



FACULTAD DE ARQUITECTURA E INGENIERÍAS

TRABAJO DE INVESTIGACIÓN DE FIN DE CARRERA

TITULADO:

**DISEÑO DE UN MODELO DE GOBERNABILIDAD Y GESTIÓN DE TI PARA EL
ÁREA DE DESARROLLO DE PROYECTOS DE SOFTWARE DE CORPORACIÓN
FAVORITA, BASADO EN LA METODOLOGÍA DEVOPS**

REALIZADO POR:

ING. CARLOS JULIO LANDÁZURI ORTIZ

DIRECTOR DEL TESIS

MSC. EDISON ESTRELLA

COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE:

**MÀSTER EN TECNOLOGÍAS DE LA INFORMACIÓN CON MENCIÓN EN SEGURIDAD Y
REDES**

Quito, 01 septiembre del 2019

DECLARATORIA

El presente trabajo de investigación titulado:

**“DISEÑO DE UN MODELO DE GOBERNABILIDAD Y GESTIÓN DE TI PARA EL
ÁREA DE DESARROLLO DE PROYECTOS DE SOFTWARE DE CORPORACIÓN
FAVORITA, BASADO EN LA METODOLOGÍA DEVOPS”**

Realizado por:

CARLOS JULIO LANDÁZURI ORTIZ

Como requisito para la obtención del Título de:

**MÁSTER EN TECNOLOGÍAS DE LA INFORMACIÓN CON MENCIÓN EN SEGURIDAD
Y REDES**

Ha sido dirigido por el profesor

MSC. EDISON ESTRELLA

Quien considera que constituye un trabajo original de su autor

MSC. Edison Estrella

DECLARACION JURAMENTADA

Yo, **CARLOS JULIO LANDÁZURI ORTIZ**, con cédula de ciudadanía **0401593421**, declaro bajo juramento que el trabajo aquí desarrollado es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional; y, que ha consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la UNIVERSIDAD INTERNACIONAL SEK, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

Carlos Julio Landázuri Ortiz

C.C: 0401593421

PROFESORES INFORMANTES

Después de revisar el trabajo presentado, lo ha calificado como apto para su defensa oral ante el tribunal examinador.

Ing. Diego Riofrío PhD.

Ing. Christian Pazmiño Msc.

Quito, septiembre de 2019

DEDICATORIA

A mis Padres por el apoyo y el ejemplo que siempre han guiado mis decisiones, a mis hermanos por su ayuda incondicional, a mis sobrinos y amigos que me inspiran cada día.

AGRADECIMIENTO

Mi agradecimiento a Dios por brindarme la fortaleza y la perseverancia para alcanzar esta nueva meta.

A la Universidad Internacional Sek por permitirme ser parte del alumnado.

Al Msc. Edison Estrella por brindarme las guías para culminar este proyecto.

A la Ing. María de Lourdes Reinoso y a Corporación Favorita por darme apertura para el desarrollo de este trabajo.

A mi familia por ser el motor que me motiva cada día alcanzar mis metas.

RESUMEN

Según Medrano (2013) a inicios de la década pasada con la aparición de los diferentes estándares de calidad se empezó a escuchar el término de gobernanza, con esto las empresas consideraron la idea de agregar a sus productos o servicios tecnológicos procesos de administración y control.

La gobernanza en TI es un eje transversal que tiene como objetivos evaluar, orientar y supervisar los procesos desarrollados en el área de tecnología, además debe garantizar la continuidad del negocio.

Corporación Favorita es una empresa dedicada a la compra y venta de productos, posicionada como la más grande del sector a nivel nacional. El equipo de desarrollo de proyectos de software pertenece al área de tecnología informática y su misión es proveer sistemas de software a todas las áreas de la empresa (Favorita C. , 2018).

Por lo mencionado, el modelo propuesto pretende fortalecer los puntos de gobernabilidad de TI con mayor oportunidad de mejora, identificados en el análisis de la situación actual de la Corporación, y de esta manera incorporar procesos de administración y control en todo lo relacionado al desarrollo de nuevas aplicaciones.

Corporación Favorita puede alinear sus actividades diarias a los parámetros de gobernanza más representativos y ajustarlos a las fases de DevOps. Con esto se logra implementar herramientas de control y administración en cada fase del desarrollo y operación.

Palabras clave: Software, metodologías, gobernanza, control, administración, DevOps

ABSTRACT

According to Medrano (2013) at the beginning of the last decade with the appearance of different quality standards, the term of governance began to be heard, with this the companies considered the idea of adding administration and control processes to their products or technological services

IT governance is a transversal axis that evaluates, guides and supervises the processes developed in the area of technology and must also ensure business continuity.

Corporación Favorita is a company dedicated to buying and selling products, it is the largest Ecuadorian company dedicated to this activity. The project development team belongs to the area of computer technology and its mission is to provide software systems to all areas of the company (Favorita C. , 2018).

The proposed model aims to strengthen the IT governance points with greater opportunity for improvement, identifying the current situation of the Corporation and thus incorporating administration and control processes in everything related to the development of new applications.

Finally, Corporación Favorita can align its daily activities to the most representative governance parameters and adjust them to the DevOps phases. With this it is possible to implement control and administration tools in each phase of software development and operation.

Keywords: software, methodologies, governance, control, administration, DevOps

CONTENIDO GENERAL

DECLARATORIA	I
DECLARACION JURAMENTADA	II
PROFESORES INFORMANTES	III
DEDICATORIA	IV
AGRADECIMIENTO	V
RESUMEN	VI
ABSTRACT	VII
CAPÍTULO I	6
INTRODUCCIÓN	6
1.1 EL PROBLEMA DE LA INVESTIGACIÓN	6
1.1.1 Formulación del problema	7
1.1.2 Diagnóstico del problema	8
1.1.3 Pronóstico	8
1.1.4 Objetivo general.....	9
1.1.5 Objetivos específicos	9
1.1.6 Justificación	9
1.1.7 Alcance	10
1.1.8 Tipo de estudio.....	11
1.1.9 Tipo de investigación.....	11
1.2 ESTADO DEL ARTE.....	11
CAPÍTULO II	15
MARCO TEÓRICO	15
2.1 GOBERNANZA DE TI.....	15
2.1.1 El gobierno corporativo y el gobierno de TI.....	15
2.1.2 Ventajas de implementación de gobernanza en TI	17
2.1.3 Parámetros de enfoque en gobernabilidad de TI.....	17
2.2 MODELOS DE GOBIERNO DE TI	18
2.2.1 Cobit 5:.....	19
2.2.2 Cobit 4.1.....	22
2.2.3 Modelo de madurez con parámetros de Cobit 4.1	25
2.3 METODOLOGÍAS DE DESARROLLO DE SOFTWARE	27

2.3.1	Ventajas de las metodologías de desarrollo de software	27
2.3.2	Comparativo de metodologías de desarrollo ágil más populares.....	28
2.4	METODOLOGÍA DEVOPS	30
2.4.1	Principios de DevOps	30
2.4.2	Acción centrada en el cliente	31
2.4.3	Mejora Continua	31
2.4.4	Ciclo de vida DevOps	32
2.4.5	Elección de la metodología de desarrollo	36
CAPÍTULO III		37
SITUACIÓN ACTUAL		37
3.1	ANTECEDENTES	37
3.1.1	Levantamiento de información	38
3.2	VISIÓN Y MISIÓN	39
3.3	ORGANIGRAMA.....	39
3.3.1	Arquitectura existente	40
3.4	NIVEL DE MADUREZ EN PROCESOS.	41
3.5	MATRIZ DE LEVANTAMIENTO DE SITUACIÓN ACTUAL	42
3.6	ANÁLISIS DE RESULTADOS	44
3.7	OPORTUNIDAD DE MEJORA.....	46
CAPÍTULO IV		47
MODELO DE GOBIERNO DE TI PROPUESTO.....		47
4.1	ARQUITECTURA DEL MODELO DE GOBIERNO.....	47
4.2	ETAPA DE PLANEACIÓN	49
4.2.1	Actores del proceso de planeación.....	49
4.2.2	Caracterización del proceso de Planeación.....	50
4.2.3	Flujo proceso solicitud y aprobación de un requerimiento de software	52
4.2.4	Flujo del sub proceso de levantamiento macro.....	54
4.2.5	Flujo del sub proceso levantamiento específico	56
1.1.1	Flujo del sub proceso de definición de arquitectura.	58
4.2.6	Flujo del sub proceso de selección de proveedor.....	59
4.2.7	Flujo del sub proceso arranque de un proyecto	61
4.3	ETAPA DE CODIFICACIÓN.....	62
4.3.1	Actores del proceso codificación	62
4.3.2	Caracterización del proceso de Codificación.....	63
4.3.3	Flujo proceso de codificación	65
4.3.4	Flujo del proceso de seguimiento de proyectos	67
4.4	ETAPA DE CONSTRUCCIÓN.....	68

4.4.1	Actores del proceso de construcción.....	68
4.4.2	Caracterización del proceso de Construcción	69
4.4.3	Flujo del proceso de ejecución de scripts	70
4.5	ETAPA DE PRUEBAS	72
4.5.1	Actores del proceso de Pruebas	72
4.5.2	Caracterización del proceso de Pruebas.....	73
4.5.3	Flujo de proceso de Pruebas (QA)	74
4.6	ETAPA DE LANZAMIENTO	75
4.6.1	Actores del proceso de Lanzamiento	75
4.6.2	Caracterización del proceso de Lanzamiento	76
4.6.1	Flujo de proceso de Lanzamiento.	77
4.7	ETAPA DE DESPLIEGUE	78
4.7.1	Actores del proceso de despliegue.....	78
4.7.2	Caracterización del proceso de Despliegue	79
4.7.3	Flujo de proceso de despliegue.	80
4.8	ETAPA DE OPERACIÓN	81
4.8.1	Actores del proceso de operación	81
4.8.2	Caracterización del proceso de Operación.....	82
4.8.3	Flujo de proceso de Configuración de ambiente.....	83
1.1.2	Flujo de proceso de capacitación de aplicativos al área de producción	84
4.9	ETAPA DE MONITOREO	86
4.9.1	Actores del proceso de monitoreo.....	86
4.9.2	Caracterización del proceso de Monitoreo	87
4.9.3	Flujo de proceso de paso a mantenimiento	88
4.10	SEGURIDAD EN DESARROLLO.	90
4.11	SISTEMATIZACIÓN DE DEVOPS	91
CAPÍTULO V.....		92
CONCLUSIONES Y RECOMENDACIONES.....		92
5.1	CONCLUSIONES	92
5.2	RECOMENDACIONES:	94
BIBLIOGRAFÍA.....		95

ÍNDICE DE TABLAS

Tabla 1 Controles Planear y organizar	23
Tabla 2 Parámetros para la evaluación de madurez de procesos	25
Tabla 3 Diferencias entre Cobit 4.1 y Cobit5	26
Tabla 4 Cuadro comparativo metodologías populares.....	29
Tabla 5 Parámetros de evaluación para nivel de madurez de procesos	41
Tabla 6 Controles Cobit 4.1	41
Tabla 7 Resultado de nivel de madurez del área.....	42
Tabla 8 Levantamiento de parámetros de gobernabilidad del área de desarrollo.	44
Tabla 9 Promedio de cumplimiento de los parámetros de Gestión de TI.	45
Tabla 11 Actores del proceso de Planeación.....	49
Tabla 12 Caracterización del proceso de Planeación	51
Tabla 13 Actores del proceso de Codificación.	62
Tabla 14 Actores del proceso de construcción.....	68
Tabla 15 Caracterización proceso de Construcción	69
Tabla 16 Actores del proceso de pruebas.....	72
Tabla 17 Caracterización proceso de Pruebas.....	73
Tabla 18 Actores del proceso de lanzamiento.....	75
Tabla 19 Caracterización proceso de lanzamiento.....	76
Tabla 20 Actores del proceso de despliegue.	78
Tabla 21 Caracterización proceso de despliegue.	79
Tabla 22 Actores del proceso de operación.	81
Tabla 23: Caracterización proceso Operación.	82
Tabla 24: Actores del proceso de monitoreo.....	86
Tabla 25: Caracterización proceso de monitoreo.....	88

ÍNDICE DE FIGURAS

Figura 1 Áreas de Enfoque del Gobierno de TI.	17
Figura 2 Objetivo de Gobierno Crear Valor	19
Figura 3 Gobierno y gestión en Cobit 5	20
Figura 4 Marco de referencia único integrado Cobit 5	20
Figura 5 Catalizadores corporativos Cobit 5.....	21
Figura 6 Áreas Clave de Gobierno y Gestión de COBIT 5.....	21
Figura 7 Los 4 dominios de Interrelación de Cobit.....	22
Figura 8 Fases de mejora continua.....	32
Figura 9 Fases definidas para DevOp.	32
Figura 10 Organigrama de área de Desarrollo de proyectos.....	39
Figura 11: Arquitectura actual de aplicaciones web	40
Figura 12 Ejemplo de tablero de indicadores de Gestión del área de Desarrollo	45
Figura 15 Modelo actual de desarrollo de aplicaciones	46
Figura 16 Arquitectura del modelo de Gobernanza propuesto.	47
Figura 17 Modelo de parámetros de Gobernanza propuesto.....	48
Figura 18 Flujo proceso de Solicitud y aprobación de un requerimiento de software.....	52
Figura 19 Flujo sub proceso de levantamiento macro.	54
Figura 20 Flujo sub proceso de levantamiento específico	56
Figura 21 Flujo del sub proceso de definición de arquitectura.	58
Figura 22 Flujo del sub proceso selección de proveedor	59
Figura 23 Flujo del sub proceso de arranque de un proyecto.	61
Figura 24 Caracterización del proceso de Codificación.	64
Figura 25 Flujo proceso de Codificación.	65
Figura 26 Flujo del proceso de seguimiento de proyectos.	67
Figura 27 Proceso de ejecución de Scripts.....	70
Figura 28 Proceso de Pruebas	74
Figura 29 Flujo del proceso de versionamiento de aplicaciones.....	77
Figura 30 Flujo del proceso de despliegue.....	80
Figura 31 Flujo del proceso de configuración de ambiente.	83
Figura 32 Flujo del proceso de capacitación a producción y locales.	84
Figura 33 Flujo del proceso de paso a mantenimiento.....	88
Figura 34 Modelo de seguridad de Open Sam con DevOps.	90
Figura 35 Parámetros de seguridad a evaluar según SWAT	91

CAPÍTULO I

INTRODUCCIÓN

1.1 El Problema de la investigación

1.1.1 Planteamiento del problema

La tecnología es un factor determinante en el giro de una empresa o un negocio. Cuando no existe orden y control sobre los procesos, las actividades pueden ser difíciles de controlar y medir, respecto a los costos y gastos operativos, afectando la gobernabilidad de sus actividades (Albarracín, 2014).

Corporación Favorita C.A. es una empresa con más de 65 años de servicio, que se encarga de comprar y vender productos de consumo masivo, posicionándose como la primera empresa de *retail* en Ecuador (Obando Tapia, 2015).

Desde su creación, el modo de operar ha dependido de la demanda de sus clientes, por lo que las áreas que conforman la empresa han tenido un crecimiento constante. En el caso del área de tecnología informática, han venido cumpliendo sus objetivos estratégicos sin una estructura formal como base de desarrollo de las actividades que realizan.

En el área de tecnología informática de Corporación Favorita C.A., se encuentra una sub-área de trabajo llamada Desarrollo de Proyectos de software, que se encarga de gestionar los proyectos que el resto de las áreas requieren, es decir cuentan con sistemas propios evitando la adquisición de licenciamiento y soporte externo.

La mayoría de sistemas, han sido desarrollados con la ayuda de proveedores externos como fábricas de software, ya que internamente no cuentan con el suficiente número de personal capacitado para realizar esta actividad y existe una gran cantidad de requerimientos solicitados,

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

por lo que se ha convertido en un limitante para gestionar y dar seguimiento a los proyectos realizados.

Esta práctica ha ocasionado que en el área se genere una pérdida de gobernabilidad con respecto a procedimientos de desarrollo, control de calidad de software y puestas en producción de las aplicaciones desarrolladas (Favorita C. , 2018).

La relación entre el desarrollo de software y el modelo de gobernabilidad, van de la mano cuando se desea conseguir sistemas escalables, medibles y con parámetros de seguridad que incorporen factores de calidad de software y buenas prácticas del uso de la información. El modelo de gobernanza soluciona las falencias de planificación y seguimiento de proyectos, control de infraestructura tecnológica e integración con los procesos posteriores al desarrollo de los sistemas (Hernández, 2014).

Sin un modelo de gobernanza en el desarrollo de proyectos de Corporación Favorita C.A., el cumplimiento de sus objetivos puede verse afectados y reflejar la falla en la experiencia del usuario final.

1.1.1 Formulación del problema

La falta de un modelo de gobernanza en el área de desarrollo de proyectos de Corporación Favorita C.A., ocasiona poca claridad en las operaciones y procesos propios del área, esto se ve reflejado en el retraso del cronograma elaborado para controlar el tiempo de desarrollo del proyecto, la aparición de incidencias de seguridad asociadas a la privacidad de los datos y el mal uso de los recursos tecnológicos.

1.1.2 Diagnóstico del problema

Para cumplir con su misión el área de Proyectos de Corporación Favorita debe incorporar servicios de empresas externas como fábricas de software, originando pérdida de gobernabilidad en lo referente a: procedimientos de desarrollo, control de calidad de software y puestas en producción de las aplicaciones creadas (Favorita C. , 2018).

1.1.3 Pronóstico

La relación entre el desarrollo de software y un modelo de gobernabilidad van de la mano cuando se desea conseguir sistemas escalables, medibles y con parámetros de seguridad que no solo cumplan con el requerimiento funcional, sino que incorporen factores de calidad de software y buenas prácticas del uso de la información (Hernández, 2014).

En caso de que el área de proyectos de Corporación Favorita no utilizará un modelo de gobernanza, el cumplimiento de sus objetivos puede verse afectado lo que se verá reflejado en la experiencia del usuario final.

1.1.4 Control del Pronóstico

El uso de un modelo de gobernanza podrá solucionar las falencias en cuanto a planificación y seguimiento de los proyectos, control de la infraestructura tecnológica y la integración con los procesos posteriores al desarrollo de los sistemas (Hernández, 2014).

1.1.5 Objetivo general

Diseñar un modelo de gobernabilidad y gestión de proyectos de software, basado en las etapas propuestas por la metodología DevOps, que mejore los procesos de desarrollo que se utiliza en el área de desarrollo de proyectos de la Corporación Favorita C.A.

1.1.6 Objetivos específicos

- Explicar la metodología DevOps mediante un estudio documental, detallando cada una de sus etapas y su campo de aplicación en los procesos de desarrollo de software.
- Definir los procesos actuales de desarrollo de software y de gobernabilidad utilizados por el área de desarrollo de proyectos y plasmarlos en una matriz de procesos.
- Especificar los procesos de mejora en función de la información recolectada en la matriz de procesos para alinearlos con las fases propuestas por DevOps.
- Proponer el plan de Gobierno de TI desarrollado, que tiene como base los procesos a ser utilizados en el área de desarrollo de proyectos de Corporación Favorita C.A., lo que a su vez genere en un documento final para su socialización

1.1.7 Justificación

Un modelo de gobernabilidad para el área de desarrollo de proyectos de Corporación Favorita C.A., servirá para planificar y desarrollar nuevos proyectos de software, lo cual ayudará tanto para el desarrollo interno de sistemas como para el desarrollo con proveedores externos. Lo que se quiere lograr es que el personal directivo tenga un mejor control sobre los proyectos, y de esta forma asegurar que el producto final entregado, cuente con parámetros de calidad, para brindar confiabilidad y satisfacción al usuario final. También, controla los

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

parámetros para el uso y acceso de las plataformas tecnológicas de empresas externas, para tener un mejor control y seguridad.

La metodología DevOps en conjunto con un marco trabajo de gobierno como Cobit.4.1 se convierte en una ventaja para el área, porque permite la iteración de todas las sub-áreas involucradas en el ciclo de vida del proyecto con lineamientos de gobernabilidad.

1.1.8 Justificación metodológica

Para conseguir los objetivos planteados en este proyecto, se realizará entrevistas como técnica de investigación, las cuales serán dirigidas a la sub gerencia del área de desarrollo de proyectos de Corporación Favorita C.A., para identificar la situación actual y los procesos utilizados en el desarrollo de un proyecto de desarrollo de software.

1.1.9 Alcance

En este proyecto se determinará la situación actual del área de desarrollo de proyectos de Corporación Favorita C.A. Para lo cual, se utilizará entrevistas enfocadas en la medición de parámetros como la misión, objetivos estratégicos y las actividades llevadas a cabo en el área, y posteriormente se desarrollará un modelo que contemple procesos, actores y responsabilidades.

Esta propuesta será enfocada en el marco de referencia de gobierno de TI - Cobit 4.1 y utilizará las fases de construcción de software propuestas por DevOps. El modelo abarcará las actividades relacionadas con la planeación, construcción y operación de los sistemas, y tomará en cuenta procesos ligados a otras áreas que pueden intervenir.

1.1.10 Tipo de estudio

El proyecto propuesto es de campo, ya que la sub gerencia del área de desarrollo de proyectos de Corporación Favorita C.A., proporcionará por medio de entrevistas, la información de los procesos actuales y la necesidad de mejora de los mismos, se propondría la creación de nuevos procesos, basados en un marco de gobierno referencial.

1.1.11 Tipo de investigación

La investigación es descriptiva, ya que se plasmará la situación actual del área en cuanto a sus procesos y falencias, para lograr determinar mejoras que se pueden implementar.

1.2 Estado del arte

1.2.1 Autor(es): Universidad de las Américas-Natalí del Rocío Núñez palacios

Título de la investigación: Desarrollo de un modelo de Gobierno de TI para empresas con servicios tercerizados.

En este estudio se realiza un análisis del crecimiento de las tecnologías y la necesidad que tienen las empresas de invertir en equipamiento tecnológico, licenciamiento y capacitación. Bajo este argumento, la subcontratación de empresas especializadas es una opción para evitar que las compañías efectúen fuertes inversiones de capital. Esta actividad es conocida como Outsourcing.

Núñez (2015) indica que la tercerización de servicios es una práctica que cada vez tiene más acogida en el medio de las Tecnologías de la Información, pero destaca que esta forma de trabajo puede originar pérdida de gobernabilidad en el uso de la información y que la utilización de un modelo de Gobierno de TI puede ayudar a gestionar y controlar los servicios tercerizados.

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

Para la elaboración del modelo propuesto se hace una comparación entre Cobit5 y la Norma ISO 38500, y se identifica a Cobit5 como el mejor marco de trabajo para la construcción de un modelo de gobierno de TI, porque se acopla a las necesidades de la institución financiera donde va a ser implementado.

1.2.2 Escuela Politécnica Nacional- Alonso René Arévalo Campos

Título de la investigación: Modelo de Gobernanza de TI para el desarrollo de software en equipos pequeños.

Para esta investigación se tomaron algunas organizaciones entre públicas y privadas dedicadas al desarrollo del software, con equipos que varían entre 5 y 25 personas. Las VSE (*Very Small Entities*) son equipos pequeños de desarrollo. En este trabajo se analiza los niveles de gobernabilidad que poseen los equipos VSE en sus actividades y el valor agregado que esto otorga a sus productos (Campos, 2017)

El modelo propuesto está basado en el trabajo de Paul Bannermen (2009) –Software Development Governance, el mismo que hace relación entre meta y gestión, cubriendo ejes de gobernabilidad como: propósito, estructura, procesos y entrega continua. Según Campos (2017) el uso de un modelo de gobierno de TI otorga valor a los productos desarrollados, pero en las conclusiones de su trabajo no brinda datos de los resultados obtenidos.ca

1.2.3 Manuel Kolp, Samedi Heng- Universite Catholique de Louvain, Belgium

Título de la investigación: Mapping IT Governance to Software Development Process: From COBIT 5 to GI Tropos

Los marcos de referencia de gobierno como Cobit5 orientado a metodologías de desarrollo como RUP o GI Tropos permiten implementar gobiernos de TI con administración de reglas para el desarrollo de aplicaciones de software (Nguyen, 2018).

