

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

“ANÁLISIS E IMPLEMENTACIÓN DE GESTIÓN DE PROCESOS DE NEGOCIOS (BPM) Y METODOLOGÍAS ÁGILES PARA OPTIMIZAR EL DESEMPEÑO FUNCIONAL DEL PROCESO DE DESARROLLO DE SOFTWARE, EN UNA EMPRESA PRIVADA, DE SOLUCIONES TECNOLÓGICAS.”

TRABAJO DE TITULACIÓN

PREVIO A LA OBTENCIÓN DEL TÍTULO

MAGISTER EN SISTEMAS DE INFORMACIÓN GERENCIAL

ING. DIANA AURORA CHIQUITO PEÑARANDA

GUAYAQUIL – ECUADOR

AÑO 2017

AGRADECIMIENTO

Ante todo agradezco a Dios por todas sus bondades, por permitirme vivir plenamente cada una de las etapas por las que he pasado.

A mis .padres por estar siempre pendiente de mis planes y brindarme su apoyo incondicional, por creer en mí, darme consejos y ánimos para avanzar y superarme.

A mis hijos, porque tengo en ellos un motivo más para no renunciar a las metas planteadas, hacen de mí una mejor persona y me enseñan a valorar lo realmente importante.

Un agradecimiento especial a mi tutor de tesis, el PhD Carlos Monsalve por el tiempo, su paciencia, el apoyo y la enseñanza ofrecida durante este proceso.

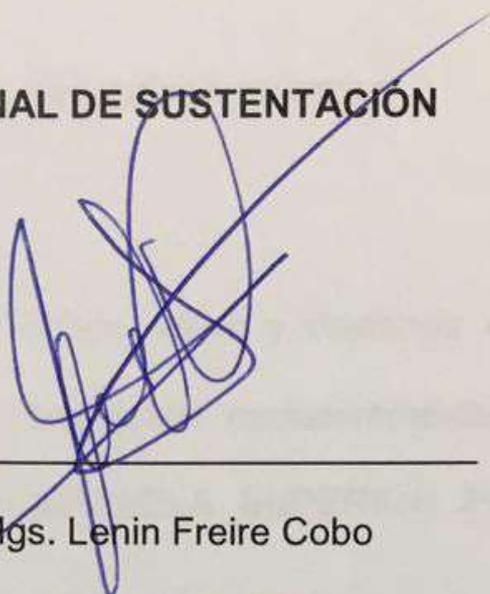
DEDICATORIA

A mis padres Andrés y Aurora porque nunca permitieron que renuncie a mis sueños y sin su apoyo hubiese sido más difícil el camino.

A David por brindarme su amor, su apoyo y estar pendiente de mis avances.

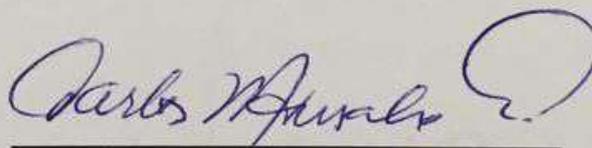
A mis tres pequeños tesoros Ethan, Valentina y Andrea quienes son el motor de mi vida, mi alegría e inspiración, solo espero ser un buen ejemplo para ellos.

TRIBUNAL DE SUSTENTACION



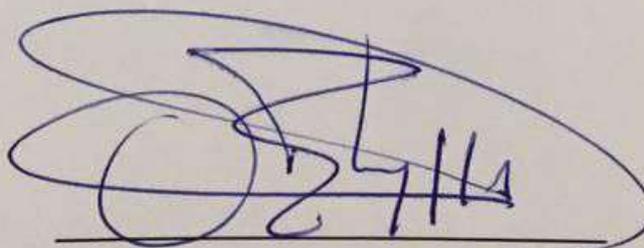
Mgs. Lenin Freire Cobo

DIRECTOR MSIG



Ph.D. Carlos Monsalve Arteaga

DIRECTOR DE PROYECTO DE TITULACIÓN



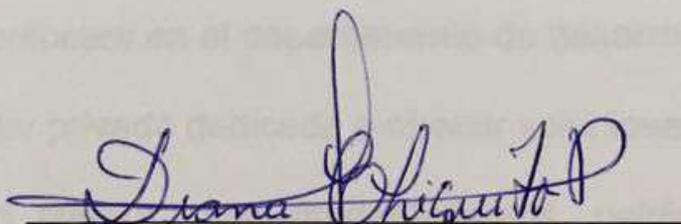
Mgs. Omar Maldonado Dañin

MIEMBRO DEL TRIBUNAL

DECLARACIÓN EXPRESA

“La responsabilidad por los hechos, ideas y doctrinas expuestas en este trabajo de titulación me corresponden exclusivamente, y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de graduación de la ESPOL)



Diana Aurora Chiquito Peñaranda

RESUMEN

El presente trabajo se concentra en el análisis e implementación de gestión de procesos de negocios apoyados con metodologías ágiles para optimizar el desempeño funcional del proceso de desarrollo de software de una empresa privada, que ofrece soluciones tecnológicas, estableciendo de esta manera en la empresa modelos de procedimientos guías a lo largo de la cadena de valor.

Este estudio se enfocará en el departamento de desarrollo de software de una empresa del sector privado dedicada a ofrecer soluciones tecnológicas, la cual actualmente no cuenta con procedimientos, políticas, ni metodologías definidas formalmente, razón por la cual nace esta necesidad.

Para corregir esta situación se plantea optimizar los procesos que se llevan a cabo durante el desarrollo de software; partiendo con el análisis respectivo, para luego alinear la operación con la estrategia de negocio y dar seguimiento a todas las actividades realizadas. Se involucrará durante el proceso de optimización a todos los interesados, trabajando de manera colaborativa, comunicativa y eficiente. Así se evitarán ambigüedades y se minimizarán los riesgos que se puedan presentar a lo largo del proceso, generando mayor

agilidad en el negocio con el fin de responder mejor a las condiciones cambiantes del mercado.

ÍNDICE GENERAL

AGRADECIMIENTO	i
DEDICATORIA	ii
TRIBUNAL DE SUSTENTACIÓN	iii
DECLARACIÓN EXPRESA	iv
RESUMEN	v
ÍNDICE GENERAL.....	vii
ABREVIATURAS Y SÍMBOLOS	xii
ÍNDICE DE TABLAS	xiii
ÍNDICE DE FIGURAS.....	xiv
INTRODUCCIÓN	xix
CAPÍTULO 1	1
GENERALIDADES	1
1.1. ANTECEDENTES.....	1
1.1. DESCRIPCIÓN DEL PROBLEMA.....	2
1.2. SOLUCIÓN PROPUESTA	4
1.3. OBJETIVO GENERAL	6
1.4. OBJETIVOS ESPECÍFICOS.....	6
1.5. METODOLOGÍA	7
CAPÍTULO 2.....	9
MARCO TEÓRICO	9
2.1. BUSINESS PROCESS MANAGEMENT (BPM)	9

2.1.1. FUNDAMENTACIÓN TEÓRICA	9
2.1.2. CICLO DE VIDA DEL PROCESO DE NEGOCIO	11
2.1.3. MODELADO DE PROCESOS DE NEGOCIO EN BPMN	15
2.1.4. ELEMENTOS BÁSICOS UTILIZADOS EN BPMN 2.0.....	16
2.1.5. BUENAS PRÁCTICAS Y PATRONES DE MODELADO	19
2.2. TIPOS DE METODOLOGÍAS PARA LA DIRECCIÓN DE PROYECTOS DE DESARROLLO DE SOFTWARE.....	21
2.2.1. METODOLOGÍAS CLÁSICAS O TRADICIONALES.....	24
2.2.1. METODOLOGÍAS ÁGILES	34
2.2.3. COMPARACIÓN ENTRE METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES.....	42
2.3. METODOLOGÍA SCRUM.....	44
2.3.1. COMPONENTES DE SCRUM.....	45
2.3.2. BENEFICIOS DE UTILIZAR LA METODOLOGÍA SCRUM	56
CAPÍTULO 3.....	58
LEVANTAMIENTO DE INFORMACIÓN	58
3.1. ANÁLISIS DE LA SITUACIÓN ACTUAL DE LA EMPRESA.....	58
3.1.1. FINES Y FUNCIONES.....	58
3.1.2. VISIÓN.....	59
3.1.3. MISIÓN.....	59
3.1.4. ORGANIGRAMA DE LA EMPRESA.....	60
3.2. ANÁLISIS DE LA SITUACIÓN DEL ÁREA OPERATIVA.....	62

3.1.1. ANÁLISIS FODA.....	63
3.1.2. CLIENTES Y USUARIOS DE LA EMPRESA.....	64
3.1.3. ESTADÍSTICA DE LOS REQUERIMIENTOS INGRESADOS AL AREA OPERATIVA	66
3.3. ANÁLISIS DE PROCESOS DEL ÁREA OPERATIVA	72
3.3.1. PROCESOS DEL ÁREA OPERATIVA	72
3.3.2. IDENTIFICACIÓN DE LOS PRINCIPALES PARTICIPANTES DE LOS PROCESO DEL ÁREA OPERATIVA.....	84
CAPÍTULO 4.....	87
ANÁLISIS, DISEÑO DE LA PROPUESTA DE IMPLEMENTACIÓN DE GESTIÓN DE PROCESOS DE NEGOCIOS CON METODOLOGÍAS ÁGILES.....	87
4.1. GESTIÓN DE PROCESOS DE NEGOCIOS APLICADO AL DESARROLLO Y MANTENIMIENTO DE SOFTWARE.....	87
4.1.1. DIAGRAMA BPMN DEL PROCESO DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE.....	88
4.1.2. DEFINICIÓN DEL MODELO DE DATOS.....	89
4.1.3. DISEÑO Y GENERACIÓN DE LAS INTERFACES DE USUARIO	98
4.1.4. DEFINICIÓN DE LAS REGLAS DE NEGOCIOS.....	106
4.1.5. DEFINICIÓN DE PARTICIPANTES Y SUS ROLES	115
4.1.6. EJECUCIÓN DEL PROCESO	117

4.2. ANÁLISIS DE LA METODOLOGÍA ÁGIL SCRUM EN FUNCIÓN DEL PROCESO LEVANTADO	123
4.2.1. RELACIÓN DE LAS ACTIVIDADES A REALIZAR DURANTE EL PROCESO DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE CON SCRUM	123
4.2.2. PASOS A SEGUIR PARA IMPLEMENTAR SCRUM EN EL PROCESO DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE.....	126
CAPITULO 5.....	128
IMPLEMENTACIÓN DE LA PROPUESTA	128
5.1. IMPLEMENTACIÓN DE PLAN DE PRUEBAS	128
5.2. IMPLEMENTACIÓN DE LAS POLÍTICAS Y PROCEDIMIENTOS ...	131
5.2.1. POLÍTICAS REFERENTES AL DESARROLLO DE SOFTWARE	131
5.2.2. PROCEDIMIENTOS	138
5.3. IMPLEMENTACIÓN DE INDICADORES EN EL PROCESO DE GESTIÓN DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE.....	141
CAPÍTULO 6.....	144
ANÁLISIS DE RESULTADOS.....	144
6.1. ANÁLISIS DE LOS RESULTADOS DE LA PROPUESTA IMPLEMENTADOS EN UN PROYECTO EN MARCHA	144
6.1.1 GESTIÓN DE REGISTRO DEL REQUERIMIENTO.....	145
6.1.2. GESTIÓN DE PLANIFICACIÓN	146
6.1.3. GESTIÓN DE DESARROLLO	149

6.1.4. GESTIÓN DE PRUEBAS.....	151
6.1.5. GESTIÓN DE LIBERACIÓN DE PRODUCTO.....	151
CONCLUSIONES Y RECOMENDACIONES.....	153
BIBLIOGRAFÍA.....	157
ANEXO 1: DIAGRAMA DEL PROCESO DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE.....	160
ANEXO 2: ACTA DE REUNIÓN DE SPRINT INICIAL.....	161
ANEXO 3: DOCUMENTO FUNCIONAL.....	163
ANEXO 4: DOCUMENTO TÉCNICO.....	165
ANEXO 5: CASOS DE PRUEBAS FUNCIONALES Y TÉCNICAS.....	166

ABREVIATURAS Y SÍMBOLOS

ATS	Anexo Transaccional Simplificado
BPD	Business Process Diagram
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Model and Notation
COTS	Commercial off the shelf
DBA	Database Administrator – Administrador de Base de Datos
ERP	Enterprise Resource Planning
INEC	Instituto Ecuatoriano de Estadísticas y Censos
MRL	Ministerio de Relaciones Laborales
OT	Orden de Trabajo
RUT	Registro Universal de Trabajo
SGC	Sistema de Gestión de Calidad
SRI	Servicios de Rentas Internas
VSS	Visual Sourcesafe

ÍNDICE DE TABLAS

Tabla 1: Elementos de BPMN [7].....	18
Tabla 2: Los doce principios del manifiesto ágil.....	37
Tabla 3: Principales metodologías ágiles.....	41
Tabla 4: Comparación entre metodologías tradicionales vs. metodologías ágiles	43
Tabla 5: Análisis FODA del Área Operativa.....	64
Tabla 6: Número de clientes que mantiene la empresa clasificadas según su tamaño.....	65
Tabla 7: Registro de RUT	90
Tabla 8: Órdenes de Trabajo	92
Tabla 9: Actividades a realizar durante el ciclo de desarrollo de software relacionadas con Scrum	124
Tabla 10: Objetivos, indicadores y métricas	141
Tabla 11: Conformación del equipo SCRUM	148

ÍNDICE DE FIGURAS

Figura 2.1: Actividad de un proceso de Negocio.....	10
Figura 2.2: Descripción de las fases de Ciclo de Vida del Proceso adaptado de [4].....	14
Figura 2.3: Metodología Clásica – Modelo en Cascada [11].....	25
Figura 2.4: Metodología Clásica – Modelo Incremental [11].....	27
Figura 2.5: Metodología Clásica – Modelo Evolutivo [11].....	29
Figura 2.6: Metodología Clásica – Modelo Espiral [11].....	31
Figura 2.7: Metodología Clásica – Desarrollo basado en componentes [11]	33
Figura 2.8: Manifiesto ágil [12].....	36
Figura 2.9: Ciclo de vida del desarrollo ágil	39
Figura 2.10: Descripción conceptual de un ciclo SCRUM [11].....	45
Figura 2.11: Ejemplo con formato de un product backlog adaptado de [13].	49
Figura 2.12: Formato de historia de usuario [13]	50

Figura 2.13: Ejemplo con formato de un Sprint Backlog [13].....	51
Figura 2.14: Tablero de Tareas [13].....	52
Figura 2.15: Burn-Down Chart adaptado de [13]	53
Figura 3.1: Organigrama de la empresa	62
Figura 3.2: Estadística de ingresos y cierres de requerimientos.....	67
Figura 3.3: Estadística de origen de requerimientos.....	69
Figura 3.4: Clasificación de los requerimientos registrados.....	70
Figura 3.5. Composición del Comité Técnico.....	73
Figura 3.6. Composición del Comité de Calidad.....	73
Figura. 4.1: Modelo de Datos.....	89
Figura. 4.2: Registro de RUT	91
Figura. 4.3: Órdenes de Trabajo.....	93
Figura 4.4: Cliente.....	94
Figura 4.5: Tipo de Requerimiento.....	94

Figura 4.6: Prioridad	95
Figura 4.7: Origen de Requerimientos	95
Figura 4.8: Recursos.....	96
Figura 4.9: Funciones	96
Figura 4.10: Tipo de Análisis	97
Figura 4.11: Sprint	97
Figura 4.12: Registro Universal de Trabajo	99
Figura 4.13: Análisis de Factibilidad de RUT	100
Figura 4.14: Informe de NO factibilidad del RUT	101
Figura 4.15: Asignación de Prioridad	102
Figura 4.16: Registro de Orden de trabajo.....	103
Figura 4.17: Registro de informe generado en reunión de Sprint Inicial	104
Figura 4.18: Registro de Documentación.....	105
Figura 4.19: Registro de Aprobación – Observaciones.....	106

Figura 4.20: Definición de Reglas de Negocio	107
Figura 4.21: Factibilidad del requerimiento	108
Figura 4.22: Validación de Documento Funcional.....	109
Figura 4.23: Validación de Pruebas Unitarias	110
Figura 4.24: Validación de Pruebas Integrales	111
Figura 4.25: Validar pase a Certificación	112
Figura 4.26: Validar Ejecución de pase a Certificación.....	113
Figura 4.27: Ejecutar pruebas con usuarios	114
Figura 4.28: Validar Certificación	115
Figura 4.29: Participantes y roles.....	116
Figura 4.30: Ejecución de proceso.....	117
Figura 4.31: Ingreso a la ejecución del proceso.....	118
Figura 4.32: Registro de RUT	118
Figura 4.33: Informe de factibilidad.....	119

Figura 4.34: Rechazo de Orden – Termino del Proceso.....	119
Figura 4.35: Asignación de Prioridad	120
Figura 4.36: Asignación de Recursos	120
Figura 4.37: Registro de Informe de reunión de sprint inicial.....	121
Figura 4.38: Registro de documentación	121
Figura 4.39: Aprobación de Documentos.....	122
Figura 4.40: Registro de Procesos Pendientes por el usuario	122
Figura 4.41: Matriz de pruebas técnicas aplicadas durante cada Sprint....	130
Figura 4.42: Matriz de pruebas funcionales aplicadas durante cada Sprint	130
Figura 6.1: Solicitud enviada por el Cliente.....	145
Figura 6.2: Planificación Inicial: Asignación de recursos y órdenes de trabajo	147

INTRODUCCIÓN

En la actualidad las organizaciones dedicadas a ofrecer soluciones tecnológicas ven la necesidad de controlar y mejorar la calidad del software que ofrecen a sus clientes/usuarios. La falta de controles durante el proceso de desarrollo tiene luego un gran impacto que puede ocasionar problemas en la fase de producción tales como retrasos en la entrega y defectos. Esto puede repercutir en la insatisfacción y/o pérdida del cliente y por lo tanto puede existir una afectación económica para la empresa.

La presente propuesta de titulación, tiene como objetivo implementar en el área de desarrollo de software de una empresa privada que ofrece soluciones tecnológicas, una gestión de procesos de negocios (BPM), complementándola con el uso de metodologías ágiles. La combinación de ambas (BPM y metodologías ágiles) permitirá tener procesos optimizados, eliminando tareas que no aporten valor. De esta manera se utilizarán los recursos disponibles (tiempo, personal, costos, etc.) de manera eficiente. Un proceso de negocio por más adecuado y bien implementado que se encuentre siempre estará susceptible para mejorarlo y optimizarlo, y las razones para que esto se de bien pueden ser internas o externas a la organización.

Las metodologías ágiles requieren de una organización sencilla del equipo de trabajo, generando que el equipo se involucre en la consecución del objetivo. Adicionalmente, las metodologías ágiles integran al cliente durante el desarrollo, de esta forma el cliente puede aportar con sus ideas y comentarios, ajustando el entregable a sus necesidades.

Como punto de partida se identificará la situación actual del proceso a optimizar, a continuación se realizará la recolección de la información que servirá para establecer indicadores de monitorización que posteriormente permitan analizar y evaluar tiempos, retrasos, progresos, desviaciones, costos y rendimiento del proceso durante su ejecución. Como resultado del análisis obtendremos los aspectos que se deben mejorar, se rediseñará el proceso actual con las respectivas mejoras, y se documentará la propuesta. Finalmente, se procederá a automatizar el proceso. De esta manera se expone la importancia por la que se debe adoptar en la empresa privada que ofrece soluciones tecnológicas la gestión de procesos de negocios apoyadas con metodologías ágiles para dar soluciones de valor agregado a los negocios de la empresa.

CAPÍTULO 1

GENERALIDADES

1.1. ANTECEDENTES

Las organizaciones están interesadas en buscar metodologías y herramientas que ayuden a mejorar su desempeño y tener una mayor agilidad en los negocios; es decir responder rápidamente a los cambios previstos y no previstos. La gestión de procesos de negocios y las metodologías ágiles, son a menudo consideradas herramientas ideales para orquestar los cambios necesarios para la consecución de los objetivos de una organización [1].

Una organización está conformada por procesos y si los conoce, esta podrá modelarlos, analizarlos, medirlos y posteriormente optimizarlos.

Al integrar el modelado de los procesos de manera inteligente con las tecnologías de la información y recursos humanos que conforman los procesos se podrá conseguir un producto final más integral, consistente y con menos grietas.

La implementación de metodologías ágiles permitirá adaptar la forma de trabajo a las condiciones de los proyectos que se realicen, logrando de esta forma tener flexibilidad e inmediatez en la respuesta para amoldar los proyectos y sus desarrollos a las circunstancias específicas del entorno.

La implementación de estos cambios en la gestión del negocio ayudará a alinear la operación con la estrategia.

1.1. DESCRIPCIÓN DEL PROBLEMA

La empresa de desarrollo de software, objeto del presente trabajo de titulación, pertenece al sector privado y ofrece soluciones Informáticas tales como un ERP Administrativo-Contable, un sistema de Facturación Electrónica, Desarrollo de software bajo pedido del cliente, Servicios de

Consultoría y Capacitaciones, por medio de los cuales genera sus ingresos. Tiene alrededor de 8 años en el mercado, de los cuales en los últimos 3 ha tenido un crecimiento significativo en cuanto al talento humano contratado y al volumen de clientes en el Ecuador.

Con el crecimiento de la empresa se empieza a evidenciar un sinnúmero de problemas en cada una de las áreas, en especial en una de las más críticas, la cual será objeto de análisis: el área de sistemas, lo que podría terminar en la generación de dificultades en toda la empresa. A continuación se detallan los problemas más comunes:

Procesos:

- Requerimientos no definidos.
- Falta de mecanismos de trazabilidad del progreso del proyecto.
- No existe gestión de riesgos.
- Poca relación con los usuarios.
- Falta de documentación de los procesos.
- Retrasos en los tiempos de entregas.

Técnicas

- Falta de estándares de codificación.
- Falta de un conjunto de prácticas ágiles.
- No se contemplan todos los posibles escenarios durante las pruebas de integración.

1.2. SOLUCIÓN PROPUESTA

La industria del software representa una actividad económica que crece rápidamente en el Ecuador, siendo así que las empresas en este sector cuando inician un proyecto de desarrollo e implantación de software, persiguen como objetivo principal la entrega a los clientes/usuarios de un producto o servicio útil y de calidad dentro de un tiempo determinado.

Es ahí cuando cobra importancia el establecimiento de políticas, metodologías y la gestión de los procesos de negocios, ya que este conjunto forma parte de las llamadas “Buenas Prácticas gerenciales”. Su adaptación es un excelente medio que permite identificar oportunidades de optimizar las tareas que se realizan, permite eliminar despilfarros y actividades que no aporten valor, además de que se

busca minimizar costos, optimizar tiempos, recursos y mejorar la calidad de sus productos.

Partiendo de lo anteriormente mencionado, lo que se propone en este trabajo de titulación, es lo siguiente:

- Modelar en “Business Process Model and Notation” (BPMN) el proceso de desarrollo de software.
- Realizar el análisis del modelo levantado para ejecutar los ajustes necesarios, de tal forma que el nuevo proceso de desarrollo de software incorpore metodologías ágiles como Scrum.
- Complementar el análisis con la definición de roles, estándares de la empresa, políticas y procedimientos.
- Probar el modelado de procesos complementado con metodología ágiles en un proyecto en marcha de la empresa.

Si tenemos en cuenta el desarrollo de las habilidades del equipo, a medida que los recursos van participando en diferentes proyectos en la organización, entonces, se irá puliendo la metodología utilizada y transformando en un estilo personal. Esto no es solo algo bueno para la

organización, sino debería ser de uso obligatorio en esta, debido a la gran competitividad del mercado.

1.3. OBJETIVO GENERAL

Implementar en el área de Sistemas, metodologías ágiles, gestión de procesos de negocios, políticas y normas, con el fin de que en la empresa privada que ofrece Soluciones Tecnológicas, se pueda establecer los modelos de procedimientos con los que trabajarán en las distintas actividades que se realicen, optimizar los procesos, proponer y desarrollar opciones de mejora continua a lo largo de la cadena de valor.

1.4. OBJETIVOS ESPECÍFICOS

1. Definir el alcance, situación actual, la estructura organizativa, los procesos críticos, los recursos disponibles, roles, responsabilidades, los factores ambientales y los objetivos de la organización.
2. Definir los conceptos claves referente a Gestión de Procesos de Negocios, metodologías ágiles, así como su aplicación en las distintas actividades de la empresa.

3. Realizar el levantamiento de información, identificación de métricas, los indicadores y matrices a utilizar en la medición.
4. Analizar la información obtenida del levantamiento de información, planteando propuestas para diseñar la solución del problema.
5. Implementar la propuesta llevando a cabo todas las actividades establecidas en el alcance del plan de acción y logrando que el proyecto cumpla con los objetivos esperados y genere valor para los interesados.
6. Analizar los resultados obtenidos en la implementación, identificando puntos de posibles mejoras.

1.5. METODOLOGÍA

Para el presente trabajo de titulación se utilizó una metodología de tipo inductiva, ya que se han obtenido conclusiones a partir de la experiencia, en acciones concretas con el fin de alcanzar la resolución o conclusión general del problema.

El tipo de metodología usado es evaluativa y aplicativa. Evaluativa porque analiza la gestión de procesos de negocios (BPM del inglés Business Process Management) y las metodologías ágiles como

SCRUM. Aplicativa porque utiliza tecnología Business Process Management System, (BPMS) para posteriormente proponer una solución adaptable a las necesidades de la organización, desarrollando un modelo para la gestión de los procesos en el área de desarrollo de Software que ayudará en el control y asignación de recursos, para de esta manera optimizar su uso en la ejecución de los procesos.

CAPÍTULO 2

MARCO TEÓRICO

2.1. BUSINESS PROCESS MANAGEMENT (BPM)

2.1.1. FUNDAMENTACIÓN TEÓRICA

Dentro de una organización se podría definir a un proceso como un conjunto de tareas o actividades que siguen una secuencia lógica, que van a ejecutarse en un tiempo concreto, fundamentalmente para conseguir un resultado empresarial. De esta forma podremos estar realizando unas tareas estructuradas, es decir un proceso para conseguir los objetivos empresariales que se desea. Se debe realizar este proceso con eficacia y eficiencia [2].

Para analizar un proceso de negocio es indispensable estudiar todas y cada una de las actividades que forman parte de él. De esta manera, si fuese necesario, se podrían introducir las mejoras requeridas.

Una actividad es una tarea que se realiza con recursos asignados y disponibles siguiendo reglas previamente definidas. Una actividad recibe una entrada que viene de una acción realizada anteriormente y debe generar una salida, es decir transmite los resultados a la acción que prosigue [3]. En la Figura 2.1 se muestra el esquema de una actividad realizada en un proceso de negocio.



