



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA:

**Análisis y Formulación de estrategia general para preparar la adopción
de la cultura DevOps en matrices bancarias de la ciudad de Guayaquil**

AUTOR:

Karen Denisse Terán Villafuerte

**Trabajo de titulación previo a la obtención del título de
INGENIERO EN SISTEMAS COMPUTACIONALES**

TUTOR:

Ing. César Adriano Salazar Tovar

Guayaquil, Ecuador

10 de septiembre, 2020



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENNERÍA

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

CERTIFICACIÓN

Certificamos que el presente trabajo de titulación fue realizado en su totalidad por Terán Villafuerte Karen Denisse, como requerimiento para la obtención del título de **Ingeniero en Sistemas Computacionales**.

TUTOR

f. 

Ing. César Adriano Salazar Tovar, Mgs

Guayaquil, a los 10 días del mes de septiembre del año 2020



UNIVERSIDAD CATOLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES

DECLARACION DE RESPONSABILIDAD

Yo, Karen Denisse Terán Villafuerte

DECLARO QUE:

El Trabajo de Titulación, **Análisis y Formulación de estrategia general para preparar la adopción de la cultura DevOps en matrices bancarias de la ciudad de Guayaquil** previo a la obtención del título de **Ingeniero en Sistemas Computacionales**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 10 días del mes de septiembre del año 2020

EL AUTOR

f. 

Karen Denisse Terán Villafuerte



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES

AUTORIZACION


Yo, Karen Denisse Terán Villafuerte

Autorizo a la Universidad Católica de Santiago de Guayaquil a la publicación en la biblioteca de la institución del Trabajo de Titulación, **Análisis y Formulación de estrategia general para preparar la adopción de la cultura DevOps en matrices bancarias de la ciudad de Guayaquil** cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 10 días del mes de septiembre del año 2020

EL AUTOR:

f.



Karen Denisse Terán Villafuerte



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES

REPORTE URKUND

URKUND	
Documento	TESIS TERÁN (028).docx (D78156317)
Presentado	2020-08-25 07:29 (-05:00)
Presentado por	cesar.salazar@cu.ucsg.edu.ec
Recibido	cesar.salazar.ucsg@analysis.orkund.com
Mensaje	Trabajo de titulación - Karen Terán Mostrar el mensaje completo
	0% de estas 53 páginas, se componen de texto presente en 0 fuentes.

TUTOR

f. _____

Ing. César Adriano Salazar Tovar, Mgs.

AGRADECIMIENTO

El desarrollo de un trabajo de investigación de dificultad como la tesis de grado no es un mérito que se logre de forma individual, existieron muchas personas y organizaciones que hicieron posible el realizar el presente trabajo de investigación, es así como se dedica unas pequeñas palabras para agradecer a todos aquellos que aportaron al desarrollo de esta tesis de grado.

En primer lugar quiero agradecer a Dios por acompañarme a lo largo de mis estudios y del desarrollo de este trabajo, hubo momentos en los que fui débil y pensé no ser capaz de lograr cumplir con este proyecto, pero él me dio las fuerzas para superar mis miedos y seguir adelante.

También quiero agradecer a mi tutor el Ing. César Salazar, quien con su apoyo y conocimientos me guio a través de cada etapa del desarrollo de este proyecto para alcanzar los resultados propuestos.

Quiero agradecer a mi profesor de la materia Tópicos el Ing. Gilberto Castro, quien me brindó su apoyo y guía durante este proceso de tesis con sugerencias y observaciones que le dieron más valor al trabajo.

Un agradecimiento especial para los funcionarios de los bancos encuestados y a los consultores entrevistados quienes me dieron la apertura y su colaboración para la recopilación de la información que fue de gran ayuda en el desarrollo de este proyecto de investigación.

Finalmente quiero agradecer a mi familia por su apoyo y fe ciega en mí, en los momentos en los que no me sentía segura de mí misma me dieron los ánimos que requería para recuperar la confianza y completar el desarrollo de este trabajo.

Muchas gracias a todos los involucrados por su apoyo constante a lo largo de estos meses.

DEDICATORIA

Esta tesis está dedicada a:

A mis padres Carmen y Jhonny quienes, con su amor y apoyo incondicional, me han permitido ser la persona que soy hoy; gracias por inculcar en mí los valores, el ejemplo de esfuerzo y el coraje para afrontar las adversidades y buscar cumplir mis metas.

A mi hermana Adriana quien de su forma particular me dio palabras de aliento y fortaleza para no rendirme durante el proceso de desarrollo de esta tesis de grado.