Esta investigación propone un mapeo de las mejores prácticas según los marcos de trabajo anteriormente mencionados, logrando la integración del gobierno de TI con los procesos de software.

Según Kolp (2018) la gobernanza dentro del desarrollo de software se puede resumir como la evaluación, dirección y monitoreo de procesos a lo largo del ciclo de vida de desarrollo.

Queda demostrado en las conclusiones que la utilización de estándares como Cobit5 ofrece un mejor marco de gestión de las actividades que intervienen en el desarrollo de software.

1.2.4 Guillermo Jiménez Marco - Universidad Politécnica de Catalunya

Título de la investigación: DevOps, la nueva tendencia en el desarrollo de sistemas TI, un caso práctico en el análisis de incidencias de software.

En este trabajo se hace un análisis de DevOps y su ciclo de vida y propone, la implementación de las etapas de desarrollo, prueba y despliegue en un sistema completo de monitorización; que abarca la mayoría de las tareas realizadas en la empresa Everis. (Jiménez, 2016)

El objetivo alcanzado es que se implementó las etapas DevOps en las diferentes áreas que intervienen en la detención y prevención de riesgos de Everis.

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

1.2.5 Lucy Ellen Lwakatare- Universitatis Ouluensis

Título de la investigación: DevOps adoption and implementation in software development practice

Según Lwakatare (2017) las fábricas de software actualmente efectúan nuevas funciones que permiten el progreso del desarrollo de forma rápida con tiempos de lanzamiento cortos y entregas parciales.

Este trabajo enfoca la relación entre la entrega continua propuesta por DevOps y la implementación de software en tiempos cortos. El análisis de varios testimonios profesionales que al utilizar las prácticas de DevOps ayudan a la mejora de aspectos del desarrollo como la velocidad y la mejora en la entrega de cambios.

Nota: No existen investigaciones iguales al propuesto en este trabajo, pero hay mucha similitud en cuanto a establecer los procesos que intervienen en un equipo de desarrollo y la implementación de un modelo de Gobierno de TI.

CAPÍTULO II

MARCO TEÓRICO

2.1 Gobernanza de TI

Naranjo (2014) define el gobierno de TI como la estructura de gestión de procesos hacia el cumplimiento de sus metas y de los objetivos empresariales.

Todos los sectores empresariales, educativos o de gobierno para su desarrollo y operación dependen de las tecnologías de la información (TI). En muchas ocasiones la eficiencia es proporcional a la inversión realizada en tecnología. Una problemática actual es que la planeación estratégica no contempla los objetivos del negocio de TI, ocasionando que la inversión en tecnología sea nula y la gestión deba acoplarse a la tecnología existente. (López, 2012)

El desconocimiento de la operativa tecnológica ocasiona que el área de TI esté aislada de las demás, por ende el gran reto es alinear los objetivos de TI con las metas estratégicas empresariales, y aunque esto parecería solo un problema de planificación va más allá, puesto que TI está sujeta a una serie de presiones regulatorias técnicas y comerciales que manejados por la urgencia hace que el área de tecnología pierda los lineamientos estratégicos (Weill, 2001)

2.1.1 El gobierno corporativo y el gobierno de TI

El gobierno corporativo es el conjunto de lineamientos, reglamentos normas y políticas que gestionan y regulan la integración de las partes de gobierno empresarial, por lo general el gobierno corporativo está a cargo de los altos mandos (Naranjo, 2014).

Una problemática actual es la falta de integración entre el gobierno corporativo y el gobierno de TI lo que genera que las tecnologías dentro de una entidad sean consideradas como un “gasto necesario” y la poca integración entre las áreas dificulta la demostración del valor agregado que las tecnologías aportan al giro del negocio. (Periñán I. L., 2011).

Existen varias instituciones que hacen esfuerzos por buscar la integración y alineación entre el gobierno de TI y el gobierno corporativo, entre las que se pueden mencionar:

- ISACA (Information Systems Audit and Control Association)
- IT GOVUK (IT Governance UK)
- ECGI (European Corporate Governance Institute)

Otras brindan lineamientos y estándares para el mismo fin por ejemplo:

- ISACA (Information Systems Audit and Control Association)
- IT GOVUK (IT Governance UK) y ECGI (European Corporate Governance Institute)
- ISO/IEC (International Organization for Standardization / International Electrotechnical Commission)
- BSI (The British Standards Institution)

Así mismo, existen entidades regulatorias por ejemplo:

- Cadbury –Report of the Committee on the Financial Aspects of Corporate Governance– (Cadbury, 1992).
- Código Olivencia de Buen Gobierno –El Gobierno de las Sociedades Cotizadas– (Olivencia, 1998).
- Company Law and Corporate Governance (European Commission, 2011)

Al momento de implantar un modelo de gobernanza dentro de TI puede existir resistencia al cambio por parte los colaboradores, para mitigar esto es importante socializar y concientizar sobre la importancia de tener una perspectiva del uso de buenas prácticas de la información y el valor agregado que eso puede dar al área (Castro, 2015).

Un adecuado órgano de gobierno de TI debe conocer a detalle la arquitectura definida para los aplicativos, de esta manera brindará los lineamientos necesarios, basados en el modelo de gobernanza implementado (Vargas, 2017).

La gobernanza de TI está muy ligada a la planificación del área, incluso en lo referente a presupuestos, donde se debe priorizar y justificar el dinero asignado a una actividad y el método de seguimiento de la misma (Vargas, 2017).

2.1.2 Ventajas de implementación de gobernanza en TI

- Buenas prácticas
- Asegura el control
- Reduce riesgos
- Optimiza la seguridad
- Claridad en la toma de decisiones
- Optimiza tareas
- Ahorro de procesos.

2.1.3 Parámetros de enfoque en gobernabilidad de TI

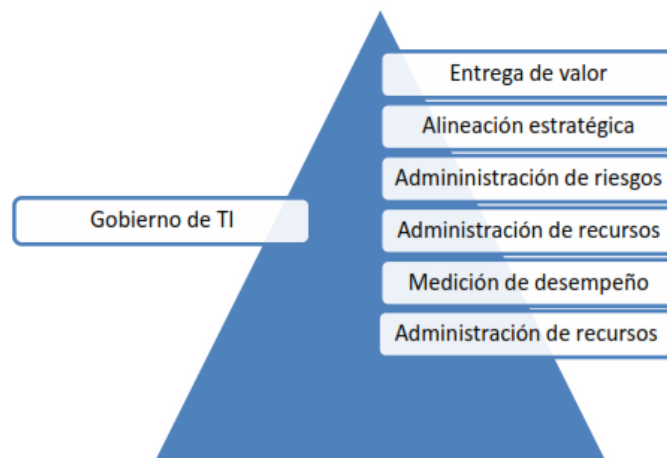


Figura 1 Áreas de Enfoque del Gobierno de TI.

Elaboración: Autor. Fuente: Muñoz (2011) Gobierno de TI. Estado del arte_Recuperado de: <https://www.redalyc.org/pdf/4115/411534384003.pdf>

2.1.4 Gobierno de TI en empresas con servicios tercerizados

Según Rantakari (2010) la tercerización de servicios consiste en asignar a un proveedor una tarea relacionada a un producto o servicio, con recursos y personal de la empresa contratada. La tercerización de servicios es recomendable solamente en tareas estratégicas y específicas del área, puesto que puede originar pérdida de gobernabilidad en las actividades desarrolladas.

En la actualidad, la tercerización de servicios tecnológicos es una estrategia muy común, puesto que en ocasiones recurrir a empresas externas resulta más beneficioso que invertir en equipo licencias o soporte (Núñez Palacios, 2015).

IT Governance Institute indica que en la tercerización de servicios la responsabilidad sigue siendo de la organización contratante; por lo tanto, ésta debe velar por la correcto uso de los recursos a través del seguimiento, monitoreo y de un plan gestión de los riesgos (Contreras, 2008).

2.2 Modelos de Gobierno de TI

Para establecer un modelo de Gobierno de TI existe una serie de marcos de trabajo de referencia, entre los más conocidos por su capacidad de implementación en empresas de diferentes tamaños esta Cobit.

Cobit es un marco de trabajo que proporciona una guía de ISACA (Asociación de Auditoría y Control de Sistemas de Información), su principal objetivo es el promover marcos de control autorizados, actualizados y aceptados, para la implementación por parte de las empresas. (Isaca, 2012)

2.2.1 Cobit 5:

Cobit 5 es un marco de referencia de gobierno y control, publicado en el año 2012, tiene como objetivo gobernar y gestionar todos los ejes corporativos, enfocándose en optimizar servicios, apoyar el cumplimiento de leyes y la gestión tecnológica. (Vivar, 2013)

Cobit 5 cumple con los siguientes principios:

- a. **Satisfacer las necesidades de las partes interesadas:** Todas las áreas de una empresa deben generar valor por ende conseguir beneficios a un óptimo costo de los recursos y con el menor riesgo.

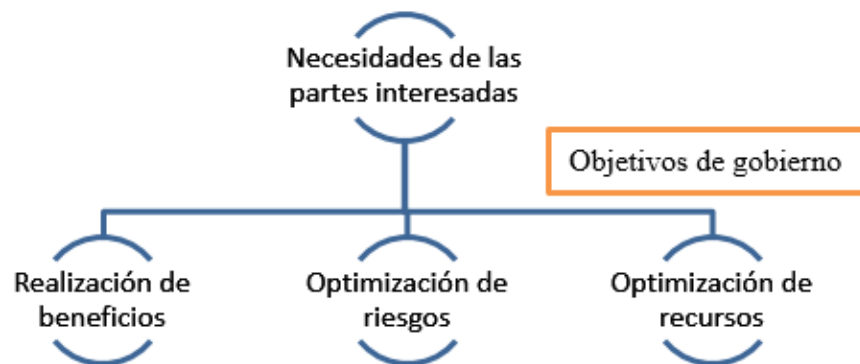


Figura 2 Objetivo de Gobierno Crear Valor

Elaboración: Autor. Fuente: ISSACA (2012) Cobit 5._Recuperado de: <https://articulosit.files.wordpress.com/2013/07/cobit5-framework-spanish.pdf>

- b. **Cubrir la Empresa de extremo a extremo:**

El objetivo es la integración del gobierno de TI con el gobierno corporativo. Cubre todas las funciones y procesos para gestionar la información empresarial y las tecnologías propias del área.

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps



Figura 3 Gobierno y gestión en Cobit 5

Elaboración: Autor. Fuente: ISSACA (2012) Cobit 5. Recuperado de: <https://articulosit.files.wordpress.com/2013/07/cobit5-framework-spanish.pdf>

- c. **Aplicar un marco de referencia único e integrado:** Permite que toda la empresa se integre a Cobit5 alineándose todos al mismo estándar, marco de referencia y políticas como un lenguaje común no solo orientado a lo tecnológico.

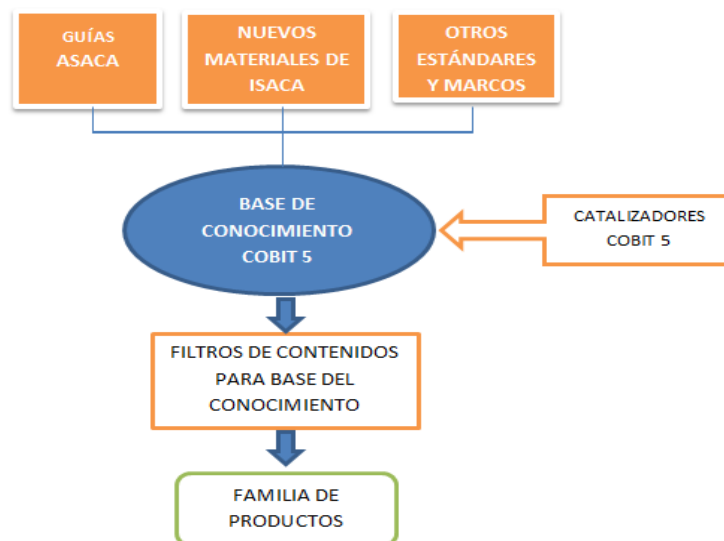


Figura 4 Marco de referencia único integrado Cobit 5

Elaboración: Autor. Fuente: ISSACA (2012) Cobit 5. Recuperado de: <https://articulosit.files.wordpress.com/2013/07/cobit5-framework-spanish.pdf>

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

- d. Hacer posible un enfoque general:** Los objetivos estratégicos de alto nivel de TI definen los diferentes objetivos, en forma de cascada.

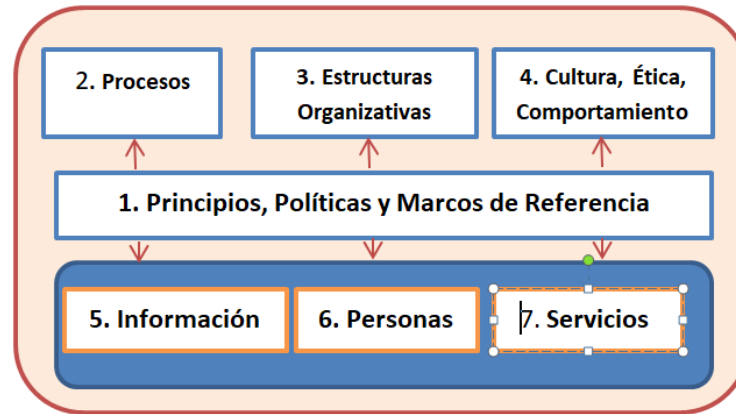


Figura 5 Catalizadores corporativos Cobit 5

Elaboración: Autor. Fuente: ISSACA (2012) Cobit 5. Recuperado de: <https://articulosit.files.wordpress.com/2013/07/cobit5-framework-spanish.pdf>

- e. Separar el gobierno de la gestión:** El gobierno y la gestión abarcan diferentes lineamientos, por ende diferentes tipos de actividades, ambos requieren estructuras organizativas diferentes.

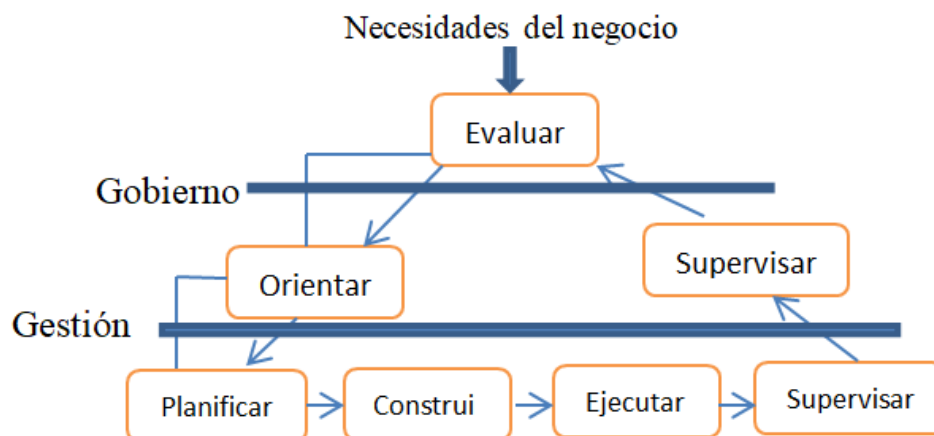


Figura 6 Áreas Clave de Gobierno y Gestión de COBIT 5

Elaboración: Autor. Fuente: ISSACA (2012) Cobit 5. Recuperado de: <https://articulosit.files.wordpress.com/2013/07/cobit5-framework-spanish.pdf>

2.2.2 Cobit 4.1

Cobit 4.1 es un marco de referencia de gobierno y control, publicado en el 2007, el cual enseña a gestionar las actividades y recursos tecnológicos con prácticas de gestión y administración (Periñán I. L., 2011).

El marco de trabajo Cobit 4.1 tiene como objetivo cubrir varias aristas del TI y está orientado a negocios y a procesos, basado en controles y mediciones (Periñán I. L., 2011).

2.2.3 Orientado al negocio:

Tiene una estructurada orientada a proveer lineamientos a proveedores de servicios, usuarios y auditores de TI, pero principalmente sirve como guía para las gerencias y dueños de procesos. (IT.GovernanceInstitute, 2007)

2.2.3.1 Orientado a procesos:

Plantea un modelo genérico de procesos distribuido de acuerdo a cuatro procesos:



Figura 7 Los 4 dominios de Interrelación de Cobit

Elaboración: Autor. Fuente: ISSACA (2007) .Cobit 4.1.

Recuperado de: <http://www.slinfo.una.ac.cr/documentos/EIF402/cobit4.1.pdf>

2.2.3.2 Planear y Organizar

Se refiere a la manera en que TI contribuye al logro de los objetivos del negocio. Estableciendo primero su visión y misión estratégica que debe alinearse a los objetivos institucionales para luego identificar los procesos que intervienen en todas las actividades de TI, asegurándose que sean medibles. (IT.GovernanceInstitute, 2007)

El objetivo de planear y organizar define los siguientes controles:

Código	Control Cobit 4.1 Planear y Organizar
PO1	Establecer un Plan Estratégico de TI
PO2	Definir la Arquitectura de la Información
PO3	Identificar la Dirección Tecnológica
PO4	Definir los Procesos, Organización y Relaciones de TI
PO5	Administrar la Inversión en TI
PO6	Socializar las Aspiraciones y la Dirección de la Gerencia
PO7	Administrar Recursos Humanos de TI
PO8	Gestionar la Calidad
PO9	Evaluar y Administrar los Riesgos de TI
PO10	Gerenciar Proyectos

Tabla 1 Controles Planear y organizar

Elaboración: Autor. Fuente: ISSACA (2007) .Cobit 4.1.
Recuperado de: <http://www.slinfo.una.ac.cr/documentos/EIF402/cobit4.1.pdf>

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

2.2.3.3 *Adquirir e implementar*

Para el éxito de las estrategias de TI, las soluciones deben ser identificadas, desarrolladas o adquiridas. Para luego ser implementadas e integradas a los procesos (IT.GovernanceInstitute, 2007)

2.2.3.4 *Entrega y dar soporte*

Se refiere a la entrega de los servicios, sean estos de administración de seguridad, soporte, administración de datos o instalaciones operativas. (IT.GovernanceInstitute, 2007)

2.2.3.5 *Monitorear y evaluar*

Todos los procesos implementados deben ser elaborados tomando en cuenta parámetros de medición y control (IT.GovernanceInstitute, 2007)

2.2.3.6 *Basado en controles*

Son 34 los objetivos de control establecidos por Cobit que cubren procesos y controles de aplicación. Cada proceso posee un objetivo de control de nivel alto y varios objetivos de control detallados. (IT.GovernanceInstitute, 2007)

2.2.3.7 *Guiado por mediciones*

Antes de la implementación es necesario una medición de la situación actual para determinar las mejoras y las herramientas necesarias para la mejora. Cobit define dos metas, la primera es un modelo de madurez que facilita la evaluación por medio de *benchmarking* y la segunda se basa en mediciones de desempeño para los procesos de TI. (IT.GovernanceInstitute, 2007)

2.2.4 Modelo de madurez con parámetros de Cobit 4.1

Cobit 4.1 propone un enfoque de modelo de madurez que se basa en Software Engineering Institute. El modelo está orientado a evaluar los procesos y la capacidad de desarrollo de software de una organización. (IT.GovernanceInstitute, 2007)

2.2.4.1 Niveles de Madurez

Para ubicar la madurez de un proceso se establecen algunos parámetros de evaluación como se muestra en la siguiente tabla.

Calificación	Parámetros de evaluación
0	No se aplican procesos administrativos en lo absoluto
2	Los procesos no tienen orden y son desorganizados
3	Los procesos siguen un patrón regular
6	Los procesos se documentan y se comunican
8	Los procesos se monitorean y se miden
10	Las buenas practicas se siguen y se automatizan

Tabla 2 Parámetros para la evaluación de madurez de procesos

Elaboración: Autor. Fuente: Espinoza (2014) Recuperado de <https://bibdigital.epn.edu.ec/handle/15000/10371>

2.2.4.2 Medición de parámetros.

Para medir la madurez de un proceso de Cobit se procede asignar una puntuación a cada control del grupo respecto a la matriz de parámetros de evaluación, donde 10 significa que las buenas prácticas se siguen y se automatizan y 0 que no se aplican procesos.

Con esto se identifica el grado de cumplimiento de cada control y brinda una pauta para determinar el nivel madurez de procesos actual (Espinoza Flores, 2014).

2.2.5 Elección del marco de referencia gobierno de TI

Cobit 4.1	Cobit 5
Cuenta con 4 dominios	Tiene 5 Dominios
Implementa 34 Procesos	Tiene 37 procesos, aumenta un proceso APO13 de Gestionar Seguridad
Incluye el cubo de COBIT 4.1 para los Criterios de la Información:	Incluye un modelo de información descriptivo sobre el nuevo framework provisto por ISACA
Objetivos de control y auditoría de fácil implementación	Prácticas de gestión y gobernanza con tiempos de implementación considerables
Se adapta a pequeñas empresas o PYMES	Está orientado a medianas y grandes empresas
Consigue mejoras en procesos de un área específica ya que sigue procesos organizados y procedimientos que dan valor al servicio.	Debe incluir a todas las áreas de la empresa. Sigue procesos organizados y procedimientos que dan valor al producto o servicio.
Poca adaptación a modelos tecnológicos	Mucha adaptación con los modelos tecnológicos
En el dominio APO , se observa mayor reorganización interna de los objetivos de control	Puede estar distribuido hasta en 5 procesos del mismo dominio.
Cuenta con un Pentágono de COBIT 4.1 para el Gobierno de TI	Nuevo dominio de gobierno denominado “EDM – Evaluar / Dirigir & Monitorear”:
Cuenta con 28 metas de TI	Cuenta con 17 metas de TI
Modelo de madurez de procesos se basa en Software Engineering Institute.	Modelo de madurez de procesos se basa “Process Capability Model”, orientado por “ISO/IEC 15504

Tabla 3 Características de Cobit 4.1 y Cobit5

Elaboración: Autor. Fuente: ISSACA (2012) Cobit 5. Recuperado de:
<https://articulosit.files.wordpress.com/2013/07/cobit5-framework-spanish.pdf>

De acuerdo al marco teórico investigado y al cuadro de características de Cobit 4.1 y Cobit 5 de la tabla (3) del apartado 2.2.5 se puede determinar que la mejor opción a implementar en el área de proyectos es el marco de trabajo Cobit 4.1, principalmente porque va dirigido a una sola área de TI y los resultados pueden ser palpables a corto plazo.

2.3 Metodologías de desarrollo de software

Según Gamboa (2018), el origen de las metodologías de desarrollo de software nacen luego del apareamiento de la ingeniería de software. El crecimiento de la demanda de desarrollo dio origen a la aparición de distintas metodologías que se enfocan en mejorar la eficiencia, productividad y competitividad, lo que se ve reflejado en la satisfacción del cliente final.

Las metodologías ágiles han tomado notoriedad en los últimos años, desplazando a las tradicionales, puesto que se enfocan en parámetros de buenas prácticas de desarrollo de software. Actualmente, existe evidencia que las metodologías ágiles otorgan un valor adicional al proyecto en lo referente a velocidad y entregas parciales del producto final (Osterweil, 2006).

2.3.1 Ventajas de las metodologías de desarrollo de software

La principal ventaja de las metodologías de desarrollo radica en que el proyecto es estructurado de acuerdo a una planeación inicial, de esta manera se puede gestionar cada tarea asignada (Sosa, 2016).

Las metodologías de desarrollo definen un marco general de actividades, las mismas que se describen a continuación:

- Desarrollo en fases: Los proyectos se dividen en diferentes tareas y cada una de ellas define sus propios actores y responsabilidades.
- Planeación del proyecto: Se identifican los objetivos del proyecto y se estructura la arquitectura de la aplicación para determinar los tiempos y recursos necesarios.
- Definición de entregables parciales y totales: De acuerdo a la planeación realizada se establece el tiempo de entrega, mismo que debe ser socializado al cliente para que exista la aprobación de las dos partes.
- Gestión de desarrollo y seguimiento: Todas las actividades definidas en la planeación deben ser gestionadas, para esto debe existir un líder de proyecto que realice el seguimiento de las actividades planteadas.

