

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, MANAGUA

UNAN-MANAGUA

FACULTAD DE CIENCIAS ECONÓMICAS

CENTRO UNIVERSITARIO DE DESARROLLO EMPRESARIAL



**TESIS DE MAESTRÍA PARA OPTAR AL TÍTULO DE:
MASTER EN ADMINISTRACIÓN FUNCIONAL DE EMPRESAS**

**TEMA DE INVESTIGACIÓN:
ANÁLISIS DE GESTIÓN DE PROCESOS DE DESARROLLO DE
SISTEMAS INFORMÁTICOS DE LA DIVISIÓN DE INFORMÁTICA Y SISTEMAS
PARA EL PERIODO 2019-2020**

**AUTOR:
ING. GABRIELA ROQUE FARIÑAS**

**TUTOR:
MSC. LUIS OCTAVIO TENORIO**

MANAGUA, NICARAGUA

DICIEMBRE 2022



i. Dedicatoria

A Dios, por ser la fuerza que me motiva, la mano que me sostiene, por su infinito amor y misericordia que cada día me demuestra y es reflejado en todas las bendiciones de mi alrededor.

A mis padres Esbeth Fariñas y Manuel Roque, por siempre confiar en mí, por apoyarme de forma incondicional en mis estudios y por los valores y educación que me han inculcado.

A mis docentes, por transmitir sus conocimientos y experiencias para poder ser profesionales exitosos de manera integral.



ii. Agradecimiento

Agradezco a Dios, por siempre poner las cosas en el momento exacto y por siempre derramar bendiciones en mi vida.

A mis amigas de curso, Nancy Hernández, Arelis Aragón, Nidia Guido y Tamara Hernández, con quienes trabajé en equipo en la realización de tareas e hicieron ameno cada día de clases.

A Carlos Miranda, por su apoyo incondicional durante todo este proceso de estudio y culminación, por sus palabras de motivación, amor y cariño para no rendirme.

A la Dirección General de Ingresos, por becar mis estudios de maestría y alcanzar este logro profesional.

A mi tutor Msc. Luis Tenorio, por su tiempo dedicado, por orientarme y corregirme en todo lo referente a la tesis.



iii. Carta Aval de Tutor/a

MSc. Ramfis Muñoz

Director / Centro de Desarrollo Empresarial PROCOMIN

Su despacho

Estimado maestro:

En cumplimiento con lo establecido en los artículos 97,20 inciso a y b y 101 del Reglamento de sistema de estudios de posgrado y educación continua SEPEC-MANAGUA, aprobado por el Consejo Universitario en sesión ordinaria No. 21-2011, del 07 de octubre 2011, Por este medio dictamino en informe final de investigación de tesis para su defensa titulada: **Análisis de gestión de procesos de desarrollo de sistemas informáticos de la División de Informática y Sistemas para el periodo 2019-2020**, realizada por la Ingeniera Gabriela Roque Fariñas, como requisito para optar el título de Master en Administración Funcional de Empresas, cumple con los requisitos establecidos en este reglamento.

Como tutor de Tesis de la Ingeniera Gabriela Roque Fariñas, considero que contiene los elementos científicos, técnicos y metodológicos necesarios para ser sometidos a Defensa ante el Tribunal Examinador.

El trabajo de Roque Fariñas, se enmarca en las líneas de investigación del programa de Maestría referido a la solución de la gestión de procesos de desarrollo de sistemas informáticos.

Dado en la ciudad de Managua, Nicaragua a los quince días de diciembre del año dos mil veintidós.

MSc. Luis Octavio Tenorio

Tutor

iv. Resumen

La División de Informática y Sistemas (DIS) de la Administración Tributaria elabora los sistemas tributarios que están a la disposición de los contribuyentes, entidades y servidores públicos con el fin de mejorar las gestiones fiscales y cumplir con los objetivos institucionales. Durante el desarrollo de estos, surgen ciertos inconvenientes que ocasionan retrasos de entrega o fallas de funcionalidad una vez puestos en ambiente productivo.

En las últimas auditorías realizadas a la DIS se hace la observación del poco control de la documentación de sus procesos. Tras esta perspectiva, se realizó un estudio para analizar la gestión de desarrollo de sistemas tributarios, logrando describir su proceso actual, el cual se asemeja a los procesos de metodologías tradicionales, con un levantamiento de requisitos y una entrega productiva al final.

También se estableció su eficiencia, detectando algunas particularidades como tiempos de desarrollo muy cortos, recursos subutilizados y gran cantidad de bitácoras recibidas por inconsistencias o requerimientos para mejoras en los procesos por escenarios no previstos, lo que sugiere funcionalidades no contempladas durante su evaluación inicial.

Se realizó diagnóstico de la situación actual a través de un análisis FODA y se determinó las causas y consecuencias del problema central, destacando tres principales que incluyen la diferencias entre las plataformas en que están desarrollados los sistemas, los formatos utilizados y la carencia de una metodología que gestione cada proceso.

Para solucionar las consecuencias detectadas, se partió de las causas para realizar una propuesta de mejora que incluye todos estos elementos antes mencionados y ayude a los líderes y altos directivos a gestionar de manera eficiente los procesos de desarrollo de sistemas, logrando reducir los tiempos de puesta en producción. minimizar los errores en los aplicativos que ocasionan molestias a los usuarios finales y contribuir al cumplimiento de los objetivos y estrategias institucionales.



Contenido

i. Dedicatoria	i
ii. Agradecimiento	ii
iii. Carta Aval de Tutor/a	iii
iv. Resumen	iv
I. Introducción	1
II. Planteamiento del problema	4
2.1. Planteamiento del problema	4
2.2. Formulación del problema	5
III. Justificación	6
IV. Antecedentes	7
4.1. Antecedentes Teóricos	7
4.2. Antecedentes de Campo	8
V. Objetivos de la investigación	10
5.1. Objetivo General	10
5.2. Objetivos específicos	10
VI. Marco Teórico	11
6.1. Perfil de la Administración Tributaria	11
6.1.1. Creación	11
6.1.2. Organización	12
6.2. Generalidades de la División de Informática y Sistemas	12
6.3. Normas Técnicas de Control Interno	14
6.4. Desarrollo de software	15
6.5. Ciclo de Vida del software	16
6.6. Metodologías tradicionales y ágiles de desarrollo de software, estándares y marcos de referencia	17



6.6.1. Metodologías tradicionales.....	19
6.6.2. Desarrollo de software ágil.....	20
6.6.3. Normas y marcos de referencia.....	25
6.7. Entornos de desarrollo y control de software.....	28
6.8. Terminología empleada en los procesos institucionales.....	30
6.9. Análisis FODA.....	32
VII. Preguntas Directrices.....	33
VIII. Operacionalización de Variables.....	34
IX. Diseño Metodológico.....	37
9.1. Tipo de Estudio.....	37
9.2. Tipo de Enfoque.....	37
9.3. Universo o Población.....	37
9.4. Selección de Muestra.....	38
9.5. Recopilación de Información.....	38
9.6. Procesamiento de Información.....	39
X. Análisis de Resultados.....	42
10.1. Descripción de la gestión actual de procesos de desarrollo de sistemas informáticos.....	42
10.1.1. Definición de procesos.....	42
10.1.2. Descripción de procesos.....	44
10.1.3. Documentos y sistemas de control durante los procesos.....	63
10.2. Eficiencia de la gestión actual de los procesos de desarrollo de sistemas informáticos.....	65
10.3. Diagnóstico de la gestión actual del proceso de desarrollo de sistemas informáticos determinando sus causas y consecuencias.....	71
10.4. Propuesta para la gestión de procesos de desarrollo de sistemas informáticos en base a buenas prácticas de desarrollo de software.....	73



10.4.1. Plataformas de desarrollo.....	73
10.4.2. Formatos de requerimientos, acta de entrega, hoja de pruebas y bitácoras. 75	
10.4.3. Metodología a implantar	77
XI. Conclusiones.....	87
XII. Recomendaciones	89
XIII. Referencias	90
XIV. Anexos	94
14.1. Anexo A: Entrevista dirigida a especialistas a cargo en informática 94	
14.2. Anexo B: Entrevista dirigida a especialistas de área de informática 95	
14.3. Anexo C: Requerimiento de Información Estadística	97
14.4. Anexo D: Guía de Observación	100
14.5. Anexo E: Guía de Revisión Documental	102
14.6. Anexo F: Instructivo para llenado de formato de requerimientos informáticos. 105	
14.7. Anexo G: Instructivo para llenado de formato de Acta de Entrega y Hoja de pruebas.....	116
14.8. Anexo H: Instructivo de llenado de formato de hoja de pruebas..	122
14.9. Anexo I: Instructivo de llenado de formato de bitácora.....	128
14.10. Anexo J: Manual Propuesto de Desarrollo de Sistemas	135
XV. Glosario de términos técnicos.....	156

Índice de Figuras

Figura 1: Ciclo de vida en cascada.....	19
Figura 2: Proceso de la ISO/IECC 12007	26
Figura 3: Proceso actual de desarrollo de sistemas informáticos	43



Figura 4: Diagrama de Flujo Expresión de la necesidad.....	46
Figura 5: Diagrama de flujo Recibimiento y análisis de requerimiento	48
Figura 6: Diagrama de flujo Desarrollo y retroalimentación de Requerimiento	50
Figura 7: Diagrama de flujo del proceso de pruebas y ajuste	52
Figura 8: Diagrama de flujo Aceptación de pruebas	54
Figura 9: Proceso Corrección de errores de los sistemas informáticos	55
Figura 10: Diagrama de flujo Recibimiento y análisis de bitácoras	57
Figura 11: Diagrama de flujo Desarrollo y retroalimentación de bitácoras..	59
Figura 12: Diagrama de flujo del proceso de pruebas y ajuste	61
Figura 13: Diagrama de flujo del proceso de Aceptación de corrección de bitácoras.....	63
Figura 14: Metodología SCRUM.....	83

Índice de Gráficos

Gráfico 1 Diagrama de Flujo Acumulado KANBAN:	85
---	----

Índice de Tablas

Tabla 1: Diferencias entre metodologías ágiles y tradicionales	18
Tabla 2 Operacionalización de Variables.....	34
Tabla 3: Recopilación de datos Entrevista 1	39
Tabla 4: Recopilación de datos Entrevista 2.....	39
Tabla 5: Recopilación de datos Control de plataformas	40
Tabla 6: Recopilación de datos Preguntas de análisis	40
Tabla 7: Recopilación de datos Comparación de cantidad de nuevos sistemas	41
Tabla 8: Expresión de la Necesidad	45
Tabla 9: Proceso de Recibimiento y Análisis de requerimiento	47
Tabla 10: Proceso de Desarrollo y retroalimentación	49
Tabla 11: Pruebas y periodo de ajuste	51
Tabla 12: Aceptación de requerimiento	53
Tabla 13: Proceso Recibimiento y Análisis de bitácoras.....	55



Tabla 14: Desarrollo y retroalimentación para bitácoras.....	58
Tabla 15: Pruebas y periodo de ajuste	60
Tabla 16: Aceptación de requerimiento	62
Tabla 17: Requerimientos por mejoras vs Requerimientos nuevos.....	67
Tabla 18: Bitácoras recibidas por inconsistencias vs Bitácoras por procesos no automatizados	68
Tabla 19: Calificación de las características para eficiencia	71
Tabla 20: Matriz Análisis FODA . Factores Internos	71
Tabla 21: Matriz Análisis FODA - Factores Externos.....	72
Tabla 22: Causas y Consecuencias del descontrol y atraso del cumplimiento de desarrollo de los requerimientos y bitácoras de los sistemas informáticos	72
Tabla 23: Plataformas de desarrollo a migrar	74
Tabla 24: Plataformas de control de versiones.....	74
Tabla 25: Plataforma de control de versiones.....	75
Tabla 26: Selección de metodologías.....	78
Tabla 27: Realización de BackLog	80
Tabla 28: Sprint Planning.....	80
Tabla 29: Daily Stand Up	81
Tabla 30: Sprint Review	82
Tabla 31: Retrospectiva	82
Tabla 32: Tablero Visual KANBAN	84
Tabla 33: Tabla para elaborar diagrama de flujo acumulado.....	85



I. Introducción

La Dirección General de Ingresos, en adelante Administración Tributaria, es la encargada de administrar las leyes fiscales y recaudar los impuestos internos del país. La administración tributaria brinda servicios para el cumplimiento de las obligaciones fiscales, entre ellos la disponibilidad de sistemas tributarios en línea para todos los contribuyentes, entidades y servidores públicos que son parte del proceso de recaudación.

La División de Informática y Sistemas (DIS) es la encargada de desarrollar estos sistemas de acuerdo a las necesidades de la administración tributaria, sin embargo, estos sistemas salen al público de manera tardía o con algunos errores detectados una vez que los usuarios finales están usándolo.

Los procesos para desarrollar los sistemas impactan de manera directa sobre estos, existen diferentes metodologías o marcos de referencia que indican como realizar estos procesos, es por eso, que se realizó un análisis de la gestión de los procesos de desarrollo para los sistemas tributarios, y determinar aquellos elementos que estén afectando la calidad y eficiencia de los mismos.

Para lograr este análisis, el desarrollo de este trabajo está compuesto por los siguientes capítulos:

El capítulo I, que es la presente introducción, sitúa al lector del tema abordado durante esta investigación y su contenido.

El capítulo II, se encuentra el planteamiento y formulación del problema detectado, describiendo los acontecimientos para establecer el problema central y su posible control de pronóstico.

El capítulo III, contiene la justificación para elaborar esta investigación, abordando su importancia, quienes serán beneficiados, quienes lo usaran y cómo lo usaran.



El capítulo IV, establece los antecedentes teóricos, referenciando las tesis relacionadas al tema abordado pero que no se llegaron a implementar, y lo antecedentes de campo, refiere a una tesis donde se logró implementar la teoría y estuvo en observación durante varios años hasta su declive.

El capítulo V, está compuesto por los objetivos de este documento, estableciendo uno general y los específicos que ayudan a lograr el objetivo general.

El capítulo VI, se presenta el marco teórico en el cual se detallan los aspectos teóricos necesarios para el desarrollo de este documento, estableciendo las bases técnicas en las que se basó la investigación.

El capítulo VII, se realizaron las preguntas directrices que están directamente relacionadas a los objetivos y la problemática de este documento.

El capítulo VIII, se realizó la operacionalización de variables, derivados de los objetivos de esta investigación, donde se determinaron los indicadores que permitieron obtener las interrogantes de los instrumentos recopilados.

El capítulo IX, abarca el diseño metodológico, estructurado por el tipo de estudio, tipo de enfoque, el universo estudiado, la muestra seleccionada, los instrumentos recopilados y cómo se procesó la información.

El capítulo X, contienen el análisis de resultados, detallando la interpretación de los datos obtenidos a través de los instrumentos recopilados como las entrevistas, guía de observación, revisión documental y la información estadística proporcionada.

Todos esto abordado en tres grandes acápite de acuerdo a los objetivos propuestos. Se realizó una descripción de todos los procesos involucrados en el desarrollo de software, creando diagramas para su mejor comprensión y especificando todos los elementos que lo componen. Además, se analizó la eficiencia que poseen estos sistemas con respecto a ciertos parámetros y se determinaron las causas y consecuencias que ocasionan la problemática principal.



También se realizó una propuesta de mejora donde se consideran las causas y a través de ahí tratar de erradicarlas. Se identificaron 3 causas principales y para cada una de ellas una propuesta que dé una solución viable.

El capítulo XI, establece las conclusiones de esta investigación de manera puntual, relacionado con los objetivos de esta investigación.

El capítulo XII, son las recomendaciones, que son sugerencias realizadas para la implementación de esta tesis.

El capítulo XIII, contiene las referencias utilizadas durante la elaboración de este documento y el capítulo XIV, muestra los anexos, aborda los instrumentos utilizados e instrumentos y manuales realizados como propuesta de solución a la tesis, referenciados en sus capítulos correspondientes.



II. Planteamiento del problema

2.1. Planteamiento del problema

En las últimas auditorías realizadas en los años 2019 y 2020 a los sistemas informáticos de la División de Informática y Sistemas se le ha hecho observaciones sobre la inexistente o desactualizada documentación de los procesos que estos conllevan, sin embargo, estos procesos se realizan de manera empírica, sin algún estándar que indique los procedimientos a seguir en cada etapa de la elaboración del sistema.

Durante las auditorías, se han detectado ciertos errores o inconsistencias en datos, documentos y debilidades en los procesos de los sistemas que son reflejados en la cantidad de bitácoras o requerimientos recibidos para superar dichas debilidades, consecuente de la poca organización en los procesos de desarrollo de los sistemas informáticos.

Los procesos a considerar para la elaboración de esta investigación serán los de análisis, desarrollo y pruebas de sistemas. Cada proceso implica diferentes etapas que los analistas involucrados realizan a manera de experiencia, es decir, según una inducción inicial realizada por otros analistas. Estos son ejecutados principalmente por los analistas desarrolladores y los analistas de control de calidad, aunque también se incluyen los cargos de jefes y líderes de áreas.

Los analistas desarrolladores por lo general eligen la plataforma en la cual desarrollar, el lenguaje de programación, el diseño de los componentes del sistema y gestor de control de versiones, ocasionando que los sistemas informáticos varíen en todas estas características.

Los analistas de control de calidad realizan sus pruebas en base un documento realizado por los analistas desarrolladores contemplando solo la parte funcional del sistema, sin diferentes escenarios definidos, evaluación de riesgos o requisitos no funcionales.



Otros elementos a considerar durante cada proceso son los requerimientos recibidos para elaboración de los sistemas, que en ocasiones están incompletos o poco comprensibles, los tiempos de ejecución son poco flexibles y ajustados, existe sobrecarga de trabajo para algunos de los analistas y sus roles no están muy definidos. El canal de comunicación es por correo electrónico para cualquier tipo de incidencia o tarea realizada, por lo que no hay un seguimiento o control centralizado para verificación del estado de desarrollo de los sistemas.

Consecuentemente a esta poca organización, se torna difícil la identificación y mitigación de riesgos, identificación de oportunidades de mejora y correcciones, establecimiento de niveles de relación entre las áreas involucradas, claridad en el flujo del proceso y establecimiento de indicadores de desempeño.

De seguir con esta forma de trabajo la eficiencia y calidad de los sistemas tributarios proporcionados por la Administración Tributaria serán afectados impactando directamente sobre los usuarios y la recaudación de impuestos.

La ausencia de una metodología y documentación que sirva como guía o manual sobre los procesos que se deben de realizar para la creación de sistemas, es uno de los principales factores a considerar para erradicar esta problemática, y por consecuente para poder identificar una metodología de acuerdo a sus necesidades y elaborar un manual acorde a lo requerido, es necesario realizar un “Análisis de gestión de procesos de desarrollo de los sistemas de la División de Informática y Sistemas”.

2.2. Formulación del problema

¿Cómo se puede mejorar la gestión de los procesos de desarrollo de sistemas informáticos de acuerdo al periodo 2019-2020?



III. Justificación

Este documento, tiene como propósito indagar, analizar y documentar los procesos que se deberían implementar para controlar de manera eficiente y segura la gestión en cada sistema informático.

Realizar un análisis de la gestión de los procesos de sistemas informáticos en la División de Informática y Sistemas es importante porque ayudará a detectar diferencias entre lo que se realiza actualmente y cómo se debería hacer, establecer la metodología de desarrollo de sistemas de acuerdo a sus necesidades, descubrir y documentar las debilidades de los procesos, mitigar los riesgos y medir la eficiencia en cada una de las etapas. Además, tener una estandarización en los procesos de desarrollo de sistemas ayudará con los objetivos institucionales, como consecuencia de mayor eficacia en los mismos.

Se beneficiarán las áreas directivas en la toma de decisiones al contar con una documentación gráfica con la cual podrán:

- ✓ Identificar fácilmente la interrelación entre los procesos
- ✓ Identificar la secuencia lógica de las actividades o tareas
- ✓ Comprensión clara de actividades y dónde podrían estar fallando
- ✓ Identificar roles y responsabilidades de los analistas en cada una de las etapas
- ✓ Identificar riesgos y puntos de control
- ✓ Facilitar la toma de tiempos de ciclo de elaboración de los sistemas
- ✓ Medir el desempeño de los analistas

Los analistas desarrolladores y analistas de control de calidad utilizarían esta guía para realizar cada uno de sus tareas de manera clara, ordenada y precisa, de forma que se optimicen los procesos y se obtengan mejores resultados.

IV. Antecedentes

4.1. Antecedentes Teóricos

En la tesis “Propuesta para la aplicación de una metodología de desarrollo e implementación de software en la División de Informática y Sistemas de la Dirección General de Ingresos” presentada por Alonso, G. en el año 2017 a la Universidad Nacional de Ingeniería para optar al título de Master en Informática Empresarial, se desarrolló una propuesta basada en un modelo ágil de software, el cual tiene ventajas sobre los modelos de ciclo de vida tradicional, al ser iterativo y flexible ante circunstancias cambiantes, entre otros beneficios.

Alonso (2017) elaboró un “diagnóstico de la situación actual mediante la evaluación de los procedimientos asociados a la gestión de proyectos de desarrollo e implementación de software” (p. 5), en esta identificó por cada área involucrada su función y las debilidades o fallas que se presentaban por proceso descrito. Realizó diagrama de procesos separando en dos principales: “la planificación de nuevos requerimientos” y “la elaboración de los requerimientos y ejecución de desarrollo”. Comparó diferentes metodologías de ciclo de vida del software ágil, seleccionó una y finalmente describió a manera general los procesos requeridos para la adecuada implantación de la metodología orientada a la gestión eficiente de proyectos.

De esta investigación se retomará parte del análisis de la situación actual y la propuesta de la metodología ágil SCRUM para el proceso de desarrollo e implementación de los softwares dentro de la División de Informática y sistemas. Servirá de apoyo para darle continuidad a su trabajo y detallar de forma más específica cada proceso.

En 2016, Rioseco L. de la Universidad de Chile, presentó la tesis titulada “ANÁLISIS Y EVALUACIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE PARA EL ÁREA DE CONSULTORÍA DE LA EMPRESA POWERDATA CHILE”, para optar al título de magister en tecnologías de la información. En esta se propuso modelar y analizar el proceso de desarrollo de software de la empresa PowerData Chile.



Este estudio contempla la definición de los procesos y métricas de las actividades para desarrollar los sistemas que solicitan a la empresa desarrolladora PowerData. Realizaron el análisis con un proyecto real de la empresa, donde definen las actividades como comercialización, planificación del proyecto, requisitos, kickoff, diseño, implementación, testing, paso a producción, cierre y control de cambios. Estos procesos o actividades fueron modelados en la herramienta EPF Composer (Eclipse Process Framework) y analizados con otra herramienta llamada AVISPA.

Durante la realización de ese estudio comprobó que había roles sobrecargados de responsabilidades y tareas con demasiados elementos de entrada que retrasaban todo el proceso. Sugirió evaluar para la empresa un modelo de ciclo de vida de software incremental o por prototipos, ya que el actual modelo de cascada crea cuellos de botella y no es flexible ante cambios. Así también recomendó revisar las responsabilidades del rol del jefe de proyectos que también ocasionó atrasos durante el proceso.

Esta tesis se relaciona con la presente investigación dado que analizan cada actividad que se realiza en la empresa para el desarrollo de software, realizan sus diagramas de procesos e identifican los puntos débiles en cada uno de ellos para luego realizar propuestas en el proceso actual.

4.2. Antecedentes de Campo

Santibañez Daniel en el año 2017, presentó memoria a la Universidad Técnica Federico Santa María, Valparaíso – Chile para optar al título de Ingeniero civil informático titulada “Diseño e implementación de una metodología de desarrollo ágil en una PYME TIC carente de metodología de desarrollo”. Esta se desarrolló en cooperación de la Universidad de Chile y una empresa (nombre desconocido) como parte del proyecto ADAPTE.

Santibañez, se propuso diseñar e implementar una metodología de desarrollo ágil, junto a las herramientas que lo soportan, que permitirían un productivo operar de una PYME TIC en Chile. Fue diseñada para los equipos de desarrollo de software



de una organización, sin perjuicio que debido a las flexibles prácticas que posee, pueda ser adaptada para ser aplicada en otras áreas en donde se practiquen disciplinas distintas.

La metodología creada se compone de 4 fases en forma secuencial, aunque a primera vista no es iterativo ni incremental como en una metodología ágil, cada fase en sí, tiene su marcado carácter iterativo y existe retroalimentación. Estas fases son inicio, elaboración, construcción y transición. Utilizó diferentes herramientas durante cada fase y diseño diagramas de proceso para cada una de ellas, además de realizar formatos para las actividades.

Fue usada por la empresa durante los años 2012 al 2014 en una docena de proyectos y fueron exitosos en su mayoría, sin embargo, hubo elementos que no fueron útiles o no pudieron implementarse debido a diferentes factores internos. La empresa cambio de liderazgo y tras múltiples conflictos fue abandonada el uso de la metodología.

Aunque esta metodología dejo de usarse en la empresa que se implementó puedo rescatar el hecho de que, tras la implementación de la metodología ágil, lograron desarrollar proyectos exitosos, fue adaptable a las PYMES, motivó y comprometió al equipo de trabajo, por lo cual hay que retomar los factores que surgieron para el abandono y mejorarlos para lograr su perduración en el tiempo.



V. Objetivos de la investigación

5.1. Objetivo General

- Analizar la gestión de procesos de desarrollo de sistemas informáticos de la División de Informática y Sistemas para el período 2019-2020 para propuesta de mejora en base a buenas prácticas de desarrollo de software.

5.2. Objetivos específicos

- Describir la gestión actual de procesos de desarrollo de los sistemas informáticos
 - Medir la eficiencia de la gestión actual de procesos de desarrollo de sistemas informáticos
 - Diagnosticar la gestión actual del proceso de desarrollo de sistemas informáticos determinando sus causas y consecuencias.
 - Diseñar propuesta de la gestión de procesos de desarrollo de sistemas informáticos en base a buenas prácticas de desarrollo de software
-



VI. Marco Teórico

6.1. Perfil de la Administración Tributaria

La Administración Tributaria cumple un papel muy importante en la recaudación y control de tributos e impuestos en un país. En Nicaragua las entidades encargadas son la Dirección General de Ingresos (DGI) y la Dirección General de Servicios Aduaneros (DGA). El tema de investigación será realizado en la Administración Tributaria.

6.1.1. Creación

La Asamblea Nacional (2000), en la Ley N° 339 Ley creadora de la Dirección General de Servicios Aduaneros y de Reforma a la Ley Creadora de la Dirección General de Ingresos, en su artículo número 2, define a la DGI y DGA, como “entes descentralizados con personalidad jurídica propia, que gozan de autonomía técnica, administrativa y de gestión de sus recursos humanos”.

La Administración Tributaria tiene como objetivo administrar los Ingresos Tributarios y las relaciones jurídicas derivadas de ellos, así como los otros Ingresos No Tributarios que se regulen a favor del Estado. Tiene jurisdicción en todo el territorio nacional y su domicilio principal en la ciudad de Managua.

Misión

Recaudar los tributos internos de manera eficaz, eficiente y transparente, promoviendo la cultura de cumplimiento voluntario, aportando al Gobierno de Reconciliación y Unidad Nacional, recursos para el Plan de Desarrollo Humano y el Programa Económico.

Visión

Ser una Administración Tributaria moderna, respetada y calificada, al servicio del pueblo nicaragüense.

6.1.2. Organización

La estructura organizacional de la Administración Tributaria está compuesta por Dirección Superior, Centro Superior de Estudios Tributarios (CSET), Asesoría Legal, Auditoría Interna, Divulgación, Oficina de Proyectos, Oficina Seguridad de la Información, Oficina de Asuntos Internos y Tributarios, División de Informática y Sistemas, División de Recursos Humanos, División de Recursos Materiales y Financieros, División de Adquisiciones, División de Planificación Estratégica, División Jurídica Tributaria, Dirección de Asistencia al Contribuyente, Dirección de Registro, Recaudación y Cobranza, Dirección de Inteligencia Fiscal, Dirección de Fiscalización, Dirección de Administraciones de Rentas, Dirección de Grandes Contribuyentes, Dirección de Catastro Fiscal y 22 Estructuras Territoriales (Administración de Rentas).

La División de Informática y Sistemas es la unidad administrativa en dependencia jerárquica con la Dirección Superior. Esta división se encarga de diseñar y desarrollar los sistemas informáticos que usa la Administración Tributaria para gestionar todo lo relacionado a los impuestos.

6.2. Generalidades de la División de Informática y Sistemas

La División de Informática y Sistemas tiene el objetivo de asegurar el diseño, desarrollo, implantación, configuración y funcionamiento de los sistemas informáticos, operativos, bases de datos, redes, telecomunicaciones, equipos y prestación de servicios requeridos por las máxima autoridad y áreas requirentes, a fin de garantizar un servicio efectivo de apoyo informático a todos los niveles de la Administración Tributaria.

Además, en el Decreto Ejecutivo N°. 20-2003 Reformas e Incorporaciones al reglamento de la Ley Creadora de la Dirección General de Servicios Aduaneros y de Reforma a la Ley Creadora de la Dirección General de Ingresos, ley no. 339, se estable en su artículo 18, División de Información y Sistemas:



- Organizar, dirigir y controlar el diseño, desarrollo, implementación y funcionamiento de los sistemas informáticos, operativos, bases de datos, redes, telecomunicaciones y equipos, para garantizar un servicio efectivo de apoyo informático a todos los niveles de la institución
- Recomendar nuevas aplicaciones tecnológicas de Hardware y Software y las formas de implantación y evaluar permanentemente el estado de los sistemas de información, equipos y software, normas y procedimientos de desarrollo, instalación, mantenimiento, operación y producción.

Esta división está estructurada de la siguiente manera:

- Oficina de Sistemas Informáticos
 - Unidad de Sistemas Tributarios
 - Unidad de Sistemas de Apoyo
 - Unidad de Contacto y Control de Calidad
- Oficina de Apoyo Tecnológico
 - Unidad de Bases de Datos y Sistemas Operativos
 - Unidad de Comunicaciones
 - Unidad de Soporte Técnico
 - Unidad Grupo de Apoyo de Sistemas Informáticos (GASI)
- Oficina de Seguridad Informática
 - Unidad Grupo de Apoyo de Sistemas Informático

Para el análisis de la gestión de los procesos de desarrollo de sistemas informáticos se estudiará los procesos que ejecutan las Unidades de Sistemas Tributarios y la Unidad de Contacto y Control de Calidad que pertenecen a la Oficina de Sistemas Informáticos. Estas áreas están en constante comunicación durante la elaboración de los sistemas, garantizando que el sistema quede listo para su posterior paso a ambiente productivo y estén disponibles para los usuarios finales.

6.3. Normas Técnicas de Control Interno

El sector público, en materia de control interno, se rigen bajo las Normas Técnicas de Control Interno (NTCI). La primera fue publica en el Diario Oficial “La Gaceta”, el 15 de junio de 1995. Posteriormente esta fue reemplazada por las NTCI aprobada el 30 de septiembre del 2004 y publicada en el Diario Oficial No. 234 del 01 de diciembre del 2004. Luego se realizó una actualización a esta norma el 06 de febrero del 2015.

Esta norma tiene el objetivo de “servir de marco de referencia para diseñar y/o ajustar los Sistemas de Administración de las Entidades del Sector Público, lo cual implica la preparación de Manuales Normativos e Instrumentos de Regulación Específicos que incorporen estas Normas” (Asamblea Nacional, NTCI, 2015).

Desde su primera versión hasta la vigente se incluye una sección para las tecnologías de información (TI). En la primera versión llamada “Sistemas Automatizados” describe a grandes rasgos y brevemente como funcionarán desde los accesos, ingreso de datos, transacciones, procesamiento de datos y medios de control. A pesar de los pocos detalles dados, exige ciertos controles sobre los procedimientos efectuados, incluyendo los de desarrollo de software.

En la versión del 2004 y 2015, se han realizado algunas modificaciones, mejorando e incluyendo nuevos controles para las tecnologías de información.