ÍNDICE

ÍNDICE.....	VIII
ÍNDICE DE TABLAS	XI
ÍNDICE DE FIGURAS.....	XII
RESUMEN.....	XIII
ABSTRACT.....	XIV
INTRODUCCIÓN	2
CAPÍTULO I.....	5
EL PROBLEMA.....	5
PLANTEAMIENTO DEL PROBLEMA.....	5
1.1 UBICACIÓN DEL PROBLEMA EN UN CONTEXTO	5
1.2 CAUSAS Y CONSECUENCIAS DEL PROBLEMA.....	5
1.3 DELIMITACIÓN DEL PROBLEMA.....	6
1.4 FORMULACIÓN DEL PROBLEMA.....	7
1.5 EVALUACIÓN DEL PROBLEMA.....	8
1.6 OBJETIVOS.....	10
1.6.1 OBJETIVO GENERAL	10
1.6.2 OBJETIVOS ESPECÍFICOS	10
1.7 ALCANCES DEL PROBLEMA.....	11
1.8 JUSTIFICACION E IMPORTANCIA.....	11
1.9 HIPÓTESIS O PREGUNTA DE INVESTIGACIÓN	12
1.9.1 PREGUNTA DE INVESTIGACIÓN	12
1.10 VARIABLES DE LA INVESTIGACIÓN	12

CAPÍTULO II.....	13
MARCO TEÓRICO	13
2.1 ELEMENTO CONTEXTUAL	13
2.2 ELEMENTOS TEÓRICOS	17
2.2.1 CULTURA DEVOPS	17
2.2.2 VALOR DEL NEGOCIO	24
2.2.3 DIFICULTADES DE TI PARA AGREGAR VALOR AL NEGOCIO 26	
2.2.4 DEVOPS Y LAS MEJORES PRÁCTICAS DE LA INSDUSTRIA 27	
2.2.5 HERRAMIENTAS DE DEVOPS	31
CAPÍTULO III.....	35
METODOLOGÍA DE LA INVESTIGACIÓN	35
3.1 TIPO DE INVESTIGACIÓN.....	35
3.2 POBLACIÓN Y MUESTRA	36
3.2.1 POBLACIÓN	36
3.3 INSTRUMENTOS DE RECOLECCIÓN DE DATOS.....	37
3.4 SELECCIÓN DE LOS ENCUESTADOS.....	39
3.5 ANÁLISIS DE RESULTADOS.....	40
CAPÍTULO IV.....	48
PROPUESTA METODOLÓGICA.....	48
4.1 GUÍA GENERAL PARA EL LEVANTAMIENTO DE INFORMACIÓN Y OBTENCIÓN DEL NIVEL DE MADUREZ	49
4.1.1 ASPECTO CULTURA	50

4.1.2	ASPECTO PROCESOS Y PRÁCTICAS.....	51
4.1.3	ASPECTO TECNOLOGÍA Y AUTOMATIZACIÓN	54
4.1.4	VALORACIÓN Y ANÁLISIS DE MADUREZ.....	57
4.1.5	ANÁLISIS DE APLICACIONES.....	65
4.2	ROAD MAP GENERAL.....	66
4.2.1	FASE 1.....	66
4.2.2	FASE 2.....	69
4.2.3	FASE 3.....	72
4.2.4	FASE 4.....	74
4.2.5	FASE 5.....	76
4.3	CLASIFICACIÓN DE HERRAMIENTAS DEVOPS	77
	CONCLUSIONES	90
	RECOMENDACIONES.....	92
	REFERENCIAS BIBLIOGRAFICAS.....	93
	ANEXOS.....	101
	Anexo 1	101
	Anexo 2	105
	Anexo 3	109
	Anexo 4	115
	Anexo 5	133

ÍNDICE DE TABLAS

Tabla 1	Tiempos de entrega de productos y servicios tecnológicos	42
Tabla 2	Metodologías de Desarrollo Ágiles.....	43
Tabla 3	Metodologías de desarrollo Tradicionales y Ágiles	43
Tabla 4	Tabla de valoración-aspecto cultural.....	58
Tabla 5	Calificación elemento transparencia.....	60
Tabla 6	Resultado Assessment	62
Tabla 7	Matriz de niveles de madurez DevOps.....	63
Tabla 8	Matriz de madurez con esquema de semáforo	64
Tabla 9	Inventario de aplicaciones.....	65
Tabla 10	Matriz de herramientas recomendadas	80

ÍNDICE DE FIGURAS

Figura 1 Historia de DevOps.....	18
Figura 2 Complementos de DevOps.....	20
Figura 3 Paquete Completo de DevOps	22
Figura 4 Valores de DevOps.....	23
Figura 5 Dimensiones del valor de negocio	24
Figura 6 Componentes de ITIL V4	28
Figura 7 Ciclo de vida DevOps	32
Figura 8 Tiempo promedio de entrega de proyectos de desarrollo	40
Figura 9 Características del Testing de software	41
Figura 10 Objetivos de los departamentos de Desarrollo y Operaciones de TI de los bancos.....	42
Figura 11 Adopción de cultura DevOps en bancos	44
Figura 12 Razones para adoptar DevOps.....	44
Figura 13 Limitaciones para adopción de DevOps.....	45
Figura 14 Modelo propuesta de adopción de cultura DevOps	48
Figura 15 Modelos de servicio de cómputo en la nube	76
Figura 16 Ciclo de vida DevOps	78