2.3.2 Comparativo de metodologías de desarrollo ágil más populares

Existen muchas metodologías de desarrollo de software, unas enfocadas a la programación en sí y algunas que toman en cuenta otros procesos adicionales al desarrollo, pero que garantizan que el producto final tenga una mejor recepción por parte del usuario final, es el caso de las metodologías ágiles (Jiménez G. M., 2016).

En la siguiente tabla se hace una comparativa de tres metodologías con mucha aceptación en el mercado de desarrollo de software. Se muestran algunos parámetros que ayudan a identificar la diferencia entre cada uno de ellas.

Parámetros	RUP	Scrum	DevOps
Framework	Análisis de diseño, orientados a objetos	Diseño basado en un proceso iterativo e incremental	Orientada a desarrollo y operación

Tipo de revisión	En cada fase se realiza una o más iteraciones, si no se termina una fase no se continua con la otra.	Sus pruebas se realizan al finalizar el proceso, enfatizando en la reutilización de los componentes.	Pruebas se realizan a lo largo de todo el proceso.
Enfoque	Iterativo	Iterativo	Iterativo
Ciclo	Ciclo formal se define a través de 4 fases, pero algunos flujos de trabajo pueden ser concurrentes.	Cada sprint (iteración) es un ciclo completo.	Ciclo orientado a 8 fases
Planificación	Plan de proyecto formal, asociado múltiples iteraciones con hitos intermedios planificados por fecha de entrega	Cada plan de la siguiente iteración se determina al final de la iteración actual	Define una etapa de planificación al inicio del ciclo
Alcance	El ámbito de aplicación está predefinido antes del inicio del proyecto pudiendo ser revisado durante el proyecto.	En vez de alcance, Scrum utiliza una cartera de proyectos, que se revalua al final de cada iteración (sprint).	El alcance se mide en cada fase
Los artefactos	Visión / ámbito de aplicación del define formatos en cada iteración	El único artefacto formal es el software operativo.	Un artefacto de sistematización por cada etapa
Tipo de proyecto / producto	Recomendado para grandes proyectos con complejidad media alta y tiempos extensos de desarrollo	Recomendado para las mejoras rápidas proyectos que no tengan tiempo de entrega limitado	Recomendado para proyectos grandes y medianos.
Facilidad de uso	Dirigido por casos de uso, estableciendo una buena arquitectura	Se basa en el un plan inicial adaptable a cambios, está basado en tareas con varias iteraciones	La sistematización de la metodología puede requerir grandes capacitaciones

Tabla 4 Cuadro comparativo metodologías populares.

Elaboración: Autor. Fuente: Torres (2016) Scrum, XP y RUP.

Recuperado de: <http://scrum-xp-rup-barrionuevo-torres.blogspot.com>

2.4 Metodología DevOps

Farías (2017) indica que el nombre de DevOps es acuñado por primera vez por Patrick Debois en 2009 en un evento tecnológico de nombre DevOps Days en Bélgica.

Muchos autores lo definen como una filosofía, otros como un movimiento y la mayoría como una metodología que abarca un conjunto de pasos que consideran el proyecto desde la idea inicial, el desarrollo, la puesta en producción y el seguimiento o mantenimiento (Alejandro, 2017).

Esta metodología reúne una serie de experiencias que han dado buenos resultados, debido a que entre sus fases no solo toma en cuenta al desarrollo, sino también a todo un conjunto de parámetros que mejoran la experiencia del cliente final (Alejandro, 2017).

DevOps no es una metodología de desarrollo estricta como pueden ser Scrum o RUP, es un conjunto de buenas prácticas, técnicas y modelos de trabajo que sirven como estrategia organizativa que sugiere una cultura de procesos.

2.4.1 Principios de DevOps

- *Objetivos orientados al cliente:* El cliente y su satisfacción es el norte de cada etapa desarrollo y operación de DevOps
- *Responsabilidad:* Se debe tomar en consideración que al optar por DevOps se debe garantizar el éxito en todo el ciclo de vida propuestas por la metodología
- *Mejora Continua:* Se debe optimizar los procesos empresariales y alinearlos con los de DevOps hasta que se convierta en una cultura organizativa
- *Automatiza todo:* Es un principio vital que busca que la operación cada vez sea automatizada en cada fase de la metodología.

- *Trabajar como un solo equipo:* Todos los roles que intervienen en el proceso de desarrollo deben estar alineados a los principios de DevOps para lograr la colaboración de todos.
- *Monitorear y probar:* Es importante que el equipo en todas sus fases pruebe y aseguren la buena práctica de procedimientos para alcanzar un producto con parámetros de calidad.
- *Integración continua:* Todas las áreas deben apoyar el proyectos siguiendo un objetivo común, tratando de que cada etapa se alinee con la siguiente logrando integrar el desarrollo en la operación y viceversa.

2.4.2 Acción centrada en el cliente

Las empresas que han optado por la utilización de DevOps como una metodología dentro de sus procesos tecnológicos han visto cambios significativos en sus resultados empresariales (Jiménez G. M., 2016).

Con la implementación de DevOps incrementa la satisfacción del cliente puesto que se obtienen mejoras en el rendimiento y la estabilidad, además otorgan un plus al producto final ya que brinda un enfoque de innovación en los proyectos.

2.4.3 Mejora Continua

DevOps en su etapa de Desarrollo permite la acción de mejoramiento en cada fase propuesta puesto que establece herramientas de mejora que ayuda a medir o mejorar los procesos, lo que origina que esa experiencia se traslade al siguiente hito de la metodología, la misma que debe implementar otras herramientas que procuren el mejoramiento de los procesos (Rayo, 2016).

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps



Figura 8 Fases de mejora continua
Elaboración: Autor. Fuente: Deducción propia

2.4.4 Ciclo de vida DevOps

DevOps define las siguientes fases dentro de su implementación:

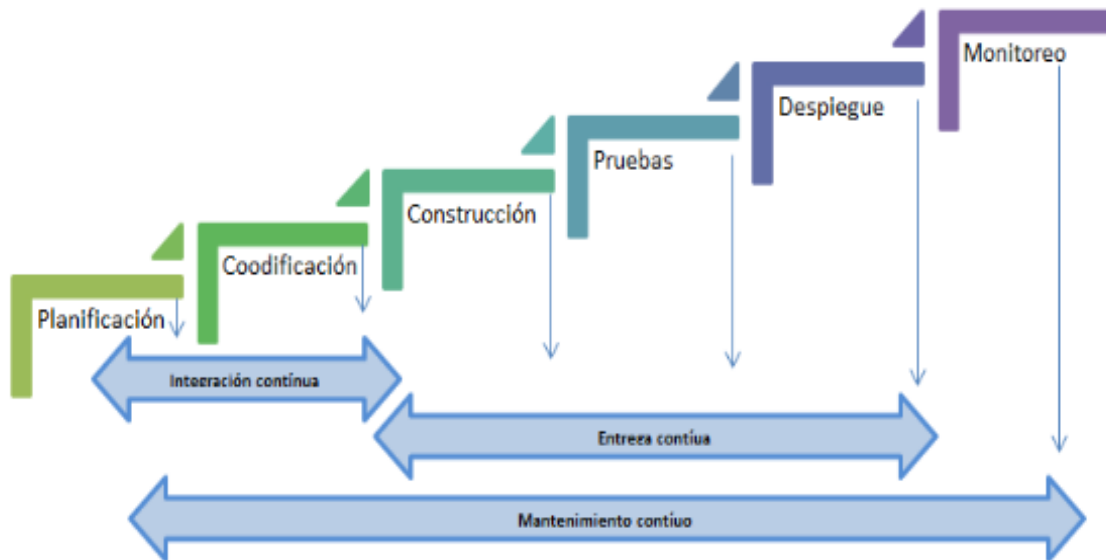


Figura 9 Fases definidas para DevOp.
Elaboración: Autor. Fuente: Pérez (2018) DevOps IT Development In.
Recuperado de: <http://uvadoc.uva.es/bitstream/10324/31469/1/TFG-I-994.pdf>

2.4.4.1 Planificación

En esta fase se establece el valor del negocio y los requisitos de la aplicación, se detallan los objetivos y las planificaciones técnicas del proyecto, con el fin de definir la viabilidad y crear el valor al producto final.

Es importante que la parte de la planificación se oriente a enfrentar cambios cortos que ayuden al grupo de desarrollo y operaciones a regular los nuevos cambios de una manera ágil (Pérez, 2018).

Actividades:

- Levantamiento de los requerimientos
- Métricas o reglas del negocio
- Actualización y elaboración de actas de meticas de la aplicación
- Calendarización y plan de construcción
- Políticas y requisitos de seguridad
- Construcción y arquitectura de la aplicación.

2.4.4.2 Codificación

En la segunda etapa de la metodología DevOps se lleva al código todas las funcionalidades antes especificadas en las reglas del negocio (Pérez, 2018). Dentro del desarrollo también se deben establecer parámetros de seguridad y la automatización de procesos como por ejemplo la construcción de repositorios de Web Services, accesos al Stash, integración entre los módulos y demás factores que garantizarán que la aplicación no tendrá ningún inconveniente para ser construida y llevada a calidad para su respectivo control de funcionamiento. En esta etapa es recomendable la utilización de un software que ayude a la gestión de tareas propias del desarrollo y su seguimiento.

2.4.4.3 Construcción

La etapa de construcción se refiere a las tareas donde se involucran todos los parámetros relacionados al despliegue de la aplicación (Pérez, 2018), mismos que deben estar listos para que no haya ningún inconveniente al momento de subir la aplicación al ambiente de pre producción, es por esto que, a esta etapa se la conoce también como pre producción esta fase incluyen tareas como:

- Preparación del hardware para la aplicación
- Creación de usuarios de aplicación y de base de datos
- Creación de empaquetados y disparadores
- Liberación y puesta en escena en pre producción.

2.4.4.4 Pruebas

La revisión de un sistema está directamente relacionada con la calidad de la versión de software (Mera, 2016), las tareas asociadas a garantizar el correcto funcionamiento de una aplicación se deben establecer en esta etapa. Todos los casos de prueba son necesarios para lograr un producto de éxito en cada compilación

Las principales activadas en esto son:

- Pruebas de sistema
- Pruebas de aceptación
- Pruebas de regresión
- Análisis de seguridad y vulnerabilidad.

2.4.4.5 Despliegue

El proceso clásico de desarrollo de software finalizaría en la fase de pruebas, pero DevOps incluye la Fase de Despliegue. En esta etapa se deben establecer los parámetros de control previos a la subida de las aplicaciones en los diferentes entornos (prueba, pre producción y producción), pudiendo ser desplegado de forma automática, pero siempre DevOps promueve la sistematización de los procesos (Bienvenido, 2018).

2.4.4.6 Lanzamiento

Etapa donde se realiza gestión de cambios, aprobaciones de versiones, automatización de versiones entre otras tareas referentes al versionamiento de software.

2.4.4.7 Operación

Esta fase se orienta a la configuración de hardware y parametrizaciones de la aplicación que se va a poner en producción garantizando que el usuario pueda operar la aplicación, además, debe asegurar la escalabilidad, persistencia, disponibilidad, redundancia y seguridad de la aplicación (Bienvenido, 2018).

2.4.4.8 Monitoreo

En esta fase se definen los parámetros de monitoreo para controlar que la aplicación se encuentra estable tanto en su funcionamiento como en su infraestructura, previo al paso a mantenimiento. Aquí se evalúa el sistema, consultando el histórico de las mediciones durante un periodo de tiempo establecido. Es decir, se analiza toda aquella información que impacte sobre la actividad del plan que incluye cambios, actualizaciones etc. (Bienvenido, 2018).

2.4.4.9 Sistematización

Si bien la sistematización no es un elemento del ciclo de vida de DevOps, es importante especificar que en la metodología en cada una de sus etapas sugiere la automatización de los procesos que intervienen tanto en el desarrollo como en la Operación de las aplicaciones, es por eso que sugiere una serie de herramientas de software que ayudarán en cada etapa del ciclo de vida de DevOps.

2.4.5 Elección de la metodología de desarrollo

Conforme al análisis comparativo entre las metodologías ágiles más populares realizado en el apartado 2.3.2, podemos establecer que para este trabajo la mejor elección es utilizar la metodología DevOps como marco de trabajo para el modelo de Gobierno de TI, puesto que esta metodología está orientada a todos los procesos y se puede enlazar con otras herramientas como Scrum, lo cual es muy útil en la fabricación de software sin embargo, no cubren la operación de las aplicaciones.

CAPÍTULO III

SITUACIÓN ACTUAL

3.1 Antecedentes

Corporación Favorita inició sus actividades hace aproximadamente 65 años, es una empresa de compra y venta de productos de primera necesidad, teniendo en la actualidad más de 160 locales en todo el Ecuador. La empresa empezó con un local en el Centro Histórico de Quito y con el paso del tiempo se han ido incorporando más locales y a su vez implementando las diferentes áreas administrativas, tal es el caso del área de Tecnología de la Información.

El área de Tecnología está formada por la Gerencia General de Tecnología y 4 sub Gerencias, dentro de estas se encuentra la Sub Gerencia de Proyectos de Software, cuya misión es proveer sistemas informáticos a las diferentes áreas de la Corporación.

El área de desarrollo de proyectos ha desarrollado un promedio de 40 proyectos de software entre desarrollo interno o con ayuda de empresas externas, por la demanda actual obliga a recurrir a fábricas de Software, pero siempre con colaboradores internos a cargo de la supervisión y seguimientos de los proyectos.

Los requerimientos son recibidos por el Sub gerente de Proyectos quién evalúa cada petición y según la urgencia y alcance del proyecto apoya con los levantamientos y elaboración de presupuestos respectivos para que las propuestas sean llevadas a los directivos de la Corporación, son quienes aprueban los sistemas que deben pasar a desarrollo.

Como se analizó en la situación actual, no existe una metodología o un conjunto de buenas prácticas que ayuden a los encargados de cada sistema a homologar los procesos de recepción, fabricación y puesta en producción de los sistemas, quedando a criterio de cada responsable la manera de hacer seguimiento de cada proyecto.

3.1.1 Levantamiento de información

Para el levantamiento de la información se elaboró un cuestionario con preguntas abiertas que se enfocan en determinar el nivel de madurez de procesos y los ejes de gobernabilidad existentes en el área de proyectos. Las preguntas fueron formuladas a la Sub Gerente de Proyectos de Corporación favorita el 25 de marzo 2019.

Las preguntas realizadas se detallan a continuación:

- a. ¿Tiene el área de proyectos un plan estratégico que esté ligado al plan estratégico de TI?
- b. ¿Existe una arquitectura bien definida para la estructura de proyectos?
- c. ¿Los colaboradores de su área tienen identificada la dirección tecnológica empresarial?
- d. ¿Las actividades que se realizan en la construcción de un sistema están definidas por procesos?
- e. ¿El área de proyectos administrar y hace seguimiento del presupuesto asignado?
- f. ¿Las directrices son socializar con todos los miembros de su equipo?
- g. ¿Existe intervención de la sub gerencia de Proyectos en la contratación del personal que se va a incorporar al área?
- h. ¿Existen parámetros de calidad que se sigan en la elaboración de software y se hace seguimiento de dichos controles?
- i. ¿Existe un plan de gestión de riesgos?
- j. ¿El área de proyectos tiene una estructura organizacional bien definida?
- k. ¿Conoce cada miembro de del equipo cuál es su rol y responsabilidad?
- l. ¿Para el desarrollo de las aplicaciones existe las bases de una metodología de desarrollo de software?

En base a la retroalimentación que se obtuvo de las entrevistas podemos precisar la información que se muestra a continuación.

3.2 Visión y misión

La visión y misión del área de Desarrollo de Proyectos está alineado a los objetivos del área de Tecnología de la Corporación el mismo que dice “Proveer soluciones informáticas a nuestros clientes internos y externos para contribuir en la consecución de los objetivos de la Corporación, con Talento Humano comprometido, trabajo en equipo liderazgo y planificación. La misión se integra con la misión corporativa que es “Ser la mejor cadena comercial de América”.

Tener clara la visión del área es un factor de gobernabilidad de TI importante ya que brinda un enfoque de hacia dónde debe estar orientado el modelo de Gobierno de TI necesario de implementar, aporta con una visión continua que cubre lo estratégico y lo operativo.

3.3 Organigrama

El organigrama del área de desarrollo de proyectos tiene una estructura jerárquica donde a la cabeza del todo el equipo está la Gerencia de TI, después se colocan las diferentes Sub gerencias y cada una de estas tiene sus propios equipos de trabajo.

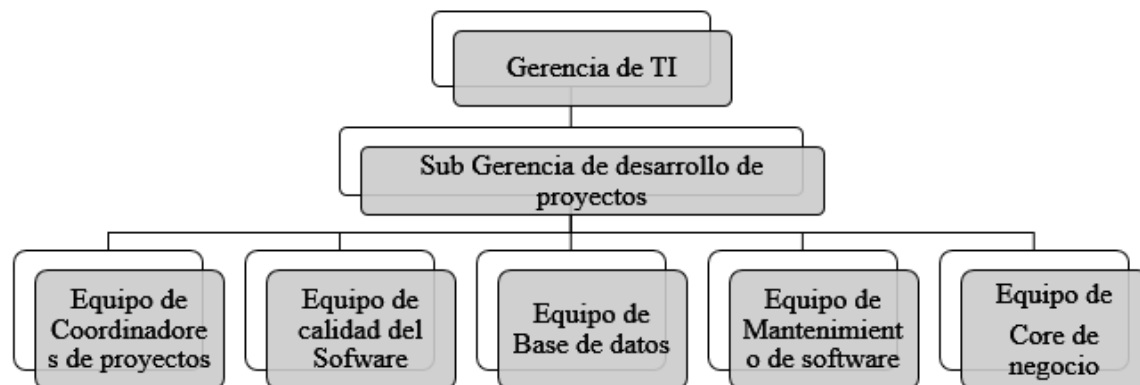


Figura 10 Organigrama de área de Desarrollo de proyectos

Fuente: Autor-Información tomada en base a entrevista realizada.

Tener un organigrama claro ayuda a cubrir en parte el aspecto de gobernabilidad referente a asignación de funciones y responsables, puesto que permite a gerentes y personal tomar decisiones rápidas lo que nuevamente puede ser una fuente de riesgos.

3.3.1 Arquitectura existente

Las aplicaciones desarrolladas a cargo del área de proyectos tienen la siguiente estructura de arquitectura.

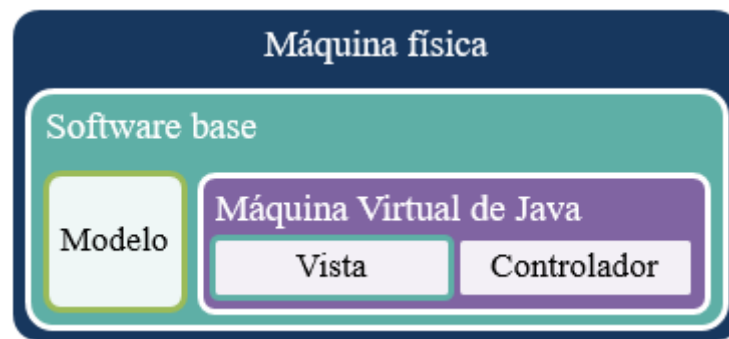


Figura 11: *Arquitectura actual de aplicaciones web*

Fuente: Autor-Información tomada del equipo de arquitectura de Corporación Favorita

La arquitectura existente en las aplicaciones desarrolladas es un modelo vista controlador, es decir, se separa los datos de la interfaz y la lógica de control usándose como componentes individuales. Este tipo de modelo de software es muy popular hoy en día y ha dado resultados positivos en las aplicaciones en uso de la Corporación. Esta es una razón para que la administración de las aplicaciones esté orientada al Desarrollo y la Operación, y DevOps ofrece un marco de trabajo referencial para el control de todas las actividades referentes a estos objetivos.

3.4 Nivel de madurez en procesos.

Para determinar los aspectos a medir en la matriz de levantamiento de la situación actual es necesario primero determinar el nivel de madurez en cuanto a proceso que tiene el área de proyectos para esto utilizamos los parámetros de mediciones de madurez basada en Cobit4.1 y aplicamos en la entrevista con la Sub gerente de proyectos.

En donde se obtuvieron los siguientes resultados:

Parámetros de evaluación aplicado	Calificación
No se aplican procesos administrativos	0
Los procesos no tienen orden	2
Los procesos siguen un patrón regular	3
Los procesos se documentan y se comunican	6
Los procesos se monitorean y se miden	8
Las buenas practicas se siguen	10

Tabla 5 Parámetros de evaluación para nivel de madurez de procesos

Fuente: Autor-Información tomada de entrevista realizada.

Código	Control Cobit 4.1 Planear y Organizar	Calificación
PO1	Establecer un Plan Estratégico de TI	8
PO2	Definir la Arquitectura de la Información	8
PO3	Identificar la Dirección Tecnológica	10
PO4	Definir los Procesos, Organización	4
PO5	Administrar la Inversión en TI	8
PO6	Socializar las Aspiraciones y la Dirección	10
PO7	Administrar Recursos Humanos de TI	6
PO8	Gestionar la Calidad	8
PO9	Evaluar y Administrar los Riesgos de TI	3
PO10	Gerencia Proyectos	10

Tabla 6 Controles Cobit 4.1

Elaboración: Autor. Fuente: *IT.GovernanceInstitute (2007)*

Recuperado de: <http://www.slinfo.una.ac.cr/documentos/EIF402/cobit4.1.pdf>

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

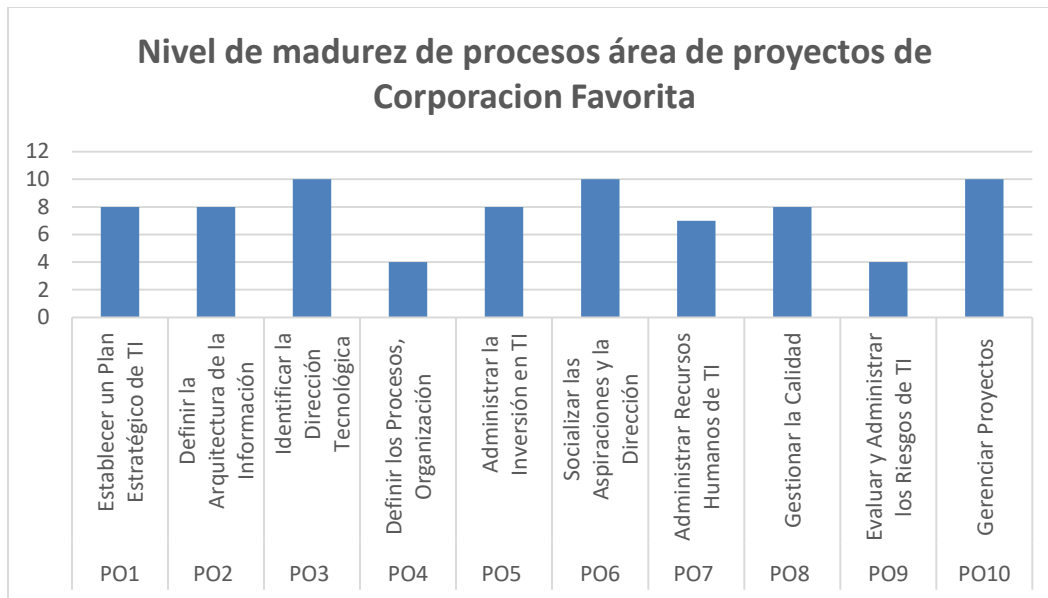


Tabla 7 Resultado de nivel de madurez del área

Fuente: Autor-Información recolectada en base a tabulación de datos

3.5 Matriz de levantamiento de situación actual

Para el levantamiento se realizó una encuesta a la Sub Gerente de Proyectos de la Corporación dónde se expusieron preguntas orientadas a los siguientes aspectos de Gobierno de TI:

- Estructura organizacional
- Marco de trabajo de procesos
- Definición de actores y roles.
- Definición de responsabilidades

Se escogieron estos parámetros, porque están relacionados al control de Cobit 4.1 P04 Referente a “Definir los procesos de la organización” el cual fue uno de los parámetros más bajos en el resultado de nivel de madurez del área.

