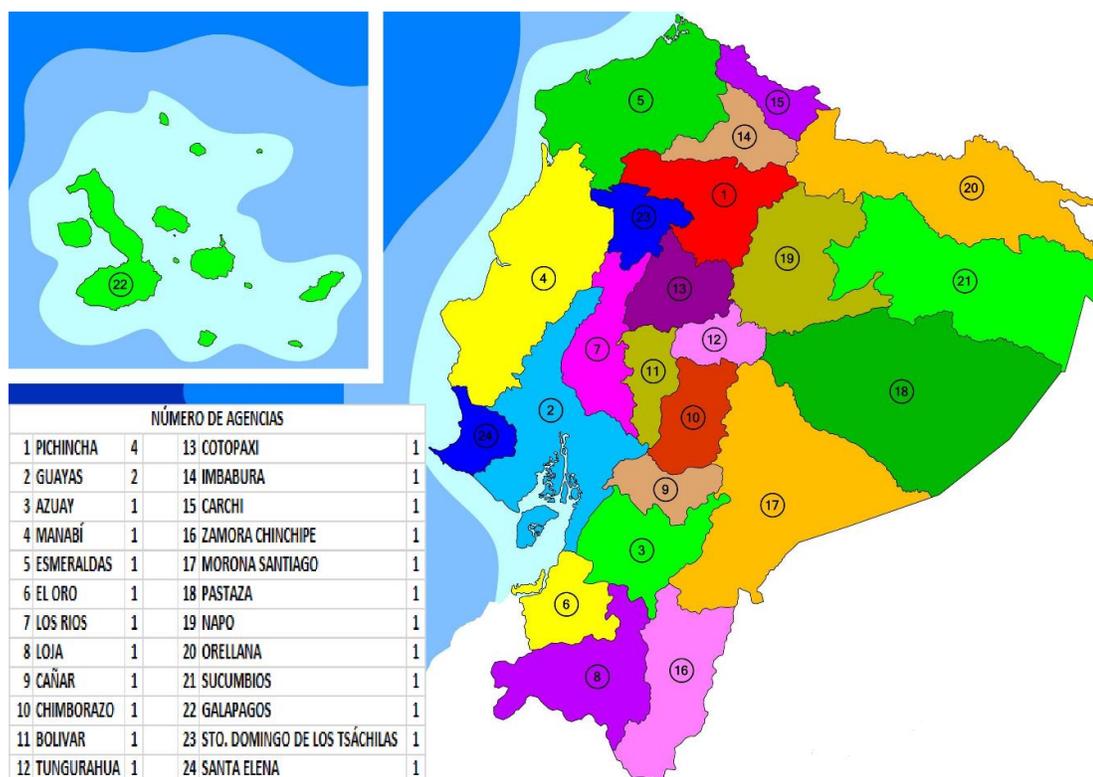


Figura 3: Número de agencias CPN.

El área como tal no cuenta con una misión y visión específica.

Aproximadamente en el mes de mayo del 2014 la empresa considero necesario implementar una herramienta que permita tener un mejor manejo de la mesa de servicios para lo cual la intuición vio la necesidad de plantearse objetivos generales y específicos que conlleven a mejorar el servicio que el área de sistemas presta a los usuarios, los mismos que se describen a continuación.

### 2.2.2. OBJETIVO GENERAL

Administrar los recursos tecnológicos y de la información de forma eficiente, eficaz y responsable; de una manera que no afecte la calidad del servicio que brinda la CPN a sus socios. (Victoria & Guapas, MANUAL DE LA GESTIÓN DE TECNOLOGÍA INFORMATICA Y TELECOMUNICACIONES, 2015)

### **2.2.3. OBJETIVOS ESPECÍFICOS**

1. Administrar el proceso de Gestión de Tecnología y Telecomunicaciones (GTT) como un modelo de gestión de entrega de servicios tecnológicos de alta calidad, asociado a la capacidad operativa de funcionamiento.
2. Asegurar la integridad, seguridad y disponibilidad de los recursos tecnológicos y de la información que maneja la CPN.
3. Regular la demanda de servicios tecnológicos sobre la demanda del negocio.
4. Asegurar la puntualidad y la calidad del servicio de GTT a los diferentes procesos de la CPN.
5. Promover el uso adecuado de la tecnología, mediante la educación, para fomentar el manejo responsable de la información a la cual se accede. (Victoria & Guapas, MANUAL DE LA GESTIÓN DE TECNOLOGÍA INFORMATICA Y TELECOMUNICACIONES, 2015)

### **2.2.4. DIAGRAMA DE FLUJO DESARROLLO DE APLICACIONES**

Para el desarrollo de las aplicaciones el área de sistemas considero necesario el generar un diagrama de actividades como se evidencia en la Figura 4, la misma que inicia con una reunión entre el subárea de desarrollo y el área solicitante de la aplicación, posterior a esta reunión se elabora un RFD (Requisitos para el Desarrollo) en el cual consta todos los requerimientos de la aplicación; una vez aprobado el RFD se empieza con el desarrollo de la aplicación y se desarrollan las pruebas por parte del subárea de desarrollo y del área interesada, por último una vez que se ha aprobado las pruebas se procede a colocar el sistema en producción para su funcionamiento.

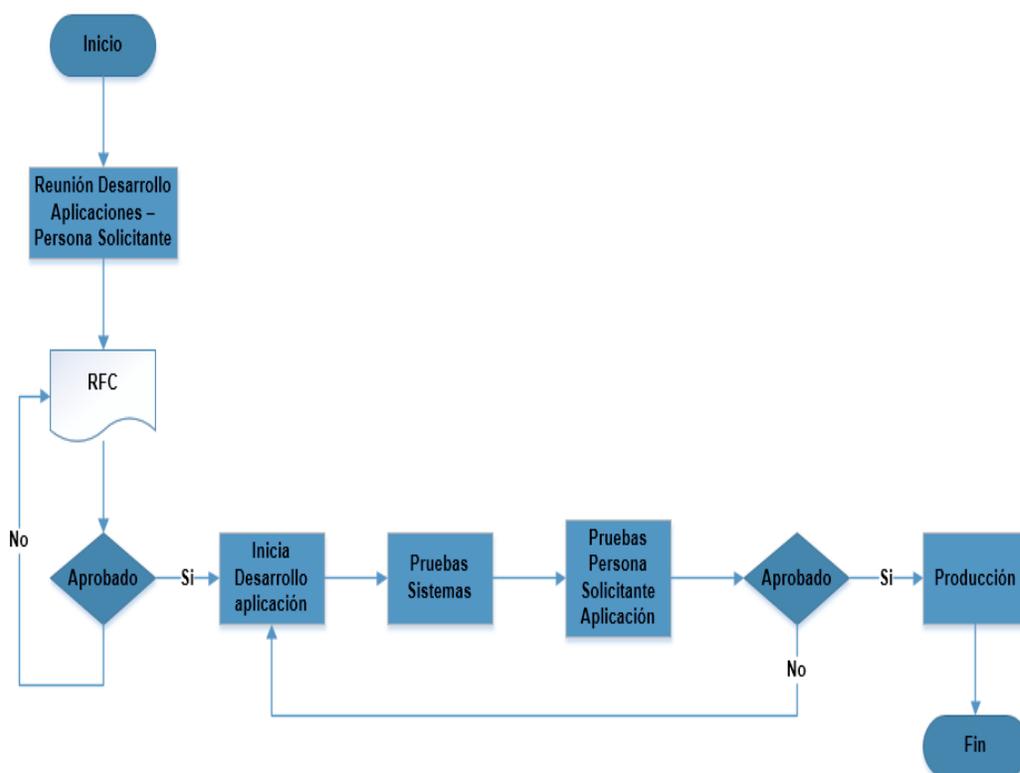


Figura 4. Proceso para el Desarrollo de Aplicaciones.

### 2.2.5. ESTRUCTURA FUNCIONAL

En la Figura 5, se observa la estructura funcional del área de sistemas que está en vigencia a partir del 2014; cabe mencionar que en el inicio de la CPN no existía el área de sistemas y no es hasta el año de 1991 dónde se crea el área con tres personas para la misma (Ver Gráfico 1) y a partir de esta fecha es donde la CPN comienza a crecer tanto en su nómina de empleados como en sus agencias en cada provincia del Ecuador (Ver Gráfico 2 y 3).

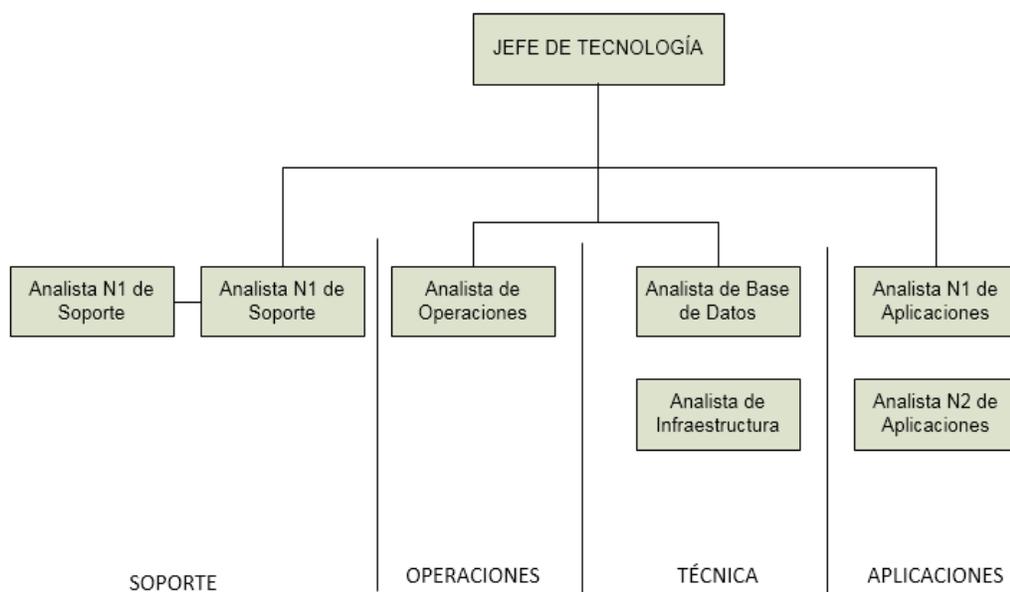


Figura 5. Estructura Funcional del área de Sistemas.

Fuente: (Victoria & Guapas, *MANUAL DE LA GESTIÓN DE TECNOLOGÍA INFORMATICA Y TELECOMUNICACIONES*, 2015)



Figura 6 Número de personas en el área de sistemas.

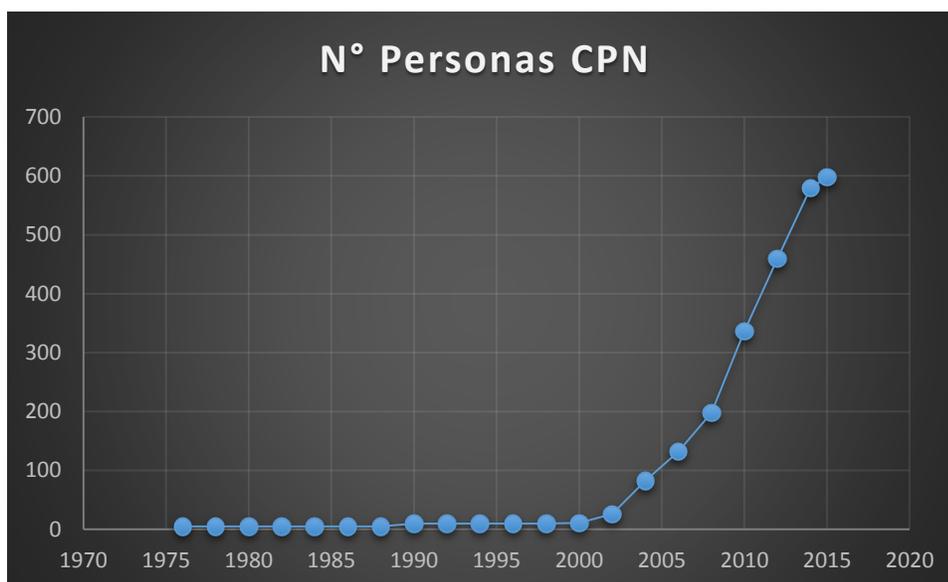


Figura 7 Número de personas en la CPN

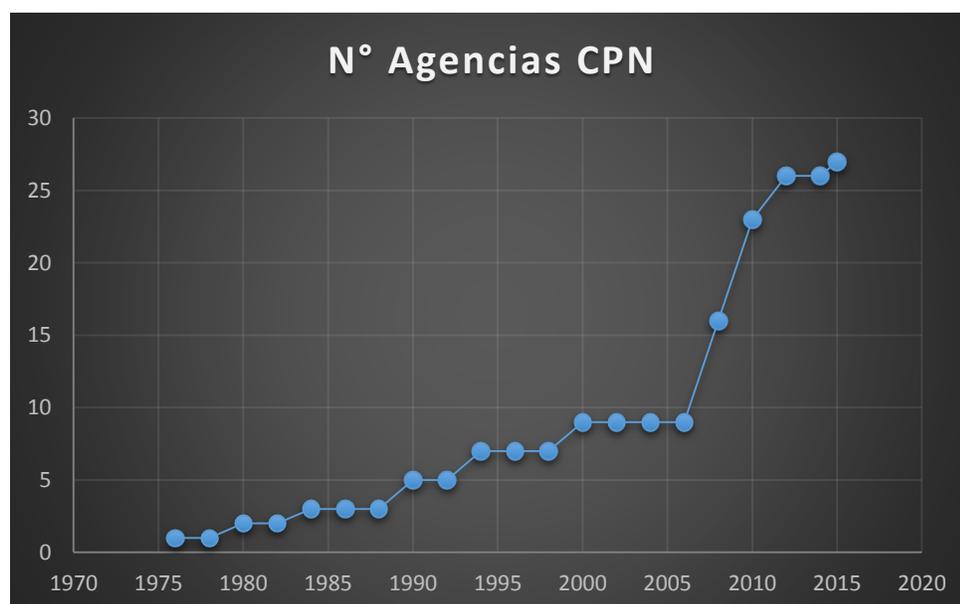


Figura 8 Número de agencias de la CPN.

### 2.2.6. SERVICIOS

A continuación, se describen los servicios que se prestan por parte del área de GTT. (Victoria & Guapas, CATALOGO DE SERVICIOS TECNOLOGICOS GTT, 2015).

Se agrupan en tres grandes servicios:

## 1. SOPORTE DE:

01



### Sistemas

Se refiere al soporte relacionado a manejo de errores, inoperatividad o degradación de rendimiento de los sistemas aplicativos vigentes en la CPN.

El servicio cubre:

- COBIS
- Nómina
- Inventarios
- Activos Fijos
- SPI
- Cooperativa Virtual
- SoftExpert
- SMS
- Huella Digital
- Punto Mático
- Image Folder
- CRM
- Extreme WEB
- Anexo Transaccional

Si requiere el servicio

Pedirlo a:

Extensión 0111

<http://www.cooperando.fin.ec/soporte>

En horario:

**Eventos Normales**

Lunes a Viernes -  
08:00-16:30

**Eventos Críticos**

Lunes a Domingo -  
00:00-23:59

Restringido en:

Períodos de cierre  
Fines de semana y  
feriado

02



### Comunicaciones

Se refiere al soporte relacionado a las actividades de revisión, restauración y mantenimiento de los equipos tecnológicos de comunicación de la CPN.

El servicio cubre:

- Enlace de Datos
- Internet
- Intranet
- Telefonía IP
- Telefonía Celular
- Correo
- Videoconferencia

Si requiere el servicio

Pedirlo a:

Extensión 0111

<http://www.cooperando.fin.ec/soporte>

En horario:

**Eventos Normales**

Lunes a viernes -  
08:00-16:30

**Eventos Críticos**

Lunes a Domingo -  
00:00-23:59

Restringido  
en:

Períodos de  
cierre  
Fines de  
semana y  
feriado

## 03

**Equipos**

Se refiere a las actividades de soporte que se brinda al usuario final de la CPN con respecto a los equipos que le han sido asignados para desempeñar sus actividades diarias.



El servicio cubre:

- Computo
- Impresoras
- Scanner
- Ofimática
- Sistema Operativo
- Antivirus

Si requiere el servicio

Pedirlo a:

Extensión 0111

<http://www.cooperando.fin.ec/soporte>

En horario:

**Eventos Normales**

Lunes a viernes - 08:00-16:30

**Eventos Críticos**

Lunes a Domingo - 00:00-23:59

Restringido

en:

Períodos de cierre

Fines de semana y feriado

## 2. SOLICITUD DE:

## 01

**Permisos**

Se refiere a las actividades que habilitan, modifican o inhabilitan el acceso a los sistemas aplicativos vigentes y comunicaciones en la CPN.



El servicio cubre:

- Acceso
- Telefonía
- Internet

Si requiere el servicio

Pedirlo a:

Extensión 0111

<http://www.cooperando.fin.ec/soporte>

En horario:

**Eventos Normales**

Lunes a viernes - 08:00-16:30

**Eventos Críticos**

Lunes a Domingo - 00:00-23:59

Restringido

en:

Períodos de cierre

Fines de semana y feriado

## 02

**Equipos**

Se refiere a las actividades de mantenimiento y cambio de equipamiento tecnológico.



El servicio cubre:

- Mantenimiento
- Cambio

Si requiere el servicio

Pedirlo a:

Extensión 0111

<http://www.cooperando.fin.ec/soporte>

En horario:

**Eventos Normales**

Lunes a viernes - 08:00-16:30

**Eventos Críticos**

Lunes a Domingo - 00:00-23:59

Restringido

en:

Períodos de cierre

Fines de semana y feriado

## 03 Conexiones

Se refiere a las actividades para el establecimiento de enlaces de datos y comunicación entre Matriz y las diferentes Sucursales de la CPN.



El servicio cubre:

- VPN
- Enlaces

Si requiere el servicio

Pedirlo a:

Extensión 0111

<http://www.cooperando.fin.ec/soporte>

En horario:

**Eventos Normales**

Lunes a viernes - 08:00-16:30

**Eventos Críticos**

Lunes a Domingo - 00:00-23:59

Restringido en:

Períodos de

cierre

Fines de

semana y

feriado

## 04 Acreditaciones y Descuentos

Se refiere a las actividades que se realizan en el sistema COBIS para reflejar acreditaciones y descuentos.



El servicio cubre:

- Valores

Si requiere el servicio

Pedirlo a:

Extensión 0111

<http://www.cooperando.fin.ec/soporte>

En horario:

**Eventos Normales**

Lunes a viernes - 08:00-16:30

**Eventos Críticos**

Lunes a Domingo - 00:00-23:59

Restringido en:

Períodos de

cierre

Fines de

semana y

feriado

### 3. DESARROLLO DE:

## 01 Reportes

Se refiere al desarrollo de nuevas aplicaciones y reportes que la CPN requiera sobre sus aplicaciones vigentes y que estas signifiquen un trabajo interno menos a 160 horas de trabajo hombre.



El servicio cubre:

- Aplicaciones Menores
- COBIS

Si requiere el servicio

Pedirlo a:

Extensión 0111

<http://www.cooperando.fin.ec/soporte>

En horario:

**Eventos Normales**

Lunes a viernes - 08:00-16:30

**Eventos Críticos**

Lunes a Domingo - 00:00-23:59

Restringido en:

Períodos de

cierre

Fines de

semana y

feriado

### **2.3. ANÁLISIS**

La CPN desde sus inicios hasta marzo del año 2014 pertenecía al segmento 3 del sector financiero, razón por la cual la entidad no reportaba ni era inspeccionada por ningún ente de control. En febrero del año 2014 se presentó auditoría externa de la Superintendencia de Economía Popular Y Solidaria (SEPS) donde después de una revisión completa y el respectivo análisis de la entidad, tomaron la decisión de trasladar a la CPN al segmento 1 por su gran cantidad de socios y activos.

Consecuente a esto el área de Sistemas fue una de las áreas que más observaciones tuvieron por parte de la SEPS y entre algunas de las observaciones una pertenece al personal de la unidad de aplicaciones indicando que no dispone de una metodología de desarrollo de software para la elaboración de sus aplicaciones.

La CPN comprometida con todos sus socios de brindar una atención de calidad y el área de sistemas de ayudar a que los productos software y servicios estén cien por ciento disponibles para los usuarios y socios ha implementado desde junio del 2014 el proyecto ITIL aplicado al área de tecnología para todos sus servicios y productos, en especial a la mesa de servicios.

Por este motivo el área de sistemas necesita implementar una metodología de desarrollo la misma que debe acoplarse a sus necesidades.

## **CAPÍTULO III. ESTADO DEL ARTE**

### **3.1. METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE**

#### **3.1.1. CONCEPTO**

Método que permite agregar cambios con rapidez en el desarrollo de aplicaciones software. En varias ocasiones, los modelos tradicionales de gestión no sirven para enfrentar un desafío que hoy en día resulta primordial el cual es incorporar cambios con rapidez y en cualquier etapa del proyecto. (EcuRed, 2015).

#### **3.1.2. HISTORIA**

El modelo Waterfall fue recién definido en 1970 por Wiston Royce en su artículo "Administrando el desarrollo de sistemas de software grandes" - todavía sin el nombre "Waterfall" y sin las etapas con las que finalmente se usó ampliamente- mientras que la ideas metodologías ágiles son anteriores. La idea de desarrollo en ciclos cortos iterativos e incrementales data de 1939 y el término Scrum como analogía del Rugby para el desarrollo de software data de 1986. (EcuRed, 2015)

De acuerdo a un estudio realizado en el Ecuador el año pasado se comprobó la efectividad del uso de metodologías ágiles en el desarrollo de software. Las implementaciones de dichas metodologías son relativamente nuevas en el país, basados en experiencias de países hermanos se ha apoyado esta innovación por el gobierno desde el año 2008. (Escuela Superior Politécnica del Litoral (ESPOL)., 2014)

En el vecino país de Colombia las metodologías ágiles van tomando más terreno en los nuevos desarrollos por lo cual han creado una comunidad desde el año 2011 en la cual comparten experiencias y nueva

información de calidad que se pueden aplicar en proyectos futuros. (Ágiles Colombia, 2015)

Agiles Argentina es una División de SADIO, la Sociedad Argentina de Informática, la cual es conformada por un conjunto de personas con ambiciones de desarrollar productos y complacer a sus usuarios y clientes. (SADIO, 2015)

Pretenden que las ideas de Desarrollo de Software Ágil puedan ayudar a esto, es por eso que tratan de aprender y crecer juntos, intercambiando conocimientos y experiencias desde el 2008. (Comunidad Latinoamericana de Metodologías Ágiles, 2015)

### **3.1.3. MANIFIESTO ÁGIL**

El 17 de febrero de 2001 diecisiete críticos de los modelos de mejora del desarrollo de software basados en procesos, fueron convocados por Kent Beck, quien un par de años antes había publicado el libro Explicando Programación Extrema, en el que exponía una nueva metodología denominada Programación Extrema, la congregación la realizaron en Snowbird, donde se trató sobre las técnicas y procesos para desarrollar software. Fue en esta reunión donde se acuñó el término “Métodos Ágiles” mismo que definía a las técnicas que estaban naciendo como alternativa a las metodologías formales a las que consideraban excesivamente “pesadas” y rígidas por su perfil normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo. (Fundación Wikimedia, Inc., 2015)

El Manifiesto Ágil menciona que se han encontrado nuevos métodos para desarrollar software, haciendo uso de ellos y brindando ayuda a otros para que lo hagan.

Los cuatro postulados del manifiesto ágil son:

- A los sujetos y su interacción, sobre los procesos y las herramientas.
- El software que funciona, es más relevante de la documentación.
- La colaboración con el cliente, ante la negociación contractual.
- La respuesta al cambio, antes que el seguimiento de un plan.  
(Fundación Wikimedia, Inc., 2015).

#### **3.1.4. CICLO DE VIDA**

En la Figura 6, se observa como un desarrollo ágil se basa en iteraciones en las que inicialmente se planifica el proyecto con los requisitos funcionales que el cliente desea. El desarrollo prosigue a dicha planificación, luego se logra ya un despliegue del desarrollo atravesando por pruebas y finalmente se comparte el proyecto con el usuario, quien es el encargado de mencionar o agregar nuevos puntos o funcionalidades que se deberán realizar en la siguiente iteración.

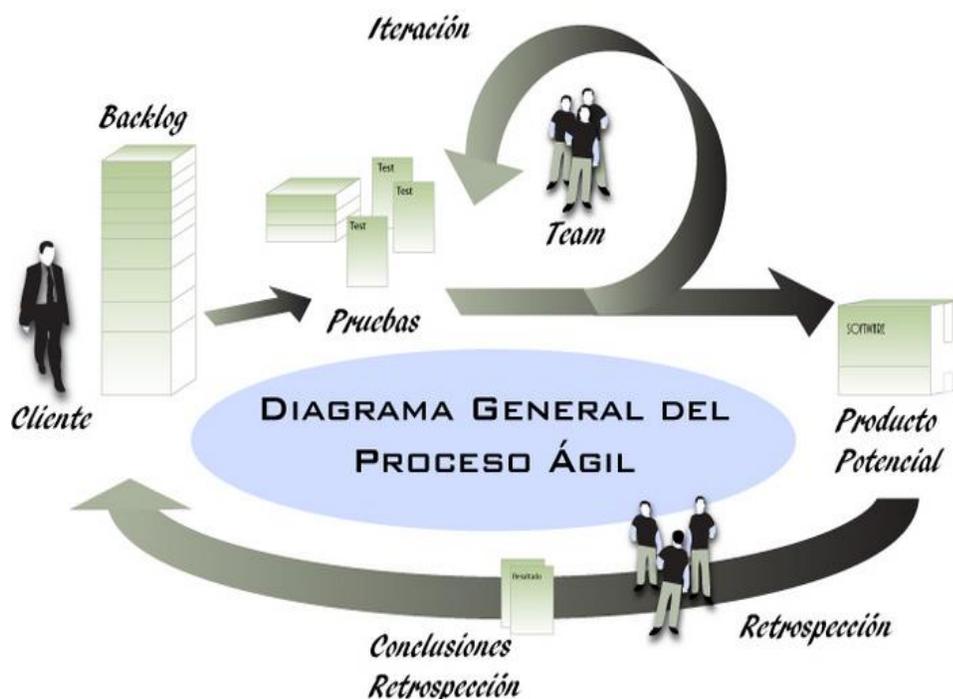


Figura 9: Diagrama General del Proceso Ágil.

Fuente: (Android, Entrepreneur, IT, Management, 2013)

### 3.1.5. TRABAJOS RELACIONADOS

A lo largo del presente trabajo se han encontrado varios artículos científicos relacionados con el tema principal, a continuación, un pequeño resumen de dichos trabajos.

En el año 2011 se creó un artículo enfocado al proceso de desarrollo de software, exponiendo las características, ventajas y desventajas de cada metodología. Se tomó como referencia 4 enfoques metodológicos: RUP, MSF, XP y Scrum. Este trabajo aportó gran conocimiento para interpretar de mejor manera como realizar comparaciones entre metodologías y así optar por la más afín a las necesidades y requerimientos iniciales.

(Pérez, 2011)

Propuesta Metodológica para el Desarrollo de Software de Investigación, es un trabajo muy bien desarrollado, con un gran nivel de

detalle en cuanto a la manera más eficiente de enfrentar un problema al obtener características que destacan cada metodología y su principal fortaleza. En el presente estudio se busca mitigar ciertos riesgos y complicaciones que tiene el personal de desarrollo al enfrentar el reto de construir un nuevo prototipo o peor aún tomar uno anteriormente desarrollado por otro personal, entender su código y su lógica para así poder generar o acoplar nuevos módulos funcionales sin que la producción se vea afectada. (Espina, 2010)

Ingeniería del software: Metodologías y ciclos de vida, es un estudio realizado por el Ministerio de Industria, Turismo y Comercio de España a través del Instituto Nacional de Tecnologías de la Comunicación realiza un análisis de los componentes, características, tipos y aplicaciones del software, además de los ciclos de vida más comunes entre las metodologías ágiles facilitando el entendimiento de los conceptos utilizados comúnmente en este tipo de trabajos. Este estudio se enfoca en cuatro partes fundamentales para aclarar cualquier duda sobre el desarrollo de software utilizando una metodología ágil.