En la última versión está contemplado en el numeral 9, nombrado como “Tecnología de la Información (TI)”. Esta consta de los siguientes acápite:

- Planeación y Organización
- Adquisición e Implementación
- Entrega y Soporte Técnico
- Monitoreo y Evaluación
- Actividades específicas de Control de la TI que deberán contemplarse como mínimo las siguientes:
 - Acceso a funciones de procesamiento
 - Transacciones rechazadas o en suspenso



- Procesamiento
- Cambios de los programas
- Estructura organizativa y procedimientos
- Uso, aprovechamiento y seguridad en recursos de tecnologías

6.4. Desarrollo de software

El software según IEEE 610 (Standard Glossary of Software Engineering Terminology) se define como “programas, procedimientos y documentación y datos asociados, relacionados con la operación de un sistema informático”. El desarrollo del software consiste en crear programas informáticos que faciliten la realización de tareas en diferentes ámbitos de la vida humana. Durante los últimos años, se ha incrementado la demanda de desarrollos de software, así como mayor exigencia por parte de los clientes y usuarios, demandando mayor calidad.

La calidad de los softwares se ha instalado como objetivo tras la denominada “Crisis del software”. El término crisis del software fue utilizado por primera vez en 1968, en la primera conferencia organizada por la OTAN sobre desarrollo de software, en la cual nació la rama de la Ingeniería de Software, y la rama que instauró el ciclo de vida del software.

Este término de crisis del software expresaba las dificultades del desarrollo de software frente al rápido crecimiento de la demanda por software, de la complejidad de los problemas a ser resueltos y de la inexistencia de técnicas establecidas para el desarrollo de sistemas que funcionaran adecuadamente o pudieran ser validados.

En este proceso, solo $\frac{1}{4}$ del tiempo de desarrollo se dedicaba a las fases de análisis, diseño, codificación y pruebas, y más de $\frac{3}{4}$ del tiempo se dedicaba a correcciones y mantenimiento. Es evidente que dedicándole solo $\frac{1}{4}$ del tiempo a las primeras fases, se arrastran errores graves, sobre todo procedentes de las fases de análisis y diseño, lo que dificultaba muchísimo la implementación, produciendo constantes paradas y retrocesos para revisar este análisis y diseño.

Algunos “síntomas” que indican que el software se encuentra en un periodo de crisis son: Baja calidad del software, tiempo y presupuesto excedido, confiabilidad cuestionable, altos requerimientos de personal para desarrollo y mantenimiento.

Es por ello, que bajo las premisas que la calidad del software, está fuertemente influenciada por la calidad de los procesos de desarrollo y mantenimiento usados para construirlo, surgen propuestas como metodologías del ciclo de vida del software, normas, estándares y buenas prácticas con el objetivo de ofrecer una mayor confiabilidad, concordancia con los requisitos exigidos, menor duración del ciclo de desarrollo de sistemas y garantizando al mismo tiempo una alta calidad del software, una reducción del tiempo de llevar el producto a un ambiente productivo, mejor comunicación y colaboración entre los desarrolladores.

6.5. Ciclo de Vida del software

Según el Laboratorio Nacional de Calidad del Software de INTECO (2009), el ciclo de vida “es el conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el software es retirado o remplazado (muere). También se denomina a veces paradigma” (pág. 24).

Sin embargo, la ISO (Internacional Organization for Standadization, s.f.), en su norma 12207 define al ciclo de vida de un software como un “marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando desde la definición hasta la finalización de su uso.”

Ambas definiciones, coinciden que contienen fases, actividades o tareas para el desarrollo del software desde el principio de su idea hasta su finalización de uso.

Entre las funciones que debe tener un ciclo de vida se pueden destacar:

- Determinar el orden de las fases del proceso de software
- Establecer los criterios de transición para pasar de una fase a la siguiente



- Definir las entradas y salidas de cada fase
- Describir los estados por los que pasa el producto
- Describir las actividades a realizar para transformar el producto
- Definir un esquema que sirve como base para planificar, organizar, coordinar, desarrollar

Los modelos del ciclo de vida tradicionales del software son: el modelo de cascada, modelo en V, modelo iterativo, modelo de desarrollo incremental y modelo en espiral. Cada uno tiene sus ventajas e inconvenientes, se puede adoptar alguno de ellos de acuerdo al proyecto que se vaya a desarrollar.

6.6. Metodologías tradicionales y ágiles de desarrollo de software, estándares y marcos de referencia

Una metodología “es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo” (Ingeniería de software, 2015, pág. 39). Lo que se busca al usar una metodología es la prolijidad, corrección y control en cada etapa del desarrollo de un programa. Lo que nos permitirá una forma sistemática para poder obtener un producto correcto y libre de errores.

Las metodologías se dividen en dos tipos, las tradicionales y las ágiles. Las metodologías tradicionales definen de manera rígida todos los requisitos cuando comienza el proyecto de creación del software/sistema, de tal forma que no permiten los cambios y los procesos de desarrollo no son nada flexibles; son lineales y no se puede comenzar una fase sin terminar la anterior ni tampoco dar vuelta atrás cuando se avanza en el proceso, por lo que si es necesario implementar algún cambio de última hora no se adaptan bien creando desajustes importantes.

Las metodologías ágiles se caracterizan por su enorme agilidad y flexibilidad en los procesos de trabajo, hay una comunicación fluida en el equipo, se adapta a los imprevistos y las necesidades que surgen durante el proceso porque los ciclos se reducen y son mucho más rápidos.

En las últimas décadas, las metodologías ágiles se han impuesto sobre las tradiciones tal como indica el estudio de Project Manager Institute (PMI,2017) , que señala que 71 % de las empresas de ingeniería de software utiliza estas metodologías. Sin embargo, hay empresas que utilizan también un enfoque híbrido que se adapte a sus necesidades.

La siguiente tabla muestra las principales diferencias entre estas dos metodologías:

Tabla 1: Diferencias entre metodologías ágiles y tradicionales

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Nota: La tabla muestra las principales diferencias de las metodologías elaboradas por un estudio en la universidad politécnica de Valencia. Fuente: Canós, J et al (2005). Metodologías ágiles en el desarrollo de software

También existen normas y marcos de referencia que estandarizan los elementos del ciclo de vida del software para lograr mejores resultados en la calidad del producto de software realizado.

Entre las metodologías, normas y marcos de referencia para desarrollo de software, podemos mencionar las siguientes como las conocidas en la industria:

6.6.1. Metodologías tradicionales

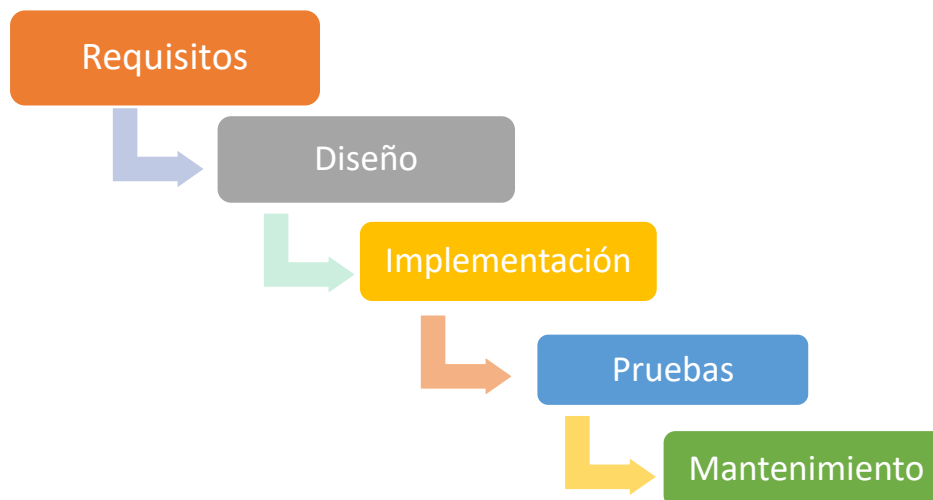
6.6.1.1. Modelo en cascada

Corresponde a una de los modelos del ciclo de vida tradicional del software. Según el Laboratorio Nacional de Calidad del Software de INTECO (2009), “es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior” (pág. 25).

Este proceso es de manera secuencial, se avanza de una fase a la siguiente sin poder regresar. Se cree que fue el primero modelo introducido y seguido ampliamente por la ingeniería del software.

Contempla las fases de:

Figura 1: Ciclo de vida en cascada



Nota. La figura muestra las fases del ciclo de vida en cascada. Fuente: Elaboración propia.



Este modelo puede ser apropiado, en general, para proyectos estables, donde los requisitos no sean cambiantes y los diseñadores predigan las áreas de problema del sistema y se produzca un diseño correcto antes de su implementación. Sin embargo, en la vida real, se da muy poco esos escenarios, y por lo general, se van cambiando los requisitos. Los resultados y mejoras se visualizan hasta el final, por lo que provoca inseguridad en los clientes que quieren ver avances en el producto. Esto implica el tener que tratar con requisitos que no se habían contemplado desde el principio del desarrollo.

A pesar de sus inconvenientes, este modelo se sigue usando en la actualidad.

6.6.2. Desarrollo de software ágil

El desarrollo ágil de software ha adquirido rápidamente popularidad en los últimos años, convirtiéndose en uno de los métodos más solicitados para el desarrollo de software, dado que adopta un enfoque diferente e innovador. Es un grupo de metodologías de desarrollo de software que se basan en principios similares.

Este tipo de metodologías, hacen hincapié en encontrar formas de complacer al cliente. Según el Laboratorio Nacional de Calidad del Software de INTECO (2009):

Las metodologías ágiles promueven generalmente un proceso de gestión de proyectos que fomenta el trabajo en equipo, la organización y responsabilidad propia, un conjunto de mejores prácticas de ingeniería que permiten la entrega rápida de software de alta calidad, y un enfoque de negocio que alinea el desarrollo con las necesidades del cliente y los objetivos de la compañía. (p. 52).

El desarrollo ágil elige hacer las cosas en incrementos pequeños con una planificación mínima, más que planificaciones a largo plazo. Las iteraciones son estructuras de tiempo pequeñas (conocidas como “timeboxes” o “sprint”) que típicamente duran de 1 a 4 semanas. De cada iteración se ocupa un equipo



realizando un ciclo de desarrollo completo, incluyendo planificación, análisis de requisitos, diseño, codificación, pruebas unitarias y pruebas de aceptación.

Esto ayuda a minimizar el riesgo general, y permite al proyecto adaptarse a los cambios rápidamente. La documentación se produce a medida que es requerida por los agentes involucrados. Una iteración puede no añadir suficiente funcionalidad para garantizar una liberación del producto al mercado, pero el objetivo es tener una versión disponible (con errores mínimos) al final de cada iteración. Se requerirán múltiples iteraciones para liberar un producto o nuevas características.

6.6.2.1. Metodología SCRUM

Se basa en el principio ágil y es conocido por sus enfoques incrementales e iterativos en su proceso de desarrollo. se puede usar para gestionar y controlar desarrollos complejos de software y productos.

Según el Laboratorio Nacional de Calidad de Software de INTECO (2009) “Los roles principales en Scrum son el ScrumMaster que mantiene los procesos y trabaja junto con el jefe de proyecto, el Product Owner que representa a las personas implicadas en el negocio y el “Team” que incluye a los desarrolladores.”

Durante cada iteración, que típicamente dura entre 2 a 4 semanas, el equipo crea un incremento de software operativo. El conjunto de características que entra en una iteración, llamados backlog, es un conjunto priorizado de requisitos de trabajo de alto nivel que se han de hacer. Estos se determinan durante la reunión de planificación de iteración.

El Product Owner, durante la reunión informa al equipo de los ítems que quiere que se completen durante la siguiente iteración. El equipo determina entonces a cuanto de eso se puede comprometer a completar. Durante la iteración, nadie puede cambiar el backlog de la iteración, los requisitos están congelados hasta la siguiente iteración.

Esta metodología SCRUM, reconoce que durante un proyecto los clientes pueden cambiar sus pensamientos sobre lo que quieren y necesitan, y que los



desafíos que no se pueden predecir, no se pueden tratar fácilmente de una forma predictiva o tradicional. Por eso, se acepta que el problema no se puede comprender o definir completamente, y trata en maximizar las habilidades del equipo para entregar rápidamente y responder a los requisitos emergentes.

El proceso de SCRUM tiene los siguientes eventos:

- Organización de trabajo pendiente: El SCRUM Master, identifica qué trabajos extraer de la lista de tareas pendientes, es decir, los trabajos a realizar.
- Sprint planning: Es una sesión de planificación donde se evaluará en qué parte del trabajo pendiente se centrará el equipo durante el sprint.
- Sprint SCRUM: Dura dos semanas, o puede ser más corto o más largo. El equipo trabajará en las tareas pendientes que se han establecido durante la sesión de planificación del sprint.
- Daily Stand Up: Es una reunión diaria para el equipo de desarrollo durante 15 minutos. Son la oportunidad para informar con respecto al trabajo que se está realizando e identificar cualquier obstáculo inesperado que haya surgido.
- Sprint Review: Presenta el trabajo durante la revisión del sprint. Tiene como máximo una duración de 4 horas para sprint por mes. El equipo presentará el trabajo terminado para la aprobación o inspección de los participantes.
- Sprint Retrospective: Se hace un análisis de cómo se desarrolló y qué podría mejorarse en el futuro.

Los artefactos utilizados en esta metodología son los siguientes:

- Product Backlog: Es el trabajo pendiente del producto, recoge la lista del trabajo que debe realizarse.
- Sprint Backlog: Es el trabajo pendiente del sprint, es decir, la serie de trabajos o productos con los que el equipo se ha comprometido durante el sprint de SCRUM.

- Incremento de producto o Release: Es lo que se entrega al final de cada sprint. Puede consistir en un nuevo producto o función, una mejora o corrección de errores, o cualquier otra cosa dependiendo del equipo.

6.6.2.2. Kanban

Kanban fue desarrollado por Taiichi Ohno, ingeniero japonés de Toyota, a fines de la década de 1940. Ohno se dio cuenta de que podía mejorar el sistema de producción de Toyota al incorporar elementos de la producción ajustada: en lugar de fabricar productos nuevos en función de la demanda anticipada, el marco Kanban de Ohno produjo y reaprovisionó productos como resultado de la demanda del consumidor; a esto se lo conoce también como el método de producción “justo a tiempo” (just in time o jit).

El marco Kanban transformó el proceso de fabricación de Toyota, de un proceso de “empuje” (los productos se introducen en el mercado) a un proceso de “extracción” (los productos se crean según la demanda del mercado). Esto significó que Toyota podía tener un nivel de inventario más bajo sin que afectara su competitividad en el mercado.

Kanban es una combinación de dos palabras japonesas: 看 (Kàn), que significa “signo” o “señal visual”, y 板 (Bǎn), que significa “tablero” (Martins, J. 2022).

Según la Guía Oficial del Método Kanban (2021), “es un método de gestión que debe aplicarse a un proceso o forma de trabajo existente. No debe cuestionarse el elegir Kanban o una metodología o un marco de trabajo. Al contrario, siempre será añadir Kanban al uso de la metodología, del marco de trabajo o de la manera de trabajar actual.” (p. 4). Hay que comprender que Kanban se aplica con sus principios y prácticas sobre un flujo y una forma de trabajo existente.

Para utilizarlo, hay que crear un tablero visible y accesible para todos los miembros del equipo. Se trazan columnas donde se anotará el estado del flujo de las tareas (tantas como sean necesarias), y así determinará el estado de cada proyecto.



Al ser un método visual permite que con un simple vistazo se conozca el estado de los proyectos y se puedan asignar nuevas tareas de manera muy efectiva.

Dentro de las prácticas generales de Kanban están:

- Visualizar: Es una clave para tener una colaboración eficaz e identificar oportunidades de mejora. Nos permite absorber y procesar una gran cantidad de información en un corto intervalo de tiempo, nos habilita la cooperación, ya que todos vemos la misma imagen.
- Limitar el Trabajo en Curso (WIP): conocido por su acrónimo en inglés Wip Work in Progress. Nos indica el número de elementos de trabajo en un determinado momento. Se centra más en el flujo de trabajo y menos en tener ocupados a los trabajadores. Se limita el WIP para equilibrar la ocupación y asegurar que haya flujo de trabajo.
- Gestionar el flujo: Es poder terminar el trabajo de la forma más fluida y predecible posible, mientras se mantiene un ritmo sostenido.
- Hacer las políticas explícitas: Como cada día se toman decisiones importantes, es necesario tener las políticas bien definidas, que deben ser acordadas entre todas las partes involucradas, incluyendo clientes, interesados y trabajadores responsables del trabajo que está en el tablero.
- Implementar ciclos de retroalimentación: Son necesarios para una entrega coordinada y para mejorar la entrega de tu servicio. Refuerza las capacidades de aprendizaje de la organización y su evolución mediante experimentos gestionados.
- Mejorar colaborativamente, evolucionar experimentalmente: Su primer principio es “comenzamos con lo que hacemos ahora” y “acordamos la búsqueda de la mejorar a través del cambio evolutivo”. Es un método para el cambio continuo y se logra que el cambio se realice colaborativamente utilizando modelos y métodos científicos. Se diseñan experimentos en entornos donde fallar es seguro, si el resultado fue

positivo se mantiene, y si no, fácilmente se puede volver al estado anterior.

6.6.3. Normas y marcos de referencia

6.6.3.1. ISO/IEC 12207: 2017

La norma ISO/IEC 12207: Ingeniería de sistemas y software — Procesos del ciclo de vida del software, “proporciona un marco de proceso común para describir el ciclo de vida de los sistemas creados por humanos, adoptando un enfoque de Ingeniería de Software.” (ISO/IEC 12207, 2017, pág. vii). Su última versión es la 2017, que reemplaza a la versión 2008.

Se centra en definir las necesidades de las partes interesadas y la funcionalidad requerida al principio del ciclo de desarrollo, documentar los requisitos y realizar la síntesis del diseño y la validación del sistema mientras se considera el problema completo. Considera tanto las necesidades comerciales como las técnicas de todas las partes interesadas con el objetivo de proporcionar un producto de calidad que satisfaga las necesidades de los usuarios y otras partes interesadas aplicables.

Este ciclo de vida abarca desde la concepción de las ideas hasta la jubilación de un sistema. Proporciona los procesos para adquirir y suministrar sistemas. Además, este marco prevé la evaluación y mejora de los procesos del ciclo de vida.

Los procesos en esta norma, se clasifican en tres tipos: procesos principales, procesos de soporte y proceso de la organización. Cada proceso está dividido en un conjunto de actividades; cada actividad se subdivide a su vez en un conjunto de tareas.

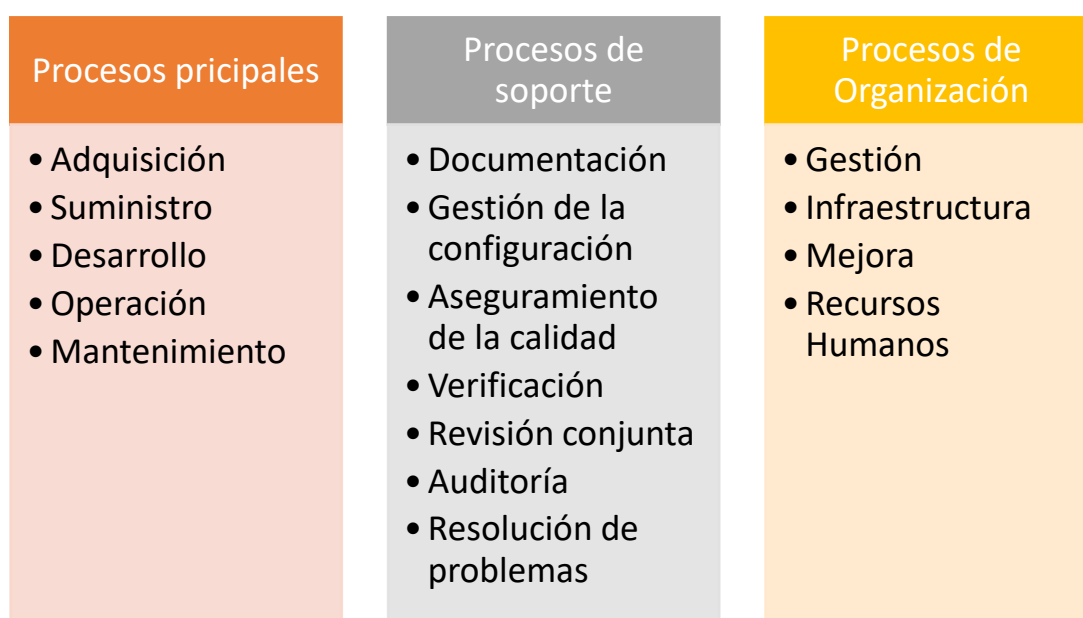
Los procesos principales son cinco procesos que dan servicio a las partes principales durante el ciclo de vida del software. Una parte principal es la que inicia o lleva a cabo el desarrollo, operación y mantenimiento de productos software.

Los procesos de apoyo al ciclo de vida son procesos que apoyan a otros procesos como parte esencial de los mismos, con un propósito bien definido, y

contribuyen al éxito y calidad del proyecto software. Un proceso de apoyo se emplea y ejecuta por otro proceso según sus necesidades.

Los procesos organizativos del ciclo de vida se emplean por una organización para establecer e implementar una infraestructura construida por procesos y personal asociado al ciclo de vida, y para mejorar continuamente esta estructura y procesos.

Figura 2: Proceso de la ISO/IECC 12007



Nota: La figura muestra los procesos contemplados en la norma ISO/IECC 12207. Fuente: Elaboración propia.

6.6.3.2. ITIL

Las siglas ITIL significan Information Technology Infrastructure Library, que traduciríamos literalmente como Biblioteca de Infraestructura de Tecnologías de Información. Según Da Silva (2021) ITIL “es una metodología que involucra una serie de buenas prácticas en la gestión de TI” (Tecnologías de la Información).



En general son un conjunto de buenas prácticas que ayudan a mejorar la prestación de un servicio de TI. Permite a las organizaciones y a los individuos ofrecer una gestión de servicios de TI rentable, alinear su gestión con la visión, la estrategia y el crecimiento de la empresa, y actuar como un punto único de contacto entre el proveedor de servicios y los usuarios finales.

En febrero del año 2019, se actualizó a la versión 4 de ITIL. Esta cuarta versión responde a los nuevos enfoques de gestión de servicios e identifica Agile, DevOps y Lean como nuevas prácticas emergentes compenetradas con ITIL.

Los procesos o etapas del ciclo de vida del servicio de ITIL son los siguientes:

- Estrategia de servicio: Facilita a las organizaciones el establecimiento de metas empresariales y el desarrollo de una estrategia que pueda satisfacer los requisitos y prioridades de los clientes.
- Diseño de servicio: Incluye el diseño de procesos y funciones. Esto se refiere tanto al diseño de los procesos, como de la tecnología, la infraestructura y los productos de la gestión del servicio.
- Transición de servicio: Se centra en mantener el estado actual del servicio mientras se implementa un nuevo cambio organizacional. Garantiza que los riesgos y los efectos estén bajo control, de manera que no haya interrupciones en ninguno de los servicios en curso.
- Operación de servicio: La operación del servicio garantiza que las tareas operacionales diarias no sean interrumpidas. Eso incluye la monitorización de la infraestructura y los servicios relacionados y permite a las empresas satisfacer los requisitos y prioridades de los clientes.
- Mejora Continua del servicio: Se trata de una parte del control de calidad que tiene como objetivo la mejora continua de los procesos de forma progresiva. Esto ocurre a lo largo del ciclo de vida del servicio.



Entre los beneficios de ITIL para las empresas se pueden mencionar, la mejora de la calidad de servicio, optimización de los costes, mayor nivel de satisfacción de los clientes, mejor gestión de riesgo y alineación de los objetivos empresariales.

6.6.3.3. COBIT

COBIT es el acrónimo de Control Objectives for Information and Related Technologies, traducido al español es Objetivos de control de la información y tecnologías relacionadas.

Es un marco de gobierno de TI para empresas que desean implementar, monitorear y mejorar las mejores prácticas de administración de TI. COBIT 2019 es la versión más reciente de este framework, en esta edición el marco se denomina «GOBIERNO DE LA INFORMACIÓN Y TECNOLOGÍA DE LAS EMPRESAS (EGIT).

Equipa a las organizaciones para prosperar, evolucionar y aprovechar mejor las oportunidades que ofrece la tecnología, proporcionando la flexibilidad para desarrollar soluciones prácticas de gobierno que se adaptan a los objetivos y el contexto de su organización.

El marco COBIT fue creado por ISACA (Asociación de Control y Auditoría de Sistemas de Información) para cerrar la brecha crucial entre los problemas técnicos, los riesgos comerciales y los requisitos de control. Se puede implementar en cualquier organización de cualquier industria para garantizar la calidad, el control y la confiabilidad de los sistemas de información.

6.7. Entornos de desarrollo y control de software

Para realizar el desarrollo de un software se emplean diferentes Entornos de Desarrollo Integrado (IDE) y algunas herramientas complementarias.

Una plataforma de desarrollo frecuentemente usada es Microsoft Visual Studio, que según Peralta (2022):



Es un entorno de desarrollo integrado, creado por la compañía Microsoft y disponible para sistemas operativos Windows, Linux y macOS, y la vez es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET. (p.1)

Este entorno de desarrollo es uno de los más usados en la institución para los diferentes sistemas que poseen. Sin embargo, también son empleados otras plataformas.

Para poder manejar las diferentes versiones de desarrollo y el trabajo en equipo se emplean las plataformas de Team Foundation o Git Hub.

El control de versiones de Team Foundation (TFVC) es “es un sistema de control de versiones centralizado.” (Microsoft, 2022). Este te permite realizar un seguimiento de los cambios realizados en el código a lo largo del tiempo. Se puede tener varias versiones del mismo proyecto y poderlas recuperar más adelante si es necesario.

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones, te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto. (Castillo, 2012). Permite a los desarrolladores administrar su proyecto, ordenando el código de cada una de las nuevas versiones que sacan de sus aplicaciones para evitar confusiones. Al tener varias versiones, no se pierden los estados anteriores cuando se va actualizar.

Permite comparar el código de un archivo para ver las diferencias entre las versiones, restaurar versiones antiguas si algo sale mal, y fusionar los cambios de distintas versiones.

Ambos softwares de control de versiones son utilizados en la institución en los diferentes entornos de desarrollo.

6.8. Terminología empleada en los procesos institucionales

Las siguientes definiciones indican el significado de palabras empleadas en la terminología en los diferentes procesos en la institución y que serán de ayuda para su comprensión.

La institución realiza un plan estratégico institucional (PEI) con vigencia de cinco años, el cual se encuentra en su página institucional. Este plan contiene “los principales desafíos que nos guiarán en la búsqueda y consolidación de una DGI moderna, respetada, calificada y al servicio del pueblo nicaragüense” (DGI, PEI 2017-2021). En este plan se contemplan los objetivos a nivel de la División de Informática y Sistemas, como parte del fortalecimiento de los principales procesos de la Administración Tributaria. Es por eso que este documento es vital para las gestiones de esta división.

En el transcurso de los procedimientos se usa cierta documentación que tiene que tiene su significado específico en la institución:

- **Bitácoras:** según Rodríguez, D. (2022), “es un cuaderno en el que se reportan los avances y resultados preliminares de un proyecto. En ella se incluyen a detalle, entre otras cosas, las observaciones, ideas, datos, avances y obstáculos en el desarrollo de las actividades que se llevan a cabo para el desarrollo del proyecto.”

En este sentido, es un documento donde se registran a detalle los acontecimientos con respecto a un proyecto. Para la institución, las bitácoras, no distante del concepto anterior, son documentos donde se detalla las inconsistencias presentadas para los procesos de los sistemas de la administración tributaria.

Inconsistencia se define como “todo aquello que no es firme, por lo tanto, la característica principal de algo cuando es inconsistente es la vulnerabilidad a variables que puedan corromper su buen funcionamiento.” (ConceptoDefinicion, 2019). Para las bitácoras, las inconsistencias representan errores o procedimientos que no se realizan correctamente en los sistemas informáticos institucionales.

Existe un formato general para quienes lo redactan. Este documento es redactado por las analistas de las administraciones de renta y las áreas normativas solicitantes, quienes son usuarios o reciben los casos presentados por los contribuyentes que tuvieron una incidencia durante algún proceso en el sistema. Luego lo remiten vía correo al área de calidad de informática para otro análisis más detallado del caso.

- **Requerimientos:** Los requerimientos “especifican qué es lo que un sistema de software debe hacer (sus funciones) y sus propiedades esenciales y deseables” (Gómez., Cervantes y González, 2019, p.28). Es decir, expresa lo que el sistema debe de realizar, su propósito.

En la institución, los requerimientos son documentos donde las áreas normativas redactan el proceso a automatizar a través de un nuevo sistema o mejora a uno existente. Estos documentos están numerados como memorándum y nomenclatura propia de cada área para su orden y control.

- **Acta de Entrega:** “es un documento donde queda de manera escrita la disposición final de una junta, reunión o acuerdo. El acta se puede hacer para muchas cosas, por ejemplo, entregas de productos, documentos, dinero, etc.” (Un documento, 2022).

Para efectos de los procesos de sistemas, este es un documento donde se detalla los objetos modificados dentro del aplicativo del sistema, ya sea formularios, esquemas o base de datos. Tienen su propia nomenclatura de numeración y son redactados por los analistas desarrolladores de software.

- **Hoja de Pruebas:** Este documento se entrega junto con el acta de entrega, y de forma general explica las pruebas a realizar a un sistema informático, y se anotan los resultados que obtuvieron los analistas quienes realizaron las pruebas indicadas. Esta hoja contiene la nomenclatura del acta de entrega a la cual está asociada e igual es redactada por el analista desarrollador de software.



6.9. Análisis FODA

Para realizar diagnósticos a las organizaciones laborales, la matriz FODA es un instrumento viable para realizar el análisis organizacional en relación a los factores que determinan el cumplimiento de las metas.

El análisis FODA, es un acrónimo de Fortalezas, Oportunidades, Debilidades y Amenazas. Según Sarli et al (2015) “consiste en realizar una evaluación de los factores fuertes y débiles que, en su conjunto, diagnostican la situación interna de una organización, así como su evaluación externa, es decir, las oportunidades y amenazas.” (p. 18).

Este análisis es aplicable para cualquier situación, individuo, producto o empresa que esté actuando como objeto de estudio. Sirve a los directivos para buscar y analizar, todas las variables que intervienen en el negocio con el fin de tomar mejores decisiones y formular las estrategias a seguir.



VII. Preguntas Directrices

- ¿Cómo es la gestión actual de procesos de los sistemas informáticos?
- ¿La gestión actual de los procesos de desarrollo de sistemas es eficiente?
- ¿Cuál es la situación actual de la gestión de los procesos de desarrollo de sistemas informáticos?
- ¿Cómo se puede mejorar la gestión de procesos de desarrollo de sistemas en base a las buenas prácticas de desarrollo de software?

VIII. Operacionalización de Variables

Tabla 2 Operacionalización de Variables

Objetivo General: Analizar la gestión de procesos de los sistemas informáticos de la División de Informática y Sistemas para el periodo 2019-2020.