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

La matriz presentada a continuación se levantó en el mes de marzo del 2019 y muestra los parámetros de Gobierno de TI evaluados al área de desarrollo de proyectos de Software. El detalle se refiere a la especificación de cada ítem y el nivel de cumplimiento, de acuerdo a dos parámetros (cumplimiento y seguimiento). Estos datos porcentuales se promedian para obtener un resultado y con esa información se pueden identificar los parámetros más críticos, por ende, tener un enfoque de la oportunidad de mejora.

Matriz de levantamiento de parámetros Cobit 4.1 Control PO4 (Marzo 2019)					
Objetivo de control (PO4)	Detalle	Nivel de cumplimiento			Promedio (%)
		Descripción	Cumplimiento %	Seguimiento %	
Estructura Organizacional (PO4.5)	Eficiencia TI	Uso de presupuesto de activos	100	100	100
		Costo mantenimiento de software	100	100	100
		IRP de proyectos en desarrollo TI	100	100	100
		Productividad	100	100	100
		Errores de los aplicativos en ambiente de producción	100	100	100
	Experiencia al cliente	Satisfacción del cliente interno	100	100	100
		Cumplimiento planificación anual de proyectos.	100	100	100
	Talento Humano	Nivel de expertiz en el área	100	100	100
	Innovación	N/A			
	País	N/A			
Marco de trabajo de Procesos (PO4.1)	Planeación	Se detalla los objetivos y las planificaciones técnicas del proyecto.	40	10	23
	Codificación	Se lleva al código todas las funcionalidades planeadas.	10	30	22
	Construcción	Creación objetos relacionados al despliegue de la aplicación.	25	10	15
	Pruebas	Garantizar el correcto funcionamiento de una aplicación	70	10	53

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

	Lanzamiento	Se establecer los parámetros de control previos a la subida de las aplicaciones	10	0	10
	Despliegue	Gestión los versionamiento de software	10	30	23
	Operación	Ajustes y parametrizaciones de la aplicación que ya se encuentra en producción	20	10	13
	Monitoreo	Paso de la aplicación a mantenimiento	10	0	7
Establecimiento de Roles (PO4.6)	Definición de Roles	Actores encargados de tareas dentro de los procesos	65	28	49
Establecimiento de Responsabilidades (PO4.6)	Asignación de responsabilidades sobre roles	Asignación de tareas a un actor.	60	35	60
			0-30	Muy urgente de implementar Urgente de Implementar Poco urgente o ya está implementado	
			31-60		
			61-100		

Tabla 8 Levantamiento de parámetros de gobernabilidad del área de desarrollo.

Fuente: Autor

3.6 Análisis de resultados

Una vez realizado el levantamiento presentado en el diagnóstico anterior (3.5) se puede identificar que los ejes más críticos encontrados son los referentes a procesos de Gobierno de TI, los actores y las responsabilidades de cada uno de ellos.

El área de TI investigada tiene clara la visión, misión y sus objetivos estratégicos, esto se debe a que en la Corporación se tiene implementada una metodología de mejora continua desde hace aproximadamente dos años. La misma que establece que cada área tenga identificados sus indicadores de gestión y el seguimiento mensual de los mismos (Figura 12).

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps



Figura 12 Ejemplo de tablero de indicadores de Gestión del área de Desarrollo
Fuente: Figura proporcionada por Sub Gerencia de proyectos

En lo referente a los procesos propios del área son los indicadores que necesitan de un modelo de Gobierno para potenciar los resultados obtenidos.

Objetivo de control (PO4)	Promedio de Calificación
Estructura Organizacional (PO4.5)	100
Marco de trabajo de Procesos (PO4.1)	20,75
Establecimiento de Roles (PO4.6)	49
Establecimiento de Responsabilidades(PO4.6)	60

Tabla 9 Promedio de cumplimiento de los parámetros de Gestión de TI.
Fuente: Autor.

En cuanto a metodologías de desarrollo de software la empresa usa una metodología para la construcción de los aplicativos que cubre únicamente las fases relacionadas a la programación, como se muestra en la siguiente figura:

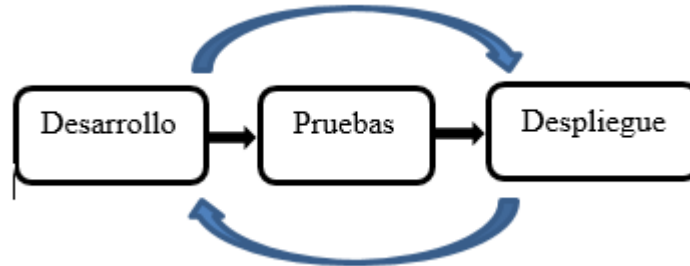


Figura 13 Modelo actual de desarrollo de aplicaciones

Fuente: Autor-Información tomada de entrevista realizada.

3.7 Oportunidad de mejora

Conforme al análisis del punto 3.6 se identifica que el modelo de Gobierno de TI basado en Cobit 4.1 debe estar basado a los siguientes parámetros:

- Identificación de los procesos del área enfocados en las fases de DevOps
- Identificar los actores de cada proceso
- Construcción de la matriz de caracterización de cada proceso
- Elaboración del diagrama de flujo del proceso
- Detalle del proceso
- Enfoque de cada lineamiento de gobernanza a las fases de la metodología DevOps

CAPÍTULO IV

MODELO DE GOBIERNO DE TI PROPUESTO

4.1 Arquitectura del modelo de gobierno

El modelo desarrollado está orientado a describir las políticas de gestión de tecnología de las diferentes actividades desarrolladas en el área de desarrollo de proyectos de Corporación Favorita, para esto se ha desarrollado una arquitectura basada en tres ejes fundamentales:

- Gobernanza de TI
- Seguridad TI
- Metodología DevOps
- Procesos internos del área.

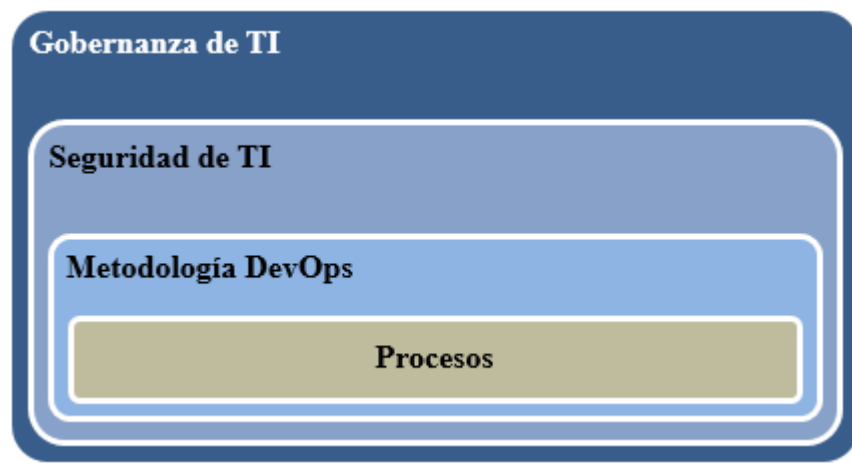


Figura 14 *Arquitectura del modelo de Gobernanza propuesto.*
Fuente: Autor- Figura proporcionada por Sub Gerencia de proyectos

Modelo General

El modelo propuesto tiene como enfoque central los parámetros de Gobierno de TI, buscando cubrir la mayoría de los procesos del área de Desarrollo de Proyectos, para esto se utiliza las buenas prácticas de la metodología DevOps. Los procesos actuales van a ser alineados a las fases de la metodología DevOps. Cada etapa de la metodología tiene un enfoque transversal con parámetros de seguridad y de automatización, que deben ser establecidos como políticas o como buenas prácticas para mitigar las brechas de seguridad en el desarrollo de las aplicaciones.

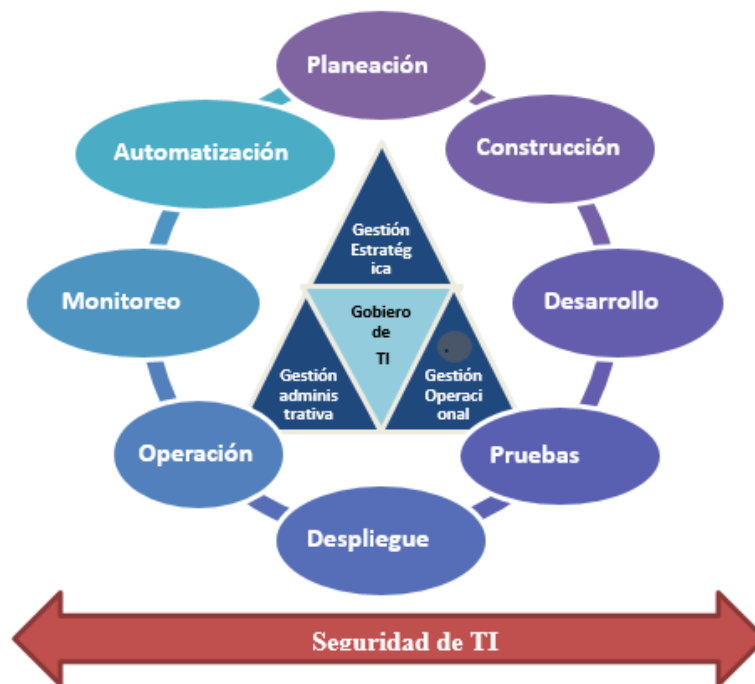


Figura 15 Modelo de parámetros de Gobernanza propuesto.

Fuente: Autor

Etapas del modelo propuesto:

4.2 Etapa de Planeación

Esta etapa describe todos los procesos relacionados con la recepción de requerimientos, planificación y aprobación de los proyectos de software. En la planeación se realizan tareas relacionadas a estimar tiempo, presupuesto y retorno de la inversión que se pretende realizar, de esta manera se puede analizar la factibilidad de empezar el desarrollo de un nuevo sistema.

4.2.1 Actores del proceso de planeación.

Actores del proceso	
Jefe de área (funcional)	Persona a cargo de un área de la Corporación o colaborador que solicita un requerimiento con la autorización del Jefe de área
Gerencia de TI	Gerente de Tecnología o Subgerentes de TI que reciben los requerimientos de desarrollo de aplicaciones
Analista técnico	Profesional técnico capacitado para realizar un levantamiento de requerimientos
Coordinador de proyectos SMX	Colaborador a cargo de organizar y hacer seguimiento de los sistemas en desarrollo.
Equipo de Arquitectura	Colaboradores a cargo de establecer los lineamientos y políticas para el desarrollo de las aplicaciones.
Comité de Proyectos	Grupo conformado por gerentes y directivos que dan paso o niegan la ejecución de un proyecto de desarrollo
Equipo de QA	Equipo de colaboradores que tiene como función probar las aplicaciones previas al despliegue en producción.

Tabla 10 Actores del proceso de Planeación.

Fuente: Autor

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

Diccionario:

- Flux: Sistema de gestión documental usado en Corporación Favorita.
- Confluence: Software de colaboración corporativa usado en Corporación Favorita.
- Jira: Software de gestión de tareas, basado en la metodología Scrum usado en Corporación Favorita.
- ROI: Plantilla de Excel dónde se establecen los rubros de un proyecto para determinar el costo y el tiempo de retorno de la inversión.

4.2.2 Caracterización del proceso de Planeación.

		CARACTERIZACIÓN DE PROCESO DE PLANEACIÓN		
Macro proceso:				
Código	Smx-001	Descripción	Proceso de Desarrollo y Operación de proyectos de software del área de TI	
Proceso:				
Código	Dev-001	Descripción	Procesos de planeación, análisis y aprobación de un requerimiento de software.	
Objetivo	Asignar un proyecto aprobado a las diferentes sub áreas del departamento de TI.			
Alcance	Recibir solicitudes, analizar el costo y tiempo de desarrollo, aprobar o descartar proyectos, asignar un responsable a un proyecto.			
Líder del proceso	Macro proceso	Gerencia de TI		
	Proceso	Gerencia de TI		
Actor	Entrada	Subprocesos	Salida	Herramienta
Gerencia de TI	Formato con Requerimiento de un nuevo proyecto	Levantamiento Macro de requerimientos	Presupuesto Inicial	Confluence Formato en Excel
Jefe de área (Funcional)	Presupuesto Inicial	Elaboración de ROI	Formato de ROI completo	Confluence Plantilla de Excel
			Acta de reunión	Confluence

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

Comité de proyectos		Formato de ROI completo		Aprobación o negación de proyectos			Acta
Requisitos aplicables							
Internos					Externos		
Reglamento interno de Corporación Favorita							
Indicadores							
Código	Nombre	Frecuencia	F análisis	R recolección	R análisis	Fuente	Valor aceptable
IRP-Dev-001	Atención de requerimientos solicitados	30 días	120 días	Reporte de solicitudes recibidas.	Gestión	Sistema Flux	80% del total de proyectos solicitados

Tabla 11 Caracterización del proceso de Planeación

Fuente: Autor

1.1.2 Flujo proceso solicitud y aprobación de un requerimiento de software

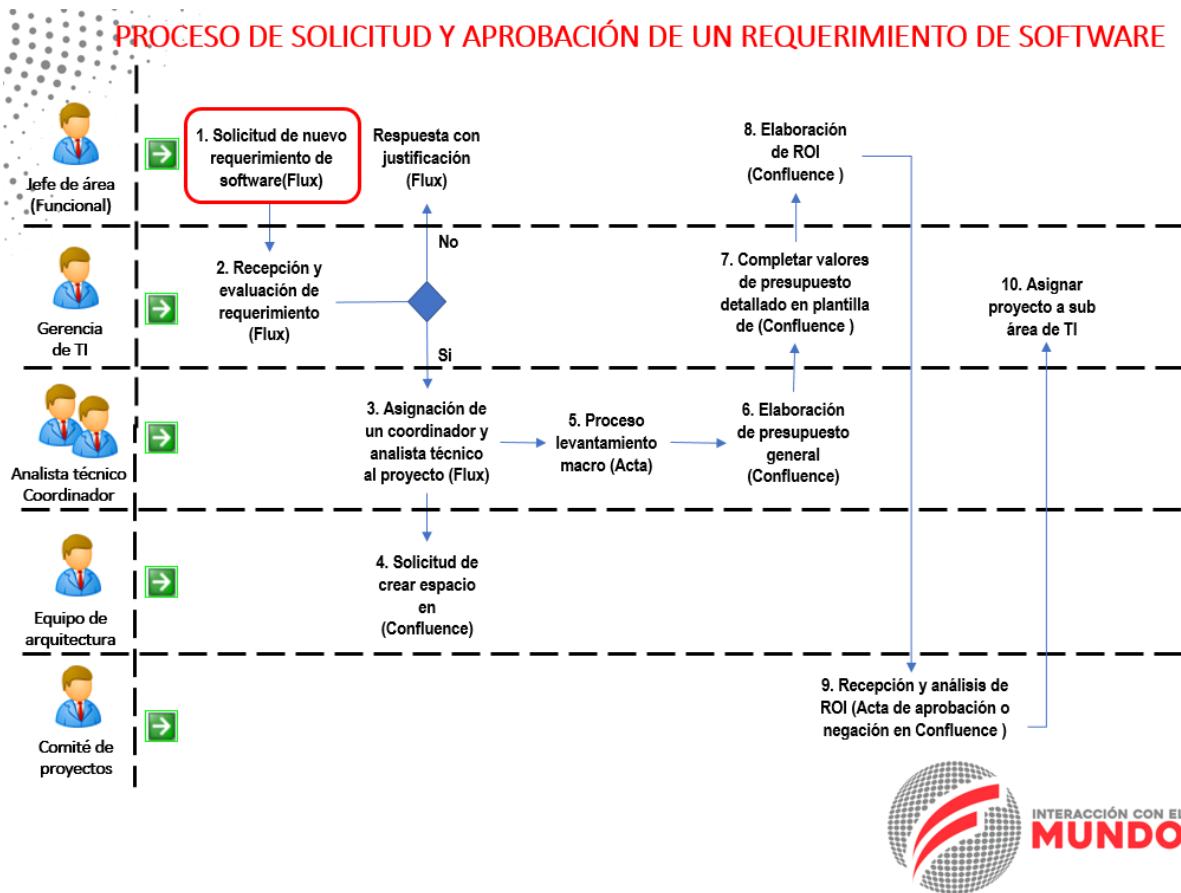


Figura 16 Flujo proceso de Solicitud y aprobación de un requerimiento de software.

Fuente: Autor

4.2.2.1 Detalle del proceso de aprobación de un requerimiento de software:

- 1. Solicitud de nuevo requerimiento de software (Flux):** El funcional envía un requerimiento a través de la herramienta Flux,
- 2. Recepción de Nuevo Requerimiento:** La Gerencia de TI o la Sub Gerencia autorizada recibe el formato de nuevo requerimiento, si se considera que el requerimiento de software es factible se asigna un coordinador y un líder técnico para continuar con el proceso.

3. **Asignación de un Coordinador y Analista técnico al proyecto (Flux):** La Gerencia o sub Gerencias de tecnología por medio de la herramienta Flux asignarán un coordinador a un proyecto, para que conjuntamente con el Analista Técnico cumplan con todas las etapas de la planificación.
4. **Solicitud de crear espacio en Confluence.** El coordinador del proyecto por medio de una tarea en Jira deberá solicitar al equipo de Core de Negocio la creación de un espacio en herramienta Confluence para anexar la información correspondiente a cada etapa del proyecto.
5. **Proceso de levantamiento macro:** El líder Técnico y el Coordinador deberán tener una o varias reuniones con el funcional, con la finalidad de establecer el tiempo en desarrollo que tomará el proyecto solicitado y el número de recursos necesarios.
6. **Elaboración de presupuesto general (Confluence):** Con la información recolectada en el levantamiento Macro, el Coordinador del proyecto deberá establecer el costo que tendrá el proyecto.
7. **Completar valores de presupuesto detallado en plantilla de Confluence.** La Sub Gerencia de Proyectos de desarrollo con la información brindada por el Coordinador respecto al costo de desarrollo del sistema llenará la Plantilla en Confluence correspondiente a costos del sistema donde además del presupuesto de desarrollo se estipulan valores como licencias servidores etc.
8. **Elaboración de ROI (Confluence).** El Funcional de proyecto con la información del presupuesto puede elaborar el archivo del ROI donde se establecen los costos y el tiempo de retorno de la inversión
9. **Recepción y análisis de ROI:** El comité de proyectos con la información de ROI podrán aprobar o negar el desarrollo del proyecto y a su vez el Coordinador deberá elaborar un acta donde consten los términos y causas con los que fue aprobado o negado el proyecto y esa misma acta colocarla en Confluence.
10. **Asignar sub área responsable del proyecto.** Una vez se aprueba un proyecto es asignado a una sub gerencia de TI para que se realice el seguimiento.

4.2.3 Flujo del sub proceso de levantamiento macro.

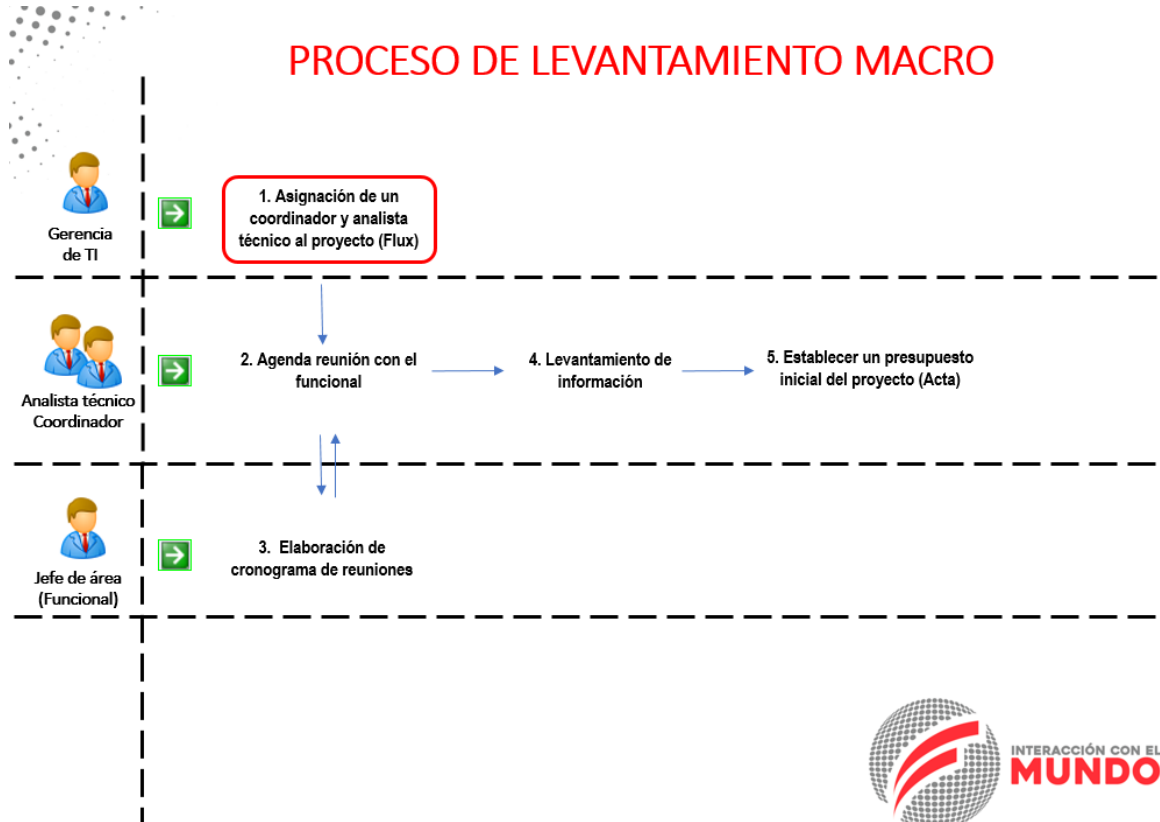


Figura 17 Flujo sub proceso de levantamiento macro.

Fuente: Autor

4.2.3.1 Detalle del sub proceso de levantamiento macro:

- 1. Asignación de un coordinador y analista técnico al proyecto (Flux):** La Gerencia o Sub Gerencias de TI por medio de la herramienta Flux asignarán un coordinador a un proyecto, para que conjuntamente con el Analista Técnico cumplan con el levantamiento Macro
- 2. Agenda reunión con el funcional.** El Analista Técnico deberá agendar una o varias reuniones para entrevistarse con el funcional y obtener la información relacionada al presupuesto general del proyecto.

3. **Elaboración del cronograma de reuniones:** Puede ser que por el tamaño del proyecto el levantamiento se lo tenga que hacer en varias reuniones, siendo así será necesario realizar un cronograma de reuniones donde los dos actores establezcan el compromiso de cumplir con los plazos.
4. **Levantamiento de información:** En base a las entrevistas mantenidas con el Funcional, el Analista Técnico deberá realizar un levantamiento general de los requerimientos solicitados por el Funcional, enfocándose en horas de trabajo y tiempo que llevará desarrollar el proyecto.
5. **Establecer un presupuesto inicial del proyecto:** Con la información recolectada en el levantamiento, el Coordinador del proyecto deberá establecer el costo que tendrá el proyecto referente a tiempo de desarrollo y número de recursos necesarios.