(Ministerio de Industria, TInstituto Nacional de Tecnologías de la Comunicación, 2009)

Desarrollo de software en proyectos de telecomunicaciones de corta duración es un estudio que se enfoca en buscar la metodología idónea que se adapte a la empresa de tecnología que requiere realizar la creación de nuevos productos software y también mejor los productos ya existentes de una manera práctica y sobre todo rápida. (Bacca, 2014)

Metodologías Ágiles es un estudio que aborda varias metodologías de desarrollo ágil tomando como precedente que hoy por hoy una de las más utilizadas es XP Extreme Programming, pero sin dejar de lado los

beneficios que presentan otras metodologías de desarrollo de software Scrum, FDD, ASD, entre otras. Cada metodología desempeña un papel importante al momento de realizar una determinada labor o cumplir con un objetivo específico, es en este punto donde cada característica se prioriza desembocando en una ventaja o desventaja para la solución que será creada a futuro. Las características comunes entre las ya mencionadas metodologías son por ejemplo la orientación a obtener resultados en el menor tiempo posible, que el prototipo presentado en cada iteración sea funcional, que sea adaptable a cualquier cambio a lo largo del desarrollo, manejar una correcta documentación sin que ésta demande mucho tiempo y por último una de las más importantes, el trato cara a cara del usuario con el equipo de trabajo, facilitando de esta manera la comunicación entre ambas partes y obteniendo mejores resultados. (Ramírez, 2008)

Metodologías para desarrollar software seguro, como el nombre del tema lo indica, este estudio se centra en analizar uno de los principales requerimientos que varias veces se pasan por alto y pasan a ser del tipo no funcional ya que en el enfoque global se mira a la seguridad como parte de la calidad del software. Cabe recalcar que en esta época los hackers y demás grupos criminales que se dedican al fraude informático evolucionan a gran velocidad descubriendo fallas y vulnerabilidades en cualquier tipo de software. Por lo tanto, al momento de seleccionar una metodología ágil para el desarrollo de software también se debe dar el tiempo y la prioridad necesaria a la seguridad que se ofrece como característica al igual que los demás elementos que llevan a la toma de decisiones. (Joaquín, 2013)

Modelos de aceptación de metodologías de desarrollo de software, nuevamente es un caso de estudio que se relaciona con la toma de decisiones al momento de escoger la más óptima metodología de desarrollo, basándose en indicadores como la recopilación de información,

fases subsecuentes, comparación entre ellas y destacando los resultados obtenidos. El estudio ha fijado como referencia 17 aspectos o factores de comparación siendo los más destacados la compatibilidad y adaptabilidad, dejando de lado a otros como el tamaño de la organización ya que no aportaron gran valor final. (Godoy Alvarez & Taype Calderón, 2015).

Taxonomía de los modelos y metodologías de desarrollo de software más utilizados toma como principal característica los costos que deberán ser asumidos por la empresa o el usuario que requiera un determinado proyecto para solventar sus necesidades. El costo es un factor sumamente importante ya que influye directamente en el presupuesto que se destina para cada proyecto y dependiendo de este un desarrollo puede continuar o detenerse por completo por el cliente. (Gómez Fuentes, 2012)

## **3.2. XP (PROGRAMACIÓN EXTREMA)**

### **3.2.1. ORIGEN**

Es una metodología ágil que se centra en potenciar las relaciones interpersonales como punto clave para el éxito del desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen ambiente laboral. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los integrantes, facilidad en las soluciones implementadas y bravura para afrontar los cambios. También se define como esencialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Agile Alliance, 2015)

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe

la filosofía de XP en dos, sin cubrir los detalles técnicos y de implantación de las prácticas. (Fundación Wikimedia, Inc., 2015)

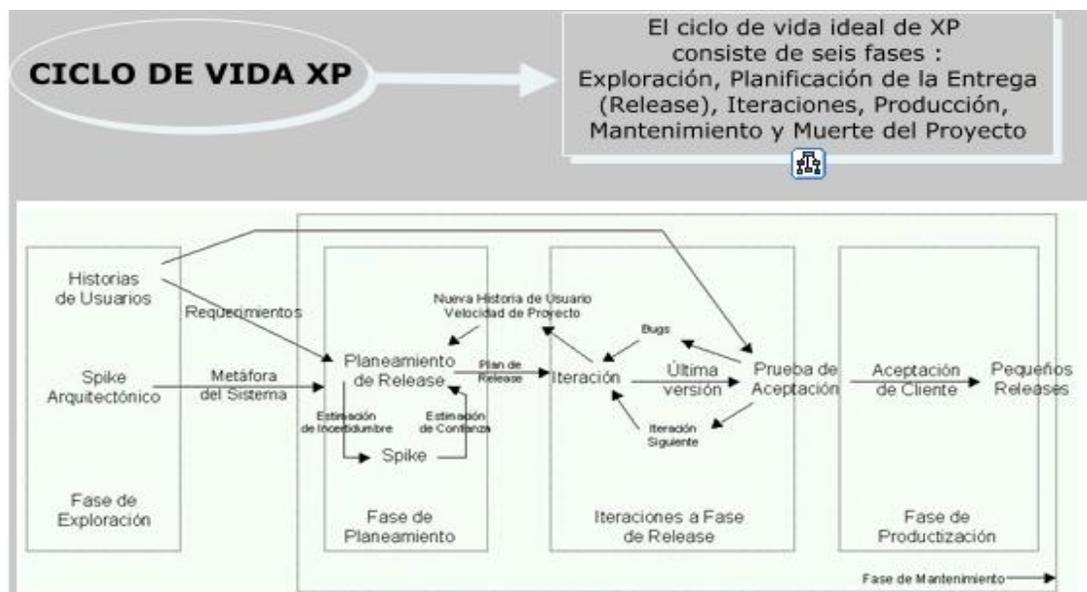


Figura 10: Ciclo de vida de eXtreme Programming.

Fuente (Ing. Software, 2012)

### 3.2.2. ENTREGABLES

**Las Historias de usuario:** Las historias de usuario son el producto inicial de la fase 1 (Exploración). Son simples tarjetas de papel en las que el cliente describe rápidamente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario no es nada rígido ya que en cualquier momento estas pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. (Highsmith, 2006)

Respecto de la información incluida en la historia de usuario, existen varias plantillas sugeridas, pero no existe una lista definitiva. Beck en su libro presenta un ejemplo de historia de usuario, en la Figura 7, se puede visualizar su descripción y contenido.

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre:</b> Elaboración de Presupuesto
<b>Usuario:</b> Desarrollo Empresarial	
<b>Modificación de Historia N°:</b> 0	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Media (Alta/Media/Baja)	<b>Puntos estimados:</b>
<b>Desarrollador Encargado:</b> Luis Ariel Castillo Estupiñan	
<b>Descripción:</b> Elaborar el presupuesto para la ejecución del evento. Depende de la disponibilidad presupuestal estipulada para la dependencia para el año vigente.	
<b>Observaciones:</b> El presupuesto está estipulado para la dependencia para el año, lo cual implica que para cada evento se tiene un máximo de recursos dependiendo de la naturaleza del mismo.	

Figura 11: Historia de Usuario

Fuente: (SlidePlayer.es.Inc., 2015)

Si al inicio no se identifican todas las historias de usuario, no es una razón para preocuparse ya que al inicio de cada iteración estarán registrados los cambios en las historias de usuario y según eso se planificará la siguiente iteración.

**Prototipo básico:** Permite testar el objeto antes de que entre en producción, detectar errores, deficiencias, etcétera. Cuando el prototipo está suficientemente perfeccionado en todos los sentidos requeridos y alcanza las metas para las que fue pensado, el objeto puede empezar a producirse.

**Cronograma de trabajo:** Es un gran apoyo en cuanto a la gestión de proyectos. Puede tratarse de un documento impreso o de una aplicación digital. El cronograma contiene un listado de actividades con las fechas previstas de su inicio y final.

**Historias de usuario no abordadas:** Son la recopilación de todas las historias de usuario no mencionadas o no tomadas en cuenta al principio y otras más que pueden surgir según la necesidad del cliente. Estas historias pueden ser totalmente nuevas o ser la modificación de una historia de usuario anterior.

**Pruebas de aceptación no superadas en las iteraciones anteriores:** Cada etapa o fase es evaluada continuamente según un plan de pruebas, del resultado de estas depende el avance que se pueda lograr ya que si dichas pruebas no son superadas en una iteración, lo mínimo que se espera es que a la siguiente esta satisfaga las necesidades del cliente.

**Tareas no terminadas en la iteración anterior:** Para cada iteración previamente se fijan las tareas que se deben cumplir o al menos tratar de hacerlo. En ocasiones los tiempos estimados no son los correctos por lo que las tareas pueden superar el tiempo estimado generando una acumulación de tareas para la siguiente iteración.

**Avances del desarrollo:** En todo equipo de trabajo debe existir al menos un encargado de documentar y formalizar los avances logrados en cuanto al desarrollo del proyecto. Deberá interactuar con la parte de los programadores, así como con el usuario teniendo en cuenta el cronograma de trabajo planteado al inicio por cada tarea.

**Documentación final:** Es el respaldo ya sea físico, digital o ambos del desarrollo del proyecto desde el inicio hasta la culminación. Consta de acuerdos iniciales, formalización, requerimientos, propuestas técnicas de solución, diagramas realizados, resultados de las pruebas y una descripción a modo de manual de usuario que explicará el funcionamiento

del proyecto completamente funcional de acuerdo a los requerimientos del cliente.

**Entrega del sistema de acuerdo a las funcionalidades requeridas por el cliente:** Entrega del proyecto demostrando completamente su funcionalidad al satisfacer las necesidades del cliente.

### 3.2.3. ROLES

**Programador:** Se encarga de crear el código y realizar pruebas unitarias del sistema. Debe haber una comunicación y coordinación apropiada entre los programadores y otros miembros del equipo.

**Cliente:** Se encarga de la creación de las historias de usuario y pruebas funcionales para validar su ejecución. Además, prioriza las historias de usuario y decide cuáles se efectúan en cada iteración. (Fundación Wikimedia, Inc., 2015)

**Encargado de pruebas (Tester):** El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Continuamente ejecuta pruebas, transmite los resultados al equipo y es responsable de las herramientas de soporte para pruebas.

**Encargado de seguimiento (Tracker):** Facilita la realimentación al equipo en el proceso XP. Verifica el nivel de acierto entre las estimaciones realizadas y el tiempo real, notificando los resultados para mejorar futuras estimaciones. También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son factibles con las limitaciones de tiempo y recursos presentes. Determina cuándo es necesario realizar

algún cambio para lograr los objetivos de cada iteración. (Fundación Wikimedia, Inc., 2015)

**Entrenador (Coach):** Es responsable del proceso global. Debe conocer a fondo el proceso y la metodología usada ya que debe guiar al su equipo de trabajo con el fin de se apliquen las mejores prácticas XP y que además se siga el proceso correctamente.

**Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

**Gestor (Big boss):** Es el lazo entre clientes y el equipo de programadores, ayuda a que este trabaje de manera efectiva, creando las condiciones apropiadas. Su labor esencial es de coordinar. (Wells, 2013)

### 3.2.4. PROCESO XP

Tabla 1  
Proceso de XP.

XP				
FASES	ACTIVIDADES	ROLES	HERRAMIENTAS	PRODUCTOS
Fase I: Exploración	1. Plantear sin mucho detalle las historias de usuario que son prioridad para la primera entrega del producto.	Cliente.	Tarjetas de papel Hojas de Cálculo.	Historias de usuario.
	2. Amoldarse con las herramientas, tecnologías y prácticas que se utilizarán en el desarrollo del proyecto.	Programador Cliente Encargado de pruebas	CASE para desarrollo de software.	No Aplica
	3. Se prueba la tecnología y se investigan las posibilidades de aplicar a la arquitectura del sistema en la construcción de un prototipo	Programador Cliente Encargado de pruebas Entrenador	CASE para desarrollo de software.	Prototipo básico
Fase II: Planificación de la Entrega	1. Establecer la prioridad de cada historia de usuario.	Programador Cliente Entrenador Gestor	Procesadores de Texto, Hojas de Cálculo, Herramienta para priorizar requisitos.	Documento con prioridad entre historias de usuario.
	2. Estimar el esfuerzo según la prioridad que se asignan a las historias de usuario utilizando como medida el punto.	Programador Cliente Entrenador Consultor	Hojas de Cálculo, CASE para desarrollo de software y Software para gestión de proyectos.	Documento que determina el esfuerzo requerido por las tareas.
				Continúa 

	3. Determinar un cronograma de actividades a realizarse.	Cliente Encargado de pruebas Encargado de seguimiento Entrenador Gestor	CASE para desarrollo de software.	Cronograma de Actividades.
Fase III: Iteraciones	1. Estructuración de varias iteraciones tomando como referencia una inicial y una final.	Programador Cliente Encargado de pruebas Encargado de seguimiento Entrenador Gestor	Hojas de Cálculo, CASE para desarrollo de software.	Documento base para seguimiento de actividades.
	2. Constituir una arquitectura del sistema que pueda ser manejada a lo largo del proyecto.	Entrenador Consultor	CASE para desarrollo de software.	Arquitectura del sistema.
	3. Realización de pruebas y seguimiento al proceso de desarrollo derivando en nuevas iteraciones.	Encargado de pruebas Encargado de seguimiento Entrenador	Herramientas de gestión de pruebas.	Plan de pruebas.
Fase IV: Producción	1. La fase de producción demanda la realización de pruebas adicionales y validaciones de rendimiento.	Programador Encargado de pruebas Encargado de seguimiento Consultor	Herramientas para pruebas funcionales, Herramientas para pruebas de carga y rendimiento.	Evaluación de resultados obtenidos en pruebas. Continúa 

	2. Las ideas planteadas y sugeridas son documentadas para su posterior implementación.	Programador Encargado de seguimiento Consultor Gestor	Procesadores de Texto, Hojas de Cálculo, Herramienta para requisitos.	Actualización de documentación historias de usuario y prioridades.
Fase V: Mantenimiento	1. El proyecto XP debe mantener la primera versión de producción en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones.	Programador Cliente Encargado de seguimiento Consultor	Procesadores de Texto, Hojas de Cálculo, CASE para desarrollo de software.	Evolución del modelo de producción.
	2. Continuidad en las tareas de soporte para el cliente.	Programador Cliente Encargado de seguimiento	Procesadores de Texto, Hojas de Cálculo, CASE para desarrollo de software.	Nuevas funcionalidades en el producto.
Fase VI: Muerte del Proyecto	1. Se genera la documentación final del sistema.	Programador Encargado de seguimiento Entrenador	Procesadores de Texto, Hojas de Cálculo.	Producto final completamente probado y documentación actualizada.

### **3.3. SCRUM (RUGBY (MELÉ))**

#### **3.3.1. ORIGEN**

Scrum es un modelo de desarrollo ágil, el cual posee las siguientes características:

- Acoger un plan de desarrollo incremental, en lugar realizar una planificación y ejecución completa del producto.
- Afirmer la calidad del resultado haciendo énfasis en el conocimiento comprendido de las personas que conforman equipos auto organizados, que en la calidad de los procesos empleados. (Schwaber, 2010)
- Trasponer las diferentes etapas del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o de cascada.

Este modelo fue identificado y definido por Ikujiro Nonaka e Hirotaka Takeuchi a principios de los 80, analizando como grandes empresas fabricaban sus nuevos productos de tecnología: Fuji-Xerox, Canon, Honda, Nec, Epson, Brother, 3M y Hewlett-Packard (Nonaka & Takeuchi, The New Product Development Game, 1986). (Palacio, 2014)

¿Qué significa SCRUM? Pues la palabra se usa en rugby y prácticamente significa melé. Al iniciar un encuentro de este juego todos los participantes se agrupan haciendo uso de su fuerza como equipo con el fin de ganar la pelota y pasársela a otro que espera atrás. Pues eso es SCRUM no son siglas ni nada por el estilo. Lo interesante de este juego en específico es que todos dependen de los demás, ya que si uno falla puede afectar a todos los miembros del grupo, por lo cual la coordinación es sumamente importante. (Wirwin, 2015)

### 3.3.2. INTRODUCCIÓN AL MODELO

El marco técnico de Scrum, por su sencillez, resulta apropiado para equipos y organizaciones que quieren comenzar a “avanzar en Scrum”.

Se conforma de un conjunto de prácticas y reglas que resultan válidos para solventar a manera de respuesta a los siguientes compendios de desarrollo ágil:

- Gestionar el progreso del avance, en lugar de la tradicional o predictiva, usada anteriormente.
- Trabajar basados en la calidad del resultado en el conocimiento tácito de las personas, más que en el explícito de los procesos y la tecnología empleada.
- Estratégicamente el desarrollo se lo realiza de manera incremental valiéndose de iteraciones (sprints) y revisiones constantes.
- Seguir los pasos del desarrollo ágil: desde el concepto o visión general de lo que el cliente necesita, construcción del producto de forma incremental a través de iteraciones rápidas que constan de fases de especulación – exploración y revisión. Estas iteraciones (en Scrum llamadas sprints) se repiten de forma continua hasta que el cliente da por cerrada la evolución del producto.

En Scrum se comienza con la visión general de lo que se desea obtener al final, de esta manera logra especificar más a fondo y dar prioridad a las actividades según sea pertinente. (Poppendieck, 2015)

Cada sprint de desarrollo o iteración finaliza con la entrega de una parte funcional del producto. La duración de cada sprint varía entre una y seis semanas, pero lo recomendable es que no se exceda de un mes.

En Scrum, el equipo monitoriza la evolución de cada sprint mediante reuniones diarias donde se revisa en conjunto el trabajo realizado por cada miembro el día anterior, y el previsto para el día en curso. Esta reunión diaria es de tiempo prefijado de 5 a 15 minutos máximo, en el cual se evalúa las tareas del día anterior y se fijan metas para cumplir hasta el fin del presente día. Esta reunión se conoce comúnmente como Scrum diario o si se desea empleando la terminología inglesa: “daily scrum” o “morning rollcall”. (Palacio, 2014)

En la Figura 8, podemos observar un resumen del proceso de Scrum.

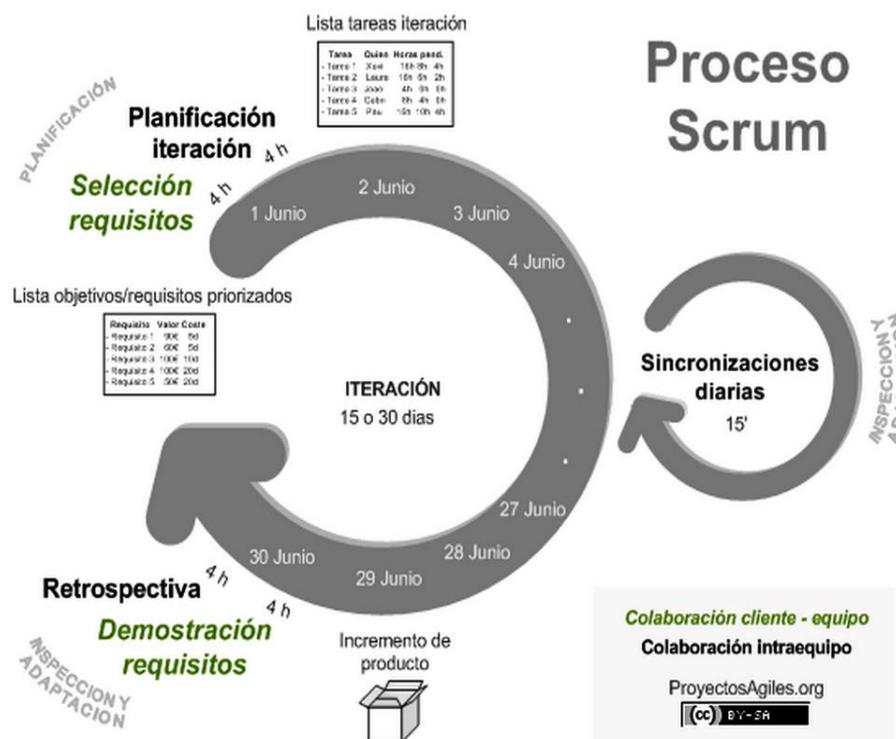


Figura 12: Proceso Scrum.

Fuente: (fundamentosagiles.org, 2015).

### 3.3.3. SCRUM TÉCNICO

En la Figura 9 se presenta el marco técnico de Scrum y está formado por:

## 1. Roles:

**El equipo de Desarrollo:** Lo forman el equipo de profesionales que se encargan de realizar el incremento de cada iteración o sprint.

Es recomendable que un equipo Scrum conste de entre 4 y 8 personas. Puesto que si se integra un grupo o equipo con más 8 resulta difícil mantener la comunicación directa, y se manifiestan con más intensidad los roces habituales de la dinámica de grupos (esto se crea a partir de grupos de 6 integrantes). En el número total de miembros del equipo de desarrollo no son considerados ni el Scrum Master ni el propietario del producto.

El equipo de desarrollo mantiene un espíritu de colaboración, y un designio común: obtener la mayor cercanía posible para la visión del cliente.

En el equipo:

- Todo el equipo conoce y comprende el enfoque del propietario del producto.
- Aportan y colaboran con el propietario del producto en la construcción de la pila del producto.
- Comparten conjuntamente el objetivo de cada sprint y la responsabilidad del logro.
- Todos los miembros son importantes y tiene la potestad de participar en las decisiones.
- Se guarda respeto frente a las opiniones y aportes de todos.

Noción:

- Grupo de personas que se encargan de la implementación de cada historia de usuario.
- Son auto-organizados.

Impacto:

- Se encarga de dividir las historias de usuario.
- Se encarga de desarrollar las tareas del sprint.
- Deben asistir a la reunión diaria de Scrum.
- Actualiza el avance de las tareas en el sprint Backlog.

Todos conocen el modelo de trabajo con Scrum.

**El propietario del producto:** El propietario del producto (product owner) se encarga de tomar las decisiones del cliente. Es responsable del valor del producto. Para facilitar la comunicación y toma de decisiones es necesario que este rol lo desempeñe una sola persona.

En resumen, el propietario de producto es quien:

- Toma la decisión de cómo será el resultado final, y el orden que se va a seguir para la construcción de 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, también se responsabiliza sobre fechas y funcionalidades de las diferentes versiones del mismo.

Noción:

- Es el responsable del producto.
- Es la persona que tiene la visión del negocio.
- Es el encargado de negociar con el equipo y con el Scrum Master la prioridad del trabajo a realizar.

Impacto:

- Incluye las historias de usuarios.

- Transformar las necesidades de negocio por parte del cliente en historias de usuario lo sumamente sencillas como para que se puedan convertir consecutivamente en tareas de desarrollo para el equipo.
- Mantener esta lista de historias de usuario (incluidas en el product backlog) debidamente priorizada enfocados al valor que ofrece cada historia para el cliente.
- Responsable de maximizar el retorno de inversión.

En los desarrollos internos para la propia empresa, suele asumir este rol el product manager o el responsable de marketing.

Respalda desarrollos para clientes externos, puesto que es el responsable del proceso de adquisición por parte del cliente.

Según las circunstancias del proyecto es posible incluso que delegue en el equipo de desarrollo, o en alguien de su confianza, sin perder la noción de que esa es una tarea suya.

**El Scrum Master:** Es responsable del cumplimiento de las reglas de un marco de Scrum técnico, certificando que se entienden en la organización, y se trabaja conforme a ellas.

Suministra 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.
- Modera las reuniones entre varios miembros según sea el caso.

- Resolución de impedimentos u obstáculos que en el sprint pueden retrasar la ejecución de las tareas.
- Gestiona el dinamismo del grupo dentro del equipo.
- Configuración, diseño y mejora continua de las prácticas de Scrum en la organización.

#### Noción:

- Es la persona que facilita las actividades que realiza el equipo.
- No es el jefe del proyecto.
- No es el jefe del equipo.

#### Impacto

- Responsable de guiar el desarrollo.
- Invita a las personas a las diferentes reuniones.
- Asegura la cooperación de todos los roles.
- Resuelve los impedimentos.

Al aumentar paulatinamente la comunicación de la organización y evolucionar hacia un marco de Scrum más experto, podría ser eliminado el rol de Scrum Master, cuando estas responsabilidades ya estén formalizadas en la organización.

**Interesados:** Una de las partes interesadas es cualquier persona que esté potencialmente afectada por el resultado del proyecto. El término se utiliza generalmente para nombrar a la gestión o los clientes.

#### Noción:

- Personas interesadas en el producto resultante del proceso de desarrollo.
- Obtienen un beneficio del proyecto (son clientes, proveedores, inversores y otros colaboradores).

## Impacto

- Participan en la reunión de planificación del sprint.
- Hacen posible el proyecto.

## 2. Artefactos:

**Pila del producto:** (product backlog) lista de requerimientos de usuario, que crece y evoluciona durante el desarrollo partiendo de la visión inicial del producto.

**Pila del sprint:** (sprint backlog) conjunto de trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.

**Incremento:** nombre que recibe cada iteración de desarrollo. Es el núcleo central que genera el pulso de avance por tiempos prefijados (time boxing).

## 3. Eventos:

**Planificación del sprint:** Reunión del equipo previo al inicio de cada sprint en la cual se decreta cuál va a ser el objetivo del sprint y las tareas necesarias para conseguirlo.

**Scrum diario:** Breve reunión diaria del equipo, en la que cada miembro responde a tres cuestiones:

1. El trabajo realizado el día anterior.
2. El que tiene previsto realizar.
3. Cosas que puede necesitar o impedimentos que deben eliminarse para poder realizar el trabajo.

Cada persona es responsable de actualizar en la pila del sprint el tiempo o esfuerzo pendiente de cada una de sus tareas, y con esta

información se actualiza a su vez el gráfico general con el que el equipo monitoriza el avance del sprint (burndown).

**Revisión del sprint:** Análisis e inspección del incremento generado, y adaptación de la pila del producto si resulta necesario.

**Retrospectiva:** Revisión de lo sucedido durante el Sprint. Reunión en la que el equipo analiza aspectos operativos de la forma de trabajo y crea un plan de mejoras para aplicar en el próximo sprint.

**Sprint:** Se denomina sprint a cada iteración de trabajo que produce una parte funcional del producto terminada y necesariamente operativa (incremento).

(Palacio, 2014)

### 3.3.4. REGLAS DE SCRUM

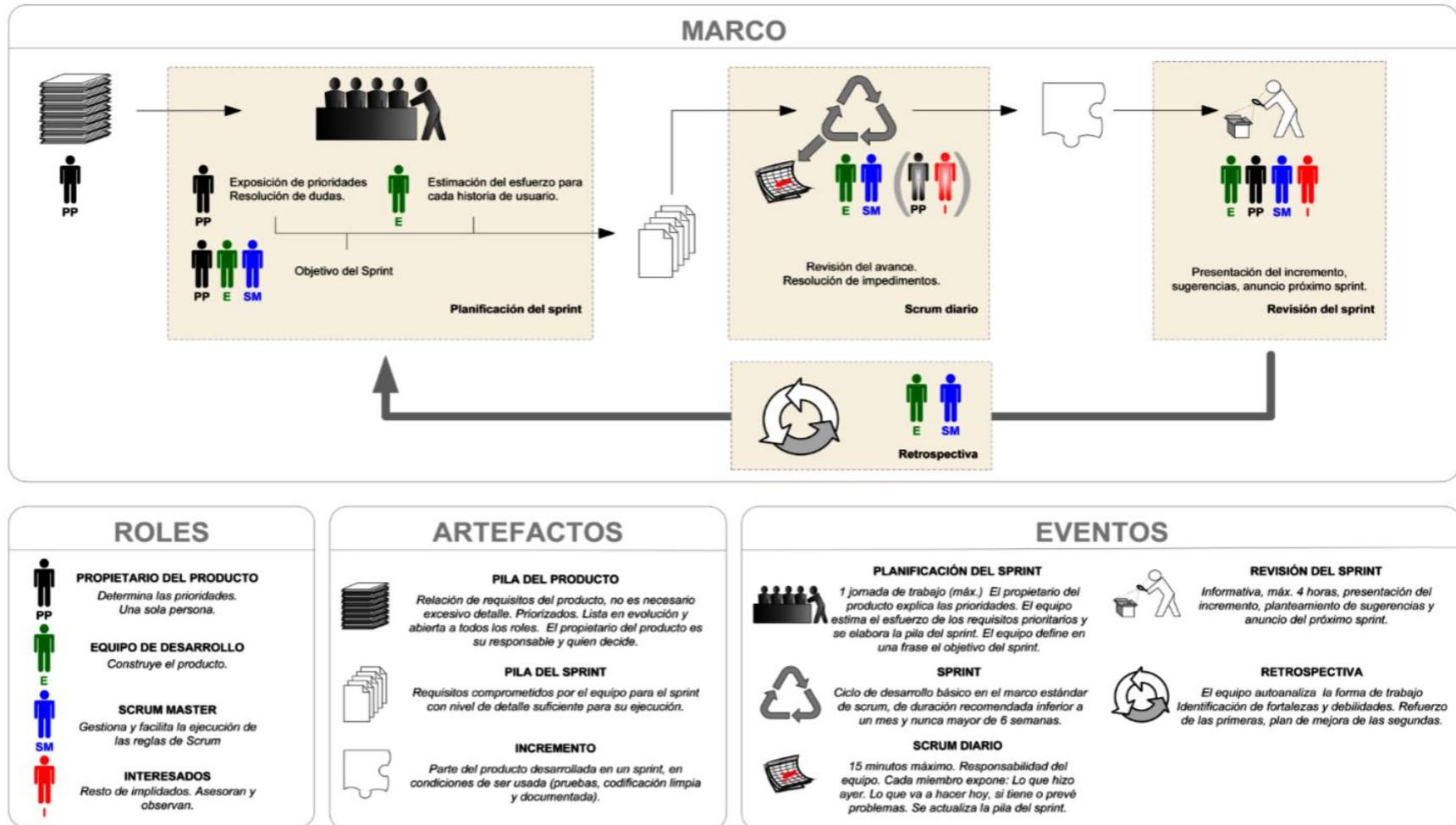


Figura 13: Marco Scrum Técnico.  
 Fuente: (MANAGER, 2015).

### 3.3.5. PROCESO SCRUM

Tabla 2  
Proceso de Scrum.