Objetivos Específicos	Variable (s)	Definición Conceptual	Definición Operacional	Indicador (es)	Ítems
Describir la gestión actual de procesos de desarrollo de los sistemas informáticos	Procesos y procedimientos	<p>Según Hitpass (2017), Proceso es “una concatenación lógica de actividades que cumplen un determinado fin, a través del tiempo y lugar, impulsadas por eventos” (p. 16).</p> <p>Los procedimientos son planes que establecen un método de actuación necesario para soportar actividades futuras. (Koontz, 2012, p. 112)</p>	<p>Conjunto de procedimientos para desarrollar un sistema informático en un periodo determinado.</p> <p>Un procedimiento son las tareas que realizan los diferentes analistas para elaborar un sistema informático.</p>	<ul style="list-style-type: none"> ✓ Nombre de procesos ✓ Objetivo del proceso ✓ Cuando inicia o finaliza un proceso ✓ Elementos de entrada y salida del proceso ✓ Tareas realizadas que conforman el proceso ✓ Responsables del proceso ✓ Personas que intervienen en el proceso ✓ Documentos o registros de control asociados al proceso ✓ Indicadores de gestión que ayudan a conocer el rendimiento y desempeño ✓ Riesgos existentes si falla un proceso ✓ Metodologías empleadas ✓ Importancia de documentación 	<ul style="list-style-type: none"> ✓ Ver anexo B, Preguntas 1, 2, 4, 5, 6, 7, 8, 9, 10, 11 y 12 ✓ Ver Anexo D ✓ Ver Anexo E

<p>Medir la eficiencia de la gestión actual de procesos de desarrollo de sistemas informáticos</p>	<p>Eficiencia</p>	<p>Eficiencia “es alcanzar los fines con el mínimo de recursos” (Koontz, 2012, p. 14)</p>	<p>Desarrollar el sistema en el menor tiempo y recursos posibles</p>	<ul style="list-style-type: none"> ✓ Cumplimiento de objetivo del proceso ✓ Tiempo empleado para desarrollar los procesos ✓ Costos del proceso 	<ul style="list-style-type: none"> ✓ Ver Anexo B, pregunta 2, 3, 4 y 9.
<p>Diagnosticar la gestión actual del proceso de desarrollo de sistemas informáticos determinando sus causas y consecuencias.</p>	<p>Tipos de Causas</p> <ul style="list-style-type: none"> ➤ Causa material ➤ Causa formal ➤ Causa eficiente ➤ Causa final 	<p>Aristóteles distingue cuatro tipos de causas (<u>teoría de las cuatro causas</u>):</p> <p>a) <u>Causa material</u>: aquello de lo que está hecho algo.</p> <p>b) <u>Causa formal</u>: aquello que un objeto es.</p> <p>c) <u>Causa eficiente</u>: aquello que ha producido ese algo (explica el <<por qué>> de las cosas).</p> <p>d) <u>Causa final</u>: aquello para lo que existe ese algo, a lo cual tiende o puede llegar a ser (explica el</p>	<p>a) <u>Causa material</u>: Aquellos documentos, controles soporte y plataformas de software para gestionar los procesos de desarrollo de sistemas informáticos</p> <p>b) <u>Causa formal</u>: El producto final del proceso de desarrollo</p> <p>c) <u>Causa eficiente</u>: Aquel agente que produce un cambio, que desencadena el proceso de desarrollo.</p> <p>d) <u>Causa final</u>: La razón de haberse creado el</p>	<p>a) Causa material:</p> <ul style="list-style-type: none"> ✓ Tipos de documentos para controlar cada proceso ✓ Sistemas o herramientas para gestionar los procesos ✓ Plataformas de desarrollo de software y control de versiones <p>b) Causa formal:</p> <ul style="list-style-type: none"> ✓ Cantidad de sistemas informáticos que se han desarrollado desde la nueva dirección ✓ Mecanismo por el cual se determinó el proceso actual <p>c) Causa eficiente:</p> <ul style="list-style-type: none"> ✓ Directivos que han liderado la AT ✓ Directivos que han liderado la DIS ✓ Objetivos estratégicos de la AT a nivel de informática <p>d) Causa final:</p> <ul style="list-style-type: none"> ✓ Índice de recaudación fiscal ✓ Cantidad de procesos automatizados para la poca intervención de servidores públicos y contribuyentes 	<ul style="list-style-type: none"> ✓ Ver Anexo B, pregunta 2, 6, 7 ✓ Ver Anexo A, pregunta 1, 2, 3, 4,5, 6 y 7

		<<para qué>> de las cosas).	proceso de desarrollo de sistemas informático		
	<ul style="list-style-type: none"> ➤ Consecuencias positivas ➤ Consecuencias negativas 	Una consecuencia es aquello que resulta a causa de una circunstancia, un acto o un hecho previo.	<p>Una consecuencia positiva es el resultado esperado y efectivo de una causa.</p> <p>Una consecuencia negativa es un resultado no esperado e indeseado de una causa</p>	<ul style="list-style-type: none"> ✓ Frecuencia de actualización de documentos según necesidades ✓ Frecuencia de actualización de plataformas según necesidades ✓ Cantidad de requerimientos nuevos y modificaciones a sistemas existentes. ✓ Cantidad de bitácoras por inconsistencias en sistemas ✓ Tiempos de tareas de los procesos 	✓ Ver Anexo C: Incisos 1,2, 3 y 4

Nota: La tabla describe las variables a estudiar Fuente: Elaboración propia

IX. Diseño Metodológico

El diseño metodológico describirá de forma detallada y precisa las estrategias y procedimientos de cómo se realizó la investigación, la organización de los procesos a desarrollarse en esta investigación, definiendo qué tipo de pruebas se realizó y de qué manera se tomaron y examinaron los datos.

9.1. Tipo de Estudio

El tipo de estudio que se realizó fue descriptivo y explicativo, ya que se identificó las características de los procesos implicados en el desarrollo de sistemas y se estableció causalidades de los procedimientos realizados.

9.2. Tipo de Enfoque

En este estudio el tipo de enfoque fue cualitativo, porque se analizó la realidad en su contexto natural de los procesos de desarrollo de sistemas informáticos, interpretando el fenómeno de estudio a través de valoraciones y entrevistas realizadas a los analistas implicados.

9.3. Universo o Población

El área de estudio fue la División de Informática y sistemas de la Dirección General de Ingresos. Se enfocó en las áreas de desarrollo de sistemas tributarios que son la Unidad de Sistemas Tributarios, Unidad de Sistemas de Apoyo y Unidad de Control de Calidad, conformada por un total de 39 analistas y 6 especialistas a cargo.

9.4. Selección de Muestra

La investigación se efectuó en los procesos realizados de desarrollo de sistemas y estuvo conformada por los especialistas a cargo y directivos involucrados en cada uno de ellos.

9.5. Recopilación de Información

Para la evaluación los procesos de desarrollo de sistemas informáticos, se formularon los instrumentos de entrevistas, información estadística solicitada al área correspondiente, se elaboró guía de observación y revisión documental.

Entrevistas y cuestionarios: Se aplicó dos entrevistas dirigidas a los directores y jefes de área de la División de Informática y Sistemas para obtener detalles de cada tarea implicada. (Ver Anexo A y B).

Información estadística: Se obtuvieron datos específicos de los sistemas para verificar detalles estadísticos de cada tarea implicada. (Ver Anexo C)

Guía de Observación: Se obtuvo información a través de la observación directa del investigador para recoger datos y documentarlos. (Ver Anexo D)

Revisión documental: Se tomaron notas de la revisión de documentos para la elaboración de manuales de procedimientos dentro de la institución. (Ver Anexo E)

9.6. Procesamiento de Información

Los datos analizar son de carácter cualitativo por lo cual se realizará un análisis cualitativo descriptivo, para ello se llenaron las siguientes tablas para evaluar los datos recolectados:

Entrevista a directores de la División de Informática y Sistemas: Las preguntas de la entrevista del Anexo A, se consensuaron de la siguiente manera comparando cada columna para obtener un dato conciso:

Tabla 3: Recopilación de datos Entrevista 1

Pregunta	Jefe 1	Jefe 2
1	[Establecimiento de los procesos de desarrollo]	[Establecimiento de los procesos de desarrollo]
2	[Incidencias]	[Incidencias]
3	[Descripción de los procesos de desarrollo]	[Descripción de los procesos de desarrollo]
4	[Importancia de los procesos]	[Importancia de los procesos]
...7	[Importancia de la documentación]	[Importancia de la documentación]

Nota: Se anotaron los datos claves de cada pregunta para comparar las respuestas. Fuente: Elaboración propia.

Entrevista a jefes de área de desarrollo: Las preguntas de la entrevista del Anexo B, se consensuaron de la siguiente manera comparando cada columna para obtener un dato conciso:

Tabla 4: Recopilación de datos Entrevista 2

Pregunta	Jefe 1	Jefe 2	Jefe 3	Jefe 4
1	[Descripción de procesos]	[Descripción de procesos]	[Descripción de procesos]	[Descripción de procesos]
2	[Objetivo]	[Objetivo]	[Objetivo]	[Objetivo]
3	[Importancia de documentación]	[Importancia de documentación]	[Importancia de documentación]	[Importancia de documentación]
4	[Metodología utilizada]	[Metodología utilizada]	[Metodología utilizada]	[Metodología utilizada]
...12	[Oportunidades de mejora]	[Oportunidades de mejora]	[Oportunidades de mejora]	[Oportunidades de mejora]

Nota: Se anotarán los datos claves de cada pregunta para comparar las respuestas. Fuente: Elaboración propia.

Información estadística

Para la primera tabla solicitada del Anexo C, se cuantificó los datos obtenidos, identificando el dato que más se repite por columna solicitada (moda), y poder obtener porcentualmente datos para análisis.

El método de la **moda** que según García, J y Ramos C. (20019), “es el valor de la variable que más se repite” (p. 28). Es decir, que de todos los valores proporcionados se contará el valor que más se repita en el conjunto de datos para su debido análisis.

Los datos procesados se consensuaron con los datos obtenidos a través de entrevistas.

Tabla 5: Recopilación de datos Control de plataformas

	Lenguaje en que está desarrollado más utilizado	Plataforma de desarrollo más utilizada	Plataforma de control de versiones más utilizada	Tipo de base de datos más utilizada
	C#	Visual Studio	GIT	MySQL
100%	61.90%	61.90%	90.48%	66.67%

Nota: Se realizará cuantificación por moda y cálculo de porcentaje. Fuente: Elaboración propia.

También se cuantificaron los datos de control sobre las plataformas y requerimientos obtenidos de la segunda tabla del Anexo C, de solicitud de información estadística, con el objetivo de estimar eficiencia y cumplimiento del sistema en los últimos 3 años.

Tabla 6: Recopilación de datos Preguntas de análisis

No.	Preguntas de análisis	SI	NO
1	¿La frecuencia de actualización de los manuales es igual para todos los sistemas?		
2	¿La frecuencia de actualización de las plataformas de desarrollo se realiza a todos los sistemas?		
3	¿La cantidad de analistas asignados a los sistemas es adecuada para el desarrollo y mantenimiento de los mismos?		

No.	Preguntas de análisis	SI	NO
4	¿La cantidad de analistas es adecuada para la atención de pruebas de los requerimientos y bitácoras?		
5	¿La cantidad de requerimientos recibidos por mejoras en los sistemas ha variado incrementalmente en los últimos tres años?		
6	¿La cantidad de requerimientos completados por mejoras fue total o parcial con respecto a los recibidos?		
7	¿La cantidad de bitácoras recibidos por procesos no automatizados ha variado incrementalmente en los últimos tres años?		
8	¿La cantidad de bitácoras resueltas por procesos no automatizados fue total o parcial en los últimos tres años?		
9	¿La cantidad de bitácoras recibidas por inconsistencias del sistema ha variado incrementalmente en los últimos tres años?		
10	¿La cantidad de bitácoras resueltas por inconsistencias del sistema ha sido parcial o total con respecto a las recibidas?		

Nota: Se consensó bajo listado de chequeo de si cumple o no para su análisis. Fuente: Elaboración propia.

Con la tercera tabla requerida del Anexo C, se realizó comparación entre las cantidades de sistemas nuevos por año y estimar un aproximado de nuevos sistemas desarrollados por año.

Tabla 7: Recopilación de datos Comparación de cantidad de nuevos sistemas

Cantidad de Sistemas nuevos			Comparación	
Año 2019	Año 2020	Año 2021	2019 - 2020	2020 - 2021
			[Igual, menor o mayor]	[Igual, menor o mayor]

Nota: La tabla mostrará la diferencia entre la cantidad de sistemas nuevos. Fuente: Elaboración propia.

Guía de observación

Se obtendrá información detallada de los procesos, observando paso a paso cada detalle para la descripción de los mismos, y poder definir los procesos que actualmente se usan para el desarrollo de sistemas.

Después de ordenar los datos en tablas se realizará un análisis de Corte y clasificación, para “identificar expresiones, pasajes o segmentos que parecen importantes para el planteamiento y luego juntarlos conceptualmente” (H, Sampieri 2014, p. 439).

X. Análisis de Resultados

10.1. Descripción de la gestión actual de procesos de desarrollo de sistemas informáticos

Actualmente la División de Informática y Sistemas no emplea ninguna metodología o estándar internacional, sin embargo, sus procesos fueron definidos bajo la primera versión de las Normas Técnicas de Control Interno (NTCI), dirigidas al sector público y se adaptaron a las necesidades de la institución, transfiriendo este conocimiento a través del tiempo.

Este mecanismo se ha utilizado por más de quince años. Hace algunos años atrás, se realizó una propuesta para mejorar la organización de las oficinas y mejora en los procesos de análisis de requerimientos, creando un cargo adicional a los establecidos en ese momento, pero tras el análisis detallado de lo que implicaría, no se tenían los recursos suficientes para implementarla, por lo cual la idea fue descartada. Se evaluó la posibilidad de aplicar alguna metodología al marco de trabajo actual considerando todos los aspectos implicados para su desarrollo, pero no se ha logrado establecer.

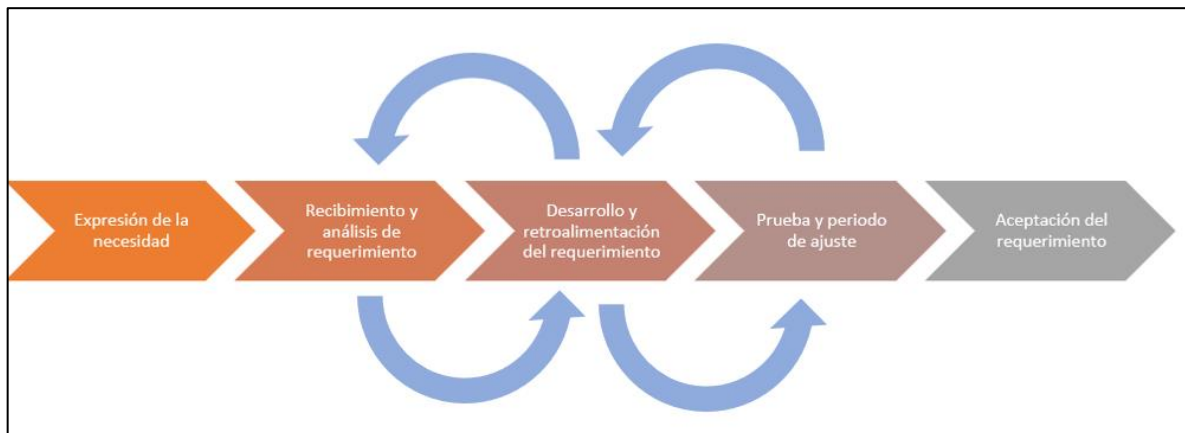
Los sistemas son realizados a la medida, debido a que los procesos en la Administración Tributaria tienen una complejidad extremadamente grande desde el vocabulario, términos utilizados hasta mantener la concordancia con la legislación y aspectos tecnológicos, es por eso que es de gran importancia a la institución para cumplir con sus objetivos y estrategias.

10.1.1. Definición de procesos

De acuerdo a los datos recopilados en entrevistas y guía de observación, se pueden definir los siguientes procesos como los actuales usados por la división de informática y sistemas para el desarrollo de los sistemas informáticos:

- Expresión de la necesidad: Esta surge por lo general, del plan estratégico institucional (PEI) y Plan Estratégico de Informática expresadas en las reuniones de directores, donde se analiza y se identifica las necesidades de sistemas o funcionalidades nuevos. Esta idea es dirigida a las direcciones específicas con el objetivo que elaboren el requerimiento de manera formal.
- Recibimiento y análisis de requerimientos: La DIS recibe el requerimiento o solicitud por el área normativa, el cual es asignado por el jefe a los analistas de desarrollo para su análisis y observaciones.
- Desarrollo y retroalimentación del requerimiento: Se va desarrollando el requerimiento mientras se realizan consultas a las áreas solicitantes en un proceso de retroalimentación para aclaración de dudas.
- Pruebas y periodo de ajuste: La Unidad de Contacto, Control y Calidad (UCC) junto con el área solicitante realizan pruebas según lo solicitado en requerimiento, donde también surgen nuevos ajustes a validaciones y errores encontrados durante las mismas.
- Aceptación del requerimiento: Una vez finalizada las pruebas tanto la UCCC como el área solicitante dan el visto bueno para su publicación en producción.

Figura 3: Proceso actual de desarrollo de sistemas informáticos



Nota: La figura muestra la secuencia actual del proceso de desarrollo de sistemas informáticos. Fuente: Elaboración propia

Los procesos descritos anteriormente únicamente describen la parte del desarrollo de sistemas informáticos antes de la publicación de su versión en producción para la disponibilidad de los usuarios finales, que este conlleva otra serie de procesos y procedimientos.

10.1.2. Descripción de procesos

Cada uno de los procesos de desarrollo contiene una serie de procedimientos que involucra elementos humanos, tecnológicos y documentos para control de los mismos. De acuerdo a lo observado y datos recopilados de entrevistas, se identificó que el desarrollo de los sistemas puede ser desencadenado por requerimientos o bitácoras, por lo cual su trato es diferente. Para su análisis se dividirán en dos grandes elementos:

- Procesos de desarrollo o ampliación de sistemas
- Procesos de corrección de errores de los sistemas existentes

En base a esto, a continuación, se detallan los procedimientos y tareas de la siguiente manera:

10.1.2.1. Procesos de desarrollo o ampliación de sistemas

Expresión de la necesidad: Este proceso surge por lo general, en reuniones de directores y jefes de áreas, donde los directores institucionales expresan la necesidad de automatizar o mejorar un proceso. Las áreas normativas o solicitantes redactan un documento donde de manera general indican lo que se necesita que realice el sistema a desarrollar para facilitar los procesos manuales. Luego se firma y sella para su autorización y después se envía a la DIS, para su desarrollo.

Los elementos involucrados en este proceso se detallan a continuación:

Tabla 8: Expresión de la Necesidad

No.	Componente	Descripción
1	Nombre del proceso	Expresión de la necesidad
2	Objetivo	Redactar un documento que refleje los requerimientos y funcionalidades que ofrecerá al usuario el sistema a desarrollar
3	Factor que indica inicio del proceso	Los directores expresan la necesidad de automatizar y mejorar procesos a las áreas normativas.
4	Especialistas involucrados	Analistas del área normativa o solicitante Jefe del área normativa o solicitante Directores de la Institución
5	Elementos de entrada	Expresión verbal de necesidad
6	Tareas realizadas que intervienen en el proceso	Redacción de documento por el analista del área, revisión por el jefe del área y autorización del documento por el director del área
7	Documento o registros de control	Se crea número de memorándum con codificación propia del área normativa o solicitante
8	Factor que indica culminación de proceso	Autorización por el director de área normativa o solicitante y envío en digital y físico a la DIS.
9	Elemento de salida	Requerimiento firmado y sellado para su envío a la DIS

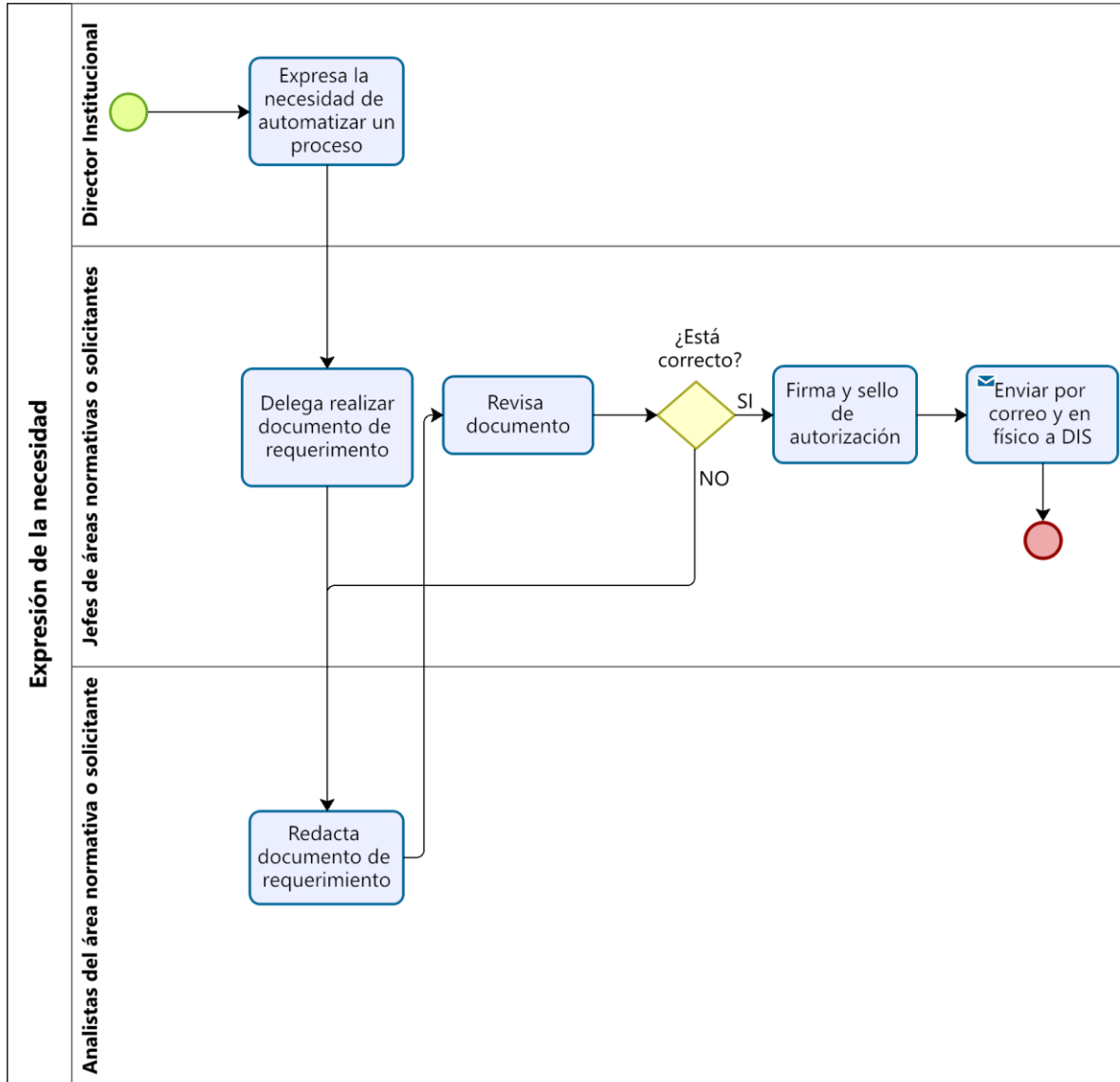
Nota: La tabla muestra los elementos que intervienen en el proceso de expresión de la necesidad. Fuente: Elaboración propia

El documento de requerimiento generado en esta etapa en un formato simple, que abarca, un número identificador del documento de memorándum con la codificación establecida por el área, los remitentes del documento, fecha de elaboración del documento, nombre del requerimiento, base legal en la cual se avala la agregación del proceso a automatizar, una descripción de lo que se quiere que el sistema realice y al final, nombre de quien elaboró el requerimiento y a quienes van sus copias.

La descripción de lo que se quiere realizar, no tiene un formato detallado en específico, por lo cual algunas áreas pueden ser muy específicas o no, y puede dificultar el análisis para la siguiente etapa.

El siguiente diagrama de flujo muestra el proceso de forma gráfica según los elementos descritos anteriormente:

Figura 4: Diagrama de Flujo Expresión de la necesidad



Nota: La figura muestra el diagrama de flujo de expresión de la necesidad.

Fuente: Elaboración propia

Recibimiento y análisis de requerimiento: El requerimiento es recibido en físico y en digital por los directivos y jefes del área de desarrollo. El jefe de área registra el requerimiento en el sistema de control de requerimientos y bitácoras, y asigna el desarrollo a los analistas a cargo.

Cuando son sistemas nuevos, los analistas son asignados de acuerdo a sus habilidades, experiencia y disponibilidad de tiempo en base a su carga laboral. Cuando son mejoras a sistemas actuales, se asigna a los analistas que están a cargo del sistema, que por lo menos son 2 en sistemas pequeños, uno a cargo y el otro de apoyo. Solo en casos especiales son más analistas en un sistema en dependencia de la extensión del desarrollo.

Los analistas, leen el requerimiento y si tienen dudas, envían mediante correo las inquietudes sobre el documento, pero avanzan a la siguiente etapa de desarrollo.

Se pueden identificar los siguientes elementos en esta etapa:

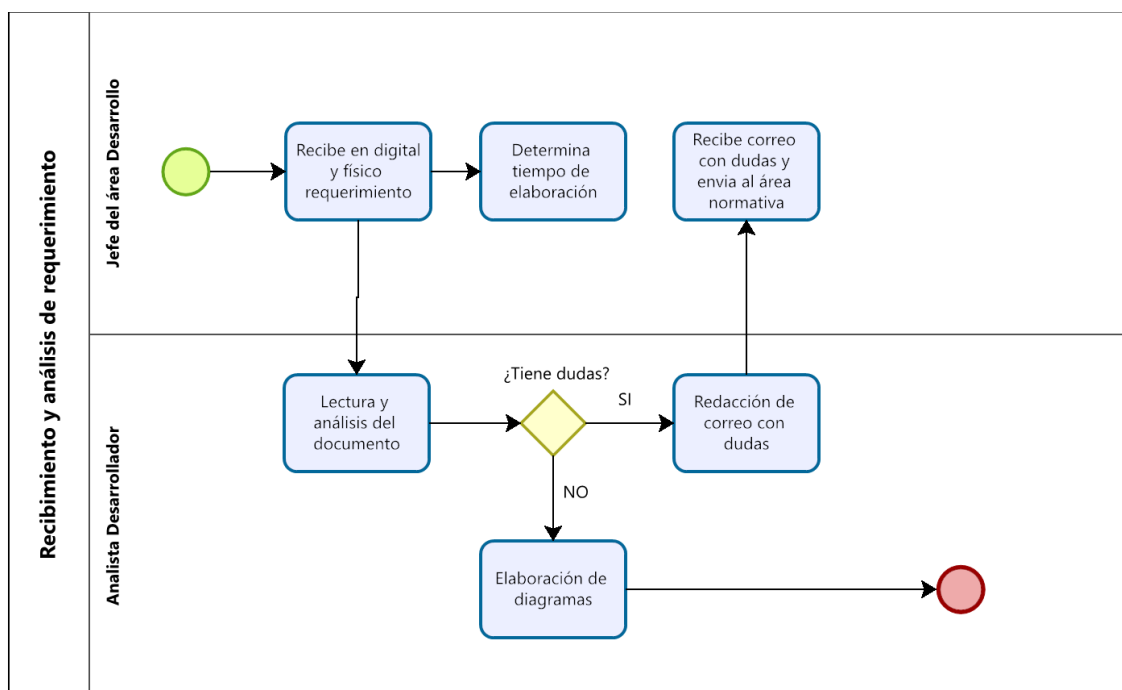
Tabla 9: Proceso de Recibimiento y Análisis de requerimiento

No.	Componente	Descripción
1	Nombre del proceso	Recibimiento y análisis de requerimiento
2	Objetivo	Determinar la funcionabilidad del sistema a desarrollar
3	Factor que indica inicio del proceso	Asignación del requerimiento a los analistas desarrolladores
4	Especialistas involucrados	Jefes a cargo Analistas desarrolladores
5	Elementos de entrada	Requerimiento en digital o físico, asignación mediante sistema de control de bitácoras y requerimientos
6	Tareas realizadas que intervienen en el proceso	El jefe del área a cargo asigna el requerimiento al analista El analista desarrollador realiza: Lectura y análisis de requerimiento Elaboración de diagramas de flujo para comprensión Elaboración de correo para dudas del documento de requerimiento El jefe y el analista estiman un tiempo para el desarrollo
8	Documento o registros de control	Registro de consultas por correo y avances por sistema de control de requerimientos y bitácoras
9	Factor que indica culminación de proceso	Continuar al siguiente paso con el desarrollo del sistema
10	Elemento de salida	Requerimiento analizado

Nota: La tabla muestra los elementos que intervienen en el proceso de recibimiento y análisis de requerimiento. Fuente: Elaboración propia

El diagrama de proceso para esta etapa de recibimiento y análisis de requerimiento es el siguiente:

Figura 5: Diagrama de flujo Recibimiento y análisis de requerimiento



Nota: La figura muestra el diagrama de flujo de recibimiento y análisis de requerimiento. Fuente: Elaboración propia

Desarrollo y retroalimentación de requerimiento: El analista desarrollador realiza las modificaciones en el código fuente del sistema, paralelamente se reciben por medio de correos las repuestas a las dudas de la etapa anterior, y estas son agregadas al desarrollo.

Algunas veces, si la situación lo amerita se realizan reuniones entre los directores de la división, los jefes de área, analistas desarrolladores y analistas del área normativa o solicitante para la aclaración de dudas. También como parte de la retroalimentación se realizan consultas vía telefónica.

El analista desarrollador por su parte no emplea ninguna metodología o estándar de desarrollo en su código fuente, ni buenas prácticas para que sea legible a otro programador de ayuda o reemplazo.

Una vez finalizada la codificación, se realiza publicación en ambiente de pruebas con ayuda de los gestores de control de código fuente. Los más usados en el área son Team Foundation y Git Hub, pero actualmente se encuentran migrando para solo usar Git Hub, debido a que es más robusto y ha demostrado que se evitan menos errores al momento de trasladar el código a proyectos productivos.

Los elementos para este proceso son los siguientes:

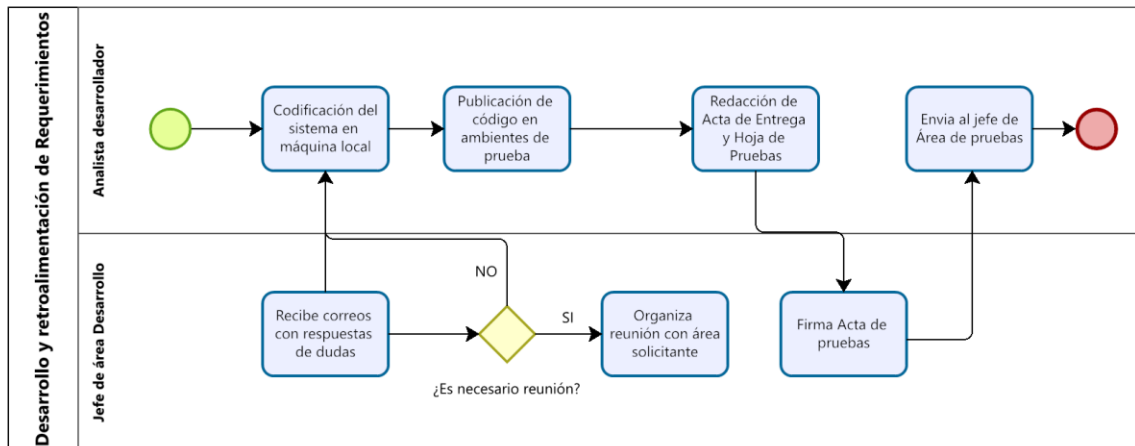
Tabla 10: Proceso de Desarrollo y retroalimentación

No.	Componente	Descripción
1	Nombre del proceso	Desarrollo y retroalimentación de requerimiento
2	Objetivo	Codificar el sistema de acuerdo a los requerimientos solicitados
3	Factor que indica inicio del proceso	Codificación del sistema en la plataforma y gestor de base de datos designado
4	Especialistas involucrados	Analista desarrollador Jefes del área Analista del área solicitante
5	Elementos de entrada	Respuestas de dudas vía correo, reuniones o llamadas telefónicas
6	Tareas realizadas que intervienen en el proceso	Codificación en su equipo local Reuniones o llamadas telefónicas para aclaración de dudas Publicación de código en ambientes de prueba Redacción de acta de entrega y hoja de pruebas
8	Documento o registros de control	Acta de Entrega y Hoja de pruebas
9	Factor que indica culminación de proceso	Entrega de los documentos de acta de entrega y hoja de pruebas al área de pruebas
10	Elemento de salida	Publicación de desarrollo en ambiente de pruebas Acta de Entrega y Hoja de pruebas

Nota: La tabla muestra los elementos que intervienen en el proceso de desarrollo y retroalimentación. Fuente: Elaboración propia

El diagrama de flujo para este proceso es:

Figura 6: Diagrama de flujo Desarrollo y retroalimentación de Requerimiento



Nota: La figura muestra el diagrama de flujo desarrollo y retroalimentación de requerimientos. Fuente: Elaboración propia

Prueba y periodo de ajuste: En esta etapa el jefe del área de pruebas recibe el acta de entrega y hoja de pruebas por parte del analista desarrollador. El jefe del área asigna el caso al analista de pruebas de acuerdo a los sistemas asignados previamente.