4.2.4 Flujo del sub proceso levantamiento específico

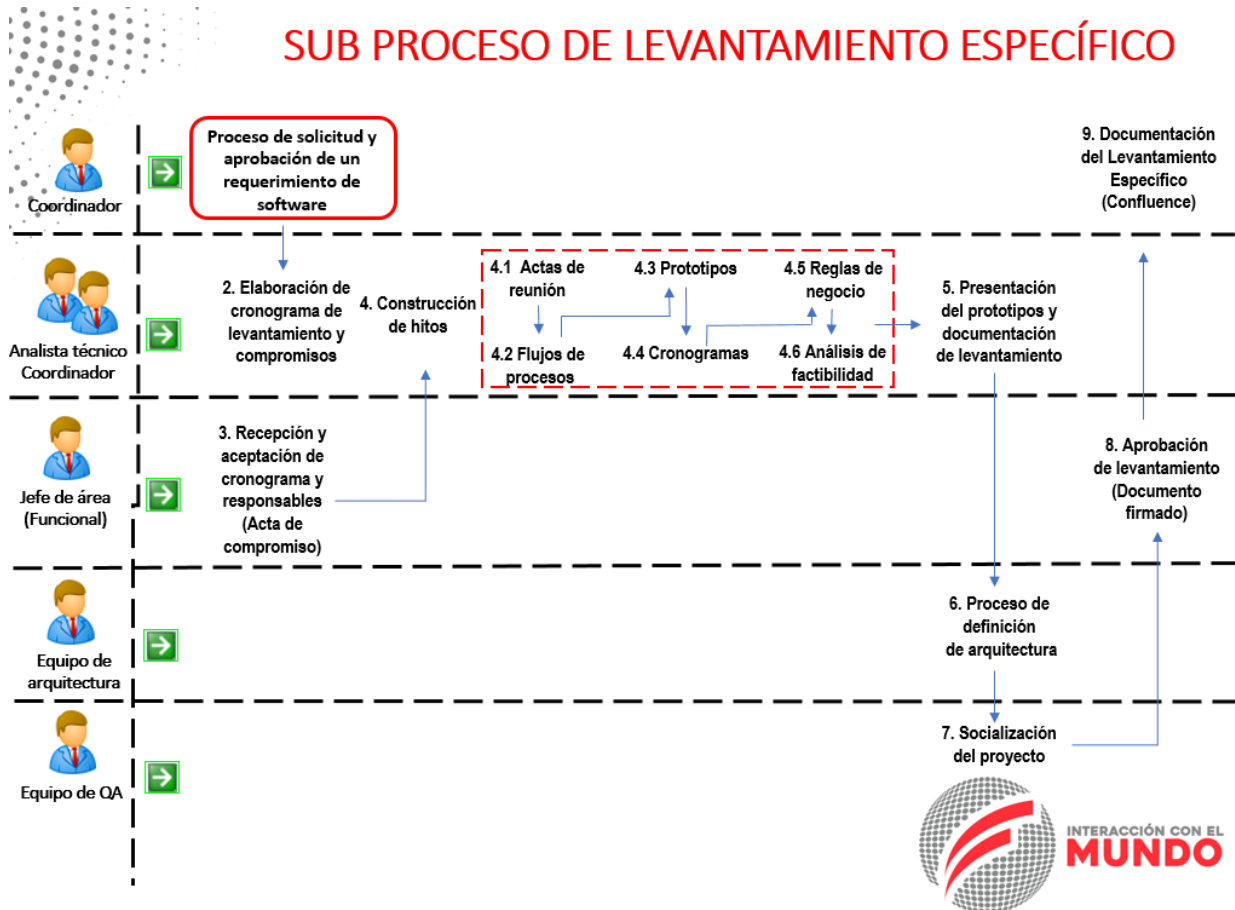


Figura 18 Flujo sub proceso de levantamiento específico

Fuente: Autor

4.2.4.1 Detalle del sub proceso de levantamiento específico:

- Asignación del proyecto y a la sub área para el levantamiento específico:** Una vez un proyecto es aprobado por el comité de proyectos, el mismo es asignado alguna sub área (Sub gerencia) de TI, dependiendo del objetivo del proyecto.

2. **Elaboración de cronograma de levantamiento y compromisos:** El levantamiento específico requiere conocer a detalle las necesidades del funcional por lo que es necesario establecer un cronograma de reuniones con fechas, tiempos y las personas que serán necesarias en el levantamiento de la información.
3. **Recepción y aceptación de cronograma y responsables del levantamiento:** EL cronograma deberá ser aceptado por las dos partes y asumir los compromisos de trabajo detallados, en caso de que no se cumpla con el cronograma el levantamiento quedará suspendido y se pasará a otro proyecto. Este compromiso debe contar con la firma de del funcional en un acta de compromiso.
4. **Construcción de hitos:** EL levantamiento debe cumplir por varias etapas conocidos como hitos donde se realizan una serie de procedimientos como: flujos de procesos, prototipos, reglas de negocio etc., Todo esto para llegar a determinar el detalle de los requerimientos del sistema y la documentación que respalde el trabajo realizado.
5. **Presentación los prototipos y documentación de levantamiento:** Toda la información generada en el levantamiento específico es necesaria para definir los prototipos de las pantallas, los cuáles deben ser expuestos al funcional para su aceptación o posibles cambios.
6. **Proceso de definición de arquitectura:** En este proceso se definen los lineamientos referentes a la arquitectura tecnológica que debe tener el sistema a desarrollar. El equipo de arquitectura deberá documentar la arquitectura de software, hardware y el modelo de datos que deberá tener la aplicación.
7. **Socialización del proyecto:** El Analista Técnico y el Coordinador del proyecto deberán brinda al equipo de QA un resumen del objetivo y los requerimientos del sistema y a su vez proporcionar acceso al levantamiento específico.
8. **Aprobación de levantamiento:** Se debe realizar un acta con los compromisos y firmas, en el cuál debe constar que Funcional está de acuerdo con los requerimientos levantados, entendiendo que cualquier cambio posterior al levantamiento puede incurrir en cambios de cronogramas y en aumento el presupuesto.

- 9. Documentación del levantamiento específico.:** EL Coordinador del proyecto es el encargado de supervisar que la información referente al levantamiento esté disponible en la plataforma Confluence.

1.1.1 Flujo del sub proceso de definición de arquitectura.

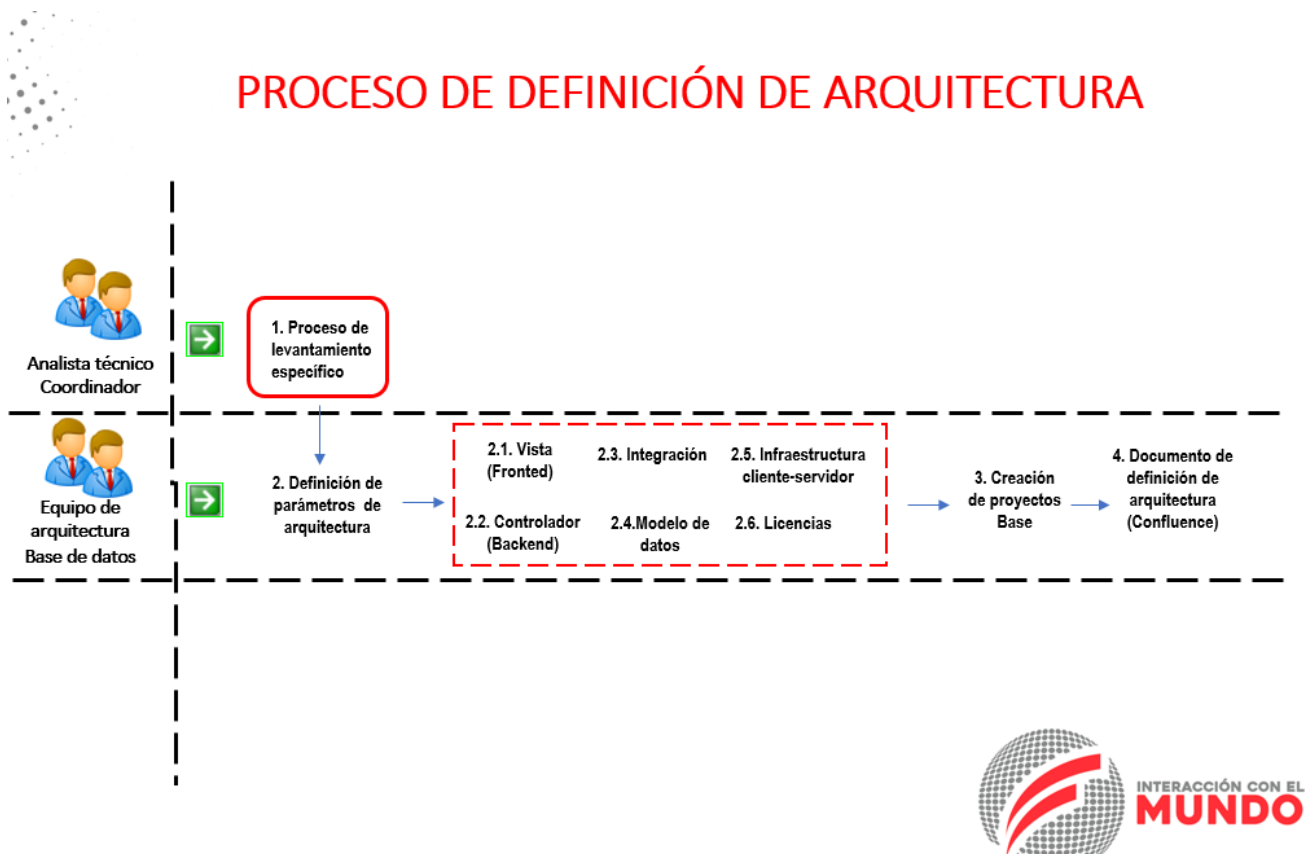


Figura 19 Flujo del sub proceso de definición de arquitectura.

Fuente: Autor

4.2.4.2 Detalle del sub proceso definición de arquitectura

- 1. Proceso de levantamiento específico:** Proceso donde se define el detalle de los requerimientos, así como los prototipos de las pantallas, los cuáles fueron aprobados por el funcional.

2. **Definición de parámetros de arquitectura:** El equipo de arquitectura define los estándares de arquitectura que debe tener un proyecto, pero siempre orientado a la arquitectura estándar de las aplicaciones existentes en la Corporación.
3. **Creación de proyectos Base:** El equipo de arquitectura deberá crear los proyectos base y los accesos necesarios a los mismos. Según el tipo de aplicación se deberá dar los estándares referentes a los desarrolladores.
4. **Documento de definición de arquitectura:** La definición de arquitectura debe plasmarse en un documento el cuál debe estar cargado en Confluence, los lineamientos deben ser claros para que el proveedor de software pueda interpretarlos.

4.2.5 Flujo del sub proceso de selección de proveedor

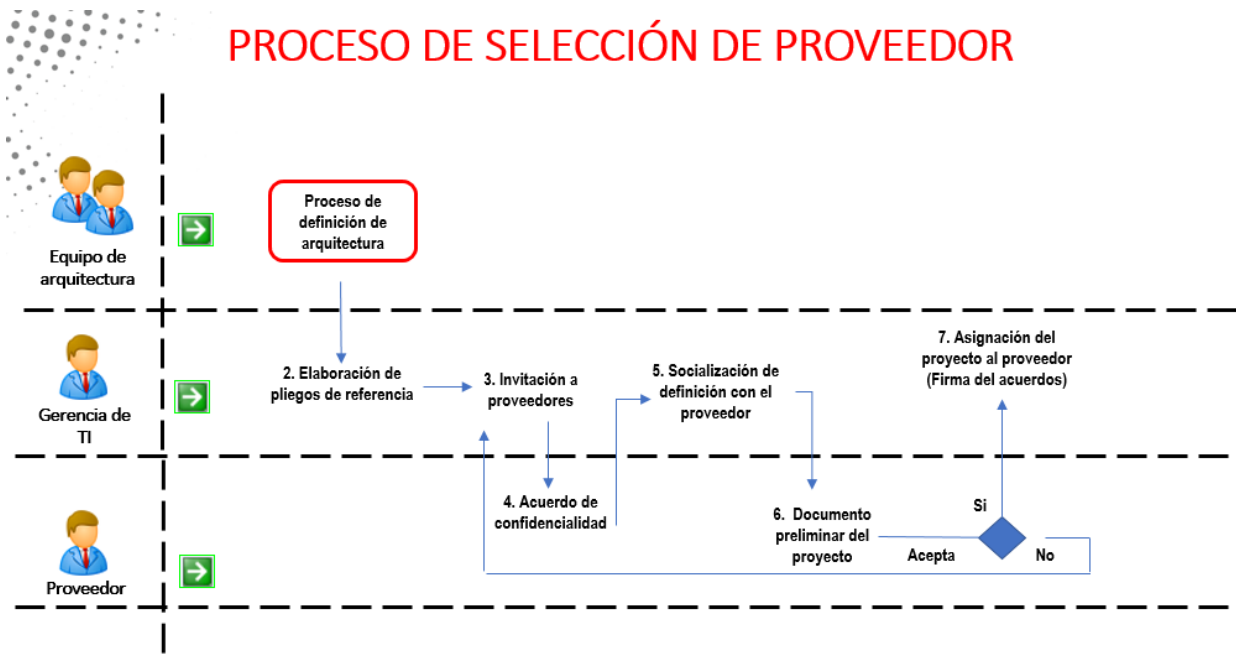


Figura 20 Flujo del sub proceso selección de proveedor
Fuente: Autor

4.2.5.1 *Detalle del sub proceso de selección del proveedor*

1. **Proceso de definición de arquitectura:** Proceso donde se define los lineamientos de arquitectura de un nuevo proyecto y plasmado en un documento cargado en Confluence.
2. **Elaboración de pliegos de referencia:** Los pliegos serán los requerimientos a detalle del sistema que se va a desarrollar, se deberá anexar las reglas de negocio y la definición de arquitectura, las reglas de negocio y toda la información necesaria para establecer el objetivo y alcance del proyecto.
3. **Invitación a proveedores:** La gerencia de TI es la encargada de elegir los proveedores y brindará la información respectiva del proyecto.
4. **Acuerdo de confidencialidad:** Previo a la socialización del proyecto al proveedor deberá firmar un acuerdo de confidencialidad sobre el proyecto que se le va a presentar.
5. **Socialización de definición con el proveedor:** Se deberá tener una o varias reuniones con el proveedor para explicar el objetivo, el alcance y las necesidades y lo que se espera en cuanto al desarrollo de la aplicación.
6. **Documento preliminar del proyecto:** Toda la información referente a los detalles del sistema (reglas de negocio, definición de arquitectura, prototipos), entre otros, serán entregados al proveedor en una socialización para su análisis.
7. **Asignación del proyecto al proveedor (Firma del acuerdos).** El proyecto es asignado al proveedor que acepto cumplir con todos los requerimientos del sistema, esto será formalizado con un acuerdo de aceptación entre las dos partes con los compromisos de trabajo.

4.2.6 Flujo del sub proceso arranque de un proyecto

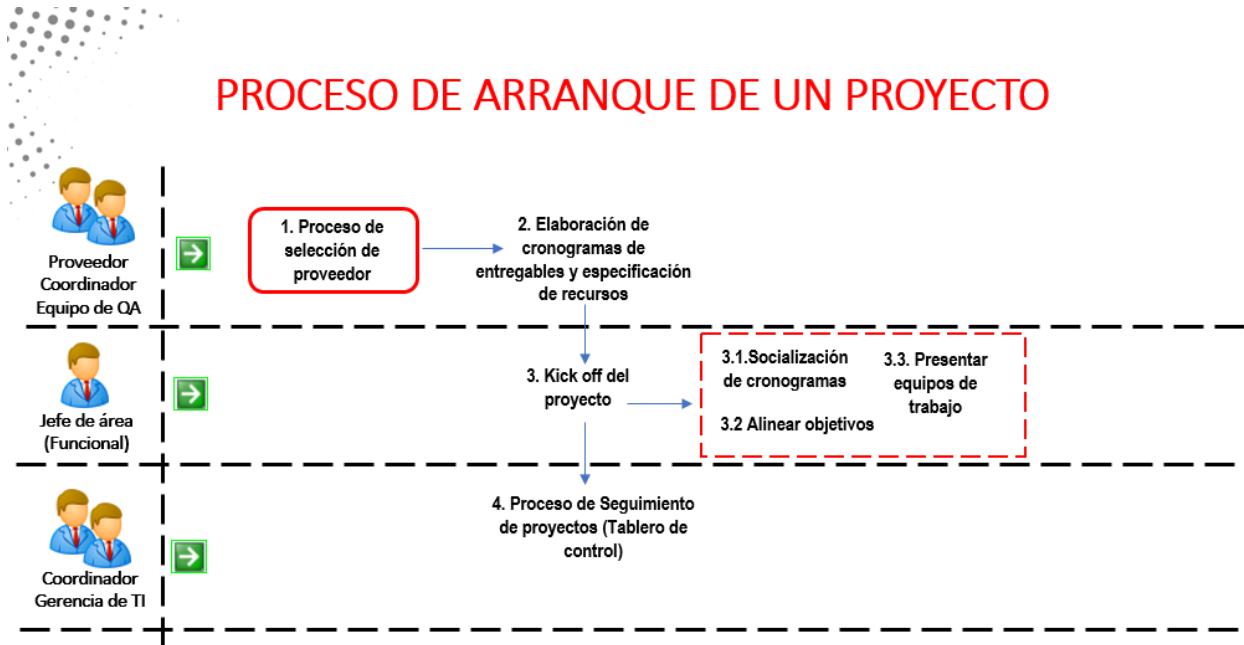


Figura 21 Flujo del sub proceso de arranque de un proyecto.
Fuente: Autor

4.2.6.1 Detalle del sub proceso de arranque de un proyecto

- 1. Proceso de selección de proveedor:** Proceso donde se selecciona un proveedor para el desarrollo un proyecto.
- 2. Elaboración de cronogramas y especificación de recursos:** El coordinador del proyecto junto el proveedor y el equipo de QA deberán detallar en un cronograma los tiempos de desarrollo, pruebas de calidad, fechas de entregables parcial o total, número de recursos entre otros aspectos.
- 3. Kick off del proyecto (Reunión de arranque del proyecto):** Con las definiciones de las reglas de negocio y los tiempos de desarrollo se puede dar paso a la etapa de codificación, este paso se lo debe socializar con el Funcional del proyecto, el mismo que debe conocer al proveedor y estar de acuerdo con los tiempos de entrega.

4. **Proceso de seguimiento de un proyecto:** Proceso mediante el cual cada coordinador de proyectos hace seguimiento de la codificación de las aplicaciones, cuidando que los tiempos de entrega y las funcionalidades de los sistemas se esté cumpliendo conforme a lo planeado.

4.3 Etapa de Codificación

La etapa de codificación está orientada a la definición del desarrollo de software que debe tener una aplicación, si bien es un proceso para el desarrollo interno de la Corporación se puede sugerir a los proveedores la utilización del mismo. La codificación integra las tareas relacionadas a la planificación de las iteraciones de codificación y luego la puesta de la aplicación en el ambiente necesario.

4.3.1 Actores del proceso codificación

Actores del proceso de codificación	
Jefe de área (funcional)	Persona a cargo de un área de la Corporación o colaborador que solicita un requerimiento con la autorización del Jefe de área
Coordinador de Proyectos SMX	Colaborador a cargo de organizar y hacer seguimiento de los sistemas en desarrollo.
Equipo de Arquitectura	Colaboradores a cargo de establecer los lineamientos y políticas para el desarrollo de las aplicaciones.
Sub área de TI	Área de TI que recibe un proyecto de software a su cargo
Líder técnico desarrollo	Profesional técnico que se encarga de administrar las tareas de desarrollo con el equipo de desarrolladores
Infraestructura SMX	Grupo de colaboradores a cargo de administrar la plataforma tecnológica de la Corporación
Equipo de QA (Calidad)	Equipo de colaboradores que tiene como función probar las aplicaciones previas al despliegue en producción.

Tabla 12 Actores del proceso de Codificación.

Fuente: Autor

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

Diccionario:

- Flux: Sistema de gestión documental usado en Corporación Favorita.
- Confluence: Software de colaboración corporativa usado en Corporación Favorita.
- Jira: Software de gestión de tareas, basado en la metodología Scrum usado en Corporación Favorita.
- Ambiente: Infraestructura tecnológica donde se despliega una aplicación.
- Hitos: Serie de etapas dentro de un mismo proyecto de desarrollo
- Git Flow: Conjunto de extensiones de GIT (Controlador de versiones) que facilitan la gestión de ramas y flujos de trabajo.
- Kambas: Método o técnica para gestionar el trabajo de desarrollo con énfasis en la entregas parciales
- Iteraciones: Diferentes pasos a seguir en el proceso de desarrollo conforme a una metodología establecida.
- Sprint: Ciclos o iteraciones que vamos a tener dentro de dentro de un proyecto Scrum.

4.3.2 Caracterización del proceso de Codificación

CARACTERIZACIÓN DE PROCESO DE CODIFICACIÓN			
Macro proceso:			
Código	Smx-001	Descripción	Proceso de Desarrollo y Operación de proyectos de software del área de TI
Proceso:			
Código	Dev-002	Descripción	Proceso de Codificación
Objetivo	Codificar en un lenguaje de programación los requerimientos indicados en la etapa de Planeación.		
Alcance	Ejecutar los Springs de tareas de desarrollo de la aplicación y hacer seguimiento del avance de cada proyecto.		

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

Líder del proceso	Macro proceso		Gerencia de TI				
	PROCESO		Equipo de desarrollo /Coordinador				
Actor	Entrada		Subprocesos		Salida	Herramienta	
Sub área de TI	Tablero de control de proyecto		Planificación y ejecución de iteraciones (springs)		Solicitud de configuración de ambiente	Flux	
Infraestructura	Solicitud de configuración de ambiente		Configurar Software en servidor		Solicitud cerrada	JIRA	
Coordinador	Accesos a un equipo (servidor) configurado		Realizar controles de funcionalidad		Acta entrega de los módulos o sistemas codificados	(We transfer Corporativo)	
Requisitos aplicables							
Internos					Externos		
Reglamento interno de Corporación Favorita							
Indicadores							
Código	Nombre	Frecuencia	Frecuencia de análisis	Recolección	Análisis	Fuente	Valor aceptable
IRP-Dev-002	Cumplimiento de desarrollo	30 días	30 días	Matriz de seguimiento de los coordinadores	Gestión	Tablero de control de proyecto	80 % de los proyectos planificados.

Figura 22 Caracterización del proceso de Codificación.
Fuente: Autor

4.3.3 Flujo proceso de codificación

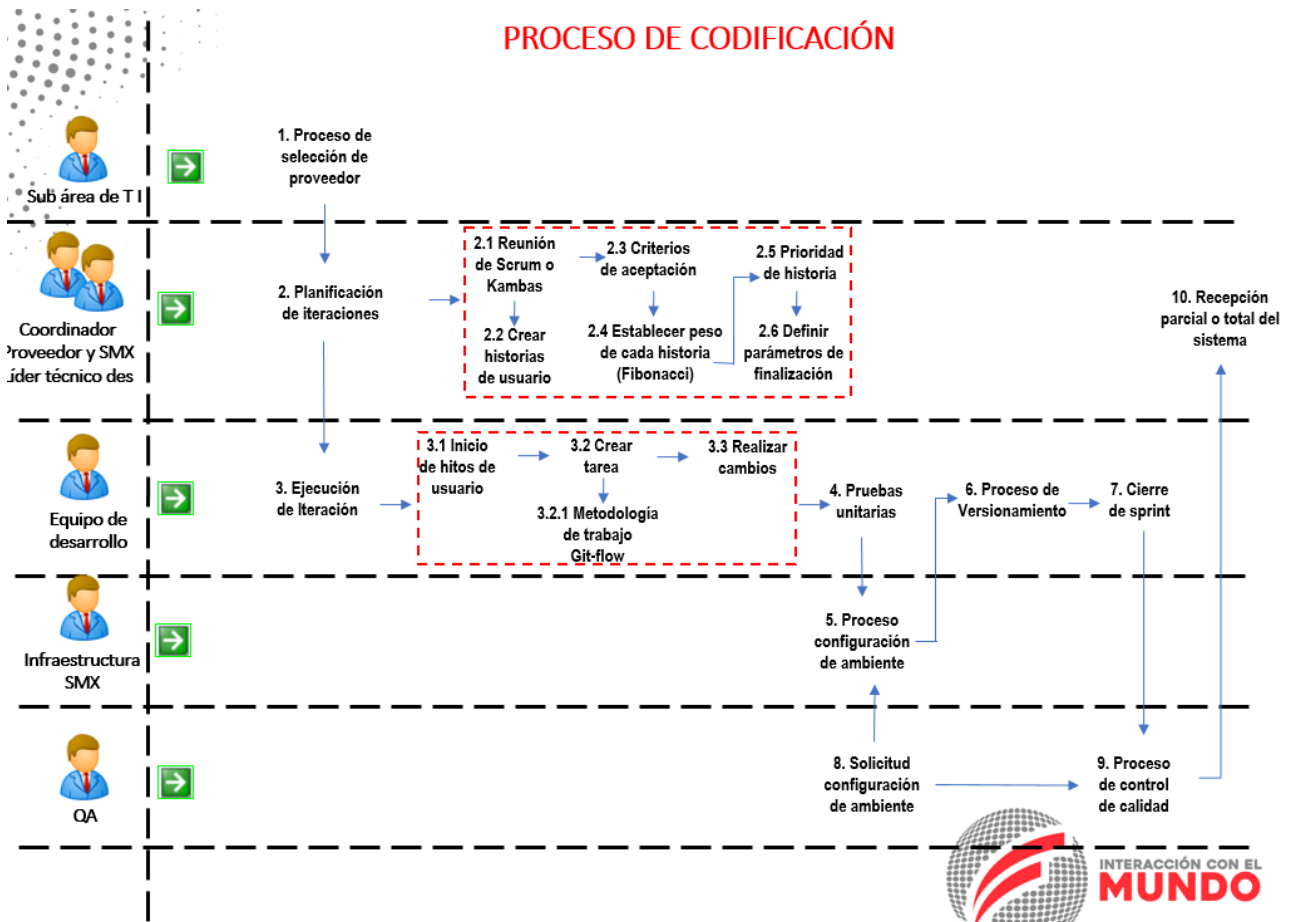


Figura 23 Flujo proceso de Codificación.