Figura 2.1: Actividad de un proceso de Negocio

Beneficios de documentar un proceso de negocio

Entre los beneficios que una organización puede lograr a partir de la documentación de sus procesos de negocios están:

- Permite tener una visión clara de la organización, además de las relaciones internas.
- Al definir las actividades realizadas por cada participante del proceso, se pueden establecer responsabilidades, así como también eliminar confusiones y desconocimientos.
- Acelera el aprendizaje de los nuevos miembros que se integren a la organización, de esta manera se consigue también abaratar costos (Inversión Financiera).
- Estandarizar las actividades que se realizan para la consecución de los objetivos, es decir, estas actividades deben ser desarrolladas de una misma forma y de manera consistente.
- Garantizar el cumplimiento de las actividades.

2.1.2. CICLO DE VIDA DEL PROCESO DE NEGOCIO

La gestión de procesos de negocios ofrece soporte en el ciclo de la vida del proceso, desde su identificación hasta su ejecución y posterior evaluación de las opciones de mejora.

El ciclo de vida del proceso de negocio está conformado por cuatro fases: Análisis y Diseño, Configuración, Ejecución y Evaluación.

Durante estas fases se puede incorporar actividades que permiten realizar mediciones y mejora continua.

En la fase de Análisis y Diseño se debe identificar y modelar explícitamente los procesos de negocio. Para esto, una de las opciones más recomendadas es utilizar una notación adecuada como lo es Business Process Model and Notation (BPMN 2.0). Esta notación a más de ofrecer una amplia gama de elementos para modelar procesos, define también un formato de intercambio de modelo que funciona con herramientas de distintos vendedores.

Es fundamental que cuando se realice un modelado este sea verificado y validado con el fin de saber si cumple con las características de calidad deseables. Para esto se usa por ejemplo herramientas de simulación.

Una vez que se cumpla con lo anteriormente expuesto, es decir el modelo del proceso ya fue validado, se procede a la fase de configuración, que es donde son implementados los sistemas de software que brindarán soporte al proceso de negocio. Se trabaja de manera similar al desarrollo de software tradicional ya que el desarrollo de estos sistemas debe pasar por pruebas de calidad [4].

En la fase de Ejecución los procesos de negocio son implementados en el día a día. Es decir, los usuarios tanto internos como externos son asignados con roles previamente definidos. El sistema BPM se encarga de registrar los datos de la ejecución real del proceso, con los tiempos que toma realizar cada actividad y las personas involucradas en estas; de esta manera se calculan las medidas de ejecución que son de interés.

Finalmente en la fase de Evaluación, utilizando técnicas de minería de procesos, todos los datos registrados en la etapa anterior se analizan para identificar nuevas oportunidades de mejora. De esta manera el ciclo de vida del proceso de negocio implica introducir poco a poco las mejoras. En la Figura 2.2

podemos observar la descripción de las fases del ciclo de vida del proceso.



Figura 2.2: Descripción de las fases de Ciclo de Vida del Proceso adaptado de [4]

BPM no es solamente una tecnología que nos permite automatizar u orquestrar los procesos; es también un conjunto de disciplinas, herramientas, métodos, técnicas que nos van a permitir identificar los procesos de negocio de la empresa, modelizar estos procesos, diseñar los procesos para una posible automatización, llevar estos procesos informativos donde tengamos la posibilidad de orquestrar todas las actividades pertenecientes a estos procesos, para luego realizar una

integración completa con los sistemas actuales de la empresa, con los datos y también con todas las aplicaciones.

2.1.3. MODELADO DE PROCESOS DE NEGOCIO EN BPMN

BPMN facilita un lenguaje común para que los interesados puedan comunicar los procesos de forma clara, completa y eficiente, pudiendo ser comprendidos. Es así como BPMN define la notación y semántica de un Diagrama de Procesos de Negocio (Business Process Diagram, BPD). BPD es un diagrama diseñado para representar gráficamente la secuencia que siguen todas las actividades que ocurren mientras se ejecuta un proceso, incluye toda la información que se considera necesaria para el análisis [5].

El modelado de procesos en BPMN permite describir un proceso de negocio en distintos niveles:

1. *Mapas de Procesos*: Las actividades son presentadas en diagramas de flujo, sin mucho detalle, solo con el nombre de la actividad y si tiene condiciones de decisiones a nivel general.

2. *Descripción de Procesos*: Aquí las descripciones del proceso, recursos, roles e información son más detalladas.
3. *Modelo de Proceso*: Los diagramas de flujo que se realizan en este nivel son más detallados, con la información necesaria para poder ser analizada y simulada. Con este modelado más detallado se puede ejecutar directamente el modelo o importarlo a herramientas que lo hagan.

2.1.4. ELEMENTOS BÁSICOS UTILIZADOS EN BPMN 2.0

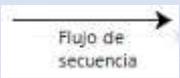
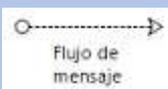
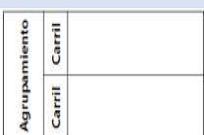
BPMN (Business Process Model and Notation) proporciona un conjunto de elementos para la representación de los procesos de negocio a través de modelos. Estos elementos permiten comprender un modelo de procesos de negocio con facilidad [6], y se clasifican en los siguientes cuatro grupos:

- **Objetos de Flujo**: Determinan el comportamiento del proceso de negocio.
- **Objetos Conectores**: Conectan los objetos de flujo y otros elementos.
- **Canales o Carriles (Swimlanes)**: Permiten crear agrupamientos o estructuras.

- Artefactos: Proporcionan mayor detalle sobre el proceso de negocio.

A continuación se muestra la Tabla 1 que contiene la descripción y la representación gráfica de los elementos de BPMN [7]:

Tabla 1: Elementos de BPMN [7]

Categoría	Elemento	Descripción	Gráfica
Objetos de Flujo	Evento	Es algo que sucede durante el proceso, afecta al flujo del proceso. Normalmente tienen una causa (disparador) o un impacto (resultado). Los eventos pueden ser de Inicio, intermedios o finales.	 Inicial Intermedio Final
	Actividad	Trabajo que se realiza durante el proceso. Pueden ser: Atómicas (Tareas). No atómicas (Subprocesos).	 Tarea Subproceso
	Compuertas	Controla convergencias y divergencias del flujo del proceso. Representa una decisión para unir caminos.	 Exclusivo Basado en evento Inclusivo Paralelo Complejo
Objetos Conectores	Flujo de Secuencia	Se utiliza para mostrar el orden en que se realizan las actividades durante un proceso.	 Flujo de secuencia
	Flujo de Mensaje	Muestra el intercambio de mensaje entre dos participantes.	 Flujo de mensaje
	Asociación	Enlaza información y otros artefactos BPMN.	 Asociación
Canales o Carriles (Swimlanes)	Piscina (Pool)	Representa a un actor o rol en un proceso. Actúa como un contenedor gráfico para particionar un conjunto de actividades. El flujo de secuencia no puede atravesar dos pools diferentes.	 Nombre Piscina Piscina
	Carril (Lane)	Permite subclasificar las actividades de un participante en función al rol que desempeña. El flujo de secuencia puede atravesar dos lanes diferentes.	 Agrupamiento Carril

Categoría	Elemento	Descripción	Gráfica
Artefactos	Objeto de Datos	Proporciona descripción de lo que realiza el proceso. Se conectan a las actividades por asociaciones.	
	Grupo	Permite realizar documentación o para propósitos de análisis pero no afecta al flujo de secuencia.	
	Anotación	Provee información adicional.	

2.1.5. BUENAS PRÁCTICAS Y PATRONES DE MODELADO

Los 7PMG (Process Modeling Guidelines) [8] son siete guías de modelado definidas, que proveen una serie de recomendaciones y buenas prácticas que permiten construir un modelo de procesos de negocio desde cero o mejorar los ya existentes. Estas guías permiten realizar un trabajo de modelado obteniendo buenos resultados, en un determinado contexto, siendo compatible en contextos similares [8].

Los beneficios de realizar un trabajo utilizando buenas prácticas, es que el resultado final es exitoso, es reconocido por ser innovador, replicable y medible. Además aporta valor al negocio. A continuación se detalla la propuesta de estas siete guías:

G1: Utilizar tanto como sea posible la menor cantidad de elementos en el modelo debido a que su tamaño puede incidir negativamente en la comprensión del modelo.

G2: Reducir las posibles rutas de cada elemento; cuanto más grande es el número de entradas y salidas que tiene un elemento resulta más difícil de entender.

G3: En lo posible se debe tratar de usar un único elemento de inicio y uno de fin en cada proceso, de lo contrario se incrementa la probabilidad de errores.

G4: Modelar de la forma más estructurada posible, balanceando las compuertas de decisión, usando las compuertas como paréntesis: una para abrir en los posibles caminos y otra de cierre para unirlos nuevamente.

G5: Evitar el uso de compuertas OR, los modelos que contienen solo compuertas AND y XOR en general contienen menos errores.

G6: Usar etiquetas «verbales» para especificar las acciones a realizar en cada tarea.

G7: Descomponer el modelo si tiene más de 50 elementos, utilizando por ejemplo, Sub-procesos de manera que el modelo general quede comprensible.

2.2. TIPOS DE METODOLOGÍAS PARA LA DIRECCIÓN DE PROYECTOS DE DESARROLLO DE SOFTWARE

El desarrollo de software puede llegar a convertirse en una tarea difícil de llevar a cabo si no se proveen de directrices para la aplicación y uso de notaciones, métodos y herramientas durante el proceso.

Para la implementación de metodologías de apoyo para la tarea de desarrollo de software, existen varias propuestas. Por una parte tenemos propuestas de metodologías tradicionales, estas metodologías se basan en realizar el control riguroso del proceso, definiendo de esta manera las actividades involucradas, los artefactos a producir, y las herramientas y notaciones que se utilizarán.

Por otro lado las metodologías ágiles están enfocadas en la efectividad de proyectos con requisitos cambiantes y cuando se requiere reducir drásticamente los tiempos de desarrollo pero conservando una alta calidad. A diferencia de las metodologías tradicionales, en las metodologías ágiles se da mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software.

En algunas organizaciones existen problemas que pueden ser recurrentes por esta razón en la actualidad tenemos varias opciones con metodologías orientadas a resolver dichos problemas, entre los problemas identificados tenemos [1]:

- No hay equidad en la carga de trabajo que se asigna a un recurso.
- Aumento de re-procesos.
- Los presupuestos asignados no son los correctos, falta de control en los mismos lo que ocasiona desfases.
- La calidad de los productos / servicios que se entregan al cliente o usuario disminuye.
- A nivel general no hay un control adecuado.

Dentro de la gestión de proyectos de desarrollo de software tenemos dos opciones definidas: la gestión tradicional y la gestión ágil, ambas opciones son buenas, para escoger la adecuada dependerá del tipo de proyecto y las características de este.

Entre los criterios que se consideran para medir el éxito del desarrollo del software tenemos [9]:

- Debe mantener un bajo coste en el desarrollo inicial.
- Debe ser de fácil entendimiento y mantenimiento.
- Debe ser compatible y portable a nuevo hardware.
- Debe satisfacer las necesidades del cliente, es decir debe hacer lo que el cliente quiere.

Según lo indican varios autores [10] la metodología de desarrollo a aplicar debe cumplir con algunos requisitos que garanticen el éxito en la implementación:

- La visión del producto debe estar clara.
- El cliente debe ser integrado al proceso, es necesario su involucramiento.
- El modelo de ciclo de vida debe estar definido.
- Los requerimientos recibidos deben ser administrados.
- Debe existir una gestión y control de cambios.
- Debe planificarse el desarrollo.
- Debe existir una gestión de la integración del proyecto, es decir incluir procesos para identificar, combinar, unificar y coordinar los diferentes procesos y actividades a realizar.

- El progreso del proyecto debe ser medido constantemente.
- Deben definirse indicadores y métricas para evaluar la calidad del producto.
- Se deben establecer criterios y condiciones para medir el riesgo.

2.2.1. METODOLOGÍAS CLÁSICAS O TRADICIONALES

Las metodologías clásicas o tradicionales se enfocan en la planificación total del trabajo a realizar, una vez que se completa la descripción de los detalles, se empieza con el ciclo del desarrollo del producto de software.

Entre sus características principales tenemos [11]:

- *Permanencia en el Entorno*: El ambiente en el que se desarrollan los proyectos ya tiene definidas sus características, siguen un patrón que son repetidos constantemente.
- *Predecibles*: Esta metodología está basada en el uso de herramientas de planificación y control, siendo así todos los esfuerzos apuntan a cumplir con los tiempos, costos y recursos.

Tipos de modelos aplicado por las Metodologías Clásicas

Entre los tipos de modelos aplicados por las metodologías clásicas tenemos:

MODELO EN CASCADA

Este modelo es recomendado para proyectos estables es decir con pocos cambios, maduros, que tenga los requerimientos claramente definidos [11]. En la Figura 2.3 podemos ver representado el esquema del modelo en cascada.

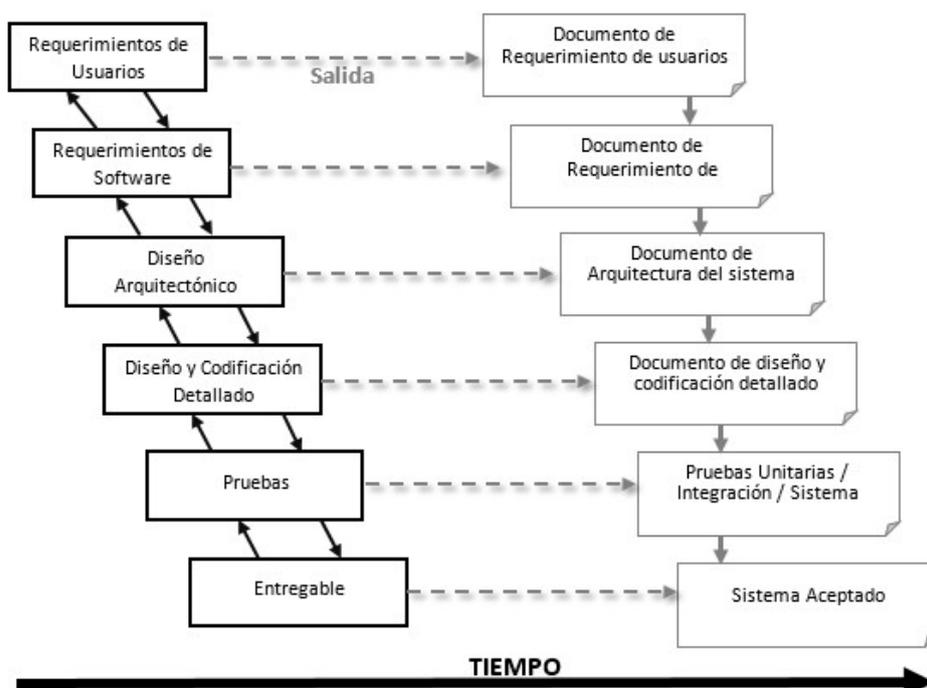


Figura 2.3: Metodología Clásica – Modelo en Cascada [11]

Características – Ventajas del Modelo en Cascada

- La etapa previa debe concluir antes de iniciar la siguiente, maneja de esta manera una secuencia ordenada.
- El trabajo final de una etapa es la entrada para la siguiente.
- Control sobre los tiempos y fechas de entrega.
- Se definen los requerimientos antes de diseñar; diseñar antes de codificar, de esta manera provee estabilidad al proyecto.
- Se debe producir documentos los cuales deben ser aprobados al final de cada etapa.
- Cada etapa concluye con entregables específicos.
- Es el más conocido, comprendido y usado.

Desventajas del Modelo en Cascada

- Poca flexibilidad.
- Es casi imposible regresar a una etapa anterior.
- Al poseer pocos puntos de visibilidad puede parecer lento.
- Se dificulta mostrar avances al cliente, por lo que este debe tener un poco de paciencia.
- Involucra poco al usuario.

- No es apropiado para proyectos en los que se deba tomar decisiones.

MODELO INCREMENTAL

Este modelo consiste en dividir el proyecto en subconjuntos (módulos) con requisitos previamente definidos de manera clara [11], estos deberán ser entregados en cada fase. Cada vez que sale una versión se va incrementando el programa con nuevas funcionalidades que satisfagan más requisitos. La Figura 2.4 presenta un esquema del modelo incremental.

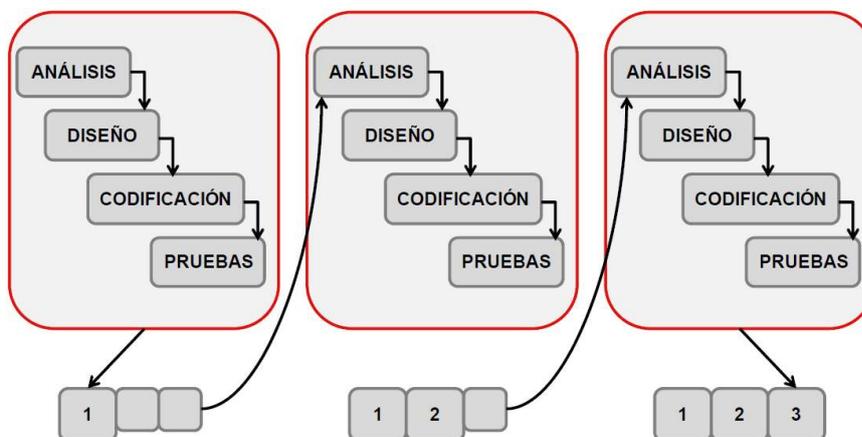


Figura 2.4: Metodología Clásica – Modelo Incremental [11]

Características – Ventajas del Modelo Incremental

- El proyecto se construye en etapas incrementales, siendo así en cada etapa se agrega funcionalidad.
- El cliente clasifica y prioriza las partes del proyecto de acuerdo a su importancia, de esta manera se empieza desarrollando primero las que tengan prioridad alta, por dicha razón estas son las más probadas.
- En cada etapa se realiza un análisis, diseño, codificación y pruebas.
- En comparación al modelo en cascada el cliente puede recibir avances a medida que se avanza en el proyecto.

Desventajas del Modelo Incremental

- Para los usuarios es difícil esperar hasta que su funcionalidad esté lista.
- Puede existir dificultad para identificar puntos comunes a todos los incrementos.
- Puede existir variación en los requerimientos a medida que se van desarrollando los incrementos.

MODELO EVOLUTIVO

Luego de la implementación de una versión inicial, para lograr que el sistema satisfaga las necesidades del usuario, con su ayuda se logra refinar para que en N versiones se obtenga el resultado deseado [11]. Las actividades concurrentes a realizar en cada versión son: especificación, desarrollo y validación. La Figura 2.5 presenta un esquema del Modelo Evolutivo.

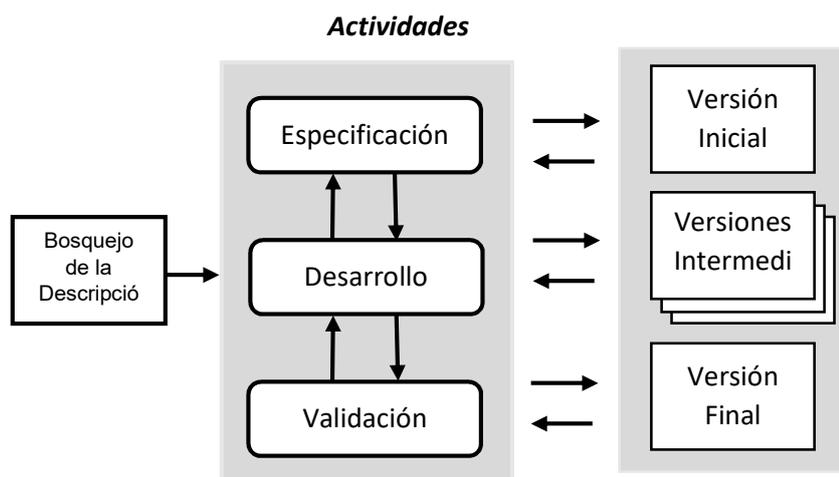


Figura 2.5: Metodología Clásica – Modelo Evolutivo [11]

Se pueden mencionar dos tipos de desarrollo evolutivo:

1. *Desarrollo Exploratorio*: Empieza desarrollando la parte que se tiene más clara, a medida que se van agregando las nuevas funcionalidades presentadas por el usuario el sistema va evolucionando.

2. *Enfoque utilizando Prototipos:* Contrario al exploratorio se empieza desarrollando la parte que el usuario aun no la tiene bien definida, se crean prototipos, al experimentar con estos, permiten definir bien los requisitos.

Características – Ventajas del Modelo Evolutivo

- Los requerimientos se van especificando de forma creciente.
- Involucra el trabajo en equipo con el usuario, de esta manera el requerimiento es más entendible, por lo tanto la calidad del desarrollo mejora.
- Comparado con el modelo en cascada, logra satisfacer, más rápido las necesidades del cliente.

Desventaja del Modelo Evolutivo

Los cambios a los que son sometidos constantemente pueden empobrecer y perjudicar su estructura, de esta manera el mantenimiento posterior puede llegar a ser costoso.

MODELO ESPIRAL

Se representa como una espiral, cada bucle o iteración representa un conjunto de actividades, estas actividades se planifican en función de un objetivo específico, analizando los riesgos y las diferentes alternativas disponibles [11].

Provee al proceso la capacidad de responder rápidamente ante cambios en los requisitos que pueden surgir eventualmente, en cualquiera de las etapas del desarrollo del software; el análisis del riesgo que se entrega cumple un papel importante en la toma de decisiones, permitiendo de esta manera tener un control sobre los costos y la duración del proyecto. La Figura 2.6 representa el esquema del Modelo Espiral.

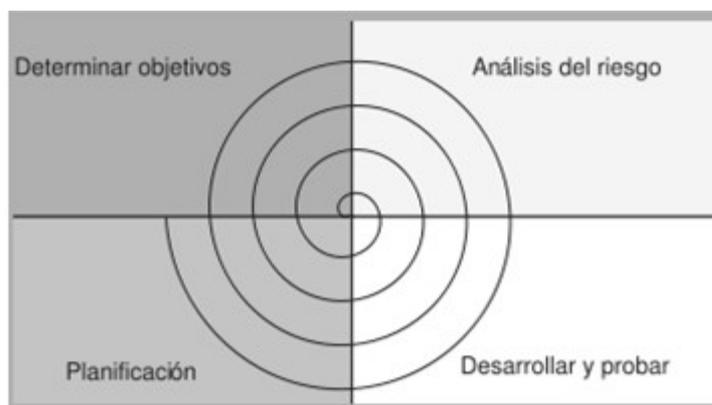


Figura 2.6: Metodología Clásica – Modelo Espiral [11]

Ventajas del Modelo Espiral

- Recomendado cuando los requerimientos del proyecto puedan sufrir cambios o no se conoce bien la aplicación.
- Cuando se pretende probar una arquitectura o tecnología.
- Cuando se requiere rapidez en el desarrollo.

Desventajas del Modelo Espiral

- Se desconoce cuándo se obtendrá un producto aceptable, ni las iteraciones que serán necesarias.
- Crea una falsa expectativa al usuario sobre la velocidad del desarrollo.

DESARROLLO BASADO EN COMPONENTES

Este modelo se basa en la existencia de un número significativo de componentes reusables [11]. El desarrollo del sistema se centra en la integración de estos componentes en un sistema en vez de desarrollarlos desde cero (Componentes y sistemas COTS Commercial-off-the-shelf). La Figura 2.7 muestra un esquema de desarrollo basado en componentes.

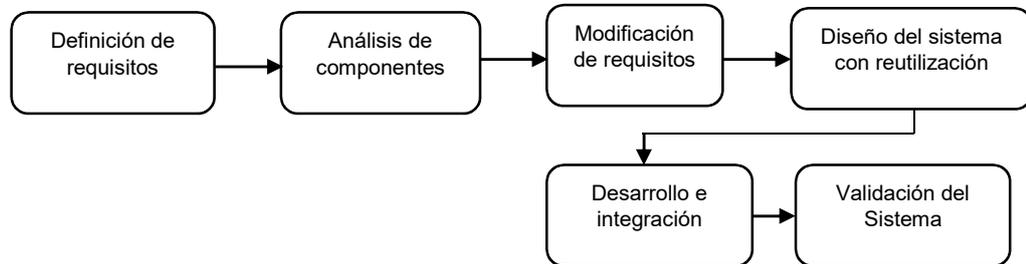


Figura 2.7: Metodología Clásica – Desarrollo basado en componentes [11]

Tipos de componentes

- Servicios Web.
- Colecciones de objetos.
- COTS (Commercial Off The Shelf).

Ventajas

- La cantidad de software a desarrollarse se reduce.
- Se puede disminuir los tiempos de entrega del software.
- Reduce los costos y riesgos.

Desventajas

- Si las nuevas versiones de los componentes reusables no se registran, se pierde el control sobre la avance del software.
- Se corre el riesgo de producir un sistema que no satisfaga las necesidades reales del usuario debido a los ajustes realizados sobre el requerimiento.

2.2.1. METODOLOGÍAS ÁGILES

Este tipo de metodologías generalmente se utiliza en proyectos que ofrezcan servicios, no tanto productos. En este tipo de metodologías se debe tomar como referencia en el punto de partida una definición cerrada de lo que se quiere lograr.

Características y Beneficios

Entre las características de los proyectos gestionados con metodologías ágiles tenemos:

- **Incertidumbre:** Se conoce la necesidad estratégica sin mayor detalle, se da amplia libertad al equipo.
- **Auto-organización:** Cada miembro del grupo cuenta con autonomía, autoaprendizaje, autodisciplina, siendo así que el equipo de trabajo puede tomar las decisiones o transferir conocimientos grupales. Es necesario realizar evaluaciones cada cierto tiempo acerca del producto o servicio que se desarrolla.
- **Desarrollo solapado:** Se desarrollan las actividades a lo largo del ciclo de vida del proyecto, estas se las realiza dependiendo de la necesidad que se va generando.

- Monitoreo sutil: A lo largo del ciclo de vida del proyecto es recomendable incorporar puntos de control, esto con el objetivo de tener un conocimiento de la situación del mismo.
- Evaluación periódica del ambiente laboral.
- Trabajo en equipo: Se recomienda implantar un sistema de reconocimiento en base a los méritos obtenidos por los recursos, así mismo establecer puntos de mejoras en los errores que pueden ocasionarse durante el desarrollo del proyecto.
- Transmisión de conocimientos: Se realizan reuniones donde los miembros del grupo de trabajo puedan intercambiar ideas, exponer y conocer todos los temas relacionados al proyecto.

El Manifiesto Ágil

Según la definición propuesta por Qumer y Henderson – Sellers [12] se puede decir que:

“La agilidad es un comportamiento persistente o habilidad, de entidad sensible, que presenta flexibilidad para adaptarse a cambios, esperados o inesperados, rápidamente; persigue la duración más corta en tiempo; usa instrumentos económicos, simples y de calidad en un ambiente dinámico; y utiliza los

conocimientos y experiencia previos para aprender tanto del entorno interno como del externo.”

El manifiesto ágil [13] numera los principales valores que identifican al desarrollo ágil. Estos valores se presentan en la Figura 2.8.