RESUMEN

El presente trabajo de titulación contiene unas guías generales a ser usadas como referencia para preparar la adopción de la cultura DevOps en bancos con casa matriz en la ciudad de Guayaquil. La investigación utilizada fue de tipo exploratoria, ya que se hizo uso de encuestas a funcionarios de bancos para conocer su estado actual frente al tema de creación y entrega de software a sus clientes, y se realizaron entrevistas a consultores (dos nacionales y uno internacional) para que, en base a su experiencia, obtener las pautas y recomendaciones para la formulación de las guías. Las guías cuentan con 3 matrices y un road map general, iniciando con una matriz guía de entrevistas para el levantamiento de información las cuales contienen: roles/áreas a entrevistar, sugerencia de documentación a obtener, sugerencias de mecanismos para obtener la información; para la evaluación y cálculo del nivel de madurez de los bancos, se formuló la matriz de valoración la cual contiene una serie de preguntas para evaluar los 3 aspectos principales de DevOps: cultura y gente, prácticas y procesos, y tecnología y automatización, así como una matriz general de madurez DevOps, ambas matrices se crearon usando los 5 niveles del enfoque CMMI recomendado por los consultores entrevistados e investigación de diferentes fuentes de consultora como MediaTek Consulting Hub que han puesto a disposición su trabajo. Para el road map general se propone cinco fases y en cada una de ellas se seguirán los 3 aspectos evaluados de la matriz de madurez. Finalmente se propone una matriz de herramientas recomendadas por DevOps las cuales fueron clasificadas acorde a las fases del ciclo de vida de esta cultura, conteniendo costos, valoraciones y características clave de cada herramienta propuesta.

Palabras Clave: *DevOps, Instituciones bancarias de Guayaquil, nivel de madurez, CMMI, Plan de acción general.*

ABSTRACT

This degree work contains general guides that are used as a reference to prepare for the adoption of the DevOps culture in banks with headquarters in the city of Guayaquil. The research used was exploratory, since surveys were made for bank officials to find out their current status regarding the creation and delivery of software to their clients, and interviews were conducted with consultants (two national and one international) so that, based on their experience, obtain the guidelines and recommendations for the formulation of the guidelines. The guides have 3 matrices and a general roadmap, starting with an interview guide matrix for gathering information, which contains: roles / areas for interviews, suggestion of documentation to obtain, suggestions of mechanisms to obtain information; for the evaluation and calculation of the maturity level of the banks, the evaluation matrix was formulated which contains a series of questions to evaluate the 3 main aspects of DevOps: culture and people, practices and processes, and technology and automation, as well as a General DevOps maturity matrix, both matrices were created using the 5 levels of the CMMI approach recommended by the interviewed consultants and from investigations performed from different sources such as consultant MediaTek Consulting Hub whose work was provided. For the general road map, five phases are proposed and in each one of them the 3 evaluated aspects of the maturity matrix will be followed. Finally, a matrix of tools recommended by DevOps is proposed, which were classified according to the phases of the life cycle of this culture, containing costs and key characteristics of each proposed tool.

Key words: *DevOps, Banking Institutions of Guayaquil, Level of maturity, CMMI, General Road Map.*

INTRODUCCIÓN

La “transformación digital” es un término que, aunque se lo empezó a usar de forma relativamente reciente, se ha dado prácticamente desde el inicio de los tiempos del ser humano. El mundo en la historia siempre ha estado en constantes cambios y descubrimientos, hace miles de años inició con la rueda y las máquinas simples, cientos de años después llegaron los primeros pasos tecnológicos de la historia humana: la imprenta, las máquinas de Babbage, Enigma, entre otros; y el nacimiento del internet en 1983.

En comparación a cientos de años atrás la evolución tecnológica se ha disparado de forma exponencial, tecnología que antes tenía una vida útil de años, hoy en día puede quedar obsoleta en cuestión de meses, y ese es el mayor reto que enfrentan las empresas del presente, adaptarse a los constantes cambios tecnológicos para seguir brindando productos y servicios a sus clientes. En pocas palabras la transformación digital es “vivir mejor con software” (RedHat, 2017).

La transformación de la banca tradicional es un punto clave hoy en día, al ser el pilar de la economía mundial, brindando productos y servicios tanto a empresas como a personas naturales, debe adaptarse a la tendencia del presente, un modelo de negocio digital, como indica Antonio Bravo director de Desarrollo de Negocios y banca digital de América del Sur en BBVA: “La digitalización de nuestro negocio está produciendo un impacto positivo en la experiencia y el servicio al cliente y además nos está llevando hacia un modelo de banca más sostenible”(González, 2019, párr. 1). Bravo hace hincapié en 2 impactos clave de la digitalización a la banca:

- Al haber canales tecnológicos de comunicación a disposición del cliente, como acceso web o móvil, aumenta la superficie de contacto entre el banco y sus clientes.
- Al establecer una comunicación no presencial entre el banco y el cliente, se crea un modelo de banca más lógico que permite a las oficinas, liberarse de las filas generadas por tareas transaccionales, para que ahora estas mismas se dediquen netamente al asesoramiento de servicios y productos financieros a sus clientes.