<b>SCRUM</b>				
<b>FASES</b>	<b>ACTIVIDADES</b>	<b>ROLES</b>	<b>HERRAMIENTAS</b>	<b>PRODUCTOS</b>
Fase 1: Reunión de planificación del sprint.	Reunión de trabajo previa al inicio de cada sprint en la que se establece el objetivo del sprint y las tareas necesarias para llegar a la consecución del mismo.	<ol style="list-style-type: none"> <li>1. Propietario del producto.</li> <li>2. Equipo de desarrollo.</li> <li>3. Scrum Master.</li> </ol>	CASE para desarrollo de software.	<ol style="list-style-type: none"> <li>1. Pila del sprint.</li> <li>2. Definir la duración del sprint y fecha de la reunión de revisión.</li> <li>3. Objetivo del sprint.</li> <li>4. Definición de tareas.</li> </ol>
Fase 2: Scrum diario.	Breve reunión diaria del equipo, en la que cada integrante evalúa y cuestiona lo siguiente: <ol style="list-style-type: none"> <li>1. El trabajo realizado el día anterior.</li> <li>2. El que tiene previsto realizar.</li> <li>3. Cosas que puede necesitar o impedimentos que deben eliminarse para poder realizar el trabajo.</li> </ol>	<ol style="list-style-type: none"> <li>1. Equipo de Desarrollo.</li> <li>2. Scrum Master.</li> <li>3. (Propietario del Producto).</li> <li>4. (Interesados).</li> </ol>	CASE para desarrollo de software.	<ol style="list-style-type: none"> <li>1. Pila del sprint y gráfico de avance (burn-down) actualizados.</li> <li>2. Identificación de posibles necesidades e impedimentos.</li> <li>3. Incremento.</li> </ol>
Fase 3: Revisión del sprint.	Análisis e inspección del incremento generado, y adaptación de la pila del producto si resulta necesario.	<ol style="list-style-type: none"> <li>1. Equipo de Desarrollo.</li> <li>2. Propietario del Producto.</li> <li>3. Scrum Master.</li> <li>4. Interesados.</li> </ol>	CASE para desarrollo de software.	<ol style="list-style-type: none"> <li>1. Feedback para el propietario del producto.</li> <li>2. Convocatoria de la reunión del siguiente sprint.</li> </ol>
Fase 4: Retrospectiva.	Revisión de lo sucedido durante el Sprint. Reunión en la que el equipo analiza aspectos operativos de la forma de trabajo. Crear un plan de mejoras para aplicar en el próximo sprint.	<ol style="list-style-type: none"> <li>1. Equipo de Desarrollo.</li> <li>2. Scrum Master.</li> </ol>	CASE para desarrollo de software.	<ol style="list-style-type: none"> <li>1. Plan de mejoras.</li> <li>2. Identificación de fortalezas y debilidades.</li> </ol>

### 3.4. DSDM (DYNAMIC SYSTEMS DEVELOPMENT METHOD)

#### 3.4.1. INTRODUCCIÓN AL MODELO

Método de desarrollo de sistemas dinámicos (en inglés Dynamic Systems Development Method o DSDM) es un método que provee un framework para el desarrollo ágil de software, implica al usuario en un desarrollo iterativo y creciente que sea sensible a los requerimientos cambiantes. (Piña, 2015)

En la Figura 10, se puede observar el diagrama con los procesos involucrados en la metodología DSDM.

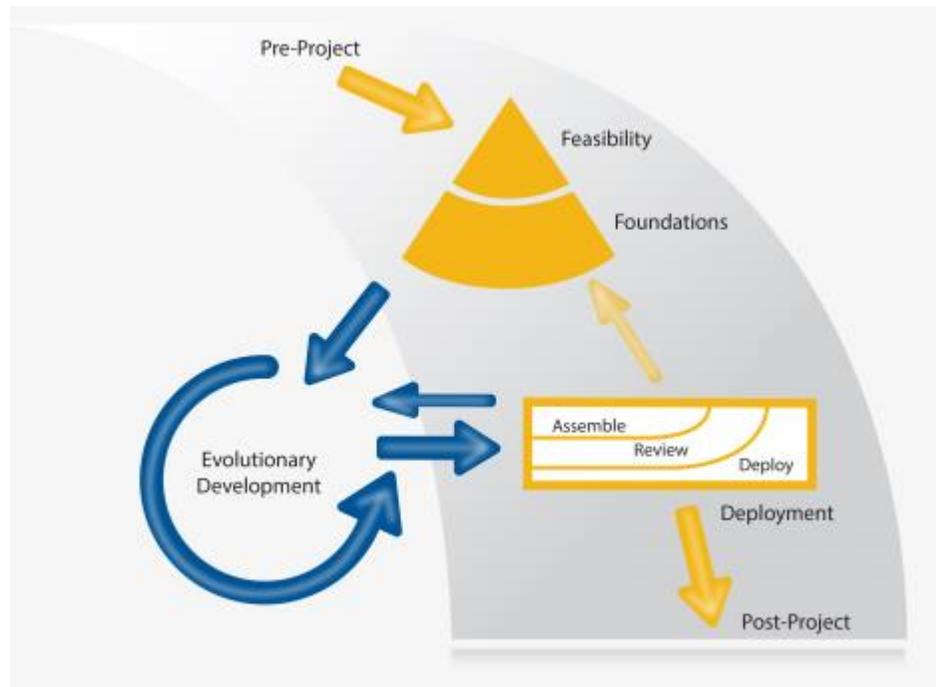


Figura 14: Diagrama de procesos DSDM.

Fuente: (DSDM Consortium, 2014)

#### 3.4.2. ROLES

Hay algunas funciones introducidas en el entorno DSDM. Es importante que los miembros del proyecto tengan que ser nombrados para diferentes

funciones antes de comenzar a ejecutar el proyecto. Cada función tiene su propia responsabilidad.

Las funciones son:

**Patrocinador Ejecutivo:** Un papel importante en la organización de usuarios que tiene la capacidad y la responsabilidad de comprometer fondos y recursos apropiados. Es responsable de la rentabilidad de la inversión.

**Visionario:** El que tiene la responsabilidad de inicializar el proyecto, asegurando que los requisitos se encuentran claros desde el principio. Otra tarea es supervisar y mantener el proceso de desarrollo en el camino correcto.

**Embajador (usuario):** trae el conocimiento de la comunidad de usuarios en el proyecto, asegura que los desarrolladores reciban suficiente cantidad de evaluaciones del usuario durante el proceso de desarrollo.

**Asesor de usuario:** puede ser cualquier usuario que representa un punto de vista importante y trae el conocimiento cotidiano del proyecto.

**Jefe de proyecto:** puede ser cualquier persona de la comunidad de usuarios o al personal de TI que gestiona el proyecto en general.

**Coordinador Técnico:** responsable en el diseño de la arquitectura del sistema y controlar la calidad técnica en el proyecto.

**Líder del equipo:** lleva a su equipo al desarrollo y consecución del proyecto, además asegura que el trabajo sea realizado de manera efectiva entre desarrolladores y usuarios.

**Solution Developer:** es el encargado de interpretar los requerimientos del sistema, determinar la herramienta adecuada para cumplir los requisitos, definir los entregables y construcción de los prototipos.

**Solución Tester:** Comprueba el funcionamiento del sistema mediante la realización de pruebas, notifica al líder del equipo las fallas o errores encontrados. Una vez que hayan sido corregidos los inconvenientes se prueba nuevamente el sistema en busca de más desperfectos. Todas las actividades que este rol desempeña deberán quedar formalmente documentadas.

**Scribe:** responsable de recopilar y registrar los requisitos, acuerdos y decisiones tomadas en todos los talleres. Los talleres son reuniones periódicas en las que se evalúan los avances del proyecto.

**Facilitador:** Responsable en la gestión de los avances de talleres, actúa como un motor para la preparación y la comunicación entre la parte técnica y los usuarios. (Rosale, 2012)

### 3.4.3. ENTREGABLES

**Prototipo de prueba:** Un prototipo se emplea a modo de prueba antes de proceder a la producción del mismo. Básicamente sirve para que sus desarrolladores puedan advertir eventuales fallas en el funcionamiento y descubrir falencias antes de continuar con la próxima actividad de la fase FMI (Modelo Funcional de Iteraciones).

**Prototipo de la opinión:** Recopila los comentarios del usuario y la documentación de acuerdos y trazabilidad generada en cada taller.

**Mantenimiento del sistema:** Con base en prototipos, se actualizará la lista de requisitos priorizados y evaluará los riesgos de posibles cambios en el sistema. Es muy importante para establecer si es factible o no otra iteración de la fase FMI.

**Listado de proyectos candidatos:** Es un listado de proyectos priorizados que se plantean inicialmente como solución a un problema o necesidad.

**Contrato del proyecto:** Es el convenio entre las partes que requieren un determinado servicio y el proveedor de los mismos. En este contrato se firman acuerdos de tiempo, costos y necesidades a satisfacer.

**Fases iniciales del sistema:** Es la recopilación y entendimiento de los requerimientos del cliente, análisis de los mismos y creación de una propuesta técnica para satisfacer necesidades.

**Sistema funcional inicial:** Es la primera entrega del sistema funcional. Es un modelo muy básico pero que permite continuar mejorando con cada iteración.

**Sistema funcional con sus respectivas pruebas:** Evaluación del sistema para comprobar su efectividad y que el funcionamiento sea el requerido por el cliente.

**Entrega de documentación final y sistema funcional:** Documentación actualizada con los requerimientos iniciales, análisis, tareas, diagramas y resultados de pruebas.

El sistema está completamente funcional de acuerdo a los requerimientos del cliente.

### 3.4.4. PROCESO DSDM

Tabla 3  
Proceso DSDM.

DSDM					
FASES	SUB FASES	ACTIVIDADES	ROLES	HERRAMIENTAS	PRODUCTOS
Fase 1: El Pre-proyecto		1. Identificación de proyectos candidatos.	Usuario Asesor de usuario Jefe de proyecto Coordinador Técnico Solution Developer Scribe	Procesadores de Texto y Hojas de Cálculo.	Listado de proyectos candidatos.
		2. Se analiza los fondos y recursos que serán utilizados.	Patrocinador Ejecutivo Scribe	No Especifica	Documento en el que constan los fondos y recursos destinados para el desarrollo proyecto.
		3. Se realiza el compromiso del proyecto.	Patrocinador Ejecutivo Visionario Jefe de proyecto Scribe.	Procesadores de Texto	Contrato del proyecto
Fase 2: Ciclo de Vida del Proyecto	Estudio de viabilidad.	1. Recopilación y entendimiento de los requerimientos del cliente.	Visionario Usuario Coordinador Técnico Solution Developer Scribe	Software para gestión de proyectos.	Documentación de los requisitos.  Continúa 

	2. Evaluar si el proyecto se ajusta a esta metodología de desarrollo.	Visionario Jefe de proyecto Coordinador Técnico Líder del equipo Solution Developer	Procesadores de Texto y Software para gestión de proyectos.	Prototipo de la opinión.
	3. Establecer una solución técnica al problema de negocio.	Visionario Coordinador Técnico Solution Developer	Procesadores de Texto y Software para gestión de proyectos.	Documento que respalda la solución al problema.
	4. Obtener un estimado inicial en costos y tiempo.	P Visionario Jefe de proyecto Coordinador Técnico Líder del equipo Facilitador.	Hojas de Cálculo y Software para gestión de proyectos.	Cronograma y presupuesto del proyecto.
Estudio de la empresa.	1. Se establecen los requisitos funcionales y la información con el fin de permitir un valor al negocio.	Visionario Usuario Asesor de usuario Coordinador Técnico Scribe	Procesadores de Texto y Hojas de Cálculo.	Documento de análisis detallado de los requisitos.
	2. Se define la arquitectura básica de la aplicación.	Visionario Usuario Coordinador Técnico Scribe	Software para gestión de proyectos.	Prototipo de prueba
Iteración del modelo funcional.	1. Enfocar todos los prototipos del DSDM para evolucionar hacia la aplicación entregable.	Visionario Usuario Jefe de proyecto Coordinador Técnico Solution Developer Scribe	Hojas de Cálculo, CASE para desarrollo de software y Software para gestión de proyectos.	Diagramación del sistema  Continúa 

	Diseño e iteración de la estructura.	1. Se revisa la construcción de prototipos durante la iteración del modelo funcional.	Visionario Usuario Jefe de proyecto Coordinador Técnico Líder del equipo	CASE para desarrollo de software y Software para gestión de proyectos.	Fases iniciales del sistema
	Implementación.	1. Se entrega una versión del sistema con el fin de capacitar al usuario.	Visionario Usuario Jefe de proyecto Líder del equipo	CASE para desarrollo de software.	Sistema funcional inicial
		2. Evaluar detalladamente los documentos de sistema.	Jefe de proyecto Líder del equipo Solution Developer Solución Tester	Procesadores de Texto y Hojas de Cálculo.	Actualización de documentación
Fase 3: Post - proyecto		1. Asegurar que el sistema funcione de manera eficaz y eficiente.	Jefe de proyecto Líder del equipo Solution Developer Solución Tester	CASE para desarrollo de software y Herramientas de gestión de pruebas.	Sistema funcional con sus respectivas pruebas
		2. Se realiza mantenimiento, mejoras y correcciones de acuerdo con los principios DSDM.	Visionario Usuario Coordinador Técnico Solution Developer Scribe Facilitador.	Software para gestión de proyectos y CASE para desarrollo de software.	Mejoras y cambios en el sistema.
		3. Los productos entregables pueden ser refinados, dependiendo del mantenimiento realizado.	Visionario Usuario Líder del equipo Scribe Facilitador.	Procesadores de Texto, Hojas de Cálculo y Software para gestión de proyectos.	Sistema funcional de acuerdo a exigencias del usuario y documentación final actualizada.

### **3.5. FDD (FEATURE DRIVEN DEVELOPMENT)**

#### **3.5.1. ORIGEN**

FDD con sus siglas en inglés Feature Driven Development es una perspectiva ágil para la creación de sistemas. Fue desarrollado por Jeff De Luca y el viejo gurú de la Programación Orientada a Objetos Peter Coad.

La metodología surgió a partir de un proyecto de software de una gran entidad financiera en Singapur en 1997, cuyo resultado solo fue un modelo de objetos complejos y ningún código con funcionamiento. ( Universidad Unión Bolivariana, 2015)

#### **3.5.2. INTRODUCCIÓN AL MODELO**

Como todos los procesos de desarrollo de buen software FDD está construido alrededor de un conjunto básico de "mejores prácticas". Las prácticas solicitadas no son nuevas, pero si esta mezcla particular de ingredientes.

Cada práctica complementa y refuerza a los demás. El resultado es un conjunto mayor que la suma de sus partes; no hay ninguna sola práctica que sustenta todo el proceso. Un equipo podría optar por implementar sólo uno o dos de las prácticas, pero no obtendría el beneficio total que se produce mediante el uso de todo el proceso FDD. (Feature Driven Development - Practices, 2015).

Al contrario de otras metodologías, FDD afirma ser conveniente para el desarrollo de sistemas críticos.

Esta metodología divide a los programadores en dos tipos:

- Dueños de clases.
- Programadores jefes.

### 3.5.3. ROLES

El FDD utiliza los siguientes roles:

**Gerente del Proyecto:** Es el líder administrativo y financiero del proyecto. Una de sus múltiples labores principales es mantener al equipo alejado de distracciones externas permitiendo que se pueda trabajar en las condiciones correctas. El Gerente del Proyecto tiene la última palabra sobre los temas enfocados al alcance, tiempo y personal. (Nebulon Pty. Ltd. , 2012)

**Jefe Arquitecto:** Responsable por el diseño total del sistema, así como de la ejecución de todas las fases del diseño. También tiene la última palabra sobre las decisiones del diseño de todo el sistema. Si así lo amerita el desarrollo.

**Director de Desarrollo:** Lleva diariamente las actividades de desarrollo y resuelve cualquier conflicto que pueda ocurrir con el equipo. Además, este rol incluye la resolución de inconvenientes referidos a los recursos. Las tareas de este rol pueden ser combinadas con las del Jefe Arquitecto o el Gerente del Proyecto.

**Jefe Programador:** Desarrollador con experiencia, el cual participa en diversas etapas del desarrollo: el análisis de los requerimientos, el diseño del proyecto, guiar a pequeños equipos en distintas fases de creación como nuevas funcionalidades, selecciona las funcionalidades en la última fase de FDD, identifica las clases y el propietario de las clases que se necesita en el equipo para la iteración. Ayudado de otros integrantes con su mismo rol resuelve problemas técnicos y de recursos, y además comparte el progreso del equipo durante la semana con todos.

**Propietario de las Clases:** Trabaja bajo la guía del Jefe Programador en las tareas de diseño, codificación, pruebas y documentación. Además, es responsable del desarrollo de las clases que se le asignaron como propias. Para cada iteración los propietarios de las clases participan en la decisión de que clase será incluida en la lista de funcionalidades de la próxima iteración.

**Expertos del dominio:** Los expertos del dominio pueden ser un usuario, un cliente, un sponsor, un analista del negocio o una mezcla de estos. Su tarea es tener el conocimiento de todos los requerimientos del sistema inicial. Este a su vez traspassa el conocimiento a los desarrolladores de manera tal que garantice que estos entreguen un sistema completo.

**Gerente de Dominio:** Lidera al grupo de expertos del dominio y resuelve sus diferencias de opinión referentes a los requerimientos del sistema.

**Equipo de Modelado:** El model team es un equipo permanente formado por expertos del dominio y desarrolladores.

**Equipo de Planificación:** Está formado por el Gerente del Proyecto, el Gerente de Desarrollo y el Jefe Programador.

**Equipo de Características:** Es un equipo de larga vida, de funciones cruzadas, cruz-componente que completa muchas características de extremo a extremo de los clientes - uno por uno. Los equipos de características son un elemento esencial para la ampliación de desarrollo ágil. ( Universidad Unión Bolivariana, 2015)

Es primordial dejar en claro que un mismo miembro del equipo puede ejecutar varios roles y un rol puede ser compartido por varias personas.

### 3.5.4. PROCESOS

FDD tiene dos procesos, el primero con tres actividades y el segundo con dos actividades como lo indica la Figura N° 11.

#### 1. Proceso de Inicio y Planeación

##### a. **Desarrollar un modelo integral:**

- Cuando comienza el desarrollo, los expertos del dominio están al tanto de la visión, el contexto y los requerimientos del sistema a construir.
- Se divide el dominio global en áreas que son analizadas detalladamente.
- Los desarrolladores construyen un diagrama de clases o de objetos por cada área.
- Se construye un modelo global del sistema.

##### b. **Construir una lista de funcionalidades:**

Una funcionalidad es un ítem útil a los ojos del cliente.

- Se elabora una lista de funcionalidades que resuma la funcionalidad general del sistema.
- La lista es elaborada por los desarrolladores y es evaluada por el cliente.
- Se divide la lista en subconjuntos según la afinidad y la dependencia de las funcionalidades.
- La lista es finalmente revisada por los usuarios y los responsables para su validación y aprobación.

##### c. **Planear por funcionalidad:**

- En este punto se procede a ordenar los conjuntos de funcionalidades conforme a su prioridad y dependencia, y se asigna a los programadores jefes.



Las primeras tres actividades agrupan gran parte del tiempo, pero a medida que se avanza en las iteraciones las otras dos van requiriendo más tiempo, y las primeras solo son para el refinamiento del sucesivo reléase.

Las últimas dos actividades se hacen en cada iteración. Cada actividad se divide en tareas y se da un criterio de comprobación.

La Figura 13 detalla cómo se desarrollan las actividades del proceso de construcción.

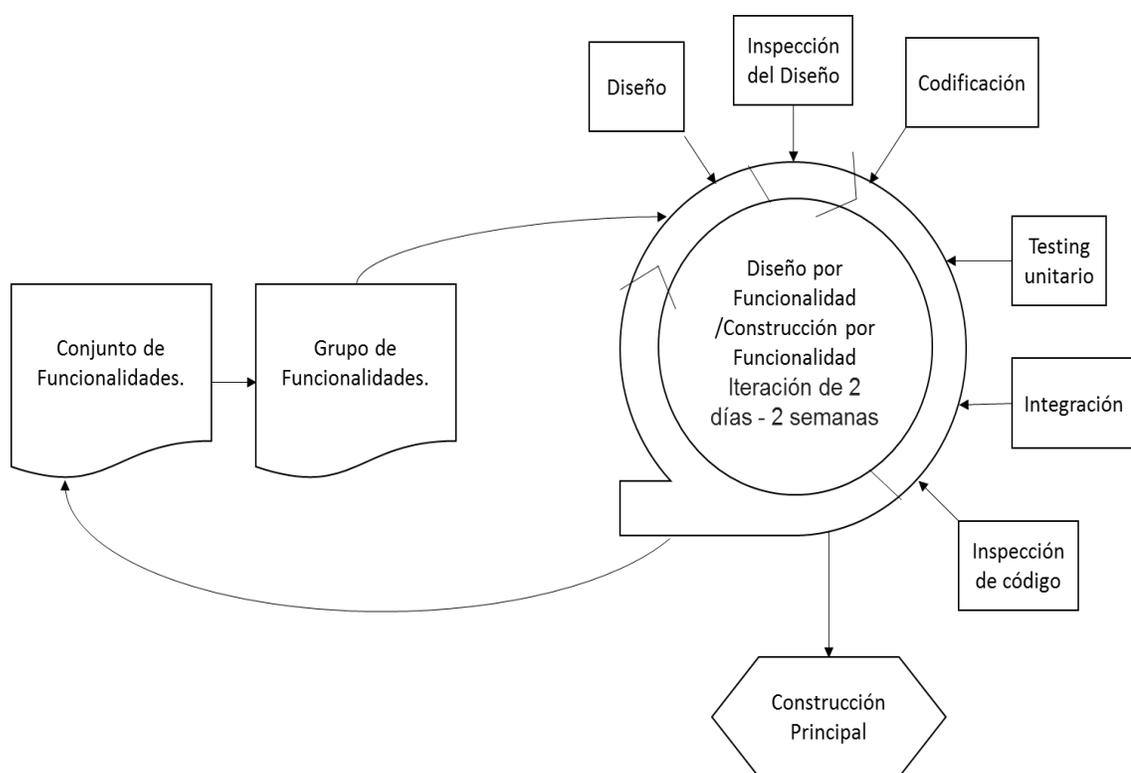
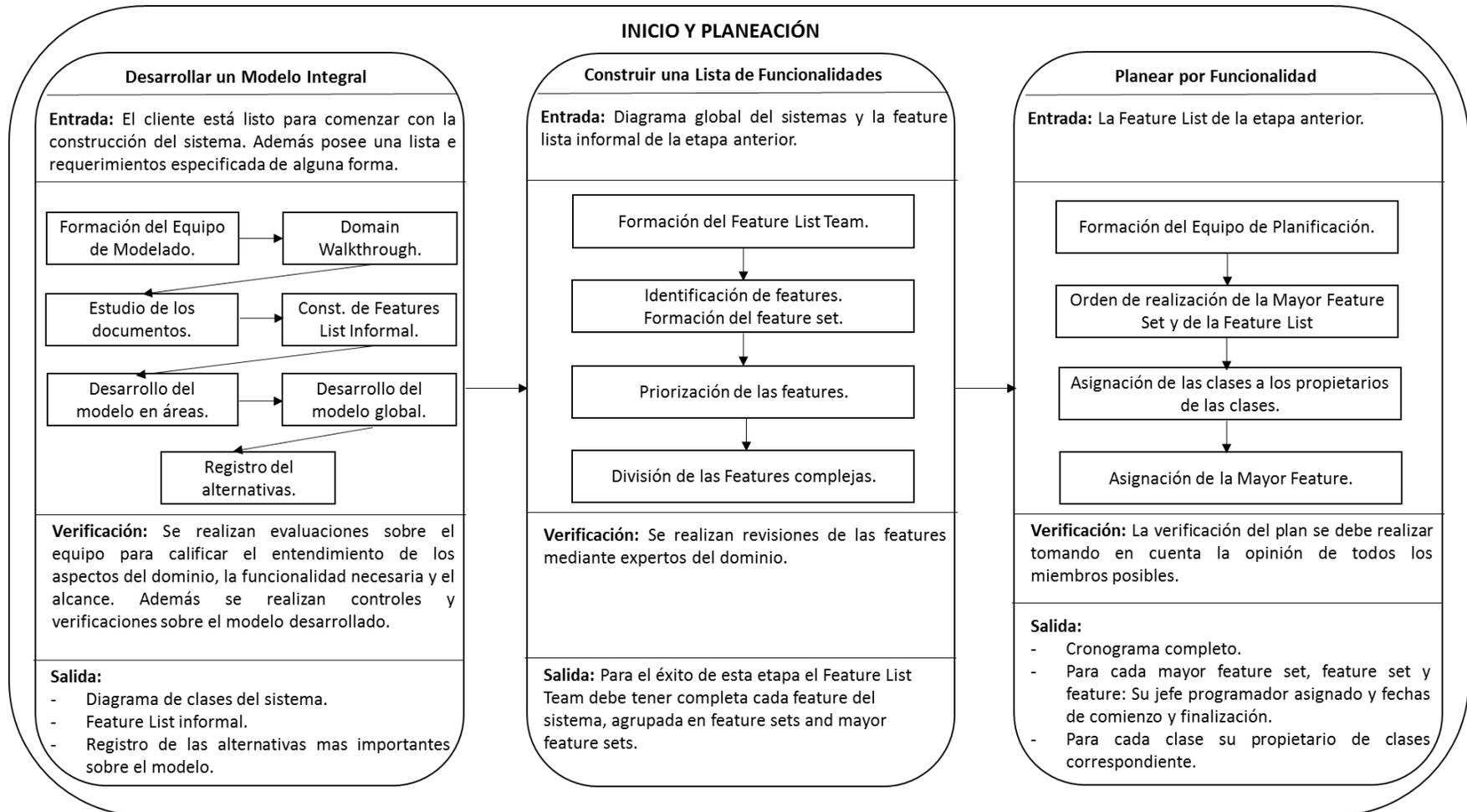


Figura 16: Iteraciones Design and Build by Feature.

Fuente: (Uruguay, 2015)

### 3.5.5. REGLAS DE FDD



Continúa

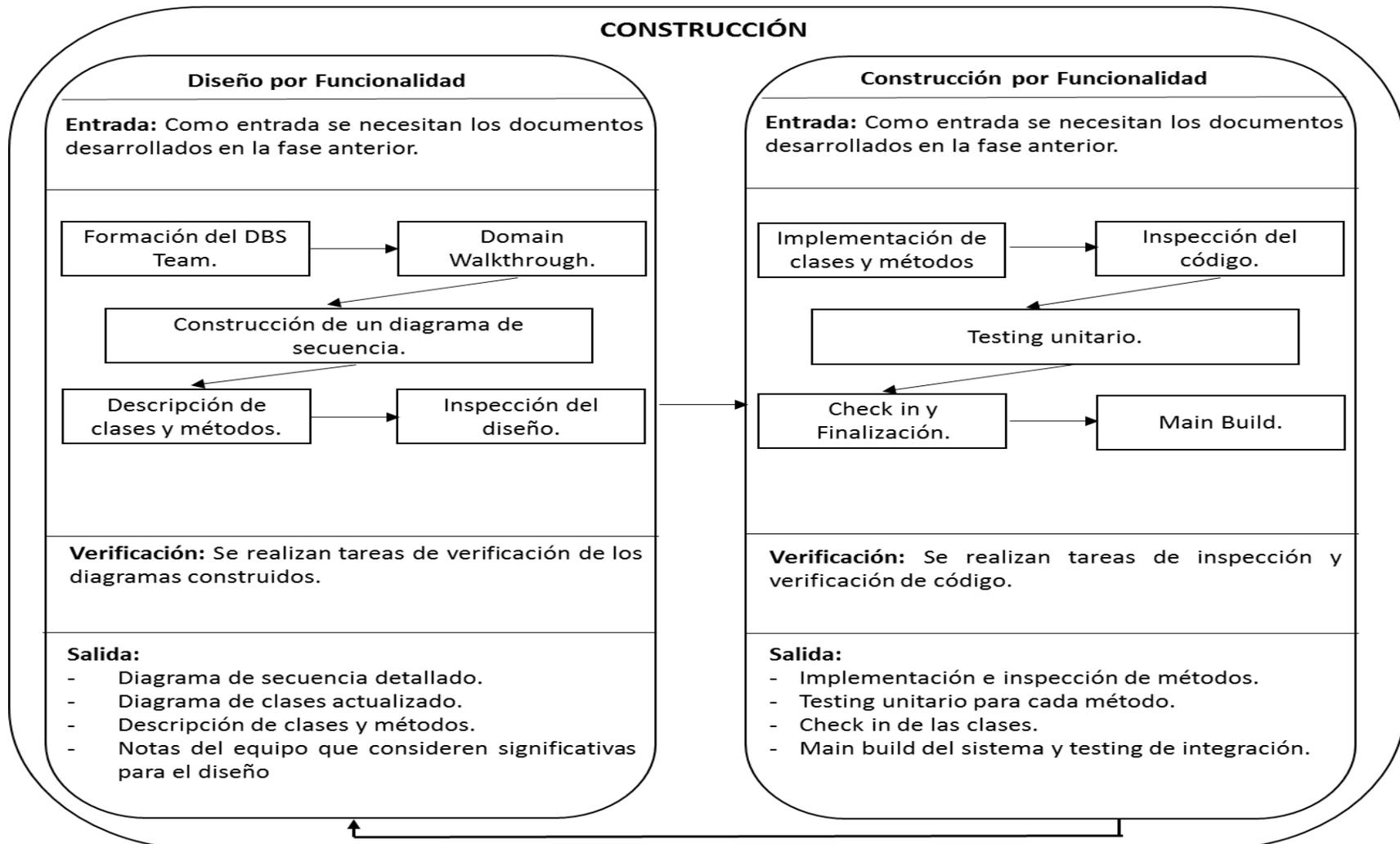


Figura 17: Reglas de FDD.

Fuente: (Uruguay, 2015).

### 3.5.6. PROCESO FDD

Tabla 4  
Proceso FDD.