Figura 2.8: Manifiesto ágil [12]

Los 12 Principios del Manifiesto Ágil

La Tabla 2 presenta el detalle de los doce principios del manifiesto ágil [13] [14]:

Tabla 2: Los doce principios del manifiesto ágil

Los dos primeros principios del manifiesto ágil representan un resumen del espíritu ágil	
1	<p>La satisfacción del cliente es considerada como alta prioridad.</p> <p>Se logra a través de la comunicación constante, realizando entregas de software tempranas que contribuyan valor. El cliente revisará el entregable y dispondrá si lo pone en marcha o necesitará algún ajuste, debiendo informar formalmente de los posibles cambios.</p>
2	<p>Dar la bienvenida a los cambios.</p> <p>Incluso en etapas tardías del desarrollo, los cambios son algo positivo, debido a que al capturar los cambios, se pueden crear ventajas competitivas para el cliente.</p>
Los siguientes principios tienen que ver directamente con el proceso de desarrollo de software a seguir	
3	<p>Entregar software que funcione entre 15 y 60 días.</p> <p>El intervalo entre los tiempos de entrega debe ser lo más corto posible.</p>
4	<p>El cliente debe ser el que guie el proceso de desarrollo.</p> <p>Los desarrolladores deben trabajar junto a la gente del negocio a lo largo de todo el proceso del proyecto.</p>
5	<p>El desarrollo del proyecto debe ser construido alrededor de individuos motivados.</p> <p>En las personas radica el factor principal del éxito, lo demás como los procesos, el entorno, la gestión, etc., pasan a segundo plano.</p>

6	El método más eficiente y efectivo de comunicación dentro de un equipo de desarrollo es el diálogo cara a cara.	Este es el método principal de comunicación, se puede crear documentación no garantiza que el equipo este informado de todo respecto al proyecto.
7	La medida principal para validar el progreso es que el software se encuentre funcionando.	El estado del proyecto se valida a través del código generado, no por la documentación que se tiene de este.
8	El desarrollo sostenible es promovido por los procesos ágiles.	Los promotores, desarrolladores y usuarios deberían ser capaces de mantener un ritmo adecuado a lo largo de la duración proyecto, se asegura que la calidad de lo que se está produciendo es la adecuada.
Los últimos principios están relacionados con el equipo de desarrollo, en cuanto a metas a seguir y organización del mismo		
9	El cuidado continuo a la calidad técnica y al buen diseño mejora la agilidad.	Para que el proyecto avance rápidamente es necesario producir código claro y robusto.
10	La simplicidad es esencial.	Si el código que se produce es simple y de alta calidad será más fácil adaptarlo a los cambios que puedan presentarse en el futuro.
11	De los equipos organizados surgen las mejores arquitecturas, requisitos y diseños.	Se informa al equipo de sus responsabilidades, y este se organiza de acuerdo a los objetivos que quieren lograr.

En intervalos habituales, el equipo debe ser capaz de ajustarse a los nuevos escenarios que pueden presentarse a lo largo del desarrollo. Puede cambiar su organización, sus reglas, sus convenciones, sus relaciones, etc., para seguir siendo ágil. El equipo analiza y evalúa opciones respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

Fases que definen el ciclo del desarrollo Ágil

En la Figura 2.9 se puede visualizar las fases por las que atraviesa el ciclo del desarrollo ágil, el cual es importante entender si se desea trabajar con estas metodologías [13].



Figura 2.9: Ciclo de vida del desarrollo ágil

Concepto: Se asigna el equipo que se encargará del desarrollo, se realiza la definición de las características que tendrá el producto de manera general.

Especulación: Se analiza la información obtenida en la fase anterior, además de establecer los límites bajo los cuales se desarrollará el producto, como por ejemplo cronogramas y costos.

Se desarrolla y revisa el producto partiendo de las ideas principales, se establecen las fechas de entregas de versiones, hitos e iteraciones.

Exploración: Las funcionalidades levantadas en la fase de especulación se agregan en el producto.

Revisión: Se realiza la revisión de lo que se ha construido y se lo valida con el objetivo que se busca.

Cierre: En la fecha establecida inicialmente se entregará una versión del producto deseado. El cierre no indica que el proyecto ha concluido, debido a que se habrá cambios que se deben realizar por mantenimiento, acercándose al producto deseado.

Revisión de Metodologías Ágiles

A pesar de la existencia del manifiesto ágil las diferentes metodologías existentes tienen sus propias características sobre las cuales se fundamentan [10]. La Tabla 3 presenta un resumen de las principales metodologías ágiles con sus características principales.

Tabla 3: Principales metodologías ágiles

Metodología Ágil	Características
Adaptative Software Development (ASD)	<p>Entre sus características principales tenemos [15]:</p> <ul style="list-style-type: none"> • Iterativa. • Orientada a los componentes de software más que a las tareas. • Tolerante a los cambios. • Guiada por los riesgos. • La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo. <p>El ciclo de vida tiene tres componentes que son:</p> <ol style="list-style-type: none"> 1. Especular 2. Colaborar 3. Aprender
Lean Development (LD)	<p>Metodología para el modelado y la generación de documentación que se encuentra alineado con los principios del desarrollo ágil y que puede ser utilizado como sustituto de UML, es una recopilación de las mejores prácticas [16], basada en siete principios:</p> <ol style="list-style-type: none"> 1. Eliminar el desperdicio 2. Crear conocimiento 3. Embeber a la calidad 4. Postergar el compromiso 5. Optimizar rol 6. Entregar rápido 7. Respetar a las personas
Extreme Programming, XP	<p>Se basa en las relaciones interpersonales como clave para el éxito en el proceso de desarrollo de software [10]. Características principales:</p> <ul style="list-style-type: none"> • Potenciar las relaciones interpersonales. • Promueve el trabajo en equipo. • Realimentación continua entre el cliente y el equipo de desarrollo. • Metodología basada en prueba y error. • Fundamentada en valores y prácticas.
Dynamic Systems development methods (DSDM)	<p>Define el marco para desarrollar un proceso de producción de software [17], entre sus características tenemos:</p> <ul style="list-style-type: none"> • Iterativo e incremental. • El equipo de desarrollo y el usuario trabajan juntos.

	<p>Propone cinco fases:</p> <ol style="list-style-type: none"> 1. Estudio de viabilidad 2. Estudio del negocio 3. Modelado funcional 4. Diseño y construcción 5. Implementación
<p>Feature-Driven Development (FDD)</p>	<p>Entre las características principales están [18]:</p> <ul style="list-style-type: none"> • Monitoreo Constante. • Propone tener etapas de cierre cada dos semanas. • Resultados periódicos y tangibles. <p>Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software.</p>

2.2.3. COMPARACIÓN ENTRE METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES

La Tabla 4 presenta la comparación entre las principales características de las metodologías tradicionales o clásicas vs las metodologías ágiles [10].

Tabla 4: Comparación entre metodologías tradicionales vs. metodologías ágiles

Metodologías tradicionales	Metodologías ágiles
Basadas en normas que provienen de estándares seguidos por el entorno de desarrollo.	Se basan en crear estrategias, métodos, criterios, provenientes de prácticas, experiencias de producción de código.
Puede existir resistencia a los cambios.	Dispuestos para cambios que se puedan presentar durante el proyecto.
Son impuestas externamente.	Son impuestas internamente.
El proceso es muy controlado, tienen numerosas normas.	El proceso es menos controlado, con pocas normas.
Contrato prefijado.	Contrato flexible y en ciertos casos puede ni existir.
Mediante reuniones el cliente puede interactuar con el equipo de desarrollo.	El cliente forma parte del desarrollo.
Grupos grandes.	Grupos pequeños (<10).
Es fundamental la arquitectura de software.	Menor importancia a la arquitectura de software.

Para el presente trabajo de titulación se trabajó con metodologías ágiles, la escogida fue SCRUM, la cual describiremos a continuación.

2.3. METODOLOGÍA SCRUM

La metodología SCRUM es un marco fundamentado en los principios ágiles que apoya el desarrollo y la gestión de productos con cierto grado de complejidad, aplica el desarrollo evolutivo de las aplicaciones de software, a través de iteraciones [19]. Su uso es especialmente conveniente en proyectos con un rápido cambio de requisitos, tal como lo son los productos de software. Es ideal para aplicarlo en organizaciones en las que su entorno presenta las características descritas en la sección 2.2.2.1.

Un proyecto que utiliza metodología Scrum se ejecuta en ciclos cortos denominados Sprint, es recomendable que la duración de cada sprint sea constante y definida por el equipo basado en su propia experiencia. Al final de cada sprint, el equipo presentará los avances obtenidos, el resultado será un producto que potencialmente se puede entregar al cliente. En la Figura 2.10 se puede observar la metodología SCRUM representada esquemáticamente.

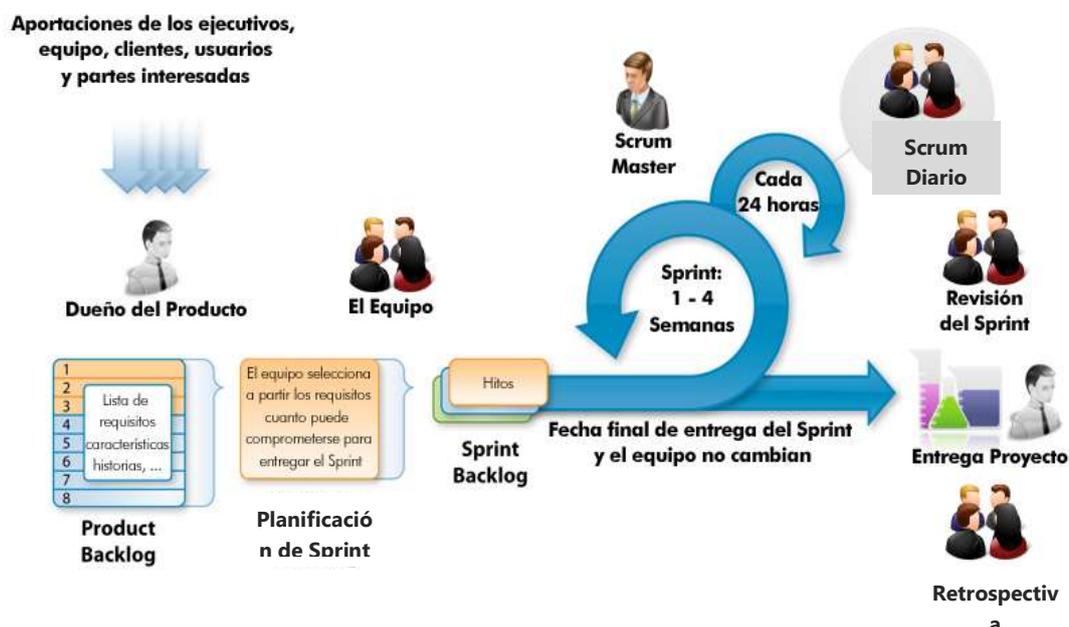


Figura 2.10: Descripción conceptual de un ciclo SCRUM [11]

2.3.1. COMPONENTES DE SCRUM

Entre los componentes del Scrum tenemos los roles, artefactos y reuniones, los cuales se detallaran a continuación:

ROLES

Los roles permiten establecer y describir las responsabilidades, los niveles de autoridad de individuos o grupo de individuos que participan a lo largo del proceso.

Los roles definidos en la metodología Scrum son:

- **Dueño del Producto (Product Owner)**

Es la persona encargada de las tomas de decisiones que se presentarán a lo largo del proyecto. Conoce al cliente y analiza las ideas presentadas por él y las ordena según la prioridad. Tiene una visión clara de lo que se espera como producto.

- **Scrum Master**

Su trabajo es asegurar que los principios, métodos y herramientas a utilizar, son correctamente aplicados durante el desarrollo. Se asegura que el equipo tenga el conocimiento y habilidades necesarias que se demandan en el desarrollo de las aplicaciones, y que se cuente con la cantidad de personas necesarias para llevar a cabo el trabajo.

Elimina todos los inconvenientes que puedan presentarse durante el desarrollo del proyecto, debe interactuar con el cliente y los gestores. El Scrum Master junto con el Dueño del Producto

adquieren el compromiso y la responsabilidad de intermediar y negociar sobre algún inconveniente que pueda presentarse en los tiempos de trabajo o en caso de ser necesario agregar o cambiar características al requerimiento de software.

- **Equipo de Desarrollo Scrum**

Es el equipo de trabajo responsable del desarrollo del producto a entregar, suele ser un grupo pequeño de entre 3-9 personas. Desde la definición de los requerimientos hasta la entrega del producto final trabajan en conjunto con los distintos stakeholders. Tienen potestad para organizarse libremente y tomar decisiones que ayuden en la consecución del objetivo.

Los roles descritos anteriormente pertenecen al grupo de las personas comprometidas con el proyecto y con el proceso Scrum. Pero existe otro grupo de individuos que si bien no forman parte del proceso Scrum, se benefician con él y son necesarios para obtener una buena retroalimentación, para revisar y planear cada sprint.

A continuación la descripción de cada uno de los roles involucrados.

- Usuarios: Personas a las cuales va destinado el producto final.
- Stakeholders: Personas a las que el desarrollo del proyecto les producirá beneficios, participan en los sprints.
- Gerentes: Participan en la selección de los objetivos y requisitos del proyecto, se encarga de la toma de decisiones finales.

ARTEFACTOS

- **Product Backlog**

Contiene una lista con todas las funcionalidades, mejoras, y corrección de errores que el equipo de desarrollo Scrum deberá realizar; es decir, todo lo que suponga una tarea a realizar. El product backlog se ordena por valor, riesgo, prioridad y necesidad. Estas especificaciones se irán incorporando al producto en cada iteración. Esta lista será creada y tratada por el cliente, partiendo del resultado de una lluvia de ideas que se

da en las reuniones, en colaboración de todo el equipo. Es necesaria tener una visión clara del producto para que posteriormente el Scrum Master, se encargue de indicar el costo estimado de completar el requerimiento [20]. En la Figura 2.11 se presenta un modelo del Product Backlog.

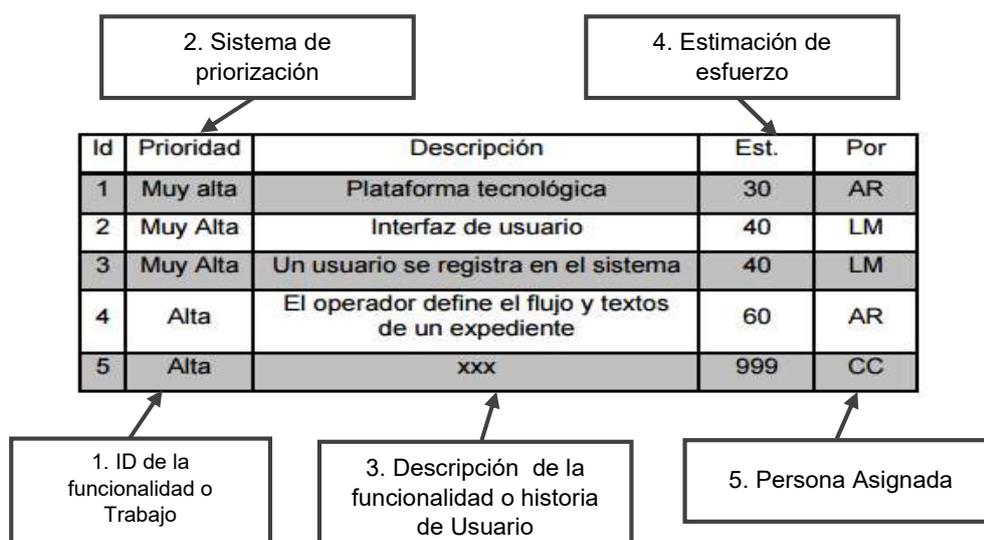


Figura 2.11: Ejemplo con formato de un product backlog adaptado de [13]

Las Historias de Usuarios (User Stories) en el product backlog: Son las descripciones de los requisitos y funcionalidades que va a tener la aplicación. Serán realizadas con la cooperación del cliente y el equipo. Se componen de las siguientes partes [13]:

- Tarjeta: Recordatorio con una breve descripción.

- Conversación: permitirá cerciorarse que lo que se entendió está correcto, y concretar el objetivo.
- Confirmación: Se fijan detalles relevantes y límites.

En la Figura 2.12 se presenta un ejemplo del formato de Historia de Usuario.

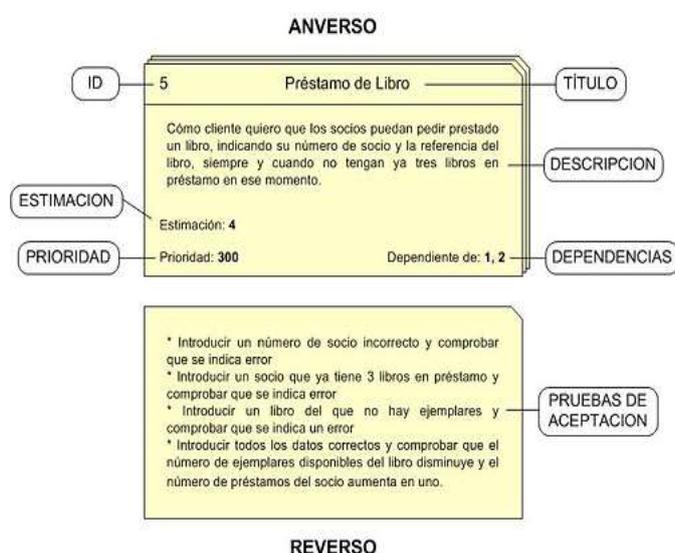


Figura 2.12: Formato de historia de usuario [13]

- **Sprint Backlog:**

Es una lista de actividades o tareas que deben realizarse durante un Sprint, partiendo de lo obtenido en el Product Backlog. Se presentan las tareas a realizar por cada uno de los objetivos/requisitos, el esfuerzo o tiempo pendiente para terminarlas y la auto-asignación que han hecho los miembros

del equipo. Al descomponer las tareas en unidades más pequeñas resulta fácil la detección de problemas que puedan presentarse y por lo tanto la toma de decisiones será rápida [21]. La Figura 2.13 presenta un modelo con el formato de un sprint Backlog.

Requisito	Tarea	Quien	Estado (No iniciada / en progreso / completada)	Dia:												
				1	2	3	4	5	6	7	8	9	10			
				Horas	1120	1088	1076	1048	1040	1032	1020	1008	992	972		
				pendientes												
Requisito A	Tarea 1	Joao	Completada	16	8											
Requisito A	Tarea 4	Laura	Completada	4												
Requisito A	Tarea 5	Laura	Completada	4												
Requisito A	Tarea 3	Gabri	Completada	8												
Requisito A	Tarea 2	Laura	Completada	16	8	4										
Requisito A	Tarea 6	Gabri	Completada	8	8	8										
Requisito A	Tarea 7	Joao	Completada	16	16	16	8									
Requisito A	Tarea 8	Laura	Completada	8	8	8										
Requisito A	Tarea 9	Laura	Completada	8	8	8	8	8								
Requisito A	Tarea 10	Laura	Completada	8	8	8	8	8	8	4						
Requisito A	Tarea 11	Joao	Completada	16	16	16	16	16	16	8						
Requisito B	Tarea 12	Gabri	Completada	16	16	16	16	16	16	16	16	16	8			
Requisito B	Tarea 13	Laura	Completada	16	16	16	16	16	16	16	16	16	8			
Requisito B	Tarea 14	Joao	En progreso	8	8	8	8	8	8	8	8	8	8	4		
Requisito B	Tarea 15	Gabri	En progreso	8	8	8	8	8	8	8	8	8	8	8		
Requisito B	Tarea 16	Laura	En progreso	8	8	8	8	8	8	8	8	8	8	8		
Requisito C	Tarea 17	Joao	No iniciada	4	4	4	4	4	4	4	4	4	4	4		
Requisito C	Tarea 18	Gabri	No iniciada	8	8	8	8	8	8	8	8	8	8	8		
Requisito C	Tarea 19	Laura	No iniciada	16	16	16	16	16	16	16	16	16	16	16		
Requisito C	Tarea 20	Joao	No iniciada	8	8	8	8	8	8	8	8	8	8	8		

Figura 2.13: Ejemplo con formato de un Sprint Backlog [13]

Para gestionar la lista de objetivos/requisitos que se debe cumplir en cada iteración se puede utilizar un tablero de tareas (Scrum Taskboard). Este contiene los estados: pendientes de iniciar, en progreso, hechas. Se coloca un listado con cada objetivo y a continuación las tareas a realizar para completarlo, estas se van ubicando hacia la derecha en el estado que les corresponde. La Figura 2.14 muestra un modelo de un tablero de tareas.

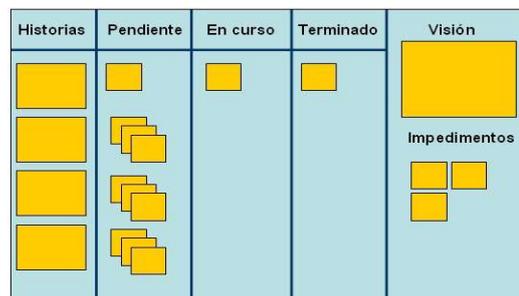


Figura 2.14: Tablero de Tareas [13]

- **Product Increment**

Es el resultado entregable de un Sprint, corresponde a la implementación de las tareas del Product Backlog especificadas en el Sprint Backlog.

El equipo de desarrollo Scrum entrega la versión finalizada y completamente operativa.

- **Burn-down chart**

Gráfica que muestra la cantidad de requisitos pendientes al inicio de cada sprint. La Figura 2.15 presenta un ejemplo de Burn-Down Chart.

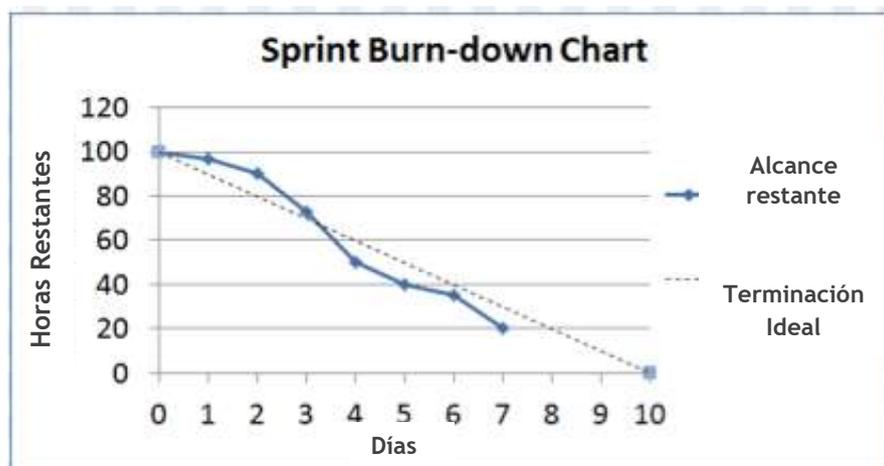


Figura 2.15: Burn-Down Chart adaptado de [13]

REUNIONES

Las reuniones se las realiza con el objetivo de sincronizar las actividades a realizar durante un proceso Scrum. Ocurren en un determinado intervalo de tiempo, con un límite de duración fijada mediante una técnica denominada TimeBox [15]. A continuación se detallan los tipos de reuniones de Scrum:

- **Planificación del Sprint**

La reunión de planificación del Sprint se la debe realizar al inicio de cada ciclo de Sprint, con lo cual se pretende identificar y definir el trabajo a efectuar en cada iteración. Se establece en un

documento los objetivos y los requisitos que tendrá el producto ordenado por prioridades.

Los asistentes para estas reuniones son: el Dueño del Producto, el Scrum Master y el equipo de desarrollo.

Durante la reunión el Dueño del Producto presenta detalladamente las historias de usuarios, el Scrum Master define la lista de tareas en el Sprint Backlog, finalmente el equipo de desarrollo expone sus dudas e inquietudes, desglosa cada historia de usuario que se escogió para el sprint en tareas y estima el tiempo de resolución de cada una de estas.

- **Scrum Diario (Daily Scrum Meeting)**

Diariamente se efectúan reuniones cuya duración máxima es de 15 minutos, los asistentes son todos los miembros del equipo de desarrollo Scrum (Scrum Team). El objetivo de estas reuniones es la evaluación del avance de las actividades, para esto hacemos las siguientes preguntas:

- ¿Qué trabajo se realizó desde la reunión anterior?
- ¿Qué trabajo se realizará hasta una nueva reunión?
- ¿Qué inconvenientes han surgido y cuál es la solución que permite continuar?

- **Demo - Revisión del Sprint (Sprint Review Meeting)**

Al concluir cada sprint se efectuará una reunión en donde los asistentes serán todo el grupo de interesados (clientes, usuarios, equipo Scrum, etc.) y cuya duración máxima será de una hora, el objetivo es realizar una revisión del incremento que se generó. Para esta reunión se presentarán los resultados finales obtenidos en un demo o prototipo; de esta exposición se obtendrán algunas ideas, rectificaciones y mejoras que se deben realizar en el producto.

Esta es una reunión de aprendizaje y evolución. Adicionalmente se realiza la convocatoria para la siguiente reunión de planificación.

- **Retrospectiva**

La metodología Scrum plantea una última reunión al finalizar cada sprint: la de retrospectiva. Esta reunión se debe realizar inmediatamente después de la presentación del demo. Los asistentes a esta reunión deben ser el equipo de desarrollo, el Dueño del Producto y el Scrum Master. El objetivo de esta reunión es que todos los miembros del equipo expongan sus opiniones sobre el sprint que recién fue terminado, se analiza el proceso, se identifican los problemas presentados a lo largo del sprint y se proponen ideas de mejoras, de esta manera se busca una evolución del proceso. Esta reunión tiene un tiempo de duración de entre 2 a 4 horas.

2.3.2. BENEFICIOS DE UTILIZAR LA METODOLOGÍA SCRUM

Entre los beneficios de utilizar la metodología ágil Scrum se pueden mencionar las siguientes [13]:

- Las expectativas del cliente en su mayoría logran ser satisfechas.
- Existe mayor flexibilidad al cambio en los requerimientos generados por la necesidad de los clientes o por evoluciones del mercado.

- El software desarrollado es de mejor calidad.
- Permite controlar los riesgos eficazmente y de manera anticipada.
- Las estimaciones de tiempos de desarrollo de las actividades dispuestas en el backlog son más acertados.
- Se consigue una mayor productividad del equipo debido a que son autónomos a la hora de organizarse, eliminando de esta manera la burocracia.
- El cliente puede comenzar a usar algunas de las funcionalidades antes de que el proyecto esté finalizado por completo.

CAPÍTULO 3

LEVANTAMIENTO DE INFORMACIÓN

3.1. ANÁLISIS DE LA SITUACIÓN ACTUAL DE LA EMPRESA

3.1.1. FINES Y FUNCIONES

Esta empresa objeto del presente trabajo de titulación, pertenece al sector privado, y brinda soluciones tecnológicas de apoyo a las áreas administrativas y financieras. Sus soluciones ofrecen a sus clientes la posibilidad de mejorar sus operaciones sin grandes inversiones, buscando fomentar la innovación, colaborando con las empresas en la introducción de cambios para mejora de sus procesos de producción, administración y financieros; dando como resultado beneficios comerciales.