Fuente: Autor

4.3.3.1 Detalle del Proceso de codificación

- 1. Proceso de selección de proveedor:** Proceso mediante el cual la Gerencia de TI seleccionó al proveedor.
- 2. Planificación de iteraciones:** El líder técnico debe organizar las iteraciones de la programación, estas deben estar alineadas a los cronogramas establecidos para cumplir con los tiempos de entrega.

3. **Ejecución de Iteración:** La ejecución de iteraciones debe contemplar el cumplimiento de las etapas propuesta por la metodología de desarrollo usadas en este caso con Scrum (hitos, tareas, ejecución de Git Flow) entre otras.
4. **Pruebas unitarias:** Una vez cumplidas las iteraciones se debe hacer pruebas unitarias al código desarrollado.
5. **Proceso configuración de ambiente:** Proceso mediante el cual se configura la infraestructura de hardware necesaria para probar la aplicación desarrollada o el módulo cumplido.
6. **Proceso de Versionamiento:** Procesos donde se realizan las pruebas de control de código estático previo para luego realizar el versionamiento de una aplicación.
7. **Cierre de sprint:** Una vez se han realizado las pruebas unitarias se deben cerrar las iteraciones o Sprints, con esto se da cumplimiento al requerimiento de programación planificado y se da paso a los siguientes procesos.
8. **Solicitud configuración de ambiente:** El equipo de QA también deberá solicitar el ambiente necesario para realizar las pruebas de calidad.
9. **Proceso de control de calidad:** Proceso en el cual se realizan las pruebas necesarias de calidad de software, previas a su paso a producción.
10. **Recepción total o parcial de la aplicación:** Una vez la aplicación cumpla con los lineamientos de calidad de software de la Corporación, y si no se hallan falencias de código detectadas por software como Sonar se podrá recibir el sistema o el módulo para dar paso al despliegue de la aplicación en el ambiente de Producción.

4.3.3.2 Flujo del proceso de seguimiento de proyectos

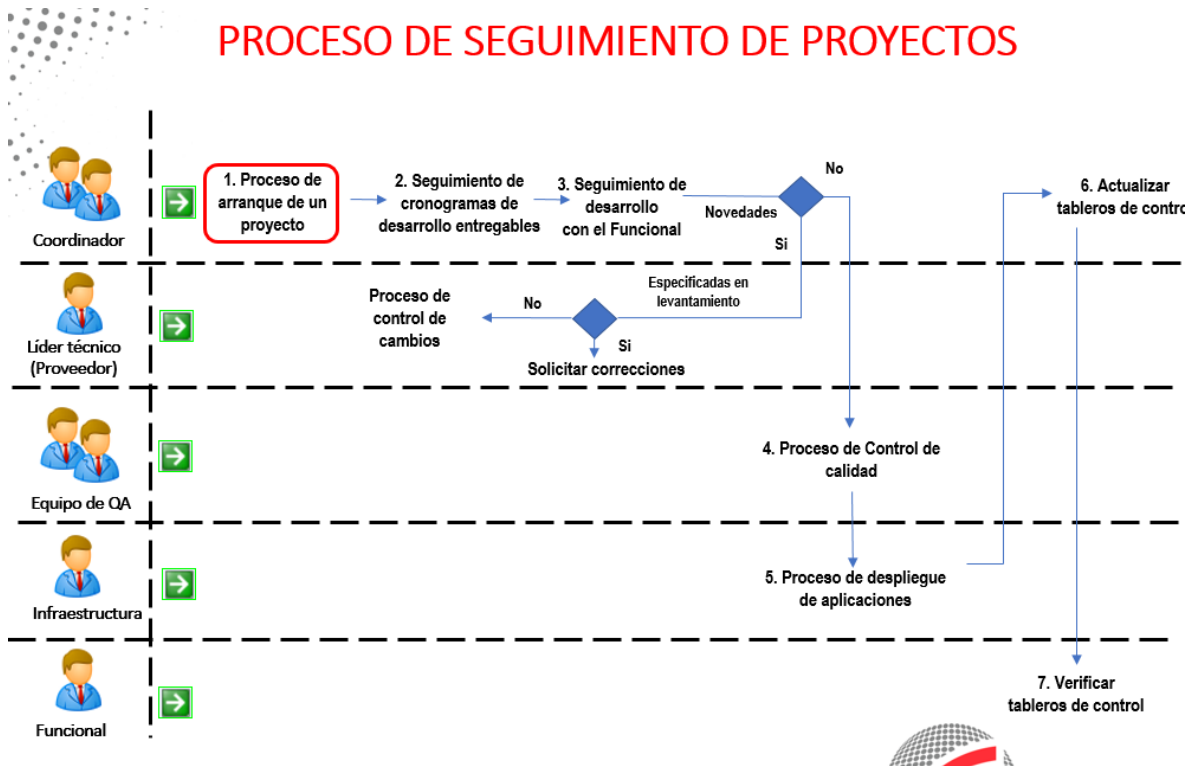


Figura 24 Flujo del proceso de seguimiento de proyectos.

Fuente: Autor

4.3.3.3 Detalle del Proceso de seguimiento de proyectos

- 1. Proceso de arranque de un proyecto:** Proceso donde se definen los calendarios de entregas totales o parciales del proyecto y se da el arranque de la codificación.
- 2. Seguimiento de cronogramas de desarrollo entregables:** El coordinador tendrá que hacer seguimiento del avance de los proyectos con el proveedor a cargo, con esto se podrá determinar con antelación si se están cumpliendo los tiempos establecidos.
- 3. Seguimiento de desarrollo con el Funcional:** Una vez terminado un módulo o una funcionalidad de un sistema, el Coordinador deberá probar su funcionalidad para cuando esté estable sea presentado al funcional para su aprobación.
- 4. Proceso de Control de calidad:** Proceso en el cuál se realizan las pruebas necesarias de calidad de software, previas a su paso a producción.

5. **Proceso de despliegue de aplicaciones:** Proceso donde se establecen los parámetros y pasos necesarios para el despliegue de una aplicación.
6. **Actualizar tableros de control:** Una vez la aplicación está en el ambiente de producción. Los coordinadores deberán actualizar los tableros de control de los proyectos para que se vaya establecido el progreso de los mismos.
7. **Verificar tableros de control:** El funcional podrá consultar los tableros de control para verificar la evolución del proyecto en la etapa de codificación.

4.4 Etapa de Construcción

La etapa de construcción está definida por la creación y ejecución de los diferentes componentes lógicos que la aplicación necesita para salir a producción, el más importante es la creación de objetos de base de datos como por ejemplo tablas, índices, constraints, definición de claves primarias y todos los componentes necesarios que interactúan con el lenguaje de programación.

Los objetos en la base de datos son creados por el administrador de la misma por medio de scripts, estos deben responder al modelo de datos definido en la arquitectura del proyecto.

4.4.1 Actores del proceso de construcción

Actores del proceso de construcción	
Líder técnico desarrollo	Profesional técnico que se encarga de administrar las tareas de desarrollo con el equipo de desarrolladores
Infraestructura SMX	Grupo de colaboradores a cargo de administrar la plataforma tecnológica de la Corporación

Tabla 13 Actores del proceso de construcción

Fuente: Autor

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

4.4.2 Caracterización del proceso de Construcción

			CARACTERIZACIÓN DE PROCESO DE CONSTRUCCIÓN				
Macro proceso:							
Código	Smx-001	Descripción	Proceso de Desarrollo y Operación de proyectos de software del área de TI				
Proceso:							
Código	Dev-003	Descripción	Proceso de construcción de parámetros lógicos para la implementación de las aplicaciones.				
Objetivo	Crear todos los objetos lógicos necesarios en la base de datos y comprobar su correcta estructura.						
Alcance	Recibir la solicitud de ejecución de scripts, verificar los objetos a crearse, ejecutar la tarea y comprobar que no se hayan generado errores.						
Líder del proceso		Macro proceso		Gerencia de TI			
		PROCESO		Infraestructura SMX			
Actor	Entrada	Subprocesos			Salida	Herramienta	
Infraestructura	area de ejecución de scripts	verificar estructura lógica de los scripts, ejecutar la tarea en base de datos.			Tarea de ejecución de script cerrada		
Requisitos aplicables							
Internos					Externos		
Reglamento interno de Corporación Favorita							
Indicadores							
Código	Nombre	Frecuencia	F Análisis	R recolección	R análisis	Fuente	Valor aceptable
IRP-Dev-003	Alteración de bases de datos	30 días	30 días	Reportes de tareas de ejecución de scripts	Gestión	Reportes de JIRA	100% de las tareas solicitadas

Tabla 14 Caracterización proceso de Construcción

Fuente: Autor.

4.4.3 Flujo del proceso de ejecución de scripts

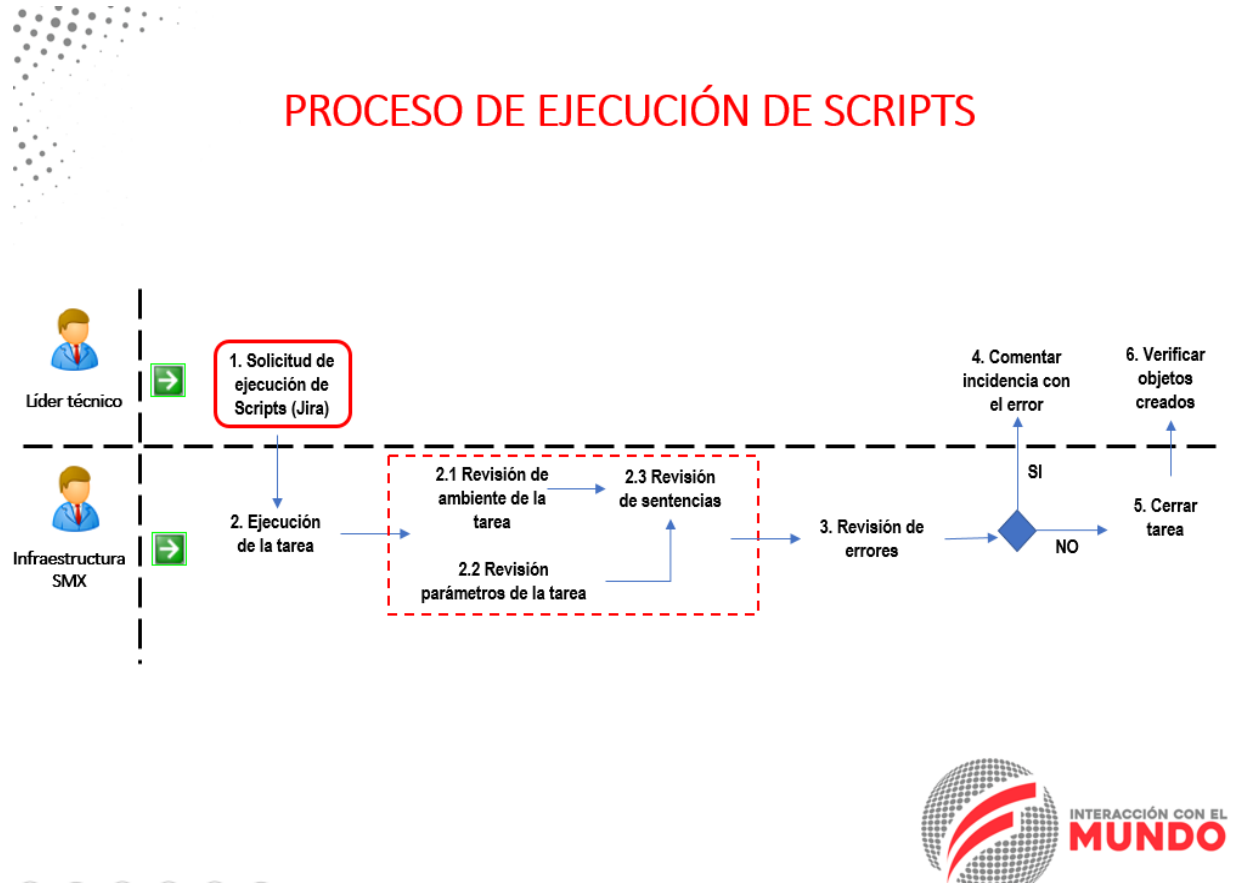


Figura 25 Proceso de ejecución de Scripts
Fuente: Autor

4.4.3.1 *Detalle del Proceso de ejecución de script*

1. **Solicitud de ejecución de Scripts (Jira):** Tarea del programa Jira donde se especifica el detalle de la tarea, como por ejemplo el ambiente donde se va a crear los objetos, las tablas afectadas, el adjunto donde se encuentra el script, la fecha que debe ser ejecutado.
2. **Ejecución de la tarea:** Para ejecutar la tarea se debe tener en consideración parámetros propios del negocio como horas y días específicos y los parámetros necesarios para la ejecución de la misma.
3. **Revisión de errores:** En algunos casos la ejecución de los scripts puede tener novedades o errores, estas deben hacerse conocer a la persona que solicita la tarea.
4. **Comentar incidencia con el error:** Si el script generó errores al momento de la ejecución se debe comentar o adjuntar el error generado o colocar la captura de la pantalla del error y cerrar la tarea para que luego de la corrección se cree una tarea nueva.
5. **Cerrar tarea:** Todas las tareas deben ser cerradas con los comentarios, en caso existir, de esta forma se tiene el registro de todas las ejecuciones realizadas en forma de bitácora en caso de ser necesarios de consultar.
6. **Verificar objetos creados:** Todos los objetos creados en la base de datos deben ser verificados por parte de la persona que solicita la tarea para constatar que la tarea fue ejecutada correctamente.

4.5 Etapa de Pruebas

En la etapa de pruebas se verifica que la aplicación cumpla con las reglas de negocio establecidas al inicio del proceso, y por otro lado, que cuente con parámetros de calidad y seguridad que garantice que no se presenten novedades cuando la aplicación se encuentre en producción, ya que estas pueden originar cambios en la programación o regresión de versiones.

4.5.1 Actores del proceso de Pruebas

Actores del proceso de pruebas	
Coordinador de Proyectos SMX	Colaborador a cargo de organizar y hacer seguimiento de los sistemas en desarrollo.
Líder técnico desarrollo	Profesional técnico que se encarga de administrar las tareas de desarrollo con el equipo de desarrolladores
Equipo de QA (Calidad)	Equipo de colaboradores que tiene como función probar las aplicaciones previas al despliegue en producción.

Tabla 15 Actores del proceso de pruebas
Fuente: Autor

Diccionario:

Kamban: Metodología para gestionar el trabajo, con parámetros de entrega justo a tiempo, tomando en cuenta la distribución equitativa de trabajo a todos los miembros del equipo.

Reglas de negocio: Lineamientos definidos al inicio del proyecto que identifican una tarea específica o una necesidad que tiene que ver con el objetivo del proyecto.

4.5.2 Caracterización del proceso de Pruebas

		CARACTERIZACIÓN DE PROCESO DE PRUEBAS					
Macro proceso:							
Código	Smx-001	Descripción	Proceso de Desarrollo y Operación de proyectos de software del área de TI				
Proceso:							
Código	Dev-004	Descripción	Verificación de cumplimiento de reglas de negocio, resolución de incidencias y verificación parámetros de calidad de software.				
Objetivo	Realizar control de calidad en las aplicaciones desarrolladas.						
Alcance	Planear las pruebas, detectar incidencias, reportar las incidencias, cerrar tareas.						
Líder del proceso		Macro proceso		Gerencia de TI			
		PROCESO		Equipo de calidad			
Actor	Entrada		Subprocesos		Salida		Herramienta
Coordinador	Cronograma de desarrollo y entregas		Planeación de pruebas		Historias de Scrum		Jira
Equipo de QA	Historias de Scrum		Desarrollo de cronograma de pruebas		Tareas de incidencias en Jira		Jira
Equipo de desarrollo	Tareas de incidencias con novedades		Resolución de novedades		Tareas de incidencias cerradas		Jira
Requisitos aplicables							
Internos					Externos		
Reglamento interno de Corporación Favorita							
Indicadores							
Código	Nombre	Frecuencia	F análisis	R recolección	R análisis	Fuente	Valor aceptable
IRP-Dev-004	Control de calidad en aplicaciones	30 días	30 días	Reportes de tareas de calidad de software	Gestión	Tareas de JIRA	100% de las tareas solicitadas

Tabla 16 Caracterización proceso de Pruebas.

Fuente: Autor

4.5.3 Flujo de proceso de Pruebas (QA)

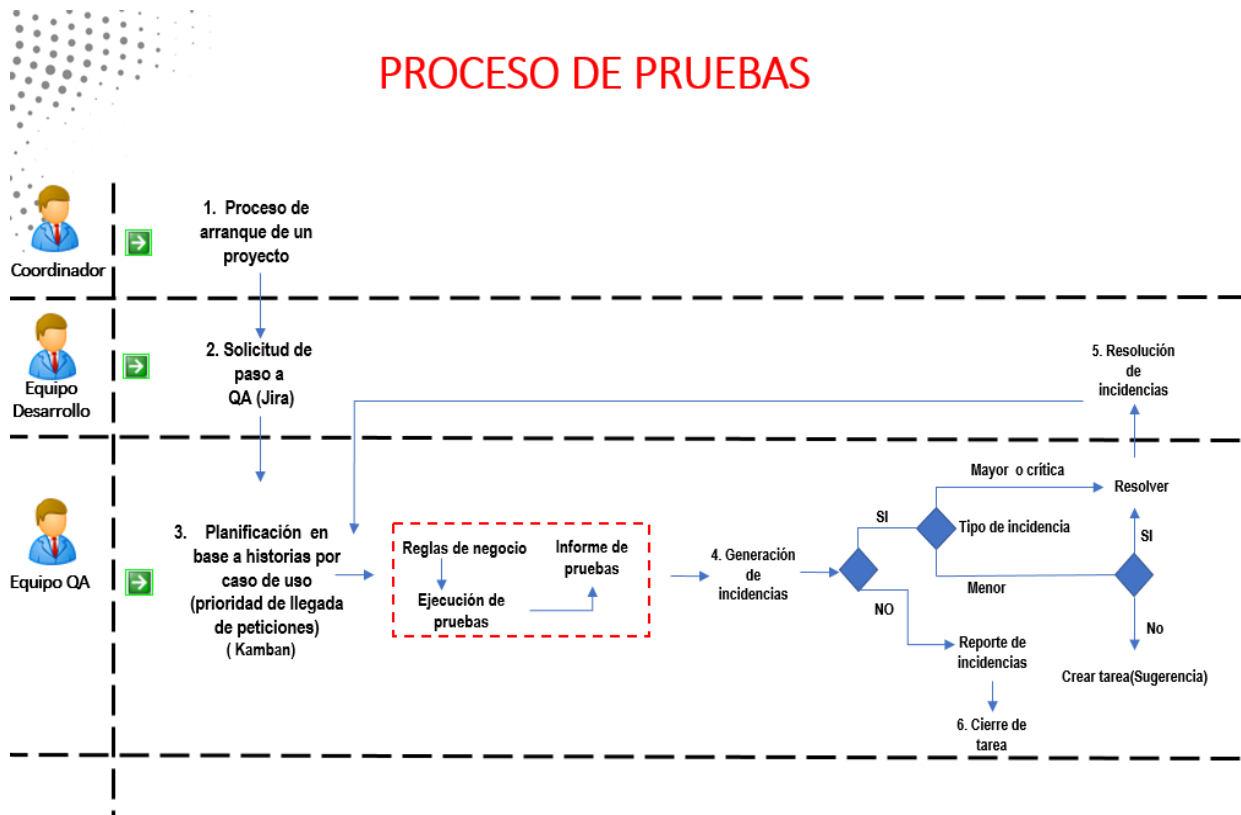


Figura 26 Proceso de Pruebas

Fuente: Autor

4.5.3.1 Detalle del Proceso de pruebas

- 1. Proceso de arranque de un proyecto:** Proceso donde se definen los calendarios de entregas totales o parciales del proyecto y se da el arranque de la codificación.
- 2. Solicitud de paso a QA (Jira):** Tareas solicitadas donde se especifican parámetros que buscan identificar incidencias de calidad en el desarrollo de una aplicación.
- 3. Planificación en base a historias por caso de uso (Kamban):** Cronogramas de actividades donde se establecen las tareas relacionadas a la funcionalidad de una aplicación de acuerdo al tiempo de desarrollo y al número de recursos en el equipo de QA.
- 4. Generación de incidencias:** Después de hacer las pruebas respectivas se pueden generar incidencias en el aplicativo probado, si dichas incidencias son críticas deben ser reportadas

al equipo de desarrollo para su corrección, si las incidencias son menores el coordinador del proyecto podrá decidir si se realizan las correcciones o si solo quedan como sugerencias.

5. **Resolución de incidencias:** Las incidencias encontradas deben ser resueltas por el equipo de desarrollo tratando en lo posible que la corrección no retrase las fechas de despliegue de los aplicativos.
6. **Cierre de tarea.** Todas las tareas deben ser cerradas con los comentarios en caso de existir, de esta forma se tiene el registro de todas las incidencias reportadas y resueltas.

4.6 Etapa de Lanzamiento

Es el procedimiento mediante el cual se hace la recepción de código fuente de un aplicativo desarrollado como por ejemplo los métodos para medir la calidad de código recibido y el versionamiento que se otorga una vez se encuentre estable.

4.6.1 Actores del proceso de Lanzamiento

Actores del proceso de Lanzamiento	
Equipo de Arquitectura	Colaboradores a cargo de establecer los lineamientos y políticas para el desarrollo de las aplicaciones.
Líder técnico desarrollo	Profesional técnico que se encarga de administrar las tareas de desarrollo con el equipo de desarrolladores

Tabla 17 Actores del proceso de lanzamiento.

Fuente: Autor

Diccionario:

SonarQube: Software para evaluar código fuente estático de un sistema con la finalidad de obtener métricas que puedan ayudar a mejorar la calidad del código desarrollado.

Nexus: Repositorio donde deben ser desplegadas las versiones generadas por el proyecto, tanto versiones estables como genéricas.