FDD				
ACTIVIDADES	TAREAS	ROLES	HERRAMIENTAS	PRODUCTOS
Actividad 1: Desarrollar un Modelo Integral.	<b>1. Formación del Equipo de Modelado:</b> Es un equipo plenamente conformado por expertos del dominio y desarrolladores.	Director del Proyecto.	No especifica.	Model Team capacitados.
	<b>2. Domain Walkthrough:</b> Un especialista del dominio efectúa un pequeño tutorial del área a ser modelada.	Equipo de Modelado.	No especifica.	Tutorial del área a ser modelada.
	<b>3. Estudio de los documentos:</b> El equipo averigua los documentos disponibles incluyendo si es necesario el modelo de componentes, los requerimientos funcionales, usuarios, etc.	Equipo de Modelado.	UML.	Informe/Lista de Documentos disponibles en la organización.
	<b>4. Construcción de Feature List informal:</b> El equipo construye una Features List antes de comenzar con la Fase 2 y especifica los artefactos necesarios para la consecución de dicha etapa.	Programador Jefe. Arquitecto Jefe.	No especifica.	Feature list informal. Documentos Necesarios próxima actividad.
	<b>5. Desarrollo del modelo en áreas:</b> El dominio global es dividido en diferentes áreas. Cada subgrupo construye específicamente un esquema para el área de dominio designada, bajo ciertas consideraciones del Chief Architect, haciendo énfasis en las clases y asociaciones, luego en los métodos y finalmente en los atributos. Los subgrupos agregan métodos en base a lo extraído de los walkthrough y de la feature list. Los subgrupos también realizan uno o más esquemas de secuencias de manera informal.	Equipo de Modelado.	UML.	Diagrama de Clases y de secuencia.
				Continúa 

	<p><b>6. Desarrollo del Modelo Global:</b> Cada subgrupo presenta su opción o sugerencia para el área de dominio especificada. El Chief Architect puede proponer una alternativa adicional si así lo desea. El Equipo de Modelado escoge una propuesta como base y se va mejorando con las demás. También se realiza un diagrama de secuencia informal de dicho modelo.</p>	<p>Jefe Arquitecto, Equipo de Modelado.</p>	<p>Procesadores de Texto, UML.</p>	<p>Propuesta base. Diagrama de secuencias informal.</p>
	<p><b>7. Registro de alternativas:</b> Un miembro del equipo registra las alternativas más importantes al modelo que el equipo valoró para referencias futuras en el proyecto.</p>	<p>Jefe Programador, Jefe Arquitecto.</p>	<p>No especifica.</p>	<p>Registros de las alternativas más importantes sobre el modelo.</p>
	<p>1. <b>Formación del Feature-List Team:</b> Consiste en un grupo permanente formado por expertos del dominio y desarrolladores.</p>	<p>Director del proyecto, Director de desarrollo.</p>	<p>No especifica.</p>	<p>Feature-List Team.</p>
	<p><b>2. Identificación de features. Formación del Feature Set:</b> Lista de rasgos informal creada en la etapa anterior. Luego de esto:</p> <ul style="list-style-type: none"> <li>a) Transforman los métodos del modelo a features.</li> <li>b) Transforman funcionalidad común en feature sets.</li> <li>c) Contienen que otras features pueden ser adicionadas para satisfacer los requerimientos del cliente.</li> <li>d) El equipo comienza con la features list.</li> </ul>	<p>Equipo de Modelado.</p>	<p>Software para gestión de proyectos.</p>	<p>Features, features sets.</p>
	<p><b>3. Priorización de las features:</b> Un subgrupo del equipo establece las prioridades de las features:</p> <ul style="list-style-type: none"> <li>a) (debe estar)</li> <li>b) (deseable que esté)</li> <li>c) (deseable si se puede)</li> </ul>	<p>Equipo de Modelado.</p>	<p>Procesadores de Texto, Hojas de Cálculo, Herramienta para priorizar requisitos.</p>	<p>Prioridades de las features. Continúa </p>

Actividad 2: Construir una Lista de Funcionalidades	d) (deseable a futuro)  El equipo considera cada feature en base a la aprobación del cliente (si se incluye la feature) y la desaprobación del cliente (si no es incluida).			
	<b>4. División de las features complejas:</b> Los desarrolladores liderados por el Jefe Arquitecto analizan la situación y el tiempo que cada tarea puede tomar. El equipo divide esas features en otras más detalladas.	Equipo de Modelado, Jefe Arquitecto.	Procesadores de Texto, Hojas de Cálculo, Herramienta para priorizar requisitos, Software para gestión de proyectos.	El equipo divide esas features en otras más pequeñas.
Actividad 3: Planear por Funcionalidad.	1. <b>Formación del Equipo de Planificación:</b> Está formado por el Project Manager, el Development Manager y el Chief Programmer.	Director del Proyecto.	No especifica.	Cronograma completo, Para cada mayor feature set, feature Set y Feature: su Chief Programmer asignado y fechas de comienzo y finalización.
	2. <b>Orden de realización de la Mayor Feature Set y de las feature lists:</b> El Equipo de Planificación determina el orden en el que serán desarrolladas las features y también se establecen los plazos de inicio y de fin de cada tarea.	Equipo de Planificación.	Procesadores de Texto, Hojas de Cálculo, Herramienta para priorizar requisitos, Software para gestión de proyectos.	Orden y plazos de desarrollo de las features.  Continúa 

	<p><b>3. Asignación de las clases a los propietarios de las clases:</b> Usando el orden de desarrollo de las features y la ponderación de las mismas, el Equipo de Planificación asigna clases a los distintos propietarios de clases.</p>	Equipo de Planificación.	Software para gestión de proyectos.	Para cada clase, su propietario de clase correspondiente.
	<p><b>4. Asignación de la Mayor Feature:</b> Usando el mismo criterio que antes el Equipo de Planificación asigna las features a los Chief Programmers.</p>	Equipo de Planificación.	Software para gestión de proyectos.	Asignación de las features a los Chief Programmers.
Actividad 4: Diseño por Funcionalidad	<p><b>1. Formación del DBS Team:</b> El Jefe Programador identifica las clases involucradas en el diseño de la feature. Luego se comunica con los Propietarios de las Clases necesarias y también con los expertos del dominio, si es preciso.</p>	Jefe Programador.	No especifica.	DBS Team Descripción de clases y métodos.
	<p><b>2. Domain Walkthrough:</b> Los expertos del dominio entregan un overview (visión general) del dominio referida a la feature en consideración.</p>	Expertos del dominio.	Software para gestión de proyectos.	Overview.
	<p><b>3. Construcción de un diagrama de secuencia:</b> Aplicando sus conocimientos de las features, el Equipo de Características construye un diagrama de secuencia detallado para la feature. Luego el Jefe Programador agrega el diagrama de secuencia</p>	Equipo de Características, Jefe Programador.	Software para gestión de proyectos., CASE para desarrollo de software.	Diagrama de secuencia detallado.
	<p><b>4. Descripción de clases y métodos:</b> Cada Propietario de las Clases actualiza la descripción de sus clases correspondiente en base al diagrama de secuencia. El incluye tipos de parámetros, tipos de retorno, mensajes envidados, etc.</p>	Propietario de clases.	Software para gestión de proyectos., CASE para desarrollo de software.	Diagrama de clases actualizado.
	<p><b>5. Inspección del diseño:</b> El Equipo de Características realiza una inspección del diseño. El Jefe Programador invita a expertos fuera del equipo para que participen, cuando siente que la complejidad lo requiere.</p>	Equipo de Características, Jefe Programador.	Software para gestión de proyectos., CASE para desarrollo de software.	Notas del equipo que consideren significativas para el diseño. Continúa 

Actividad 5: Construcción por Funcionalidad	<p><b>1. Implementación de clases y métodos:</b> Cada Propietario de las Clases implementa los métodos de sus correspondientes clases en base al diagrama de secuencia. Además, realiza varias pruebas para validar la efectividad de los métodos.</p>	Equipo de Características, Propietario de las Clases.	Software para gestión de proyectos., CASE para desarrollo de software.	Implementación e inspección de métodos, Testing unitario para cada método.
	<p><b>2. Inspección del código:</b> El Jefe Programador establece en el cronograma un reconocimiento del código que puede realizar antes o después del testing unitario. El Equipo de Características traslada dicha inspección y si lo requiere puede pedir la ayuda a expertos externos al equipo.</p>	Equipo de Características, Jefe Programador.	CASE para desarrollo de software y Herramientas de gestión de pruebas.	Inspección del código.
	<p><b>3. Testing unitario:</b> Cada Propietario de las Clases realizar pruebas unitarias de las clases asignadas. El Jefe Programador ayuda al Propietario de las Clases sobre los datos de prueba que deben utilizarse.</p>	Equipo de Características, Propietario de las Clases, Jefe Programador.	CASE para desarrollo de software y Herramientas de gestión de pruebas.	Testing unitario de las clases asignadas.
	<p><b>4. Check in y finalización (de la iteración):</b> Una vez que el código fue implementado, inspeccionado y testeado, cada Propietario de las Clases, si así lo desea puede hacer un Check-in al CMS. Cuando todas las clases están ingresadas el Jefe Programador comunicara acerca de la finalización.</p>	Equipo de Características, Propietario de las Clases, Jefe Programador.	CASE para desarrollo de software, Procesadores de Texto.	Check in de las clases, informar acerca de la finalización del ingreso de todas las clases.
	<p><b>5. Main Build:</b> Luego que la iteración llega a su fin se realiza un main build de la funcionalidad en el cual se integra la característica mejorada. Dicho main build se realiza mientras la siguiente iteración comienza.</p>	Equipo de Características.	CASE para desarrollo de software, Herramientas de gestión de pruebas.	Main build del sistema y testing de integración.

## **3.6. ASD (Adaptive Software Development)**

### **3.6.1. ORIGEN**

La técnica de Adaptive Software Development fue desarrollada por Jim Highsmith y Sam Bayer a comienzos de 1990. Está basada en la adaptación continua a circunstancias cambiantes. En esta metodología no hay un ciclo de planificación-diseño-construcción del software, sino un ciclo especular-colaborar-aprender. (Universidad Unión Bolivariana, 2015)

James Highsmith III, consultor de Cutter Consortium, desarrolló ASD hacia el año 2000 con la finalidad inicial de ofrecer una alternativa a la idea propia de CMM (Capability Maturity Model) Nivel 5 de que la optimización es la única solución frente a los problemas de mayor complejidad. Este método ágil pretende abrir una nueva opción entre el “desarrollo monumental de software” y el “desarrollo accidental”, o entre la burocracia y la adhocracia. Se debería buscar más bien, afirma Highsmith, “el rigor estrictamente necesario”; para ello hay que situarse en coordenadas apenas un poco fuera de la anarquía y practicar menos control que el que se piensa es necesario, el pensador de quien Highsmith toma estas ideas es John Holland, el creador del algoritmo genético y seguramente el investigador actual más importante en materia de procesos emergentes. (Universidad Unión Bolivariana, 2015)

### **3.6.2. DEFINICIÓN**

El método ágil ASD es un modelo de implementación de patrones ágiles para desarrollo de software y se enfoca en aplicar las ideas que se produjeron en el mundo de los sistemas confusos, adaptación continua del proceso al trabajo.

ASD reconoce que los requerimientos del cliente son siempre cambiantes. La iniciación de un proyecto implica definir una misión, establecer las características y las fechas y descomponer el proyecto en una serie de pasos individuales, cada paso puede comprender entre cuatro y ocho semanas. (Universidad Unión Bolivariana, 2015)

### 3.6.3. ENTREGABLES

Los entregables de ASD son:

**Documentación de requerimientos:** Se almacena la información de requerimientos obtenida en reuniones con clientes, de aquí se derivan las iteraciones.

**Cronograma de Actividades:** Puede tratarse de un documento impreso o de una aplicación digital; en cualquier caso, el cronograma incluye una lista de actividades o tareas con las fechas previstas de su comienzo y final. (Definición.de, 2015)

**Documentación de iteraciones:** Se fijan las iteraciones viables que se van a desarrollar a lo largo del proyecto, pueden ser modificadas o eliminadas a través del tiempo.

**Prototipo funcional:** Sirve para ir probando, evaluando y corrigiendo las funcionalidades del proyecto.

**Mantenimiento del prototipo:** Son cambios que el cliente realiza luego que el sistema se encuentre funcional.

**Plan de pruebas:** Se aplican pruebas para evaluar si los resultados obtenidos satisfacen las necesidades del cliente, si se encuentran errores

se notifica al líder del proyecto para que estos sean corregidos en la siguiente iteración.

**Documentación del Proyecto:** Es el respaldo ya sea físico, digital o ambos del desarrollo del proyecto desde el inicio hasta la culminación. Consta de: acuerdos iniciales, formalización, requerimientos, propuestas técnicas de solución, diagramas realizados, resultados de las pruebas y una descripción a modo de manual de usuario que explicará el funcionamiento del proyecto completamente funcional de acuerdo a los requerimientos del cliente.

**Evaluación de resultados obtenidos:** Se analizan los resultados obtenidos para aclarar si es necesaria o no una siguiente iteración. De no ser necesaria una nueva iteración se procede con la finalización de proyecto.

#### 3.6.4. ROLES

Los principales roles de ASD son:

**Cliente:** Es el encargado de definir y transmitir los requerimientos al líder del proyecto. Además, debe estar estrechamente relacionado con todo el equipo de desarrollo durante todo el proceso de creación del nuevo software.

**Líder del Proyecto:** lleva a su equipo al desarrollo y consecución del proyecto, además asegura que el trabajo sea realizado de manera efectiva entre desarrolladores y usuarios del sistema informático

**Programador:** Tiene a cargo la codificación de los componentes a desarrollar en cada iteración, el posee los materiales y lleva a cabo la implementación de los requerimientos capturados. (RIEHLE, 2007)

**Desarrollador:** Tiene a cargo la construcción lógica del software y la continua refinación de la misma en cada iteración. Estará a cargo de la creación de la interfaz del software. (RIEHLE, 2007)

**Tester:** Tiene a su cargo la generación de pruebas al sistema a partir de los requerimientos extraídos y analizados, comprueba el funcionamiento del sistema mediante la realización de pruebas, notifica al líder del equipo las fallas o errores encontrados. Una vez que hayan sido corregidos los inconvenientes se prueba nuevamente el sistema en busca de más desperfectos. Todas las actividades que este rol desempeña deberán quedar formalmente documentadas. (RIEHLE, 2007)

### 3.6.5. FASES

ASD utiliza un cambio enfocado en el ciclo de vida, que consta tres componentes los cuales son: especular, colaborar y aprender.

**Especular:** Es la fase inicial en la que se establecen los principales objetivos y metas del proyecto en conjunto y comprender las limitaciones o riesgos con las que se manejara el proyecto. En ASD se realizan estimaciones de tiempo teniendo en cuenta que pueden sufrir cambios. Pese a esto, son necesarias para la correcta atención del equipo de desarrollo que trabaja dentro de términos de tiempo en forma que puedan priorizar sus tareas.

Se resuelve el número de iteraciones para ser consumidas en el proyecto, prestando atención a las características que pueden ser usadas por el cliente al final de la iteración. Son por tanto necesarios, marcar objetivos prioritarios dentro de las mismas iteraciones.

Estos pasos se pueden volver a examinar varias veces antes de que el equipo y los clientes estén satisfechos con el resultado. Ofrece más espacio para explorar, para darse cuenta que no todo es seguro, permitiendo desviarse del plan sin ningún temor. Muchas veces no seguir del plan original puede ser considerado como un error, más que una oportunidad de aprendizaje, es ahí donde la especulación incita a explorar y a experimentar. Si se admite que no se conoce todo, se está más dispuesto a aprender. (Universidad Unión Bolivariana, 2015)

**Colaborar:** Es la fase donde se centra la mayor parte del desarrollo manteniendo una componente cíclica. La coordinación asegura que lo aprendido por un equipo se transmite al resto y no tenga que volver a ser aprendido por los otros equipos puesto que los conocimientos son heredados entre los miembros.

Las aplicaciones complicadas requieren, la recolección y el análisis de una gran cantidad de información, lo cual no puede ser controlado por una sola persona. A su vez aplicaciones con ambientes cambiantes como las de e-commerce producen un gran flujo de datos, los cuales se pueden manejar por una persona, o un grupo pequeño, ya que estos no pueden saberlo todo. (Rica, 2015)

**Aprender:** La última etapa finaliza con una serie de ciclos de colaboración, su trabajo radica en aprisionar lo que se ha aprendido, tanto positivo como negativo. Es un elemento crítico para la eficacia de los equipos.

Se debe evaluar el conocimiento constantemente realizando retroalimentaciones y reuniones grupales, al final de cada ciclo iterativo,

en lugar de al final del proyecto, ya que esto ayuda a soportar y solucionar de una mejor manera el constante cambio que puede tener el proyecto y su adaptación.

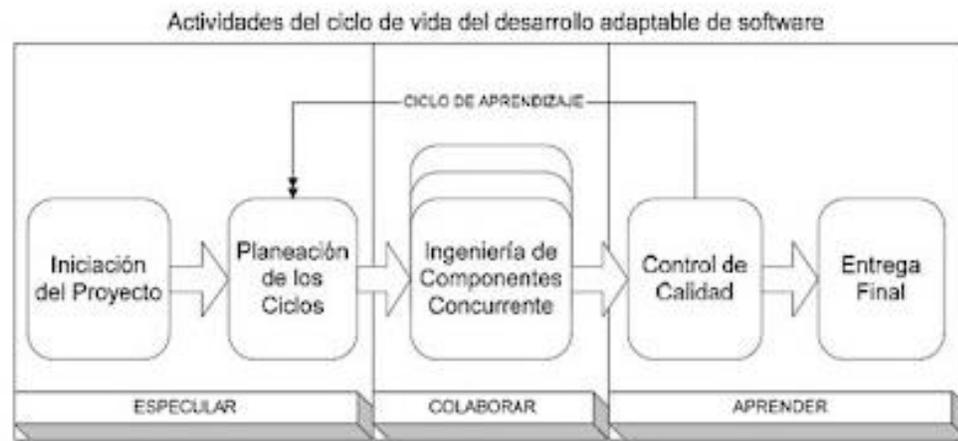


Figura 18: Ciclo de Vida FDD.

(Murillo, 2013)

Jim Highsmith identifica cuatro tipos de aprendizaje en esta etapa:

**Calidad del producto desde un punto de vista del cliente:** Es la única medida legítima de éxito, pero, además, dentro de las metodologías ágiles, los clientes poseen un valor importante.

**Calidad del producto desde un punto de vista de los desarrolladores:** Se trata de evaluar la calidad de los productos desde un enfoque técnico. Ejemplos de esto incluyen el apego a las normas y objetivos conforme a la arquitectura.

**La gestión del rendimiento:** Este es un proceso de evaluación para conocer lo que se ha aprendido, mediante el empleo de los métodos utilizados por el equipo.

**Situación del proyecto.** Como paso previo a la planificación de la siguiente iteración del proyecto, es el punto de partida para la construcción de la siguiente serie de características. (Rica, 2015)

### 3.6.6. PROCESO DE ASD

Tabla 5  
Proceso de ASD.

FASES	ACTIVIDADES	ROLES	HERRAMIENTAS	PRODUCTOS
FASE I ESPECULAR	<b>1. Inicio para determinar la misión del proyecto:</b> Se especula y se adapta los requerimientos para el producto que el cliente necesita.	Cliente Líder del Proyecto Desarrolladores	Procesadores de Texto, Hojas de Cálculo, Herramienta para requisitos.	Documentación de requerimientos.
	<b>2. Fijación del marco temporal del proyecto:</b> Tener una proyección o estimación del proyecto que se va a realizar.	Cliente Líder del Proyecto	CASE para desarrollo de software.	Cronograma de Actividades.
	<b>3. Determinación del nº de iteraciones y la duración de cada una:</b> Analizar factores críticos y esenciales para asignar prioridades en actividades de la iteración.	Cliente Líder del Proyecto Programador Desarrollador	Hojas de Cálculo, CASE para desarrollo de software	Documentación de iteraciones.
	<b>4. Definición del objetivo de cada iteración.</b> - Fijar un objetivo específico a cada iteración.	Cliente Líder del Proyecto Programador Desarrollador	Procesadores de Texto	Priorización de iteraciones.
	<b>5. Asignación de funcionalidad a cada iteración:</b> Tener claro la funcionalidad y beneficio que se obtendrá en cada iteración.	Programador Desarrollado	Hojas de Cálculo, CASE para desarrollo de software	No Específica.  Continúa 

FASE II COLABORAR	<b>1. Construir las funcionalidades a ser desarrolladas durante la iteración.</b>	Líder del Proyecto Programador Desarrolladores Tester	CASE para desarrollo de software.	Prototipo funcional.
	<b>2. Se exploran nuevas alternativas de solución al proyecto.</b>	Cliente Líder del Proyecto Programador Desarrolladores	CASE para desarrollo de software.	Mantenimiento del prototipo.
FASE III APRENDER	<b>1. Calidad del producto desde un punto de vista del cliente.</b> Se considera la única medida veraz de éxito puesto que los clientes tienen un valor importante.	Líder del Proyecto Programador Desarrolladores Tester	Herramientas de gestión de pruebas.	Plan de pruebas
	<b>2. Calidad del producto desde un punto de vista de los desarrolladores.</b> Se trata de la estimación de la calidad de los productos desde un enfoque técnico.	Programador Desarrolladores Tester	Herramientas de gestión de pruebas.	Plan de pruebas
	<b>3. La gestión del rendimiento.</b> Este es un proceso de evaluación del conocimiento adquirido en el proyecto.	Líder del Proyecto Programador Desarrolladores Tester	CASE para desarrollo de software.	Documentación del Proyecto.
	<b>4. Situación del proyecto.</b> Como paso previo a la organización de la siguiente iteración del proyecto.	Cliente Líder del Proyecto Programador Desarrolladores	Procesadores de Texto, Hojas de Cálculo, Herramienta para requisitos.	Evaluación de resultados obtenidos.

### 3.7. DEFINICIÓN DE CARACTERÍSTICAS DE EVALUACIÓN

La ISO 9126 (Sicilia, 2016) es un estándar internacional para la evaluación del Software. Está supervisado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos. El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente:

- modelo de calidad,
- métricas externas,
- métricas internas y
- calidad en las métricas de uso.

Para este caso se utilizará el modelo de calidad establecido en la primera parte del estándar, ISO 9126-1 cuyo estándar ha sido desarrollado en un intento de identificar los atributos clave de calidad para el software. El estándar identifica 6 atributos clave de calidad:

**Funcionalidad.** - El grado en que el software satisface las necesidades indicadas por los siguientes subatributos:

- Idoneidad
- Corrección
- Interoperabilidad
- Conformidad
- Seguridad

**Fiabilidad.** - Cantidad de tiempo que el software está disponible para su uso. Está referido por los siguientes subatributos:

- Madurez
- Tolerancia a fallos
- Facilidad de recuperación

**Usabilidad.** - Grado en que el software hace óptimo el uso de los recursos del sistema. Está indicado por los siguientes subatributos:

- Facilidad de comprensión
- Facilidad de aprendizaje
- Operatividad

**Eficiencia.** - Grado en que el software hace óptimo el uso de los recursos del sistema. Está indicado por los siguientes subatributos:

- Tiempo de uso
- Recursos utilizados

**Mantenibilidad.** - Facilidad con que una modificación puede ser realizada. Está indicada por los siguientes subatributos:

- Facilidad de análisis
- Facilidad de cambio
- Estabilidad
- Facilidad de prueba

**Portabilidad.** - La facilidad con que el software puede ser llevado de un entorno a otro. Está referido por los siguientes subatributos:

- Facilidad de instalación
- Facilidad de ajuste
- Facilidad de adaptación al cambio

La ISO 9126 distingue entre fallos y no conformidad, siendo un fallo el no cumplimiento de los requisitos previos, mientras que la no conformidad

afecta a los requisitos especificados. Una distinción similar es hecha entre la validación y la verificación.

El estándar provee un entorno para que las organizaciones definan un modelo de calidad para el producto software, sin embargo, cada organización tiene tarea de especificar su propio modelo. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos de calidad.

Este estándar está pensado para los desarrolladores, adquirentes, personal que asegure la calidad y evaluadores independientes, responsables de especificar y evaluar la calidad del producto software.

Por tanto, puede servir para validar la completitud de una definición de requisitos, identificar requisitos de calidad de software, objetivos de diseño y prueba, criterios de aseguramiento de la calidad, etc.; por tales razones la ISO-9126 será aplicada en el desarrollo del último objetivo de este trabajo.

Para el análisis de las cinco metodologías de desarrollo se ha tomado en cuenta las capas de la ingeniería de software, así como también se ha definido algunas características, las cuáles han sido las más comunes de las cinco metodologías de desarrollo del caso de estudio, así como también algunas características que debe tener toda metodología ágil de desarrollo de software.

A pesar de la variedad de propuestas de proceso de software, existe un conjunto de actividades fundamentales que se encuentran presentes en todos ellos:

- Requisitos de Software
- Diseño

- Construcción
- Validación y Verificación
- Despliegue
- Roles
- Ciclo de vida
- Tiempo de Interactividad
- Número máximo de personas en la metodología

Tabla 6  
Indicadores de evaluación.