Con el objetivo de abordar de mejor manera la transformación digital, nace el marco Agile o Ágil el cual, a diferencia de su predecesor la metodología tradicional o cascada, adopta un enfoque diferente que se aleja de la planificación de todo el proyecto, aceptando la incertidumbre; Agile se basa en la idea de responder al cambio en lugar de cargarlo como un costo adicional o simplemente ignorarlo. El marco Agile ha sido una auténtica mejora en cuanto a proyectos de desarrollo de software se refiere, pero una desventaja de este es que se ve limitado a las etapas de desarrollo, pruebas e implementación del software, sin embargo, para muchas empresas la agilidad en estas etapas ya no es suficiente para alcanzar sus nuevas metas y mantenerse competitivos, y es así como ante una necesidad de mayor agilidad nace una nueva cultura (Brown, 2020).

DevOps nace como un pilar dentro del contexto de la transformación digital, este marco o cultura se centra en la entrega de servicios rápidos de TI haciendo uso de prácticas ágiles, esto se logra mediante la mejora en la colaboración entre el departamento de desarrollo y el equipo de operaciones, es esta precisamente la diferencia más marcada respecto a Agile ya que DevOps involucra ambos departamentos (desarrollo/operaciones) (Novoseltseva, 2018).

Las empresas del Ecuador, específicamente hablando de los bancos, se verían altamente beneficiados de la implementación de la cultura DevOps en sus modelos de negocios, ya que los conduciría a la transformación digital y por ende a una mejora sustancial de la entrega de sus servicios y productos a sus clientes, esto se ha podido evidenciar en casos de implementación exitosos como el de la consultora Cognizant de España, quienes realizaron el estudio previo y la implementación de la cultura DevOps en un importante banco mundial, entre los resultados obtenidos se encuentra la reducción de defectos funcionales y de tiempos de inactividad de aplicaciones, un ahorro en costos gracias a la automatización y una mejora sustancial con los ciclos de lanzamiento de software (Consultora Cognizant, s. f.); otro caso exitoso es la implementación en un banco importante que fue ejecutado por la consultora Nisum, este banco a pesar de contar con agilidad en sus áreas de desarrollo, no podían alcanzar las metas que tenían planteadas, es por ello que optan

por DevOps, y los resultados que obtuvieron superaron sus expectativas (Consultora Nisum, 2017).

Partiendo de lo mencionado anteriormente, la implementación de una cultura como DevOps, requiere de preparación de las empresas, un exhaustivo análisis de su estado actual y una planificación para la adopción de esta cultura y con ello agilizar sus procesos para lograr la transformación digital.

Por ello, el objetivo de este trabajo es preparar una guía general que permita a los bancos con casa matriz en la ciudad de Guayaquil realizar la autoevaluación del estado de sus áreas de desarrollo y operación, en relación con las mejores prácticas recomendadas por DevOps, e identificando su estado de madurez, los bancos puedan trazar su plan de acción o road map para la adopción de DevOps.

CAPÍTULO I

EL PROBLEMA

PLANTEAMIENTO DEL PROBLEMA

1.1 UBICACIÓN DEL PROBLEMA EN UN CONTEXTO

Las empresas hoy en día se han visto obligadas a tener que dejar su forma de trabajar tradicional para adaptarse a las nuevas tendencias tecnológicas que el mundo les exige usar. Los bancos al ser empresas de suma importancia, considerados del sector estratégico, deben ser las primeras en encaminarse a la transformación digital, mediante la adopción de marcos que les permitan trabajar de forma ágil. La forma tradicional de operación de las empresas, sin ser la excepción los bancos, ha sido mantener sus áreas de desarrollo y operaciones independientes funcionalmente; las razones por las cuales estas estructuras se mantienen son diversas, pero independientemente de la razón, el resultado es el mismo, el potencial organizacional se ve inhibido (Leclercq, 2019); como consecuencia de esto el tiempo que le toma iniciar el desarrollo de un producto o servicio hasta su liberación a producción es muy alto, esto afecta su competitividad en el mercado ya que otros bancos al tener esquemas más ágiles, son capaces de lanzar soluciones tecnológicas, ya sea para el cliente o para uso interno, de manera más rápida y eficiente que aquellos bancos que no han logrado la agilidad en sus procesos (Zauzich, 2015). Los bancos se encuentran en una carrera por adaptarse a las tendencias tecnológicas del mundo, la cual se aceleró por la actual pandemia mundial, lograr la digitalización se ha convertido en un objetivo primordial para mantener la ventaja competitiva y para ello adoptar modelos o culturas ágiles a sus procesos de desarrollo y operaciones de TI es un hecho no una opción.