Durante este proceso, el analista de pruebas, va documentando cada paso realizado en el sistema, esto a través de captura de pantallas. Realiza las pruebas en ambiente designado, verifica base de datos y sistemas afines al proceso. También registra las pruebas en el sistema de control de requerimientos y bitácoras.

Si el analista de pruebas detecta alguna inconsistencia durante las pruebas, las reporta mediante correo y sistema de control de requerimientos y bitácoras; entonces es donde surge el periodo de ajuste. Los analistas de pruebas tienen que esperar a que los analistas desarrolladores, indiquen que han corregido las inconsistencias por los mismos medios reportados, para así retomar las pruebas.

También realizan pruebas el área normativa o solicitante en ambiente de pruebas, los links de ambientes para realizar las pruebas son notificados a través de correo electrónico por parte del jefe del área de desarrollo. Por el mismo medio de correo, se notifican los errores por parte de esta área normativa o solicitante.

Luego vuelve un ciclo de pruebas, reportar inconsistencias, corrección por parte del analista desarrollador hasta culminar con todo el proceso correctamente.

Adicionalmente, si el área normativa detecta durante pruebas que necesita alguna mejora al proceso y no fue contemplado en el requerimiento original, entonces lo solicita vía correo, y posteriormente se recibe un memorándum de ampliación al requerimiento con las mejoras solicitadas vía correo.

Los elementos involucrados en esta etapa son:

Tabla 11: Pruebas y periodo de ajuste

No.	Componente	Descripción
1	Nombre del proceso	Prueba y periodo de ajuste
2	Objetivo	Garantizar que el sistema esté libre de errores y funcione correctamente en ambiente productivo
3	Factor que indica inicio del proceso	Recibimiento de acta de entrega y hoja de pruebas por parte del analista desarrollador
4	Especialistas involucrados	Analista desarrollador Jefe de área de pruebas Analista de pruebas Analistas del área normativa o solicitante
5	Elementos de entrada	Acta de entrega y Hoja de pruebas
6	Tareas realizadas que intervienen en el proceso	Por parte del analista de pruebas: Revisión de hoja de pruebas y requerimiento para realizar los procesos contemplados en estos documentos Realización de pruebas en los ambientes correspondientes verificando los cambios en las bases de datos y sistemas a fines al desarrollo. Rellenar formato de hojas de pruebas Firma del formato de hoja de pruebas por parte del jefe del área una vez culminada las pruebas Por parte del área normativa o solicitante: Notificación de errores a través de correo electrónico

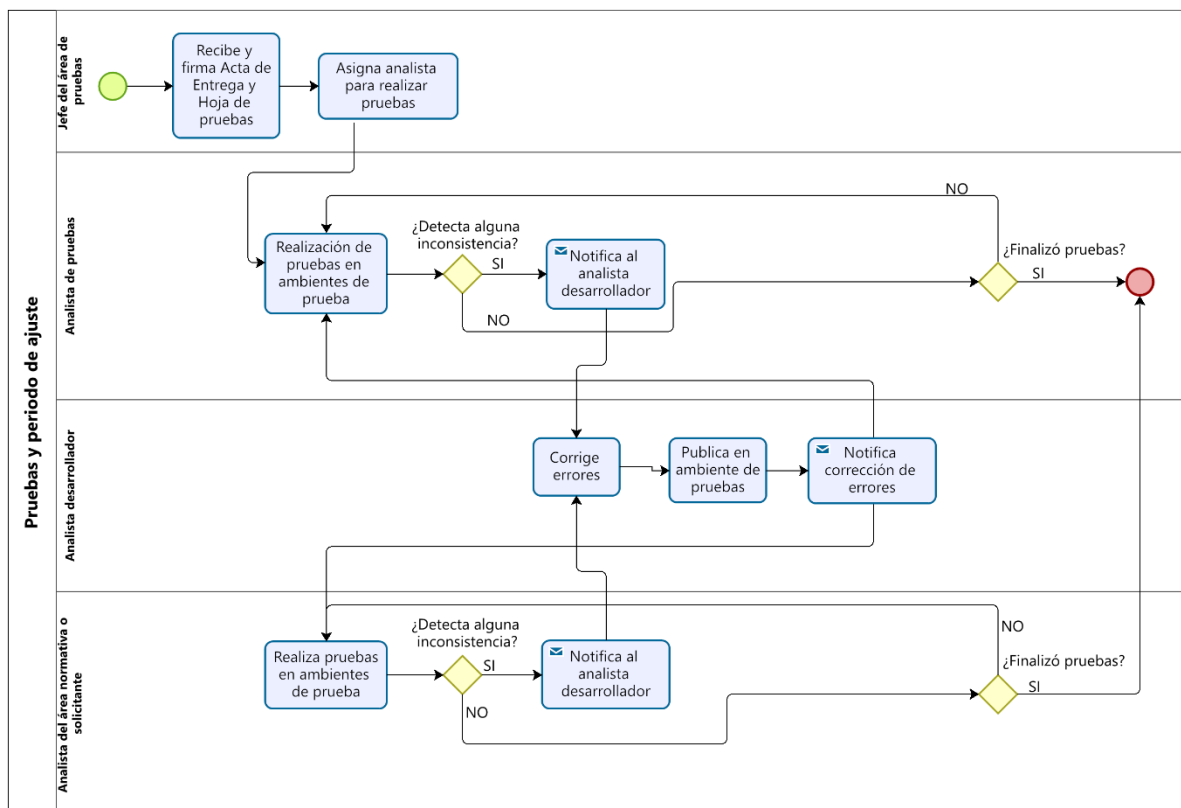
No.	Componente	Descripción
		Solicitud de algún cambio o mejora a través de correo para posteriormente ser enviado por memorándum.
8	Documento o registros de control	Hoja de pruebas Formato de Plan de Pruebas Registro en el sistema de control de requerimiento y bitácoras.
9	Factor que indica culminación de proceso	Procesos de requerimiento completados correctamente
10	Elemento de salida	Hoja de pruebas firmada por el jefe de área de pruebas remitidas al área de desarrollo

Nota: La tabla muestra los elementos que intervienen en el proceso de pruebas y periodo de ajuste. Fuente: Elaboración propia

El diagrama de flujo es el siguiente:

Nota: La figura muestra el diagrama de flujo de pruebas y periodo de ajuste.

Figura 7: Diagrama de flujo del proceso de pruebas y ajuste



Fuente: Elaboración propia

Aceptación del requerimiento: Una vez finalizada las pruebas y que los resultados fueron satisfactorios, entonces mediante correo y memorándum de aceptación, se notifica al área de desarrollo que aceptan los cambios para ser publicados en producción.

Adicionalmente, también se recibe el memorándum con mejoras al proceso, si en esa ocasión se necesitó.

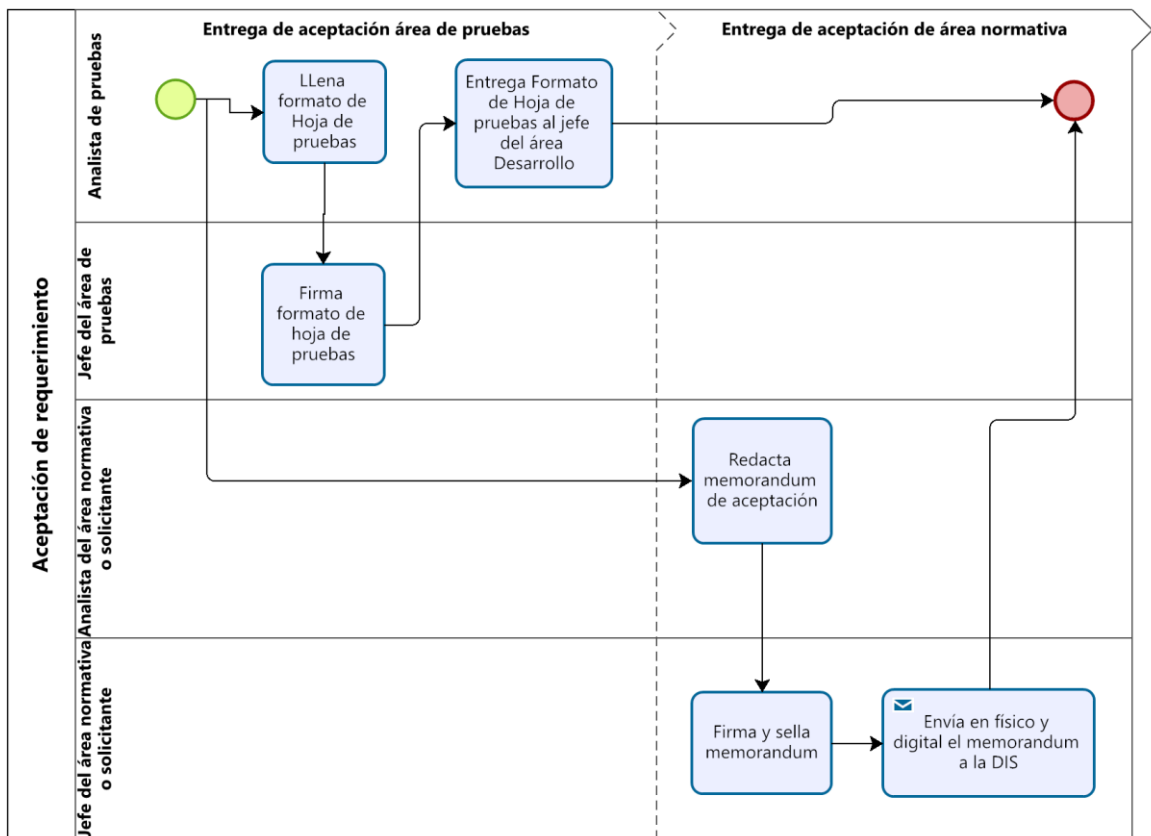
Tabla 12: Aceptación de requerimiento

No.	Componente	Descripción
1	Nombre del proceso	Aceptación del requerimiento
2	Objetivo	Documentar que las pruebas fueron satisfactorias y su aceptación para publicación en producción
3	Factor que indica inicio del proceso	Finalización de pruebas
4	Especialistas involucrados	Analista de pruebas Analista del área normativa o solicitante
5	Elementos de entrada	Hoja de pruebas Formato de Plan de Pruebas
6	Tareas realizadas que intervienen en el proceso	Redacción de documento para aceptación de requerimiento que corresponde realizarlo al área normativa o solicitante Redacción en hoja de pruebas que las pruebas fueron satisfactorias
8	Documento o registros de control	Memorándum de aceptación Hoja de pruebas Formato de Plan de Pruebas Registro en el sistema de control de requerimiento y bitácoras.
9	Factor que indica culminación de proceso	Redacción del memorándum de aceptación
10	Elemento de salida	Memorándum de aceptación

Nota: La tabla muestra los elementos que intervienen en el proceso de aceptación de requerimiento. Fuente: Elaboración propia

El diagrama de flujo para este último proceso es:

Figura 8: Diagrama de flujo Aceptación de pruebas



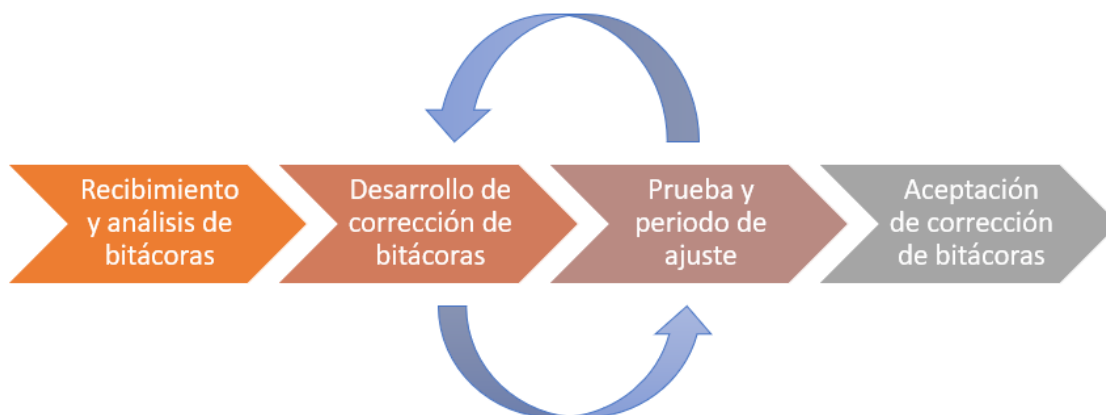
Nota: La figura muestra el diagrama de flujo de aceptación de requerimiento.

Fuente: Elaboración propia

10.1.2.2. Procesos de corrección de errores de los sistemas existentes

Algunas modificaciones al sistema no se realizan a través de requerimiento, sino a través de bitácoras, que son documentos donde comunican alguna falla que se dio en algún proceso del sistema para ser corregida. Los procesos para este tipo de casos son los siguientes:

Figura 9: Proceso Corrección de errores de los sistemas informáticos



Nota: La figura muestra la secuencia actual de los procesos de corrección de errores de los sistemas informáticos. Fuente: Elaboración propia

Recibimiento y análisis de bitácoras: Las bitácoras son recibidas por los analistas de las áreas normativas o solicitantes, quiénes interactúan directamente con los sistemas o reciben los casos reportados por los contribuyentes que son usuarios de éstos . Los elementos para este proceso son:

Tabla 13: Proceso Recibimiento y Análisis de bitácoras

No.	Componente	Descripción
1	Nombre del proceso	Recibimiento y análisis de bitácoras
2	Objetivo	Verificar falla del procedimiento del sistema
3	Factor que indica inicio del proceso	Recibimiento de bitácora a través de correo electrónico
4	Especialistas involucrados	Analistas del área normativa o solicitante Analista de pruebas
5	Elementos de entrada	Bitácora recibida por el área normativa o solicitante
6	Tareas realizadas que intervienen en el proceso	Análisis de caso presentado en bitácora Replica en ambientes de prueba para confirmación de situación presentada y verificación en la base de datos si es necesario Llenado de formato de bitácora para enviar al área de desarrollo Registro de bitácora por sistema de ticket y libro de registro de bitácoras del área.
8	Documento o registros de control	Bitácora recibida del área solicitante Libro de registro de control de bitácoras Sistema de ticket

9	Factor que indica culminación de proceso	Envío de bitácora al área de desarrollo
10	Elemento de salida	Bitácora con análisis del analista de pruebas

Nota: La tabla muestra los elementos que intervienen en la variante del proceso de recibimiento y análisis de requerimientos. Fuente: Elaboración propia

El área de pruebas realiza la réplica de los casos presentados en los ambientes de pruebas, que son sitios similares a los productivos, pero estos difieren porque contienen múltiples desarrollos de requerimientos u otras bitácoras que también están en pruebas. Esto genera que a veces se dificulte las pruebas del caso presentado ocasionando retrasos y la no verificación de funcionalidad de forma idéntica como debería de esperarse en producción.

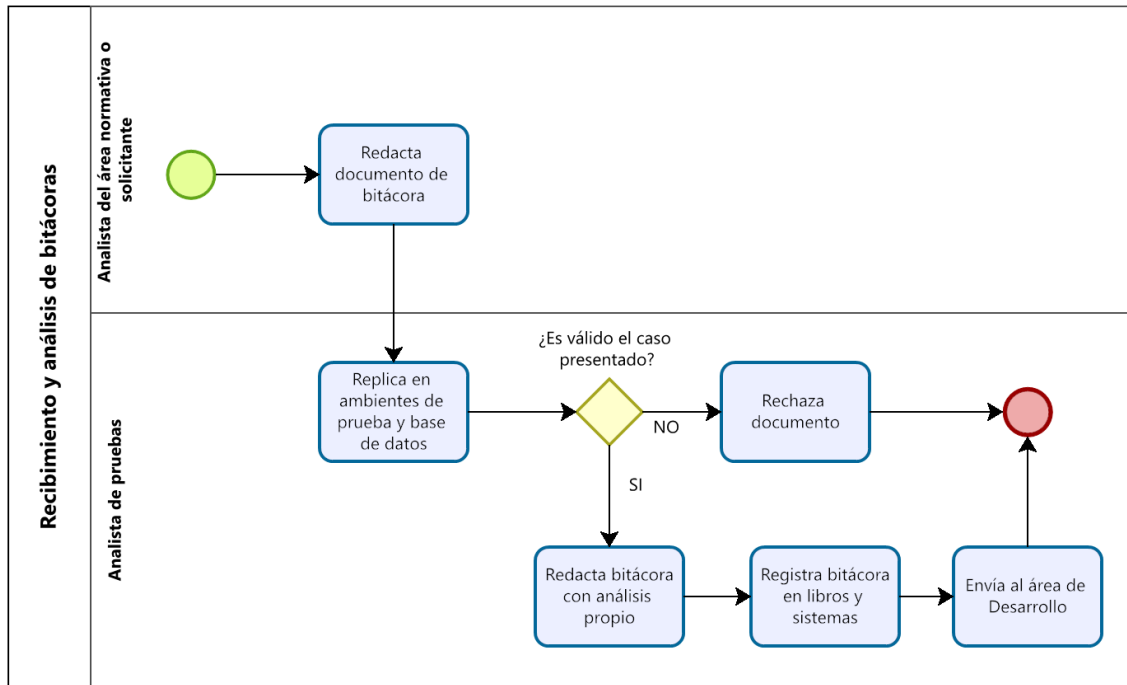
Además de estos sitios, el analista de calidad, puede verificar en las bases de datos de prueba los registros de los casos presentados por el área que reporta la inconsistencia, para verificar que se hayan guardado correctamente.

Las pruebas que realiza son de funcionalidad, y no abarcan las pruebas de rendimiento, bugs ni seguridad del sistema, como se haría en una metodología internacional o estándar.

Una vez verificado el caso y si es necesario realizar una corrección, el analista registra la bitácora en libro de registro, firma el jefe del área de desarrollo y se entrega para su asignación.

El diagrama de flujo para este proceso es el siguiente:

Figura 10: Diagrama de flujo Recibimiento y análisis de bitácoras



Nota: La figura muestra el diagrama de flujo de recibimiento y análisis de bitácoras. Fuente: Elaboración propia

Desarrollo de corrección de bitácoras: El jefe de desarrollo, asigna la bitácora al analista desarrollador que tienen a cargo el sistema que presentó el caso. Los analistas desarrolladores replican el proceso reportado con su proyecto local, verifican los datos de entrada al proceso a través del código fuente y base de datos, e identifican a qué se debe el problema presentado.

Una vez identificado, realizan la modificación en el código fuente y realizan publicación en ambiente de pruebas con ayuda del gestor de código fuente.

Posteriormente se redacta un documento de Acta de entrega con los objetos modificados en el sistema y una hoja de pruebas firmado por el jefe del área de desarrollo, donde en el formato básico solo pide ciertos elementos del proceso, hay columnas que no se ocupan nunca y que de acuerdo al detalle que el analista que lo redacta, puede ser de ayuda o no al analista de pruebas.

Los elementos para el caso de bitácoras lo podemos consensuar de la siguiente manera:

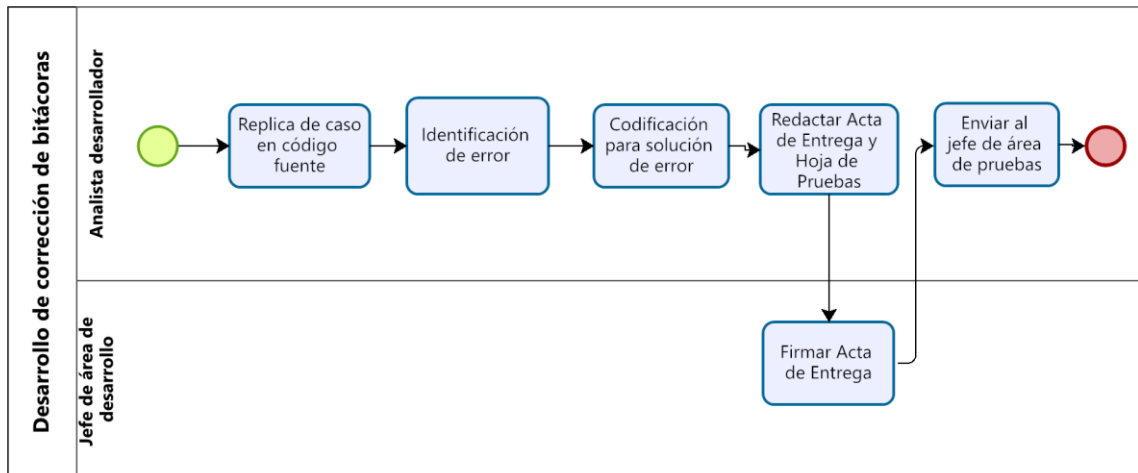
Tabla 14: Desarrollo y retroalimentación para bitácoras

No.	Componente	Descripción
1	Nombre del proceso	Desarrollo de corrección de bitácoras
2	Objetivo	Codificar el sistema para solucionar problemas presentados durante algún proceso
3	Factor que indica inicio del proceso	Codificación del sistema en la plataforma y gestor de base de datos designado
4	Especialistas involucrados	Analista desarrollador
5	Elementos de entrada	Bitácora
6	Tareas realizadas que intervienen en el proceso	Réplica del caso presentado en bitácora en código fuente y base de datos Identificación del error Codificación para solución del error Redacción de acta de entrega y hoja de pruebas.
8	Documento o registros de control	Bitácora Acta de Entrega y Hoja de pruebas
9	Factor que indica culminación de proceso	Entrega de los documentos de acta de entrega y hoja de pruebas al área de pruebas
10	Elemento de salida	Publicación de desarrollo en ambiente de pruebas Acta de Entrega y Hoja de pruebas

Nota: La tabla muestra los elementos que intervienen en el proceso de desarrollo y retroalimentación para el caso de bitácoras. Fuente: Elaboración propia

El diagrama de flujo es el siguiente:

Figura 11: Diagrama de flujo Desarrollo y retroalimentación de bitácoras



Nota: La figura muestra el diagrama de flujo desarrollo y retroalimentación de bitácoras. Fuente: Elaboración propia

Prueba y periodo de ajuste: El jefe del área de pruebas recibe el acta de entrega y hoja de pruebas por parte del analista desarrollador. El jefe del área asigna el caso al analista de pruebas que realizó la bitácora.

Al igual que las pruebas para un requerimiento, el analista de pruebas, va documentando cada paso realizado en el sistema, a través de captura de pantallas. Realiza las pruebas en ambiente designado, verifica base de datos y sistemas afines al proceso. También registra las pruebas en el sistema de control de requerimientos y bitácoras.

Si el analista de pruebas detecta alguna inconsistencia durante las pruebas, las reporta mediante correo y sistema de control de requerimientos y bitácoras; entonces es donde surge el periodo de ajuste. Los analistas de pruebas tienen que esperar a que los analistas desarrolladores, indiquen que han corregido las inconsistencias por los mismos medios reportados, para así retomar las pruebas.

También realizan pruebas el área normativa o solicitante en ambiente de pruebas, los links de ambientes para realizar las pruebas son notificados a través de correo electrónico por parte del jefe del área de desarrollo. Por el mismo medio de correo, se notifican los errores por parte de esta área normativa o solicitante.

Luego vuelve un ciclo de pruebas, reportar inconsistencias, corrección por parte del analista desarrollador hasta culminar con todo el proceso correctamente.

Los elementos involucrados en esta etapa son:

Tabla 15: Pruebas y periodo de ajuste

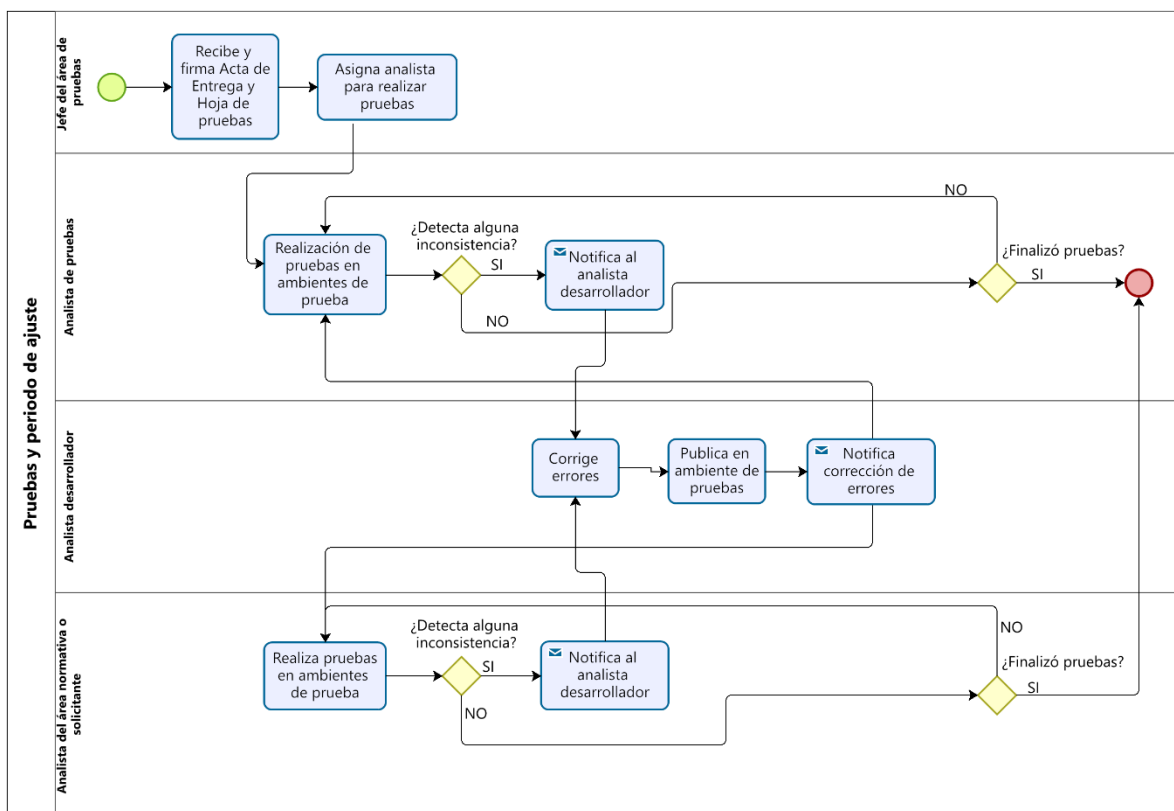
No.	Componente	Descripción
1	Nombre del proceso	Prueba y periodo de ajuste
2	Objetivo	Garantizar que el proceso reportado por la bitácora se realice correctamente
3	Factor que indica inicio del proceso	Recibimiento de acta de entrega y hoja de pruebas por parte del analista desarrollador
4	Especialistas involucrados	Analista desarrollador Jefe de área de pruebas Analista de pruebas Analistas del área normativa o solicitante
5	Elementos de entrada	Acta de entrega y Hoja de pruebas
6	Tareas realizadas que intervienen en el proceso	Por parte del analista de pruebas: Revisión de hoja de pruebas y requerimiento para realizar los procesos contemplados en estos documentos Realización de pruebas en los ambientes correspondientes verificando los cambios en las bases de datos y sistemas a fines al desarrollo. Rellenar formato de hojas de pruebas Firma del formato de hoja de pruebas por parte del jefe del área una vez culminada las pruebas Por parte del área normativa o solicitante: Realización de pruebas en los ambientes correspondientes Notificación de errores a través de correo electrónico
8	Documento o registros de control	Hoja de pruebas Formato de Plan de Pruebas Registro en el sistema de control de requerimiento y bitácoras.

No.	Componente	Descripción
9	Factor que indica culminación de proceso	Procesos de requerimiento completados correctamente
10	Elemento de salida	Hoja de pruebas firmada por el jefe de área de pruebas remitidas al área de desarrollo

Nota: La tabla muestra los elementos que intervienen en el proceso de pruebas y periodo de ajuste. Fuente: Elaboración propia

El diagrama de flujo es el siguiente:

Figura 12: Diagrama de flujo del proceso de pruebas y ajuste



Nota: La figura muestra el diagrama de flujo de pruebas y periodo de ajuste. Fuente: Elaboración propia

Aceptación de corrección de bitácoras: Una vez finalizada las pruebas y que los resultados fueron satisfactorios, el analista de pruebas, rellena la hoja de pruebas con los datos satisfactorios y lo entrega al jefe del área de pruebas y el jefe del área

de desarrollo para que ambos firmen el documento como aceptación que todo está correctamente.

Los analistas del área normativa o solicitante envían correo notificando que aceptan los cambios para ser publicado en producción.

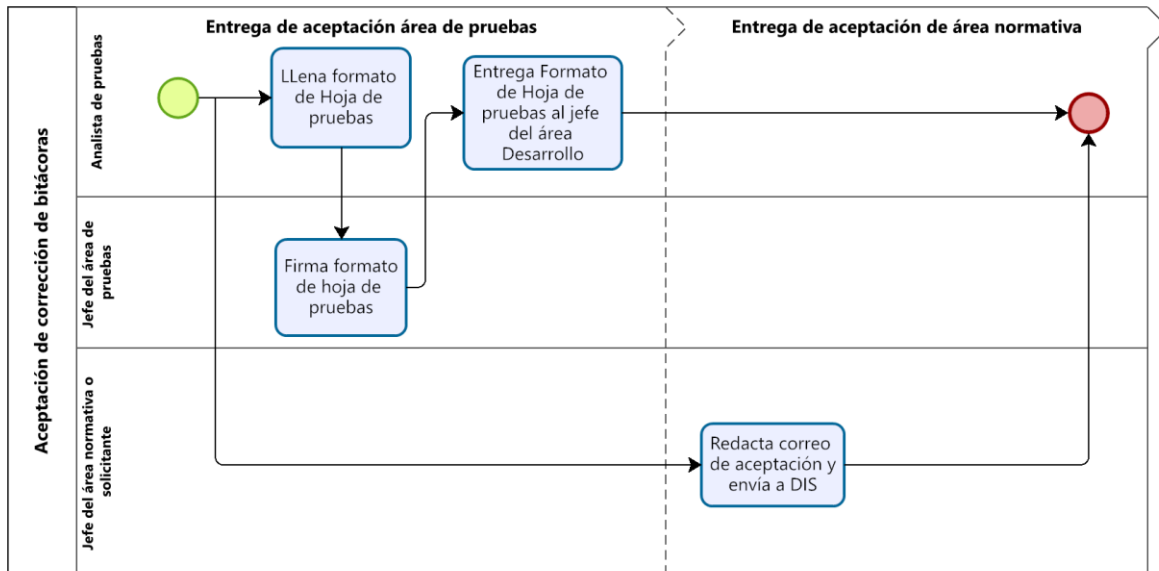
Tabla 16: Aceptación de requerimiento

No.	Componente	Descripción
1	Nombre del proceso	Aceptación de corrección de bitácora
2	Objetivo	Documentar que las pruebas fueron satisfactorias y su aceptación para publicación en producción
3	Factor que indica inicio del proceso	Finalización de pruebas
4	Especialistas involucrados	Analista de pruebas Analista del área normativa o solicitante
5	Elementos de entrada	Hoja de pruebas
6	Tareas realizadas que intervienen en el proceso	Redacción de correo para aceptación de publicación en producción que corresponde realizarlo al área normativa o solicitante Redacción en hoja de pruebas que fueron satisfactorias
8	Documento o registros de control	Correo de aceptación Hoja de pruebas Formato de Plan de Pruebas Registro en el sistema de control de requerimiento y bitácoras.
9	Factor que indica culminación de proceso	Correo de aceptación para culminación Hoja de pruebas finalizada y firmada
10	Elemento de salida	Correo de aceptación para culminación Hoja de pruebas finalizada y firmada

Nota: La tabla muestra los elementos que intervienen en el proceso de aceptación de requerimiento. Fuente: Elaboración propia

El diagrama de flujo para este proceso es:

Figura 13: Diagrama de flujo del proceso de Aceptación de corrección de bitácoras



Nota: La figura muestra el diagrama de flujo de aceptación de corrección de pruebas. Fuente: Elaboración propia

10.1.3. Documentos y sistemas de control durante los procesos

Durante los procesos se utilizan varios documentos, sin embargo, durante observación se identificó que estos no poseen muchos detalles necesarios del proceso, y cuando el analista que recibe el documento no está claro con algo de los formatos, tiene que recurrir donde el analista que lo redactó para aclarar las dudas y proceder con lo solicitado.