<b>Característica</b>	<b>Porcentaje</b>	<b>Descripción</b>
<b>Define Entregables.</b>	20%	Indica que si la metodología genera algún producto entregable (Gráficos, diagramas, documentos, etc.).
<b>Fases bien definidas.</b>	15%	Indica que si sus fases pueden ser entendidas y aplicadas desde el inicio hasta el final del proceso.
<b>Roles.</b>	10%	Función que debe cumplir cada uno de los integrantes del equipo de desarrollo.
<b>Técnicas.</b>	10%	Se refiere a que, si la metodología maneja, emplea o indica que métodos utiliza en todo su proceso. Ejemplo: Lluvia de ideas.
<b>Herramientas.</b>	10%	Se refiere a que, si la metodología maneja, emplea o indica que instrumentos utiliza en todo su proceso. Ejemplo: UML.
<b>Define criterios de calidad.</b>	10%	Indica que si toman en cuenta aspectos funcionales y no funcionales esenciales para garantizar la calidad en el software. Deben ser medibles y además es necesario que sean incorporados desde el inicio del proyecto.
<b>Núm. Personas (mín. 4 - máx. 8).</b>	10%	Indica el número mínimo y máximo de personas que debe tener la metodología.
<b>Tiempo Interactividad hasta 6 semanas.</b>	10%	Se refiere al tiempo que dura una iteración en la metodología de desarrollo.
<b>Ciclo de vida.</b>	5%	Describe el desarrollo de software, desde la fase inicial hasta la fase final para garantizar que el software cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo.
<b>TOTAL</b>	100%	

### 3.7.1. ANÁLISIS

### 3.7.2. CUADRO COMPARATIVO

Tabla 7  
Cuadro comparativo de las metodologías de desarrollo de estudio

Característica	Sub Característica	Indicadores a evaluar	XP	%	SCRUM	%	DSDM	%	FDD	%	ASD	%
<b>Fases bien definidas (15%)</b>												
	<b>Requisitos de Software (5%)</b>		<b>6</b>	<b>4.10%</b>	<b>6</b>	<b>4.50%</b>	<b>6</b>	<b>4.30%</b>	<b>5</b>	<b>3.60%</b>	<b>7</b>	<b>4.50%</b>
	Es el proceso de investigar las condiciones y capacidades requeridas que deben cumplirse o estar presentes en el sistema o subsistema para satisfacer un contrato, estándar, una especificación u otro documento. (IEEE)	Obtiene información sobre el sistema a desarrollarse. (1%)	1	1.00%	1	1.00%	1	1.00%	1	0.80%	1	1.00%
		Prepara y realiza sesiones de elicitación/negociación. (0.5%)	1	0.50%	1	0.50%	1	0.30%	0	0.00%	1	0.50%
		Identifica los objetivos del sistema. (1%)	1	1.00%	1	1.00%	1	1.00%	1	0.80%	1	1.00%
		Identifica los requisitos funcionales (RF). (0.5%)	1	0.30%	1	0.50%	1	0.50%	1	0.50%	1	0.40%
		Identifica los requisitos no funcionales (RNF). (0.5%)	1	0.30%	1	0.50%	1	0.50%	1	0.50%	1	0.30%
		Prioriza objetivos, RF y RNF. (1%)	1	1.00%	1	1.00%	1	1.00%	1	1.00%	1	0.80%
		Define técnicas de elicitación. (0.5%)	0	0.00%	0	0.00%	0	0.00%	0	0.00%	1	0.50%
	<b>Diseño (3%)</b>		<b>2</b>	<b>1.25%</b>	<b>2</b>	<b>1.50%</b>	<b>4</b>	<b>2.15%</b>	<b>4</b>	<b>3.0%</b>	<b>4</b>	<b>1.90%</b>
	Es el proceso de utilizar la información recolectada en la etapa de análisis al diseño del producto. La principal tarea de la etapa de	Diseña la arquitectura física del sistema. (1%)	1	0.75%	0	0.00%	1	0.75%	1	1.00%	1	0.80%
		Diseña la estructura física de datos del sistema. (1%)	1	0.50%	1	1.00%	1	0.60%	1	1.00%	1	0.40%
		Especifica el entorno tecnológico del sistema. (0.5)	0	0.00%	1	0.50%	1	0.40%	1	0.50%	1	0.30%

diseño es desarrollar un modelo o las especificaciones para el producto.	Completa todas las especificaciones de diseño. (0.5)	0	0.00%	0	0.00%	1	0.40%	1	0.50%	1	0.40%
<b>Construcción (3%)</b>											
Consiste en utilizar los modelos creados durante la etapa de diseño para crear los componentes del sistema.	Prepara entorno de desarrollo. (0.5)	1	0.50%	1	0.50%	1	0.40%	1	0.50%	1	0.40%
	Prepara entorno de pruebas. (0.25)	1	0.20%	1	0.25%	1	0.20%	1	0.25%	1	0.10%
	Prepara entorno de procedimientos de operación. (0.25)	1	0.20%	1	0.25%	1	0.15%	1	0.25%	1	0.10%
	Desarrolla los componentes del sistema. (0.25)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%
	Prueba los componentes del sistema. (0.25)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%
	Realiza pruebas de integración. (0.5)	1	0.50%	1	0.50%	1	0.50%	1	0.50%	1	0.40%
	Elaboración de Manuales de administración. (0.5)	1	0.20%	0	0.00%	1	0.50%	1	0.50%	1	0.50%
	Elaboración de Manuales de usuario. (0.5)	1	0.20%	0	0.00%	1	0.50%	0	5.00%	1	0.50%
<b>Validación y Verificación (2%)</b>		<b>3</b>	<b>1.35%</b>	<b>3</b>	<b>1.50%</b>	<b>3</b>	<b>1.10%</b>	<b>4</b>	<b>1.5%</b>	<b>3</b>	<b>0.95%</b>
Consiste en comprobar que los requisitos documentados correspondan con las necesidades de los clientes y usuarios (Validación) y comprobar que la especificación cumpla con los criterios de calidad oportunos (Verificación).	Diseña las pruebas del sistema. (0.5%)	1	0.35%	1	0.50%	1	0.25%	1	0.50%	1	0.20%
	Realizar las pruebas del sistema. (0.5%)	1	0.50%	1	0.50%	1	0.50%	1	0.50%	1	0.40%
	Realizar las pruebas de certificación. (0.5%)	0	0.00%	1	0.50%	0	0.00%	1	0.50%	0	0.00%
	Establecer procedimientos de producción. (0.5%)	1	0.50%	0	0.00%	1	0.35%	1	0.00%	1	0.35%
<b>Despliegue (2%)</b>		<b>6</b>	<b>1.50%</b>	<b>4</b>	<b>1.25%</b>	<b>6</b>	<b>1.45%</b>	<b>5</b>	<b>1.50%</b>	<b>6</b>	<b>1.50%</b>
Consiste en poner a disposición del cliente el producto.	Instalación y configuración en el servidor de producción. (0.5)	1	0.30%	1	0.50%	1	0.30%	1	0.50%	1	0.30%
	Capacitaciones a administradores y usuarios. (0.5)	1	0.20%	0	0.00%	1	0.20%	0	0.00%	1	0.20%

		Poner en producción el software. (0.25%)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%
		Ejecutar pruebas unitarias, de integración. (0.25%)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%
		Socializar de Manuales técnicos, administración y de usuario. (0.25%)	1	0.25%	0	0.00%	1	0.20%	1	0.25%	1	0.25%
		Depuraciones. (0.25%)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%
<b>TOTAL</b>			<b>25</b>	<b>10.50%</b>	<b>21</b>	<b>10.75%</b>	<b>27</b>	<b>11.75%</b>	<b>25</b>	<b>17.10%</b>	<b>28</b>	<b>11.35%</b>
<b>Ciclo de vida (5%)</b>												
	<b>Iterativo (1.5%)</b>		3	1.30%	3	1.50%	3	1.10%	3	1.50%	3	1.30%
	Se centra en mejorar y revisar el producto ya creado.	Prioriza requerimientos en cada iteración. (0.5%)	1	0.40%	1	0.5%	1	0.40%	1	0.50%	1	0.50%
		Permite ser evaluado constantemente. (0.5%)	1	0.40%	1	0.5%	1	0.30%	1	0.50%	1	0.30%
		Cada iteración cumple un objetivo. (0.5%)	1	0.50%	1	0.5%	1	0.40%	1	0.50%	1	0.50%
	<b>Incremental (1.5%)</b>		3	1.20%	3	1.50%	3	1.10%	3	1.50%	3	1.00%
	Se centra en desarrollar por partes el producto software, para después integrarlas a medida que se completan.	Acopla las nuevas mejoras a la iteración anterior. (0.5%)	1	0.40%	1	0.5%	1	0.30%	1	0.50%	1	0.30%
		Realiza una mini cascada de desarrollo de cada uno de los incrementos del sistema. (0.5%)	1	0.30%	1	0.5%	1	0.30%	1	0.50%	1	0.20%
		El conjunto de incrementos es el producto final. (0.5%)	1	0.50%	1	0.5%	1	0.50%	1	0.50%	1	0.50%
	<b>Iterativo - Incremental (2%)</b>		3	2.00%	3	2.00%	3	1.80%	2	1.50%	3	1.90%
	Al final de cada iteración se consigue una versión estable del software, añadiendo además nuevas funcionalidades a las versiones anteriores.	Agrega nuevas funcionalidades en relación a versiones anteriores. (0.5%)	1	0.50%	1	0.5%	1	0.40%	1	0.50%	1	0.40%
		Al final de cada iteración muestra un resultado funcional. (0.5%)	1	0.50%	1	0.5%	1	0.40%	0	0.00%	1	0.50%
		Agrupar todas las mejoras hasta llegar al producto final. (1%)	1	1.00%	1	1.0%	1	1.00%	1	1.00%	1	1.00%
<b>TOTAL</b>			<b>9</b>	<b>4.50%</b>	<b>9</b>	<b>5.00%</b>	<b>9</b>	<b>4.00%</b>	<b>8</b>	<b>4.50%</b> Continúa	<b>9</b> a	<b>4.20%</b> 

Roles (10%)												
	<b>Líder de Proyecto (1.5%)</b>		<b>6</b>	<b>1.35%</b>	<b>6</b>	<b>1.50%</b>	<b>6</b>	<b>1.30%</b>	<b>6</b>	<b>1.50%</b>	<b>6</b>	<b>1.35%</b>
		Tiene una persona que lidere el proyecto (alcance, tiempo, personal) (Planificación, Seguimiento y Gestión) (0.25%)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%
		Protege al equipo de distracciones y permite trabajar en las condiciones apropiadas. (0.25%)	1	0.20%	1	0.25%	1	0.20%	1	0.25%	1	0.25%
		Asegura que el trabajo sea realizado de manera efectiva. (0.25%)	1	0.25%	1	0.25%	1	0.20%	1	0.25%	1	0.25%
		Motiva y organiza el equipo de trabajo para lograr un objetivo definido. (0.25%)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.20%
		Planifica y realiza las reuniones de control del equipo de desarrollo en el tiempo establecido. (0.25%)	1	0.20%	1	0.25%	1	0.20%	1	0.25%	1	0.20%
		Planifica y asigna las tareas de la forma más razonable posible. (0.25%)	1	0.20%	1	0.25%	1	0.20%	1	0.25%	1	0.20%
		<b>Gestor con el Usuario (1%)</b>	<b>4</b>	<b>1.00%</b>	<b>4</b>	<b>1.00%</b>	<b>4</b>	<b>0.80%</b>	<b>4</b>	<b>0.60%</b>	<b>4</b>	<b>0.80%</b>
		Favorece la relación entre usuarios y desarrolladores (0.25%)	1	0.25%	1	0.25%	1	0.20%	1	0.15%	1	0.20%
		Cubre las necesidades del equipo. (0.25%)	1	0.25%	1	0.25%	1	0.20%	1	0.15%	1	0.20%
		Experiencia en manejo y conflictos. (0.25)	1	0.25%	1	0.25%	1	0.20%	1	0.15%	1	0.20%
		Habilidades de comunicación oral y escrita. (0.25)	1	0.25%	1	0.25%	1	0.20%	1	0.15%	1	0.20%
		<b>Especialista en Requisitos (1.5%)</b>	<b>6</b>	<b>1.35%</b>	<b>6</b>	<b>1.45%</b>	<b>6</b>	<b>1.30%</b>	<b>6</b>	<b>1.45%</b>	<b>6</b>	<b>1.25%</b>
		Tiene una persona que sea experta en elicitación de requisitos. (0.25%)	1	0.20%	1	0.20%	1	0.20%	1	0.20%	1	0.20%
		Autoconfianza y Comprensión del dominio. (0.25%)	1	0.25%	1	0.25%	1	0.20%	1	0.25% Continúa	1	0.20% a 

	Uso de especificaciones formales para requerimientos y requisitos (formatos estándar de documentos, UML). (0.25%)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%
	Capacidad para detectar y resolver conflictos. (0.25)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.20%
	Habilidad para convencer. (0.25)	1	0.20%	1	0.25%	1	0.20%	1	0.25%	1	0.20%
	Habilidad para moderar y para empatizar (0.25)	1	0.20%	1	0.25%	1	0.20%	1	0.25%	1	0.20%
<b>Arquitecto de Software (1.5%)</b>		<b>6</b>	<b>1.10%</b>	<b>6</b>	<b>1.20%</b>	<b>5</b>	<b>1.10%</b>	<b>6</b>	<b>1.10%</b>	<b>5</b>	<b>1.10%</b>
	Llevar a cabo el diseño global del sistema. (0.25%)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%
	Definir las vistas de la arquitectura de una aplicación. (0.25%)	1	0.10%	1	0.10%	0	0.10%	1	0.10%	0	0.10%
	Dar soporte técnico-tecnológico a desarrolladores, clientes y expertos en negocios. (0.25%)	1	0.20%	1	0.25%	1	0.15%	1	0.15%	1	0.15%
	Conceptualizar y experimentar con distintos enfoques arquitectónicos. (0.25%)	1	0.10%	1	0.15%	1	0.10%	1	0.10%	1	0.10%
	Crear documentos de modelos y componentes y especificaciones de interfaces. (0.25%)	1	0.20%	1	0.20%	1	0.25%	1	0.25%	1	0.25%
	Validar la arquitectura contra requerimientos. (0.25%)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%
<b>Diseñador de Base de Datos (1.5%)</b>		<b>3</b>	<b>1.40%</b>	<b>3</b>	<b>1.50%</b>	<b>3</b>	<b>1.30%</b>	<b>3</b>	<b>1.50%</b>	<b>3</b>	<b>1.40%</b>
	Habilidad para diseñar y construir la base de datos del sistema. (0.5%)	1	0.40%	1	0.50%	1	0.30%	1	0.50%	1	0.40%
	Habilidad para administra la base de datos. (0.5%)	1	0.50%	1	0.50%	1	0.50%	1	0.50%	1	0.50%
	Una comprensión del lenguaje y del entorno de implementación. (0.5%)	1	0.50%	1	0.50%	1	0.50%	1	0.50%	1	0.50%
<b>Programador (1.5%)</b>		<b>6</b>	<b>1.45%</b>	<b>6</b>	<b>1.50%</b>	<b>6</b>	<b>1.35%</b>	<b>6</b>	<b>1.50%</b>	<b>6</b>	<b>1.35%</b>
	Experiencia en codificación de los componentes a desarrollar. (0.25%)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%

		Interpretar los modelos establecidos en los requerimientos capturados. (0.25%)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%
		Planificar su propio trabajo en el contexto del equipo de desarrollo del proyecto. (0.25%)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%
		Aplicar pruebas unitarias. (0.25%)	1	0.20%	1	0.25%	1	0.20%	1	0.25%	1	0.20%
		Documentar el código para que resulte interpretable y utilizable por quienes lo necesiten. (0.25%)	1	0.25%	1	0.25%	1	0.20%	1	0.25%	1	0.20%
		Alto conocimiento tecnológico y de lenguajes de programación. (0.25%)	1	0.25%	1	0.25%	1	0.20%	1	0.25%	1	0.20%
	<b>Tester (1.5%)</b>		<b>6</b>	<b>1.40%</b>	<b>6</b>	<b>1.35%</b>	<b>6</b>	<b>1.30%</b>	<b>6</b>	<b>1.50%</b>	<b>6</b>	<b>1.35%</b>
		Capacidad de abstracción y modelado para entender y simular el comportamiento del sistema bajo prueba. (0.25%)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%
		Facilidad de comunicación oral y escrita para interactuar con desarrolladores y usuarios. (0.25%)	1	0.25%	1	0.25%	1	0.20%	1	0.25%	1	0.20%
		Creatividad para generar ideas e imaginar los problemas que podrían existir. (0.25%)	1	0.25%	1	0.25%	1	0.20%	1	0.25%	1	0.25%
		Pensamiento crítico para evaluar las ideas, hacer deducciones y vincular lo observado con los criterios de calidad de la empresa. (0.25%)	1	0.20%	1	0.15%	1	0.20%	1	0.25%	1	0.20%
		Generación documentación que respalde las pruebas. (0.25%)	1	0.20%	1	0.20%	1	0.20%	1	0.25%	1	0.20%
		Aptitudes para el trabajo en equipo. (0.25%)	1	0.25%	1	0.25%	1	0.25%	1	0.25%	1	0.25%
		<b>TOTAL</b>	<b>37</b>	<b>9.05%</b>	<b>37</b>	<b>9.50%</b>	<b>36</b>	<b>8.45%</b>	<b>37</b>	<b>9.15%</b>	<b>36</b>	<b>8.60%</b>
<b>Técnicas (10%)</b>												
	<b>Requisitos de Software (2%)</b>		<b>2</b>	<b>1.45%</b>	<b>2</b>	<b>1.50%</b>	<b>2</b>	<b>1.35%</b>	<b>2</b>	<b>1.50%</b> Continú	<b>2</b> a	<b>1.35%</b> ➔

	Realiza reuniones. (0,75%)	1	0.75%	1	0.75%	1	0.75%	1	0.75%	1	0.75%
	Lluvia de ideas. (0,75%)	1	0.70%	1	0.75%	1	0.60%	1	0.75%	1	0.60%
	Casos de uso. (0,50%)	0	0.00%	0	0.00%	0	0.00%	0	0.00%	0	0.00%
<b>Diseño (2%)</b>		<b>3</b>	<b>1.00%</b>	<b>3</b>	<b>1.75%</b>	<b>3</b>	<b>0.95%</b>	<b>2</b>	<b>0.015</b>	<b>3</b>	<b>1.00%</b>
	Se realiza un análisis de la factibilidad técnica. (0,50%)	1	0.40%	1	0.50%	1	0.45%	0	0.00%	1	0.50%
	Diseña interfaces. (0,75%)	1	0.30%	1	0.75%	1	0.25%	1	0.75%	1	0.25%
	Modela base de datos. (0,75%)	1	0.30%	1	0.50%	1	0.25%	1	0.75%	1	0.25%
<b>Construcción (2%)</b>		<b>3</b>	<b>2.00%</b>								
	Crea el prototipo básico funcional. (0,75%)	1	0.75%	1	0.75%	1	0.75%	1	0.75%	1	0.75%
	Realiza constantemente pruebas unitarias del software. (0,75%)	1	0.75%	1	0.75%	1	0.75%	1	0.75%	1	0.75%
	Valida que el producto al final de cada iteración funcione de acuerdo a los requisitos. (0,50%)	1	0.50%	1	0.50%	1	0.50%	1	0.50%	1	0.50%
<b>Validación y Verificación (2%)</b>		<b>3</b>	<b>1.90%</b>	<b>3</b>	<b>2.00%</b>	<b>3</b>	<b>1.70%</b>	<b>3</b>	<b>2.00%</b>	<b>3</b>	<b>1.90%</b>
	Técnicas basadas en la experiencia. (0,50%)	1	0.45%	1	0.50%	1	0.40%	1	0.50%	1	0.40%
	Detecta y corrige requisitos funcionales innecesarios e incorrectos. (0,75%)	1	0.75%	1	0.75%	1	0.70%	1	0.75%	1	0.75%
	Realiza pruebas de aceptación del usuario. (0,75%)	1	0.70%	1	0.75%	1	0.60%	1	0.75%	1	0.75%
<b>Despliegue (2%)</b>		<b>3</b>	<b>1.80%</b>	<b>2</b>	<b>1.50%</b>	<b>3</b>	<b>2.00%</b>	<b>3</b>	<b>2.00%</b>	<b>3</b>	<b>1.80%</b>
	Realiza pruebas de funcionamiento en ambiente de desarrollo. (0,75%)	1	0.75%	1	0.75%	1	0.75%	1	0.75%	1	0.70%
	Realiza pruebas de funcionamiento en ambiente de producción. (0,75%)	1	0.75%	1	0.75%	1	0.75%	1	0.75%	1	0.75%
	Se realiza capacitación del funcionamiento del software. (0,50%)	1	0.30%	0	0.00%	1	0.50%	1	0.50%	1	0.35%
	<b>TOTAL</b>	<b>14</b>	<b>8.15%</b>	<b>13</b>	<b>8.75%</b>	<b>14</b>	<b>8.00%</b>	<b>13</b>	<b>9.00%</b>	<b>14</b>	<b>8.05%</b>
Continúa 											

Herramientas (10%)											
<b>Requisitos de Software (2%)</b>		<b>3</b>	<b>1.95%</b>	<b>3</b>	<b>2.00%</b>	<b>3</b>	<b>1.75%</b>	<b>3</b>	<b>2.00%</b>	<b>3</b>	<b>1.85%</b>
	Realiza entrevistas a principales actores involucrados con el sistema. (0,75%)	1	0.75%	1	0.75%	1	0.75%	1	0.75%	1	0.75%
	Todos los participantes contribuyen con sus ideas a través de una lluvia de ideas. (0,75%)	1	0.70%	1	0.75%	1	0.60%	1	0.75%	1	0.70%
	Realizar reuniones continuas con los usuarios. (0,50%)	1	0.50%	1	0.50%	1	0.40%	1	0.50%	1	0.40%
<b>Diseño (2%)</b>		<b>3</b>	<b>2.00%</b>	<b>3</b>	<b>1.95%</b>	<b>3</b>	<b>1.75%</b>	<b>2</b>	<b>1.00%</b>	<b>3</b>	<b>1.85%</b>
	Utiliza herramientas CASE. (0,50%)	1	0.50%	1	0.50%	1	0.45%	1	0.50%	1	0.50%
	Herramientas de planificación. (0,75%)	1	0.75%	1	0.75%	1	0.70%	1	0.50%	1	0.75%
	Herramientas de seguimiento de requisitos. (0,75%)	1	0.75%	1	0.70%	1	0.60%	0	0.00%	1	0.60%
<b>Construcción (2%)</b>		<b>2</b>	<b>1.30%</b>	<b>2</b>	<b>1.50%</b>	<b>2</b>	<b>1.05%</b>	<b>2</b>	<b>1.50%</b>	<b>2</b>	<b>1.30%</b>
	Permite el uso de lenguajes de 4ta generación. (0,5%)	1	0.30%	1	0.50%	1	0.30%	1	0.50%	1	0.30%
	Establece un Plan de Pruebas del Software. (1%)	1	1.00%	1	1.00%	1	0.75%	1	1.00%	1	1.00%
	Define el lenguaje de programación. (0,5%)	0	0.00%	0	0.00%	0	0.00%	0	0.00%	0	0.00%
<b>Validación y Verificación (2%)</b>		<b>3</b>	<b>1.40%</b>	<b>2</b>	<b>0.30%</b>	<b>3</b>	<b>1.70%</b>	<b>1</b>	<b>0.50%</b>	<b>3</b>	<b>1.75%</b>
	Herramientas de gestión. (0,50%)	1	0.30%	1	0.20%	1	0.40%	1	0.50%	1	0.40%
	Herramientas de medida. (0,75%)	1	0.50%	1	0.10%	1	0.70%	0	0.00%	1	0.70%
	Herramientas de soporte. (0,75%)	1	0.60%	0	0.00%	1	0.60%	0	0.00%	1	0.65%
<b>Despliegue (2%)</b>		<b>3</b>	<b>1.95%</b>	<b>2</b>	<b>1.50%</b>	<b>3</b>	<b>2.00%</b>	<b>3</b>	<b>2.00%</b>	<b>3</b>	<b>1.75%</b>
	Ambiente de desarrollo. (0,75%)	1	0.75%	1	0.75%	1	0.75%	1	0.75%	1	0.70%
	Ambiente de producción. (0,75%)	1	0.75%	1	0.75%	1	0.75%	1	0.75%	1	0.75%
	Curso de capacitación a usuarios. (0,50%)	1	0.45%	0	0.00%	1	0.50%	1	0.50% Continú	1 a	0.30% ➔

		<b>TOTAL</b>		<b>14</b>	<b>8.60%</b>	<b>12</b>	<b>7.25%</b>	<b>14</b>	<b>8.25%</b>	<b>11</b>	<b>7.00%</b>	<b>14</b>	<b>8.50%</b>
<b>Define Entregables (20%)</b>													
	<b>Requisitos de Software (2.5%)</b>		<b>2</b>	<b>2.50%</b>	<b>2</b>	<b>2.50%</b>	<b>2</b>	<b>2.25%</b>	<b>2</b>	<b>2.50%</b>	<b>2</b>	<b>2.50%</b>	
		Documento de requisitos del sistema (1,25%)	1	1.25%	1	1.25%	1	1.25%	1	1.25%	1	1.25%	
		Documento de requisitos del software. (1,25%)	1	1.25%	1	1.25%	1	1.00%	1	1.25%	1	1.25%	
	<b>Diseño (2.5%)</b>		<b>2</b>	<b>1.00%</b>	<b>2</b>	<b>1.00%</b>	<b>2</b>	<b>1.00%</b>	<b>4</b>	<b>2.00%</b>	<b>2</b>	<b>1.00%</b>	
		Diagramas de estado. (0,5%)	0	0.00%	0	0.00%	0	0.00%	0	0.00%	0	0.00%	
		Diagramas de interacción (secuencia y colaboración). (0,5%)	0	0.00%	0	0.00%	0	0.00%	1	0.50%	0	0.00%	
		Diagramas de actividad. (0.5%)	0	0.00%	0	0.00%	0	0.00%	1	0.50%	0	0.00%	
		Diagrama de clases. (0.5%)	1	0.50%	1	0.50%	1	0.50%	1	0.50%	1	0.50%	
		Plan de pruebas de programas. (0.5%)	1	0.50%	1	0.50%	1	0.50%	1	0.50%	1	0.50%	
	<b>Construcción (5%)</b>		<b>1</b>	<b>1.50%</b>	<b>1</b>	<b>1.00%</b>	<b>1</b>	<b>1.50%</b>	<b>1</b>	<b>2.00%</b>	<b>1</b>	<b>1.50%</b>	
		Manual de usuario del sistema. (2%)	1	1.50%	0	0.00%	1	1.50%	1	2.00%	1	1.50%	
		Diagrama de navegación. (1%)	0	0.00%	1	1.00%	0	0.00%	0	0.00%	0	0.00%	
		Diagrama de distribución. (1%)	0	0.00%	0	0.00%	0	0.00%	0	0.00%	0	0.00%	
		Diagrama de componentes. (1%)	0	0.00%	0	0.00%	0	0.00%	0	0.00%	0	0.00%	
	<b>Validación y Verificación (5%)</b>		<b>3</b>	<b>4.25%</b>	<b>3</b>	<b>5.00%</b>	<b>3</b>	<b>3.60%</b>	<b>3</b>	<b>3.00%</b>	<b>3</b>	<b>4.10%</b>	
		Plan de pruebas del sistema (2%)	1	2.00%	1	2.00%	1	2.00%	1	2.00%	1	2.00%	
		Informe de los resultados de las pruebas. (2%)	1	1.50%	1	2.00%	1	1.00%	1	1.00%	1	1.50%	
		Certificación de las pruebas. (1%)	1	0.75%	1	1.00%	1	0.60%	1	0.00%	1	0.60%	
	<b>Despliegue (5%)</b>		<b>1</b>	<b>5.00%</b>	<b>1</b>	<b>5.00%</b>	<b>1</b>	<b>5.00%</b>	<b>0</b>	<b>0.00%</b>	<b>1</b>	<b>5.00%</b>	
									Continúa		a	➔	

		Documento de aceptación del sistema. (5%)	1	5.00%	1	5.00%	1	5.00%	0	0.00%	1	5.00%
<b>TOTAL</b>			<b>9</b>	<b>14.25%</b>	<b>9</b>	<b>14.50%</b>	<b>9</b>	<b>13.35%</b>	<b>10</b>	<b>9.50%</b>	<b>9</b>	<b>14.10%</b>
<b>Define criterios de calidad (10%)</b>												
	<b>Funcionalidad (2%)</b>		<b>3</b>	<b>1.90%</b>	<b>3</b>	<b>2.00%</b>	<b>3</b>	<b>1.95%</b>	<b>3</b>	<b>0.113</b>	<b>3</b>	<b>1.95%</b>
	La capacidad de la metodología para proveer las funciones que satisfacen las necesidades explícitas e implícitas cuando un proyecto se desarrolla bajo condiciones específicas.	Posee el número de personas requeridas. (0,75)	1	0.75%	1	0.75%	1	0.75%	1	10.00%	1	0.75%
		Permite la gestión de cualquier tipo de proyecto. (0,50)	1	0.50%	1	0.50%	1	0.45%	1	0.50%	1	0.45%
		Permite adaptarse a requisitos cambiantes e imprecisos. (0,75)	1	0.65%	1	0.75%	1	0.75%	1	0.75%	1	0.75%
	<b>Usabilidad (3%)</b>		<b>6</b>	<b>2.70%</b>	<b>6</b>	<b>3.00%</b>	<b>6</b>	<b>2.80%</b>	<b>6</b>	<b>3.00%</b>	<b>6</b>	<b>2.80%</b>
	La cualidad de un sistema, aplicación, herramienta u objeto que nos indica la facilidad con que éste se puede utilizar. Esto implica que "la herramienta" sea fácil de aprender a usar, que sus opciones (si las tiene) sean fáciles de recordar, que sea entendible, que su manipulación sea muy intuitiva, etc.	Arquitectura de la Interfaz. (0,5%)	1	0.50%	1	0.50%	1	0.50%	1	0.50%	1	0.50%
		Percepción. (0,5%)	1	0.40%	1	0.50%	1	0.40%	1	0.50%	1	0.40%
		Funcionalidad. (0,5%)	1	0.50%	1	0.50%	1	0.50%	1	0.50%	1	0.50%
		Facilidad de Aprendizaje. (0,5%)	1	0.50%	1	0.50%	1	0.50%	1	0.50%	1	0.50%
		Manejo de Errores. (0,5%)	1	0.40%	1	0.50%	1	0.50%	1	0.50%	1	0.50%
		Nivel de Seguridad. (0,5%)	1	0.40%	1	0.50%	1	0.40%	1	0.50%	1	0.40%
	<b>Fiabilidad (3%)</b>		<b>4</b>	<b>2.40%</b>	<b>4</b>	<b>3.00%</b>	<b>4</b>	<b>2.10%</b>	<b>4</b>	<b>0.03</b>	<b>4</b>	<b>2.10%</b>
	La capacidad de la metodología para mantener un nivel específico de funcionamiento cuando se está utilizando bajo	No presenta fallas de manera frecuente. (0,75)	1	0.60%	1	0.75%	1	0.60%	1	0.75%	1	0.60%
		Recuperar datos en caso de fallas. (0,75)	1	0.60%	1	0.75%	1	0.50%	1	0.75%	1	0.50%
		Es capaz de reestablecer el nivel de desempeño deseado. (0,75)	1	0.60%	1	0.75%	1	0.50%	1	0.75% Continúa	1	0.50% →

	requerimientos especificados. Las limitaciones en fiabilidad son debido a fallas en los requerimientos, diseño, e implementación.	La metodología evaluada reacciona adecuadamente cuando se presentan fallas. (0,75)	1	0.60%	1	0.75%	1	0.50%	1	0.75%	1	0.50%
	<b>Eficiencia (2%)</b>		<b>3</b>	<b>1.95%</b>	<b>3</b>	<b>2.00%</b>	<b>3</b>	<b>1.80%</b>	<b>3</b>	<b>0.02</b>	<b>3</b>	<b>1.85%</b>
	La capacidad de la metodología para proveer un desempeño adecuado, de acuerdo a la cantidad de recursos utilizados y bajo las condiciones planteadas.	Evaluada el tiempo de respuesta y de ejecución adecuada. (0,75)	1	0.75%	1	0.75%	1	0.65%	1	0.75%	1	0.70%
		Cumple con los plazos de desarrollo establecidos inicialmente. (0,75)	1	0.75%	1	0.75%	1	0.75%	1	0.75%	1	0.75%
		Mantiene un uso de recursos adecuado en cada tarea. (0,50)	1	0.45%	1	0.50%	1	0.40%	1	0.50%	1	0.40%
	<b>TOTAL</b>		<b>16</b>	<b>8.95%</b>	<b>16</b>	<b>10.00%</b>	<b>16</b>	<b>8.65%</b>	<b>16</b>	<b>19.25%</b>	<b>16</b>	<b>8.70%</b>
<b>Núm. Personas mínimo 4 - máximo 8 (10%)</b>												
	<b>Metodología ágil.</b>		1	10.00%	1	10.00%	0	0.00%	0	0.00%	0	0.00%
		Posee un equipo para el desarrollo de mínimo 4 y máximo 8 personas. (10%)	1	10.00%	1	10.00%	0	0.00%	0	0.00%	0	0.00%
	<b>TOTAL</b>		<b>1</b>	<b>10.00%</b>	<b>1</b>	<b>10.00%</b>	<b>0</b>	<b>0.00%</b>	<b>0</b>	<b>0.00%</b>	<b>0</b>	<b>0.00%</b>
<b>Tiempo Interactividad hasta 6 semanas (10%)</b>												
	<b>Interactividad.</b>		1	10.00%	1	10.00%	1	10.00%	1	10.00%	1	10.00%
		Posee un límite de 6 semanas para cada interactividad. (10%)	1	10.00%	1	10.00%	1	10.00%	1	10.00%	1	10.00%
	<b>TOTAL</b>		<b>1</b>	<b>10.00%</b>								
			<b>126</b>	<b>84.00%</b>	<b>119</b>	<b>85.75%</b>	<b>126</b>	<b>72.45%</b>	<b>121</b>	<b>85.50%</b>	<b>127</b>	<b>73.50%</b>

### 3.7.3. VENTAJAS Y DESVENTAJAS

Tabla 8

Ventajas y Desventajas de las metodologías de desarrollo.