1.2 CAUSAS Y CONSECUENCIAS DEL PROBLEMA

De acuerdo con lo indicado por uno de los arquitectos de soluciones de la empresa Maint S.A, en general los frameworks reconocidos mundialmente ya sea para gestión de proyectos, agilidad, calidad, etc; establecen criterios de buenas prácticas de aplicación general, sin embargo, para su aplicación las

empresas deben realizar grandes esfuerzos en términos de estudio, investigación o utilización de consultoría especializada para aplicar dichas guías generales en sus propios entornos. DevOps no es la excepción, esta cultura ágil no posee guías que indiquen los pasos a seguir para poder facilitar la realización de una autoevaluación del estado de madurez de las matrices bancarias de Guayaquil, y empresas en general, para definir los pasos a seguir en su adopción de DevOps. Seguramente esto se debe a que, como en muchos casos se requiere de la contratación de consultores o personal especializado en DevOps, se vuelve un tema comercial ya que es más conveniente ofrecer servicios de consultoría para la implementación de DevOps que contar con guías que le permitan a la misma empresa autoevaluarse para iniciar la adopción de esta cultura. La falta de guías que puedan ser accesibles para bancos de Guayaquil los obliga a tener que contratar consultoras o profesionales muy calificados de DevOps, por lo general extranjeros, lo cual resulta en costos elevados, siendo en algunos casos, un obstáculo en la decisión de adopción de esta cultura.

1.3 DELIMITACIÓN DEL PROBLEMA

El problema se delimita en los siguientes campos:

Campo:	Cultura DevOps
Área:	Matrices bancarias de Guayaquil
Aspecto:	Interacción entre las áreas de desarrollo y operación de TI de las matrices bancarias de Guayaquil
Tema:	Preparación de la documentación necesaria para la elaboración de una guía general, que permita la autoevaluación del estado de madurez de las áreas de desarrollo y operación de TI de las matrices bancarias de la ciudad de Guayaquil

1.4 FORMULACIÓN DEL PROBLEMA

La alta competitividad en el ámbito bancario de la ciudad de Guayaquil obliga a estas entidades a buscar innovarse constantemente para ofrecer servicios y productos de calidad a su clientela, la evolución tecnológica que se viene dando desde hace siglos, ha provocado que las industrias, y los bancos incluidos, tengan que cambiar sus modelos de negocio para mantenerse a la vanguardia tecnológica y competir con sus adversarios a nivel global y tendiente a la completa digitalización (*Revista Asobanca | Asobanca*, 2019., p. 6).

Un claro ejemplo de esto son las Fintech que se han instituido como el futuro de la banca virtual, bancos totalmente digitales han obligado a la banca tradicional a tener que emprender su camino hacia la transformación digital (*Medrano*, 2019.), para ello es necesario la adopción de procesos ágiles en las distintas áreas de la organización, que les posibilite el desarrollo y entrega de productos y servicios que generen valor para sus clientes internos y externos.

Una de las soluciones más empleadas en la banca y que ha logrado resultados que han superado las expectativas es la cultura DevOps, muchos bancos a nivel mundial y nacional se están viendo beneficiados de esta cultura; sin embargo para las matrices bancarias de Guayaquil y en general cualquier banco es un gran desafío la transición de una cultura de trabajo tradicional a otra en la que se rompe los paradigmas que la cultura tradicional ha mantenido, como ya se mencionó anteriormente, las empresas optan por la contratación de empresas consultoras, en su mayoría extranjeras, o profesionales expertos en DevOps, los cuales representan altos costos, y otras deciden postergar la adopción de esta cultura.

Dada esta problemática se formula el problema de la siguiente manera:

¿Qué necesitan las matrices bancarias de la ciudad de Guayaquil para realizar su autoevaluación y definir un plan de acción para la adopción de la cultura DevOps?

1.5 EVALUACIÓN DEL PROBLEMA

La falta de agilidad en los procesos del ciclo de vida del desarrollo de aplicaciones es un problema latente en las empresas de distintos sectores de la economía del Ecuador y del mundo, sin embargo, el presente estudio, será limitado a las matrices bancarias de Guayaquil. Los bancos se encuentran en una carrera por adaptarse a las tendencias tecnológicas del mundo la cual se aceleró por la pandemia mundial, lograr la digitalización se ha convertido en un objetivo primordial para mantener la ventaja competitiva, para ello adoptar modelos o culturas ágiles a sus procesos de desarrollo y operaciones de TI se ha convertido en una necesidad, acorde a uno de los arquitectos de soluciones de la empresa Maint S.A la falta de una guía destinada al uso de las matrices bancarias de Guayaquil ralentiza y en ocasiones hasta evita que estas matrices puedan prepararse en su adopción para la cultura DevOps dado que no todas las entidades tienen la opción de contratar personal especializado que realice el estudio y trace el road map a seguir para la adopción de la misma.