Las características, prestaciones y buen funcionamiento de los productos y servicios ofrecidos por la empresa garantizan el control, seguridad y correcto uso de la información generada por las transacciones realizadas en los negocios del cliente, convirtiéndose así en un activo muy valioso para él.

3.1.2. VISIÓN

La visión que tiene la empresa es la siguiente:

“Establecernos como una reconocida empresa de tecnología de Información, que brinda a sus clientes la posibilidad de mejorar sus operaciones, mediante la oferta de excelentes productos y servicios, identificada principalmente por fomentar y crear innovación en todo su entorno, con una sólida cultura, basados siempre en la calidad y el principio de bienestar general para todos sus integrantes.”

3.1.3. MISIÓN

La misión que tiene la empresa es la siguiente:

“Nuestro compromiso es ayudar al proceso de producción y crecimiento de las empresas a través de nuestro talento con alto

grado de profesionalismo y conocimientos en el área de tecnología de la Información, logrando ser aliado en innovación tecnológica de nuestros clientes, proporcionándoles soluciones y aplicaciones acorde a sus necesidades, adaptados a los cambios tecnológicos.”

3.1.4. ORGANIGRAMA DE LA EMPRESA

Actualmente la organización cuenta con tres áreas definidas, las cuales se detallan a continuación:

- **Gerencia de Operaciones:** Es un pilar fundamental para la empresa ya que aquí es donde se controlan los procesos de elaboración de los productos y servicios destinados a satisfacer las necesidades del cliente. Tiene como responsabilidad velar que los entregables cumplan con las normas de calidad requeridas, utilizando sus recursos de manera eficiente. Esta área cuenta con las secciones de Soporte y Desarrollo (Ver Figura 3.1).
- **Gerencia Comercial:** Se encarga de elaborar planes de acción para dar a conocer al público los bienes y servicios que ofrece la empresa. Mediante el estudio de mercado. Realiza

análisis de tendencias para luego realizar publicidad y presentar promociones (Ver Figura 3.1).

- **Gerencia Administrativa:** Encargada de asesorar a la gerencia general de la empresa en la formulación de políticas, planes y programas en la administración del talento humano, así como en el uso de los recursos físicos y financieros. El área administrativa comprende los departamentos de: Recursos Humanos, Cobranzas y Contabilidad (Ver Figura 3.1).

Cada una de estas áreas cuenta con sus respectivas divisiones, y con personal especializado que reportan directamente a la Gerencia General, esta a su vez es controlada por el Directorio y los Accionistas. La Figura 3.1 muestra el detalle del organigrama de la organización.

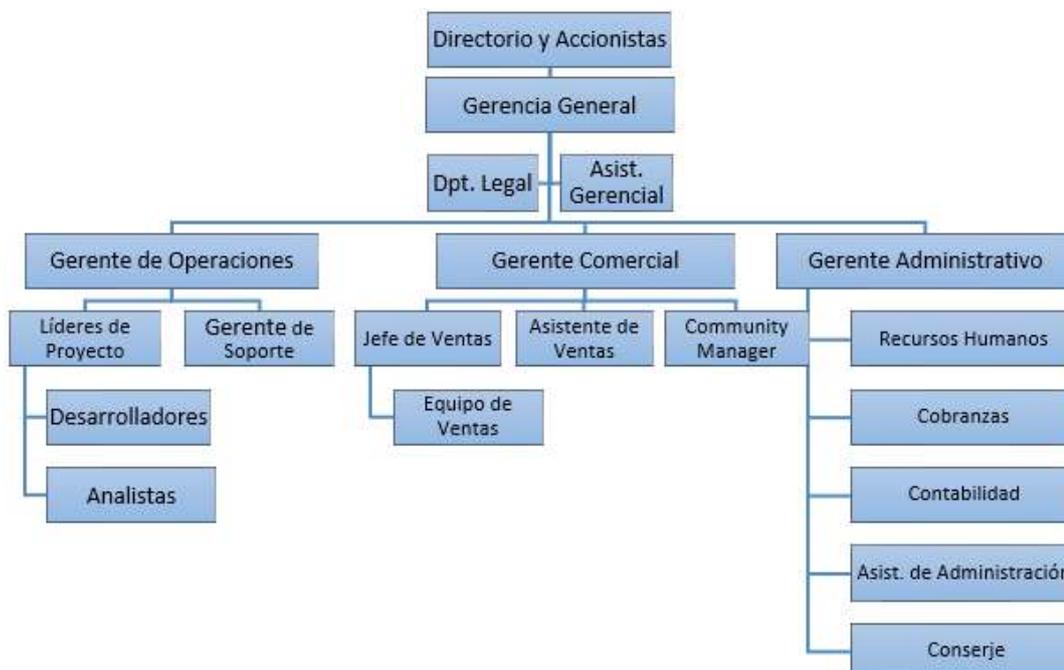


Figura 3.1: Organigrama de la empresa

3.2. ANÁLISIS DE LA SITUACIÓN DEL ÁREA OPERATIVA

Previo al análisis del proceso de desarrollo y mantenimiento de software objeto de estudio, es necesario entender el entorno que lo afecta directamente. Para esto es preciso revisar cómo se maneja el área operativa.

El área operativa actualmente cuenta con los sub-departamentos de Soporte de Usuario y de Desarrollo y Mantenimiento de Software. Al trabajar conjuntamente permiten acelerar la entrega y mejorar el

desempeño de las aplicaciones, estimular la innovación y por sobre todo lograr la satisfacción de los clientes/usuarios.

3.1.1. ANÁLISIS FODA

El análisis FODA permite tener un enfoque claro de la situación actual del área de operaciones el cual es objeto de estudio. En esta evaluación se muestran los pros y contras del área operativa, partiendo de eso se pueden obtener pautas válidas para el avance del estudio. Cabe destacar que existe un claro desbalance en la parte organizacional dentro del área de operaciones que es la encargada de llevar a cabo el desarrollo de los requerimientos solicitados por los clientes/usuarios de la empresa, esta área se encarga de proporcionar los recursos destinados para la atención directa con el usuario. En la Tabla 5 se puede apreciar los aspectos identificados como parte de la situación actual del área en cuanto a manejo de las operaciones:

Tabla 5: Análisis FODA del Área Operativa

	Fortalezas	Debilidades
Aspectos Internos	<ul style="list-style-type: none"> • Recursos comprometidos. • Predisposición del personal para aprender. • Apoyo de otras áreas. 	<ul style="list-style-type: none"> • Requerimientos no definidos. • Falta de enfoque en la satisfacción del cliente. • Falta de experiencia en algunos colaboradores. • Falta de procedimientos, políticas y metodologías. • Falta de documentación. • Falta de calidad del producto / servicio que ofrece la empresa. • Falta de organización. • Falta de control sobre la manipulación de los fuentes (versionamiento). • Falta de control de cambios.
	Oportunidades	Amenazas
Aspectos Externos	<ul style="list-style-type: none"> • Crear vínculos de relación con el cliente a corto plazo, es decir involucrarlo en el proceso. • Capacidad de colocar nuevos servicios donde el cliente. • Obtener referencias por parte del cliente. 	<ul style="list-style-type: none"> • Competencia.

3.1.2. CLIENTES Y USUARIOS DE LA EMPRESA

Los clientes/usuarios tienen un rol primordial dentro de los procesos que se manejan en el área operativa, ya que ellos son los que solicitan los requerimientos y se benefician con el

resultado final, de ahí que el objetivo principal de la organización y del área operativa es lograr su satisfacción.

En la Tabla 6 se puede observar el número de clientes que mantiene la empresa, sumando un total de 246, clasificados de acuerdo a su tamaño (volumen ventas anual (V) y el número trabajadores (T)), esta categorización está dada por el Instituto Ecuatoriano de Estadísticas y Censos INEC [22].

Tabla 6: Número de clientes que mantiene la empresa clasificadas según su tamaño

<i>Empresa</i>	Tamaño	Nº de empresas clientes	Porcentaje de representación
<i>Grande</i>	V: \$5'000.001 en adelante. T: 200 en adelante.	7	2.85%
<i>Mediana B</i>	V: \$2'000.001 a \$5'000.000. T: 100 a 199.	12	4.88%
<i>Mediana A</i>	V: \$1'000.001 a \$2'000.000. T: 50 a 99.	112	45.53%
<i>Pequeña</i>	V: \$ \$100.001 a \$1'000.000. T: 10 a 49.	95	38.62%
<i>Microempresa</i>	V: < a \$100.000. T: 1 a 9.	20	8.12%
	TOTAL	246	100%

Como se puede observar en la Tabla 6, podemos determinar que la mayoría de sus clientes (84,15%) pertenecen a empresas de las categorías pequeña y mediana (A); se conoce que la mayor parte de los ingresos de la organización provienen de estos sectores.

3.1.3. ESTADÍSTICA DE LOS REQUERIMIENTOS INGRESADOS AL AREA OPERATIVA

Un RUT (Registro Único de Trabajo) es un documento por medio del cual el cliente/usuario ingresa su requerimiento. Partiendo del ingreso del RUT se registran OT (órdenes de trabajo), las cuales se van aperturando según la necesidad y son asignadas al personal de soporte de I nivel, analista funcional, desarrollador, siendo así un RUT puede tener relacionadas varias OT.

Estadística de requerimientos recibidos y atendidos por mes

La Figura 3.2 representa el número de RUTs que ingresaron al área operativa durante el período de junio de 2016 a marzo de 2017; adicionalmente se presenta el número de requerimientos

que se alcanzaron a atender, recibieron la aprobación del usuario/ cliente y se dieron por cerrados.

Según el gráfico durante este período se recibió un total de 1905 requerimientos, de los cuales se logró cerrar un total de 1492 quedando pendientes 413 requerimientos, es decir el 22%.

Esta información se la obtuvo directamente de la base de datos de la empresa.

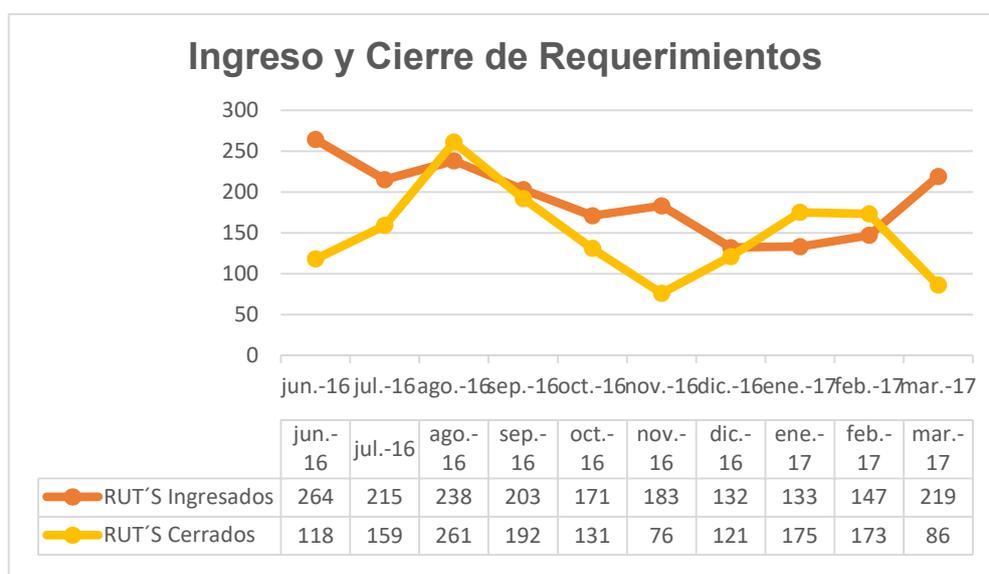


Figura 3.2: Estadística de ingresos y cierres de requerimientos

Estadísticas del origen de los requerimientos

El origen de los RUT puede ser: Interno o por clientes. Los requerimientos internos nacen dentro de la empresa, por lo general son creados cuando existe alguna regulación de gobierno como las que llegan por parte del SRI (Servicios de Rentas Interna), MRL(Ministerio de Relaciones Laborales), etc. y deben ser desarrolladas como por ejemplo es el caso de modificaciones en la estructura del ATS (Anexos transaccional simplificado).

Los requerimientos de clientes son reportados por estos, y tienen por lo general mayor nivel de prioridad. Pueden ser por necesidad de soporte, defectos del sistema presentados en producción, desarrollo a medida, etc.

La Figura 3.3 representa la cantidad de requerimientos reportados durante el periodo de junio 2016 a marzo de 2017, clasificados según su origen, con lo cual se obtuvo que del total de los 1905 requerimientos ingresados, 1812 tienen como origen el cliente, los 93 restantes fueron de origen interno.

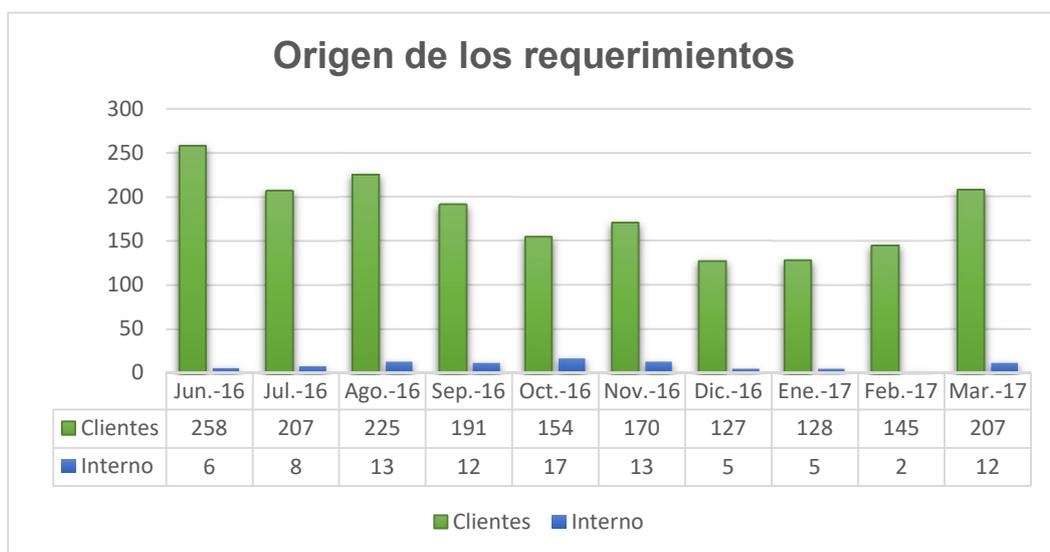
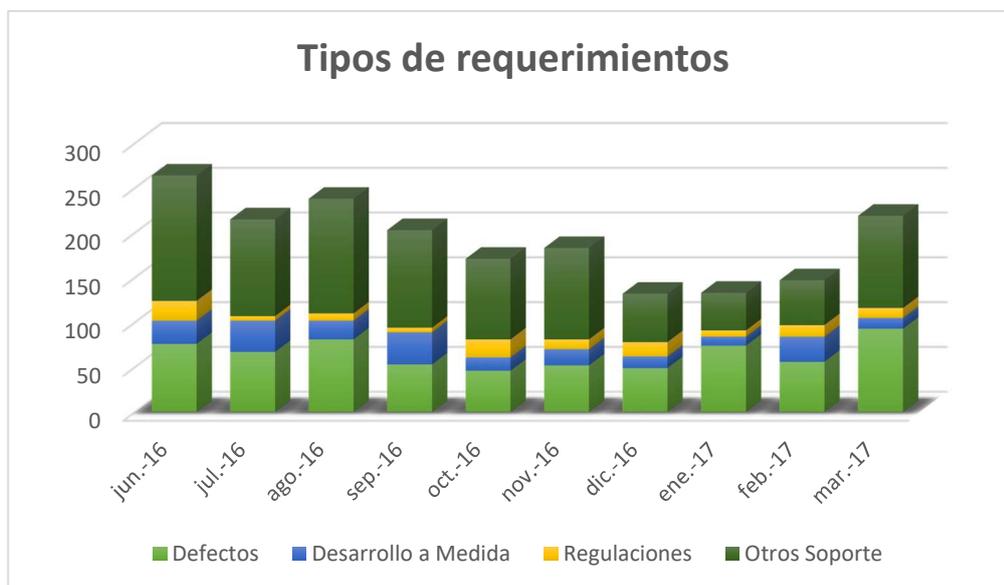


Figura 3.3: Estadística de origen de requerimientos

Clasificación de requerimientos recibidos

La Figura 3.4 presenta la clasificación de los requerimientos recibidos por parte de los usuarios durante el periodo de junio 2016 a marzo de 2017.



	jun-16	jul-16	ago-16	sep-16	oct-16	nov-16	dic-16	ene-17	feb-17	mar-17	% Promedio
Otros Soporte	140	108	128	109	90	102	54	42	50	103	49%
Regulaciones	22	5	8	5	20	11	16	7	13	11	6%
Desarrollo a Medida	26	35	21	36	15	18	13	10	28	12	11%
Defectos	76	67	81	53	46	52	49	74	56	93	34%

Figura 3.4: Clasificación de los requerimientos registrados

Como podemos observar en el gráfico en promedio el 34% de los requerimientos ingresados durante este período corresponden a defectos presentados en ambiente de producción, esto ocasiona malestar por parte de los clientes/usuarios y deben ser atendidos con prioridad alta.

Los requerimientos presentados por regulaciones de gobierno o tributarias representaron un promedio de 6% del total registrado

durante ese período, mientras que los registrados por desarrollo a medida o personalizaciones correspondieron al 11%. Estos tipos de requerimientos deben ser analizados profundamente y contar siempre con el usuario final involucrado en el avance del desarrollo del mismo.

Asignación de recursos para atención de requerimientos en soporte de primer nivel

El 47% de los requerimientos que ingresaron por origen de clientes debe ser atendido en primera instancia por soporte de I nivel.

La asignación de estos requerimientos para atención en soporte de primer nivel por lo general es de inmediato, siendo así que el promedio en demora de asignación obtenido en días durante el período de junio 2016 a marzo de 2017 es de 0.5 días.

3.3. ANÁLISIS DE PROCESOS DEL ÁREA OPERATIVA

3.3.1. PROCESOS DEL ÁREA OPERATIVA

El área de operaciones cuenta con los sub-departamentos de Soporte y Desarrollo. El Sub-departamento de Desarrollo a su vez está compuesto por los Desarrolladores y los Analistas Funcionales, los cuales están relacionados entre sí. Las actividades realizadas en cada una de las áreas crean valor para las demás áreas.

El área operativa cuenta con dos comités que trabajan y organizan los procesos:

- Comité Técnico
- Comité de Calidad

Comité Técnico

El Comité Técnico es el encargado de evaluar, aprobar, analizar y diseñar, los diferentes sistemas que se desarrollan, lo cual realiza durante sus reuniones quincenales. En la Figura 3.5 se puede observar cómo está conformado este comité.



Figura 3.5. Composición del Comité Técnico

Comité de Calidad

Entre las funciones del Comité de Calidad están la aprobación de procedimientos y políticas de los cambios que se realicen en la forma de trabajo, la evaluación de las formas de trabajo, mejoras de documentos y la realización de reuniones ordinarias una vez al mes. En la Figura 3.6 se puede observar cómo está



Figura 3.6. Composición del Comité de Calidad

El trabajo de los sub-departamentos junto a los comités antes descritos involucra los procesos listados a continuación los cuales brevemente se describen en esta sección:

- **Proceso de Soporte Técnico y de Usuario:**

Es un proceso de apoyo dirigido por el Coordinador de Soporte junto con el jefe del área de operaciones, los involucrados en este proceso trabajan directamente con el usuario final. Está encargado del soporte básico de primer nivel, es decir de la recepción, revisión y análisis del requerimiento. En algunos casos este proceso brinda la solución inmediata, pero si el requerimiento necesita otro tipo de solución, por ejemplo defectos encontrados en producción o nuevas funcionalidades, entonces se escala la solicitud al proceso de desarrollo y mantenimiento de software.

El área de soporte es la encargada de realizar el registro en la plataforma de todos los incidentes-requerimientos reportados a través de:

- Correo electrónico a la cuenta de soporte.
- Contratos recibidos del área comercial.

Los requerimientos pueden ser:

1. Nuevos requerimientos por actualizaciones reguladas por entes tributarios, gubernamentales, etc.
2. Desarrollo a medida.
3. Defectos presentados en producción.
4. Instalaciones y reinstalaciones de las aplicaciones.
5. Nuevos requerimientos de soporte de primer nivel y/o capacitación.
6. Actualización de versión.

Los tres primeros casos requerirán el paso al proceso de desarrollo y mantenimiento, por lo cual el líder de soporte debe registrar la respectiva Orden de Trabajo (OT). Los casos restantes los podrá atender directamente el personal de soporte. Para el tercer caso, primero pasa por soporte de primer nivel donde se valida que efectivamente existe el defecto en producción.

Finalmente el personal de soporte para poder cerrar una orden de trabajo (OT) que ingreso a desarrollo, luego de la

actualización, debe asegurarse que el cliente se encuentre satisfecho y proceda a certificar el trabajo realizado.

- ***Proceso de desarrollo y mantenimiento de software:***

Este proceso es dirigido por el líder de proyectos, junto con el jefe del área de operaciones. El propósito del proceso de desarrollo y mantenimiento de software es la producción eficaz y eficiente de productos de software de calidad que reúnan las condiciones y especificaciones solicitadas por el cliente/usuario. De cómo se lleve a cabo este proceso y el resultado obtenido depende la satisfacción y fidelidad del Cliente/Usuario. Actualmente se trabaja bajo la creatividad y juicio de los involucrados.

Internamente se desarrollan los siguientes subprocesos:

- ***Gestión de proyectos***

Dentro de los tipos de requerimientos que involucran desarrollo tenemos dos tipos, los registrados por:

- Defectos de la aplicación en producción y,

- Nuevas Funcionalidades (incluyen actualizaciones reguladas por ley, desarrollo a medida, mantenimiento de funciones).

El proceso de desarrollo y mantenimiento inicia con la recepción del requerimiento, este es evaluado y validado por el líder del área, junto con un analista funcional designado por este y el usuario final. En base a ciertos criterios previamente definidos se establece la prioridad del mismo y se realizan las estimaciones de tiempo (Inicio y finalización del requerimiento).

En este punto el subproceso de Gestión de Proyectos se manejará dependiendo del tipo de requerimiento que se tenga. Para el caso de nuevas funcionalidades el líder de proyecto evalúa las condiciones necesarias para su desarrollo, y se establecen las fechas de reuniones con el cliente/usuario, ya que es de vital importancia mantener al usuario involucrado con el objetivo de analizar a un nivel más bajo el requerimiento y entender bien sus necesidades. El responsable de llevar a cabo las reuniones es el analista funcional designado por el líder del proyecto.

El analista funcional planificará las reuniones que crea convenientes con el usuario, para luego presentar al equipo de trabajo el resultado obtenido: la documentación de especificaciones, documento que debe ser validado por el usuario líder.

El líder de proyecto debe elaborar el Plan del Desarrollo del requerimiento o proyecto. Dicho plan contiene el cronograma de planificación y define el personal que necesita involucrar en todas las actividades a realizar (programadores, analistas funcionales, control de calidad, administrador de base de datos). Posteriormente se realiza la Sesión Kick Off (arranque de proyecto), en donde se valida la planificación realizada.

El líder de proyecto realiza un constante seguimiento a las actividades que se realizan posteriormente tales como desarrollo del software, elaboración de casos de prueba, el control de calidad, capacitación de usuarios, control del entregable en el

ambiente de producción y finalmente el cierre del proyecto, con el propósito de asegurarse que todas las actividades se ejecuten de acuerdo a lo acordado en la planificación del proyecto.

- **Desarrollo de software**

Para dar Inicio a este subproceso se necesita tener listo el Informe Funcional y el Técnico, el cual es responsabilidad del analista funcional y del arquitecto de software.

A través de este documento se plantean:

- La arquitectura de la solución.
- Las interfaces de comunicación a desarrollar.
- Implementación de las reglas de negocio indicadas en la especificación funcional.

Dicho documento es validado por el comité técnico junto con el líder del proyecto, quienes tendrán que aprobarlo.

Una vez aprobado el documento, el analista inicia con la elaboración de los casos de pruebas y además se crea en el repositorio de versionamiento todo lo necesario para que se inicie

con la programación del sistema, para lo cual actualmente se utiliza Visual Sourcesafe (VSS).

Por otro lado, el analista desarrollador gestiona las condiciones del ambiente de trabajo que necesita para comenzar con la codificación y desarrollo de los programas. Una vez que concluya la fase de programación, realiza las pruebas unitarias y ajusta lo que sea necesario. Cuando culmina tales pruebas realiza la integración de los componentes respectivos para iniciar las pruebas de integración y los ajustes correspondientes.

Finalmente, cuando el entregable se encuentra sin incidencias congela la versión de los programas fuentes en el repositorio de versionamiento.

- **Control de calidad**

Una vez concluido el desarrollo del requerimiento y realizadas las respectivas pruebas se elabora el documento de Solicitud de Pase a Certificación, esto es responsabilidad del analista

desarrollador. Este entregable es validado por el comité de calidad. El analista funcional revisa que en la documentación entregada se encuentre todo lo necesario para la ejecución del pase en los ambientes de certificación, tal como: congelamiento de los programas fuentes, evidencias de las pruebas de integración, manual para realizar el despliegue de los componentes, entre otros. Luego de verificar que todo esté en orden, el analista de control de calidad se encarga de realizar el despliegue de la solución en el ambiente de certificación.

Si el analista funcional encuentra alguna inconsistencia o error al momento de aplicar el pase, revierten todos los pasos realizados para dejar el ambiente sin ninguna alteración, pero además lo comunica al desarrollador y espera hasta que la inconsistencia sea corregida. Caso contrario, de realizarse correctamente el despliegue, el siguiente paso será que el analista de control de calidad inicie con las pruebas del sistema.

Como resultado de tales pruebas funcionales, el analista de control de calidad, en acuerdo con el analista funcional,

determinan si las funcionalidades implementadas se encuentran en buenas condiciones como para iniciar las pruebas con los usuarios principales.

El principal resultado de este proceso es el informe de las pruebas que elabora el analista de control de calidad. En este documento se evidencian las observaciones o errores que se han encontrado, tanto del lado del analista como del usuario.

- ***Seguimiento y control***

Este *proceso* se realiza de manera constante una vez que el plan de proyecto esté aprobado y además se hayan iniciado las etapas siguientes al análisis de requerimientos. El jefe de proyecto en función a su análisis y seguimiento, a través de las distintas reuniones que tiene con el equipo de proyecto, actualiza periódicamente el cronograma de trabajo y elabora el informe de estado.