Metodología	Ventajas	Desventajas
<b>XP</b>	<ul style="list-style-type: none"> <li>• Da lugar a una programación sumamente organizada.</li> <li>• Ocasiona eficiencias en el proceso de planificación y pruebas.</li> <li>• Propicia la satisfacción del programador.</li> <li>• Fomenta la comunicación entre los clientes y los desarrolladores.</li> <li>• Facilita los cambios.</li> <li>• Permite ahorrar mucho tiempo y dinero.</li> <li>• Puede ser aplicada a cualquier lenguaje de programación.</li> <li>• El cliente tiene el control sobre las prioridades.</li> <li>• Se hacen pruebas continuas durante el proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>• Es recomendable emplearla solo en proyectos a corto plazo.</li> <li>• En caso de fallar, las comisiones son muy altas.</li> <li>• Requiere de un rígido ajuste a los principios de XP.</li> <li>• Puede no siempre ser más fácil que el desarrollo tradicional.</li> </ul>
<b>SCRUM</b>	<ul style="list-style-type: none"> <li>• El cliente está satisfecho ya que recibe lo que necesita y esperaba.</li> <li>• El coste en términos de proceso y Management es mínimo, llevando a un resultado más rápido y barato.</li> <li>• Ayuda a la empresa a ahorrar tiempo y dinero</li> <li>• Permite realizar proyectos en los que la documentación de los requerimientos de negocios no está muy clara como para ser desarrolladas.</li> <li>• Los problemas se identifican por adelantado en las reuniones diarias y por lo tanto se pueden resolver rápidamente.</li> <li>• Hay visibilidad clara del desarrollo del proceso.</li> <li>• Poco control que insiste en la información frecuente del proceso</li> </ul>	<ul style="list-style-type: none"> <li>• Si no existe una fecha definitiva de finalización del proyecto es posible que se siga solicitando, y añadiendo, nueva funcionalidad.</li> <li>• Si una tarea no está bien definida, los costes de tiempo y dinero estimados del proyecto no serán demasiado exactos. En ese caso, la tarea se puede extender sobre varios sprints.</li> <li>• Si los miembros del equipo no están centrados y convencidos, el proyecto nunca se completará o incluso fallará.</li> <li>• Además de los recursos sin suficiente experiencia, la falta de dirección firme puede llevar a los proyectos a no completarse o incluso fallar.</li> <li>• Si se practican controles muy estrictos sobre los miembros del equipo, puede ser Continúa ➡</li> </ul>

	<p>en el trabajo mediante reuniones regulares.</p> <ul style="list-style-type: none"> <li>• Las reuniones diarias hacen posible medir la productividad individual. Esto lleva a la mejor en la productividad de cada uno de los miembros del equipo.</li> <li>• Es fácil entregar un producto de calidad en el tiempo estipulado.</li> <li>• Puede trabajar con cualquier tecnología o lenguaje de programación.</li> <li>• Hace el proceso del desarrollo de software más centrado y manejable.</li> </ul>	<p>extremadamente frustrante para ellos, llevando a la desmoralización y el fallo del proyecto.</p> <ul style="list-style-type: none"> <li>• El control de la calidad del proyecto es difícil de implementar y cuantificar a menos que el equipo de test puedan llevar a cabo testeos de regresión después de cada sprint.</li> </ul>
<b>DSDM</b>	<ul style="list-style-type: none"> <li>• Permite la gestión de proyectos eficiente y un fuerte control sobre el ciclo de vida del proyecto.</li> <li>• Enfoque prioritario Requisito útil en la entrega de las funcionalidades más importantes primero.</li> </ul>	<ul style="list-style-type: none"> <li>• La documentación es compleja y requiere mucho tiempo.</li> <li>• Presupuesto (Dinero)</li> <li>• Continua implicación del usuario.</li> </ul>
<b>FDD</b>	<ul style="list-style-type: none"> <li>• Está pensada para equipos y proyectos grandes.</li> <li>• Detalla más el proceso.</li> <li>• Detalla la iteración cero.</li> <li>• Contempla fase la arquitectura y el diseño.</li> <li>• Define roles, como el jefe de proyecto y el arquitecto.</li> <li>• Manejo jerárquico de los requisitos.</li> <li>• Metodología se puede utilizar en caso de aplicaciones que son complejas debido a la documentación y los informes que se crean.</li> </ul>	<ul style="list-style-type: none"> <li>• Muchas personas para todo el proceso.</li> <li>• La complejidad es tanto que no hay ningún punto en el uso de este método para proyectos más pequeños.</li> <li>• Menos comunicación dentro y fuera del equipo. Por lo tanto los equipos aprenden menos de otras personas y equipos</li> </ul>
<b>ASD</b>	<ul style="list-style-type: none"> <li>• Atención a la evaluación y control.</li> <li>• Difunde la colaboración.</li> <li>• Proceso adaptativo</li> <li>• Proceso iterativo incremental.</li> <li>• Se utiliza para poder aprender de los errores e iniciar nuevamente el ciclo de desarrollo.</li> </ul>	<ul style="list-style-type: none"> <li>• No escalable.</li> <li>• Exceso de dependencia de la comunicación interhumana.</li> <li>• No hay modelos específicos prescritos.</li> <li>• Los errores y cambios que no son detectados con anterioridad afectan la calidad del producto y su costo total.</li> </ul>

#### **3.7.4. SELECCIÓN DE LA METODOLOGÍA QUE SE ACOPLE A LA CPN.**

Una buena selección de una metodología; y también el buen desarrollo de software; es producto de una minuciosa elección de estándares, normas y características para trabajar mediante una Metodología de Desarrollo de Software fija. La investigación realizada ha permitido entender y comprender que el diseño y el desarrollo de software no es una labor corriente, por mucho tiempo en la Cooperativa de Ahorro y Crédito Policía Nacional Ltda., este trabajo se lo ha llevado adelante sin una metodología definida.

Se sabe por la investigación realizada que las metodologías ágiles se basan en técnicas provenientes de prácticas de producción de código; además están preparadas para cambios durante todo el proyecto las cuales son impuestas internamente por el equipo con procesos menos controlados, con pocos principios. Así como también con un contrato flexible e incluso inexistente, donde el cliente es parte del desarrollo y los grupos son pequeños por lo que se incluyen pocos artefactos. Al contrario, las metodologías tradicionales se basan en normas provenientes de estándares seguidos por el entorno de desarrollo; con una fuerte resistencia a los cambios.

Todo lo anteriormente mencionado, nos ayuda a concluir algunas cosas; es claro que antes de desarrollar un producto software, necesitamos comprender el punto de vista del producto, la cual se la puede obtenerla en base a la relación con el cliente, y un previo establecimiento del modelo de ciclo de vida del software, así como la gestión de los requisitos, el plan de desarrollo y también la parte de la integración del proyecto. Esto nos permitirá tener un amplio panorama donde podremos ver las medidas de progreso del proyecto, así como los indicadores para evaluar la calidad, las maneras de medir el riesgo, y saber con ello como gestionar los cambios y así establecer una línea de meta; lo que nos ayuda para poder seleccionar una metodología de desarrollo de software; pues al ver lo que realmente necesitamos, podremos ver que es lo que nos proporciona

una metodología tradicional y una ágil, independientemente de cuál seleccionemos para llevar a cabo el desarrollo de nuestro software, ya que tenemos variedades de metodologías tanto tradicionales como ágiles.

Se debe tener claro que para seleccionar una metodología ágil de desarrollo de software no existe un criterio o alguna guía que nos indique como se debe llevar una correcta selección.

Se puede decir que la formulación para poder seleccionar una buena metodología se basa en una correcta selección por criterios de expertos y por conocimiento de desarrollo en la rama de los analistas y diseñadores que nos guie a la necesidad del cliente y la metodología que se acople a dicha necesidad.

Para este proyecto se definió que se necesita una metodología ágil de desarrollo así que se realizó el análisis de cinco metodologías ágiles como se puede observar en la Tabla 7 y para poder evaluar a cada una de estas metodologías se utilizaron los criterios como: herramientas, métodos, procesos y criterios de calidad, así como criterios que debe tener una metodología ágil de desarrollo como número máximo de personas y tiempo de interactividad, así como la norma ISO/IEC 9126.

La norma mencionada en el párrafo anterior permite evaluar y especificar la calidad del software desde distintas perspectivas, las cuales están asociadas a la auditoria del software, aseguramiento de la calidad, mantenimiento, soporte, evaluación, uso, desarrollo, requerimientos, y adquisición. Puede ser usada por grupos de aseguramiento de la calidad, evaluadores independientes, y desarrolladores, responsables de especificar y evaluar la calidad del software.

La ISO 9126 ha sido el que se ha establecido como modelo de calidad base en la comunidad científica y, por ello, se ha considerado más apropiado seguir las indicaciones y terminología de dicha norma.

Mediante el análisis realizado se identificó que la metodología de desarrollo de software SCRUM es la que más se acopla a las necesidades de la Cooperativa de Ahorro y Crédito Policía Nacional Ltda.

## CAPÍTULO IV. EJECUCIÓN DEL ESTUDIO.

### 4.1. DISEÑO DEL PROTOTIPO.

<i>Documento de Requerimiento de Desarrollo</i>	
<b>Aprobaciones del Proyecto:</b>	<b>Identificación del Requerimiento: 1</b>
<b>Nombre del Requerimiento:</b> Módulo Mis Décimos	<b>Versión del Documento:</b> 1.0 <b>Fecha:</b> 04-01-2016
<b>Objetivo:</b> Desarrollar una aplicación que permita a los socios de la CPN ingresar y registrar los datos necesarios para obtener el contrato de ahorro programado para la acumulación de sus décimos.	<b>Fecha propuesta para implementación:</b>
<b>División y Área solicitante:</b> Negocios. <b>Patrocinador:</b>	<b>Líder del Proyecto:</b> Tlga. Pamela Herrán <b>Teléfono:</b> 0987082447 <b>Líder Back-Up:</b>
<b>Beneficios al Negocio:</b> La Cooperativa Policía Nacional no cuenta con un producto que permita generar ahorros programados para los décimos a mensualizar a través de la web, por lo que es necesario crear este módulo a fin de incrementar las captaciones. <b>Beneficios Operativos:</b> •	<b>Propósito del Requerimiento</b> <input type="checkbox"/> RG – Legal / Regulatorio <input type="checkbox"/> CP – Corrección problema <input type="checkbox"/> MJ – Mejora (Perfeccionamiento) <input checked="" type="checkbox"/> NV – Nuevo
<b>Productos/Servicios impactados:</b> <i>Software a modificar o Desarrollar</i>	<b>Impacto al proceso:</b> <input type="checkbox"/> Bajo <input type="checkbox"/> Medio <input checked="" type="checkbox"/> Alto
<b>Antecedentes:</b> Ante la ley propuesta por el gobierno de depositar mensualmente los valores del décimo tercer y cuarto sueldo nace la iniciativa de desarrollar un módulo en la cual el socio de la CPN pueda acumular el valor de sus décimos en una cuenta de ahorro programado y así nuestros socios puedan ahorrar dichos valores y evitar el gasto innecesario que se presentaría si los valores le fueran acreditados en su sueldo.	
<b>Como se maneja Actualmente</b>	
<b>Funciones y propósito a alto nivel de cada una (ingresos, autorizaciones, pantallas, reportes, controles, seguridades):</b>	
a) Actualmente no existe un módulo que permita realizar la solicitud de acumulación de décimos.	
<b>Describa lo que desea que se desarrolle con ejemplos de pantallas y controles que se debería controlar</b>	
<ul style="list-style-type: none"> <li>• El socio accederá al sistema mediante su número de cédula y su clave del cajero automático en el cual deberá ingresar:               <ul style="list-style-type: none"> <li>- Número de Cédula</li> <li>- Apellidos</li> </ul> </li> </ul>	
<i>Uso Interno</i>	

Continúa 

*Documento de Requerimiento de Desarrollo*

---

- Nombres
- Agencia
- Grado Policial
- Años de servicio
- Valor Descontar.

Una vez ingresada la información el socio procederá a guardar la solicitud y el sistema permitirá imprimir el contrato de cuenta décimos una vez que el socio acepte los términos y condiciones del mismo.

Los socios que ingresan la solicitud se podrán ver en el módulo de reportes web mediante la generación de un reporte (Lista de socios).  
El reporte permitirá observar el comportamiento de los contratos ingresados para la acumulación de décimos.

**REPORTE DE CONTROL:** (jefatura comercial)

- Nombres y Apellidos
- No. cédula
- No. socio
- Login del usuario que ingresó el contrato
- Monto
- Fecha de ingreso del contrato a la web
- Fecha de apertura del contrato al sistema
- Fecha de pago final
- No. cuenta
- Motivo para poder clasificar a las cuentas décimas de los demás contratos de ahorro programado.

Describe, de manera general, una a una las funciones que serán afectadas

**Criterios de Aceptación:**

**Pruebas en QA:**

- Que sea socio de la cooperativa
- Que la clave de la tarjeta de débito sea correcta
- Que el monto de descuento mínimo responda al grado policial ingresado.
- Que acepte a través de un visto en los términos y condiciones lo que ahí detalla.
- Que reporte satisfactoriamente la transacción realizada
- Que solo se ingrese un contrato por cada número de cédula ingresado.

Describe la razonabilidad que utilizará para verificar el requerimiento

Área	Responsable	Rol / Firma	Fecha
Negocios	Tiga. Pamela Herrán	Jefe Comercial y Negocios Solicita	
Sistemas	Sr. Darwin Tipán	Analista N2 de Aplicaciones Recibe	

*Pág. 1*

Figura 19 Diseño del experimento.

## **4.2. DESARROLLO DEL PROTOTIPO.**

### **4.2.1. REQUISITOS DE SOFTWARE**

Para la demostración de la metodología seleccionada se ha decidido utilizarla en el desarrollo de una aplicación que permita a los socios de la CPN (Servicio Activo, Servicio Pasivo, Familiares de Policías y Empleados Civiles) solicitar la acumulación de sus décimos (tercero y cuarto) mediante una cuenta de ahorro programado. Dicha aplicación se ha denominado MIS DÉCIMOS cuyos requisitos se los puede encontrar en el archivo [4 Requisitos Desarrollo](#).

### **4.2.2. DISEÑO**

La CPN es una entidad financiera cuyo core bancario es COBIS, y debido a la gran cantidad de transacciones que se realizan y para que dichas transacciones sean más rápidas en su tiempo de respuesta Cobis maneja una base de datos no relacional (Las bases de datos no relacionales pueden manejar enormes cantidades de datos así como no generan cuellos de botella.); por ese motivo las tablas de las bases de datos de la CPN se relacionan por códigos, en la figura N° 17 se puede observar las tablas que utilizaremos en el desarrollo de la aplicación.

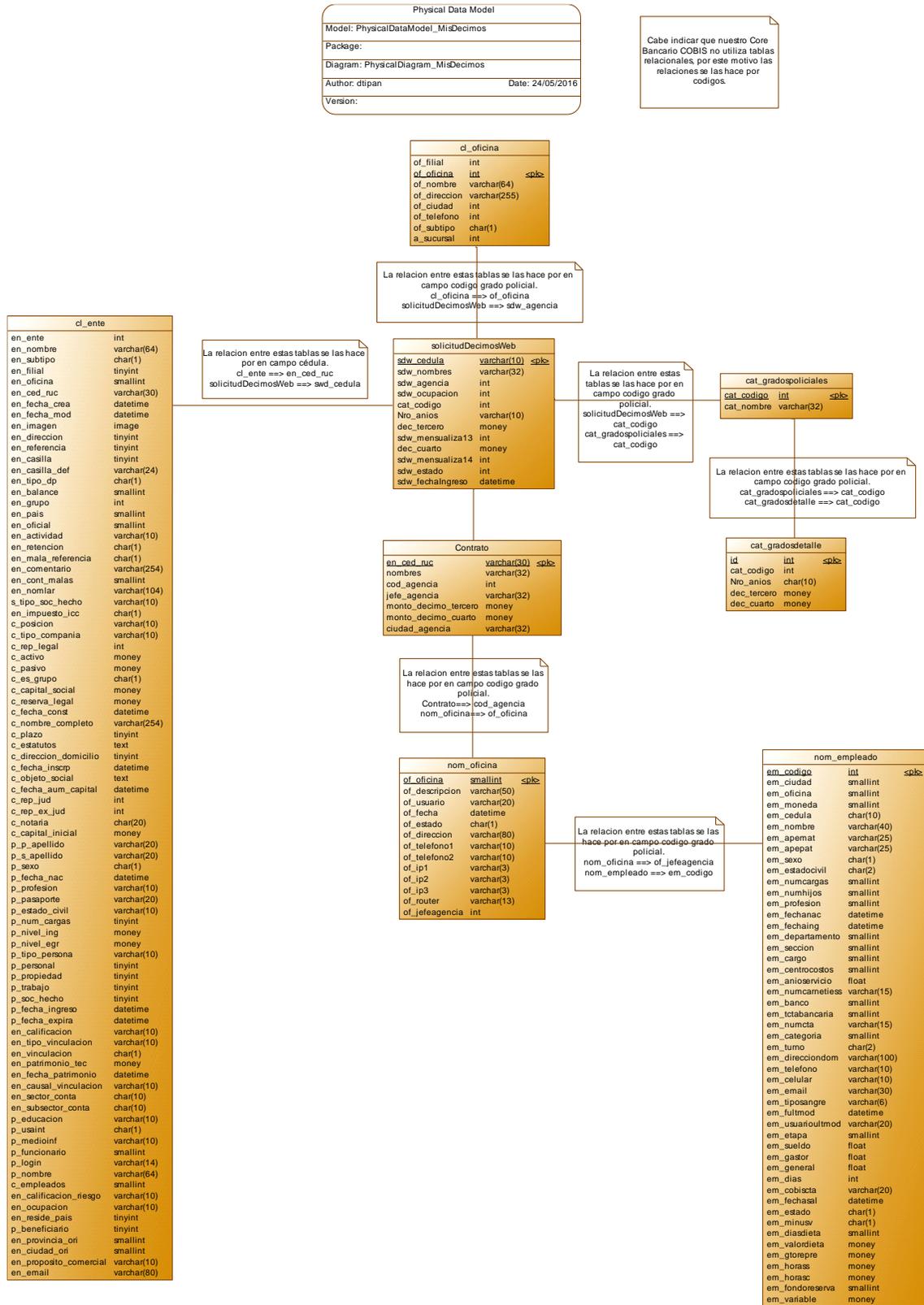


Figura 20 Diagrama de Base de Datos.

En la figura N° 18 se muestra el diagrama de secuencias de la solicitud de acumulación de décimos de los socios (Servicio Activo y Servicio Pasivo) donde podemos identificar las iteraciones entre los objetos del sistema MIS DÉCIMOS.

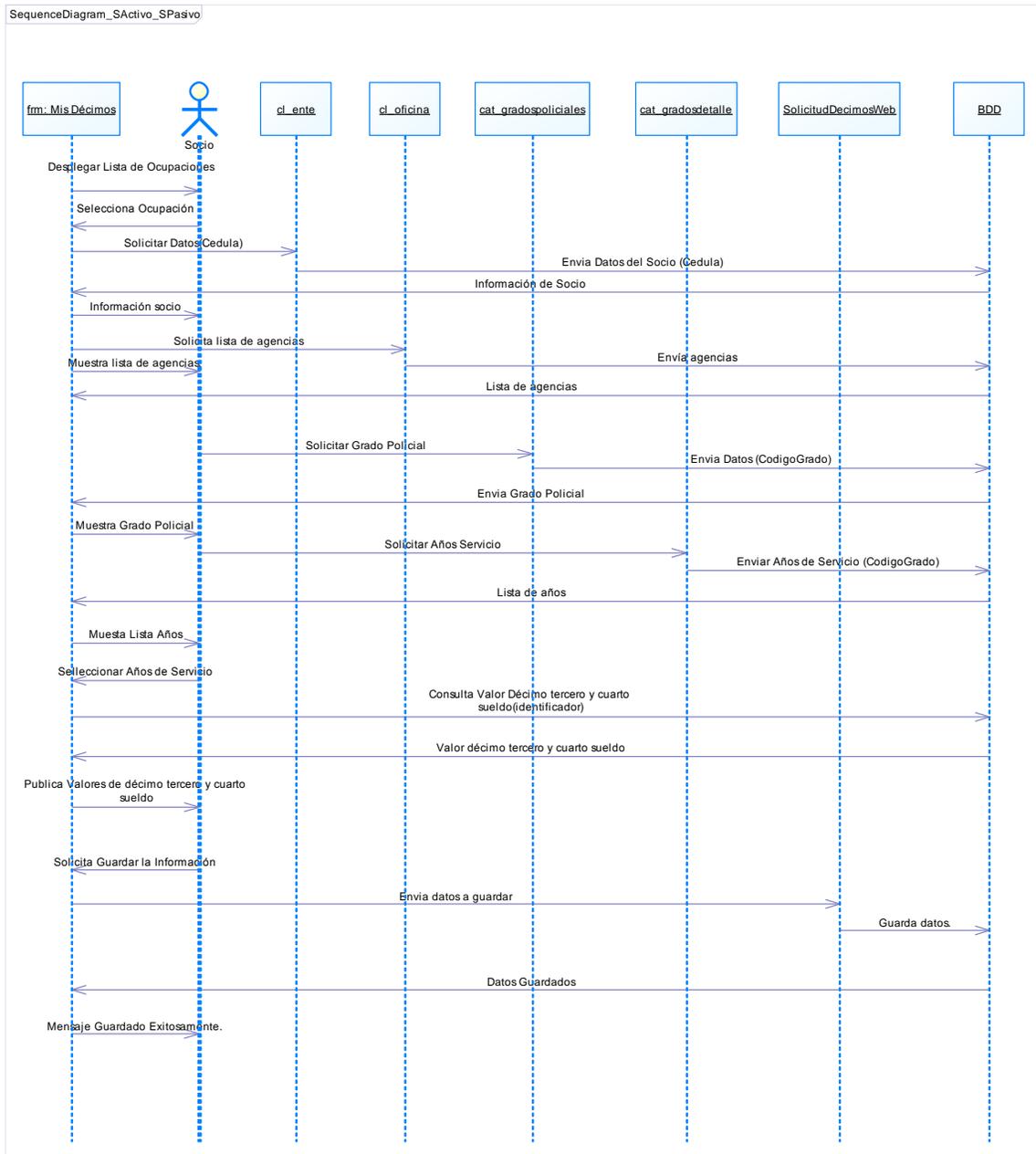


Figura 21 Diagrama de Secuencia Servicio Activo y Servicio Pasivo.

En la figura N° 19 se muestra el diagrama de secuencias de la solicitud de acumulación de décimos de los socios (Familiares y Empleados Civiles) donde podemos identificar las iteraciones entre los objetos del sistema MIS DÉCIMOS.

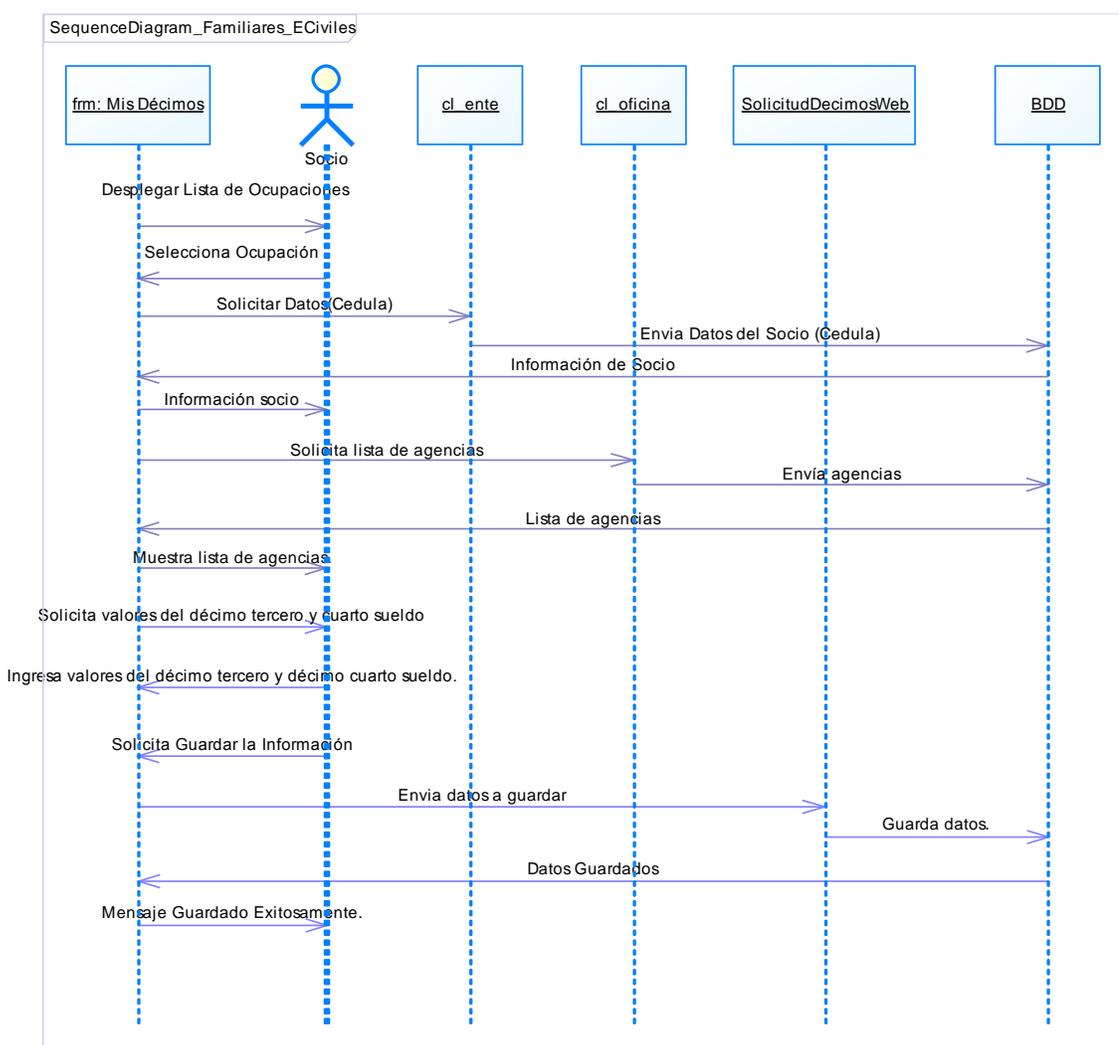


Figura 22 Diagrama de secuencia Familiares y Empleados Civiles.

### 4.2.3. CONSTRUCCIÓN

En esta etapa se especifican las herramientas con las que se ha desarrollado y construido el Módulo MIS DÉCIMOS y los equipos que se necesitan para su ejecución, así como también los entregables según la metodología escogida.

#### 1. PROTOTIPO

Pantalla Principal de Acceso al Módulo. (Pantalla 1)

Diagrama de la pantalla de Login (Pantalla 1). El título es "Pantalla Login". Hay dos campos de entrada de texto: "Cédula" y "Clave Tarjeta". Debajo de "Clave Tarjeta" hay un botón que dice "INGRESAR".

Figura 23 Pantalla Prototipo Login MIS DÉCIMOS.

Una vez ingresado tenemos la pantalla donde el asociado escogerá a qué tipo de socio pertenece. (Pantalla 2)

Diagrama de la pantalla Tipo de Socio (Pantalla 2). El título es "Tipo de Socio". Hay cuatro botones que representan los tipos de socio: "Servicio Activo", "Servicio Pasivo", "Familiares" y "Empleado Civil".

Figura 24 Pantalla Prototipo Tipo de Socio MIS DÉCIMOS.

Una vez que el socio haya seleccionado cualquiera de las cuatro opciones se le mostrará la pantalla en la que podrá ingresar sus datos para solicitar la acumulación de sus décimos. (Pantalla 3)

Solicita Acumulación socios Servicio Activo y Pasivo

Cédula	<input type="text"/>
Nombres Completos	<input type="text"/>
Agencia	<input type="text"/>
Ocupación	<input type="text"/>
Grado Policial	<input type="text"/>
Tiempo de Servicio	<input type="text"/>
Valor Décimo Tercero	<input type="text"/> <input type="checkbox"/> Habilitar
Valor Décimo Cuarto	<input type="text"/> <input type="checkbox"/> Habilitar

Conozco y acepto los términos y condiciones de la CUENTA MIS DECIMOS.

Figura 25 Pantalla Ingreso Acumulación Décimos (Activos y Pasivos).

Solicita Acumulación socios Familiares y Empleados Civiles

Cédula	<input type="text"/>
Nombres Completos	<input type="text"/>
Agencia	<input type="text"/>
Ocupación	<input type="text"/>
Valor Décimo Tercero	<input type="text"/> <input type="checkbox"/> Habilitar
Valor Décimo Cuarto	<input type="text"/> <input type="checkbox"/> Habilitar

Conozco y acepto los términos y condiciones de la CUENTA MIS DECIMOS.