Para el formato de requerimiento, este posee número de requerimiento, el remitente y a quien va dirigido, fecha de realización, nombre del requerimiento, una breve descripción de lo que se quiere realizar junto con la base legal y a continuación empieza la redacción de lo que se espera que el sistema realice; sin embargo, esta parte, no está estructurada ni específica, así que a veces se puede olvidar ciertos detalles de incluir para su mejor comprensión o resultan ser ambiguos.

El formato de Hoja de pruebas, posee una columna de descripción general, donde el analista desarrollador de forma general indica los pasos a realizar según



lo desarrollado, hay una columna de verificación si ese pasó está “Bueno” o “Malo”, aunque la columna de “Malo”, nunca se utiliza, pues cuando se detecta un error, se reporta para ser corregido, y no se tacha la hoja hasta su entrega final en la columna Bueno.

También posee una columna de observaciones del analista de pruebas y firma. En la parte inferior se sitúan los espacios para firma de los jefes del área de desarrollo y área de pruebas. Este formato carece de más especificaciones que documente todos los aspectos para la realización de las mismas incluyendo las incidencias.

10.2. Eficiencia de la gestión actual de los procesos de desarrollo de sistemas informáticos

Para proporcionar valor y evitar consecuencias negativas a las partes interesadas se hace necesario evaluar la eficiencia del software.

La Eficiencia según Koontz (2012) es “alcanzar los fines con el mínimo de recursos” (p. 14). La norma ISO/IEC 25000, conocida como SQuaRE (*System and Software Quality Requirements and Evaluation*), es un marco de referencia para evaluar la calidad del software, en la división 25010, se evalúa la eficiencia de desempeño que es una “característica que representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones” (ISO 25010).

Esta característica se subdivide a su vez en las siguientes subcaracterísticas: comportamiento temporal que considera los tiempos de respuesta y procesamiento, la utilización de recursos referente a las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas y la capacidad evaluando el grado en que el software cumple con los requisitos.

Considerando lo anterior para determinar la eficiencia actual de los procesos de desarrollo, se analizó ciertas frecuencias y cantidades con respecto a manuales, plataformas, bitácoras atendidas y requerimientos realizados para su análisis.

La frecuencia de actualización de manuales es poco frecuente para todos los sistemas, con más frecuencia se ha realizado mayormente a los manuales de usuario, no así a los manuales técnicos y diccionario de datos. Lo que sugiere que todos los cambios que se han realizado en los sistemas desde la última actualización del manual, no están contemplados.

Las plataformas de desarrollo se actualizan cuando ya la tecnología está muy desfasada e impide realizar cambios al sistema que son necesarios para el correcto funcionamiento, aunque se ha realizado en algunos, otros siguen en plataformas antiguas, adaptando los controladores a las funcionalidades solicitadas.

Para los desarrollos nuevos se tratan de usar plataformas más actualizadas, siempre en coordinación con el área de base de datos para evaluar el soporte de estas nuevas tecnologías. Esto también sugiere que los sistemas se visualizan de manera distinta por los diferentes controles empleados en cada plataforma.

Para sistemas grandes que demandan mayor frecuencia de cambios los equipos de desarrolladores están conformados por varios analistas, por el contrario, sistemas más pequeños o hasta de poco uso o frecuencia de actualizaciones, son atendidos por 1 o dos analistas. Por lo tanto, podemos decir que, a pesar de ser pocos analistas por sistema, los desarrollos son atendidos de acuerdo a su magnitud.

A los analistas de pruebas cada seis meses, se les reasignan las administraciones de rentas y áreas normativas a atender, que son áreas solicitantes de respuestas de bitácoras y requerimientos.

Los analistas que están en esta área de pruebas atienden a un conjunto de estas áreas solicitantes, se considera la frecuencia que demanda cada área y que estén dos analistas por grupo de áreas. Otro factor a considerar es que con esto se trata que todos los analistas conozcan todos los procesos por si alguno llega a fallar por alguna circunstancia, alguien más pueda atender los casos que presentan estas áreas.

Cuando un analista se va de la institución, hay rotación de personal. A pesar de los esfuerzos que la Dirección ha realizado por implementar políticas para retener al personal, el hecho que se desarrollen los sistemas a la medida implica el conocimiento único del analista. Según lo mencionado en entrevistas realizadas, reponer a esa persona son seis meses aproximadamente en términos de productividad, es por eso que, para disminuir ese impacto, se trata que no exista analistas únicos en el dominio del tema.

La siguiente tabla muestra la comparación por año de los nuevos requerimientos versus los requerimientos recibidos por mejoras en los sistemas, analizados de los datos del instrumento de información estadística:

Tabla 17: Requerimientos por mejoras vs Requerimientos nuevos

Año	% Requerimientos por mejoras	% Requerimientos nuevos
2019	38.27	38.15
2020	16.05	23.33
2021	45.68	38.52
Total	100.00	100.00

Nota: La tabla muestra en porcentajes los requerimientos recibidos por mejoras y nuevos. Fuente: Elaboración propia.

La cantidad de requerimientos recibidos por mejoras, fue de 38.27% recibido en el año 2019, un 16.05% en el año 2020 y un 45.68%, durante el año 2021. Consecuentemente se puede decir que durante el año que 2020 fue un año poco común, ya que disminuyó bastante, posiblemente por causas externas, como lo es la pandemia del covid-19, que afectó a nivel internacional.

Hay que considerar que los sistemas surgen con funciones o procedimientos básicos, y en base a ley, se agregan más funciones de poco a poco para tener todos los procesos automatizados. Sin embargo, no todos los casos son así, puesto que muchos de estos requerimientos son mejoras por escenarios no contemplados durante el requerimiento original y se tienen que realizar ajustes para que se puedan gestionar estos casos a través de los sistemas.

Relacionado a la cantidad de requerimientos recibidos por mejoras a los sistemas, no todos los requerimientos son completados por diferentes razones, entre ellas, prioridades cambiantes entre los requerimientos a desarrollar, el requerimiento debido a las circunstancias actuales dejó de ser necesario para su aplicación en producción, entre otras. Algunos requerimientos son desarrollados, pero no puestos en prueba o producción. La carga laboral también impacta en las entregas. Esta cifra corresponde a un 9%.

Existen procesos que todavía no están automatizados por lo que algunos procedimientos, deben de realizarse desde el área de informática. Estas son

recibidas por bitácoras y por lo general son modificaciones en la base de datos. Algunos sistemas reciben más modificaciones que otros. Hay requerimientos que contemplan estos cambios pero que todavía no han podido publicarse en producción.

La siguiente tabla muestra las cifras correspondientes a este caso:

Tabla 18: Bitácoras recibidas por inconsistencias vs Bitácoras por procesos no automatizados

Año	% Bitácoras recibidas por inconsistencias	% Bitácoras por procesos no automatizados
2019	29.15	27.27
2021	47.77	31.45
2022	23.08	41.28
Total	100.00	100.00

Nota: La tabla muestra en porcentajes las bitácoras recibidas por inconsistencias vs las bitácoras recibidas por procesos no automatizados. Fuente: Elaboración propia.

En su totalidad, las bitácoras de procesos no automatizados son resueltas, ya que son de poco análisis y requieren de pocas modificaciones en las bases de datos. Sin embargo, consumen gran volumen de las bitácoras recibidas, que por consecuencia consumen tiempo que podría emplearse para otro tipo de desarrollos, ya que los mismos analistas desarrolladores atienden estos casos presentados por bitácoras y retrasan el desarrollo de los requerimientos recibidos.

Como se van agregando nuevos procesos en los sistemas, a veces se presentan inconsistencias, por lo que la cantidad de bitácoras recibidas por este motivo también ha incrementado. Casi todas en su totalidad se resuelven, pero quedan algunas que por diversos motivos no se resuelven o se contemplan a través de requerimientos de mejoras.

La cantidad de sistemas nuevos que se desarrollan por año incrementan año con año, siempre como parte de los objetivos estratégicos del PEI y la ayuda a los altos directivos en la toma de decisiones.

Dentro del PEI 2017-2021, hay una sección nombrada “La Tecnología como apoyo a la gestión y el fortalecimiento institucional”, y dentro de este se contempla la adición de nuevos procedimientos automatizados, mejoras a los sistemas existentes y la implementación de un sistema de gestión de TIC’S basado en las mejores prácticas y estándares de COBIT v5 para alinear las tecnologías de información a los objetivos de negocio de forma óptima.

Actualmente este último objetivo no se logró realizar, debido (según entrevista realizada a los especialistas de informática), a que se ha dado prioridad a otros planes y proyectos después del intento fallido de golpe de estado sufrido en Nicaragua durante el año 2018. Este evento tuvo un impacto en el país que afectó también en cierto grado a la institución y la obligó a replantear los objetivos y redirigir los recursos a los de mayor necesidad.

Los tiempos de desarrollo de los requerimientos van en dependencia del tipo de proyecto, pero mayormente determinado por las prioridades y necesidades del momento establecidas por la alta dirección, por lo que se puede o no cumplir con estos tiempos de desarrollo.

Las plataformas de desarrollo más usadas en los sistemas desarrollados con el 61.90% es Visual Studio .NET, se poseen las licencias y las metodologías de codificación de desarrollo de sistemas que se han venido actualizando a través de los años y es más fácil adaptarlas a estas plataformas de visual por ser robusta.

La plataforma de control de versiones que ha demostrado ser de mayor utilidad por tener varias versiones del proyecto de acuerdo a las necesidades es GIT, de la cual el 90.48% de los sistemas la utilizan.

El lenguaje de programación más utilizado por los sistemas con un 61.90%, es C#, utilizado en todos los proyectos desarrollados en la plataforma de Visual Studio. El tipo de base de datos más utilizada con un 66.67% es MySQL. Seguidamente por SQL Server con un 28.57% y 4.76% la base de datos de Natural.

Estos datos nos indican que sus plataformas, base de datos y lenguajes difieren para algunos sistemas, incluso algunos llegan a estar un poco

desactualizados. Sin embargo, aunque son pocos los diferentes, son sistemas importantes que deben interrelacionarse entre sí.

Estos sistemas se tienen que comunicar unos con otros para las distintas validaciones en los servicios tributarios que brinda la institución, pero al estar desarrollados en distintos lenguajes, se tienen que usar canales alternativos para enlazar los datos y obtener la información necesaria, lo que dificulta el desarrollo de mejoras en estos sistemas.

Con todo lo anterior descrito se puede evaluar las siguientes características brindándole una puntuación a cada uno de la siguiente manera, donde: 1: no cumple, 2: bajo, 3: aceptable o regular, 4: bueno o adecuado y 5: excelente.

En cuanto al comportamiento temporal de los sistemas que se desarrollan, considerando tiempos, a veces no se cumplen porque no se evalúan de manera exhaustiva todos los elementos que implica el desarrollo, obteniendo una calificación de 3.

Para la utilización de recursos, tienen los elementos humanos y tecnológicos, pero de igual manera la rotación de personal y carga laboral por analistas incide en la calidad del producto final de software, en este caso los sistemas desarrollados. Además, los diferentes tipos de plataformas de desarrollo influye en la comunicación entre todos los sistemas para brindar un mejor servicio tributario. La calificación será de 3.

La capacidad evaluando el grado en que se cumplen con los requisitos es de 2, ya que la cantidad de bitácoras recibidas por inconsistencias de los sistemas y la cantidad de requerimientos por mejoras por escenarios no contemplados es moderada.

La eficiencia en los sistemas es moderada a baja y se puede optimizar si se establecen mejores mecanismos para controlar el proceso de desarrollo de software.

Tabla 19: Calificación de las características para eficiencia

Aspectos a considerar	Calificación				
	1	2	3	4	5
Comportamiento temporal					
Utilización de recursos					
Capacidad					

Nota: La tabla muestra la calificación asignada a cada uno de las características en estudio para eficiencia.

10.3. Diagnóstico de la gestión actual del proceso de desarrollo de sistemas informáticos determinando sus causas y consecuencias

Para diagnosticar la situación del proceso de desarrollo de sistemas informáticos, se utilizó el análisis de la matriz FODA. Estos datos fueron obtenidos de los instrumentos de guía de observación (Anexo D) y las entrevistas realizadas a los especialistas de informática (Anexos A y B).

Tabla 20: Matriz Análisis FODA . Factores Internos

Fortalezas	Debilidades
F1. Se posee licencias y plataformas de desarrollo actualizadas	D1. Plataformas de desarrollo, control de versiones y base de datos distintas entre los sistemas dificultando la comunicación, enlace y obtención de datos de un sistema a otro
F2. Personal con dominio técnico del giro del negocio y desarrollo de sistemas	D2. Poco entendimiento de los formatos de requerimientos, bitácoras y hojas de pruebas.
F3. Los líderes de las áreas participan activamente en los procesos de desarrollo de software	D3. Ninguna metodología de desarrollo de software implementada
F4. Los analistas trabajan en equipo y bajo presión	D4. Retraso en entrega de requerimientos y resolución de bitácoras por priorizar casos del día a día.
F5. Personal altamente capacitado en maestrías, post grados, certificaciones y cursos de especialización.	D5. Omisión de pruebas en distintos escenarios
	D6. La mayoría de los procesos y cambios relacionados a los sistemas únicamente son conocidos por los analistas involucrados o están en las actas de pruebas o de instalación.

Fortalezas	Debilidades
	D7. Poca o inexistente documentación de los procesos y sistemas.

Fuente: Elaboración propia

Tabla 21: Matriz Análisis FODA - Factores Externos

Oportunidades	Amenazas
O1. Los líderes y directivos se encuentran dispuestos a incorporar una metodología que mejore los procesos de desarrollo de sistemas.	A1. Rotación de personal
O2. Implementar nuevas tecnologías que faciliten los procesos en los sistemas	A2. Desfases por tecnologías cambiantes en plataformas de desarrollo, control de versiones y base de datos.
O3. Mejorar el proceso de desarrollo de sistemas informáticos generando usuarios satisfechos	A3. Aumento de costos para capacitaciones y uso de tecnologías lo que reduce la disponibilidad del presupuesto asignado
O4. Mejorar comunicación entre todas las áreas involucradas en los procesos	A4. Resistencia al cambio para implementar las nuevas metodologías de desarrollo
O5. Cumplimiento de objetivos estratégicos institucionales en menor tiempo	
O6. Documentar los procesos de desarrollo de sistemas	

Fuente: Elaboración propia

Tras el análisis del proceso actual y su eficiencia, se puede determinar cómo problema central el descontrol y atraso del cumplimiento de desarrollo de los requerimientos y bitácoras de los sistemas informáticos.

Las causas y consecuencias se pueden sintetizar de la siguiente manera:

Tabla 22: Causas y Consecuencias del descontrol y atraso del cumplimiento de desarrollo de los requerimientos y bitácoras de los sistemas informáticos

Problema: “Descontrol y atraso del cumplimiento de desarrollo de los requerimientos y bitácoras de los sistemas informáticos”	
Causas	Consecuencias
Plataformas de desarrollo, control de versiones y bases de datos distintas entre los sistemas	Dificultad de comunicación, enlazar y obtener datos de un sistema a otro
	Visualización y funcionamiento de componentes distintos entre los sistemas

	Dificultad para implementar nuevas tecnologías
Formato de requerimiento y bitácoras poco específicos	Poco entendimiento del objetivo del desarrollo
	No desarrollar lo que el área normativa o solicitante necesitan
Formato de hojas de pruebas poco específicas	Dificultad para entender las pruebas
	Pruebas limitadas al escenario especificado
Ninguna metodología de desarrollo de software implementada	Retraso en entrega de requerimientos y resolución de bitácoras
	Constantes bitácoras de corrección de errores en el sistema
	Constantes requerimientos de mejoras a los procesos por escenarios no contemplados en el requerimiento original
	Usuarios poco satisfechos
	Documentos o sistemas de control no estandarizados, dispersos y poco precisos.
	Objetivos de PEI no cumplido para implementación de metodologías para informática.
	Tiempo de desarrollo corto
	Ambientes de pruebas muy diferentes a los productivos

Nota: La tabla muestra las causas y consecuencias del proceso actual de desarrollo de sistemas. Fuente: Elaboración propia

10.4. Propuesta para la gestión de procesos de desarrollo de sistemas informáticos en base a buenas prácticas de desarrollo de software

Considerando que el problema principal en la gestión de procesos de desarrollo de sistemas informáticos es el descontrol y atraso del cumplimiento de desarrollo de los requerimientos y bitácoras, se realizan las siguientes propuestas en cuanto a plataformas de desarrollo, formatos a emplear y metodología a implementar para optimizar los procesos y lograr mejores resultados.

10.4.1. Plataformas de desarrollo

Hay sistemas que están en plataformas que actualmente se consideran obsoletas, y que dificultan la comunicación con otros sistemas que se tienen que interrelacionar para brindar los servicios tributarios, por lo que se propone migrar y

unificar los sistemas a una plataforma de desarrollo, control de versiones y base de datos más actualizada, de forma paulatinamente por el impacto que esto tendrá.

Por ejemplo, hay sistemas que ya están en plataformas lo suficientes robustas, por lo que estos quedarían en sus plataformas y los demás se adaptarían a estas.

A continuación, se detallan en porcentajes los sistemas a migrar de una plataforma de desarrollo a otra:

Tabla 23: Plataformas de desarrollo a migrar

Porcentajes de sistemas %	de	Plataforma de desarrollo actual	Plataforma a migrar
4.76		Natural	Visual Studio 2019
33.33		PHP/APACHE	Visual Studio 2019
15.38		Visual Studio 2010	Visual Studio 2019
61.54		Visual Studio 2015	Conservar en esta plataforma
23.08		Visual Studio 2019	Conservar en esta plataforma

Nota: La tabla muestra la plataforma actual y la propuesta. Fuente: Elaboración propia

Las plataformas de Visual Studio se van actualizando a través del tiempo, por lo que se recomienda migrar estas plataformas cada 6 años a la versión actualizada.

Para la plataforma de control de versiones, los porcentajes de sistemas sería la siguiente:

Tabla 24: Plataformas de control de versiones

Porcentaje de sistemas %	Plataforma de control de versiones actual	Plataforma a migrar
4.76	Team Foundation	Git Hub
90.48	Git Hub	Conservar en esta plataforma

Nota: La tabla muestra la plataforma actual y la propuesta. Fuente: Elaboración propia

Para las bases de datos, los porcentajes de gestores quedarían de la siguiente forma:

Tabla 25: Plataforma de control de versiones

Porcentaje de sistemas %	Gestor de base de datos	Plataforma a migrar
4.76	Natural	SQL
66.67	Mysql	SQL
28.57	SQL	Conservar en esta plataforma

Nota: La tabla muestra la plataforma actual y la propuesta. Fuente: Elaboración propia

10.4.2. Formatos de requerimientos, acta de entrega, hoja de pruebas y bitácoras.

Como una de las principales causas del descontrol y atraso de los procesos de gestión de desarrollo de sistemas informáticos son los formatos usados en requerimientos, acta de entrega, hoja de pruebas y bitácoras, a continuación, se proponen estos formatos para los documentos mencionados.

La institución posee su propia guía técnica para la elaboración de documentos normativos, este tiene el objetivo de “Homogenizar la elaboración, actualización y aprobación de los diferentes documentos normativos que se emiten en la DGI” (DGI, 2021, p. 1). Este documento te indica la estructura que debe poseer los diferentes documentos normativos elaborados en la institución incluyendo los manuales de procesos, manuales de procedimientos e instructivos.

En base a esta guía se formulan las siguientes propuestas de formatos con sus respectivos manuales:

Requerimiento: El formato propuesto (Ver Anexo F) contendrá los siguientes elementos: número de requerimiento, destinatario y remitente, fecha de creación, datos generales como nombre del requerimiento, descripción y objetivo, el tipo de requerimiento, el nombre del sistema que se creará o modificará, la base legal que avala la modificación o creación, los tipos y características de los usuarios del sistema, el contexto operativo, las funciones del sistema con sus elementos mínimos necesarios y los requerimientos no funcionales. Al final indicar quien lo elaboró y las personas que tendrán una copia de éste.

Acta de Entrega: El formato propuesto para el Acta de Entrega (Ver Anexo G) contiene número de acta de entrega, destinatario y remitente, URL del ambiente de pruebas, número de remisión que originó el desarrollo, objetivo que se espera lograr, módulos o procesos afectados, objetos modificados incluyendo formularios, tablas de base de datos y script ejecutados, fecha y hora de entrega y recibido, firmas de los jefes del área de desarrollo y del área de pruebas.

Hoja de Pruebas: Este formato acompaña al acta de entrega. El formato propuesto (Ver Anexo H) consta de los apartados de:

- **Datos Generales:** Se coloca el número de acta de entrega con la cual está vinculada, el número de remisión que generó la prueba, el nombre del analista desarrollador, el nombre del analista de pruebas, la fecha y hora de inicio de pruebas, fecha y hora de finalización de pruebas, la URL del ambiente de pruebas y la base de datos a consultar de pruebas.
- **Pruebas a realizar:** Se repetirá el cuadro de pruebas propuesta por cada proceso a probar, identificándolo con un número consecutivo anteponiendo la letra P y guion, colocar el nombre del módulo o proceso, el cargo y nombre del usuario para realizar esa prueba, describir el procedimiento para esa prueba, anotar cualquier observación relevante del caso, el resultado esperado y el resultado obtenido. El campo de número de pruebas, el nombre del módulo y descripción del procedimiento tendrán que ser rellenados por el analista desarrollador, el resto de campos deberá ser rellenado por el analista de pruebas.
- **Incidencias encontradas:** Se enlistará todas las incidencias que surgieron durante pruebas, especificando la fecha de notificación, la fecha de solución, el estado en el que esté, entre pendiente o resulta y cualquier observación al respecto.
- **Pantallas de pruebas:** Adjuntar por proceso las pantallas que se realizaron durante pruebas.

- Anexos: Adjuntar cualquier documento, correo, tablas, gráfico, entre otros que ayuden sobre el entendimiento del caso.

Bitácoras: Para obtener un mayor entendimiento de los casos presentados por errores en los sistemas, es necesario detallar ciertos elementos que en formato actual no se indican, por lo tanto, el formato propuesto (Ver Anexo I) contiene los siguientes apartados:

- Datos del caso: se describe los elementos generales como el número de bitácora, fecha en que se realizó, analista que remitió el caso, el área a la que pertenece el analista, la severidad del caso presentado distinguiéndose entre crítico, mayor o menor y una breve descripción de que consiste el caso.
- Análisis y evaluación del caso: Este apartado debe de rellenarse por el analista de calidad, indicando URL del ambiente de pruebas, usuario y cargo con que realizó las pruebas, los pasos que realizó, detallar su análisis y posible solución, colocar su nombre, el día que lo revisó y el día que lo remitió a la unidad de desarrollo.
- Solución del caso: Este apartado debe ser llenado por el analista desarrollador, indicando los procedimientos implementados para la solución, su nombre, el día que lo revisó y el día que lo remitió nuevamente al área de calidad para su notificación de solución.

10.4.3. Metodología a implantar

El área de informática carece de una metodología que le ayude en la gestión de sus procesos de desarrollo de sistemas, por lo tanto, es de mucha importancia implantar una que se adecue a sus necesidades, para evitar retrasos en entregas, errores que no se subsanan y un excesivo consumo de recursos con resultados poco deseados.

Para poder seleccionar una metodología adecuada que se sea de utilidad, es necesario evaluar ciertas características. Tinoco et al., (2010), en su artículo de “Criterios de selección de metodologías de desarrollo de software” estableció

diferentes formas de seleccionar una metodología, entre ellas está la de criterios por presencia. Se toma como parámetros:

- La metodología con mayor presencia en internet
- La metodología mejor documentada
- Metodologías certificadas y con training
- Metodologías con comunidades
- Metodologías más utilizadas por empresas. Presencia empresarial.
- Metodologías más utilizadas en proyectos de software.

De este análisis concluyó que la de mayor peso es la metodología SCRUM.

Tabla 26: Selección de metodologías.

Metodología	Mayor presencia en Internet	Mejor documentación	Certificadas y con training	Comunidades	Presencia empresarial	Proyectos de software	Total
Agile Project Management (APM)	2	1	3	5	1	1	11
Dynamic Systems development methods (DSDM)	1	3	5	5	4	4	22
Scrum	5	2	5	5	5	5	27
Test Driven Development	3	4	3	2	2	2	16
Extreme Programming (XP)	4	5	3	2	3	3	19
Total	15	15	19	19	15	15	95

Fuente: Tinoco et al., (2010). Criterios de selección de metodologías de desarrollo de software

Sin embargo, dada las necesidades dentro de la institución los proyectos pueden ser desarrollos pequeños o grandes, y esta característica es muy importante al momento de seleccionar la metodología que mejor se adecue a lo que queremos lograr.

Cuando se tiene proyectos grandes que se puede visualizar e ir entregando por partes, es recomendable usar SCRUM, no obstante, no todos los casos pueden ser considerados de la misma forma, puesto que hay desarrollo donde no se puede planear ninguna semana de trabajo como se hace en SCRUM, y es necesario disminuir los tiempos de entrega a nuestros usuarios, por lo tanto, es recomendable usar KANBAN, o también una combinación de ambas “SCRUMBAN”.

Esta combinación consiste en incorporar elementos de una metodología a la otra. Si se está usando KANBAN, pero se quiere mostrar avances al usuario, entonces se puede agregar el “review” de SCRUM, y así sucesivamente dependiendo de las necesidades del proyecto.

Se recomienda usar ambas metodologías en dependencia del proyecto a realizar y así incorporar las metodologías ágiles al entorno de informática de forma óptima.

SCRUM para proyectos grandes

La metodología SCRUM fomenta que los trabajos se realicen siempre en equipo, para alcanzar los objetivos más rápidamente. Se compone de bloques de tiempos fijos de corta duración en los que se debe obtener un resultado.

Esta metodología define los siguientes roles, que en la institución serían los siguientes:

- **Product Owner:** Es la persona a cargo de la lista de trabajo pendiente del product backlog. Está conectado a las necesidades del usuario y se centra en transmitir el punto de vista del usuario a su equipo y a otros ejecutivos involucrados. En este caso se hará necesario crear un nuevo cargo que se encargue de ejercer este cargo, debe de poseer los conocimientos técnicos y del giro institucional para que se comunique con las áreas normativas o solicitante de los sistemas o mejoras a estos, y pueda transmitir todas sus necesidades en el product backlog.
- **SCRUM Master:** Es la persona que dirige los distintos eventos de SCRUM. Debe de promover las reuniones diarias de actualización y organizar las reuniones de planificación, revisión y análisis retrospectivo del sprint. Este rol el correspondería al líder del área desarrolladora, quien estaría liderando al equipo y asegurar que se lleve todo a cabo.
- **Team SCRUM:** Son los miembros que están trabajando en el sprint. Este estaría conformado por los analistas desarrolladores y pruebas.

Eventos del SCRUM

La dinámica para realizar el trabajo sería de la siguiente manera:

- Iteraciones: Cada sprint tendrá una duración máxima de 3 semanas y serán entre 8 y 12 sprint dependiendo del proyecto.
- El Product Owner se reunirá con las áreas normativas para escribir las “historias de usuario” que son los requerimientos básicos del producto, detalla los criterios de aceptación de cada una de las historias de usuario y define el mínimo producto viable de cada proyecto. Prepara el backlog.

Tabla 27: Realización de BackLog

No.	Componente	Descripción
1	Especialistas involucrados	Product Owner Analista del área normativa solicitante
2	Elementos de entrada	Historias de usuario
3	Tareas realizadas que intervienen en el proceso	Establecer los criterios de aceptación de las historias de usuarios Definir si el proyecto es viable
4	Elemento de salida	Backlog con las historias de usuario definidas por importancia
5	Tiempo	1 semana

Nota: Tabla de elementos involucrados en proceso de realización de backlog. Fuente: Elaboración propia

- Posteriormente se realiza el Sprint Planning, donde se planeará la duración de cada sprint. El Product Owner lleva elaborado el backlog, explica cada historia y sus criterios, el equipo decidirá cuántas historias son capaz de hacer durante el sprint. Se pueden ayudar del tablero visual como herramienta de KANBAN.

Tabla 28: Sprint Planning

No.	Componente	Descripción
1	Especialistas involucrados	Product Owner Scrum Master Team Scrum
2	Elementos de entrada	Product Backlog Release del producto del sprint anterior (Cuando es la segunda hasta ultima iteración)

3	Tareas realizadas que intervienen en el proceso	Planificación de tareas definiendo el sprint backlog y cronograma de actividades Actualización del product backlog Actualización de tablero visual de KANBAN
4	Elemento de salida	Sprint backlog
5	Tiempo	3 horas

Nota: Tabla de elementos involucrados en proceso de Sprint Planning.

Fuente: Elaboración propia

- Daily Stand Up: Se deberá realizar una reunión diaria de 15 a 20 minutos diarios para informar al equipo el trabajo que se está elaborando.

Responderán tres preguntas:

¿Qué hice ayer?

¿Qué voy hacer hoy?

¿Qué impedimentos tengo?

Se puede utilizar una técnica de “Burn Down chart” o gráfico de trabajo pendiente, para ver el estado del progreso del sprint.

Tabla 29: Daily Stand Up

No.	Componente	Descripción
1	Especialistas involucrados	Scrum Master Team Scrum
2	Elementos de entrada	Spring Backlog Tablero visual Kanban Historias de usuario Burn Down chart
3	Tareas realizadas que intervienen en el proceso	Descripción de tareas realizadas por los analistas desarrolladores
4	Elemento de salida	Tablero visual Kanban actualizado
5	Tiempo	15 a 20 minutos

Nota: Tabla de elementos involucrados en proceso de Daily Stand Up.

Fuente: Elaboración propia

- Sprint Review: Al finalizar el sprint el scrum, el equipo desarrollador y el scrum master presentará la versión de iteración del producto al Product Owner y las áreas solicitantes, para su debido análisis y mejoras.

Tabla 30: Sprint Review

No.	Componente	Descripción
1	Especialistas involucrados	Scrum Master Team Scrum Product Owner Áreas solicitantes
2	Elementos de entrada	Release del producto construido en el sprint
3	Tareas realizadas que intervienen en el proceso	Presentación del producto finalizado para ser puesto en producción
4	Elemento de salida	Revisión, aprobación, publicación en producción.
5	Tiempo	1 a 2 horas

Nota: Tabla de elementos involucrados en proceso de Sprint Review.

Fuente: Elaboración propia

Hay que tener en cuenta que cada finalización de sprint conlleva a una publicación en producción. Las metodologías ágiles tratan de realizar el producto en partes pequeñas que sean funcionales y que se puedan publicar en producción.

- Retrospectiva: El Team scrum, junto con el scrum master y el Product Owner, se reúnen para analizar lo que hicieron durante el sprint y mejorar aquellos elementos que consideren necesarios, así el siguiente sprint se espera sea más optimizado.

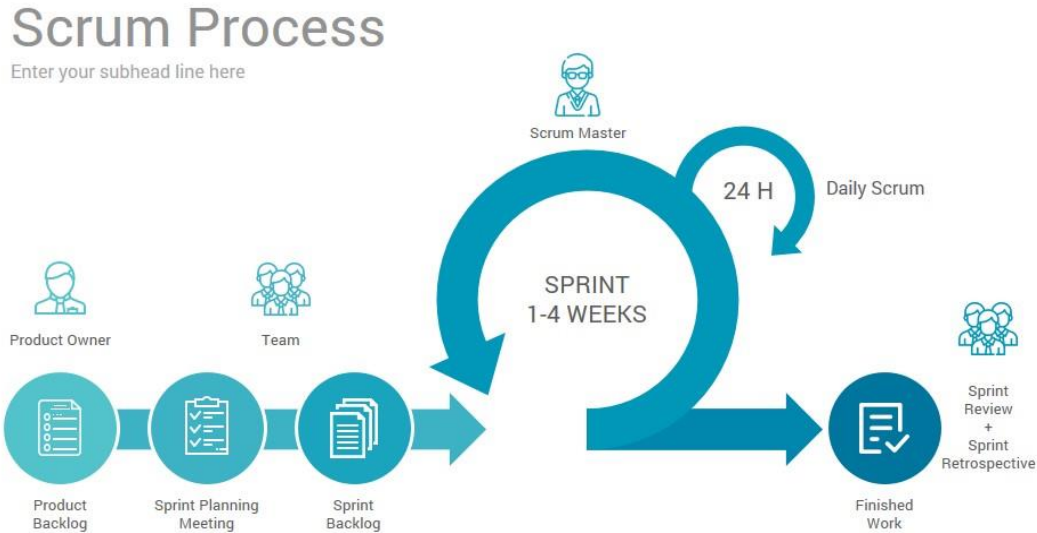
Tabla 31: Retrospectiva

No.	Componente	Descripción
1	Especialistas involucrados	Scrum Master Team Scrum Product Owner
2	Elementos de entrada	Release del producto construido en el sprint Product backlog
3	Tareas realizadas que intervienen en el proceso	Revisión del cumplimiento del producto backlog Revisión del tablero visual Burn down chart
4	Elemento de salida	Revisión, aprobación, publicación en producción.
5	Tiempo	1 a 2 horas

Nota: Tabla de elementos involucrados en proceso Retrospectiva. Fuente: Elaboración propia

La siguiente imagen abordada por Fechina, A. describe el proceso de SCRUM:

Figura 14: Metodología SCRUM



Nota: La figura muestra el proceso de SCRUM. Fuente: Fechina, A. (2018)

KANBAN para proyectos pequeños

Para desarrollos pequeños que se perciba que tendrá una duración corta de una semana o menos tiempo, se recomienda usar KANBAN. Esta metodología es bastante flexible, todos los cambios se pueden ingresar de acuerdo a su prioridad y manejarlas a través de flujos acumulativos por los estados. Puede ser uno o varias analistas los que desarrollen las tareas. Esta metodología es adaptativa, no se necesita muchas reglas para llevarlo al nivel que se necesita. A diferencia de SCRUM que tiene más reglas, no se pueden ingresar cambios hasta en el siguiente sprint y tiene que ser un equipo para realizarlo.