Dicho informe es revisado con el líder usuario y otros integrantes del proyecto. Como resultado de esta reunión, se identifica si existen cambios en relación a lo planificado. De ser así, el jefe de proyecto elabora el documento "Control de Cambio" en el cual se especifican las modificaciones a realizar en el proyecto y como estas impactan en el costo y tiempo. Este documento es validado por el líder usuario, su aceptación implica actualizar el plan del proyecto y el cronograma de trabajo. No obstante, independientemente de dicha aceptación todos los cambios identificados son documentados en la bitácora de cambios del proyecto.

3.3.2. IDENTIFICACIÓN DE LOS PRINCIPALES PARTICIPANTES DE LOS PROCESO DEL ÁREA OPERATIVA

A continuación se mencionan los principales participantes involucrados en los procesos del área operativa con la respectiva descripción de sus funciones:

Líder de proyecto de desarrollo

- Dirige y coordina los proyectos de desarrollo y mantenimiento de las aplicaciones.
- Supervisa las funciones y los recursos de análisis funcional, técnico y programación, asegurando la adecuada explotación de los recursos asignados, con el objetivo de satisfacer las necesidades de los usuarios.

Analista funcional

- Analiza y documenta los procesos, requerimientos técnicos, requerimientos de negocio, etc.
- Realiza el levantamiento de requerimientos de usuario de manera detallada para posterior envío a desarrollo.

- Detecta eventuales omisiones ocurridas durante el desarrollo de las aplicaciones, para luego solicitar la corrección al programador.
- Elabora los casos de prueba, y la forma en que se visualizarán los resultados esperados por las modificaciones.

Analista – Desarrollador

- Se encarga del desarrollo de sistemas de diversa complejidad, efectuando análisis de requerimientos, diseños físicos, diseños lógicos, ejecutando tareas de programación.
- Realiza el mantenimiento de las aplicaciones de software.

Analista de control de calidad

- Encargado de ejecutar los casos de pruebas para el control de calidad del software, en el ambiente dispuesto para esto.
- Evalúa los resultados obtenidos de las pruebas.
- Obtiene la validación y aprobación de las definiciones del usuario.
- Verifica el cumplimiento de los requerimientos desde el punto de vista del usuario.

Arquitecto de Software

- Encargado de realizar la documentación con especificaciones técnicas del requerimiento.

Coordinador de soporte de usuario

- Supervisa y coordina la asistencia y soporte al usuario, garantizando la calidad y oportuna prestación del servicio.

Analista de Soporte de técnico y de usuario de primer nivel

- Se encarga de la atención directa al cliente.
- Soluciona las novedades de primer nivel presentadas por los usuarios.
- Realiza las implementaciones del sistema, verifica que las infraestructuras donde se las realizarán sean las adecuadas.
- Realiza las capacitaciones a los usuarios internos y externos de las nuevas funcionalidades o sobre los cambios por mantenimiento realizadas sobre las mismas.

CAPÍTULO 4

ANÁLISIS, DISEÑO DE LA PROPUESTA DE IMPLEMENTACIÓN DE GESTIÓN DE PROCESOS DE NEGOCIOS CON METODOLOGÍAS ÁGILES

4.1. GESTIÓN DE PROCESOS DE NEGOCIOS APLICADO AL DESARROLLO Y MANTENIMIENTO DE SOFTWARE

Para el análisis, diseño e implementación de la Gestión de Procesos de Negocio se utilizaron herramientas que ofrece Bizagi Studio, debido a que estas herramientas ofrecen un ambiente colaborativo donde los usuarios involucrados en el proceso pueden trabajar simultáneamente.

Bizagi Studio divide las tareas de diseño e implementación del Sistema de Gestión de Negocio en seis fases las cuales se mencionan a continuación:

1. Diagrama BPMN del Proceso de Desarrollo y Mantenimiento de software.
2. Definición del modelo de datos.
3. Diseño y generación de las interfaces de usuario.
4. Definición de las reglas de negocio.
5. Definición de los participantes y sus roles.
6. Ejecución del proceso.

4.1.1. DIAGRAMA BPMN DEL PROCESO DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE

Basados en la descripción detallada del proceso de desarrollo y mantenimiento de software realizado en la sección 3.3.1, se definió el diagrama BPMN del proceso en estudio, el cual podemos observar en el Anexo 1

4.1.2. DEFINICIÓN DEL MODELO DE DATOS

Posterior a la definición del diagrama BPMN del proceso de desarrollo y mantenimiento de software, se procedió con la definición del modelo de datos, que es donde se organiza la información que será utilizada a través de las interfaces de usuario, a lo largo de todo el proceso. La Figura 4.1 muestra el modelo de datos para el proceso de desarrollo y mantenimiento de software:

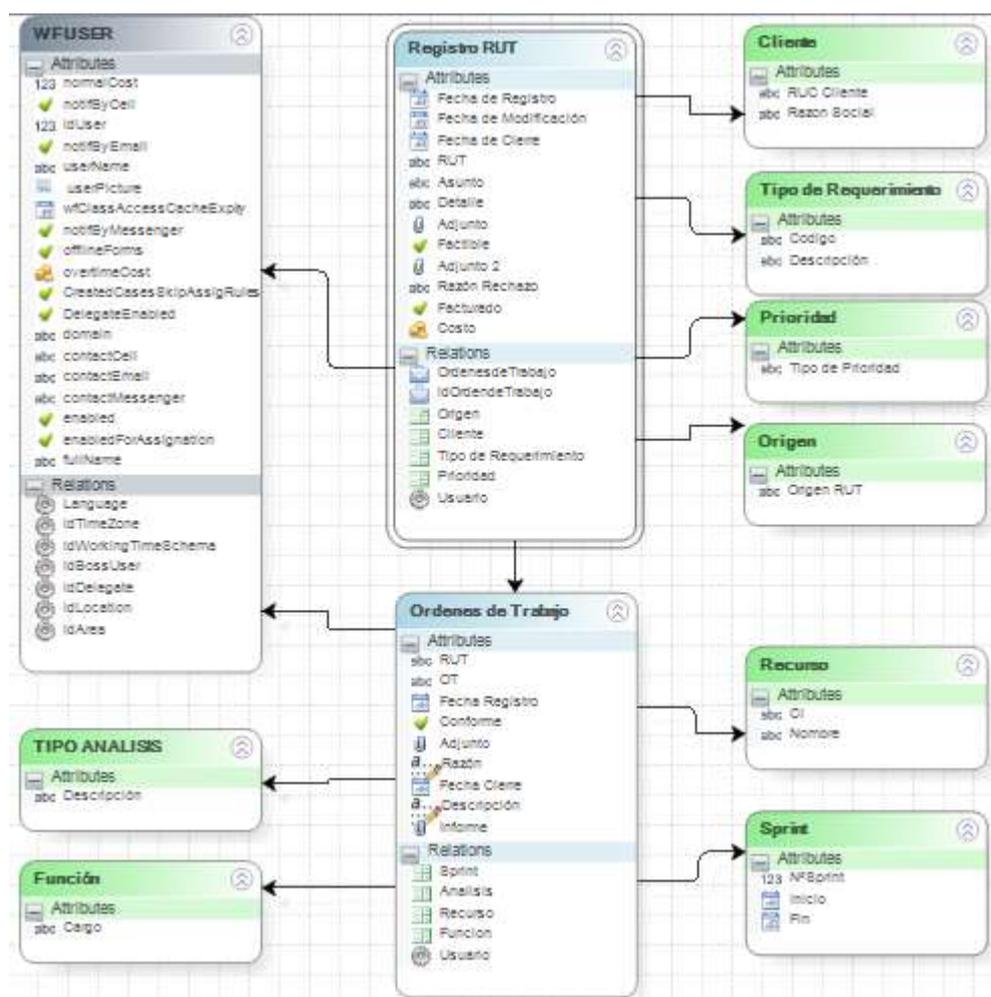


Figura. 4.1: Modelo de Datos

Es importante realizar la definición del modelo de datos de manera correcta, con el fin de garantizar que las actividades solicitarán la información necesaria y por consiguiente el proceso se concluirá con éxito. Para el presente caso se definieron las siguientes entidades, las cuales se detallarán según su tipo:

- Entidades Maestras:

En Bizagi las Entidades Maestras se presentan en color azul y almacenarán la información que se relaciona directamente a los casos de los procesos. Entre las Entidades Maestras que se definieron tenemos:

- Registro de RUT

Guardará los datos referentes al ingreso del requerimiento, tal como se presenta en la Tabla 7:

Tabla 7: Registro de RUT

Nombre	Descripción
Fecha de Registro	Fecha de registro del requerimiento.
Fecha de Modificación	Fecha de modificación del requerimiento.
Fecha de Cierre	Fecha de cierre del requerimiento.
Origen	Origen del requerimiento (seleccionar parámetros de catálogo predefinido).
Cliente	Cliente que solicita el requerimiento (seleccionar parámetros de catálogo predefinido).

Nombre	Descripción
Tipo de requerimiento	Tipo de requerimiento (seleccionar parámetros de catálogo predefinido).
Prioridad	Prioridad que se le asigna al requerimiento (seleccionar parámetros de catálogo predefinido).
Asunto	Resumen del requerimiento.
Detalle	Descripción detallada del requerimiento.
Adjunto	Adjuntar archivos.
Factible	SI/NO factibilidad del requerimiento.
Adjunto2	Adjuntar archivo sobre razón de rechazo.
Razón Rechazo	Detalle del rechazo.
Facturado	SI/NO requerimiento es facturado.
Costo	Valor de cobro por el requerimiento.
Usuario	Usuario que registra el requerimiento.
RUT	Código del requerimiento.

En la Figura 4.2 se muestran los atributos parametrizados en la entidad Registro de RUT, así como su respectivo tipo de dato.

Attribute list for Registro RUT

Attributes List

+ Add - Remove

Display Name	Name	Type	State
Fecha de Registro	FechaCreacion	Date - time	
Fecha de Modificación	FechaModificacion	Date - time	
Fecha de Cierre	FechaCierre	Date - time	
Origen	Origen	Origen	
Cliente	Cliente	Cliente	
RUT	RUT	String	
Tipo de Requerimiento	TipodeRequerimiento	Tipo de Requerimiento	
Prioridad	Prioridad	Prioridad	
Asunto	Asunto	String	
Detalle	Detalle	String	
Adjunto	Adjunto	File	
Factible	Factible	Boolean (Yes - No)	
Adjunto 2	Adjunto2	File	
Razón Rechazo	RazonRechazo	String	
Facturado	Facturado	Boolean (Yes - No)	
Costo	Costo	Currency	
Usuario	Usuario	WFUSER	

Figura. 4.2: Registro de RUT

- Órdenes de trabajo

En esta entidad se guardarán los datos referentes a las órdenes de trabajo que se generan por un RUT. Un RUT puede tener varias órdenes de trabajo asignadas al equipo dependiendo de las actividades a realizar tal como: análisis, desarrollo, certificación, etc. En la Tabla 8 se presenta la descripción.

Tabla 8: Órdenes de Trabajo

Nombre	Descripción
RUT	Código de RUT.
OT	Código de Orden de Trabajo.
Sprint	Número de Sprint.
Análisis	Tipo de Análisis.
Fecha de registro	Fecha de registro de Orden de trabajo.
Recurso	Recurso asignado a la Orden de trabajo (seleccionar parámetros de catálogo predefinido).
Función	Función del recurso asignado (seleccionar parámetros de catálogo predefinido).
Fecha de Cierre	Fecha de cierre de la Orden de trabajo.
Usuario	Usuario que registra la OT.
Conforme	SI/NO Aprobado.
Adjunto	Archivo Adjunto generado luego de la reunión de sprint inicial.
Razón	Razón de Rechazo.

En la Figura 4.3 se muestran los atributos parametrizados en la entidad Registro de Órdenes de trabajo, así como su respectivo tipo de dato.

Display Name	Name	Type	State
RUT	RUT	String	
OT	OT	String	
Sprint	Sprint	Sprint	
Analisis	Analisis	Analisis	
Fecha Registro	FechaRegistro	Date - time	
Recurso	Recurso	Recurso	
Funcion	Funcion	Función	
Fecha Cierre	FechaCierre	Date - time	
Usuario	Usuario	WFUSER	
Conforme	Conforme	Boolean (Yes - No)	
Adjunto	Adjunto	File	
Razon	Razon	Extended Text	

Figura. 4.3: Órdenes de Trabajo

- Entidades Paramétricas

En Bizagi las Entidades Paramétricas se presentan en color verde y almacenarán las listas con valores predefinidos (Catálogos). A continuación la descripción de las entidades paramétricas que se utilizarán en el modelo:

- Cliente

En esta entidad registraremos los clientes que mantiene la empresa: RUC y Razón Social. En la Figura 4.4 se presentan los

atributos parametrizados en la entidad Cliente, así como su respectivo tipo de dato.



The screenshot shows a window titled 'Attribute list for Cliente'. Inside, there is a section 'Attributes List' with 'Add' and 'Remove' buttons. Below is a table with columns: Display Name, Name, Type, and State. The table contains two rows:

Display Name	Name	Type	State
RUC Cliente	RUC	abc String	
Razon Social	RazonSocial	abc String	

Figura 4.4: Cliente

- Tipo de Requerimiento

En esta entidad registraremos el catálogo de tipos de requerimientos predefinidos por la organización, los cuales pueden ser, entre otros: Por Defectos, Nuevos requerimientos, etc. En la Figura 4.5 se presentan los atributos parametrizados en la entidad Tipo de Requerimiento, así como su respectivo tipo



The screenshot shows a window titled 'Attribute list for Tipo de Requerimiento'. Inside, there is a section 'Attributes List' with 'Add' and 'Remove' buttons. Below is a table with columns: Display Name, Name, Type, and State. The table contains two rows:

Display Name	Name	Type	State
Codigo	Codigo	abc String	
Descripción	Descripcion	abc String	

Figura 4.5: Tipo de Requerimiento

- Prioridad

En esta entidad registraremos el catálogo de prioridades predefinidas por la organización, las cuales pueden ser: Alto, Medio Alto, Medio, Bajo. En la Figura 4.6 se presentan los atributos parametrizados en la entidad Prioridad, así como su respectivo tipo de dato.



Attribute list for Prioridad

Attributes List

+ Add - Remove

Display Name	Name	Type	State
Tipo de Prioridad	Tipo_Prioridad	abc String	

Figura 4.6: Prioridad

- Origen

En esta entidad registraremos el catálogo de origen de requerimientos predefinidos por la organización, los cuales pueden ser: Interno / Externo. En la Figura 4.7 se presentan los atributos parametrizados en la entidad Origen, así como su respectivo tipo de dato.



Attribute list for Origen

Attributes List

+ Add - Remove

Display Name	Name	Type	State
Origen RUT	OrigenRUT	abc String	

Figura 4.7: Origen de Requerimientos

- Recursos

En esta entidad registraremos el catálogo de los recursos disponibles predefinidos por la organización. En la Figura 4.8 se presentan los atributos parametrizados en la entidad Recursos, así como su respectivo tipo de dato.



The screenshot shows a window titled "Attribute list for Recurso". Inside, there is a section labeled "Attributes List" with "Add" and "Remove" buttons. Below this is a table with the following data:

Display Name	Name	Type	State
CI	CI	abc String	
Nombre	Nombre	abc String	

Figura 4.8: Recursos

- Función

En esta entidad registraremos el catálogo de los Funciones o Cargos predefinidos por la organización. En la Figura 4.9 se presentan los atributos parametrizados en la entidad Función, así como su respectivo tipo de dato.



The screenshot shows a window titled "Attribute list for Función". Inside, there is a section labeled "Attributes List" with "Add" and "Remove" buttons. Below this is a table with the following data:

Display Name	Name	Type	State
Cargo	Cargo	abc String	

Figura 4.9: Funciones

- Tipo de Análisis

En esta entidad registraremos el catálogo predefinido por la empresa de los tipos de análisis que se llevará a cabo durante el proyecto, los cuales pueden ser: Análisis Funcional, Análisis Técnico, Pruebas unitarias, Pruebas Integrales, Pruebas de Certificación. En la Figura 4.10 se presentan los atributos parametrizados en la entidad tipo de análisis, así como su respectivo tipo de dato.

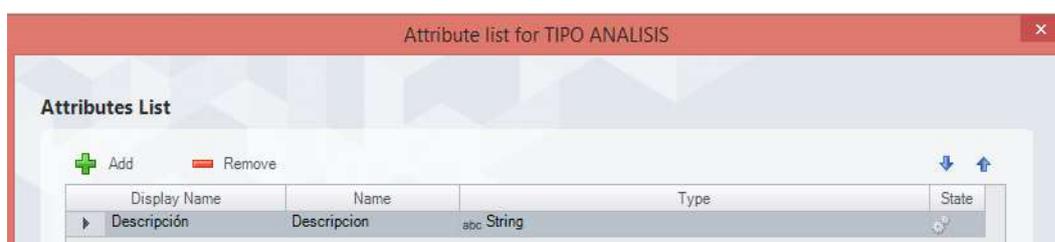


Figura 4.10: Tipo de Análisis

- Sprint

En esta entidad registraremos el catálogo de los Sprint predefinidos por la organización. En la Figura 4.11 se presentan los atributos parametrizados en la entidad sprint, así como su respectivo tipo de dato.



Figura 4.11: Sprint

- Entidades del Sistema

En Bizagi las Entidades del Sistema se presentan en color gris. Para este modelo utilizaremos la entidad WFUser, la cual almacena los usuarios finales del proyecto. Esta entidad no permite modificar sus atributos.

4.1.3. DISEÑO Y GENERACIÓN DE LAS INTERFACES DE USUARIO

En esta sección se define la interfaz que facilitará la interacción con los usuarios del proceso.

Las siguientes figuras representan el diseño de las interfaces de usuario creadas para organizar los datos generados por las actividades:

- Registro Universal de Trabajo - RUT

En esta pantalla se registrarán los requerimientos (RUT). Entre los datos de ingreso obligatorio tenemos:

- Fecha de Registro
- Usuario

- Origen del Requerimiento
- RUT (Código)
- Tipo de Requerimiento
- Asunto
- Detalle

En la Figura 4.12 podemos ver el diseño de la pantalla para el registro de requerimientos.

REGISTRO UNIVERSAL DE TRABAJO - RUT

Fecha de Registro: M/d/yyyy

Fecha de Modificación: M/d/yyyy

Usuario: Item 1

Origen: Item 1

Cliente: Item 1

RUT: abc

Tipo de Requerimiento: Item 1

Detalle de Requerimiento

Asunto: abc

Detalle: abc

Adjunto: No files uploaded

Figura 4.12: Registro Universal de Trabajo

- Informe de Factibilidad de RUT

Esta pantalla traerá los datos registrados en el paso anterior para que posteriormente el líder del proyecto defina si el RUT es factible de realizar.

En la Figura 4.13 podemos ver el diseño de la pantalla para el registro del análisis de factibilidad del requerimiento.

ANÁLISIS DE FACTIBILIDAD DE RUT

Fecha de Registro:	1/1/1900	Fecha de Modificación:	M/d/yyyy
Usuario:	Item 1		

Origen: Item 1 Cliente: Item 1

RUT: abc Tipo de Requerimiento: Item 1

Detalle de Requerimiento

Asunto: abc

Detalle: abc

Adjunto: No files uploaded

Factibilidad

Factible: Yes No

Figura 4.13: Análisis de Factibilidad de RUT

- Informe de NO Factibilidad

En el caso que en el paso anterior se defina como NO factible la realización del requerimiento, será mandatorio registrar la razón

del rechazo y la fecha de cierre para poder dar por terminado el requerimiento.

En la Figura 4.14 podemos ver el diseño de la pantalla para el registro de un informe negativo de factibilidad de un requerimiento.

INFORME DE NO FACTIBILIDAD DE RUT			
Fecha de Registro:	1/1/1900	Fecha de Modificación:	M/d/yyyy
Usuario:	Item 1	Fecha de Cierre:	M/d/yyyy
▼ Datos Generales			
Origen:	Item 1	Cliente:	Item 1
RUT:	abc	Tipo de Requerimiento:	Item 1
▼ Detalle de Requerimiento			
Asunto:	abc		
Detalle:	abc		
Adjunto:	No files uploaded		
▼ Factibilidad			
Factible:	Yes	Adjunto 2:	No files uploaded
Razón Rechazo:	abc		

Figura 4.14: Informe de NO factibilidad del RUT

- Asignación de Prioridad a RUT

En caso de que el requerimiento si sea factible de desarrollar, se presentará esta ventana donde se procederá a la asignación de prioridad.

En la Figura 4.15 podemos ver el diseño de la pantalla para asignación de prioridad.

The screenshot shows a web form titled "ASIGNACIÓN DE PRIORIDAD A RUT". The form is divided into several sections:

- Header:** "ASIGNACIÓN DE PRIORIDAD A RUT"
- Form Fields:**
 - Fecha de Registro: 1/1/1900
 - Fecha de Modificación: M/d/yyyy (with a calendar icon)
 - Usuario: Item 1
 - Origen: Item 1
 - Cliente: Item 1
 - RUT: abc
 - Tipo de Requerimiento: Item 1
 - Priorización (expanded):
 - Prioridad: Item 1 (with a dropdown arrow)

Figura 4.15: Asignación de Prioridad

- Registro de Órdenes de Trabajo

En esta pantalla se procederá al registro de las órdenes de trabajo que se deriven de un requerimiento, las cuales serán asignadas a cada participante.

En la Figura 4.16 podemos ver el diseño de la pantalla para el registro de las órdenes de trabajo.



The screenshot shows a web form titled "REGISTRO DE ORDEN DE TRABAJO". Below the title is a section labeled "Asignación de recursos". The form contains six input fields arranged in two columns. The left column has: "Fecha Registro:" with a date picker showing "M/d/yyyy"; "OT:" with a text input containing "abc"; and "Recurso:" with a dropdown menu showing "Item 1". The right column has: "RUT:" with a text input containing "abc"; "Sprint:" with a dropdown menu showing "Item 1"; and "Funcion:" with a dropdown menu showing "Item 1".

Figura 4.16: Registro de Orden de trabajo

- Registro de Informe de Sprint Inicial

En esta pantalla se registrará la documentación con el detalle del requerimiento, generada luego de realizada la reunión de inicio de Sprint, y que posteriormente se utilizará como base para la elaboración de los documentos funcionales y técnicos.

En la Figura 4.17 podemos ver el diseño de la pantalla para el registro del informe de la reunión de inicio de sprint.

INFORME DE REUNIÓN SPRINT INICIAL			
Fecha de Registro:	1/1/1900	Fecha de Modificación:	M/d/yyyy
Usuario:	Item 1		
▼			
Origen:	Item 1	Cliente:	Item 1
RUT:	abc	Tipo de Requerimiento:	Item 1
▼ Detalle de Requerimiento - Informe Sprint Inicial			
Asunto:	abc		
Detalle:	abc		
Adjunto:	No files uploaded		

Figura 4.17: Registro de informe generado en reunión de Sprint Inicial

- Registro de Documentación

En esta pantalla se registrará la documentación generada por los diferentes participantes involucrados en el requerimiento.

En la Figura 4.18 podemos ver el diseño de la pantalla para el registro de documentación.

REGISTRO DE DOCUMENTACIÓN

Documentación

Fecha:	M/d/yyyy	RUT:	abc
OT:	abc	TIPO:	Item 1
Recurso:	Item 1	Funcion:	Item 1

Detalle

Descripción:	abc	Informe:	No files uploaded
--------------	-----	----------	-------------------

Figura 4.18: Registro de Documentación

- Observaciones – Devuelto

En el caso que durante la revisión de la documentación registrada tenga observaciones y sea rechazada en esta pantalla se realizará el registro de dichas observaciones.

En la Figura 4.19 podemos ver el diseño de la pantalla para el registro de las Observaciones generadas durante la revisión de la documentación.

The screenshot displays a web form titled "OBSERVACIONES - DEVUELTO". It is organized into three main sections, each with a dropdown arrow on the left:

- Datos Generales:** Contains six fields in a two-column layout:
 - Fecha: 1/1/1900
 - RUT: abc
 - OT: abc
 - TIPO: Item 1
 - Recurso: Item 1
 - Funcion: Item 1
- Detalle:** Contains two fields:
 - Descripción: abc
 - Informe: No files uploaded
- Aprobación:** Contains three fields:
 - Conforme: A radio button group with "Yes" selected and "No" unselected.
 - Razon Rechazo: abc
 - Informe 2: No files uploaded

Figura 4.19: Registro de Aprobación – Observaciones

4.1.4. DEFINICIÓN DE LAS REGLAS DE NEGOCIOS

A lo largo del proceso nos encontramos con ciertos puntos determinados donde se deben verificar condiciones específicas, las que tendrán un desenlace: verdadero o falso.

En la Figura 4.20 podemos observar cómo se configuran las reglas de negocio en la herramienta Bizagi.

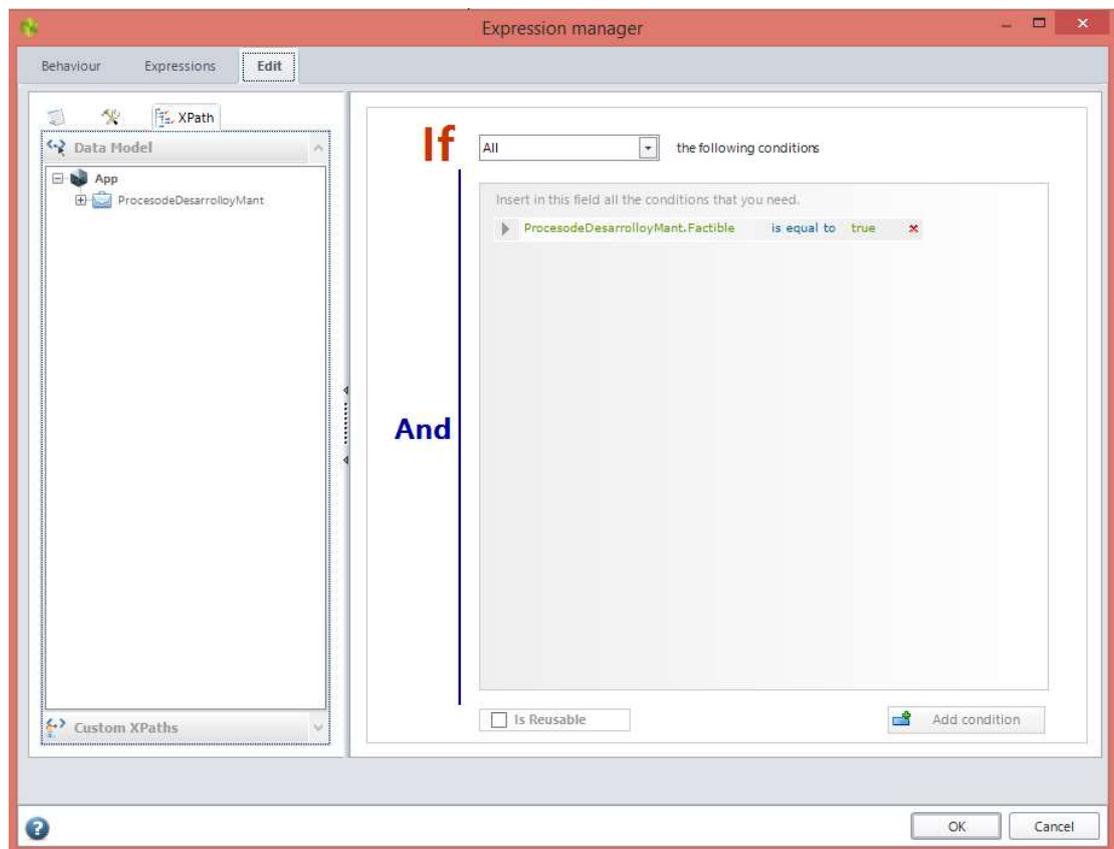


Figura 4.20: Definición de Reglas de Negocio

Para el proceso de desarrollo y mantenimiento de software, se plantearon reglas para las disyuntivas de procesos en los siguientes casos:

a) Fase de especificaciones

Reglas

- I. El líder de sistemas debe validar la factibilidad del requerimiento:

- Si: Debe determinar la prioridad del requerimiento.
- No: Debe elaborar el informe de No Factibilidad, y posteriormente se da por terminado el proceso.