Figura 26 Pantalla Ingreso Acumulación Décimos (Familiares y Civiles).

En la tercera pantalla el sistema cargará automáticamente los siguientes datos:

- Número de cédula.
- Nombres Completos.
- Ocupación.

Será necesario ingresar los siguientes datos para socios Activos y Pasivos:

- Agencia.
- Grado Policial.
- Tiempo de Servicio
- Valor décimo tercero y/o Valor Décimo Cuarto.

Y para socios Familiares y Empleados Civiles será necesario ingresar la siguiente información.

- Agencia.
- Valor décimo tercero y/o Valor Décimo Cuarto.

## 2. ACCESO AL SISTEMA

En cada sistema existe o debe existir un control de acceso a los datos, donde esta permita garantizar la seguridad de los mismos. Para esta aplicación se ha decidido la incorporación de un Login en la cual valide el ingreso de los socios mediante su número de cédula y su clave de la tarjeta del cajero automático.

## 3. NIVEL DE SEGURIDAD

- Autenticación (Usuario y Clave).

## 4. ARQUITECTURA

La arquitectura lógica que se utilizó para el desarrollo de esta aplicación es de 3 capas.

- Capa 1. Modelo de Presentación.
- Capa 2. Modelo de Negocio.
- Capa 3. Modelo de Datos.

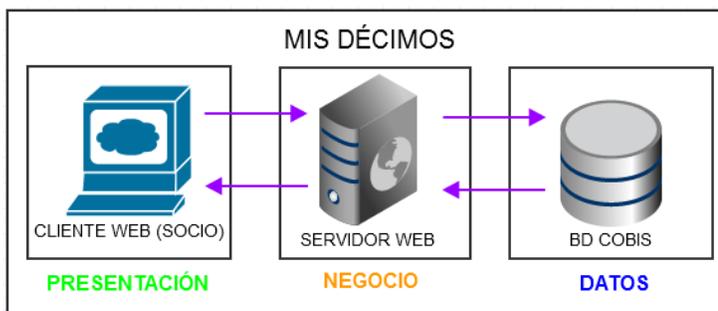


Figura 27 Arquitectura Módulo Mis Décimos

A continuación, en la Figura 25 se presenta la estructura de arquitectura del módulo Mis Décimos a través de un diagrama de despliegue en el cual se puede diferenciar las tres capas que lo conforman. Por una parte, se encuentra el navegador web que puede ser cualquiera que escoja el usuario, por ejemplo, Chrome, Mozilla Firefox, Safari, etc. El servidor de aplicaciones es un Windows Server 2007 llamado Srvteamf03, la base de datos es Cobis 15.7 finalmente el protocolo de comunicación en este caso es TCP/IP.

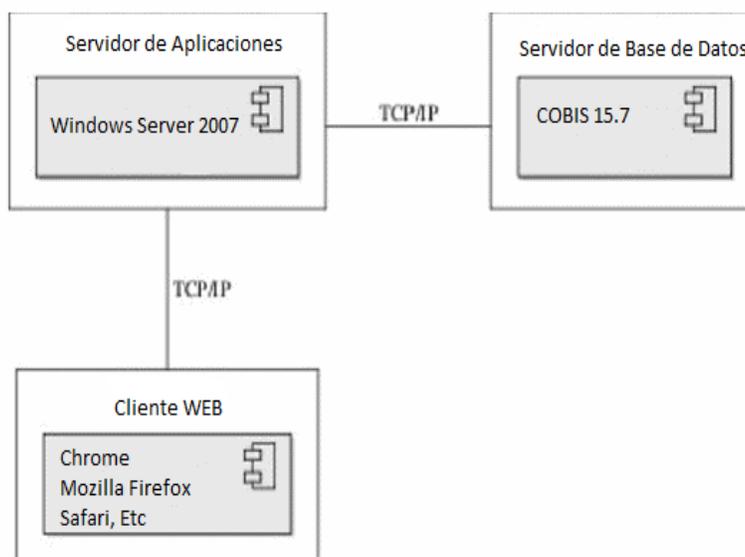


Figura 28 Arquitectura Módulo Mis Décimos Diagrama de Despliegue

## 5. BASE DE DATOS

- Sybase 15.7

## 6. SOLUCIÓN

- Login



Figura 29 Pantalla Login MIS DÉCIMOS.

- Tipo de socio



Figura 30 Pantalla Tipo de Socio MIS DÉCIMOS.

- Solicita Acumulación Décimos (Servicio Activo y Servicio Pasivo)

Servicio Activo - Google Chrome  
190.110.196.94:18081/ServicioActivo.aspx

**CUENTA MIS DÉCIMOS**

Cédula: 0102284734

Nombres: CASTRO QUEZADA MARIO VICENTE

Agencia: Guayaquil

Ocupación: Servicio Activo

Grado: SUBOFICIAL MAYOR

Tiempo de Servicio: 2DO. AÑO

Usted puede incrementar sus DÉCIMOS si lo desea.

Valor 13° Sueldo: 222.00  Deshabilitar Contrato.

Valor 14° Sueldo: 31.00  Deshabilitar Contrato.

Acepto y acepto los términos y condiciones de la CUENTA MIS DÉCIMOS.

[Ver Términos y Condiciones](#)

**GUARDAR** **IMPRIMIR**

Se ha guardado exitosamente.

Figura 31 Pantalla Ingreso Acumulación Décimos (Activos y Pasivos).

- Solicita Acumulación Décimos (Familiars y Empleados Civiles)

Empleados Civiles - Google Chrome  
190.110.196.94:18081/EmpleadosCiviles.aspx

**CUENTA MIS DÉCIMOS**

Cédula: 1718093170

Nombres: TIPAN COYAGO DARWIN MAURICIO

Agencia: Matriz

Ocupación: Empleado Civil

Usted puede incrementar sus DÉCIMOS si lo desea.

Valor 13° Sueldo: 0  Habilitar Contrato.

Valor 14° Sueldo: 31.00  Deshabilitar Contrato.

Acepto y acepto los términos y condiciones de la CUENTA MIS DÉCIMOS.

[Ver Términos y Condiciones](#)

**GUARDAR** **IMPRIMIR**

Se ha guardado exitosamente.

Figura 32 Pantalla Ingreso Acumulación Décimos (Familiars y Civiles).

- Impresión Contrato.

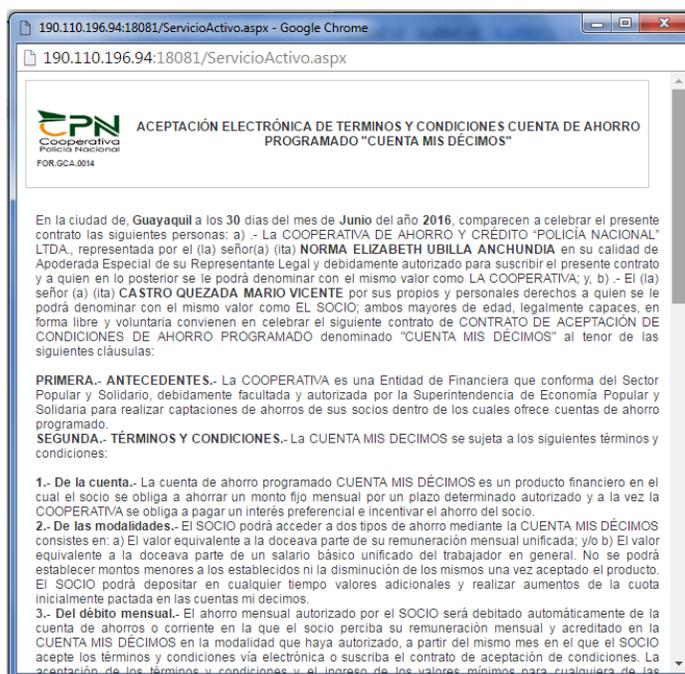


Figura 33 Contrato de Acumulación Décimos pág. 1/2.

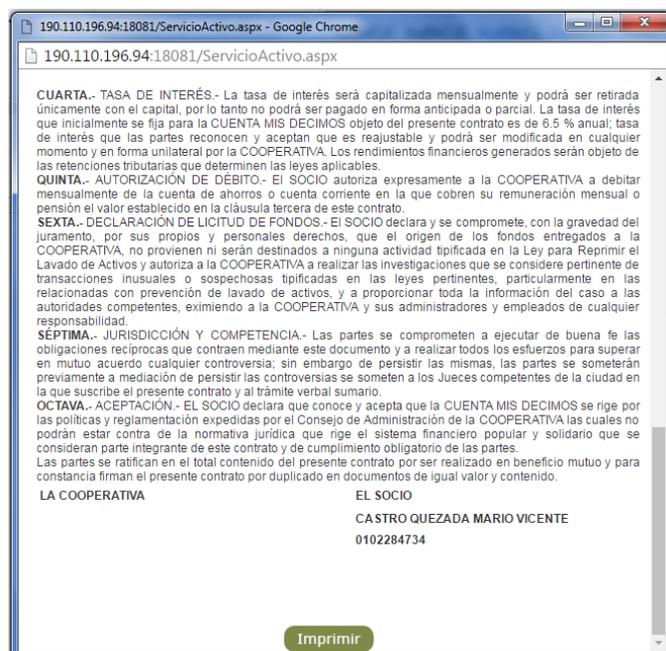


Figura 34 Contrato de Acumulación Décimos pág. 2/2.

- Listado de solicitudes de acumulación de décimos ingresadas por la web.



Figura 35 Sistema Reportes (Listado Solicitudes Décimos).

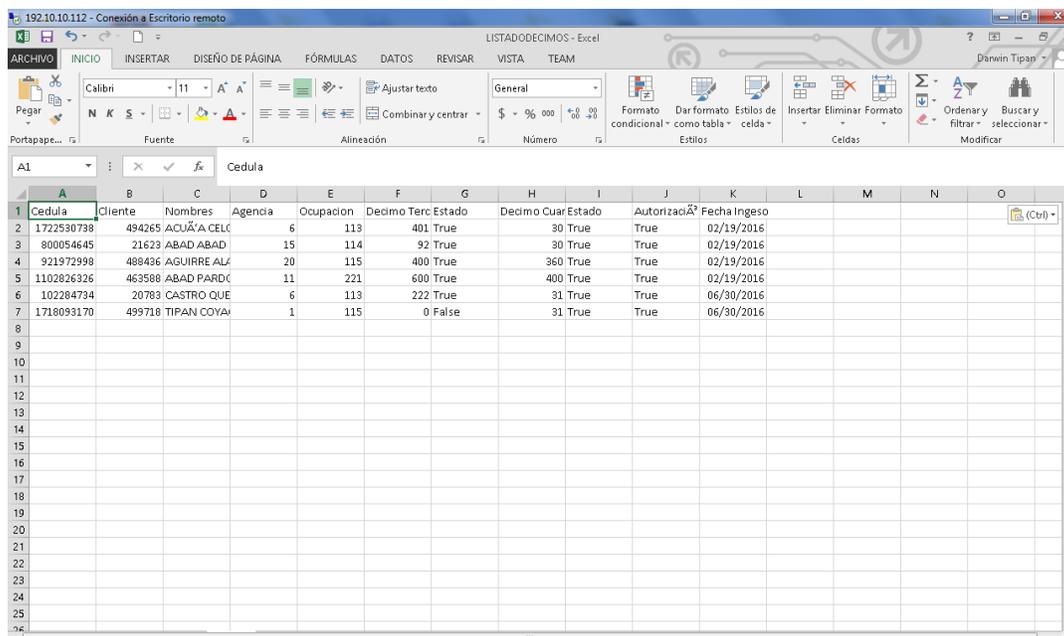


Figura 36 Listado Solicitudes Décimos (Exportado del Sistema de Reportes).

#### 4.2.4. VERIFICACIÓN Y VALIDACIÓN

Para la realización de las pruebas se ha tomado en cuenta el documento [Plan de Pruebas](#) donde se puede realizar la verificación y validación del módulo MIS DÉCIMOS y la obtención de óptimos resultados esperados por el cliente.

Los resultados obtenidos de las pruebas realizadas se las puede observar en el documento [Resultado Pruebas](#) dónde se especifica los casos de prueba y el resultado obtenido de cada uno al ejecutarlo.

#### 4.2.5. DESPLIEGUE.

Una vez que las pruebas del sistema han sido aplicadas y aceptadas (ver documento [Aceptacion Pruebas](#)) se procede a la liberación en el ambiente de producción del módulo desarrollado:

##### 1. Instalación.

- a) Publicar la solución en el servidor de aplicaciones.- Como podemos ver en la figura N° 32 y N° 33 se procede a publicar la solución desde Visual Studio.

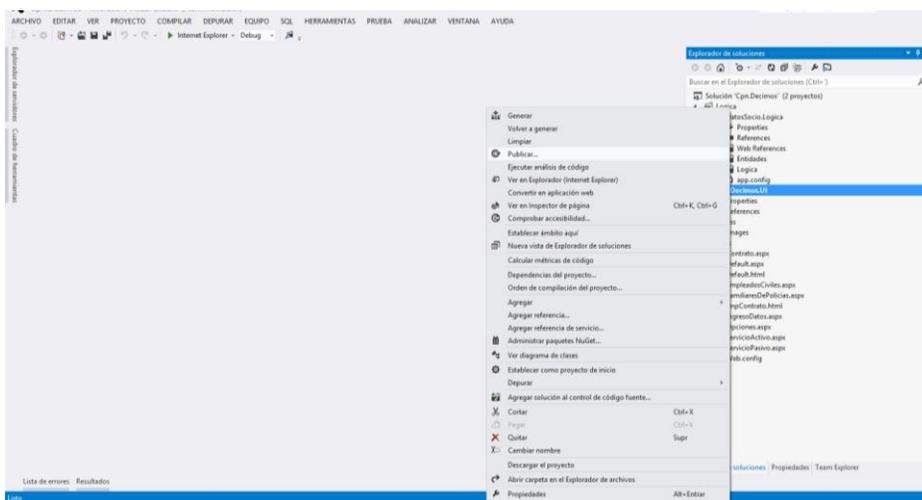


Figura 37 Se publica la solución desde Microsoft Visual Studio.

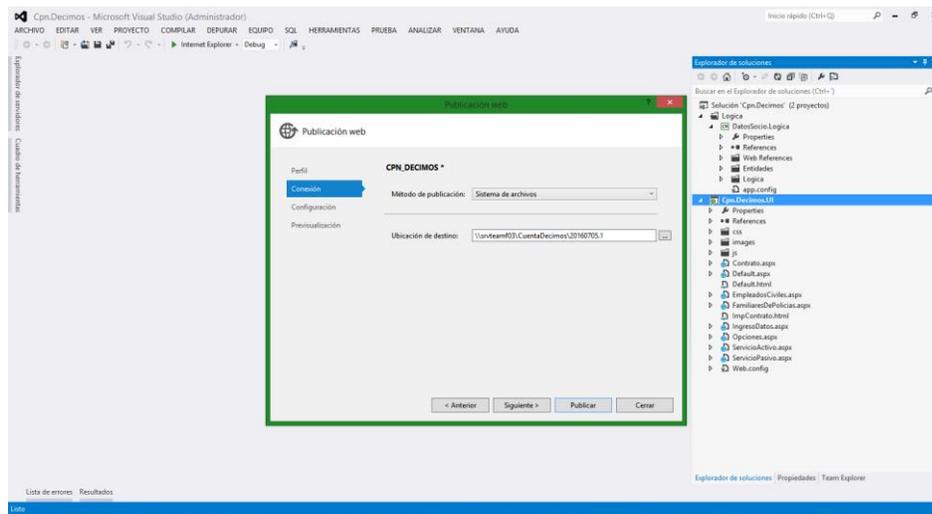


Figura 38 Se publica la solución en el servidor de aplicaciones.

b) En las figuras 35, 36, 37 y 38, se configura el módulo en el servidor de aplicaciones.

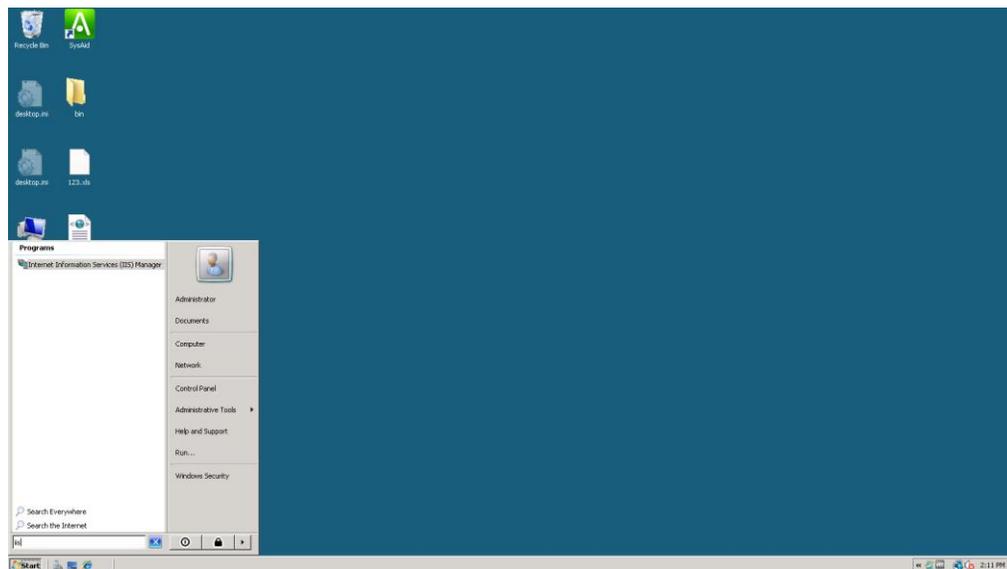


Figura 39 Seleccionar el IIS.

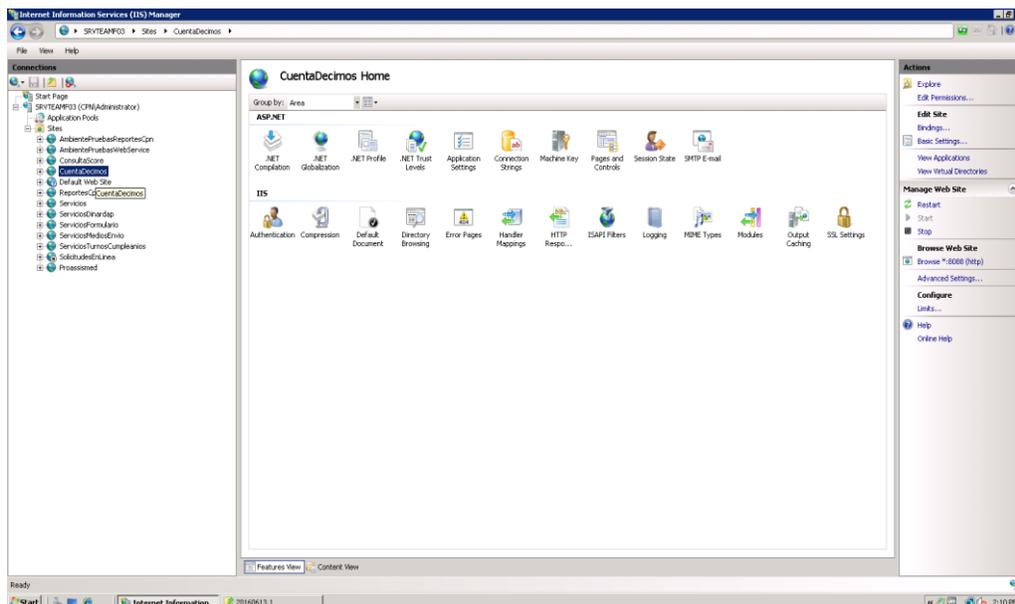


Figura 40 Selección de la aplicación en el IIS.

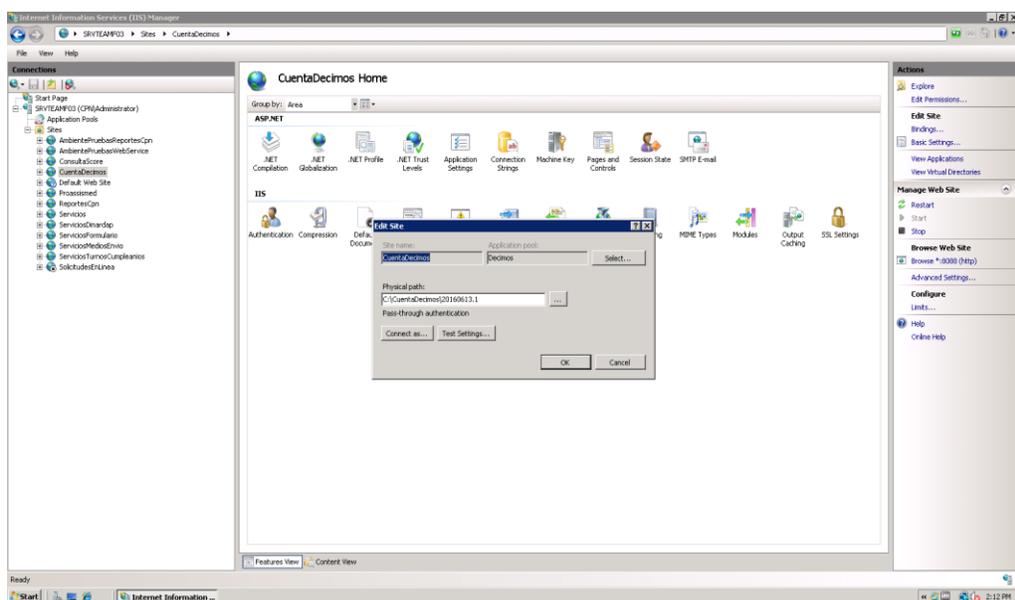


Figura 41 Direccionamiento del sitio a la ubicación del módulo.

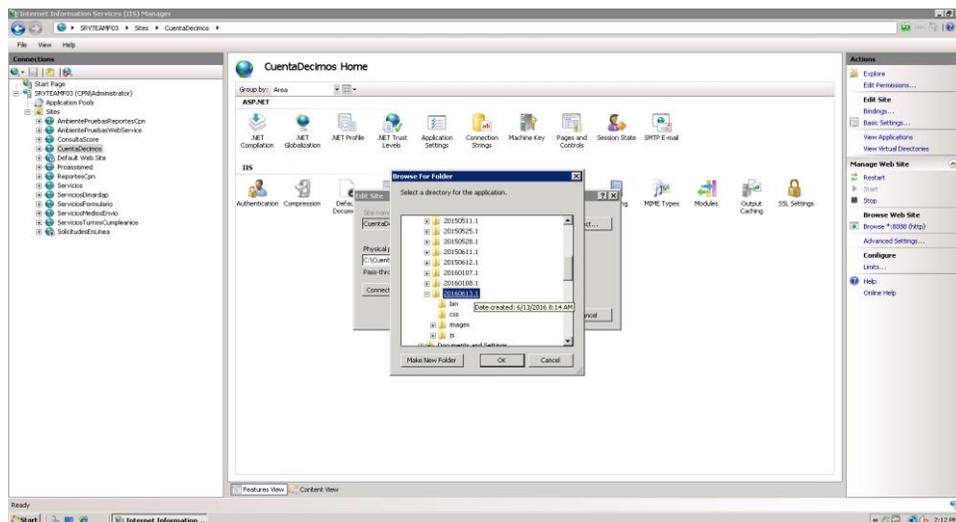


Figura 42 Selección del módulo publicado.

## 2. Soporte.

El soporte y entrenamiento para la utilización del software es muy importante y algo que muchos desarrolladores de software descuidan. Los usuarios, por naturaleza, se oponen al cambio porque conlleva una cierta inseguridad, es por ello que es primordial instruir de forma adecuada a los futuros usuarios del software.

Para el caso de la CPN, se procedió a instruir al personal de soporte (ver Figura 5) dos horas durante dos días; para que de este modo puedan capacitar a las personas de servicio al cliente y así se pueda brindar la debida ayuda a todos los socios de la Cooperativa.

Una vez realizada la inducción al personal indicado se tuvieron algunas novedades como las que se mencionan a continuación:

- Socios no se acordaba su clave debido a que utilizaban su huella digital para retirar dinero.
- Los socios no estaban al tanto de que si colocaban tres veces mal su clave, se les bloqueaba la misma al igual que en los cajeros automáticos.

- Una vez realizado el ingreso de la solicitud, el socio no podía modificar los datos ingresados.

#### 4.2.6. ANÁLISIS DE RESULTADOS.

Con el fin de lograr los objetivos planteados al inicio de esta tesis, se procedió a comparar los resultados obtenidos con la elaboración del módulo MIS DÉCIMOS y a compararlos con otros sistemas de igual o menor tamaño desarrollados en la CPN sin ninguna metodología de desarrollo de software. Para esto se utilizaron indicadores de desempeño los mismos que se pueden observar en la tabla N° 9 así como se realizaron gráficas en Excel, para una mejor comprensión de los resultados.

Tabla 9  
Indicadores de desempeño análisis de resultados.

Característica	Porcentaje	Descripción
<b>Funcionalidad</b>	30%	La capacidad del producto de software para proveer las funciones que satisfacen las necesidades explícitas e implícitas cuando el software se utiliza bajo condiciones específicas.
<b>Fiabilidad</b>	20%	La habilidad de la unidad funcional de realizar una función requerida. (ISO/IEC 2382-14:1997). La capacidad del producto de software para mantener un nivel específico de funcionamiento cuando se está utilizando bajo requerimientos especificados. El desgaste o envejecimiento no ocurre en el software. Las limitaciones en fiabilidad son debido a fallas en los requerimientos, diseño, e implementación.
<b>Usabilidad</b>	15%	La capacidad del producto de software de ser entendido, aprendido, usado y atractivo al usuario, cuando es utilizado bajo las condiciones especificadas. Algunos aspectos de funcionalidad, fiabilidad y eficiencia también afectarán la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no son clasificados como usabilidad
<b>Eficiencia</b>	20%	La capacidad del producto de software para proveer un desempeño adecuado, de acuerdo a la cantidad de recursos utilizados y bajo las condiciones planteadas. Continúa ➡

<b>Mantenimiento</b>	10%	Capacidad del producto de software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, y especificaciones de requerimientos funcionales.
<b>Portabilidad</b>	5%	La capacidad del software para ser trasladado de un entorno a otro. El entorno puede incluir entornos organizacionales, de hardware o de software.
<b>Costo</b>	0%	
<b>TOTAL</b>	100%	

Para el análisis se ha utilizado el método MoSCow que es una técnica de priorización de requisitos que se basa en el hecho de que, aunque todos los requisitos se consideren importante, es importante destacar los requisitos vitales que aportan mayor valor al negocio y que son realmente considerado obligatorios. El análisis de resultados completo se lo puede encontrar en el documento [Análisis de Resultados Metodología Scrum en la CPN](#).

En la tabla N° 10 podemos observar los puntajes totales del análisis de resultados de siete aplicaciones desarrolladas sin ninguna metodología de desarrollo frente al sistema Mis Décimos en la cual se utilizó la metodología Scrum.

Tabla 10  
Matriz análisis de resultados.

	100%	WS Garantías	WS SolicitudesLinea	WS MediosEnvio	WS Cumpleaños	Proassismed	WS DatosTarjetaHabiente	WS CobraSueldo	Mis Décimos
<b>Funcionalidad</b>	<b>30%</b>	5%	5%	5%	5%	4%	5%	5%	19%
		<b>2%</b>	<b>2%</b>	<b>2%</b>	<b>2%</b>	<b>1%</b>	<b>2%</b>	<b>2%</b>	<b>6%</b>
<b>Fiabilidad</b>	<b>20%</b>	159%	159%	156%	138%	138%	113%	125%	244%
		<b>32%</b>	<b>32%</b>	<b>31%</b>	<b>28%</b>	<b>28%</b>	<b>23% Continúa</b>	<b>25% úa</b>	<b>49%</b> 

<b>Usabilidad</b>	<b>15%</b>	54%	54%	54%	54%	63%	54%	54%	157%
		<b>8%</b>	<b>8%</b>	<b>8%</b>	<b>8%</b>	<b>9%</b>	<b>8%</b>	<b>8%</b>	<b>24%</b>
<b>Eficiencia</b>	<b>20%</b>	100%	100%	100%	100%	100%	100%	100%	225%
		<b>20%</b>	<b>45%</b>						
<b>Mantenimiento</b>	<b>10%</b>	45%	45%	45%	45%	45%	45%	45%	95%
		<b>4%</b>	<b>9%</b>						
<b>Portabilidad</b>	<b>5%</b>	118%	118%	118%	118%	118%	118%	118%	205%
		<b>6%</b>	<b>10%</b>						
<b>Costo</b>	<b>0%</b>	0%	0%	0%	0%	0%	0%	0%	0%
		<b>0%</b>							

De la matriz de análisis de resultados anterior se puede generar la Figura N° 39 donde se puede observar cada uno de los sistemas analizados y a simple vista se puede observar que el sistema Mis Décimos que utilizó la metodología Scrum tiene el porcentaje más alto que el de los otros sistemas.