Para la implementación de KANBAN, se han identificado los siguientes estados que estarán en el tablero, que será mis columnas:

- Por Hacer: Son todas las tareas según las historias de usuario o requerimientos que se deban de realizar
- Haciendo: Serán todas las tareas que en ese momento se estén ejecutando

- Probando: Son todas las tareas que estén pruebas
- Hecho: Todas aquellas que se han completado

Para definir las filas del tablero se debe de considerar lo que se va a realizar, se puede iniciar sin filas, o agrupar según la necesidad, ya sea por analistas desarrolladores, o prioridad de tareas, las filas son adaptativas de acuerdo a lo que se quiera visualizar.

Se deben de establecer mínimos y máximos por columna para evitar cuellos de botella. Para la columna de “Haciendo” el máximo de tareas debe de ser de 5 y para la columna de probando será de 6 tareas. La columna “Por hacer” será variable, según la cantidad de analistas que estén desarrollando y será evaluado si se puede aceptar más elementos, se propone un máximo de 10 tareas. Esto ayudará a tener más eficiencia, optimización y productividad.

Tabla 32: Tablero Visual KANBAN

Máximo 10	Mínimo 4 y Máximo 5		
Por hacer	Haciendo	Probando	Hecho

Nota: La figura muestra la dinámica de un tablero de KANBAN. Fuente: Elaboración propia.

Se usará el diagrama de flujo acumulado, que se evaluará de forma diaria. Se construye cuantificando la cantidad de tareas que hay en cada columna, y se grafican por día. La interpretación de esta gráfica ayudará a tomar acciones lo antes posible.

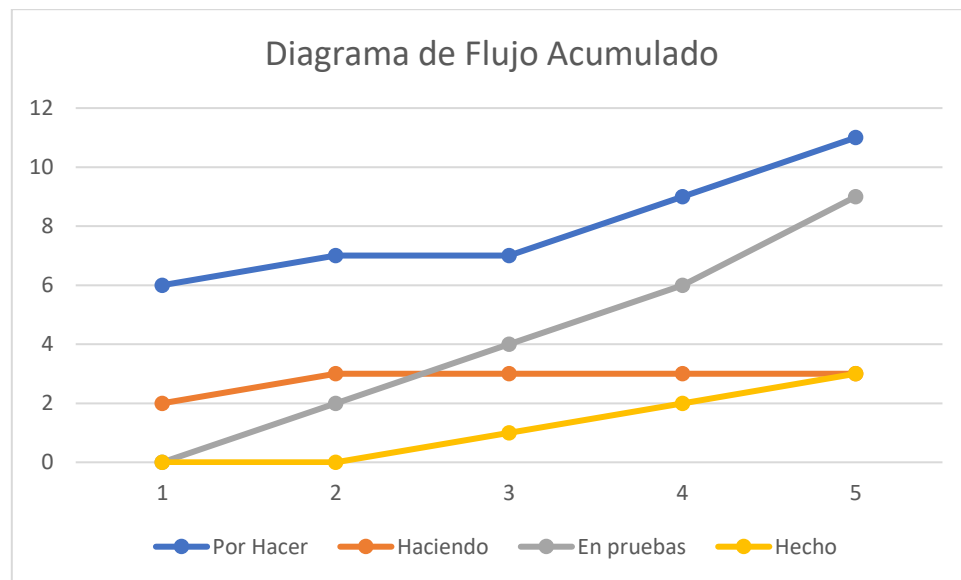
Ejemplo:

Tabla 33: Tabla para elaborar diagrama de flujo acumulado

Día	Por Hacer	Haciendo	En pruebas	Hecho
1	6	2	0	0
2	7	3	2	0
3	7	3	4	1
4	9	3	6	2
5	11	3	9	3

Nota: La tabla muestra cómo se debe de rellenar los campos para elaborar diagrama de flujo acumulado. Fuente: Elaboración propia

Gráfico 1 Diagrama de Flujo Acumulado KANBAN:



Nota: La figura muestra el diagrama de flujo acumulado según ejemplo. Fuente: Elaboración propia

Las líneas deberían de ir casi juntas, pero en este ejemplo se puede observar una gran separación entre haciendo y pruebas, detectando un cuello de botella en la columna de pruebas, que no está fluyendo a la columna de “Hecho”. Lo mismo pasa con la columna “Por Hacer”, hay muchos trabajos que aún no se pueden hacer, ya que en promedio solo se pueden estar haciendo 3 tareas y no hay capacidad

para más. Entonces con esta información se puede centrar en gestionar lo que esté pasando en esas columnas y darle solución lo más pronto posible.

Kanban deberá utilizarse para pequeños desarrollos, como son algunas mejoras en los proyectos existentes o nuevas funcionalidades menores, ya que se adapta a su marco de trabajo actual, pero les dará una herramienta de control sobre las tareas que se están realizando en el momento y tratar de realizarlas todas lo más antes posible. Para proyectos de tamaños intermedios, se puede usar una combinación de ambas metodologías, “SCRUMBAN”, para obtener todos los beneficios de una metodología ágil.

Para incorporar las metodologías ágiles mencionadas, se debe de realizar de forma paulatina, ya que implica grandes cambios en la cultura organizacional. Se puede iniciar implantando KANBAN ya que es más adaptativa, al marco de trabajo actual e ir incorporando elementos de la metodología SCRUM para así de forma gradual adaptarse y familiarizarse a esta metodología. Paralelamente a esta incorporación gradual, utilizar los formatos realizados en este acápite para minimizar los errores en los procesos que se usan.

Para conocer a manera detallada cada procedimiento según el tipo de proyecto a desarrollar, se ha redactado un manual de procedimientos, según formato institucional, con los pasos a realizar para el desarrollo de Software (Ver Anexo J). Este contiene de manera específica la implementación de las metodologías SCRUM y KANBAN, y las buenas prácticas para desarrollar software.

XI. Conclusiones

El primer objetivo de esta tesis es describir la gestión actual de proceso de desarrollo de los sistemas informáticos, del cual podemos concluir, que se logró establecer los elementos involucrados en el proceso actual y visualizarlos de forma gráfica a través de diagramas de flujo. Se detectó que los procesos no están sujetos a ninguna metodología y se establecieron según las NTCl en su primera versión, que contenía a grandes rasgos los controles mínimos a tener dentro de los procesos de las tecnologías de Información además de asemejarse a una metodología tradicional.

Para el segundo objetivo, se planteó medir la eficiencia de la gestión actual de procesos de desarrollo de sistemas informáticos, donde se caracterizó tras los parámetros de comportamiento temporal, utilización de recursos y capacidad según la eficiencia de desempeño de software de la ISO/IEC 25000 división 25010. Este análisis determinó que la eficiencia es moderada a baja.

El tercer objetivo se diagnosticó la situación actual a través de FODA y se determinó las causas y consecuencias de la gestión actual del proceso de desarrollo de los sistemas informáticos, donde se identificó tres causas principales ligadas a múltiples consecuencias extraídas de la descripción de todos los procesos actuales. Entre las causas se identifican las diferentes plataformas en las que están desarrollados los sistemas, los formatos que usan para las tareas que no son explícitos y ninguna metodología para gestionarlos y sea de mayor calidad los productos de software desarrollados.

Como último objetivo, se diseñó una propuesta de la gestión de procesos de desarrollo de sistemas informáticos considerando las causas identificadas del objetivo anterior. Se realizó una propuesta de actualización y unificación de plataformas de los sistemas tributarios para que ayude a simplificar las tareas y pueda adaptarse a nuevas tecnologías y metodologías. Se diseñó manuales de instructivos de llenado de formatos para los documentos de requerimientos, acta de entrega, hoja de pruebas y bitácoras.

Como parte final, se realizó la propuesta de implementar una metodología para el desarrollo de los sistemas. Se consideró que los tipos de requerimientos que llegan pueden ser proyectos grandes o pequeños, por lo tanto, no se puede utilizar una sola metodología. Para grandes proyectos que se pueda subdividir en partes funcionales se recomienda utilizar la metodología SCRUM, y para proyectos pequeños, como son algunas mejoras en los sistemas existentes, incorporar KANBAN en la metodología actual.

XII. Recomendaciones

Se recomienda usar herramientas de software específicas de implementación de SCRUM y KANBAN para poder incorporar las metodologías ágiles al marco de trabajo actual y sea más fácil y rápido documentar estos procesos y sean de utilidad.

Para darle seguimiento a la eficiencia, una vez implementadas las metodologías y el uso de los nuevos formatos, se recomienda evaluar de forma periódica los parámetros según la ISO/IECC 25000 u otro metodología o marco de referencia, para dar un monitoreo de los resultados obtenidos.

Si se llega a detectar aún algunos fallas o problemas para los sistemas, se recomienda siempre evaluar las causas y consecuencias para actualizar las analizadas en este estudio, y lograr tomar medidas al respecto e identificar si se logró implementar correctamente las metodologías sugeridas.

Para saber el impacto organizacional y de recursos del uso de las metodologías propuestas, se recomienda realizar un estudio de viabilidad de ambas metodologías a mayor detalle.

XIII. Referencias

- Alonso, G. (2017). Propuesta para la aplicación de una metodología de desarrollo e implementación de software en la División de Informática y Sistemas de la Dirección General de Ingresos. [Documento proyecto de grado, Universidad Nacional de Ingeniería]. Repositorio SIBIUNI. <http://ribuni.uni.edu.ni/1949/1/91611.PDF>
- Asamblea Nacional de la República de Nicaragua (1995). Normas Técnicas de Control Interno. https://www.academia.edu/19729319/Normas_Tecnica_de_Control_Interno_NTCl_Nicaragua
- Asamblea Nacional de la República de Nicaragua (2015, 14 de abril). Normas Técnicas de Control Interno. La Gaceta, Diario Oficial N°. 67 <http://legislacion.asamblea.gob.ni/normaweb.nsf/3133c0d121ea3897062568a1005e0f89/752bf68fc53a51b6062584ff0052248a>
- Canós, J et al (2005). Metodologías ágiles en el desarrollo de software. http://fcaenlinea1.unam.mx/anexos/1728/Unidad_1/u1_act2_1.pdf
- Castillo, L. (2012). ¿Qué es GitHub? Conociendo GitHub. <https://conociendogithub.readthedocs.io/en/latest/data/introduccion/#:~:text=GitHub%20es%20una%20plataforma%20de%20desarrollo%20colaborativo%20de,de%20forma%20privada%2C%20creando%20una%20cuenta%20de%20pago.>
- Concepto definicion.de, Redacción. (2019). Definición de Inconsistencia. <https://concepto definicion.de/inconsistencia/>
- Da Silva, Douglas. Web (2021). ¿Qué es ITIL y para qué sirve? Análisis detallado. Web Content / SEO associate, LATAM. <https://www.zendesk.com.mx/blog/itil-que-es-para-que-sirve/>
- Decreto No. 20-2003. Texto Consolidado De Reformas E Incorporaciones Al Reglamento De La Ley Creadora De La Dirección General De Servicios Aduaneros Y De Reforma A La Ley Creadora De La Dirección General De Ingresos, Ley N°. 339. (2022, 21 de abril). La Gaceta, Diario Oficial N°. 72.

<http://legislacion.asamblea.gob.ni/normaweb.nsf/b92aeea87dac762406257265005d21f7/443c0dfb29645bd7062588b8007a5f67?OpenDocument>

- Dirección General de Ingresos (2017). Plan Estratégico Institucional 2017-2021. <https://www.dgi.gob.ni/pdfInfo/PlanEstrategico>
- Dirección General de Ingresos, División Estratégica, Oficina de Organizaciones y métodos. (Marzo, 2021). Guía Técnica para la Elaboración de documentos normativos.
- Fechina, A. (2018). Metodología SCRUM, ¿Qué es? <https://winred.es/management/metodologia-scrum-que-es/gmx-niv116-con24594.htm>
- García, R y Ramos, C. (2009). Estadística Empresarial. Servicio de publicaciones Universidad de Cádiz.
- Gómez, M. C., Cervantes, J. y Gonzáles P. P. (2019). *Fundamentos de Ingeniería de Software*. Universidad Autónoma Metropolitana. http://www.cua.uam.mx/pdfs/conoce/libroselec/Fundamentos_Ing_SW-VF.pdf
- Herrera, H. (2013). Conceptos fundamentales de Aristóteles. <https://www3.gobiernodecanarias.org/medusa/ecoblog/johergon/2013/05/09/conceptos-fundamentales-de-aristoteles/>
- Instituto Nacional de Tecnologías. (2009). *Ingeniería del software: Metodologías y ciclos de vida*. https://www.academia.edu/9795641/INGENIER%C3%8DA_DEL_SOFTWA RE_METODOLOG%C3%8DAS_Y_CICLOS_DE_VIDA_Laboratorio_Nacional_de_Calidad_del_Software
- ISO IEC 25010 (s.f.). Eficiencia de desempeño. <https://iso25000.com/index.php/normas-iso-25000/iso-25010/21-eficiencia-de-desempeno>
- Koontz, H., Weihrich, H. y Cannice, M. (2012). Administración una perspectiva global y empresarial. Decimocuarta edición. Editorial Mc Graw Hill.

https://www.soy502.com/sites/default/files/administracion_una_perspectiva_global_y_empresarial_14_edi_koontz.pdf

- Laboratorio Nacional de Calidad del Software de INTECO (2009). Ingeniería de Software: Metodologías y ciclos de vida.
- Ley N° 339 Ley Creadora de La Dirección General de Servicios Aduaneros y de Reforma a la Ley Creadora de la Dirección General de Ingresos. (2000, 09 de marzo). Asamblea Nacional de la República de Nicaragua. Diario Oficial N°. 69 del 6 de abril de 2000. <http://legislacion.asamblea.gob.ni/normaweb.nsf/b92aaea87dac762406257265005d21f7/362c48eeb82ff5cf0625720200550a02#:~:text=LEY%20N%C2%B0.%20339%2C%20aprobada%20el%209%20de%20marzo,DE%20NICARAGUA%20Hace%20saber%20al%20pueblo%20nicarag%C3%BCense%20que%3A>
- Martins, J. (2022). ¿Qué es la metodología Kanban y cómo funciona? <https://asana.com/es/resources/what-is-kanban>
- Peralta, K. (2022). Concepto de Microsoft Visual Studio, ¿Qué es y para qué sirve Microsoft Visual Studio? <https://www.espaciohonduras.net/microsoft-visual-studio-concepto-y-que-es-y-para-que-sirve-microsoft-visual-studio>
- Project Management Institute (2017). Aumento de las tasas de Éxito. Pulse of the Profession. https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf?v=58a71b4b-ad6d-470b-8a06-34d7d66c9454&sc_lang=temp=es-ES
- Rioseco, L. (2016). ANÁLISIS Y EVALUACIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE PARA EL ÁREA DE CONSULTORÍA DE LA EMPRESA POWERDATA CHILE. [Documento proyecto de grado, Universidad de Chile]. Repositorio Uchile. <http://repositorio.uchile.cl/bitstream/handle/2250/144486/An%C3%A1lisis-y-evaluaci%C3%B3n-del-proceso-de-desarrollo-de-software-para-el-%C3%A1rea-de-consultor%C3%ADa-de-la-Empresa-PowerData-Chile.pdf?sequence=1&isAllowed=y>

- Rodríguez, D. (2022), ¿Qué es una bitácora de trabajo? El Consejo Salvador. <https://elconsejosalvador.com/trabajo/que-es-una-bitacora-de-trabajo-2.html>
- Santibañez, D. (2017). Diseño e implementación de una metodología de desarrollo ágil en una PYME TIC carente de metodología de desarrollo. [Documento de grado, Universidad de Chile]. Repositorio USM.CH. <https://repositorio.usm.cl/bitstream/handle/11673/23007/3560900231959UTFSM.pdf?sequence=1&isAllowed=y>
- Sarli, R, González S & Ayres, N. (2015). Análisis FODA. Una herramienta necesaria. Facultad de odontología, UNCuyo. Volumen 9. No. 1. https://bdigital.uncuyo.edu.ar/objetos_digitales/7320/sarlirfo-912015.pdf
- Significados.com. "Consecuencia". en: <https://www.significados.com/consecuencia/>
- Tinoco et al., (2010). Criterios de selección de metodologías de desarrollo de software. Revista de la Facultad de Ingeniería Industrial 13(1): 70-74 (2010) UNMSM. <https://www.redalyc.org/articulo.oa?id=81619984009>
- Un Documento (2022). Cómo hacer un acta de entrega de documentos. <https://undocumento.com/legal/como-hacer-un-acta-de-entrega-de-documentos/>

XIV. Anexos

14.1. Anexo A: Entrevista dirigida a especialistas a cargo en informática

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, MANAGUA

UNAN-MANAGUA

FACULTAD DE CIENCIAS ECONÓMICAS

CENTRO DE DESARROLLO EMPRESARIAL, PROCOMIN



MAESTRÍA EN ADMINISTRACIÓN FUNCIONAL DE EMPRESAS

Entrevista 1 dirigida a especialistas a cargo en Informática y Sistemas

Fecha: ____/____/____ Hora Inicio: _____ Hora Fin: _____

La presente entrevista tiene como objetivo recopilar información para el estudio “**Análisis de gestión de procesos de desarrollo de sistemas informáticos de la División de Informática y Sistemas para el periodo 2019-2020**” y optar al título de Máster en Administración Funcional de Empresas. Los datos proporcionados por el especialista de informática, tendrán carácter de confiabilidad, servirán para indagar sobre las causas del proceso de desarrollo de los sistemas informáticos. De ello, se presentarán los resultados, conclusiones y recomendaciones para la mejora de los procesos. Agradezco de antemano por todo el apoyo que pueda brindar a la investigación.

1. ¿Cómo se establecieron los procesos de desarrollo de sistemas informáticos?
2. ¿De qué forma ha incidido la rotación de personal en los procesos de desarrollo de sistemas en la AT?
3. Explique los procesos de desarrollo de sistemas que se emplean actualmente.
4. De acuerdo a su conocimiento ¿Cuál es la importancia de desarrollar e implementar sistemas informáticos en la AT?
5. ¿En qué manera los sistemas informáticos ayudan a cumplir los objetivos estratégicos de la Administración Tributaria?
6. De acuerdo a su experiencia, ¿Qué oportunidades de mejora existen en los procesos realizados actualmente?
7. ¿Cuál es la importancia de contar con los procesos documentados?

14.2. Anexo B: Entrevista dirigida a especialistas de área de informática

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, MANAGUA

UNAN-MANAGUA

FACULTAD DE CIENCIAS ECONÓMICAS

CENTRO DE DESARROLLO EMPRESARIAL, PROCOMIN



MAESTRÍA EN ADMINISTRACIÓN FUNCIONAL DE EMPRESAS

Entrevista dirigida a especialistas de área de informática y sistemas

Fecha: ____/____/____ Hora Inicio: _____ Hora Fin: _____

La presente entrevista tiene como objetivo recopilar información para el estudio **“Análisis de gestión de procesos de desarrollo de sistemas informáticos de la División de Informática y Sistemas para el periodo 2019-2020”** y optar al título de Máster en Administración Funcional de Empresas. Los datos proporcionados por el especialista de informática, tendrán carácter de confiabilidad, servirán para conocer los procedimientos que se realizan en el proceso de desarrollo de sistemas informáticos. De ello, se presentarán los resultados, conclusiones y recomendaciones para la mejora de los procesos. Agradezco de antemano por todo el apoyo que pueda brindar a la investigación.

1. De acuerdo a sus conocimientos, ¿Cuáles son los procesos que se realizan para desarrollar los sistemas informáticos?
2. ¿Cuál es el objetivo de cada uno de los procesos mencionados?
3. ¿Cuál es la importancia de la utilización de documentación tales como: acta de entrega de remisión y hoja de pruebas, que se utiliza para realizar cada tarea?
4. ¿Cuál es la metodología utilizada y que factores son tomados en consideración para establecer el tiempo de realización cada tarea?

5. ¿Cómo son seleccionados los responsables de cada uno de los procesos?
6. ¿Qué sistemas o herramientas usan para controlar los procesos?
7. ¿Cuáles de las plataformas usadas para desarrollar los sistemas ha dado resultados más satisfactorios en cuanto a complejidad, eficiencia y control?
8. ¿Qué software de control de versiones utilizado durante el desarrollo de sistemas ha dado mejores resultados para obtener los cambios a través del tiempo?
9. ¿Qué indicadores utilizan para medir el rendimiento o desempeño para estos procesos?
10. ¿Cuáles son los planes de contingencia para anticiparse ante riesgos existentes durante el fallo de alguno de los procesos?
11. ¿Cuál es la importancia de tener los procesos documentados?
12. De acuerdo a su experiencia, ¿Qué oportunidades de mejora existen en los procesos realizados actualmente?

14.3. Anexo C: Requerimiento de Información Estadística

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, MANAGUA
UNAN-MANAGUA



FACULTAD DE CIENCIAS ECONÓMICAS
CENTRO DE DESARROLLO EMPRESARIAL, PROCOMIN
MAESTRÍA EN ADMINISTRACIÓN FUNCIONAL DE EMPRESAS

Requerimiento de Información Estadística

Fecha: ____/____/____

El presente requerimiento de información estadística tiene como objetivo recopilar información para el estudio **“Análisis de gestión de procesos de desarrollo de sistemas informáticos de la División de Informática y Sistemas para el periodo 2019-2020”** y optar al título de Máster en Administración Funcional de Empresas. Los datos proporcionados por el especialista de informática, tendrán carácter de confiabilidad, servirán para conocer datos estadísticos de los procesos de desarrollo de sistemas informáticos. De ello, se presentarán los resultados, conclusiones y recomendaciones para la mejora de los procesos. Agradezco de antemano por todo el apoyo que pueda brindar a la investigación.



Llene las tablas con los datos indicados:

1. Datos técnicos de los sistemas:

Nombre del Sistema	Lenguaje en que está desarrollado	Plataforma de desarrollo	Plataforma de control de versiones	Tipo de base de datos

2. Datos de control sobre las plataformas y requerimientos recibidos

Nombre del Sistema	Frecuencia de actualización de manuales	Frecuencia de actualización de plataformas de desarrollo	Cantidad de analistas asignados al desarrollo del sistema	Cantidad de analistas asignados a las pruebas del sistema	Cantidad de requerimientos recibidos por mejoras en el año 2019, 2020 y 2021	Cantidad de requerimientos completados por mejoras en el año 2019, 2020 y 2021	Cantidad de bitácoras recibidas por procesos no automatizados en los años 2019, 2020 y 2021	Cantidad de bitácoras resueltas por procesos no automatizados en los años 2019, 2020 y 2021	Cantidad de bitácoras recibidas por inconsistencias del sistema en los años 2019, 2020 y 2021	Cantidad de bitácoras resueltas por inconsistencias del sistema en los años 2019, 2020 y 2021
					2019 2020 2021	2019 2020 2021	2019 2020 2021	2019 2020 2021	2019 2020 2021	2019 2020 2021

3. Indique la cantidad de requerimientos por sistemas nuevos en los últimos 3 años

Cantidad de Sistemas nuevos		
Año 2019	Año 2020	Año 2021



4. Variación de desarrollos de sistemas a través del tiempo

Rango de tiempo	Cantidad de sistemas informáticos desarrollados
2000-2006	
2007-2011	
2012-2017	
2018-2022	

14.4. Anexo D: Guía de Observación

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, MANAGUA
UNAN-MANAGUA
FACULTAD DE CIENCIAS ECONÓMICAS
CENTRO DE DESARROLLO EMPRESARIAL, PROCOMIN
MAESTRÍA EN ADMINISTRACIÓN FUNCIONAL DE EMPRESAS



Guía de observación

La presente guía de observación tiene como objetivo detallar los procedimientos empleados para el desarrollo de un sistema que se utilizará en el estudio **“Análisis de gestión de procesos de desarrollo de sistemas informáticos de la División de Informática y Sistemas para el periodo 2019-2020”** y optar al título de Máster en Administración Funcional de Empresas. Los datos proporcionados por el especialista de informática, tendrán carácter de confiabilidad, servirán para conocer datos estadísticos de los procesos de desarrollo de sistemas informáticos. De ello, se presentarán los resultados, conclusiones y recomendaciones para la mejora de los procesos. Agradezco de antemano por todo el apoyo que pueda brindar a la investigación.

Fecha Inicio: ____/____/____ **Fecha Fin:** ____/____/____

Codificación utilizar	a	Proceso
A-UST		Recibimiento de requerimiento
A-UST		Análisis y desarrollo de requerimiento
B-USAT		Recibimiento de requerimiento
B-USAT		Análisis y desarrollo de requerimiento
C-UCC		Recibimiento de pruebas por requerimiento o bitácoras
C-UCC		Gestión de bitácoras

ítem	Descripción	Check	Observaciones
1	Dependencia Encargada		
2	Proceso		
3	Factor que lo desencadena		
3.1	Requerimiento de sistema		
3.2	Bitácora		
3.3	Correo electrónico		
3.4	Llamada telefónica		
3.5	Otro		
4	Especialistas involucrados		
4.1	Especialista desarrollador		
4.2	Especialista de pruebas		
4.3	Especialista a cargo de la unidad		
5	Descripción de proceso observado		
5.1	Revisión del documento		
5.2	Reunión previa al desarrollo		
5.3	Asignación de tareas		
5.4	Realización de tareas		
5.5	Firma de documentos		
5.6	Registro de datos en sistemas o controles		
5.7	Uso de todos los campos del formato utilizado		
5.8	Es detallada la información de los formatos		
6	Factor que indica culminación de proceso		
6.1	Entrega de documento formal		
6.2	Envío de correo		
7	Cumplimiento del procedimiento	SI	

14.5. Anexo E: Guía de Revisión Documental

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, MANAGUA UNAN-MANAGUA



FACULTAD DE CIENCIAS ECONÓMICAS CENTRO DE DESARROLLO EMPRESARIAL, PROCOMIN MAESTRÍA EN ADMINISTRACIÓN FUNCIONAL DE EMPRESAS Guía de Revisión Documental

La presente guía de revisión documental tiene como objetivo recopilar información que se utilizará para el estudio “**Análisis de gestión de procesos de desarrollo de sistemas informáticos de la División de Informática y Sistemas para el periodo 2019-2020**” y optar al título de Máster en Administración Funcional de Empresas. Los datos proporcionados por el especialista de informática, tendrán carácter de confiabilidad, servirán para conocer datos estadísticos de los procesos de desarrollo de sistemas informáticos. De ello, se presentarán los resultados, conclusiones y recomendaciones para la mejora de los procesos. Agradezco de antemano por todo el apoyo que pueda brindar a la investigación.

Fecha Inicio: ____ / ____ / ____ **Fecha Fin:** ____ / ____ / ____

Documento	Datos a obtener
Guía técnica para la Elaboración de documentos normativos	Formato para elaborar manual de procesos en la institución
Plan Estratégico Institucional (PEI) anterior al periodo 2017-2011	Objetivos dirigidos para el área de informática
Ley 339 Ley Creadora De La Dirección General De Servicios Aduaneros y De Reforma a la Ley Creadora de la Dirección General de Ingresos	Artículos 2, 3,5, 7, 10
Decreto Ejecutivo No. 20-2003 De Reformas e Incorporaciones al Reglamento de la Ley Creadora de La Dirección General de Servicios Aduaneros y de Reforma a la Ley Creadora de la Dirección General De Ingresos, Ley N°. 339	Artículo 18

Guía técnica para la Elaboración de documentos normativos			
No.	Pregunta	Si cumple	No cumple
1	¿Se define el objetivo del documento?		
2	¿Describe las reglas básicas para redactar el documento?		
3	¿Contiene la descripción de la portada para el documento a elaborar?		
4	¿Se indica los elementos a contener cada parte del documento?		
5	¿Contiene el procedimiento administrativo para la aprobación y autorización del documento?		
6	¿Ejemplifica formatos de instructivos?		

Plan Estratégico Institucional (PEI)			
No.	Pregunta	PEI 2017-2021	PEI 2012-2016
1	¿Contiene estrategias para el área de informática referente a la mejora de procesos de desarrollo de software?		
2	¿Describe el objetivo de la estrategia?		
3	¿Describe cómo se va a realizar?		
4	¿Se cumplió la estrategia de manera parcial o completa?		

Ley 339			
No.	Pregunta	Capítulo	Artículo
1	¿Qué artículo define la naturaleza de la DGI?	I	2
2	¿Qué artículo establece los objetivos la DGI?	I	3
3	¿Qué artículo describe las funciones de la DGI?	I	6
4	¿Qué artículo determina la jurisdicción de la DGI?	I	7
5	¿Qué artículo especifica la organización de la DGI?	II	10



Decreto Ejecutivo No. 20-2003

No.	Pregunta	Capítulo	Artículo
1	¿Qué artículo establece la estructura organizacional y funcional para la División de Informática?	II	18



14.6. Anexo F: Instructivo para llenado de formato de requerimientos informáticos.

Logo de
gobierno

Logo
institucional

Dirección General de Ingresos

División de Informática y Sistemas

DIS

Instructivo para llenado de formato de requerimientos
informáticos

(Versión 1)

Octubre, 2022

I. Introducción

Como parte de los procesos de desarrollo dentro de la División de Informática y Sistemas, es necesario poseer los formatos idóneos para poder realizar correctamente cada paso que conlleve a un desarrollo eficaz y que minimicen los errores.

El formato de requerimiento de sistemas informáticos es el documento que “especifica qué es lo que un sistema de software debe hacer (sus funciones) y sus propiedades esenciales y deseables” (Gómez., Cervantes y González, 2019, p.28).

Siendo el primer paso para desarrollar un sistema el levantamiento de los requerimientos para elaborar un sistema, el presente instructivo describe el proceso y se indica el formato para este fin.

II. Objetivo

- Instruir sobre el llenado de formato para redactar requerimientos de sistemas informáticos por las áreas normativas y/o solicitantes

III. Aspecto Legal:

Decreto Ejecutivo N°. 20-2003 Reformas e Incorporaciones al reglamento de la Ley Creadora de la Dirección General de Servicios Aduaneros y de Reforma a la Ley Creadora de la Dirección General de Ingresos, ley no. 339, se establece en su artículo 22, División de Información y Sistemas:

- Realizar en forma técnica y especializada, el servicio de apoyo del procesamiento automático de la Administración Tributaria a través del Sistema de Información Tributaria (SIT).
- Recomendar nuevas aplicaciones tecnológicas de Hardware y Software y las formas de su implantación y evaluar permanentemente el estado de los sistemas de información, equipos y software, normas y procedimientos de desarrollo, instalación, mantenimiento, operación y producción.