4.6.2 Caracterización del proceso de Lanzamiento

			CARACTERIZACIÓN DE PROCESO DE LANZAMIENTO				
Macro proceso:							
Código	Smx-001		Descripción		Proceso de Desarrollo y Operación de proyectos de software del área de TI		
Proceso:							
Código	Dev-005		Descripción		Proceso de versionamiento de un aplicativo.		
Objetivo	Generar una versión de un aplicativo que vaya a desplegar						
Alcance	Recepción de código fuente, análisis estático del código versionamiento del aplicativo.						
Líder del proceso		Macro proceso		Gerencia de TI			
		Proceso		Equipo de Arquitectura			
Actor	Entrada		Subprocesos			Salida	Herramienta
Equipo de Arquitectura	Código Fuente		Análisis de código estático			Versión en Nexus	Nexus Sonar Qube
Requisitos aplicables							
Internos						Externos	
Reglamento interno de Corporación Favorita							
Indicadores							
Código	Nombre	Frecuencia	F análisis	R recolección	R análisis	Fuente	Valor aceptable
IRP-Dev-005	Versionamiento de aplicaciones	30 días	30 días	Reporte de aplicaciones en Nexus	Gestión	Herramienta Nexus	100% de las tareas solicitadas

Tabla 18 Caracterización proceso de lanzamiento.

Fuente: Autor

4.6.1 Flujo de proceso de Lanzamiento.

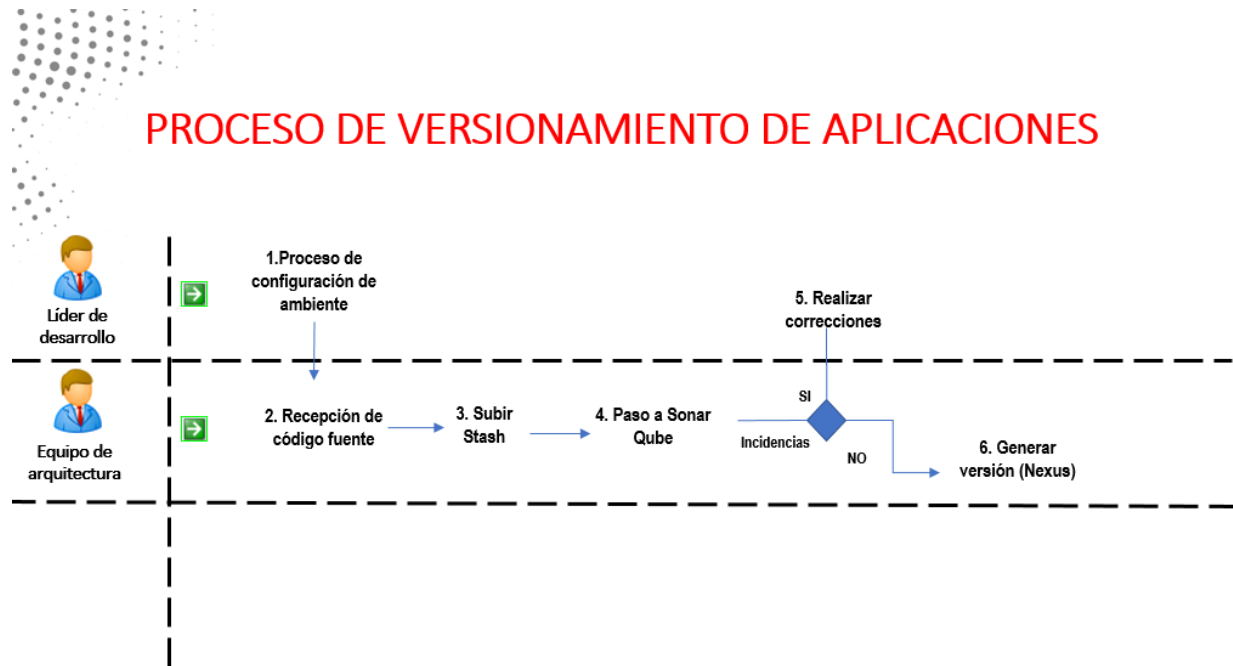


Figura 27 Flujo del proceso de versionamiento de aplicaciones.

Fuente: Autor

4.6.1.1 Detalle del Proceso de Lanzamiento

- 1. Proceso de configuración de ambiente:** Proceso mediante el cual se configura la infraestructura necesaria para probar la aplicación desarrollada o el módulo cumplido
- 2. Recepción de código fuente:** Actividad mediante la cual el proveedor hace entrega del código fuente del sistema desarrollado para las pruebas respectivas.
- 3. Subir Stash:** El código recibido debe ser subido a la herramienta Stash.
- 4. Paso a SonarQube:** Al correr el código fuente por esta herramienta permitirá identificar diferentes incidencias de seguridad que puede tener la aplicación desarrollada, dependiendo de este análisis la aplicación podrá ser desplegada.

5. **Realizar correcciones:** Si el reporte de las herramientas de calidad es muy alto se debería indicar al proveedor para su corrección y mejoramiento.
6. **Generar versión en Nexus:** Una vez el código de la aplicación se encuentre en parámetros aceptables de calidad se podrá generar una versión en el software Nexus.

4.7 Etapa de despliegue

En esta etapa se realizan todas las configuraciones necesarias para poner una aplicación en un ambiente determinado, identificando parámetros de configuración necesarios para que al momento de subir los empaquetados no genere errores que interfieran en el funcionamiento de otras aplicaciones.

4.7.1 Actores del proceso de despliegue

Actores del proceso de despliegue	
Coordinador de Proyectos SMX	Colaborador a cargo de organizar y hacer seguimiento de los sistemas en desarrollo.
Líder técnico desarrollo	Profesional técnico que se encarga de administrar las tareas de desarrollo con el equipo de desarrolladores
Infraestructura SMX	Grupo de colaboradores a cargo de administrar la plataforma tecnológica de la Corporación
Base de datos	Equipo de colaboradores que tiene a cargo la administración de la data existente en la Corporación

Tabla 19 Actores del proceso de despliegue.

Fuente: Autor

4.7.2 Caracterización del proceso de Despliegue

			CARACTERIZACIÓN DE PROCESO DE DESPLIEGUE				
Macro proceso:							
Código	Smx-001	Descripción	Proceso de Desarrollo y Operación de proyectos de software del área de TI				
Proceso:							
Código	Dev-006	Descripción	Proceso de despliegue de una aplicación en un ambiente determinado para sus pruebas u operación.				
Objetivo	Desplegar en un ambiente específico para una aplicación desarrollada.						
Alcance	Configurar componentes necesarios para la puesta en el ambiente indicado una aplicación desarrollada.						
Líder del proceso	Macro proceso		Gerencia de TI.				
	Proceso		Equipo de Infraestructura.				
Actor	Entrada		Subprocesos		Salida		Herramienta
Líder técnico	Aplicación o modulo finalizado		Creación de tarea con parámetros de la subida		Tarea de subida de aplicación		Jira
Coordinador	Tarea de subida de aplicación		Aprobación o negación de la tarea		Tarea de subida de aplicación		Jira
Infraestructura	Tarea de subida de aplicación		Planeación y ejecución de la tarea		Tarea cerrada		Jira
Requisitos aplicables							
Internos					Externos		
Reglamento interno de Corporación Favorita							
Indicadores							
Código	Nombre	Frecuencia	F análisis	R recolección	R análisis	Fuente	Valor aceptable
IRP-Dev-006	Despliegue de aplicación	30 días	30 días	Tareas puestas en la herramienta JIRA	Gestión	Reportes de Jira	100% de las tareas solicitadas

Tabla 20 Caracterización proceso de despliegue.

Fuente: Autor

4.7.3 Flujo de proceso de despliegue.

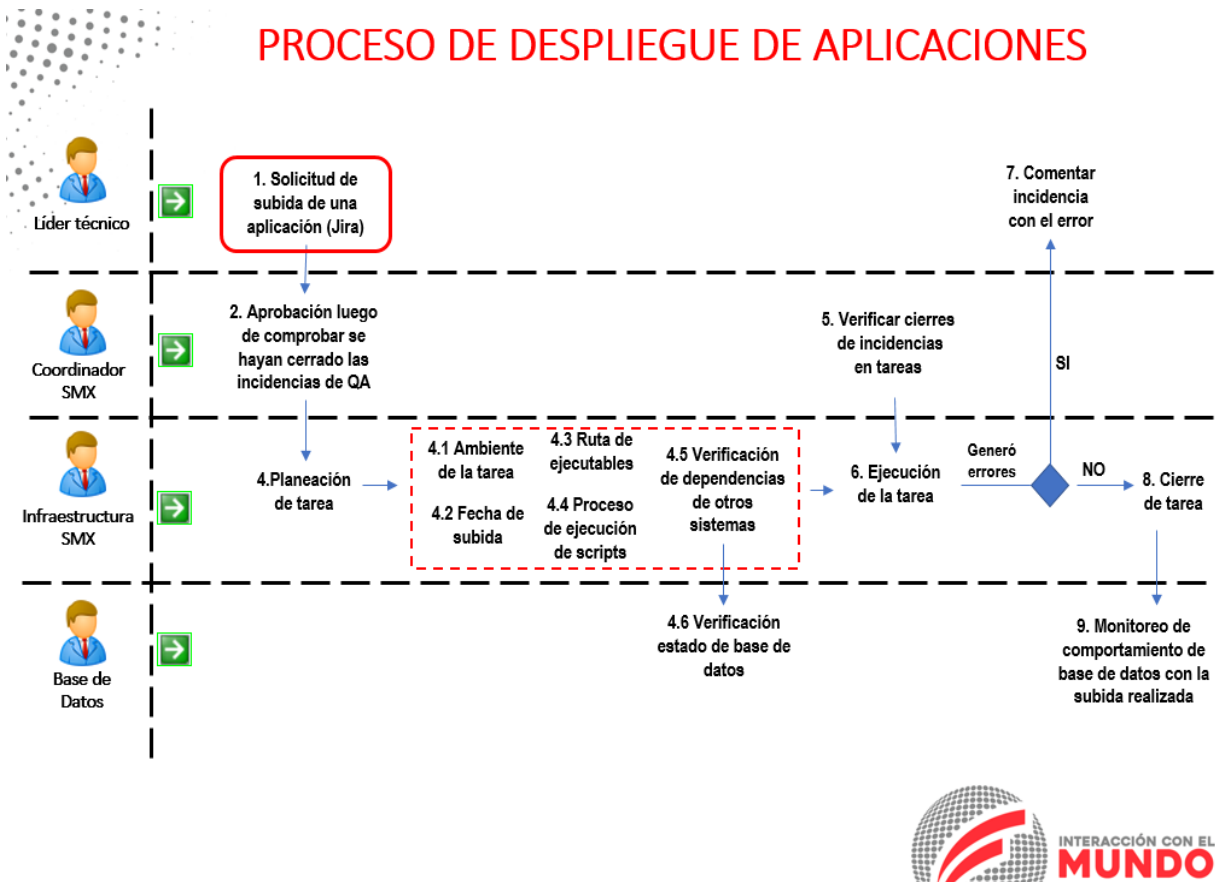


Figura 28 Flujo del proceso de despliegue

Fuente: Autor

4.7.3.1 Detalle del Proceso de Despliegue

- 1. Solicitud de subida de una aplicación (Jira):** Solicitud puesta en la herramienta Jira con la especificación “subida de aplicación”.
- 2. Aprobación luego de comprobar se hayan cerrado las incidencias de QA:** El coordinador del proyecto es el responsable de verificar que todas las incidencias de calidad hayan sido cerradas y deberá comentar en la tarea que se aprueba la ejecución de la tarea y reasignar la tarea al equipo de infraestructura.
- 3. Planeación de tarea:** Una vez la tarea se encuentra a cargo del área de infraestructura se deberá revisar todos los parámetros como: Ambiente de la tarea, fecha de ejecución,

ejecución de scripts necesarios para el despliegue, dependencia con otros sistemas entre otros.

4. **Ejecución de la tarea:** Ejecutar la tarea en el ambiente indicado luego de revisar que se encuentren todos los datos necesarios para la ejecución de la misma.
5. **Comentar incidencia con el error:** Si al momento de subir la aplicación genera errores se debe hacer conocer al líder técnico para su corrección
6. **Despliegue de la aplicación:** Si la aplicación no generó errores se desplegará, esta subida debe ser monitoreada en la base de datos para identificar si el despliegue realizado originó cambios significativos en la base de datos.

4.8 Etapa de operación

En la etapa de Operación se encuentra la configuración del equipamiento de hardware necesario para que la aplicación pueda estabilizarse dentro de la plataforma tecnológica existente en la Corporación, garantizando con esto la disponibilidad y la escalabilidad de la misma.

4.8.1 Actores del proceso de operación

Actores del proceso de operación	
Coordinador de Proyectos SMX	Colaborador a cargo de organizar y hacer seguimiento de los sistemas en desarrollo.
Líder técnico desarrollo	Profesional técnico que se encarga de administrar las tareas de desarrollo con el equipo de desarrolladores
Infraestructura SMX	Grupo de colaboradores a cargo de administrar la plataforma tecnológica de la Corporación

Tabla 21 Actores del proceso de operación.

Fuente: Autor

Diccionario:

JBoss: Servidor de aplicación Java EE de código abierto compatible con la mayoría de sistemas operativos.

4.8.2 Caracterización del proceso de Operación

		CARACTERIZACIÓN DE PROCESO DE OPERACIÓN					
Macro proceso:							
Código	Smx-001	Descripción		Proceso de Desarrollo y Operación de proyectos de software del área de TI			
Proceso:							
Código	Dev-007	Descripción		Proceso de configuración de le hardware para la Operación de una aplicación.			
Objetivo	Configurar el equipamiento necesario para el despliegue de una aplicación.						
Alcance	Recibir requerimientos, configurar equipos para despliegue de aplicaciones.						
Líder del proceso	Macro proceso		Gerencia de TI.				
	Proceso		Equipo de Infraestructura.				
Actor	Entrada		Subprocesos		Salida		Herramienta
Coordinador	Cronograma de desarrollo y entregas		Realizar especificaciones técnicas del equipo o instancias		Solicitud de configuración de ambiente		Jira
Infraestructura	Solicitud de configuración de ambiente		Configurar Software en servidor		Solicitud cerrada		Jira
Requisitos aplicables							
Internos					Externos		
Reglamento interno de Corporación Favorita							
Indicadores							
Código	Nombre	Frecuencia	F análisis	R recolección	R análisis	Fuente	Valor aceptable
IRP-Dev-007	Configuración de ambiente	30 días	30 días	Tareas puestas en la herramienta JIRA	Gestión	Reportes de Jira	80% de las tareas solicitadas

Tabla 22: Caracterización proceso Operación.

Fuente: Autor

4.8.3 Flujo de proceso de Configuración de ambiente

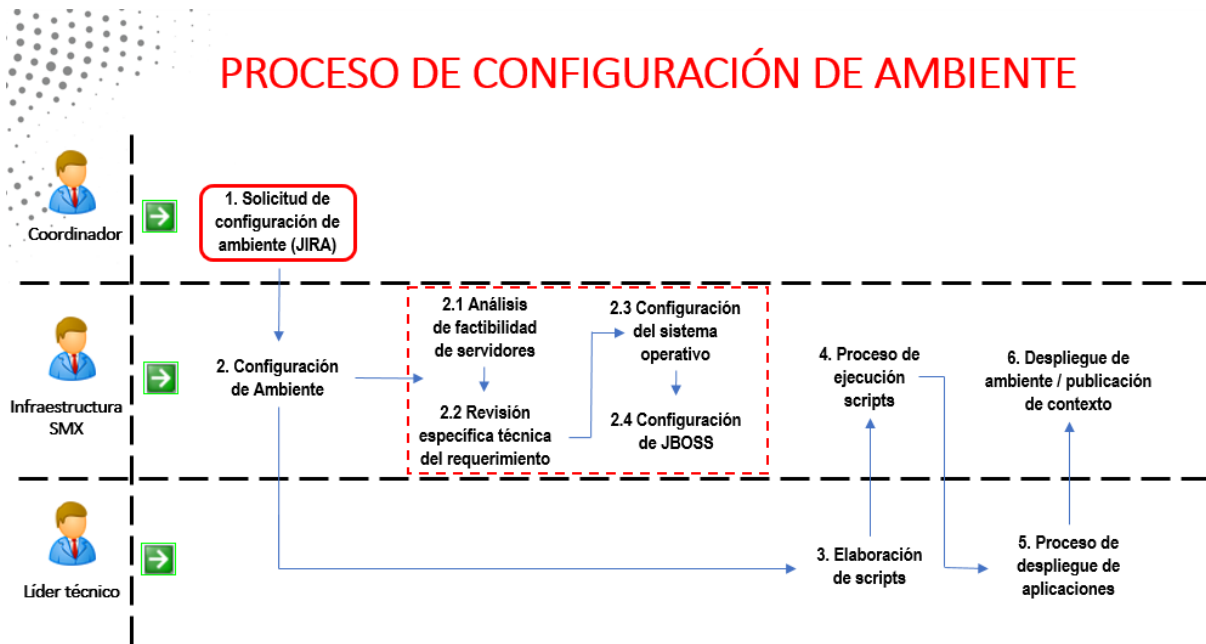


Figura 29 Flujo del proceso de configuración de ambiente.

Fuente: Autor

4.8.3.1 Detalle del Proceso de Configuración de ambiente

- 1. Solicitud de configuración de ambiente (JIRA):** Solicitud puesta en la herramienta Jira con la especificación “Configuración de ambiente”.
- 2. Configuración de Ambiente:** En esta actividad se contemplan varias configuraciones necesarias como: factibilidad de servidores, configuración de sistemas operativos, configuración de Jboss entre otras.
- 3. Elaboración de scripts.** En ocasiones antes de subir una aplicación en un ambiente ya configurado es necesario antes la ejecución de scripts.

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

4. **Proceso de ejecución scripts:** Proceso mediante se ejecutan sentencias en la base de datos o también alteraciones a los objetos de la base de datos.
5. **Proceso de despliegue de aplicaciones:** Proceso mediante el cual se pone en un ambiente específico una aplicación.
6. **Despliegue de ambiente / publicación de contexto.** Actividad que define que el equipamiento de hardware necesario para una aplicación está realizado, con esto se puede hacer el despliegue de un sistema desarrollado.

1.1.2 Flujo de proceso de capacitación de aplicativos al área de producción y locales.

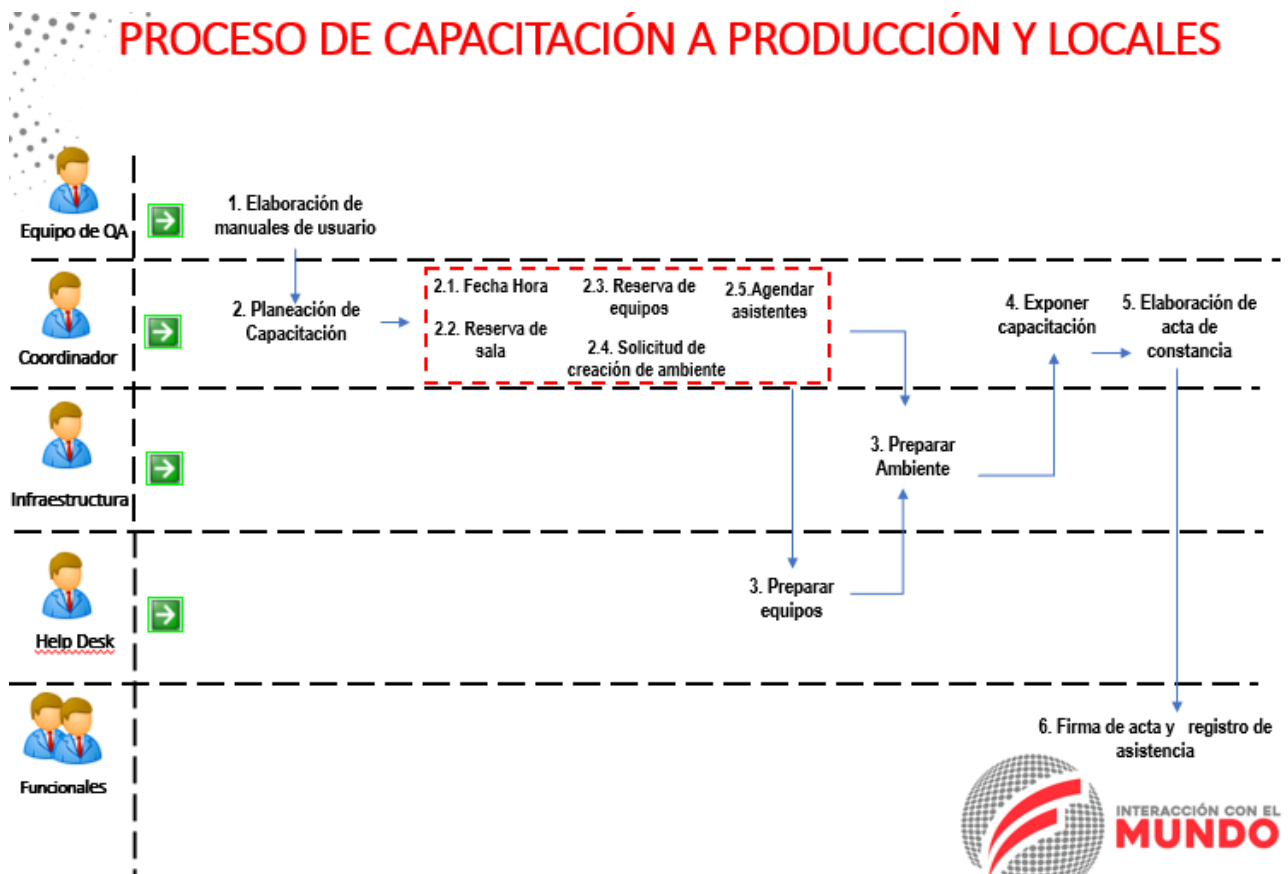


Figura 30 Flujo del proceso de capacitación a producción y locales.

Fuente: Autor

4.8.3.2 *Detalle de proceso de capacitación de aplicativos al área de producción y locales*

- 1. Elaboración de manuales de usuario:** El equipo de QA son los encargados de elaborar los diferentes manuales técnicos y de usuario correspondiente a los aplicativos que van a salir a producción.
- 2. Planeación de capacitación:** El coordinador a cargo de un proyecto deberá planificar la capacitación con los involucrados (locales, oficinas o centro de distribución) teniendo en cuenta la fecha lugar y equipos necesarios para la capacitación y agendar a los asistentes.
- 3. Preparar ambiente:** El coordinador deberá planificar con el equipo de Infraestructura la creación de un ambiente específico para montar la aplicación si fuera necesario, caso contrario los aplicativos deben probarse en el ambiente de pre producción.
- 4. Preparar equipos:** El equipo de help desk es el responsable de proveer el equipamiento necesario para las capacitaciones en conformidad al pedido realizado por el coordinador.
- 5. Exponer capacitación.** La capacitación deberá ser llevada a cabo dentro del tiempo establecido cumpliendo el cronograma indicado en el acta inicial de la reunión.
- 6. Elaboración de acta de constancia:** Una vez finalizada la capacitación se deberá elaborar un acta donde consten los temas tratados a detalle y los resultados obtenidos, así como si existe alguna novedad encontrada.
- 7. Firma de acta y registro de asistencia:** Todos los asistentes deberán firmar el acta para constancia de que fueron capacitados sobre los temas específicos. En algunos casos las personas que reciben la capacitación deberán capacitar a otros colaboradores por lo que en el acta debe constar este compromiso con la fecha y responsable de la actividad.

4.9 Etapa de monitoreo

En esta etapa se realiza un monitoreo de la aplicaciones ya desplegadas en Producción, se debe analizar el comportamiento de la aplicación frente al *pool* de conexión con los servidores de aplicación, con respecto a la base de datos y la compatibilidad con otras aplicaciones. Si en un tiempo determinado la aplicación cumple con los parámetros establecidos se puede hacer la recepción definitiva por parte del equipo de mantenimiento.

4.9.1 Actores del proceso de monitoreo

Actores del proceso de monitoreo	
Jefe de área (funcional)	Persona a cargo de un área de la Corporación o colaborador que solicita un requerimiento con la autorización del Jefe de área
Coordinador de Proyectos SMX	Colaborador a cargo de organizar y hacer seguimiento de los sistemas en desarrollo.
Equipo de Arquitectura	Colaboradores a cargo de establecer los lineamientos y políticas para el desarrollo de las aplicaciones.
Líder técnico desarrollo	Profesional técnico que se encarga de administrar las tareas de desarrollo con el equipo de desarrolladores
Infraestructura SMX	Grupo de colaboradores a cargo de administrar la plataforma tecnológica de la Corporación
Coordinador de Mantenimiento	Colaborador que está a cargo de la recepción y de recibir la capacitación un sistema específico para el mantenimiento

Tabla 23: Actores del proceso de monitoreo.