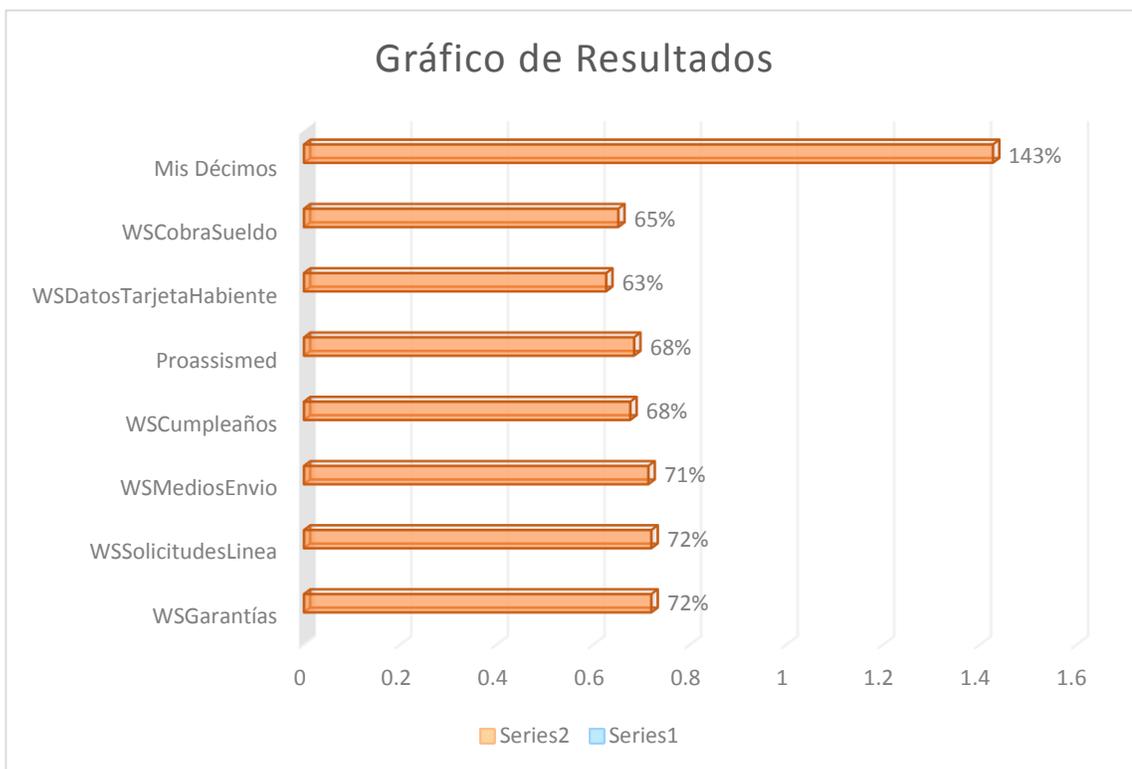


Figura 43 Resultado entre proyectos.

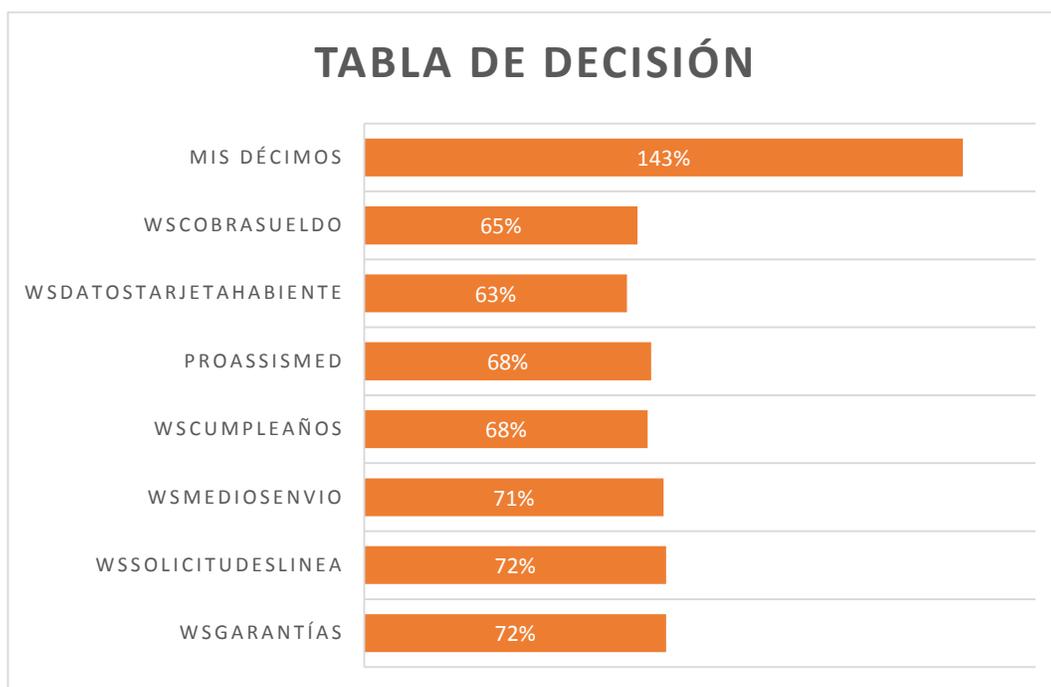


Figura 44 Resultados del análisis.

Es necesario indicar que en el desarrollo del módulo MIS DÉCIMOS se obtuvo mejores resultados en aspectos como los requisitos, los entregables, las pruebas del módulo y el tiempo de entrega de dicho sistema como se puede observar las figuras de la 41 a la 44.

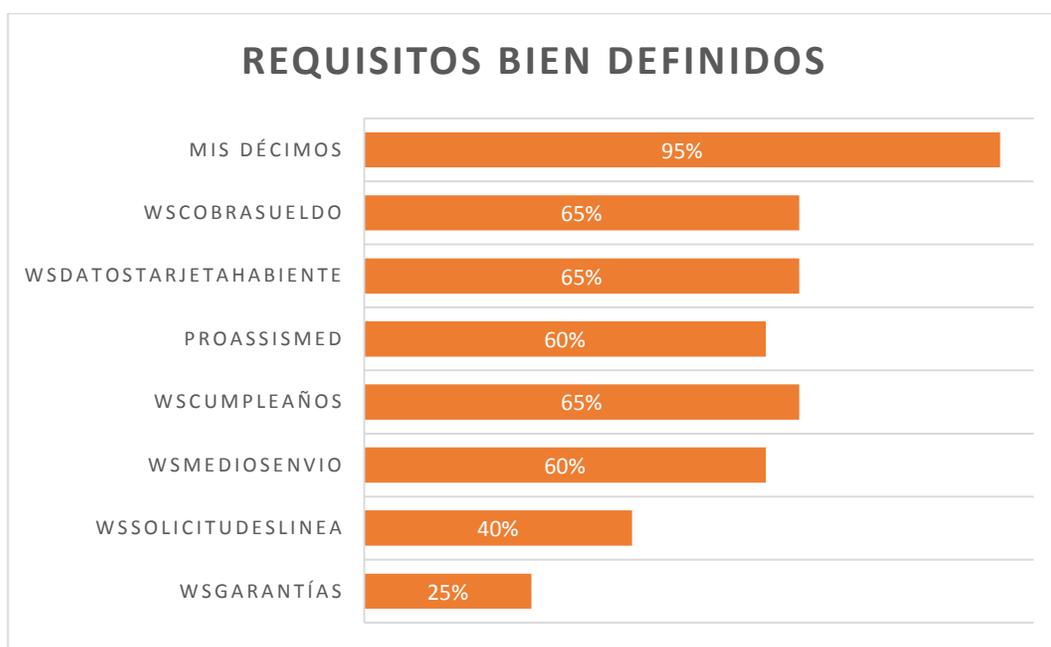


Figura 45 Resultados requisitos.

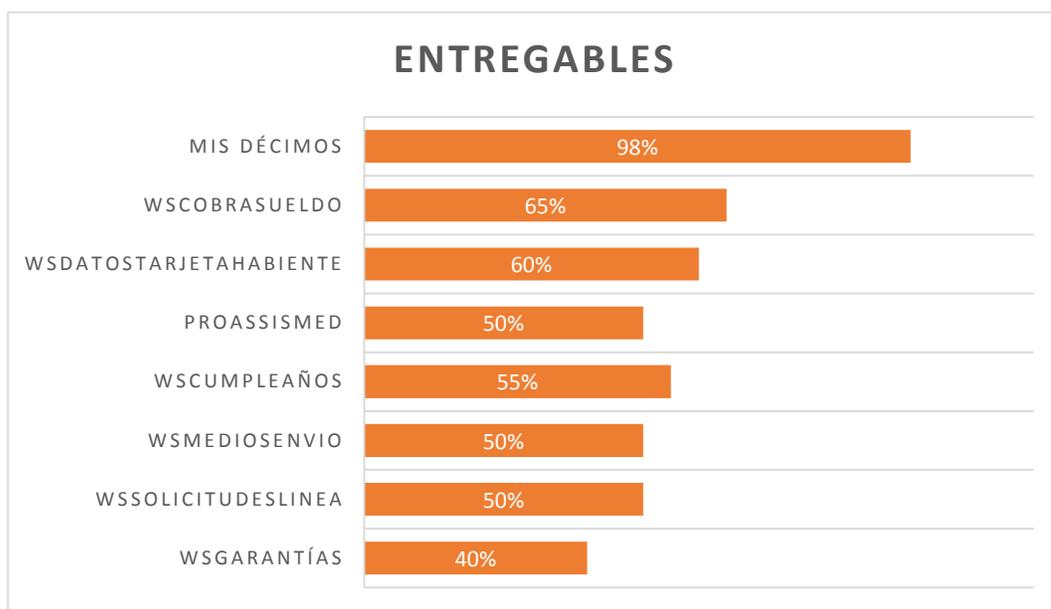


Figura 46 Resultado Entregables.

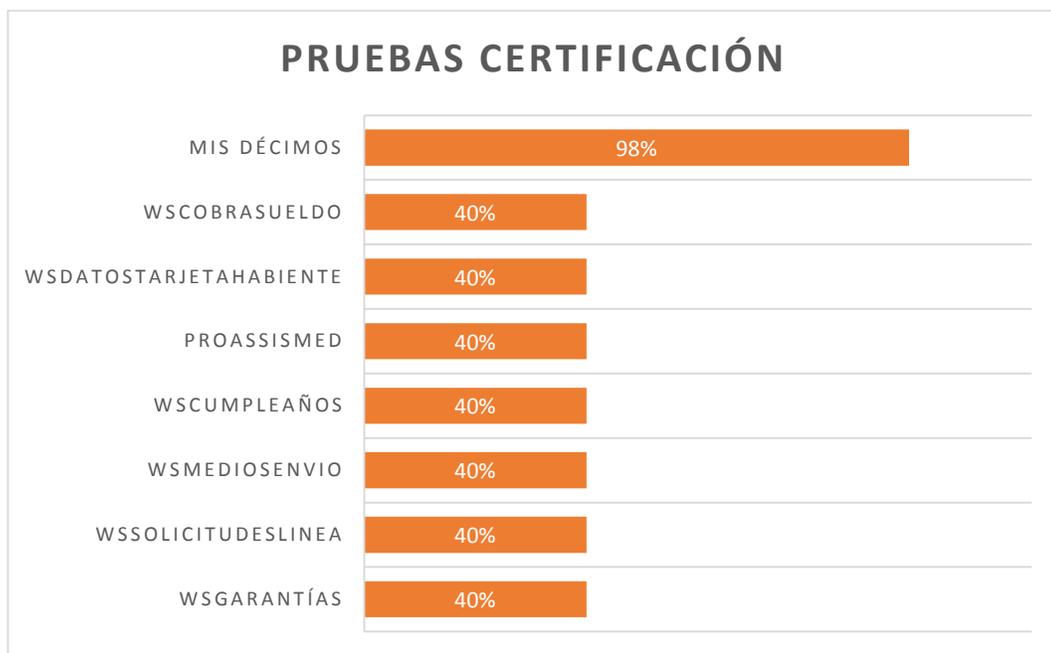


Figura 47 Resultado Pruebas.

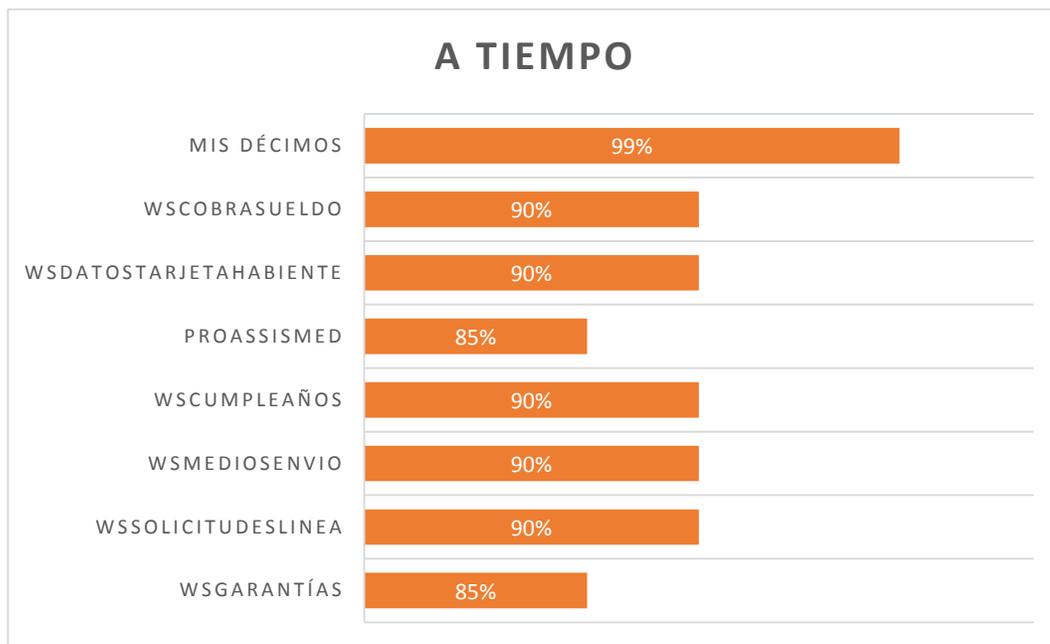


Figura 48 Resultado Tiempo.

De acuerdo a los resultados obtenidos se puede concluir que:

- El desarrollo de un sistema sin la utilización de una metodología genera inconvenientes como requisitos mal planteados, incumplimiento de los tiempos estipulados y disgusto con el usuario.
- Una metodología de desarrollo de software acorde a las necesidades de la empresa y acorde a las características de sus proyectos generan sistemas de acuerdo a lo que el cliente necesita, creando confianza y satisfacción del producto obtenido.
- La construcción de un módulo aplicando una metodología de desarrollo, genera mayor satisfacción al cliente.

## **CAPÍTULO V. CONCLUSIONES, RECOMENDACIONES Y LÍNEAS FUTURAS.**

### **5.1. CONCLUSIONES.**

- Se determinó que la CPN, no aplicaba las etapas de un producto software a detalle (Fases bien definidas, Ciclo de vida, Roles, Técnicas, Herramientas, Entregables, Criterios de Calidad), generando un problema para el departamento de sistemas de la CPN ya que no podía llevar de manera correcta el desarrollo de sus aplicaciones, generando inconvenientes con los usuarios.
- Se pudo realizar un análisis cualitativo de cinco metodologías de desarrollo de software existentes y al final se pudo obtener un modelo de selección basado en resultados.
- La metodología seleccionada (SCRUM), permitió al equipo de desarrollo priorizar los módulos que son de mayor valor para negocio y a la organización de una manera iterativa, recibiendo constante retroalimentación del área de negocio para adaptar la construcción del producto a las cambiantes necesidades del proyecto.
- Se pudo comprobar que en la construcción de la aplicación (MIS DÉCIMOS) utilizando la metodología de desarrollo SCRUM se generó mejor satisfacción al cliente ahorrando tiempo, comparado con desarrollos similares que no hacían uso de alguna metodología, lo cual fue un estímulo en la motivación del equipo de desarrollo.

### **5.2. RECOMENDACIONES**

- Toda empresa que desarrolle aplicaciones debe tener establecida una metodología de desarrollo software.

- Cada empresa/equipo de trabajo debe escoger la metodología que más se acople a sus características y con mayor énfasis a las necesidades de sus proyectos.
- Al seleccionar una metodología de desarrollo de software, ésta se debe llevar a cabo con personas de experiencia, puesto que existen varias en el mercado con sus respectivas características y ventajas, que finalmente pueden ser un factor determinante en la consecución de un proyecto.
- Las características a ser evaluadas en una metodología de desarrollo de software pueden ser creadas por el jefe de proyectos en función a los factores que puedan tener los proyectos, en base a su experiencia personal y principalmente basado en las normas e indicadores propios de cada metodología.

### **5.3. LÍNEAS FUTURAS.**

- A pesar de que los objetivos planteados con la realización de esta investigación fueron cumplidos, se recomienda tomar esta propuesta solo como la primera fase de un estudio mucho más ambicioso y continuar esta investigación.
- Otra alternativa sería buscar nuevas características, propiedades de las metodologías que no se han contemplado en la investigación realizada y que pueden tener un peso importante a la hora de elegir una metodología u otra.

## **GLOSARIO DE TÉRMINOS**

### **CASE**

(Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas o programas informáticos destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

### **CMS**

Es un repositorio en el cual se almacena toda la información del proyecto como ser documentación, código fuente, etc.

### **CPN**

Cooperativa de Ahorro y Crédito Policía Nacional Ltda.

### **DSDM**

Método de desarrollo de sistemas dinámicos.

### **Esfuerzo**

El trabajo realizado por una o varias personas para concluir una tarea o cumplir una meta a pesar de las dificultades que se encuentren en el camino.

### **Feature**

Son pequeñas funcionalidades que el cliente quiere.

### **Feature List**

Lista que agrupa toda la funcionalidad del sistema.

### **FMI**

Modelo Funcional de Iteraciones

**FODA**

Fortalezas, Oportunidades, Debilidades y Amenazas.

**GTT**

Gestión de Tecnología y Telecomunicaciones.

**ITIL**

Information Technology Infrastructure Library (Biblioteca de Infraestructura de Tecnologías de Información).

**KPI**

Key Performance Indicators (Indicador clave de desempeño).

**RFD**

Requerimientos de Desarrollo.

**Walkthrough**

Es una forma de revisar el software por pareja. Walkthrough se diferencia de una revisión técnica del software en la forma de inspección pues mencionamos que se interactúa directamente con el código.

**XP**

EXtreme Programming (Programación Extrema).

## BIBLIOGRAFÍA

- Universidad Unión Bolivariana. (12 de 05 de 2015). FDD. Obtenido de [http://ingenieriadesoftware.mex.tl/61162\\_FDD.html](http://ingenieriadesoftware.mex.tl/61162_FDD.html)
- Agile Alliance. (10 de Febrero de 2015). Agile software community . Obtenido de <https://www.agilealliance.org/>
- Ágiles Colombia. (19 de Mayo de 2015). Ágiles Colombia. Obtenido de Sitio web de Ágiles Colombia: <http://agilescolombia.org/>
- Android, Entrepreneur, IT, Management. (29 de Diciembre de 2013). Mariouz blog. Obtenido de sitio web de Mariouz blog: <https://mariouzblog.wordpress.com/>
- Bacca, A. (DICIEMBRE de 2014). Desarrollo de software en proyectos de telecomunicaciones de corta duración. BOGOTÁ, BOGOTÁ D.C, COLOMBIA.
- Comunidad Latinoamericana de Metodologías Ágiles. (19 de Mayo de 2015). Comunidad Latinoamericana de Metodologías Ágiles. Obtenido de Sitio web de Comunidad Latinoamericana de Metodologías Ágiles: <http://www.agiles.org/argentina>
- Cooperativa Policia Nacional. (22 de 02 de 2013). Estatuto. Quito, Pichincha, Ecuador.
- Cooperativa Policía Nacional. (21 de 04 de 2015). BALANCE SOCIAL COOPERATIVO. Obtenido de <http://www.cooperando.fin.ec/images/pdf/BALANCE%20SOCIAL.pdf>
- Cooperativa Policia Nacional. (21 de 04 de 2015). Cooperativa Policia Nacional. Obtenido de <http://www.cooperando.fin.ec/index.php/2014-01-30-19-41-51>
- Cooperativa Policia Nacional. (21 de 04 de 2015). Cooperativa Policia Nacional Valores. Obtenido de <http://www.cooperando.fin.ec/index.php/2014-01-30-19-43-56/principios-cooperativos>
- Definición.de. (28 de Junio de 2015). Definición.de. Obtenido de Definición.de: <http://definicion.de/cronograma/>
- DSDM Consortium. (12 de Noviembre de 2014). Driving Strategy Delivering More. Obtenido de El Marco DSDM Ágil de Proyectos: <http://www.dsdm.org/dig-deeper/book/dsdm-agile-project-framework>

- EcuRed. (15 de Mayo de 2015). EcuRed. Obtenido de EcuRed: [http://www.ecured.cu/index.php/Metodolog%C3%ADa\\_%C3%A1gil](http://www.ecured.cu/index.php/Metodolog%C3%ADa_%C3%A1gil)
- Escuela Superior Politécnica del Litoral (ESPOL). (2 de Febrero de 2014). Revista Tecnológica ESPOL - ISSN 1390-3659. Obtenido de Sitio web Revista Tecnológica ESPOL: <http://www.rte.espol.edu.ec/index.php/tecnologica/article/view/312>
- Espina, H. A. (2010). Propuesta Metodológica para el Desarrollo de Software de Investigación. Santiago, Chile, Chile.
- Feature Driven Development - Practices. (19 de 09 de 2015). Obtenido de [http://www.pearsonhighered.com/assets/hip/us/hip\\_us\\_pearsonhighered/samplechapter/0130676152.pdf](http://www.pearsonhighered.com/assets/hip/us/hip_us_pearsonhighered/samplechapter/0130676152.pdf)
- Fundación Wikimedia, Inc. (17 de Mayo de 2015). Manifiesto ágil. Obtenido de Sitio web de Wikipedia: [http://es.wikipedia.org/wiki/Manifiesto\\_%C3%A1gil](http://es.wikipedia.org/wiki/Manifiesto_%C3%A1gil)
- Fundación Wikimedia, Inc. (18 de Mayo de 2015). Programación extrema. Obtenido de Sitio web de Wikipedia: [http://es.wikipedia.org/wiki/Programaci%C3%B3n\\_extrema](http://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema)
- fundamentosagiles.org. (06 de 07 de 2015). Scrum. Obtenido de <http://www.proyectosagiles.org/que-es-scrum>
- Godoy Alvarez, D. A., & Taype Calderón, R. S. (2015). Modelos De Aceptación De Metodologías De Desarrollo De Software. LIMA, PERÚ.
- Gómez Fuentes, M. d. (MARZO de 2012). Taxonomía de los modelos y metodologías de desarrollo de software más utilizados. México DF, México.
- Highsmith, J. (25 de Julio de 2006). Agile Project Management . Obtenido de Adaptive Software Development: <http://www.adaptivesd.com/>
- Ing. Software. (7 de Agosto de 2012). METODOLOGIA XP. Obtenido de IUTY 072301 PNF en Informática: <http://ingsoftware072301.obolog.es/metodologia-xp-2012877>
- Introducción a Visual Studio. (14 de 11 de 2007). Obtenido de Introducción a Visual Studio: [https://msdn.microsoft.com/es-es/library/fx6bk1f4\(v=vs.90\).aspx](https://msdn.microsoft.com/es-es/library/fx6bk1f4(v=vs.90).aspx)

- IT Infrastructure Library. (4 de Agosto de 2013). Métricas ITIL - KPIs ITIL. Obtenido de AXELOS Limited: [http://wiki.es.it-processmaps.com/index.php/M%C3%A9tricas\\_ITIL\\_-\\_KPIs\\_ITIL](http://wiki.es.it-processmaps.com/index.php/M%C3%A9tricas_ITIL_-_KPIs_ITIL)
- Jack Herrick. (13 de Marzo de 2015). wikiHow. Obtenido de Cómo hacer una revisión literaria: <http://es.wikihow.com/hacer-una-revisi%C3%B3n-literaria>
- Joaquín, B. A. (DICIEMBRE de 2013). Metodologías para desarrollar software seguro. Zacatecas, México.
- LOGIC. (06 de 07 de 2015). SCRUM: metodología “ágil” para tus proyectos. Obtenido de <http://logic.com/scrum-metodologia-agil-para-tus-proyectos/>
- MANAGER, S. (04 de 05 de 2015). SCRUM MANAGER. Obtenido de <http://www.scrummanager.net/>
- Martinez Fustero Eduardo, L. 8. (14 de 02 de 2014). Las 8 grandes ventajas de las metodologías ágiles. Recuperado el 2 de 12 de 2014, de Las 8 grandes ventajas de las metodologías ágiles: <http://comunidad.iebschool.com/iebs/agile-scrum/que-es-agile/>
- Medina Velandia, L. N., Escobar Escobar, Á., & Arenas Prada, A. A. (2 de JUNIO de 2009). Método para el desarrollo de proyectos de software. Bogotá,, Colombia.
- Ministerio de Industria, TInstituto Nacional de Tecnologías de la Comunicación. (2009). Ingeniería del software: Metodologías y ciclos de vida. España.
- Murillo, C. (12 de Marzo de 2013). RECONOCIMIENTO GENERAL Y DE ACTORES. Obtenido de Ciclos de vida del software web: <http://reconocimientogeneralydeactores.blogspot.com/>
- NEBULON. (19 de 09 de 2015). FDD. Obtenido de <http://www.nebulon.com/articles/fdd/download/fddprocessesA4.pdf>
- Nebulon Pty. Ltd. . (01 de Noviembre de 2012). Feature Driven Development. Obtenido de <http://www.featuredrivendevelopment.com/>
- Palacio, J. (2014). Scrum Manager I. Scrum Manager.
- PALLAROSO, M. A. (22 de 07 de 2015). Aplicar la metodología de desarrollo ágil FDD. Obtenido de repositorio.espe.edu.ec/bitstream/21000/5297/2/T-ESPE-033275-P.pptx

- Pérez, O. A. (1 de Abril de 2011). Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM. Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM.
- Piña, P. L. (15 de Enero de 2015). METODOS DE DESARROLLO DE SISTEMAS DINAMICOS. Obtenido de <https://ingsfotwarebi.files.wordpress.com/2015/03/metodo-de-desarrollo-de-sistemas-dinamicos.docx>.
- Poppendieck, T. (21 de Octubre de 2015). The Lean Mindset. Obtenido de <http://www.poppendieck.com/>
- Pressman, R. (1997). Ingeniería del Software: Un enfoque práctico. McGraw Hill.
- Ramírez, I. D. (5 de DICIEMBRE de 2008). METODOLOGÍAS ÁGILES. HABANA, CUBA.
- Rengifo, A. (20 de Noviembre de 2014). Portafolio Ingeniería del Software. Obtenido de Web de Portafolio Ingeniería del Software: [http://arengifoingsoft.blogspot.com/2014\\_11\\_01\\_archive.html](http://arengifoingsoft.blogspot.com/2014_11_01_archive.html)
- Rica, U. L. (11 de 05 de 2015). ASD. Obtenido de <https://www.academia.edu/5002829/ASD>
- RIEHLE, D. (7 de Septiembre de 2007). A Comparison of the Value Systems of Adaptive Software. Obtenido de How Methodologies May: <http://www.riehle.org/computer-science/research/2000/xp-2000.html>
- Rosale, R. A. (25 de Julio de 2012). MÉTODO DE DESARROLLO DE SISTEMAS DINÁMICOS “DSDM”. Obtenido de utmfci: <https://sites.google.com/site/utmfci/system/app/pages/recentChanges>
- SADIO. (12 de Febrero de 2015). Sociedad Argentina de Informática e Investigación Operativa. Obtenido de Sociedad Argentina de Informática e Investigación Operativa: <http://www.sadio.org.ar/>
- SAMANIEGO, M. (03 de 10 de 2015). SISTEMA WEB GRANJA AVICOLA USANDO METODOLOGIA FDD. Obtenido de <http://repositorio.espe.edu.ec/bitstream/21000/5297/1/T-ESPE-033275.pdf>
- Schwaber, K. (22 de Agosto de 2010). SOFTWARE IN 30 DAYS. Obtenido de <http://www.controlchaos.com/>

- Sicilia, M. A. (06 de 05 de 2016). ISO 9126. Obtenido de <http://garciagregorio.webcindario.com/ms/iso9126.pdf>
- SlidePlayer.es.Inc. (12 de 05 de 2015). XP Extreme Programming. Obtenido de SlidePlayer.es.Inc.: <http://antares.itmorelia.edu.mx/~kaos/http://www.xumarhu.net/>
- Universidad Unión Bolivariana. (16 de Febrero de 2015). METODO AGIL. Obtenido de ASD (ADAPTIVE SOFTWARE DEVELOPMENT): [http://ingenieriadesoftware.mex.tl/61154\\_ASD.html](http://ingenieriadesoftware.mex.tl/61154_ASD.html)
- Uruguay, U. O. (10 de 06 de 2015). Metodología FDD. Obtenido de <http://fi.ort.edu.uy/innovaportal/file/2021/1/metodologiafdd.pdf>
- Victoria, P., & Guapas, M. (21 de 04 de 2015). CATALOGO DE SERVICIOS TECNOLOGICOS GTT. Quito, Pichincha, Ecuador.
- Victoria, P., & Guapas, M. (21 de 04 de 2015). MANUAL DE LA GESTIÓN DE TECNOLOGÍA INFORMATICA Y TELECOMUNICACIONES. Quito, Pichincha, Ecuador.
- Wells, D. (08 de Octubre de 2013). Extreme Programming. Obtenido de <http://www.extremeprogramming.org/>
- Wirwin. (17 de 07 de 2015). Metodología SCRUM para la dirección de proyectos informáticos. Obtenido de <https://ejecucion.wordpress.com/2009/06/10/metodologia-scrum-para-la-direccion-de-proyectos-informaticos/>