Es evidente ya que es un hecho que la transformación digital es prácticamente una obligación que se debe llevar a cabo por cualquier empresa que desee ser competitiva en el mercado actual, las matrices bancarias de Guayaquil se encuentran en una constante competencia por ser los primeros en lanzar productos y servicios de índole tecnológico a sus clientes. Como indica Juan Carlos Arcila, presidente del Congreso Latinoamericano de Banca Digital, Innovación y Tecnología: “hoy en día es difícil encontrar un banco que no haya desarrollado un portal web transaccional o que no esté incursionando en la banca móvil” (Diario El Telégrafo, 2019, para. 2). Aunque las matrices bancarias poseen ciertos productos o servicios ofertados de forma digital, aún no han alcanzado un alto nivel de digitalización de sus servicios y la demanda de usuarios más exigentes e informados (nativos digitales) los obliga a desarrollar y lanzar al mercado más herramientas que faciliten las transacciones y demás servicios bancarios de manera no presencial, por ello la adopción de una cultura o marco de trabajo que les permita alcanzar una mayor rapidez en el desarrollo e implementación de soluciones tecnológicas hacia los clientes es una problemática evidente que necesita ser atendida.

Es relevante ya que la transformación digital no es un tema o preocupación reciente, la urgencia por adaptarse a las condiciones cambiantes se ha dado desde el siglo XVIII con lo que se conoció como la Industria 1.0, la etapa más actual de esta transformación se conoce como Industria 4.0 en la cual nuevas tecnologías como Big Data, Machine Learning, han revolucionado la forma en la que las empresas laboran (Martínez, 2017), las empresas del área financiera entre ellas los bancos son de los principales negocios que deben adaptar sus modelos tradicionales a modelos ágiles para poder ser competitivos en el mercado; un ejemplo de esta competencia presente entre los bancos de Guayaquil fue el lanzamiento de la primera billetera digital desarrollada por Banco del Pacífico (Cordova, 2018), este producto fue puesto al público el 16 de Enero del 2018 y obligó a los demás bancos a buscar formas de competir, como Banco Machala que lanzó su propia billetera móvil en octubre del mismo año. Los bancos que han agilizado sus procesos en sus áreas de desarrollo y operaciones de TI crean y entregan ofertas de valor a sus clientes con mayor rapidez y el desarrollo de estas es más eficiente. En estos momentos de pandemia es importante que los bancos de Guayaquil adopten culturas ágiles a sus procesos para proveer productos y servicios que ayuden a la sociedad a realizar sus transacciones con el menor costo posible tanto en tiempo como en facilidad manteniéndose competitivos y en crecimiento en el mercado ecuatoriano.

La transformación digital mediante el cambio de estructuras de trabajo tradicionales a ágiles es un tema que se ha dado desde la primera revolución industrial, y aunque se ha enfatizado en varios bancos a nivel mundial, representa un cambio cultural muy complejo para ser adoptado de forma rápida y sencilla por las empresas en el mundo sin ser la excepción nuestro país Ecuador. Este trabajo es original, plantea una solución que pueda servir como un acelerador en el proceso.

Es factible ya que se cuenta con amplia información a la cual se puede hacer referencia para el estudio de la problemática, así como casos que aportan con la experiencia como lo es la implementación en importantes bancos internacionales realizados por empresas consultoras de reconocimiento mundial; adicionalmente gracias a las herramientas

tecnológicas con las que se cuenta y a la operatividad que no ha cesado en esta pandemia, es posible realizar investigaciones vía remota con personal de las firmas consultoras, y los propios bancos para recopilar y obtener información que sustenten debidamente el problema y desarrollar la solución propuesta.

Es concreto debido a que a pesar de que es una problemática aplicable a cualquier empresa que desarrolle sus propios productos y/o servicios; el presente trabajo se enfocará en las matrices bancarias de la ciudad de Guayaquil.

1.6 OBJETIVOS

1.6.1 OBJETIVO GENERAL

El objetivo general planteado es el siguiente:

Establecer una estrategia general que sirva como guía en el análisis del estado de madurez de las matrices bancarias de la ciudad de Guayaquil, con áreas de desarrollo y operaciones, para definir un plan de acción que los prepare para adoptar la cultura DevOps en sus modelos de negocio

1.6.2 OBJETIVOS ESPECÍFICOS

Los respectivos objetivos específicos planteados son los siguientes:

- Definir una guía general para realizar la evaluación del estado actual y mediante la comparación con las mejores prácticas de DevOps, obtener la madurez de las matrices bancarias de la ciudad de Guayaquil
- Establecer un road map general para que sea utilizado como guía para elaborar el road map específico de las matrices bancarias de Guayaquil, que buscan prepararse para adoptar DevOps
- Clasificar las herramientas recomendadas por DevOps teniendo como criterio de clasificación las etapas del ciclo de vida de las aplicaciones y su integración con operación

1.7 ALCANCES DEL PROBLEMA

El alcance del proyecto se delimita en los siguientes entregables:

- Guía para levantamiento de información del assestment
- Matriz general de comparación entre estado actual y mejores prácticas de DevOps
- Plan de acción general guía acorde a nivel de madurez
- Inventario de herramientas recomendadas en una cultura DevOps y clasificadas acorde a las etapas del ciclo de vida de las aplicaciones y su integración con operación

No es parte del alcance de este proyecto la implementación de la guía, tampoco la implementación de DevOps.