Fuente: Autor

4.9.2 Caracterización del proceso de Monitoreo

		CARACTERIZACIÓN DE PROCESO DE MONITOREO		
Macro proceso:				
Código	Smx-001	Descripción	Proceso de Desarrollo y Operación de proyectos de software del área de TI	
Proceso:				
Código	Dev-008	Descripción	Proceso donde se analiza el estado de la aplicación para el debido proceso a mantenimiento.	
Objetivo	Pasar una aplicación al área de Mantenimiento			
Alcance	Analizar estado de las aplicaciones entregadas por los proveedores, cargar entregables en un ambiente propio, verificar si cumplen con check list de requerimientos para el paso a manteniendo de una aplicación.			
Líder del proceso	Macro proceso	Gerencia de TI.		
	Proceso	Equipo de Mantenimiento de aplicaciones web.		
Actor	Entrada	Subprocesos	Salida	Herramienta
Coordinador	Aplicación o modulo codificado.	Monitoreo de la aplicación por un mes	Solicitud de paso a Mantenimiento	Jira
Coordinador de Mantenimiento	Solicitud de paso a Mantenimiento	Verificación de check list de parámetros	Agenda de reuniones para capacitación	Jira
Líder de Mantenimiento	Agenda de reuniones para capacitación	Capacitación funcional y técnica de las aplicaciones	Solicitud de paso a Mantenimiento cerrada	Jira
Requisitos aplicables				
Internos			Externos	
Reglamento interno de Corporación Favorita				
Check list para recepción de aplicaciones				

Indicadores							
Código	Nombre	Frecuencia	F análisis	R Recolección	R análisis	Fuente	Valor aceptable
IRP-Dev-008	Aplicaciones en mantenimiento	30 días	30 días	Tareas puestas en la herramienta JIRA	Gestión	Reportes de Jira	100% de las tareas solicitadas

Tabla 24: Caracterización proceso de monitoreo.

Fuente: Autor

4.9.3 Flujo de proceso de paso a mantenimiento

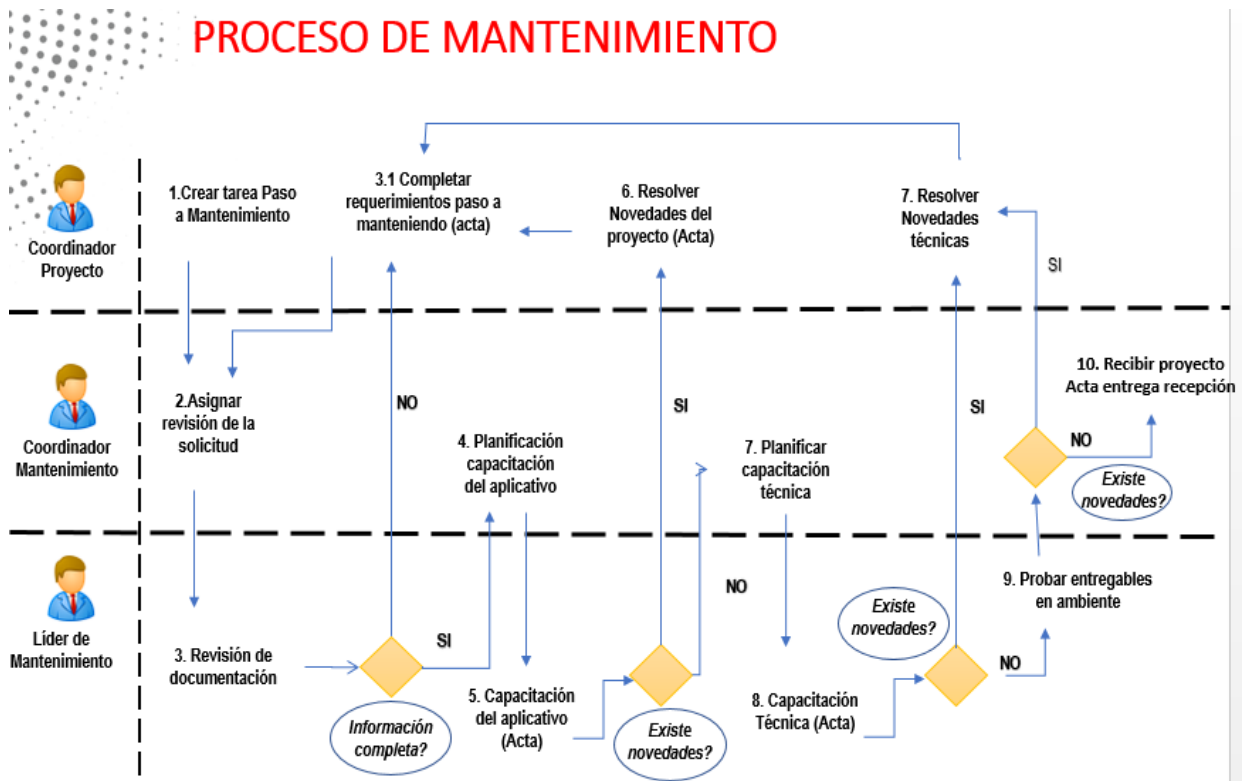


Figura 31 Flujo del proceso de paso a mantenimiento.

Fuente: Autor

4.9.3.1 *Detalle del proceso de paso a mantenimiento*

- 1. Crear tarea paso a mantenimiento:** El coordinador del proyecto deberá crear una tarea en JIRA de “paso a mantenimiento” y asignar la tarea del coordinador de Mantenimiento
 - 2. Asignar revisión de la solicitud:** El Coordinador de Mantenimiento deberá designar un líder de mantenimiento y asignar la tarea de recibir el sistema.
 - 3. Revisión de documentación:** El líder de Mantenimiento revisará la información referente al proyecto en base a un chek list de manteniendo.
 - 4. Completar requerimientos paso a mantenimiento (acta):** Si en la revisión de la documentación hay novedades se deberá indicar al coordinador del proyecto con las observaciones en un acta.
 - 5. Planificación capacitación del aplicativo:** Después que la documentación esté completa, el Coordinador de mantenimiento deberá planificar la capacitación del aplicativo
 - 6. Capacitación de la applicativa (acta):** Si en la revisión de la documentación hay novedades se deberá realizar la capacitación de la funcionalidad del sistema por parte del equipo de desarrollo al Líder de mantenimiento.
 - 7. Resolver novedades del proyecto (acta):** Si en la capacitación hay novedades se deberá indicar al Coordinador del proyecto con las observaciones en un acta.
 - 8. Planificar capacitación técnica:** Después que las observaciones de la capacitación del aplicativo hayan sido gestionadas, el Coordinador de mantenimiento deberá planificar la capacitación técnica.
 - 9. Capacitación técnica :** Si en la capacitación técnica hay novedades se deberá indicar al Coordinador del proyecto con las observaciones en un acta.
 - 10. Probar entregables en ambiente:** Una vez se hayan recibido las respectivas capacitaciones se deberá probar los entregables de la aplicación en un ambiente propio.
- Recibir proyecto acta entrega recepción:** Cumplida todas la actividad anterior se puede recibir el aplicativo para lo cual se debe elaborar un acta de entrega recepción con el detalle de módulos.

4.9.4 Seguridad en desarrollo.

Cada fase de DevOps sugiere la implementación de un modelo de madurez para el aseguramiento de software. Por esta razón la Corporación debe aumentar las estrategias de seguridad de software.

Para la implementación del modelo de seguridad se sugiere que la Corporación adopte los lineamientos de Open Sam puesto que esta herramienta que ayuda tener un enfoque en los siguientes aspectos:

- Evaluar las prácticas de seguridad en Software existentes en la organización.
- Construir un programa de seguridad en Software balanceado en iteraciones bien definidas.
- Demostrar mejoras concretas en el programa de aseguramiento de Software.
- Definir y medir actividades relacionadas con seguridad en la organización.

Con la implementación de un modelo de seguridad el enfoque entre Open Sam y DevOps quedaría como en la siguiente figura:

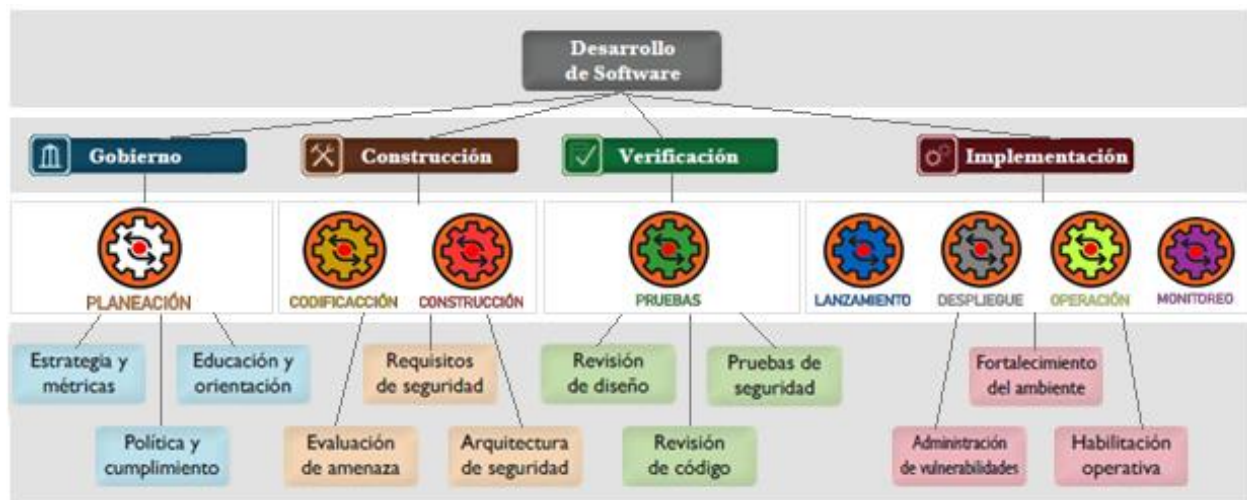


Figura 32 Modelo de seguridad de Open Sam con DevOps.

Fuente: Autor.

Como sugiere la herramienta SWAT (Security web Applications Technologies) todas las aplicaciones desarrolladas por o para la Corporación deberán cumplir los siguientes parámetros de seguridad:

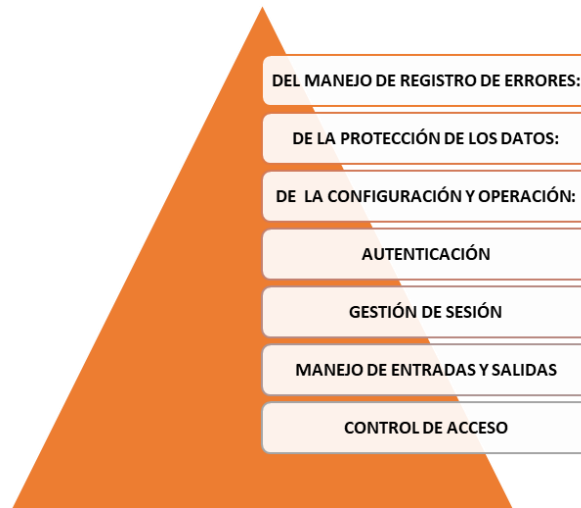


Figura 33 *Parámetros de seguridad a evaluar según SWAT*

Fuente: Autor

4.10 Sistematización de DevOps

DevOps tiene un enfoque de entrega continua por lo que aparte del modelo de gobierno de TI y la identificación de los procesos que intervienen en el desarrollo de aplicaciones es necesario la implementación de herramientas que logren sistematizar cada iteración tanto del desarrollo como de la Operación.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Después del desarrollo de este trabajo se puede concluir que:

Se desarrolló un modelo de Gobierno de TI para el área de proyectos de Corporación Favorita, apoyado en un estudio de la situación actual del área en lo referente a Gobernabilidad de TI. La información fue recolectada en base a entrevistas a los miembros de equipo principalmente a la Sub Gerente a cargo del área.

La realización de entrevistas a los miembros claves de la organización es una buena práctica de levantamiento de información de la fuente directa, esta información sumada a la aplicación de la metodología DevOps permitió crear un modelo de Gobierno de TI.

Para la propuesta del modelo de gobierno se seleccionó a Cobit 4.1 como el mejor marco de gobierno, puesto que Cobit 5 está orientado a toda la organización. Dentro del Cobit 4.1 se aplicó el marco de referencia de nivel de madurez del área en cuanto a procesos.

Con esta información se pudo identificar que uno de los puntos más bajos es el referente al control PO4 DEFINIR PROCESOS, con este dato el levantamiento de matriz de la situación actual se lo enfocó en los controles referentes a: estructura organizacional, marco de trabajo de procesos, definir roles y responsabilidades.

El modelo de Gobierno de TI desarrollado comprende el conjunto de cada proceso y el detalle de cada iteración. Por cada proceso se realizó una matriz de caracterización donde se explican los objetivos, actores, entradas, salidas, modo de recolección de la información y forma

de medición. Con esto se logró establecer las tareas y responsabilidades de cada actor, logrando fortalecer los parámetros presentados en la matriz de levantamiento inicial.

Se pudo determinar que la utilización de DevOps como marco de trabajo requiere la implementación de un modelo de madurez de seguridad. Se realizó un diagrama donde se explica la forma como Open Sam y DevOps pueden combinarse para lograr cumplir con parámetros de seguridad en cada etapa del desarrollo y la operación.

Con fecha viernes 12 de julio en la sala 1 de las oficinas administrativas de Corporación Favorita se brindó una presentación sobre el modelo de Gobierno de TI propuesto. En dicha reunión estuvieron presentes todos los colaboradores pertenecientes al área de desarrollo de proyectos de software de la Corporación, incluida la Sub Gerente a cargo del área como consta en el acta de asistencia firmada por los asistentes. En esta reunión se aprobó este trabajo y se planteó la necesidad de implementar lo más pronto posible el modelo de gobierno.

De la experiencia recolectada en esta investigación se puede concluir que un modelo como el que se propone en este trabajo puede ser implantado en empresas medianas o grandes y que tengan alta demanda en desarrollo, porque la sistematización de DevOps puede llegar a requerir fuertes inversiones en cuanto a licenciamiento e infraestructura.

5.2 Recomendaciones:

- Para la implementación del modelo de Gobierno de Tecnología propuesto se recomienda utilizar un indicador de Gestión en el área de TI y un plan de acción para el seguimiento mensual de la implementación.
- Se recomienda que los modelos de gobierno de TI sean socializados con todos los involucrados para lograr el compromiso de todos.
- El modelo de Gobierno de TI debe ir de la mano con un modelo de madurez de seguridad por lo que se recomienda para trabajos posteriores se implemente un modelo de seguridad para el desarrollo de software en el área de proyectos que mejore el existente.
- Una vez implantado el modelo de Gobierno de TI propuesto en el área de desarrollo de proyectos de la Corporación se recomienda se implemente un modelo de Gobierno de TI para toda el área de Tecnología acorde a los objetivos organizacionales.

BIBLIOGRAFÍA

- Albarracín, E. J. G., Erazo, S. C. R., & Palacios, F. C. (2014). Influencia de las tecnologías de la información y comunicación en el rendimiento de las micro, pequeñas y medianas empresas colombianas. *Estudios Gerenciales*, 30(133), 355-364. Obtenido de <https://www.redalyc.org/pdf/212/21232399004.pdf>
- Alejandro, F., & Karina, I. (2017). *Definición de un ambiente de construcción de aplicaciones empresariales a través de Devops, Microservicios y Contenedores* (Bachelor's thesis). Obtenido de <http://dspace.utpl.edu.ec/bitstream/20.500.11962/21205/1/Far%C3%ADas%20Alejandro%20Ivonne%20Karina%20tesis.pdf>
- Vargas Alvarado, S. M. *Modelo de gobierno de TI como apoyo a los procesos administrativos: Caso Universidad de los Llanos* (Doctoral dissertation, Universidad Nacional de Colombia-Sede Manizales). Obtenido de <http://www.bdigital.unal.edu.co/56375/>
- Campos, A., & René, A. (2017). *Propuesta de modelo de gobernanza de TI para el desarrollo de software para equipos tipo VSE (entidades muy pequeñas)* (Master's thesis, Quito, 2017.).
- Bannerman, P. L. (2009). *Software development governance: A meta-management perspective*. Obtenido de <https://ieeexplore.ieee.org/document/5071329> Obtenido de <https://bibdigital.epn.edu.ec/bitstream/15000/17646/1/CD-8087.pdf>
- Bienvenido, A. (2018). *Iniciación al ciclo de vida de DevOps*. Obtenido de <https://www.tecnologias-informacion.com/devops.html>
- Castro-Fuentes, M., Solana-González, P., & Pérez-González, d. Análisis de la implantación del gobierno de ti en la administración de Cantabria a través del marco de ITIL. Obtenido de https://www.researchgate.net/publication/309012929_Analisis_de_la_Implantacion_del_Gobierno_de_TI_en_la_Administracion_de_Cantabria_a_traves_del_Marco_ITIL
- Contreras, F. G., & Reinoso, I. T. (2008). Externalización de funciones: algunas reflexiones teóricas. *Estudios gerenciales*, 24(107), 107-135. Obtenido de https://www.icesi.edu.co/revistas/index.php/estudios_gerenciales/article/view/261/html
- Espinoza Flores, B. D., & Coronel Marquez, J. A. (2014). *Evaluación del grado de madurez de los procesos del área de TI de la empresa Fiduecuador SA* (Bachelor's thesis, Quito: EPN, 2015.). Obtenido de <https://bibdigital.epn.edu.ec/handle/15000/10371>
- Favorita, C. (2018). Gobierno Cooperativo. *Revista de Informe de accionistas 2017*.
- Gamboa, J. Z. (2018). Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. *INNOVA Research Journal*, 3(10), 20-33. Obtenido de <https://dialnet.unirioja.es/descarga/articulo/6777227.pdf>
- Governance, Institute IT. (2007). *Cobit 4.1*. Obtenido de <https://www.isaca.org/Knowledge-Center/cobit/Documents/COBIT4.pdf>

Diseño de un modelo de gobernabilidad de TI para el área de Proyectos de Corporación Favorita, basado en DevOps

- Hernández, M. G. (2014). Escuela de Ingeniería Informática Magíster en Ingeniería Informática (Doctoral dissertation, Pontificia Universidad Católica de Valparaíso). Obtenido de http://opac.pucv.cl/pucv_txt/txt-6500/UCD6592_01.pdf
- Isaca. (2012). *COBIT*. Obtenido de <https://articulosit.files.wordpress.com/2013/07/cobit5-framework-spanish.pdf>
- Jaramillo Wilches, W. E. (2016). *Aplicación de la metodología RUP y el patrón de diseño MVC en la construcción de un sistema de gestión académica para la Unidad Educativa Ángel de la Guarda* (Bachelor's thesis, PUCE). Obtenido de <http://repositorio.puce.edu.ec/bitstream/handle/22000/11264/Documento%20Disertaci%C3%B3n%20Wendy%20Jaramillo.pdf?sequence=1&isAllowed=y>
- Jiménez Marco, G. (2016). DevOps, la nueva tendencia en el desarrollo de sistemas TI, un caso práctico en el análisis de incidencias de software. Obtenido de <https://pdfs.semanticscholar.org/5252/3f8ba5b45f1aa136a1eb33e105ec641b2294.pdf>
- López, J. S. (2012). *Metodología para gestionar inversiones de TI en Instituciones de Educación Superior del Sector Privado basada en VAL IT Y COBIT*. Obtenido de <http://documentos.redclara.net/bitstream/10786/775/1/36-METODOLOGIA%20PARA%20INVERSIONES%20DE%20TI-1604-2013.pdf>
- Perinán, M., Lucia, I., López Revelo, J. S., & Manguel Villalba, M. (2014). Metodología para gestionar inversiones de TI en instituciones de educación superior del sector privado basada en VAL IT y COBIT.
- Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2016). An exploratory study of devops extending the dimensions of devops with practices. *ICSEA 2016*, 104. Obtenido de https://www.researchgate.net/profile/Luigi_Lavazza/publication/307576316_ICSEA_2016_The_Eleventh_International_Conference_on_Software_Engineering_Advances/links/57c9a36a08ae3ac722af8728.pdf#page=105
- Martín Medrano, Ó. (2013). Evaluación de la gobernanza TIC por controles COBIT utilizando técnicas de decisión multicriterio difusas e híbridas. Obtenido de <https://repositorio.unican.es/xmlui/bitstream/handle/10902/3884/Oscar%20Martin%20Medrano.pdf?sequence=1>
- Mera, J. A. (2016). *Análisis del proceso de pruebas de calidad de software*. Obtenido de <https://revistas.ucc.edu.co/index.php/in/article/view/1482>
- Naranjo Calderón, F. E. (2014). *Evaluación técnica del gobierno de TI de la Dirección Provincial del Consejo de la Judicatura de Pichincha* (Master's thesis, Universidad de las Fuerzas Armadas ESPE. Maestría en Evaluación y Auditoría de Sistemas Tecnológicos.). Obtenido de <http://repositorio.espe.edu.ec/jspui/handle/21000/9905>
- Nguyen, V. H., Kolp, M., Wautelet, Y., & Heng, S. (2018). Mapping IT Governance to Software Development Process: From COBIT 5 to GI-Tropos. In *ICEIS (2)* (pp. 665-672). Obtenido de <https://pdfs.semanticscholar.org/9d2a/35048b97468b955bf24cdf7a085ad8e68da8.pdf>

- Núñez Palacios, N. D. R. (2015). *Desarrollo de un modelo de gobierno y gestión de TI para empresas con los servicios de TI tercerizados. Caso de estudio: empresa de la industria financiera* (Master's thesis, Quito, Universidad de las Américas, 2015). Obtenido de <http://dspace.udla.edu.ec/bitstream/33000/5120/1/UDLA-EC-TMGSTI-2015-29.pdf>
- Obando Tapia, C. A. (2015). *Supermercados en Ecuador: oligopolios e implicaciones de la Ley Orgánica de Regulación y Control del Poder del Mercado* (Bachelor's thesis, Quito/PUCE/2015). Obtenido de <http://repositorio.puce.edu.ec/bitstream/handle/22000/8393/Disertaci%C3%B3n%20Carolina%20Obando%20Tapia.pdf?sequence=1>
- Osterweil, L. J. (2006). Proceedings of the 28th international conference on Software engineering. *University of Southern California University Park Campus, Los Angeles* (págs. 12-29). Shanghai, China: ICSE'06, May 20–28, 2006, Shanghai, China.
- Pérez Hoyos, L. (2018). DevOps: IT Development in the Era of Digitalization. Obtenido de <http://uvadoc.uva.es/handle/10324/31469ray>
- Rantakari, L. (2010). Governance in business process outsourcing: case study on call center outsourcing. Obtenido de http://epub.lib.aalto.fi/fi/ethesis/pdf/12260/hse_ethesis_12260.pdf
- Rayo, A. M. (2016). *DevOps ¿Es posible el cambio cultural en las organizaciones?* Obtenido de <https://www.netmind.es/knowledge-center/devops-es-posible-el-cambio-cultural/>
- Sosa, L. M. O. (2016). Modelo de Clasificación y Evolución de Metodología de Desarrollo de Sistemas de Información. *Tekhne*, 1(7). Obtenido de <http://revistasenlinea.saber.ucab.edu.ve/temas/index.php/tekhne/article/view/2852>
- Vivar, G., & Carlos, J. (2013). *Desarrollo del marco de referencia Cobit 5.0 para la Gestión del área de TI de la empresa Blue Card* (Bachelor's thesis, QUITO/PUCE/2013). Obtenido de <http://repositorio.puce.edu.ec/bitstream/handle/22000/6078/T-PUCE-6320.pdf?sequence=1&isAllowed=y>
- Weill, P., Subramani, M., & Broadbent, M. (2001). Strategic Agility. *Group*. Obtenido de <https://www.ics.uci.edu/~wscacchi/Tech-EC/EC-EB-Infrastructure/IT-strategy-infrastructure-invest-SMR-2002.pdf>