IV. Descripción del Procedimiento

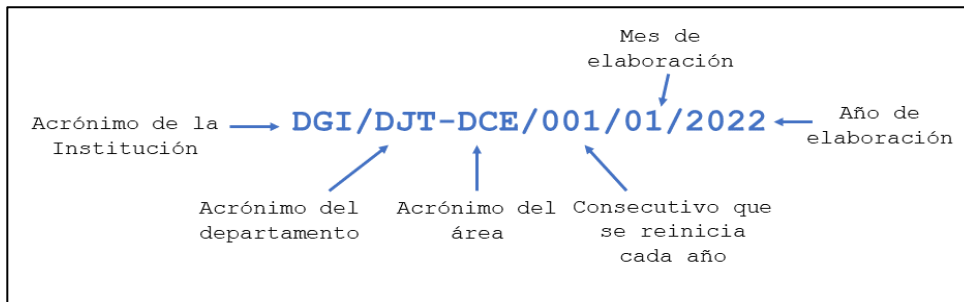
A continuación, se detalla cómo se debe de rellenar cada campo del formato especificado en el inciso V de este instructivo:

4.1. Número de requerimiento: Se debe de colocar en negrita el número de requerimiento con la nomenclatura DGI/ Acrónimo del departamento - acrónimo del área /Consecutivo/Mes/Año en curso.

El consecutivo debe de iniciar en 001 y debe de reiniciarse cada año.

El Mes debe ser el mes que se está elaborando el requerimiento y debe constar de dos dígitos entre los meses de enero a septiembre colocar un número cero adelante.

Ejemplo:



Fuente: Elaboración propia

- 4.2. **Para:** Destinatario a quién va dirigido el requerimiento, se debe de colocar nombre y cargo.
- 4.3. **De:** Remitente quien dirige el documento, se debe colocar nombre y cargo.
- 4.4. **Fecha:** Fecha de elaboración del documento.
- 4.5. **Nombre del requerimiento:** Breve referencia del requerimiento
- 4.6. **Descripción:** Resumen en un párrafo en que consiste el contenido del requerimiento
- 4.7. **Objetivo:** Redactar el para qué del requerimiento, qué se quiere lograr con el requerimiento. Debe de ser de la siguiente manera:

Specific - **E**specífico

Measurable - **M**edible

Attainable - **A**lcanzable

Relevant - **R**elevante

Time based - **T**emporal

Específico: Tiene que ser lo más específico posible

Medible: Tiene que tener una forma de medirse

Alcanzable: Que se pueda construir con el alcanzable que se está definiendo.

Relevante: Que sea importante automatizar este proyecto

Time based: En cuanto tiempo espera obtener este resultado.

4.8. Tipo de requerimiento: Marcar con una "X", la casilla del tipo de requerimiento que se está elaborando. "Nuevo Sistema" corresponde cuando no existe el sistema a desarrollar, "Ampliación a sistema existente" indica que se va anexar un nuevo proceso o módulo y "Mejora a proceso de sistema existente" se usará cuando se va agregar nuevas validaciones o mejoras en un proceso actual del sistema.

4.9. Nombre del sistema: Se colocará el acrónimo del sistema y su nombre completo.

4.10. Base Legal: Especificar nombre, número de ley y artículos en los que se avala el desarrollo de modificaciones o creación del sistema.

4.11. Tipos y características de usuario: Se describirán los tipos de usuarios que usarán el sistema, especificando su rol dentro de este, privilegios de acceso y frecuencia de uso.

4.12. Contexto Operativo: Se describirá el contexto donde operará el software, especificando Plataformas de hardware, sistemas operativos, otros sistemas con los que interactuará.

Ejemplo:

- El sistema debe ser accesible en los horarios de trabajo definidos por la institución.
- Cada usuario posee su usuario y contraseña previamente establecida.

- El sistema obtendrá datos de los usuarios provenientes de "x" sistema.
- Se espera un uso de 200 usuarios por día.
- Los usuarios cuentan con Computadoras Personales estándar o dispositivo móvil con capacidades de navegación por sistemas web

4.13. Funciones del software: Se debe detallar en este apartado los siguientes elementos mínimos a contener:

4.13.1. Identificador: Es el número de la función a desarrollar, con la nomenclatura de estructura: letra F seguido de serie por tres dígitos, empezando por el número 001 hasta llegar a todos las funciones requeridas.

4.13.2. Nombre del proceso: Breve referencia del proceso a describir.

4.13.3. Tipo de usuario: Indicar el tipo de usuario que realizará las acciones dentro de esta función.

4.13.4. Datos de entrada: Especificar todos los datos que se deben de digitar o cargar al sistema.

4.13.5. Descripción del proceso: Detallar cada paso del proceso y si es necesario adjuntar una tablas, imágenes o diagramas para la explicación.

4.13.6. Datos de salida: Indicar qué se espera que el sistema muestre o realice al ejecutar la acción.

4.13.7. Validaciones: Enumerar las validaciones pertinentes a esta función.

4.13.8. Criterios de aceptación: Especificar qué condición indica que resultado exitoso el proceso descrito.

4.13.9. Prioridad: Se indicará si la prioridad de esta función o proceso o es alta, media o baja.

- 4.13.10. **Diagramas de flujo:** Se colocarán los diagramas correspondientes al proceso descrito.
- 4.14. **Requerimientos no funcionales:** describen una restricción o limitación sobre el sistema. Restringen los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, el tipo de proceso de desarrollo a utilizar, fiabilidad, tiempo de respuesta, capacidad de almacenamiento. Agrupar según sea necesario por disponibilidad, integridad, interoperabilidad, desempeño o seguridad de la información.
- 4.15. **Elaborado por:** Indicar quien elaboró el requerimiento, colocar iniciales, cargo y correo institucional
- 4.16. **CC:** Colocar nombre y cargos de las personas a quienes se les enviará copia de este requerimiento, por ejemplos jefes inmediatos de los analistas de desarrollo y pruebas.

V. Formato e instructivo

Número del requerimiento: (DGI/Acrónimo del departamento - acrónimo del área/Consecutivo/Mes/Año en curso)

Para: (Nombre - Cargo)

De: (Nombre - Cargo)

Fecha: (Fecha de elaboración del requerimiento)

Datos Generales	
Nombre del requerimiento:	(Breve referencia del requerimiento)
Descripción:	(Resumen en un párrafo)
Objetivo:	(Objetivo general del requerimiento)

Especificaciones	
Tipo de requerimiento	<input checked="" type="checkbox"/> Nuevo Sistema
	<input type="checkbox"/> Ampliación a sistema existente
	<input type="checkbox"/> Mejora a proceso de sistema existente
Nombre del sistema	(Acrónimo y nombre completo del sistema)
Base Legal	Nombre y No. De Ley, artículos.

Contenido:

1. Tipos y características de usuario

(Describe los tipos de usuario que utilizarán el sistema. Agrega información como: frecuencia de uso, experiencia técnica, privilegios de acceso, otros)

Tipo Usuario	Descripción o función	Privilegios de acceso	Frecuencia de uso
(Cargo o rol que desempeñara en el sistema)	(Descripción breve de su función)	(Menús u opciones de acceso)	(Expresado en horas, días, semanas, meses o años)

2. Contexto Operativo

(Describe el contexto operativo en el que operará el software. Incluye: Plataformas de hardware, sistemas operativos, otros sistemas con los que interactuará)

3. Funciones del software

(Esta sección, describe las funciones del software. Describe las funciones en procesos, pantallas o tipos de usuario)

3.1. Identificador: Nomenclatura F001 (Letra F seguido de serie por tres dígitos)

3.1.1. Nombre del Proceso: Nombre del proceso o módulo a afectar.

3.1.2. Tipo de usuario: (nombre del cargo del usuario quien realiza el proceso)

- 3.1.3. **Datos de Entrada:** Datos que haya que digitar o cargar al sistema
- 3.1.4. **Descripción del proceso:** descripción de lo que el sistema realizará. Si es necesario para su explicación adjuntar propuesta de interface para su mejor comprensión.
- 3.1.5. **Datos de salida:** (Lo que se espera que el sistema realice, como mostrar mensajes de alerta u otros)
- 3.1.6. **Validaciones:** (Condiciones requeridas para que sean validos los datos digitados)
- 3.1.7. **Criterios de Aceptación:** (Una vez finalizado el proceso qué condición indica que resultado exitoso)
- 3.1.8. **Prioridad:** (Alta, media o baja)
- 3.1.9. **Diagramas de flujo:** (diagrama de flujo del proceso)

4. Requerimientos no funcionales

- 4.1. **Disponibilidad:** Cuánto tiempo el sistema debe de estar disponible para acceder o usar
- 4.2. **Integridad:** Qué se hace para que toda la información esté completa, integra y verificable en todo momento
- 4.3. **Interoperabilidad:** La preparación y la capacidad del software para interactuar con otras aplicaciones o sistemas
- 4.4. **Desempeño:** Tiempo de respuesta, tamaños máximos de archivos, tiempos de carga, etc.
- 4.5. **Seguridad de la información:** No permitir el acceso a terceros no autorizados. Que controles se usarán para este fin.



Elaborado por: (Nombre del analista quien lo elabora, cargo y correo institucional)

CC: (Nombres y cargos de las personas a quien les llega una copia).

Nota: El requerimiento debe ser firmado y sellado por el jefe del área remitente.



14.7. Anexo G: Instructivo para llenado de formato de Acta de Entrega y Hoja de pruebas

Logo de
gobierno

Logo
institucional

Dirección General de Ingresos

División de Informática y Sistemas

DIS

Instructivo para llenado de formato de Acta de Entrega y Hoja de Pruebas de requerimientos o modificaciones a los sistemas informáticos

(Versión 1)

Octubre, 2022

I. Introducción

Como parte de los procesos de desarrollo dentro de la División de Informática y Sistemas, es necesario poseer los formatos idóneos para poder realizar correctamente cada paso que conlleve a un desarrollo eficaz y que minimicen los errores.

El formato de acta de entrega para sistemas informáticos es el documento donde se detalla los objetos modificados dentro del aplicativo del sistema, ya sea formularios, esquemas o base de datos. La hoja de pruebas que va adjunta con el acta de entrega, es el documento donde se explica las pruebas a realizar a un sistema informático.

El presente instructivo describe el proceso para llenado de ambos formatos.

II. Objetivo

- Instruir sobre el llenado de formato para redactar acta de entrega y hoja de pruebas para modificación o creación de sistemas informáticos

III. Aspecto Legal:

Decreto Ejecutivo N°. 20-2003 Reformas e Incorporaciones al reglamento de la Ley Creadora de la Dirección General de Servicios Aduaneros y de Reforma a la Ley Creadora de la Dirección General de Ingresos, ley no. 339, se establece en su artículo 22, División de Información y Sistemas:

- Realizar en forma técnica y especializada, el servicio de apoyo del procesamiento automático de la Administración Tributaria a través del Sistema de Información Tributaria (SIT).
- Recomendar nuevas aplicaciones tecnológicas de Hardware y Software y las formas de su implantación y evaluar permanentemente el estado de los sistemas de información, equipos y software, normas y procedimientos de desarrollo, instalación, mantenimiento, operación y producción.

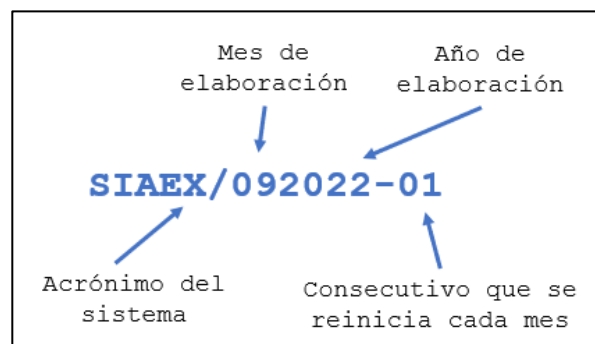
IV. Descripción del Procedimiento

A continuación, se detalla cómo se debe de rellenar cada campo del formato especificado en el inciso V de este instructivo:

- 4.1. Número del Acta de Entrega:** Se coloca el número del acta de entrega con la siguiente nomenclatura: Acrónimo del sistema/Mes y año actual-consecutivo de dos dígitos. El consecutivo debe de reiniciarse cada mes.

Ejemplo:

Imagen1: Nomenclatura de Acta de Entrega



- 4.2. **Entrega:** Se debe de colocar el nombre de la unidad a la que pertenece el analista, guion, la palabra "Analista" seguido del nombre del analista desarrollador.
- 4.3. **Recibe:** Se debe colocar el nombre de la unidad de pruebas que recibe, guion, nombre y cargo del jefe de la unidad.
- 4.4. **URL o IP de ambiente de pruebas:** Se debe colocar la URL o IP del ambiente donde se realizarán las pruebas.
- 4.5. **Remisión:** Corresponde al número de requerimiento, bitácora o solicitud que origina el desarrollo
- 4.6. **Objetivo:** Objetivo que se espera lograr con el desarrollo.
- 4.7. **Módulos o procesos afectados:** Se deben enlistar los módulos o proceso afectados por el desarrollo
- 4.8. **Objetos modificados:** Especificar los elementos que se han modificado para el desarrollo, pueden contener uno o todos los elementos que se mencionan a continuación:
- 4.8.1. **Formularios afectados:** Detallar capa y nombre del formulario y a la par los métodos afectados.
- 4.8.2. **Tablas de base de datos afectadas:** Especificar nuevas tablas o modificaciones sobre tablas existentes y los campos.
- 4.8.3. **Scripts ejecutados:** Enlistar los scripts a ejecutar en la base de datos de prueba
- 4.9. **Fecha y hora entrega:** Colocar fecha y hora de entrega del acta.
- 4.10. **Fecha y hora recibe:** Colocar fecha y hora que se recibe el acta.
- 4.11. **Entregado por:** Nombre y firma del jefe del área de desarrollo



4.12. **Recibido por:** Nombre, cargo y firma del jefe del área de pruebas

V. **Formato e instructivo**

ACTA DE ENTREGA

Número del Acta de Entrega:

(Acrónimo del sistema/Mes y año actual-consecutivo de dos dígitos)

Entrega: (Unidad a la que pertenece - Analista seguido del nombre)

Recibe: (Unidad que recibe - Nombre y cargo del jefe de la unidad)

URL o IP de ambiente de pruebas: (<https://nombredelsitio.com>)

Remisión: (Número de requerimiento, bitácora o solicitud que origina el desarrollo)

Objetivo: (Objetivo a lograr con el desarrollo)

Módulos o procesos afectados: (Enumerar los módulos o proceso afectados)

Objetos modificados:

➤ **Formularios afectados**

No.	Capa y Nombre del formulario	Métodos afectados

➤ **Tablas de Base de datos afectadas**

IP del servidor	
Nombre de base de datos	
Tabla afectada	
Campos afectados	(nombre del campo y tipo de dato)

➤ **Scripts ejecutados:** (Escribir las sentencias que se ejecutaron en base de datos de prueba)



(Lugar, día, mes y año. - Hora) (Lugar, día, mes y año. - Hora)

Fecha y hora de Entrega

Fecha y hora Recibe

Entregado por:
(Nombre jefe del área)

Recibido por:
(Nombre jefe del área)



14.8. Anexo H: Instructivo de llenado de formato de hoja de pruebas

Logo de
gobierno

Logo
institucional

Dirección General de Ingresos

División de Informática y Sistemas

DIS

Instructivo para llenado de formato de Hoja de Pruebas
de requerimientos o modificaciones a los sistemas
informáticos

(Versión 1)

Octubre, 2022

I. Introducción

Como parte de los procesos de desarrollo dentro de la División de Informática y Sistemas, es necesario poseer los formatos idóneos para poder realizar correctamente cada paso que conlleve a un desarrollo eficaz y que minimicen los errores.

El formato de hoja de pruebas que va adjunta con el acta de entrega, es el documento donde se explica las pruebas a realizar a un sistema informático, con el objetivo de detectar errores antes que esté disponible al usuario final.

El presente instructivo describe el proceso para llenado de este formato.

II. Objetivo

- Instruir sobre el llenado de formato para redactar hoja de pruebas para modificación o creación de sistemas informáticos

III. Aspecto Legal:

Decreto Ejecutivo N°. 20-2003 Reformas e Incorporaciones al reglamento de la Ley Creadora de la Dirección General de Servicios Aduaneros y de Reforma a la Ley Creadora de la Dirección General de Ingresos, ley no. 339, se establece en su artículo 22, División de Información y Sistemas:

- Realizar en forma técnica y especializada, el servicio de apoyo del procesamiento automático de la Administración Tributaria a través del Sistema de Información Tributaria (SIT).
- Recomendar nuevas aplicaciones tecnológicas de Hardware y Software y las formas de su implantación y evaluar permanentemente el estado de los sistemas de información, equipos y software, normas y procedimientos de desarrollo, instalación, mantenimiento, operación y producción.

IV. Descripción del Procedimiento

A continuación, se detalla cómo se debe de rellenar cada campo del formato especificado en el inciso V de este instructivo:

- 4.1. No. Acta de Entrega:** Colocar el número del acta de entrega a la que pertenece la hoja de pruebas. Campo digitado por el analista desarrollador.
- 4.2. No. De Remisión:** Colocar el número de remisión según acta de entrega. Campo digitado por el analista desarrollador.
- 4.3. Analista de pruebas:** Colocar nombre del analista quien realiza las pruebas. Campo digitado por el analista de pruebas.
- 4.4. Analista desarrollador:** Colocar el nombre del analista quien desarrollo los cambios del sistema. Campo digitado por el analista desarrollador.
- 4.5. Fecha y hora Inicio:** Con el formato de día, mes y año - Hora. Campo digitado por el analista de pruebas.

- 4.6. Fecha y hora de Finalización:** Con el formato de día, mes y año - Hora. Campo digitado por el analista de pruebas.
- 4.7. URL o IP Pruebas:** Colocar la URL del ambiente de pruebas. Campo digitado por el analista desarrollador
- 4.8. Base de datos de pruebas:** colocar la base de datos que se vincula con el ambiente de pruebas. Campo digitado por el analista desarrollador
- 4.9. Pruebas a realizar:** Por cada proceso a probar, repetir el cuadro del formato, asignando un número consecutivo anteponiendo una letra P y guion,
- 4.10. Incidencias encontradas:** En esta sección se deberá de enlistar cada incidencia detectada durante pruebas, detallando: la descripción, fecha en que se notificó, la fecha en que se solucionó, el estado que podrá variar entre pendiente o resuelto y por último anotar cualquier observación sobre la incidencia. Campo digitado por el analista de pruebas.
- 4.11. Fecha y hora entrega:** Colocar fecha y hora de entrega del acta. Campo digitado por el jefe que entrega.
- 4.12. Fecha y hora recibe:** Colocar fecha y hora que se recibe el acta de entrega y hoja de pruebas. Campo digitado por el jefe que recibe al finalizar las pruebas.
- 4.13. Entregado por:** Nombre y firma del jefe del área de pruebas
- 4.14. Recibido por:** Nombre, cargo y firma del jefe del área de desarrollo
- 4.15. Pantallas de pruebas:** [Enumerar por proceso las pantallas de pruebas]

4.16. Anexos: Adjuntar correo sobre aclaraciones realizadas durante pruebas, tablas o cualquier documento relevante sobre el caso

V. Formato e instructivo

Hoja de Pruebas

5.1. Datos Generales

No. Acta de Entrega: [Número de acta de entrega]	No. Remisión: [Número de remisión según acta de entrega]
Analista de pruebas: [nombre del analista de pruebas]	Analista desarrollador: [nombre del analista desarrollador]
Fecha y hora Inicio: [día, mes y año - Hora]	Fecha y hora de Finalización: [día, mes y año - Hora]
URL o IP Pruebas:	Base de datos pruebas:

5.2. Pruebas a realizar

[En este apartado se describirán en detalle cada una de las pruebas realizadas que se hayan identificado como necesarios para verificar la funcionabilidad completa del sistema. Se deberá repetir una tabla por cada caso de prueba que se defina.]

No. Prueba [P-001]	[Módulo o Proceso]
Cargo de usuario: [Indicar el cargo de usuario con el cual acceder]	Usuario: [Usuario con que realizó la prueba]
Descripción: [Descripción del procedimiento realizado]	
Observaciones: [Anotar cualquier particularidad sobre el caso, incluyendo diferentes escenarios que se puedan dar]	

Resultado esperado: [lo que se espera que realice el sistema]
Resultado obtenido: [lo que el sistema realizó]

5.3. Incidencias encontradas

No.	Descripción	Fecha Notificación	Fecha Solución	Estado	Observación
				[Pendiente o Resuelta]	

[Lugar, día, mes y año. - Hora]

[Lugar, día, mes y año. - Hora]

Fecha y hora de Entrega

Fecha y hora Recibe

Entregado por:

Recibido por:

[Nombre jefe del área]

[Nombre jefe del área]

5.4. Pantallas de pruebas

[Enumerar por proceso las pantallas de pruebas]

5.5. Anexos

[Adjuntar correo sobre aclaraciones realizadas durante pruebas, tablas o cualquier documento relevante sobre el caso]



14.9. Anexo I: Instructivo de llenado de formato de bitácora

Logo de
gobierno

Logo
institucional

Dirección General de Ingresos

División de Informática y Sistemas

DIS

Instructivo para llenado de formato de Bitácora de
incidencias de los sistemas informáticos

(Versión 1)

Octubre, 2022

VI. Introducción

Como parte de los procesos de desarrollo dentro de la División de Informática y Sistemas, es necesario poseer los formatos idóneos para poder realizar correctamente cada paso que conlleve a un desarrollo eficaz y que minimicen los errores.

El formato de bitácora son documentos donde se detalla las inconsistencias presentadas para los procesos de los sistemas de la administración tributaria. Las inconsistencias representan errores o procedimientos que no se realizan correctamente en los sistemas informáticos institucionales.

El presente instructivo describe el proceso para llenado de este formato.

VII. Objetivo

- Instruir sobre el llenado de formato para redactar hoja de pruebas para modificación o creación de sistemas informáticos

VIII. Aspecto Legal:

Decreto Ejecutivo N°. 20-2003 Reformas e Incorporaciones al reglamento de la Ley Creadora de la Dirección General de Servicios Aduaneros y de Reforma a la Ley Creadora de la Dirección General de Ingresos, ley no. 339, se establece en su artículo 22, División de Información y Sistemas:

- Realizar en forma técnica y especializada, el servicio de apoyo del procesamiento automático de la Administración Tributaria a través del Sistema de Información Tributaria (SIT).
- Recomendar nuevas aplicaciones tecnológicas de Hardware y Software y las formas de su implantación y evaluar permanentemente el estado de los sistemas de información, equipos y software, normas y procedimientos de desarrollo, instalación, mantenimiento, operación y producción.

IX. Descripción del Procedimiento

A continuación, se detalla cómo se debe de rellenar cada campo del formato especificado en el inciso V de este instructivo:

9.1. Acrónimo del sistema: En la parte superior colocar el acrónimo del sistema seguido del nombre al que pertenece la bitácora.

9.2. Datos del caso: Se indica los datos generales como:

- **No.:** Número de bitácora, con la nomenclatura de Primeras tres letras del Sistema-Mes año-consecutivo anual. Ejemplo: EXO-092022-220
- **Fecha:** Fecha en que se realizó la bitácora con el formato de día-mes-año dd-mm-aaaa.
- **Analista:** Nombre del analista del área normativa o administración de renta que presenta el caso.
- **Área:** Área o departamento a la que pertenece el analista.
- **Severidad:** Se distinguirá entre las siguientes categorías:

Crítica, cuando el caso requiere de atención inmediata porque afectan o impiden el funcionamiento principal del sistema y puede salir de servicio.

Mayor, cuando no funciona correctamente uno o algunos procesos, pero otros sí funcionan correctamente.

Menor, cuando el error no tiene impacto en la funcionabilidad. Ejemplo: corrección de textos.

- **Referencia del caso:** Breve descripción del caso que indique el error presentado

9.3. Análisis y Evaluación del caso: Se detallará el análisis realizado por el analista de control de calidad.

- **Ambiente de pruebas:** Colocar URL del ambiente en que se realizó la prueba.
- **Usuario:** Colocar el usuario que se usó para replicar el caso
- **Cargo o rol de usuario:** Indicar el cargo o rol que desempeña el usuario con el que se replicó el caso.
- **Pasos reproducidos:** Enlistar cada paso detallando los procedimientos o circunstancias específicas que llevaron a generar el caso
- **Análisis de situación y propuesta de solución:** Realice observaciones sobre el caso indicando posibles causas y describa la propuesta para solucionar el caso presentado.
- **Revisado por el analista:** Indicar nombre del analista del área de calidad.

- **Revisado el día:** Indicar el día que se realizó la revisión de la bitácora. Formato día, mes y año (dd-mm-aaaa).
- **Remitido el día:** Indicar el día que se remitió al área de desarrollo. Formato día, mes y año (dd-mm-aaaa).

9.4. Solución del caso: Campo llenado por el analista desarrollador. Se debe de indicar:

- **Procedimientos implementados para la solución:** Describir los cambios correctivos que se realizaron
- **Revisado por el analista:** Indicar nombre del analista del área de desarrollo.
- **Revisado el día:** Indicar el día que se realizó la revisión de la bitácora. Formato día, mes y año (dd-mm-aaaa).
- **Remitido el día:** Indicar el día que se remitió al área de calidad. Formato día, mes y año (dd-mm-aaaa).

9.5. Anexos: Adjuntar documentos, correos, pantallas, consultas y todos aquellos elementos que ayuden al entendimiento y soporte del caso.

X. Formato e instructivo

Logo de gobierno	División de Informática y Sistemas Dirección General de Ingresos Bitácora		Logo institucional
[Acrónimo - Nombre del sistema]			
Datos del Informe			
No. [Primeras tres letras del Sistema-Mes año-consecutivo anual]		Fecha: [Fecha en que se realizó. Formato dd-mm-aaaa]	
Analista quien reporta el caso: [nombre del analista que remite el caso]	Área: [área del analista]	Severidad:	
		Crítica	
		X	Mayor
		Menor	
Referencia del caso: [breve descripción del caso que indique el error presentado]			

Datos del caso			
Módulo o proceso:		[nombre del módulo o proceso]	
RUC: [número RUC del contribuyente que presenta el caso]		Razón Social: [razón social del contribuyente que presenta el caso]	
Descripción: [describir el problema presentado]			
Pasos para reproducir: [Enlistar cada paso detallando los procedimientos o circunstancias específicas que llevaron a generar el caso]			
Si repite los pasos, ¿vuelve a suceder otra vez?			SI
			NO
			X
Resultado esperado: [Que debió de realizar el sistema]			
Propuesta de solución: [Describe la propuesta para solucionar el caso presentado]			

Análisis y Evaluación del caso - UCCC			
Ambiente de pruebas:		[URL del ambiente de pruebas]	
Usuario: [usuario que uso en pruebas]		Cargo o rol de usuario: [cargo o rol que desempeña el usuario]	
Pasos reproducidos: [Enlistar cada paso detallando los procedimientos o circunstancias específicas que llevaron a generar el caso]			
Análisis de situación y propuesta de solución: [Indique posibles causas y describa la propuesta para solucionar el caso presentado]			
Revisado por el analista:		Revisado el día:	Remitido el día:

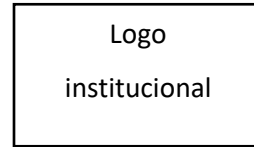
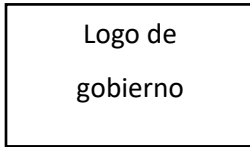


Solución del caso - [Acrónimo Unidad de Desarrollo]		
Procedimientos implementados para la solución: [Describir los cambios correctivos que se realizaron]		
Revisado por el analista:	Revisado el día:	Remitido el día:
Firma y fecha del jefe desarrollo		
Firma y fecha del jefe de calidad		

Anexos: [Adjuntar correos, pantallas visuales, consultas, etc.]



14.10. Anexo J: Manual Propuesto de Desarrollo de Sistemas



Dirección General de Ingresos
División de Informática y Sistemas
DIS

Manual de procedimientos para el desarrollo de sistemas
informáticos
(Versión 1)

Octubre, 2022

I. Introducción

La Dirección General de Ingresos (DGI), a través de la División de Informática y Sistemas (DIS), desarrolla y da mantenimiento a los sistemas tributarios que están a la disposición de contribuyentes, entidades y servidores públicos como herramientas para la recaudación fiscal.

Por tal motivo, es necesario definir las funciones de los involucrados, documentos utilizados y establecer los procedimientos necesarios para el desarrollo de sistemas informáticos, bajo las metodologías ágiles (SCRUM y KANBAN) y buenas prácticas de programación, de tal manera que garantice la eficiencia de los sistemas y minimicen los riesgos existentes durante estos procesos.

El proceso de desarrollo o ampliación de sistemas se dividirá en dos tipos: Requerimientos Mayores y Requerimientos Menores, clasificados de acuerdo al esfuerzo requerido para el cumplimiento del desarrollo.

II. Objetivo

- Estandarizar los procesos de desarrollo de sistemas informáticos desde las buenas prácticas y metodologías ágiles de desarrollo de software.

III. Aspecto Legal:

Decreto Ejecutivo N°. 20-2003 Reformas e Incorporaciones al reglamento de la Ley Creadora de la Dirección General de Servicios Aduaneros y de Reforma a la Ley Creadora de la Dirección General de Ingresos, ley no. 339, se establece en su artículo 22, División de Información y Sistemas:

- Realizar en forma técnica y especializada, el servicio de apoyo del procesamiento automático de la Administración Tributaria a través del Sistema de Información Tributaria (SIT).
- Recomendar nuevas aplicaciones tecnológicas de Hardware y Software y las formas de su implantación y evaluar permanentemente el estado de los sistemas de información, equipos y software, normas y procedimientos de desarrollo, instalación, mantenimiento, operación y producción.

IV. Glosario

Para efectos del presente manual, se entenderá por:

Área usuaria/solicitante: Áreas normativas o administrativas que son las encargadas de solicitar los requerimientos de sistemas nuevos o modificación a uno existente.

SCRUM: Metodología de desarrollo ágil, basado en un marco de trabajo simple que promueve la colaboración en los equipos para lograr desarrollar productos complejos.

KANBAN: es un método ágil, muy popular, de gestión del flujo de trabajo para definir, gestionar y mejorar los servicios que proporciona el trabajo de conocimiento. Te ayuda a visualizar el trabajo, maximizar la eficiencia y mejorar continuamente. El trabajo se representa en tableros Kanban, lo que te permite optimizar la entrega de trabajo a través de múltiples equipos y manejar, incluso los proyectos más complejos en un solo entorno.

Product Owner: Rol dentro de la metodología SCRUM que levantará los requerimientos de las áreas solicitantes para plantearlos en el Product Backlog, dirige y da seguimiento al proyecto a desarrollar.



SRUM Master: Rol dentro de la metodología SCRUM, que actúa como facilitador entre el Product Owner y el Team SCRUM.

TEAM SCRUM: Rol dentro de la metodología SCRUM, es el equipo de analistas desarrolladores encargados de programar el sistema.

Sprint: Iteración de desarrollo dentro de la metodología SCRUM

Product BackLog: Es el trabajo pendiente del producto, recoge la lista del trabajo que debe realizarse.

Sprint BackLog: Es el trabajo pendiente del sprint, es decir, la serie de trabajos o productos con los que el equipo se ha comprometido durante el sprint de SCRUM.

Incremento de producto o Release: Es lo que se entrega al final de cada sprint.

Burndown Chart: El Burndown Chart es un gráfico de trabajo pendiente a lo largo del tiempo que muestra la velocidad a la que se están completando los objetivos, requisitos, o historias de usuarios. Permite extrapolar si el equipo podrá completar el trabajo en el tiempo estimado.

Bugs: Son errores o defectos en el software que hace que un programa no funcione correctamente. Son ocasionados por múltiples causas como código, lógica del programador, incompatibilidad de una librería o un error en la plataforma de desarrollo.