1.8 JUSTIFICACION E IMPORTANCIA

La cultura DevOps es una excelente opción al momento de buscar la transformación digital, y es un hecho que dentro del ámbito de la banca la adopción de esta cultura ha traído beneficios que han superado las expectativas de los empresarios, prueba de esto son los distintos bancos internacionales como: Aktia Bank y los Bancos que son casos de éxito de las consultoras Cognizant y Nizum.

La ventaja de los casos de implementación previamente mencionados es que se ha dado en países en los cuales el conocimiento y la experiencia con DevOps es mayor que la que hay en el Ecuador, por lo que para algunas de las matrices bancarias de Guayaquil, aunque buscan mantenerse competitivas con sus pares frente a los cambios tecnológicos, no tienen perfiles con el dominio y experiencia para iniciar su preparación para la adopción de DevOps y en muchos casos no cuentan con el capital para contratar consultores extranjeros.

Por ello el desarrollo de una guía general se justifica y se considera que podría resultar de gran ayuda a las matrices bancarias de la ciudad de Guayaquil para iniciar su preparación para la adopción de DevOps.

1.9 HIPÓTESIS O PREGUNTA DE INVESTIGACIÓN

A través de esta investigación se desea responder de forma positiva o negativa la pregunta de investigación:

1.9.1 PREGUNTA DE INVESTIGACIÓN

¿Es posible el desarrollo de una guía general para la preparación de la adopción de la cultura DevOps en las matrices bancarias de la ciudad de Guayaquil?

1.10 VARIABLES DE LA INVESTIGACIÓN

- **Variable Independiente:** Análisis y estudio de la cultura DevOps y sus mejores prácticas
- **Variable dependiente:** Guía general para que las matrices bancarias puedan realizar su evaluación y definir el road map para la adopción de DevOps

CAPÍTULO II

MARCO TEÓRICO

El Marco Teórico permite conocer todos los fundamentos que serán de utilidad en el contexto del proyecto. A continuación, se muestra el sustento teórico que permitirá tener una visión más clara de los conceptos y teorías que son parte de este proyecto de investigación.

2.1 ELEMENTO CONTEXTUAL

La banca tiene sus orígenes varios siglos atrás, los primeros vestigios de lo que se conocía como “banco” datan del año 3,400 a.C. en Mesopotamia, siendo la realidad más aceptada al banco actual la encontrada en 1473 conocido como “El monte de crédito sobre prenda” fundado en Monte de Bolonia (J.C,2016.). La evolución de la banca, desde su concepción, se ha visto directamente influenciada por las nuevas tecnologías, todo negocio que se figura mantenerse competitivo en el mercado debe ser consciente de los constantes cambios y adaptarse a las tecnologías, requerimientos del mercado y metodologías de la actualidad. Esto se puede evidenciar en las 4 etapas que, según Fanjul y Valdunciel, se divide la historia de la banca:

- Primera etapa: Se origina en los años sesenta, en ese entonces los objetivos que buscaba la banca en general era una mejora en la productividad acompañada de una reducción de costos, sin embargo, estos objetivos se vieron limitados por la falta de la capacitación del personal, así como por la falta de herramientas tecnológicas que agilizaran los procesos.
- Segunda etapa: En los años setenta el objetivo a conseguir era la mejora del servicio ofrecido, para ello se introdujo en el negocio bancario el teleproceso, sin embargo, al igual que con la primera etapa hubo aspectos que limitaron el cumplimiento del objetivo, en este caso al ser relativamente nuevo el concepto del sistema de telecomunicaciones se presentaban problemas con

su implementación y uso, también hubo problemas con la homologación normativa.

- Tercera etapa: En los años ochenta se comienza a incorporar el concepto de ofimática, la apertura de puntos de venta y la mayor novedad, el sistema de banca electrónica, durante esta etapa los problemas originados fueron por parte de los servicios telemáticos, principalmente la incompatibilidad de los equipos.
- Cuarta etapa: En los años noventa se introduce por primera vez el sistema de banca virtual, con lo cual se busca a diferencia de la tercera etapa, reducir la cantidad de puntos de venta, lograr una mayor automatización de los procesos, y magnificar la comunicación con el cliente a través de la red.

(Avendaño Carbellido, 2018)

A partir de la cuarta etapa el objetivo planteado no ha cambiado, buscar la automatización para mejorar la rapidez con la que se entrega valor a los clientes mejorando la calidad y seguridad de los productos ofertados, sin embargo, alcanzar este objetivo es un gran desafío para la banca.

Desde sus inicios la banca, así como muchos otros negocios, han mantenido una estructura de trabajo bastante tradicional en la cual, de forma general, consiste de una estructura jerárquica donde cada área funcional responde a una jefatura, entre estos departamentos existen dependencias, como lo es el caso de los departamentos de Desarrollo y Operaciones de TI en el que Operaciones va a depender de Desarrollo para la puesta en producción y Desarrollo depende de Operaciones para el testeado, y aunque la dependencia entre estos departamentos es evidente e irrefutable, hay tiempos de espera muy dilatados entre los procesos que ejecutan estos departamentos incrementando los tiempos del proceso del ciclo de desarrollo de software y su respectiva puesta en producción y soporte (Sutil, 2017), todavía existen barreras que limitan la efectividad y velocidad con la que es posible para un banco producir soluciones.