A continuación en la Figura 4.21 se presenta el diagrama que identifica esta regla:



Figura 4.21: Factibilidad del requerimiento

b) Fase de Análisis y Diseño

Reglas:

- II. El analista funcional elabora el Documento Funcional del proyecto, este documento deber ser validado si está correcto por el arquitecto de software.
 - Si: El arquitecto de software elabora la documentación técnica.

- No: El arquitecto de software elabora el documento y devuelve las observaciones encontradas al analista funcional.

A continuación la Figura 4.22 muestra el diagrama que identifica esta regla:

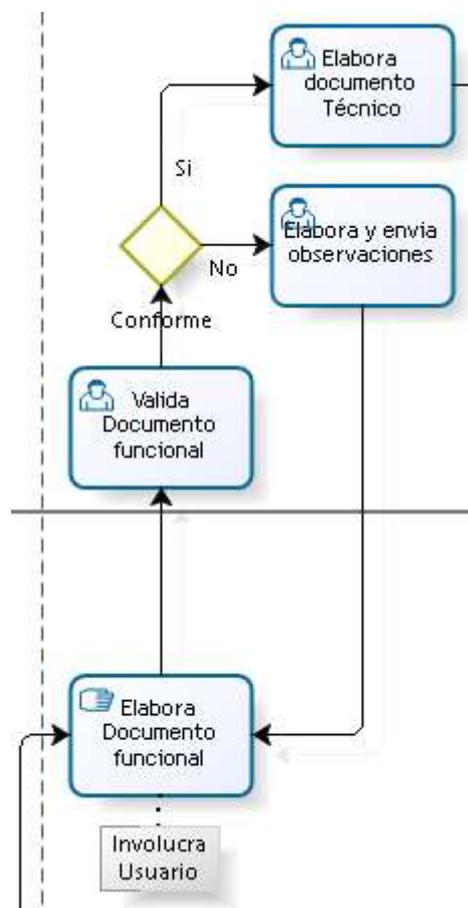


Figura 4.22: Validación de Documento Funcional

c) Fase de desarrollo

Reglas

III. El desarrollador realiza las pruebas unitarias y deber validar si estas fueron exitosas:

- Si: Debe integrar y empaquetar la solución, para proceder luego con las pruebas integrales.
- No: Regresa a desarrollo y realiza las respectivas correcciones.

A continuación la Figura 4.23 muestra el diagrama que identifica esta regla:

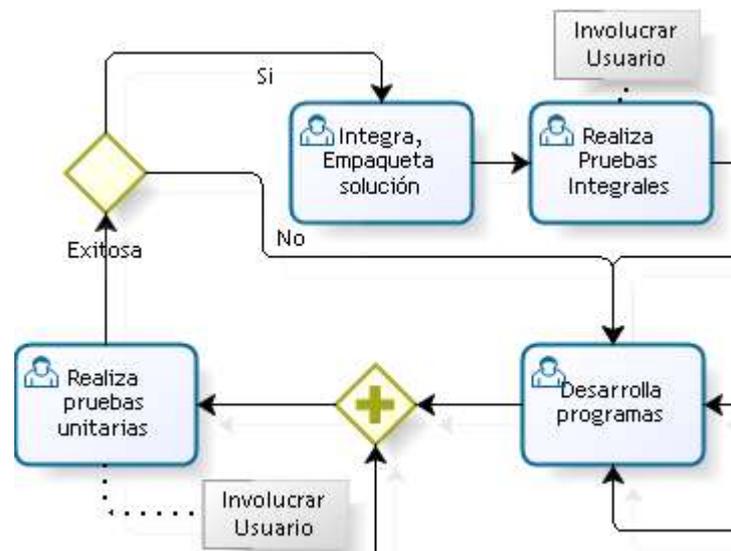


Figura 4.23: Validación de Pruebas Unitarias

IV. El desarrollador realiza las pruebas integrales, valida si estas son exitosas.

- Si: Congela versión de programas fuentes, para luego solicitar el pase a certificación.
- No: Regresa a desarrollo y realiza las respectivas correcciones.

A continuación en la Figura 4.24 se presenta el diagrama que identifica esta regla:

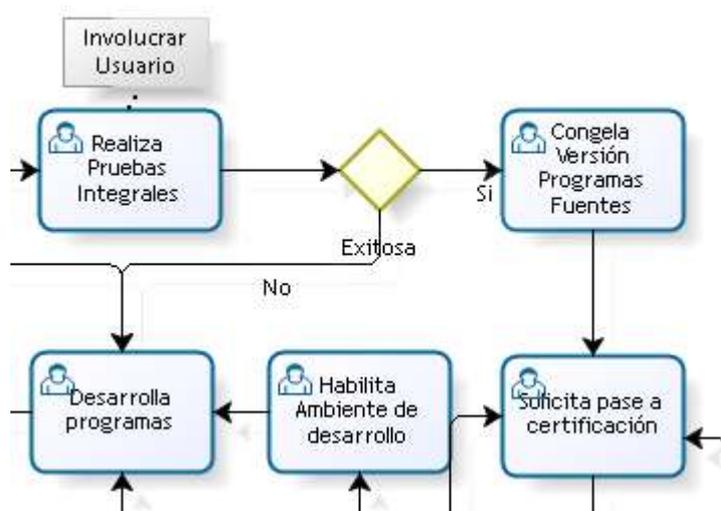


Figura 4.24: Validación de Pruebas Integrales

V. Una vez concluidas exitosamente las pruebas integrales, el desarrollador procede a elaborar un informe con los cambios realizados a nivel de Front end y back end. Solicita pase a certificación. El analista de control de calidad se

encarga de validar que los documentos enviados en la solicitud del pase estén completos y ejecuta posteriormente las pruebas de certificación.

- Si: Analista Funcional se encarga de preparar el ambiente y ejecuta el pase para certificación. Valida si todo está en orden.
- No: El Analista de control de calidad regresa las observaciones encontradas a desarrollo.

A continuación la Figura 4.25 muestra el diagrama que identifica esta regla:



Figura 4.25: Validar pase a Certificación

- VI. Una vez que se valida que la documentación del pase a producción es correcta el analista funcional prepara el ambiente de pruebas y ejecuta el pase para certificación.
- Si: Conforme con la ejecución el analista de control de calidad ejecuta pruebas de calidad. Utiliza las matrices de pruebas correspondientes.
 - No: Notifica para revertir el pase.

A continuación la Figura 4.26 muestra el diagrama que identifica esta regla:

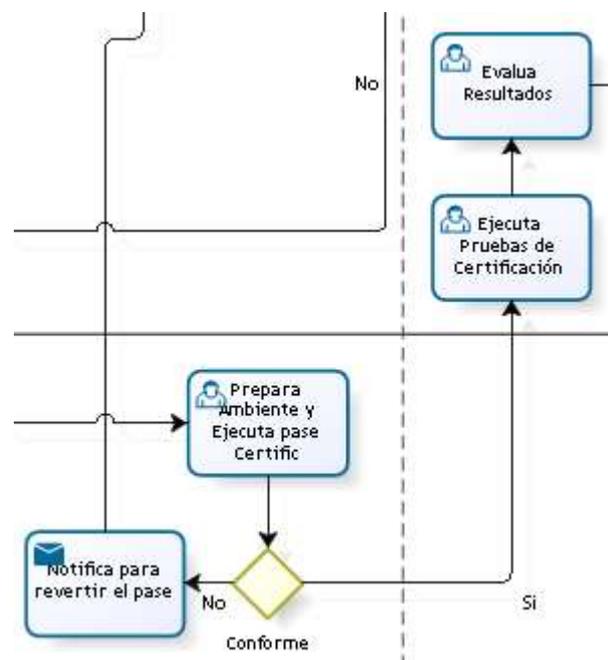


Figura 4.26: Validar Ejecución de pase a Certificación

d) Fase de Pruebas

Reglas

VII. El Analista de control de calidad ejecuta y evalúa el resultado de las pruebas realizadas, posteriormente define si necesita realizar pruebas con usuarios.

- Si: Debe ejecutar las pruebas con usuarios.
- No: Elabora informe de control de calidad.

A continuación la Figura 4.27 muestra el diagrama que identifica esta regla:

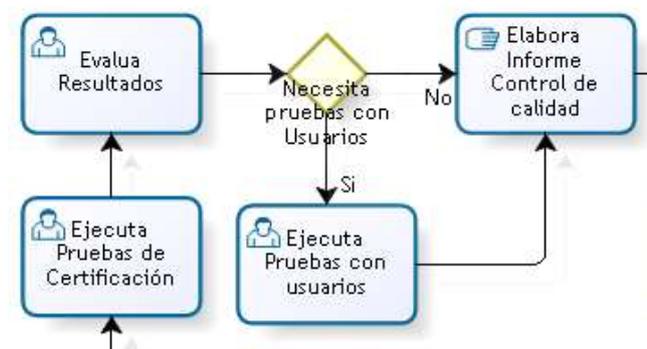


Figura 4.27: Ejecutar pruebas con usuarios

VIII. El analista de control de calidad evalúa los resultados obtenidos en las pruebas y elabora el respectivo informe de calidad.

- Si: Está conforme con el resultado obtenido, Solicita pase a producción.

- No: Elabora informe y devuelve a desarrollo para realizar las respectivas correcciones.

A continuación la Figura 4.28 muestra el diagrama que identifica esta l

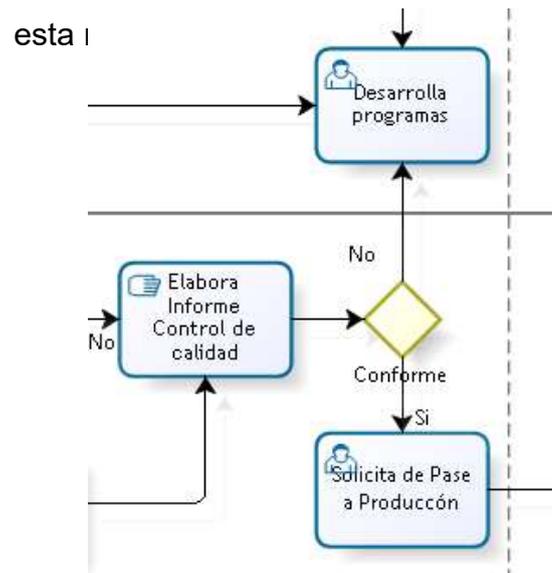


Figura 4.28: Validar Certificación

4.1.5. DEFINICIÓN DE PARTICIPANTES Y SUS ROLES

En esta sección se definen los participantes del proceso y los roles que estos tendrán. Para el proceso de desarrollo y mantenimiento de software usaremos tres tipos diferentes de asignaciones:

- La actividad Registrar Requerimiento será siempre ejecutada por el Creador del caso.

- La actividad Aprobar Requerimientos siempre será ejecutada por el Jefe del creador del caso.
- Todas las actividades del registro de documentación serán siempre ejecutadas por los Analistas (Funcional, Desarrollo, Arquitecto, Certificador).

En la Figura 4.29 podemos observar cómo se configuran los roles en la herramienta Bizagi.

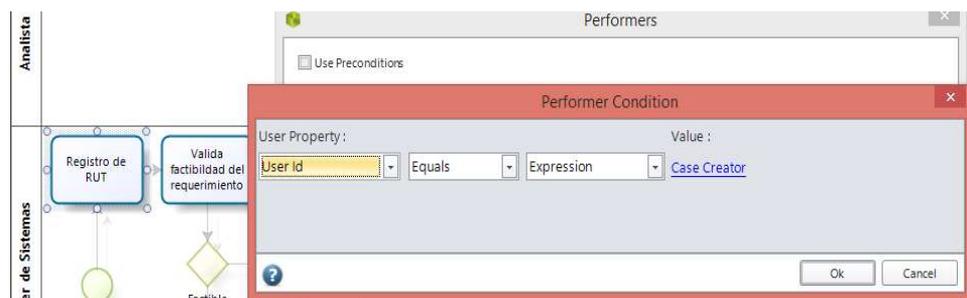


Figura 4.29: Participantes y roles

4.1.6. EJECUCIÓN DEL PROCESO

Luego de haber completado las fases anteriormente descritas, para la ejecución del flujo de procesos diseñados en Bizagi los usuarios necesitan tener acceso a Internet o Intranet

Para ejecutar se debe hacer clic sobre el icono que se resalta en la Figura 4.30:

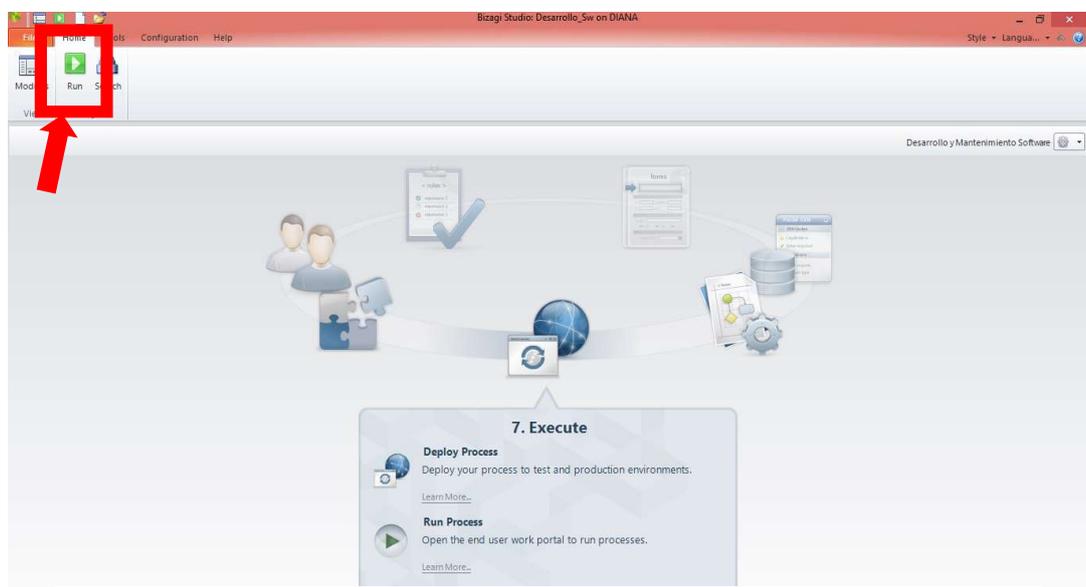


Figura 4.30: Ejecución de proceso

A continuación las principales pantallas presentadas durante la ejecución del proceso:

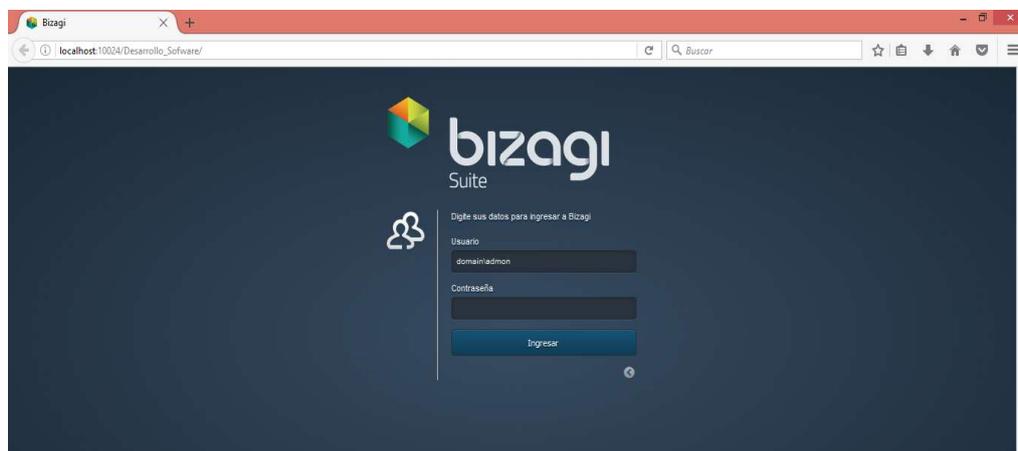


Figura 4.31: Ingreso a la ejecución del proceso

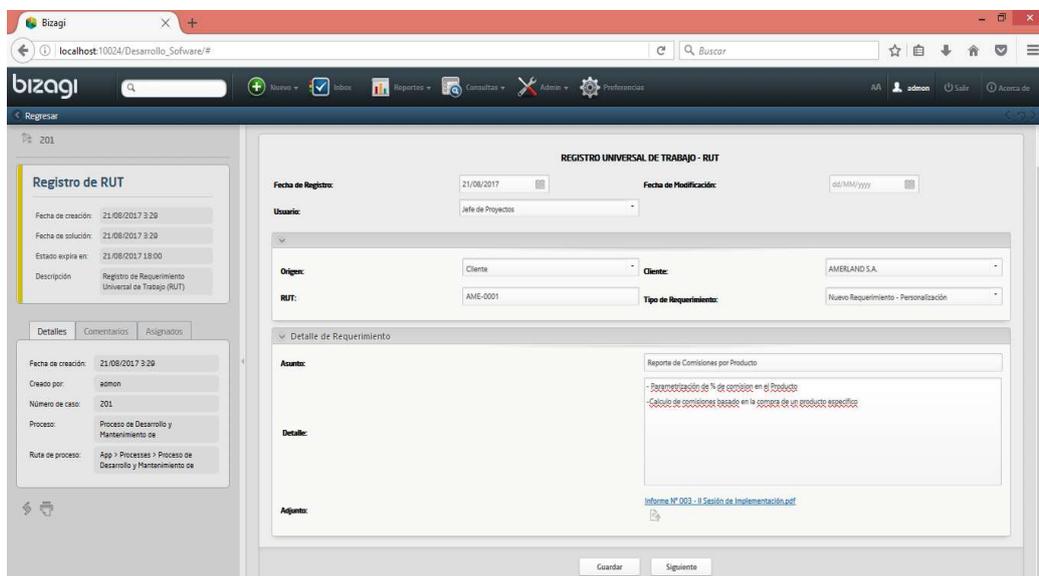


Figura 4.32: Registro de RUT

Informe de No Factibilidad de RUT

Fecha de Registro: 21/08/2017 Fecha de Modificación: dd/MM/yyyy
 Usuario: Jefe de Proyectos Fecha de Cliente: 21/08/2017

Datos Generales

Origen: Cliente Cliente: AMERLAND SA
 RUT: AME-0001 Tipo de Requerimiento: Nuevo Requerimiento - Personalización

Detalle de Requerimiento

Asunto: Reporte de Comisiones por Producto
 Detalle: - Parametrización de % de comisión en el Producto
 - Cálculo de comisiones basado en la compra de un producto específico
 Adjunto: [Informe N° 003 - II Sesión de Implementación.pdf](#)

Factibilidad

Factible: No Adjunto 2: No hay archivos
 Razón Rechazo: Faltan especificaciones. Aclarar idea

Figura 4.33: Informe de factibilidad

Rechazo de Orden – Terminación del Proceso

Usted no tiene más actividades pendientes en este caso

Figura 4.34: Rechazo de Orden – Terminación del Proceso

ASIGNACIÓN DE PRIORIDAD A RUT

Fecha de Registro: 21/08/2017 Fecha de Modificación: 21/08/2017

Usuario: Jefe de Proyectos

Origen: Interno Cliente: Defecto

RUT: INT-0002 Tipo de Requerimiento: Defecto

ASIGNACIÓN DE PRIORIDAD

Prioridad: [Dropdown menu]

Detalle de Requerimiento

Asunto: Cálculo de IESS

Detalle: Error en el redondeo de decimales en el cálculo del IESS

Adjunto: No hay archivos

Factibilidad

Factible: Si Adjunto 2: No hay archivos

Razón Rechazo:

Determina Prioridad

Fecha de creación: 20/08/2017 18:30

Fecha de solución: 20/08/2017 18:30

Estado expira en: 21/08/2017 12:00

Detalles Comentarios Asignados

Fecha de creación: 20/08/2017 18:30

Creado por: admon

Número de caso: 151

Proceso: Proceso de Desarrollo y Mantenimiento de

Ruta de proceso: App > Procesos > Proceso de Desarrollo y Mantenimiento de

Figura 4.35: Asignación de Prioridad

REGISTRO DE ORDEN DE TRABAJO

Asignación de recursos

Fecha Registro: 21/08/2017 RUT: INT-0001

OT: 1 Spies: Por favor seleccione...

Recurso: Andrea Valverde Función: Jefe de Proyectos

Guardar Siguiente

Asigna Recursos

Fecha de creación: 20/08/2017 18:30

Fecha de solución: 20/08/2017 18:30

Estado expira en: 21/08/2017 18:00

Detalles Comentarios Asignados

Fecha de creación: 20/08/2017 18:30

Creado por: admon

Número de caso: 151

Proceso: Proceso de Desarrollo y Mantenimiento de

Ruta de proceso: App > Procesos > Proceso de Desarrollo y Mantenimiento de

Figura 4.36: Asignación de Recursos

Informe de Reunión Sprint Inicial

Fecha de Registro: 20/08/2017 Fecha de Modificación: 20/08/2017

Usuario: Jefe de Proyectos

Origen: Interno Cliente: RAMIREZ BULLE EDINSON ARTURO

RUT: INT-0001 Tipo de Requerimiento: Defecto

Detalle de Requerimiento - Informe Sprint Inicial

Asunto: Model: Facturación - Cálculo IVA emisoro

Detalle: Pasa a cambio en parametro Nro los sigue calculando el 24%

Adjunto: IMP-00035.pdf

Priorización

Prioridad: Alta

Facilidad

Faciles: 9 Adjunto 2: No hay archivos

Estado Reclamos:

Guardar Siguiente

Figura 4.37: Registro de Informe de reunión de sprint inicial

Registro de Documentación

Fecha: 21/08/2017 RUT: 1

DT: 1 Tipo: Análisis Funcional

Recurso: Andrea Valverde Funcion: Jefe de Proyectos

Detalle

Descripción: Análisis Funcional Informe: No hay archivos

Guardar Siguiente

Figura 4.38: Registro de documentación

Valida Documento funcional

Fecha de creación: 20/08/2017 18:30
 Fecha de solución: 20/08/2017 18:30
 Estado expira en: 21/08/2017 18:00

APROBACIÓN DE DOCUMENTOS

Datos Generales

Fecha:	21/08/2017	RUT:	1
OT:	1	TIPO:	Análisis Funcional
Recurso:	Andrea Valverde	Función:	Jefe de Proyectos

Detalle

Descripción: Análisis Funcional Informe: No hay archivos

Aprobación

Confirmar: Si No

Razon Rechazo: Documentación Incompleta Informe 2: No hay archivos

Guardar Siguiente

Figura 4.39: Aprobación de Documentos

Bandeja de entrada

Todos los casos

1 Solicita pase a certificación
 Proceso: Proceso de Desarrollo y Mantenimiento de
 Fecha creación proceso: 20/08/2017 12:42
 Actividad vence en: 20/08/2017 15:15
 Fecha Solución: 20/08/2017 12:42

51 Asigna Recursos
 Proceso: Proceso de Desarrollo y Mantenimiento de
 Fecha creación proceso: 20/08/2017 15:48
 Actividad vence en: 20/08/2017 15:50
 Fecha Solución: 20/08/2017 15:48

151 Registro de RUT
 Proceso: Proceso de Desarrollo y Mantenimiento de
 Fecha creación proceso: 20/08/2017 18:30
 Actividad vence en: 21/08/2017 18:00

Solicita pase a certificación Trabajar aquí

Fecha de creación: 20/08/2017 12:42
 Estado expira en: 20/08/2017 15:15

Detalle / Asignados

Fecha de creación: 20/08/2017 12:42
 Creación por: admon
 Número de caso: 1
 Proceso: Proceso de Desarrollo y Mantenimiento de
 Ruta de proceso: App > Procesos > Proceso de Desarrollo y Mantenimiento de

Figura 4.40: Registro de Procesos Pendientes por el usuario

4.2. ANÁLISIS DE LA METODOLOGÍA ÁGIL SCRUM EN FUNCIÓN DEL PROCESO LEVANTADO

La organización ha decidido adoptar la metodología Scrum como apoyo al proceso de desarrollo y mantenimiento de software debido a que la forma como se manejaba anteriormente el proceso no era efectiva. La organización entiende que se enfrenta a nuevos desafíos propios del cambio, finalmente lo que se desea es poder decidir si la metodología Scrum es la mejor opción.

4.2.1. RELACIÓN DE LAS ACTIVIDADES A REALIZAR DURANTE EL PROCESO DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE CON SCRUM

En la Tabla 9 se puede apreciar en base al diagrama BPMN levantado con las actividades que forman parte del proceso de desarrollo y mantenimiento de software como estas se relacionan con el ciclo de desarrollo de software y las etapas de Scrum.

Tabla 9: Actividades a realizar durante el ciclo de desarrollo de software relacionadas con Scrum

Ciclo de desarrollo	Rol	Tareas	Ciclo Scrum	Rol Scrum
Especificaciones	Líder de sistemas	Registro de Rut. Se registra y establece: 1. Origen del requerimiento: * Interno * Cliente 2. Clasificación del requerimiento según los siguientes criterios: * Nuevas funcionalidades * Mantenimiento – Defectos	Fase de Planificación	Dueño del Producto
		Valida factibilidad del requerimiento. En el caso de ser factible el proceso continúa normalmente. En caso que NO resulte factible el proceso termina.		
		Determina Prioridad Recursos. Según la urgencia e importancia del requerimiento.		
		Asignación de Recurso. *Se evalúa la carga de trabajo y en función de la complejidad y la prioridad del requerimiento.		
		Dirige reunión de planificación de Sprint.		
Análisis y Diseño	Analista funcional	Elabora Documento Funcional. Se desarrolla un plan preliminar de diseño, implementación y pruebas.	Fase de Planificación	Equipo de desarrollo Scrum
	Arquitecto de software	Valida documento funcional. En el caso que NO este conforme con el documento funcional entregado: Elabora y envía Observaciones al analista funcional.		
		Elabora Documento Técnico. Determinar objetos a modificar (Front-End, Back- end), restricciones, riesgos y liberación del software.		
		Elabora casos de prueba funcionales en base a todos los escenarios posibles. Se debe involucrar al usuario.		
	Analista funcional	Crea repositorio de control de versión.		
	Arquitecto de software	Elabora casos de pruebas técnicas.		