V. **Ámbito de aplicación**

5.1. **Generales**

- El proceso de requerimientos mayores considera a aquellos proyectos mayores a 2 semanas y máximo de 6 meses totales de esfuerzo.
- El proceso de requerimientos menores, considera a aquellos proyectos menores a 2 semanas.
- El proceso de requerimientos mayores se trabajará con la metodología SCRUM. Estos proyectos se podrán dividir en varios módulos o sprint funcionales para ser puestos en producción.
- El proceso de requerimientos menores, se trabajará con la metodología KANBAN adaptada al marco de trabajo actual. Estos proyectos son mejoras o cambios mínimos en los sistemas que no requieran de un equipo y su tiempo sea corto.
- El Proceso de Control de Cambios, es aplicable para cualquier cambio de funcionalidad requerido para el Proyecto (Mayor o Menor), que no se encuentre establecido en el Alcance original del mismo

5.2. Específicas

Involucrados:

- a) Analista del área usuaria/solicitante
- b) Product Owner
- c) SCRUM Master
- d) Team SCRUM
- e) Líderes de unidad

Documentos del proceso de requerimiento Mayor

Documento	Criterio	Involucrados
Product BackLog	Correctamente redactado, organizado por prioridades y comprensible	Product Owner Áreas solicitantes

Sprint BackLog	Tareas alcanzables durante el tiempo establecido para el Sprint	Product Owner SCRUM Master Team SCRUM
Incremento	Proyecto funcional y listo para ser instalado en producción	Team SCRUM
Burndown Chart	Actualizado diario	

Documentos del proceso de requerimiento Menor

Documento	Criterio	Involucrados
Requerimiento de nuevo sistema o ampliación	Claro y preciso con los requerimientos de sistema.	Área solicitante
Acta de Entrega	Completo y validado	Analista desarrollador Líder de unidad
Hoja de Pruebas	Completo y validado	Desarrollo Líder de Unidad de pruebas

5.3. Control de cambios

Los cambios que se necesiten realizar a un requerimiento por cualquier motivo que se origine, debe de ser remitido a través de un memorándum de Ampliación o Mejora con referencia al requerimiento original.

VI. Consideraciones Generales para el desarrollo de código del software

- Los nombres de los sistemas deben de hacer referencia al proceso general que se abordará

Durante la programación de los sistemas los analistas desarrolladores deben de:

- Utilizar nombres claros para variables, métodos o funciones. Ejemplo: En vez de "nm" usar "numero_de_mes".
- Evitar el uso de literales sino constantes. EJEMPLO: En vez de usar "cf" declarar: "CONSTANTS: febrero TYPE string VALUE '02'"
- Organizar el código. En vez de realizar todo el proceso de una sola vez, organizarlo por bloques a través de subrutinas.
- Utilizar comentarios de lo que puede significar un método o variable.
- Testear el código. Probar el código programado de manera local
- Realizar control de versiones: Los analistas desarrolladores deberán de gestionar los cambios al código a través de la herramienta GIT.
- Simplificar al máximo. Se recomienda evitar la construcción de código complejo siempre que sea posible. De esta manera, se encontrarán menos bugs y se ahorrará tiempo en resolver errores.
- No reproducir fragmentos idénticos de código. Aunque se haya pensado en un código estable, no se debe copiar y pegar fragmentos de código para aprovecharlos en otros módulos. En cambio, es necesario tratar de encapsular esa parte del código en una función y aprovecharla cuando sea necesario.
- Nombrar las tablas de las bases de datos con el prefijo del nombre del sistema seguido de un guion bajo y posterior el nombre de la tabla de forma clara que tenga consistencia con el contenido.

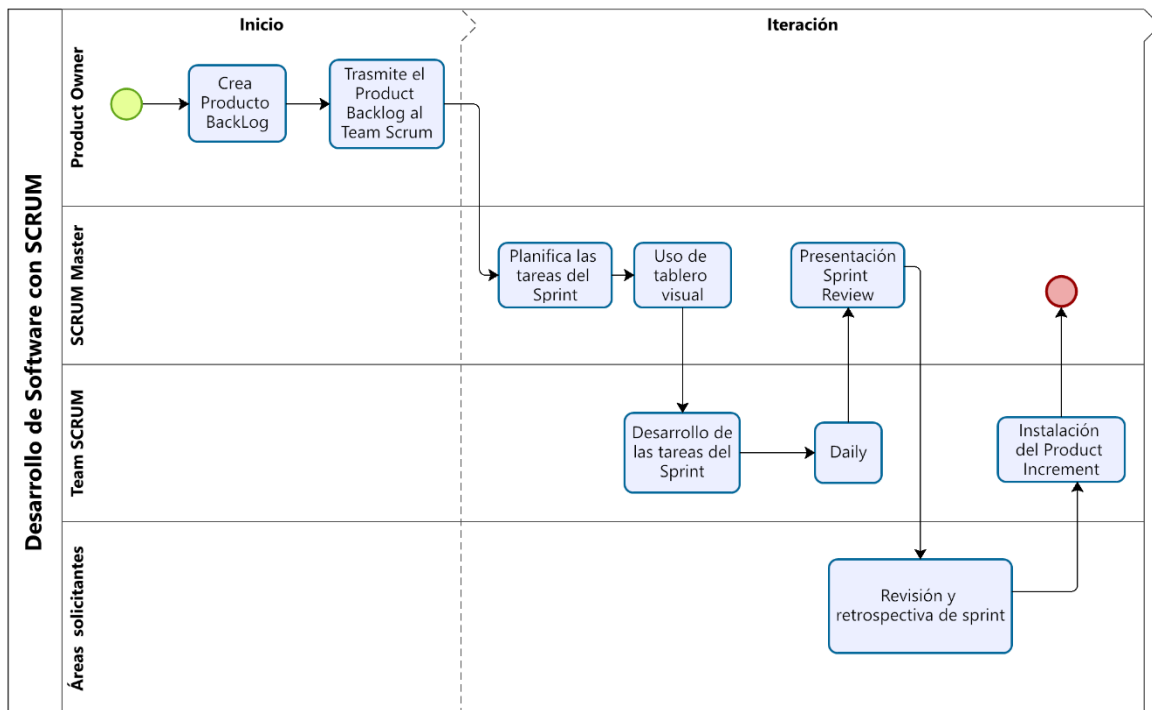
VII. Aspectos Técnicos

Para la elaboración de nuevos proyectos se deberán usar las siguientes plataformas:

Tipo de plataforma	Gestor	Versión
Entorno de desarrollo IDE	Microsoft Visual Studio	2019
Entorno de ejecución runtime	.NET framework	7.0.1
Base de datos	Base de datos SQL Server	2018
Modelado	Bizagi Modeler	
Arquitectura/Enfoque	CodeFirst	
Lenguaje	C#	

VIII. Diagrama de Flujo

Requerimientos Mayores



Requerimientos Menores

Los procesos serán:

- Expresión de la necesidad
- Recibimiento y análisis de requerimiento
- Desarrollo y retroalimentación del requerimiento
- Prueba y periodo de ajuste
- Aceptación del requerimiento

Diagrama para "Expresión de la necesidad"

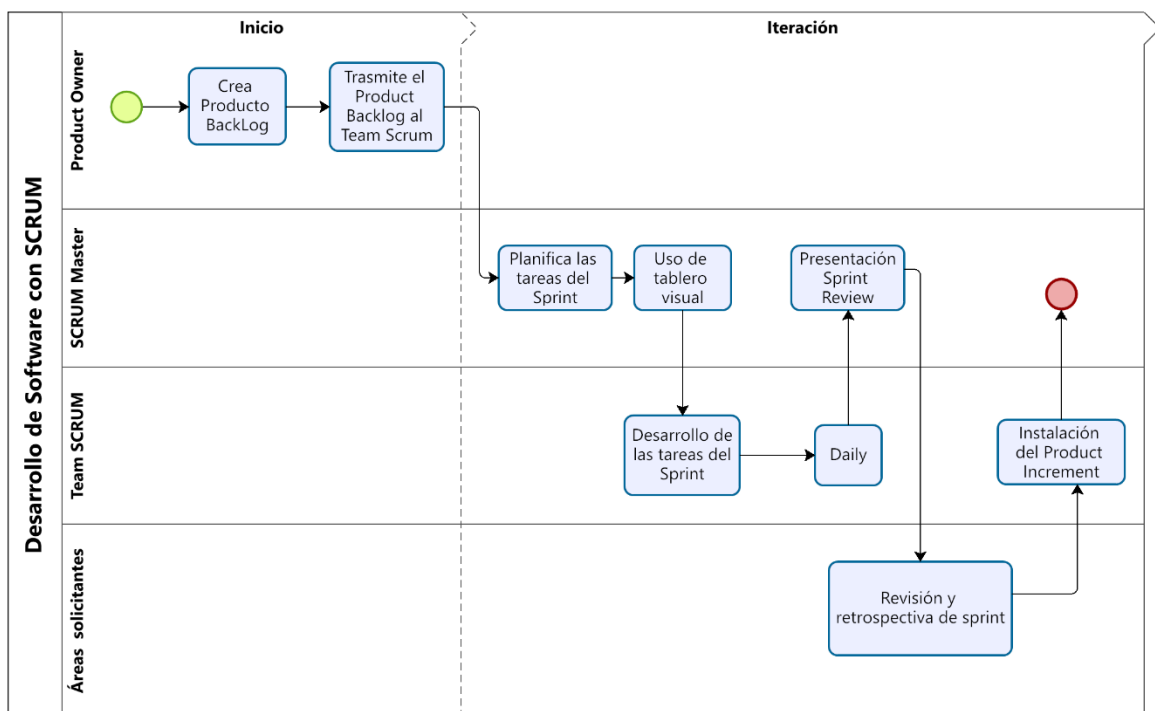


Diagrama para "Recibimiento y análisis de requerimiento"

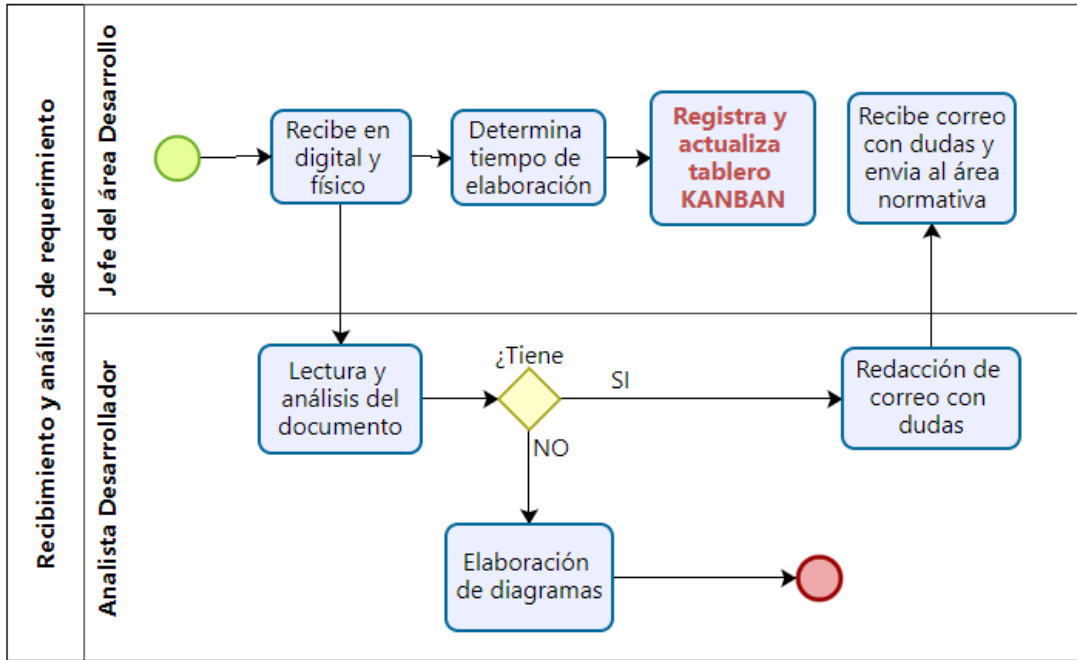


Diagrama para "Desarrollo y retroalimentación del requerimiento"

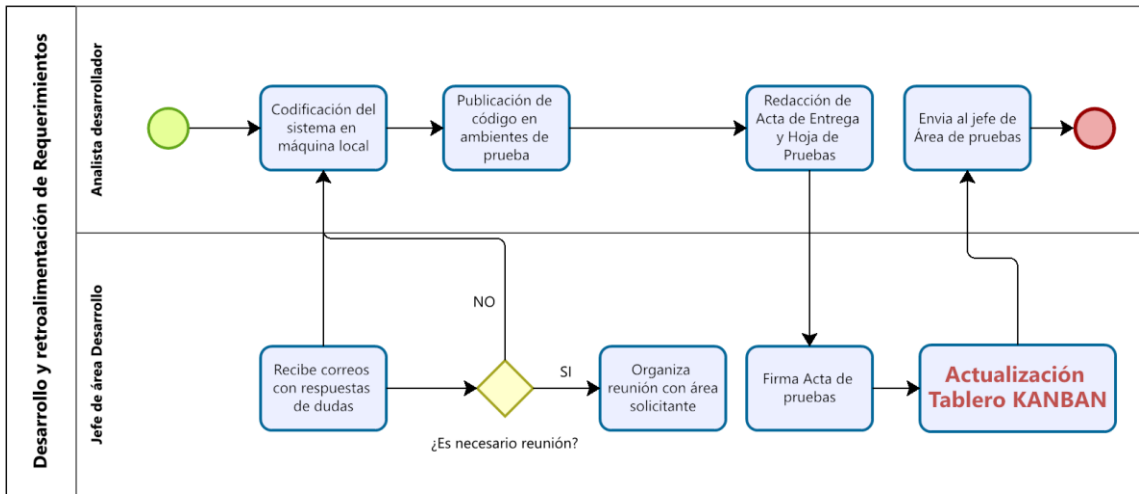


Diagrama para "Prueba y periodo de ajuste"

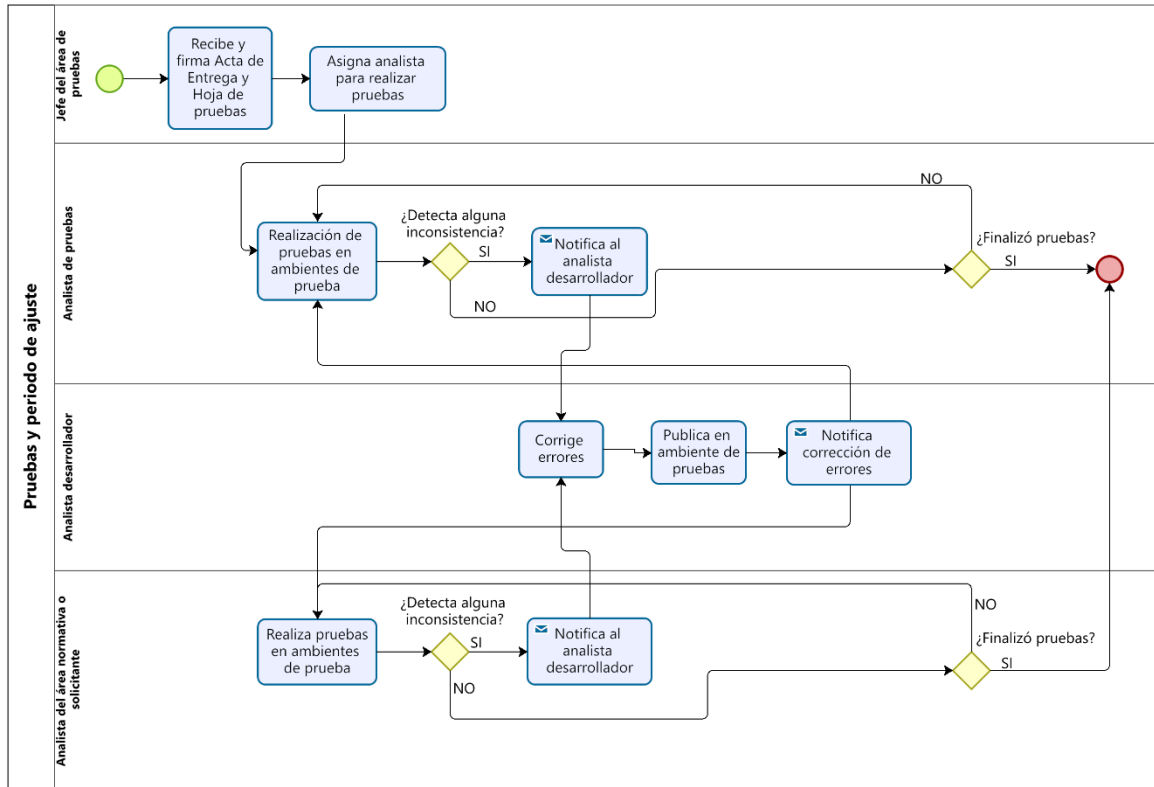
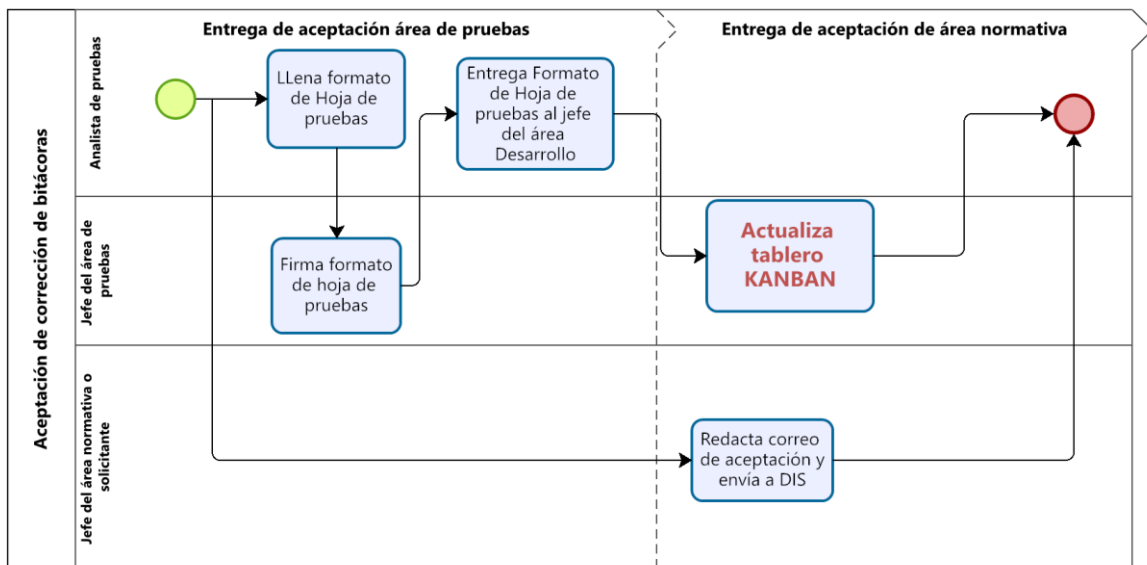


Diagrama para "Aceptación del requerimiento"



XI. Descripción del Procedimiento

Requerimientos Mayores

1. Crear BackLog

El Product Owner se reunirá con el jefe y analistas de las áreas normativas solicitantes para listar las historias de usuario.

Las historias de usuario describen las características y necesidades de un software desde la perspectiva de un usuario.

Con estas historias se crea el BackLog del proyecto, el product Owner deberá de organizarlas por orden de prioridad.

Los elementos para este paso son:

No.	Componente	Descripción
1	Especialistas involucrados	Product Owner Analista del área normativa solicitante
2	Elementos de entrada	Historias de usuario
3	Tareas realizadas que intervienen en el proceso	Establecer los criterios de aceptación de las historias de usuarios Definir si el proyecto es viable
4	Elemento de salida	Backlog con las historias de usuario definidas por importancia
5	Tiempo	1 semana

2. Planificar Sprint

El Product Owner, junto con el Scrum Master y el Team Scrum, se reúnen para planificar las tareas a realizar en el Sprint. El elemento resultante es el Sprint BackLog.

No.	Componente	Descripción
1	Especialistas involucrados	Product Owner Scrum Master Team Scrum
2	Elementos de entrada	Product Backlog Release del producto del sprint anterior (Cuando es la segunda hasta ultima iteración)
3	Tareas realizadas que intervienen en el proceso	Planificación de tareas definiendo el sprint backlog y cronograma de actividades Actualización del product backlog Actualización de tablero visual de KANBAN
4	Elemento de salida	Sprint backlog
5	Tiempo	3 horas

3. Daily

Cada día durante 15 a 20 minutos, el team y el SCRUM Master se reunirán para informar al equipo de trabajo lo que se está elaborando:

Responderán tres preguntas:

- ¿Qué hice ayer?
- ¿Qué voy hacer hoy?
- ¿Qué impedimentos tengo?

Se utilizará la técnica de "Burn Down chart" o gráfico de trabajo pendiente, para ver el estado del progreso del sprint.

No.	Componente	Descripción
1	Especialistas involucrados	Scrum Master Team Scrum
2	Elementos de entrada	Spring Backlog Tablero visual Kanban Historias de usuario Burn Down chart
3	Tareas realizadas que	Descripción de tareas realizadas por los analistas desarrolladores

	intervienen en el proceso	
4	Elemento de salida	Tablero visual Kanban actualizado
5	Tiempo	15 a 20 minutos

4. Desarrollo de Sprint

El Sprint tendrá una duración máxima de 3 semanas y serán entre 8 a 12 sprint dependiendo del tamaño del proyecto.

El Team SCRUM programará las tareas del sprint y enviarán a pruebas partes del código para que en conjunto con los analistas de calidad detecten bugs del sistema.

Utilizar formatos de Acta de Entrega y Hoja de pruebas.

5. Sprint Review:

Al finalizar el sprint el scrum, el equipo desarrollador y el scrum master presentará la versión de iteración del producto al Product Owner y las áreas solicitantes, para su debido análisis y mejoras.

No.	Componente	Descripción
1	Especialistas involucrados	Scrum Master Team Scrum Product Owner Áreas solicitantes
2	Elementos de entrada	Release del producto construido en el sprint
3	Tareas realizadas que intervienen en el proceso	Presentación del producto finalizado para ser puesto en producción
4	Elemento de salida	Revisión, aprobación, publicación en producción.
5	Tiempo	1 a 2 horas

En esta etapa el Release del producto es publicado en ambiente productivo. Utilizar formato de acta de publicación.

6. Retrospectiva

El Team scrum, junto con el scrum master y el Product Owner, se reúnen para analizar lo que hicieron durante el sprint y mejorar aquellos elementos que consideren necesarios, así el siguiente sprint se espera sea más optimizado.

No.	Componente	Descripción
1	Especialistas involucrados	Scrum Master Team Scrum Product Owner
2	Elementos de entrada	Release del producto construido en el sprint Product backlog
3	Tareas realizadas que intervienen en el proceso	Revisión del cumplimiento del producto backlog Revisión del tablero visual Burn down chart
4	Elemento de salida	Revisión, aprobación, publicación en producción.
5	Tiempo	1 a 2 horas

7. Inicio de nuevo Sprint

Finalizada las etapas anteriores, se deberá de iterar el Sprint desde el paso 2 al 6, hasta completar las tareas del BackLog.

Requerimientos Menores

1. Expresión de la necesidad

El área solicitante es la encargada de redactar el requerimiento para el desarrollo de un nuevo sistema o mejoras a uno existente. Se deberá usar el formato adjunto para redacción de requerimiento especificando todos los campos solicitados. Se pueden ayudar junto con el Product Owner quien orientará sobre cualquier duda al respecto.

Este requerimiento deberá ser redactado por el analista del área solicitante y autorizado con firma y sello del jefe de área.

2. Recibimiento y análisis de requerimiento

El Líder de la Unidad de desarrollo recibirá en físico y en digital el requerimiento. Determinará el tiempo de elaboración y registrará en la columna "Por hacer" del tablero KANBAN. También se registrará en el sistema de control de requerimientos.

El analista desarrollador leerá y analizará el requerimiento, si tiene alguna duda, redactará correo que enviará al líder del área, y éste será encargado de remitirlas al área solicitante.

Si el analista desarrollador no tiene ninguna duda, procederá de ser necesario a realizar diagramas adicionales al proceso del requerimiento como guía para el desarrollo.

No.	Componente	Descripción
1	Nombre del proceso	Recibimiento y análisis de requerimiento
2	Objetivo	Determinar la funcionabilidad del sistema a desarrollar
3	Factor que indica inicio del proceso	Asignación del requerimiento a los analistas desarrolladores
4	Especialistas involucrados	Jefes/Lider a cargo Analistas desarrolladores
5	Elementos de entrada	Requerimiento en digital o físico, asignación mediante sistema de control de requerimientos
6	Tareas realizadas que intervienen en el proceso	El jefe del área a cargo asigna el requerimiento al analista. Registra en tablero KANBAN. El analista desarrollador realiza: Lectura y análisis de requerimiento

			Elaboración de diagramas de flujo para comprensión Elaboración de correo para dudas del documento de requerimiento El jefe y el analista estiman un tiempo para el desarrollo
8	Documento registros control	o de	Registro de consultas por correo y avances por sistema de control de requerimientos y bitácoras
9	Factor indica culminación de proceso	que de	Continuar al siguiente paso con el desarrollo del sistema
10	Elemento salida	de	Requerimiento analizado

3. Desarrollo y retroalimentación del requerimiento

El analista desarrollador realiza las modificaciones en el código fuente del sistema, si hubo dudas, paralelamente se reciben por medio de correos las repuestas a las dudas de la etapa anterior, y estas son agregadas al desarrollo.

Se deberán emplear las consideraciones generales para la codificación del software descritas en este manual. Una vez finalizada la codificación, se realizan publicación en ambiente de pruebas con ayuda de los gestores de control de código fuente. Actualizar tablero KANBAN.

Utilizar formato de Acta de Entrega y Hoja de pruebas. También utilizar el control de versiones Git Hub.

No.	Componente	Descripción
1	Nombre del proceso	Desarrollo y retroalimentación de requerimiento
2	Objetivo	Codificar el sistema de acuerdo a los requerimientos solicitados
3	Factor que indica inicio del proceso	Codificación del sistema en la plataforma y gestor de base de datos designado

4	Especialistas involucrados	Analista desarrollador Jefes del área Analista del área solicitante
5	Elementos de entrada	Respuestas de dudas vía correo, reuniones o llamadas telefónicas
6	Tareas realizadas que intervienen en el proceso	Codificación en su equipo local Reuniones o llamadas telefónicas para aclaración de dudas Publicación de código en ambientes de prueba Redacción de acta de entrega y hoja de pruebas
8	Documento registros control o de	Acta de Entrega y Hoja de pruebas
9	Factor que indica culminación de proceso	Entrega de los documentos de acta de entrega y hoja de pruebas al área de pruebas
10	Elemento de salida	Publicación de desarrollo en ambiente de pruebas Acta de Entrega y Hoja de pruebas Actualización de tablero KANBAN

4. Prueba y periodo de ajuste

El jefe del área de pruebas recibe el acta de entrega y hoja de pruebas por parte del analista desarrollador. El jefe del área asigna el caso al analista de pruebas de acuerdo a los sistemas asignados previamente.

Esta asignación se realizará cada 6 meses a los analistas para redistribuir la carga de trabajo.

El analista de pruebas, va documentando cada paso realizado en el sistema, esto a través de captura de pantallas. Realiza las pruebas en ambiente designado, verifica base de datos y sistemas afines al proceso. También registra las pruebas en el sistema de control de requerimientos y bitácoras.

Se notificará cualquier inconsistencia detectada a través del sistema de control de requerimientos y bitácoras. Cuando reciban respuesta continuar las pruebas. Cuando se hayan realizado todas las pruebas satisfactoriamente, se procederá al siguiente paso de aceptación. El jefe del área de pruebas deberá de firmar el acta de pruebas ya culminada.

Los analistas del área normativa, también realizarán las pruebas y notificarán errores. Una vez culminadas las pruebas, se procederá al siguiente paso.

Los elementos para esta etapa son:

No.	Componente	Descripción
1	Nombre del proceso	Prueba y periodo de ajuste
2	Objetivo	Garantizar que el sistema esté libre de errores y funcione correctamente en ambiente productivo
3	Factor que indica inicio del proceso	Recibimiento de acta de entrega y hoja de pruebas por parte del analista desarrollador
4	Especialistas involucrados	Analista desarrollador Jefe de área de pruebas Analista de pruebas Analistas del área normativa o solicitante
5	Elementos de entrada	Acta de entrega y Hoja de pruebas
6	Tareas realizadas que intervienen en el proceso	Por parte del analista de pruebas: Revisión de hoja de pruebas y requerimiento para realizar los procesos contemplados en estos documentos Realización de pruebas en los ambientes correspondientes verificando los cambios en las bases de datos y sistemas a fines al desarrollo. Rellenar formato de hojas de pruebas Firma del formato de hoja de pruebas por parte del jefe del área una vez culminada las pruebas

No.	Componente	Descripción
		Por parte del área normativa o solicitante: Notificación de errores a través de correo electrónico Solicitud de algún cambio o mejora a través de correo para posteriormente ser enviado por memorándum.
8	Documento o registros de control	Hoja de pruebas Formato de Plan de Pruebas Registro en el sistema de control de requerimiento y bitácoras.
9	Factor que indica culminación de proceso	Procesos de requerimiento completados correctamente
10	Elemento de salida	Hoja de pruebas firmada por el jefe de área de pruebas remitidas al área de desarrollo

5. Aceptación del requerimiento

La forma de aceptación por parte de los analistas de pruebas, es el formato de Hoja de Pruebas, debidamente relleno y firmado por el jefe del área.

La forma de aceptación por parte del área normativa, deberá ser remitida a través de correo y en físico un memorándum de aceptación de las pruebas del requerimiento.

El líder del área de desarrollo deberá de actualizar tablero KANBAN.

Los elementos para este proceso son:

No.	Componente	Descripción
1	Nombre del proceso	Aceptación del requerimiento
2	Objetivo	Documentar que las pruebas fueron satisfactorias y su aceptación para publicación en producción

3	Factor que indica inicio del proceso	Finalización de pruebas
4	Especialistas involucrados	Analista de pruebas Analista del área normativa o solicitante
5	Elementos de entrada	Hoja de pruebas Formato de Plan de Pruebas
6	Tareas realizadas que intervienen en el proceso	Redacción de documento para aceptación de requerimiento que corresponde realizarlo al área normativa o solicitante Redacción en hoja de pruebas que las pruebas fueron satisfactorias
8	Documento o registros de control	Memorándum de aceptación Hoja de pruebas Formato de Plan de Pruebas Registro en el sistema de control de requerimiento y bitácoras.
9	Factor que indica culminación de proceso	Redacción del memorándum de aceptación Actualización en tablero KANBAN
10	Elemento de salida	Memorándum de aceptación

XII. Formato e instructivo

Ver instructivos de llenado de formatos para:

- Requerimiento
- Acta de Entrega
- Hoja de Pruebas

XV. Glosario de términos técnicos

Área usuaria/solicitante: Áreas normativas o administrativas que son las encargadas de solicitar los requerimientos de sistemas nuevos o modificación a uno existente.

SCRUM: Metodología de desarrollo ágil, basado en un marco de trabajo simple que promueve la colaboración en los equipos para lograr desarrollar productos complejos.

KANBAN: es un método ágil, muy popular, de gestión del flujo de trabajo para definir, gestionar y mejorar los servicios que proporciona el trabajo de conocimiento. Te ayuda a visualizar el trabajo, maximizar la eficiencia y mejorar continuamente. El trabajo se representa en tableros Kanban, lo que te permite optimizar la entrega de trabajo a través de múltiples equipos y manejar, incluso los proyectos más complejos en un solo entorno.

Product Owner: Rol dentro de la metodología SCRUM que levantará los requerimientos de las áreas solicitantes para plantearlos en el Product Backlog, dirige y da seguimiento al proyecto a desarrollar.

SRUM Master: Rol dentro de la metodología SCRUM, que actúa como facilitador entre el Product Owner y el Team SCRUM.

TEAM SCRUM: Rol dentro de la metodología SCRUM, es el equipo de analistas desarrolladores encargados de programar el sistema.

Sprint: Iteración de desarrollo dentro de la metodología SCRUM

Product BackLog: Es el trabajo pendiente del producto, recoge la lista del trabajo que debe realizarse.

Sprint BackLog: Es el trabajo pendiente del sprint, es decir, la serie de trabajos o productos con los que el equipo se ha comprometido durante el sprint de SCRUM.

Incremento de producto o Release: Es lo que se entrega al final de cada sprint.



Burndown Chart: El Burndown Chart es un gráfico de trabajo pendiente a lo largo del tiempo que muestra la velocidad a la que se están completando los objetivos, requisitos, o historias de usuarios. Permite extrapolar si el equipo podrá completar el trabajo en el tiempo estimado.

Bugs: Son errores o defectos en el software que hace que un programa no funcione correctamente. Son ocasionados por múltiples causas como código, lógica del programador, incompatibilidad de una librería o un error en la plataforma de desarrollo.

Software: Conjunto de programas y rutinas que permiten al ordenador ejecutar determinadas tareas, así como la documentación correspondiente.