Ciclo de desarrollo	Rol	Tareas	Ciclo Scrum	Rol Scrum
Desarrollo	Analista desarrollador	Habilita ambiente de desarrollo.	Fase de Desarrollo	Equipo de desarrollo Scrum
		Desarrolla programas en base a las especificaciones funcionales y técnicas.		
		Realiza pruebas unitarias. En el caso de que estas pruebas NO resulten exitosas, regresa al paso de desarrollo de programas.		
		Integra, empaqueta solución.		
		Realiza pruebas integrales. Integra el software modificado con el sistema existente. En el caso de que estas pruebas NO resulten exitosas, regresa al paso de desarrollo de programas.		
	Congela versión programa fuentes.			
	Solicita pase a certificación.			
	Analista de control de calidad	Valida solicitud de pase. En caso de NO estar conforme devuelve observaciones a desarrollo.		
	Analista funcional	Prepara el ambiente y ejecuta pase para certificación. En caso de NO estar conforme Notifica para revertir el pase.		
Pruebas	Analista de control de calidad	Ejecuta pruebas de certificación.	Fase de Revisión.	
		Evalúa Resultados de las pruebas.		
		Ejecuta pruebas de certificación con Usuarios.		
		Elabora Informe de control de calidad , se adjuntan pruebas de todos los escenarios probados. En el caso de que las pruebas de calidad de NO resulten exitosas, regresa al paso de desarrollo de programas.		
		Solicita pase a producción.		
Liberación del Producto	Analista funcional	Actualiza manuales de usuarios.	Fase de Finalización	Scrum Master
		Entrega Versión.		
	Líder de sistemas y Jefe de Área Operativa	Ejecuta el pase en producción.		
		Seguimiento y control de la actualización.		
		Dirige reunión de retrospectiva.		
		Elabora Informe de cierre.		

4.2.2. PASOS A SEGUIR PARA IMPLEMENTAR SCRUM EN EL PROCESO DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE

A continuación se establecen los pasos que seguirá la organización para hacer posible la implementación de la metodología ágil Scrum.

1. Definir un responsable de producto (Dueño del Producto):

Persona que conoce a detalle el producto y tiene la visión clara de lo que se necesita, tendrá en cuenta los riesgos, y la viabilidad del proyecto. Para este caso la persona idónea para esta asignación es el líder del área operativa.

2. Definir el equipo de desarrollo (Scrum Team):

Se requiere que el equipo tenga las habilidades necesarias para hacer realidad la visión del responsable de producto.

3. Identificar el Scrum Master:

Persona responsable de guiar a todo el equipo en la aplicación de Scrum y de eliminar del camino todo aquello que los detenga. Se establecerá como Scrum Master al líder de desarrollo.

4. Elaborar y priorizar la lista de objetivos (backlog):

Tener en cuenta todo lo que se debe realizar para cumplir con la visión del producto. Para elaborarla se deberá consultar con todos los

interesados para cerciorarse de que representa lo que quiere el cliente y además lo que es factible.

- 5. Planificar los Sprints:** Cada sprint durará máximo 4 semanas, el equipo revisará el backlog y realizará una proyección de cuánto pueden realizar en cada sprint. Una vez definido no se puede cambiar ni añadir nada. El equipo será capaz de trabajar de forma autónoma a lo largo del sprint.
- 6. Mostrar en un lugar visible el tablero de tareas (Scrum Taskboard)** con sus tres columnas: pendiente, en proceso y hecho.
- 7. Scrum Diario:** todas las mañanas, antes de arrancar con cualquier actividad, se realizará una reunión de no más de 15 minutos, donde se evaluará el avance y posibles riesgos que puedan presentarse.
- 8. Revisión y demostración del Sprint:** El equipo realizará una reunión abierta en la que se explicará lo que ha logrado pasar a la columna de «hecho» durante el sprint.
- 9. Retrospectiva del Sprint:** Posterior a la entrega de lo conseguido en el sprint se efectuará una reunión donde se reflexionará sobre lo que ha ido bien, lo que podría hacerse mejor y lo que se podría perfeccionar en el siguiente sprint.

CAPITULO 5

IMPLEMENTACIÓN DE LA PROPUESTA

5.1. IMPLEMENTACIÓN DE PLAN DE PRUEBAS

Contar con un plan de pruebas definido permite detallar lo que se quiere probar y cómo se van a ejecutar las pruebas. Una de las debilidades que actualmente tiene la empresa objeto de estudio es la calidad del producto / servicio que ofrece a sus clientes, ya que muchos de los requerimientos registrados son por defectos en producción. Por esta razón, se implementará el uso de una matriz de pruebas con el fin de mejorar la calidad del producto.

El plan de pruebas se aplicará a todos los requerimientos ingresados a desarrollo.

En las Figuras 4.41 y 4.42 se presentan los formatos de la matriz de casos de pruebas técnicas y funcionales a utilizar.

Sprint _____ Proyecto: _____ Metodo Utilizados _____
 Fecha de Inicio _____ Responsable: _____ Componentes _____
 Fecha de Final _____ Objetivo: _____

Escenario				Participantes:			Tipo de Prueba		Ambiente		
ID	Fecha	Módulo	Casos	Camino	Especifica SP	Genera Log	Nombre Sp	Parámetros	Resultados esperados	Resultados Obt	Observaciones
<Id. del caso>	<Fecha de prueba>	<Módulo afectado>	<Descripcion del escenario>	<Condiciones previas antes de llegar al caso>	<Condiciones generadas por el caso>	<Descripción del flujo normal>	<Casos alternos que pueda tener el flujo>	<Excepciones del flujo Normal>	<Resultados esperados en la pruebas>	<Resultados obtenidos en las pruebas>	<Otros datos que aporten a la implementación>

Figura 4.41: Matriz de pruebas técnicas aplicadas durante

Sprint _____ Proyecto: _____
 Fecha de Inicio _____ Responsable: _____
 Fecha de Final _____ Objetivo: _____

Escenario				Participantes:			Tipo de Prueba		Ambiente		
ID	Fecha	Módulo	Sub-Escenario	Pre condición	Post condicio	Flujo Normal	Flujos Alternos	Excepciones	Resultados esperados	Resultados Obt	Observaciones
<Id. del caso>	<Fecha de prueba>	<Módulo afectado>	<Descripcion del escenario>	<Condiciones previas antes de llegar al caso>	<Condiciones generadas por el caso>	<Descripción del flujo normal>	<Casos alternos que pueda tener el flujo>	<Excepciones del flujo Normal>	<Resultados esperados en la pruebas>	<Resultados obtenidos en las pruebas>	<Otros datos que aporten a la implementación>

Figura 4.42: Matriz de pruebas funcionales aplicadas durante cada Sprint

5.2. IMPLEMENTACIÓN DE LAS POLÍTICAS Y PROCEDIMIENTOS

Establecer políticas dentro de la organización permitirá a los empleados saber cómo manejar situaciones específicas y procedimientos sobre cómo llevar a cabo determinadas tareas. A continuación se detallan las políticas y procedimientos a implementar:

5.2.1. POLÍTICAS REFERENTES AL DESARROLLO DE SOFTWARE

POLÍTICA PARA LA RECEPCIÓN DE REQUERIMIENTOS DE DESARROLLO

- Todo requerimiento que implique modificación a las aplicaciones o sistemas existentes, así como nuevos desarrollos deben ser ingresados mediante una Orden de Trabajo (OT) clasificados por solicitud de cambios, mantenimiento de las aplicaciones o nuevos requerimientos de desarrollo.
- Las OT ingresadas por solicitud de cambios o nuevos requerimientos de desarrollo, deben tener la autorización del Jefe del área Operativa, y haber sido revisadas por los comités técnico y de calidad, y en el caso de que sea un requerimiento que tendrá costo para el cliente deberá tener

su aceptación, antes de realizar el análisis o cualquier diseño inicial.

POLÍTICA PARA EL ANÁLISIS DE REQUERIMIENTOS

- Todo requerimiento autorizado conlleva a un análisis del mismo, su objetivo es establecer especificaciones que detallen las necesidades a ser cubiertas.
- En la definición de nuevos requerimientos deben participar tanto los recursos de la empresa involucrados (líder, desarrollador, analista funcional) como los usuarios solicitantes.
- En el caso de nuevos requerimientos se elaborará un plan para reuniones con el usuario, en donde por cada reunión se incluirá fecha, hora y lugar, participantes y un apartado de los aspectos relevantes se trataron en la reunión. (Funciones que realizan el usuario y problemas que necesita resolver).
- Luego de presentar las especificaciones del requerimiento se deben definir las diferentes alternativas de desarrollo con sus ventajas y desventajas, estas deberán ser documentadas, para posteriormente elegir la más conveniente.

POLÍTICA PARA LA ASIGNACIÓN DE RECURSOS HUMANOS

- El jefe del área de desarrollo será el encargado de la asignación de recursos para un requerimiento, y revisará la carga de trabajo y la capacidad de cada recurso, así como también el nivel de prioridad de la OT.

POLÍTICA DE DESARROLLO DE APLICACIONES

- El desarrollo de aplicaciones de software debe estar dirigido a satisfacer las necesidades de manejo de información.
- Los nuevos proyectos de desarrollo de sistemas de información, tanto los de origen interno como los de origen externo, deberán ser debidamente formalizados y gestionados siguiendo la metodología y estándares establecidos en el respectivo manual de desarrollo de sistemas.
- El Comité Técnico y el Comité de Calidad garantizarán que el proceso de desarrollo sea llevado de la manera más óptima.

- Todo desarrollo de aplicaciones de software, tanto de origen interno como de origen externo, deberá ser evaluado por el comité técnico de la organización.
- El proceso de desarrollo de aplicaciones de software debe contemplar las etapas de análisis de requerimientos, diseño, desarrollo de la programación, pruebas de calidad, puesta en producción e implementación.
- El proceso de desarrollo y/o mantenimiento de software debe de contar con estándares de programación, versionamiento, documentación y pruebas.
- El proceso de desarrollo y/o mantenimiento de software debe contener procedimientos de pruebas funcionales y no funcionales. Las pruebas no funcionales deben incluir pruebas de vulnerabilidades y seguridades.
- El proceso de desarrollo y/o mantenimiento de software, así como las pruebas de certificación deben manejarse en ambientes independientes dispuestos para ello.
- El Cliente/Usuario solicitante de cualquier requerimiento de nuevos desarrollos deberá participar activamente durante la etapa de análisis, desarrollo e implementación.

POLÍTICAS PARA ASEGURAMIENTO DE LA CALIDAD

- El plan de pruebas para aceptación del requerimiento debe ser coherente con los requisitos y las especificaciones funcionales.
- Las pruebas a realizarse deben ser de los siguientes tipos:
 - Pruebas unitarias ejecutadas en ambiente de desarrollo, por el desarrollador del requerimiento a fin de certificar su funcionalidad en forma aislada.
 - Pruebas conjuntas ejecutadas en ambiente de desarrollo, por todos los desarrolladores. Su objetivo es validar la funcionalidad del sistema completo.
 - Pruebas funcionales, ejecutadas al liberar cambios en el ambiente de pruebas, realizadas por el analista funcional junto a los usuarios solicitantes del requerimiento. Estas pruebas de aceptación del requerimiento deberán efectuarse antes de su liberación al ambiente de producción.
- El Comité de Calidad será el responsable de velar por el cumplimiento de los estándares establecidos para todo el ciclo de vida de desarrollo del proyecto.

***POLÍTICAS PARA EL MANEJO DE ESTÁNDARES DE
DESARROLLO Y CONTROL DE LA DOCUMENTACIÓN***

- El comité técnico se apegará a una metodología estándar para el análisis y desarrollo de los requerimientos, a fin de definir los aspectos más importantes que deberán ser tomados en cuenta durante el ciclo de vida del desarrollo del requerimiento.
- El Comité Técnico es el responsable de dar a conocer y vigilar la correcta aplicación de la metodología seleccionada para el desarrollo de sistemas (SCRUM), por todas las personas involucradas en el área de desarrollo.
- Toda la documentación generada, incluyendo la documentación técnica (análisis, diseño, documentación de los programas, manuales de usuario), debe cumplir con el estándar general.

POLÍTICAS DE CONTROL DE ACCESO A LOS CÓDIGOS

FUENTE

- Se debe proporcionar al Área de Desarrollo los programas fuentes solicitados para su modificación, teniendo en cuenta la correlación programa fuente /ejecutable.
- Se debe llevar un registro actualizado de todos los programas fuente en uso, el cual debe tener:
 - Nombre del programa.
 - Versión.
 - Fecha/hora de última modificación.
 - Fecha / hora de compilación.
 - Estado (en modificación, en producción).
- Gestionar las distintas versiones de la aplicación.
- Todo programa objeto en producción debe tener un único programa fuente asociado que avale su origen.
 - Restringir el acceso sin control que permita la generación y/o manipulación de los programas fuente.
 - Realizar las copias de seguridad y pruebas de restauración de los programas fuente.

POLÍTICA PARA CONTROL DE CAMBIOS EN DESARROLLO

- El área de desarrollo es responsable de elaborar, publicar y vigilar la correcta aplicación del procedimiento de control de cambios.

5.2.2. PROCEDIMIENTOS

PROCEDIMIENTOS PARA ANALIZAR Y DOCUMENTAR REQUERIMIENTOS DE DESARROLLO DE SOFTWARE

- Se recibe la solicitud de desarrollo y/o mantenimiento de software registrado en el sistema de Gestión de Órdenes de Trabajo. Se analiza factibilidad.
- Cuando un requerimiento no es viable se realiza el respectivo informe que se da a conocer a los usuarios interesados.
- Se establecen prioridades, se asignan los recursos que van a estar involucrados en el proyecto de desarrollo.
- En el caso de ser nuevos desarrollos de software, se coordinan reuniones de trabajo en donde participarán todos los interesados (analistas, líder, cliente / usuario final, soporte de usuario, etc.) en donde se detallarán los requerimientos,

funciones específicas que deberá tener incluido el sistema. El número de reuniones que se deben mantener dependerá de la complejidad del requerimiento.

- Elaboración del documento funcional en base a lo recopilado en las reuniones con los interesados. Para la elaboración de los documentos e informes se deberá seguir el estándar definido por la organización.
- Determinar en un documento técnico el plan que seguirá el equipo de desarrollo. Se establecerá el diseño de estructura de programación (lenguaje de programación, interfaz, etc) y de base de datos (definición de diccionario de datos).
- Una vez entregado el producto final deberán también entregarse actualizados los manuales de usuarios y de sistema.

PROCEDIMIENTO DE DESARROLLO Y MANTENIMIENTO DE SISTEMAS

- Recibe y analiza la documentación funcional y técnica.
- De acuerdo al esquema planteado inicia la programación (creación del código fuente).
- Realizar pruebas en coordinación con el usuario solicitante.

- Documentar los resultados de las pruebas unitarias, así como el desarrollo del sistema.
- Documentar los resultados de pruebas de integración (comportamientos de datos, datos utilizados que interactúan entre los módulos, desempeño de la interfaz, etc.).
- Realizar dos respaldos de los programas modificados. (código fuente, estructura de tablas, procedimientos y funciones.) Un respaldo será entregado al área de certificación y el otro será entregado al líder del área para su posterior liberación en producción.
- Presentar el sistema terminado, realizar la solicitud para las respectivas pruebas de certificación.

PROCEDIMIENTO DE CONTROL DE CAMBIOS

Con el objetivo de reducir al máximo los riesgos de alteración de los sistemas de información, se efectuarán controles estrictos durante la ejecución de cambios. Entre los controles a realizar están:

- Validar que los cambios fueron propuestos por usuarios autorizados y que se respete los procesos.

- Revisar los controles y los procedimientos de integridad para garantizar que no serán comprometidos por los cambios.
- Solicitar la revisión del comité técnico y el comité de calidad a fin de garantizar que se cumplan con los requerimientos de seguridad que debe cumplir el software.
- Mantener un control de versiones para todas las actualizaciones de software.

5.3. IMPLEMENTACIÓN DE INDICADORES EN EL PROCESO DE GESTIÓN DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE

Debido a que en la empresa objeto de estudio no existe control sobre los proyectos de desarrollo de software llevados a cabo, para mejorar tal situación se implementarán métricas e indicadores en el proceso de Gestión de Desarrollo de Software. En la Tabla 10 se presentan los objetivos, indicadores y métricas a implementar.

Tabla 10: Objetivos, indicadores y métricas

1º Objetivo		
Incrementar el nivel de cumplimiento en las fechas de entrega de los proyectos de desarrollo de software.		
Indicadores	Métricas	Rangos
Porcentaje de requerimientos entregados dentro del tiempo establecido.	Número de Actividades cerradas en tiempo / Número de Actividades del Proyecto	Bueno: $\geq 90\%$ Regular: $\geq 75\%$ y $< 90\%$ Malo: $< 75\%$

2° Objetivo		
Mejorar la precisión de las estimaciones de las actividades de los proyectos.		
Indicadores	Métricas	Rangos
Porcentaje de actividades cerradas dentro del tiempo planificado.	Número de Actividades cerradas en tiempo / # Actividades del Proyecto	Bueno: $\geq 80\%$ Regular: $\geq 70\%$ y $< 80\%$ Malo: $< 70\%$
3° Objetivo		
Incrementar el nivel de involucramiento de los usuarios durante la construcción.		
Indicadores	Métricas	Rangos
Porcentaje de requerimientos en los que el usuario ha estado involucrado dentro del plazo establecido.	Número de requerimientos validados con el usuario dentro del plazo / Número de requerimientos totales del proyecto	Bueno: $\geq 90\%$ Regular: $\geq 75\%$ y $< 90\%$ Malo: $< 75\%$
4° Objetivo		
Minimizar el número de defectos presentados en etapa de producción.		
Indicadores	Métricas	Rangos
Número de ciclos de la etapa de certificación.	Número de ciclos en la etapa de certificación	Muy bueno = 1 Bueno: = 2 Regular: = 3 Malo: ≥ 4
5° Objetivo		
Mejorar el proceso de gestión de requerimientos.		
Indicadores	Métricas	Rangos
Número de cambios aprobados por nuevas o cambios de definiciones.	Números de cambios	Bueno: ≤ 3 Regular: ≥ 4 y ≤ 6 Malo: ≥ 7
6° Objetivo		

Mejorar el control de toda la documentación asociada a los proyectos.		
Indicadores	Métricas	Rangos
Número de no conformidades u observaciones de auditoría (internas o externas).	Número de no conformidades + observaciones de auditoría	Bueno: ≤ 3 Regular: > 3 y ≤ 5 Malo: ≥ 6

CAPÍTULO 6

ANÁLISIS DE RESULTADOS

6.1. ANÁLISIS DE LOS RESULTADOS DE LA PROPUESTA IMPLEMENTADOS EN UN PROYECTO EN MARCHA

Para el presente proyecto de titulación se realizó la implementación de la propuesta de Gestión de Procesos de Negocio y Metodología Scrum, aplicando también las políticas y procedimientos descritos en el capítulo anterior, sobre un nuevo requerimiento que ingresó al área operativa.

6.1.1 GESTIÓN DE REGISTRO DEL REQUERIMIENTO

El requerimiento sobre el cual se aplicó la propuesta provino de un cliente. El cliente a través de correo electrónico solicitó agregar una nueva funcionalidad a la opción de Proformas dentro del Módulo de Facturación. En la Figura 6.1 se presenta la solicitud enviada por el cliente a través de correo electrónico.

Cliente	XYZ S.A.
Fecha de Solicitud	08/08/2017
Tipo de requerimiento	Nueva Funcionalidad
Módulo	Facturación
Opción	Proformas y Pedidos
Descripción	
1.- Al momento de ingresar productos compuestos en una proforma cargar los componentes de este. 2.- Permitir Cambiar, eliminar o ingresar nuevos componentes, sin afectar a la configuración inicial del componente 3.- Imprimir la Proforma mostrando el producto principal y los compuestos	

Figura 6.1: Solicitud enviada por el Cliente

Esta solicitud previamente fue gestionada por el departamento comercial en conjunto con el jefe de operaciones para establecer el costo del mismo. Al ser un requerimiento solicitado por un cliente tiene mayor prioridad que los requerimientos internos.

6.1.2. GESTIÓN DE PLANIFICACIÓN

Como primer paso el jefe del área de operaciones realiza el registro del requerimiento en la plataforma de Bizagi.

El líder de sistemas inmediatamente realiza la planificación inicial del trabajo. A continuación se detallan los puntos tratados en esta planificación:

- Se define el Sprint en el que se realizará el requerimiento en base a la prioridad, en este caso como estamos iniciando con el piloto el código de sprint corresponde al primero.
- Define el equipo Scrum, es decir se asignan los recursos con sus respectivas Órdenes de Trabajo.

En la Figura 6.2 se presenta el resumen de la planificación inicial realizada por el líder de sistemas.

Asignación de Recursos Generación de Ordenes de Trabajo		
RUT	XYZ-00055	
Sprint	000001	
Fecha de Inicio	14/08/2017	
Fecha de Fin	25/08/2017	
Módulo	Facturación	
Opción	Proformas y Pedidos	
Tipo de requerimiento	Nueva Funcionalidad	
Prioridad	Alta	
ROL	RECURSO	OT
Líder de sistemas	Ing. Carlos Gomez	XYZ-00055-001
Analista funcional	Ing. Luris Pérez	XYZ-00055-002
Analista desarrollador	Ing. Jackson Gamboa	XYZ-00055-003
	Ing. Geovany Giron	XYZ-00055-004
Arquitecto de software	Ing. David Maldonado	XYZ-00055-005
Analista de control de calidad	Ing. Jonathan Díaz	XYZ-00055-006
Usuario Certificador (Cliente)	Jefe de Facturación	

Figura 6.2: Planificación Inicial: Asignación de recursos y órdenes de trabajo

Posterior a la planificación inicial se realizó la primera reunión de planificación del Sprint en donde asistieron todos los involucrados en el proyecto, incluyendo el Cliente representado por el Jefe de Facturación quien es uno de los beneficiarios del requerimiento y conoce el giro del negocio. La Tabla 11 presenta la manera como quedó conformado el equipo SCRUM.

Tabla 11: Conformación del equipo SCRUM

Dueño de Producto	SCRUM Master	Equipo de Desarrollo SCRUM
Líder de Sistema Cliente (Jefe de Facturación)	Líder de Sistema	Analista desarrollador Analista funcional Arquitecto de software Analista de control de calidad

Como resultado de esta reunión se generó un acta donde se establecieron los siguientes puntos:

- Se definieron los objetivos del requerimiento.
- Se definieron los requisitos del requerimiento, el equipo se comprometió a entregar la funcionalidad en base a estas definiciones.
- Se fijaron tiempos y fechas de entrega, para poder realizar un monitoreo constante de avances y resultados.

En el Anexo 2 podemos revisar el acta de acuerdos generada en esta reunión.

Posterior a la reunión de inicio del sprint el analista funcional con apoyo del Cliente/Usuario elaboró el documento funcional en donde se especificó a detalle el requerimiento a desarrollar. En el Anexo 3 podemos revisar el Documento Funcional generado.

El arquitecto de software validó el Documento Funcional y preparó el documento con las especificaciones técnicas del requerimiento a nivel de Front end y Back end, los cuales fueron entregados al analista desarrollador. En el Anexo 4 podemos revisar el Documento Técnico generado.

Una vez que se validaron los documentos funcional y el técnico, el analista funcional junto con el arquitecto de software procedieron con la elaboración de los casos de pruebas funcionales y técnicas respectivamente. Para la elaboración de los casos de pruebas se contó con el apoyo del cliente. En el Anexo 5 podemos revisar el documento con los casos de pruebas funcionales y técnicas definidas.

La fase de planificación finalizó con la creación del repositorio para control de versión, a cargo del analista funcional.

6.1.3. GESTIÓN DE DESARROLLO

Con la documentación entregada en la fase de planificación el analista desarrollador procedió con la programación del requerimiento. Realizó las respectivas pruebas unitarias y de integración.

En este punto es donde diariamente el equipo Scrum se reúne 15 minutos para tratar los avances obtenidos, posibles problemas que enfrenta y como solucionarlos.

Una vez concluido con éxito el desarrollo del requerimiento se procedió a congelar la versión del programa fuente, además se elaboró el respectivo informe, y se solicitó el pase a certificación.

El analista de control de calidad se encargó de realizar la respectiva validación de la solicitud del pase a producción, es decir revisó que la documentación de los cambios realizados sobre la aplicación esté completa y correcta.

Una vez conforme con la solicitud del pase a certificación, este fue entregado al analista funcional, este se encargó de preparar y ejecutar el pase en ambiente de certificación.

6.1.4. GESTIÓN DE PRUEBAS

El analista de control de calidad procedió a ejecutar los casos de pruebas funcionales; inicialmente las realizó sólo, pero como este requerimiento tenía por origen un cliente, posteriormente fue necesario involucrarlo en una segunda fase de pruebas.

Para finalizar esta etapa el analista de control de calidad procedió a elaborar el respectivo Informe de Calidad, y la solicitud del pase a Producción.

6.1.5. GESTIÓN DE LIBERACIÓN DE PRODUCTO

Luego de finalizadas las pruebas y entregadas las evidencias de estas, el analista funcional procedió con la entrega de la versión con los programas actualizados. En esta fase el analista funcional también se encargó de actualizar los manuales de

usuario, agregando el uso de la nueva funcionalidad desarrollada.

Una vez aprobados todos los cambios el líder de sistemas ejecutó el pase en producción, realizó el respectivo seguimiento y control de la actualización ya en producción verificando que todo esté en orden.

Posteriormente el líder de sistemas fue el encargado de organizar la reunión final de Sprint donde se realizó la revisión y presentación del Sprint y la retrospectiva en donde se revisaron los puntos a mejorar y lecciones aprendidas a lo largo del sprint.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Después de analizar, diseñar e implementar una propuesta de gestión de procesos de negocio junto con la metodología ágil SCRUM en un proyecto en marcha de un proceso de desarrollo y mantenimiento de software, en una empresa privada, que ofrece soluciones tecnológicas, obtenemos las siguientes conclusiones:

1. Para la empresa fue trascendental la implementación de Gestión de Procesos de Negocio debido a que permitió tener claro el panorama de todo el flujo del proceso llevado a cabo durante el desarrollo y mantenimiento de software; y por consiguiente se obtuvo mayor control de todas las actividades realizadas en el mismo, para de esta manera poder plantear mejoras al proceso de una manera simple.

2. El modelado del proceso de desarrollo y mantenimiento de software a través de BPMN permitió a los involucrados comprender de mejor manera las actividades realizadas durante el proceso, así como definir los responsables de llevarlas a cabo.
3. La implementación de un plan de pruebas bien definido durante el proceso de desarrollo y mantenimiento de software se convirtió en un mecanismo que marca la diferencia en el resultado final, entregando un producto de software con mejor calidad, esto nos ayuda a aumentar la satisfacción general del cliente, disminuir costos y optimizar recursos.
4. La definición e implementación de indicadores en el proceso de gestión de desarrollo y mantenimiento de software permiten conocer el estado del proceso, ofreciendo información útil que permitirá posteriormente una mejor toma de decisiones en base a los resultados obtenidos.
5. La aplicación y estandarización de una metodología bien definida para el manejo de todos los proyectos de desarrollo y mantenimiento de software, en este caso SCRUM, están relacionados con el uso eficiente y eficaz de los recursos destinados para el proyecto, y mejoras en los servicios que se le ofrecen a los clientes ya que estos en todo momento se sienten tomados en cuenta.
6. Durante las etapas de desarrollo del requerimiento se mantuvo comunicación constante con el usuario lo que permitió una definición clara del requerimiento, por lo tanto las pruebas realizadas ayudaron a

obtener un producto final acorde a los estándares propuestos por el usuario.

7. Para el proyecto en marcha elegido en el que se implementó las metodologías SCRUM y BPM pese a que el tiempo de duración del proceso desde que se registró el requerimiento hasta que salió a producción aumentó en tres días, en comparación de que no se lo hubiese aplicado el resultado fue favorable debido a:

1. No se presentaron defectos en producción, gracias a que las pruebas fueron más y mejores detalladas. Esto evitó utilizar tiempos adicionales en encontrar y corregir estos defectos, ya que se hubiese tenido que volver a registrar un requerimiento y realizar todo un proceso para corregir estos defectos.
2. Cliente / Usuario satisfecho puesto que se pudo cumplir con sus expectativas, al encontrarse involucrado pudo conocer el proceso y comprender los tiempos que se necesitó para la entrega.
3. El equipo de trabajo se encuentra más motivado e involucrado en el proceso.

Recomendaciones

Luego de haber implementado la propuesta de Gestión de Procesos de Negocio y la Metodología ágil SCRUM, en un requerimiento puesto en marcha en la empresa, se recomienda:

1. Aplicar la presente propuesta para los demás requerimientos venideros, buscando siempre obtener retroalimentación de todos los involucrados durante el proceso, con el objetivo de ir mejorando continuamente la manera como se los llevan a cabo.
2. Revisar periódicamente (cada semestre) la estrategia de la organización, sus políticas, procedimientos e indicadores.
3. Trabajar siempre en las oportunidades de mejoras que se puedan presentar.
4. Realizar el estudio y análisis de los tiempos del proceso, de tal manera que se logre la estandarización y optimización de los mismos.

BIBLIOGRAFÍA

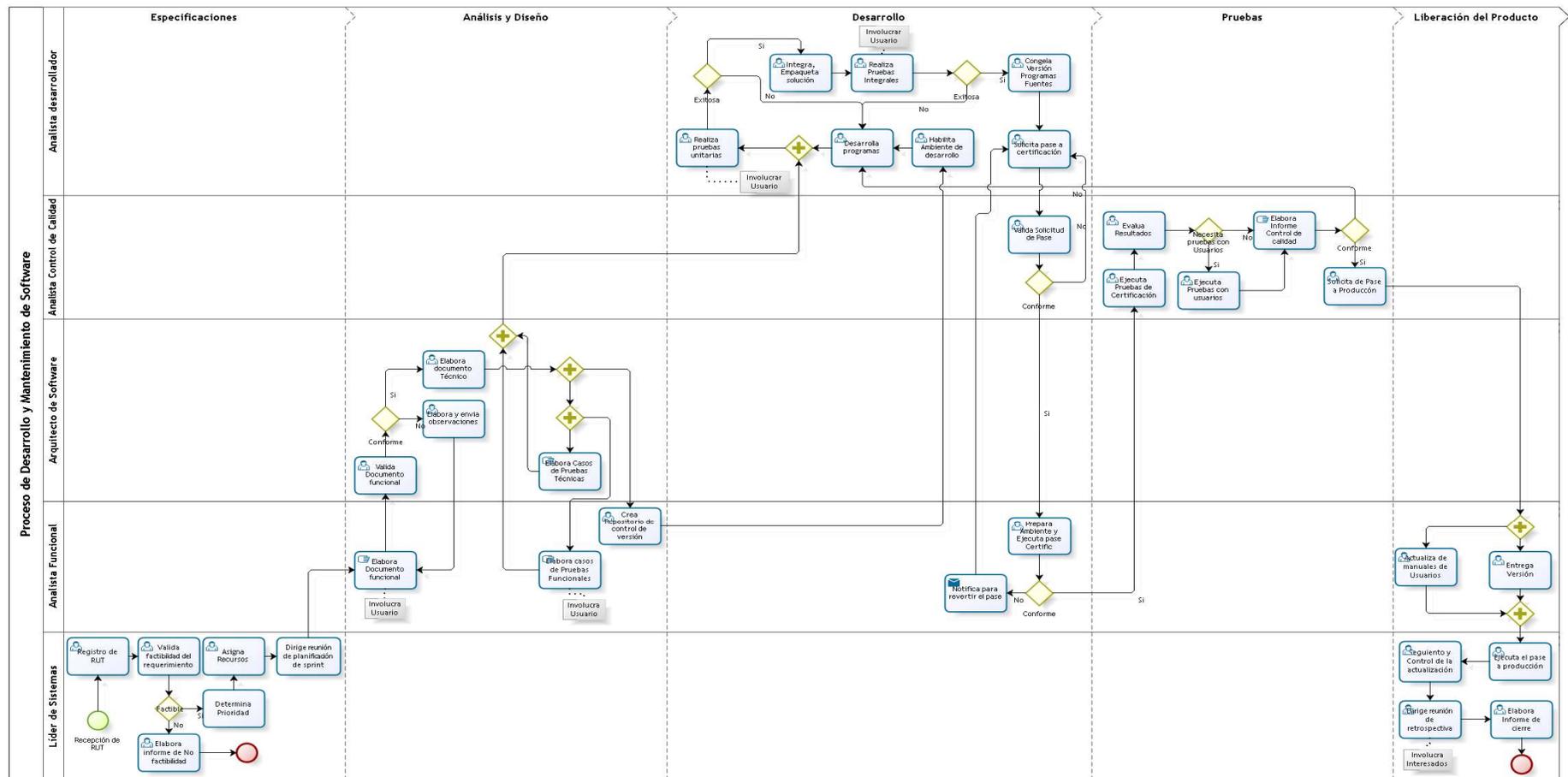
- [1] A. León y V. Cristina, «Diseño de una metodología para el direccionamiento de proyectos de software para la empresa Gestorinc SA», 2014.
- [2] V. M. Magliano, P. Bazán, y J. N. Martínez Garro, «Análisis metodológico para la utilización de Process Mining como tecnología de optimización y respaldo de la implementación de procesos de negocio bajo el marco de BPM», en *XV Workshop de Investigadores en Ciencias de la Computación*, 2013.
- [3] F. García y C. David, «Propuesta de rediseño de procesos de negocio de la Compañía de Tecnologías de la Información y la Comunicación Telefónica Colombia-Movistar: proceso de gestión de cambios tecnológicos», 2015.
- [4] A. Delgado y D. Calegari, «Business Process Management (BPM): aspectos clave para la construcción de software de soporte e impacto en la mejora continua de las organizaciones», *INNOTEC Gest.*, n.º 6, pp. 40–51, 2014.
- [5] E. T. Cano, «Optimización y cuantificación de procesos utilizando BPM- Process Optimization and Quantification using BPM», *Apunt. Univ.*, n.º 1, 2013.

- [6] R. Giandini, G. Pérez, y C. Pons, «Un lenguaje de Transformación específico para Modelos de Proceso del Negocio», en *XXXVI Conferencia Latinoamericana de Informática (CLEI 2010)*, 2010, vol. 18.
- [7] J. F. G. Estupiñan, «Una Mirada a BPMN como Herramienta Estándar para el Modelado de Procesos de Negocio», *Fray Ismael Leonardo Ballesteros Guerr. OP–Decano Div. Arquít. E Ing. Univ. St. Tomás Secc. Tunja*.
- [8] J. Mendling, H. A. Reijers, y W. M. van der Aalst, «Seven process modeling guidelines (7PMG)», *Inf. Softw. Technol.*, vol. 52, n.º 2, pp. 127–136, 2010.
- [9] E. Espino Berrocal, R. Tito Alegre, y others, «Propuesta de mejora en la gestión de proyectos de desarrollo de software», Universidad Peruana de Ciencias Aplicadas-UPC, 2014.
- [10] O. T. Gómez, P. P. R. López, y J. S. Bacalla, «Criterios de selección de metodologías de desarrollo de software», *Ind. Data*, vol. 13, n.º 2, pp. 70–74, 2010.
- [11] Ph.D. Mónica Villavicencio, «Modelos de procesos para el desarrollo del software», presentado en MSIG, Ingeniería en Software, Escuela Superior Politécnica del litoral.
- [12] A. Qumer y B. Henderson-Sellers, «An evaluation of the degree of agility in six agile methods and its applicability for method engineering», *Inf. Softw. Technol.*, vol. 50, n.º 4, pp. 280–295, 2008.
- [13] M. T. Gallego, «Metodología Scrum», *Gest. Proy. Informáticos*
Httpopenaccess *Uoc*

Eduwebappso2bitstream10609178851mtrigasTFC0612memoria Pdf, 2012.

- [14] P. Letelier, «Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)», 2006.
- [15] J. Highsmith, *Adaptive software development: a collaborative approach to managing complex systems*. Addison-Wesley, 2013.
- [16] M. Poppendieck y T. Poppendieck, *Lean Software Development: An Agile Toolkit: An Agile Toolkit*. Addison-Wesley, 2003.
- [17] J. Stapleton, *DSDM, dynamic systems development method: the method in practice*. Cambridge University Press, 1997.
- [18] S. R. Palmer y M. Felsing, *A practical guide to feature-driven development*. Pearson Education, 2001.
- [19] R. Nazareno, H. Leone, y S. M. Gonnet, «Trazabilidad de procesos ágiles: un modelo para la trazabilidad de procesos Scrum», en *XVIII Congreso Argentino de Ciencias de la Computación*, 2013.
- [20] Marco Morales, «Product Manager – Agile en acción (Scrum)», 05-may-2015. .
- [21] «Lista de tareas de la iteración (Sprint Backlog)». [En línea]. Disponible en: <https://proyectosagiles.org/lista-tareas-iteracion-sprint-backlog/>.
- [22] Instituto Ecuatoriano de Estadísticas y Censos (INEC), «Directorio de Empresas y Establecimientos».

ANEXO 1: DIAGRAMA DEL PROCESO DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE



ANEXO 2: ACTA DE REUNIÓN DE SPRINT INICIAL

N° SPRINT	000001	1
Fecha	21-08-2017	

REUNIÓN DE INICIO DE SPRINT

OBJETIVOS DE LA REUNIÓN
Reunión para determinar las funcionalidades o historias de usuario que se van a incluir en el próximo sprint.

DATOS DEL PROYECTO	
Origen	Cliente
Empresa	INPEX SA
Tipo de requerimiento	Nueva Funcionalidad
Proyecto	Modulo Facturación - Funcionalidad de Proformas y pedidos.
Patrocinador Principal	INPEX SA
ASISTENTES	
Lider de sistemas	Ing. Carlos Gómez
Analista Funcional	Ing. Luris Perez
Analista Desarrollador	Ing. Jack Gamboa Ing. Geovany Giron
Arquitecto de Software	Ing. David Maldonado
Analista de control de calidad	Ing. Jonathan Diaz
Cliente / Usuario	Jefe de Facturación Vendedor
Otros Interesados	Lider de soporte de Usuario Analista de Soporte de Usuario

DEFINICIÓN DE RESPONSABILIDADES	
Responsabilidades del Dueño del Producto(Product Owner)	<ul style="list-style-type: none"> Asistencia a la reunión. Exposición y explicación de las historias que necesita para la próxima iteración y posibles restricciones de fechas que pudiera tener.
Responsabilidades del Scrum Master	<ul style="list-style-type: none"> Moderación de la reunión Decide cuales son los requisitos a implementar para la entrega. La decisión siempre se consulta con el la siguiente entrega. Esta decisión siempre se consulta con el cliente
Responsabilidades del equipo SCRUM	<ul style="list-style-type: none"> Confección de la pila del sprint. Auto-asignación del trabajo

N° SPRINT	000001	2
Fecha	21-08-2017	

PILA DEL SPRINT	
Modulo	Facturación
Opción	Proformas y Pedidos
Objetivo del Requerimiento	Agregar funcionalidad que permita desglosar productos compuestos al momento de elaborar la proforma.
Detalle	<ol style="list-style-type: none"> 1. En la opción de parámetros básicos del Módulo de Facturación se debe crear parámetro: <i>Desea descomponer Productos Compuestos</i> 2. Al seleccionar un producto validar Tipo de producto (SIMPLE / COMPUESTO) 3. En caso de ser Producto compuesto Deberá preguntar: <i>¿Desea descomponer el Producto? Si o no,</i> 4. En el caso de Si cargar automáticamente los componentes (si en Parámetros básicos esta parametrizado como Si en la opción de descomponer Productos compuestos. <ol style="list-style-type: none"> 1. Estos componentes podrán ser modificados dentro de la proforma sin afectar la plantilla creada en inventario. 2. Se podrán agregar, modificar o eliminar ítems correspondientes al componente. 3. Ajustar formato de impresión de proforma

Definición de fechas	
FECHA INICIO	21-08-2017
FECHA ENTREGA	01-09-2017



mastering

 Servicios Empresariales

ANEXO 3: DOCUMENTO FUNCIONAL



RUT	IMP-00001
CLIENTE	IMPEX SA
DESARROLLADOR	JACKSON GAMBOA
ANALISTA FUNCIONAL	LURIS PEREZ
SPRINT	000001
FECHA INICIO	21-08-2017
FECHA FIN	01-09-2017

DOCUMENTO FUNCIONAL

MODULO: FACTURACIÓN

OPCIÓN: PROFORMAS y PEDIDOS

Proformas de Productos Compuestos

Situación actual:

- En el módulo de inventario en la opción Productos el usuario puede registrar Productos de Subtipo: Compuestos

Definir un producto con subtipo Compuesto

Defino las componentes que forman parte del producto compuesto.

- b) Al traer un producto compuesto desde la opción de Proformas, esta no trae sus componentes (estos componentes son parametrizados en el módulo de Inventario / Producto / botón Componentes).



Figura 2. Llamada a un producto Compuesto

Solución requerida:

1. En la opción de parámetros básicos del Módulo de Facturación se debe crear parámetro: **Desea descomponer Productos Compuestos**
2. Al traer un producto compuesto desde la opción de Proformas, ya sea utilizando la tecla F3 o digitando el código:

Si está marcada la opción **Desea descomponer Productos Compuestos**,

Deberá preguntar: **¿Desea descomponer el Producto? Si o no.**

En caso de que la respuesta sea Si, Deberá traer todos los componentes que conforman dicho producto y cargarlo en el detalle de los Items...

1. Estos componentes podrán ser modificados dentro de la proforma sin afectar la plantilla creada en inventario.
2. Se podrán agregar, modificar o eliminar Items correspondientes al componente.
3. Ajustar formato de impresión de proforma

En caso que no dese descomponer solo deberá traer el ítem del producto como lo hace hasta ahora.

ANEXO 4: DOCUMENTO TÉCNICO



RUT	IMP-00001
CLIENTE	IMPEX SA
DESARROLLADOR	JACKSON GAMBOA GEOVANNY GIRON
ARQUITECTO DE SOFTWARE	DAVID MALDONADO
SPRINT	000001
FECHA INICIO	21-08-2017
FECHA FIN	01-09-2017

DOCUMENTO TÉCNICO

BACK - END

TABLAS

FCONTRL			
Motivo	Campo	Tipo	Presentación
Nuevo Campo	DESCRIPCION	varchar(11)	Desea Desglosar Items de productos compuestos en la proforma

PROCEDIMIENTOS ALMACENADOS

Procedimientos almacenados a modificar:

1. dbo.SP_FA_IN_CC_FCONTRL
2. dbo.SP_FA_DETALLE_COMPONENTES
3. dbo.SP_FA_OBTIENE_PEDIDO
4. dbo.SP_FA_UPDATE_PEDIDO

FRONT- END

Librerías a Modificar:

1. SAI_GENERAL.PBL
2. FUNCION2.PBL
3. FA_WIN.PBL
4. FA_REPORT.PBL

mastering
Servicios Empresariales

Escenario		Proformas y Pedidos		Participantes:		Tipo de Prueba			Ambiente		
ID	Fecha	Modulo	Sub-Escenario	Pre condición	Post condiciones	Rufo Normal	Rujos Alternos	Excepciones	Resultados esperados	Resultados Obtenidos	Observaciones
3	25/08/2017	Facturación	Crear Proforma usando Productos Simples	PB: Opción Desea desglosar Items de Producto - NO	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
4	25/08/2017	Facturación	Editar Proforma usando Productos Simples	PB: Opción Desea desglosar Items de Producto - NO	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
5	25/08/2017	Facturación	Anular Proforma usando Productos Simples	PB: Opción Desea desglosar Items de Producto - NO	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
6	25/08/2017	Facturación	Impresión Proforma usando Productos Simples	PB: Opción Desea desglosar Items de Producto - NO	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
7	25/08/2017	Facturación	Crear Proforma usando Productos Compuestos	PB: Opción Desea desglosar Items de Producto - NO	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
8	25/08/2017	Facturación	Editar Proforma usando Productos Compuestos	PB: Opción Desea desglosar Items de Producto - NO	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
9	25/08/2017	Facturación	Anular Proforma usando Productos compuestos	PB: Opción Desea desglosar Items de Producto - NO	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
10	25/08/2017	Facturación	Impresión Proforma usando Productos Compuestos	PB: Opción Desea desglosar Items de Producto - NO	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
11	25/08/2017	Facturación	Crear Proforma usando Productos combinados (Simple y Compuestos)	PB: Opción Desea desglosar Items de Producto - NO	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
12	25/08/2017	Facturación	Editar Proforma usando Productos combinados (Simple y Compuestos)	PB: Opción Desea desglosar Items de Producto - NO	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
13	25/08/2017	Facturación	Anular Proforma usando Productos combinados (Simple y Compuestos)	PB: Opción Desea desglosar Items de Producto - NO	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
14	25/08/2017	Facturación	Impresión Proforma usando Productos combinados (Simple y Compuestos)	PB: Opción Desea desglosar Items de Producto - NO	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		

Escenario		Proformas y Pedidos		Participantes:		Tipo de Prueba			Ambiente		
ID	Fecha	Modulo	Sub-Escenario	Pre condición	Post condiciones	Rufo Normal	Rujos Alternos	Excepciones	Resultados esperados	Resultados Obtenidos	Observaciones
15	25/08/2017	Facturación	Crear Proforma usando Productos Simples	PB: Opción Desea desglosar Items de Producto - SI	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
16	25/08/2017	Facturación	Editar Proforma usando Productos Simples	PB: Opción Desea desglosar Items de Producto - SI	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
17	25/08/2017	Facturación	Anular Proforma usando Productos Simples	PB: Opción Desea desglosar Items de Producto - SI	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
18	26/08/2017	Facturación	Impresion Proforma usando Productos Simples	PB: Opción Desea desglosar Items de Producto - SI	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
19	25/08/2017	Facturación	Crear Proforma usando Productos Compuestos	PB: Opción Desea desglosar Items de Producto - SI	-	Preguntar: Desea desglosar Items Compuestos - SI	-	-	Automaticamente carga los componentes del producto seleccionado		
20	25/08/2017	Facturación	Crear Proforma usando Productos Compuestos : Cambio de Items	PB: Opción Desea desglosar Items de Producto - SI	-	Cambiar Items de la carga realizada	-	-	Permitir cambiar Items predefinidos		
21	25/08/2017	Facturación	Crear Proforma usando Productos Compuestos : Anadir Items	PB: Opción Desea desglosar Items de Producto - SI	-	Añadir Items a la carga realizada	-	-	Permitir agregar Items predefinidos		
22	25/08/2017	Facturación	Crear Proforma usando Productos Compuestos : Eliminar de Items	PB: Opción Desea desglosar Items de Producto - SI	-	Eliminar Items a la carga realizada	-	-	Permitir eliminar Items predefinidos		
23	25/08/2017	Facturación	Editar Proforma usando Productos Compuestos : Cambio de Items	PB: Opción Desea desglosar Items de Producto - SI	-	Cambiar Items de la carga realizada	-	-	Permitir cambiar Items predefinidos		
24	25/08/2017	Facturación	Editar Proforma usando Productos Compuestos : Anadir Items	PB: Opción Desea desglosar Items de Producto - SI	-	Añadir Items a la carga realizada	-	-	Permitir agregar Items predefinidos		
25	25/08/2017	Facturación	Editar Proforma usando Productos Compuestos : Eliminar de Items	PB: Opción Desea desglosar Items de Producto - SI	-	Eliminar Items a la carga realizada	-	-	Permitir eliminar Items predefinidos		
26	25/08/2017	Facturación	Anular Proforma usando Productos compuestos	PB: Opción Desea desglosar Items de Producto - SI	-	Función Normal de La opción	-	-	No debe alterarse la Función normal de la Opción		
27	25/08/2017	Facturación	Impresion Proforma usando Productos Compuestos	PB: Opción Desea desglosar Items de Producto - SI	-	Función Normal de La opción	-	-	Presentar formato predefinido para productos compuestos		

Escenario		Proformas y Pedidos		Participantes:		Tipo de Prueba			Ambiente		
ID	Fecha	Modulo	Sub-Escenario	Pre condición	Post condiciones	Rujo Normal	Rujos Alternos	Excepciones	Resultados esperados	Resultados Obtenidos	Observaciones
28	25/08/2017	Facturación	Crear Proforma usando Productos Compuestos	PB: Opción Desea desglosar Items de Producto - SI	-	Preguntar: Desea desglosar Items Compuestos - NO	-	-	No debe alterarse la Función normal de la Opción		
29	25/08/2017	Facturación	Crear Proforma usando Productos combinados (Simple y Compuestos)	PB: Opción Desea desglosar Items de Producto - SI	-	Preguntar: Desea desglosar Items Compuestos - SI	-	-	Automaticamente carga los componentes del producto seleccionado		
30	25/08/2017	Facturación	Crear Proforma usando Productos combinados (Simple y Compuestos) - Cambios de Items	PB: Opción Desea desglosar Items de Producto - SI	-	Preguntar: Desea desglosar Items Compuestos - SI	-	-	Permitir cambiar items predefinidos		
31	25/08/2017	Facturación	Crear Proforma usando Productos combinados (Simple y Compuestos) - Añadir Items	PB: Opción Desea desglosar Items de Producto - SI	-	Función Normal de la opción	-	-	Permitir agregar items predefinidos		
32	25/08/2017	Facturación	Crear Proforma usando Productos combinados (Simple y Compuestos) - Eliminar Items	PB: Opción Desea desglosar Items de Producto - SI	-	Función Normal de la opción	-	-	Permitir eliminar items predefinidos		
33	25/08/2017	Facturación	Editar Proforma usando Productos combinados (Simple y Compuestos) - Cambiar Items	PB: Opción Desea desglosar Items de Producto - SI	-	Función Normal de la opción	-	-	Permitir cambiar items predefinidos		
34	25/08/2017	Facturación	Editar Proforma usando Productos combinados (Simple y Compuestos) - Añadir Items	PB: Opción Desea desglosar Items de Producto - SI	-	Función Normal de la opción	-	-	Permitir agregar items predefinidos		
35	25/08/2017	Facturación	Editar Proforma usando Productos combinados (Simple y Compuestos) - Eliminar Items	PB: Opción Desea desglosar Items de Producto - SI	-	Función Normal de la opción	-	-	Permitir eliminar items predefinidos		
36	25/08/2017	Facturación	Anular Proforma usando Productos combinados (Simple y Compuestos)	PB: Opción Desea desglosar Items de Producto - SI	-	Función Normal de la opción	-	-	No debe alterarse la Función normal de la Opción		
37	25/08/2017	Facturación	Impresión Proforma usando Productos combinados (Simple y Compuestos)	PB: Opción Desea desglosar Items de Producto - SI	-	Función Normal de la opción	-	-	Presentar formato predefinido para productos compuestos		



XYZ S.A.
DOMINIUM CONTROL

CASOS DE PRUEBAS TÉCNICAS

Sprint: 000001 Proyecto: IMP-00001 Metodo Utilizados: _____
 Fecha de Inicio: 21/08/2017 Responsable: David Maldonado Componentes: _____
 Fecha de Final: 01/09/2017 Objetivo: Nueva Funcionalidad - Proformas y Pedidos

ID	Fecha	Proformas y pedidos		Participantes	Desarrollo		Tipo de Prueba	Parámetros	Ambiente	Desarrollo	
		Módulo	Casos		Específicos SP	Genera Log				Nombre Sp	Resultados esperados
1	28/08/2017	BD	Alter Table FCCONTROL	Back - End	-	-	-	-	Ejecutar Alter		
2	28/08/2017	BD	Añadir Nuevo parametro	Back - End	SP_FA_IN_CC_FCCONTROL	-	SP_FA_IN_CC_FCCONTROL	@DESGPROD	S / N		
3	28/08/2017	BD	Tree componentes	Back - End	SP_FA_DETALLE_COMPONENTES	-	SP_FA_DETALLE_COMPONENTES	-	Consulta detalle de componentes		
4	28/08/2017	BD	Valida tipo de Producto: Simple	Back - End	SP_FA_OBTIENE_PEDIDO	-	SP_FA_OBTIENE_PEDIDO	-	Funcionamiento Normal		
5	28/08/2017	BD	Valida tipo de Producto: Compuesto	Back - End	SP_FA_OBTIENE_PEDIDO	-	SP_FA_OBTIENE_PEDIDO	-	Carga de items		
6	28/08/2017	BD	Guarda Proforma	Back - End	SP_FA_UPDATE_PEDIDO	-	SP_FA_UPDATE_PEDIDO	-	Guarda datos a la BD		
7	28/08/2017	General	w_parametros	Front - End	SP_FA_IN_CC_FCCONTROL	Parámetros Básico	SP_FA_IN_CC_FCCONTROL	@DESGPROD	Parámetro Nuevo. Desaglosar items		
8	28/08/2017	Facturación	de_proformas	Front - End	SP_FA_OBTIENE_PEDIDO	Proformas y pedido	SP_FA_OBTIENE_PEDIDO	-	Pregunta si desea descomponer items		
9	28/08/2017	Facturación	de_componentes	Front - End	SP_FA_DETALLE_COMPONENTES	Productos	SP_FA_DETALLE_COMPONENTES	-	pantalla de componentes		
10	28/08/2017	BD	Guarda Proforma	Front - End	SP_FA_UPDATE_PEDIDO	-	SP_FA_UPDATE_PEDIDO	-	Guardar cambios realizados		