El sector de la banca siempre ha estado a la par en cuanto a soluciones tecnológicas se refiere, sin embargo en una época en la que la competencia

de la banca nace con un “Know-how” bastante elevado en nuevas tecnologías (NUEVO FINANCIERO, 2018) es necesario un cambio para que la banca tradicional pueda competir con estos emprendimientos. Un ejemplo de esta competencia son las famosas Fintech, estas relativamente nuevas empresas que actúan como intermediarios entre la banca y sus clientes ofrecen servicios financieros de forma netamente digital (BBVA, 2015), otro ejemplo de competencia son los Neobancos, quienes al igual que las Fintech buscan brindar servicios financieros enfocados a un nicho de mercado que prefiere el uso de tecnología web o móvil para completar sus operaciones bancarias, en lugar de asistir a una sucursal física para realizarlo, pero a diferencia de las Fintech estas si son entidades financieras que implementan la filosofía Fintech (BBVA, 2019); la velocidad con la que la banca entrega valor a sus clientes bajo un esquema de trabajo tradicional en ocasiones lo coloca en desventaja frente a estos nuevos “startups” que captan la atención de las nuevas generaciones nativas digitales, así como del mercado tecnológicamente activo.

La metodología ágil aparece como opción para remediar la falta de “agilidad” en el desarrollo y entrega de valor ya que, a diferencia de su contraparte, la metodología tradicional, las prácticas ágiles permiten:

- Capacidad para responder a los cambios que surgen al largo del desarrollo
- Entrega continua de software funcional en plazos cortos
- Cliente y equipo de desarrollo trabajan en conjunto
- Se elimina trabajo innecesario, predomina la simplicidad
- Mejora continua de los procesos y equipo de desarrollo
(Rebaza, 2007)

En algunas matrices bancarias de la ciudad de Guayaquil manejan un esquema de “TI de dos velocidades ”, es decir que dependiendo de la naturaleza del proyecto a desarrollar se opta ya sea por una metodología de desarrollo tradicional tipo cascada o una ágil (Heredia, 2015), sin embargo estas metodologías por sí solas, aunque siguen siendo eficientes y permiten el desarrollo y entrega de valor, no es suficiente; para lograr competir con las

startups como Fintechs, se requiere agilidad a nivel de todo el ciclo de vida de desarrollo del software incluyendo su entrega y operación. La cultura DevOps ha surgido como una opción bastante recomendada para lograr un mayor nivel de agilidad y velocidad en el desarrollo y entrega de software en la banca, ejemplos de casos de éxito son el de la consultora Nisum, la cual se encargó de implementar la cultura DevOps en un importante banco internacional, este banco a pesar de contar con un marco ágil no estaba satisfecho con la calidad y velocidad en que se entregaba software y se implementaba en ambiente de producción, con DevOps en conjunto con Agile fue posible lo siguiente:

- Reducir en un 66% los tiempos de paso a producción
- Mayor calidad con el aumento de la cobertura de pruebas en cada requerimiento
- Más ciclos de feedback para que los desarrolladores puedan trabajar en nuevas características
- Mayor fidelidad de los clientes

(Consultora Nisum, 2017)

Un similar caso de éxito se dio con la consultora española Cognizant quienes asistieron a un importante banco con cobertura mundial a mejorar el enfoque de desarrollo y despliegue de software, esto se logró de igual manera con la adopción de la estrategia DevOps para la automatización de procesos de despliegue y creación ya existentes, este caso dio resultados bastante optimistas que superaron las expectativas del banco:

- Reducción del 19% en defectos funcionales
- Reducción del 40% de inactividad de aplicaciones
- Ahorro del 90% en costos mediante automatización

(Consultora Cognizant, s. f.)

En la ciudad de Guayaquil Banco del Pacífico se encuentra en proceso de adopción de esta cultura por parte de la consultora Maint S.A. (Jorge Gonzalez, Gerente de proyectos Maint S.A, comunicación personal, 2020). Así como los previamente mencionados hay casos de éxito en bancos nacionales e internacionales que han posicionado a esta cultura como la

favorita para mejorar y agilizar el proceso de desarrollo y despliegue de aplicaciones en la banca.

2.2 ELEMENTOS TEÓRICOS

2.2.1 CULTURA DEVOPS

La cultura DevOps, como la palabra “cultura” la define es un movimiento, ideología o filosofía que promueve un conjunto de valores, principios y prácticas que buscan afianzar la relación entre los departamentos de desarrollo y operaciones del área de TI. Una definición más semántica define a DevOps como una palabra compuesta que, reúne las mejores prácticas de TI existentes de ITIL, Agile y Lean enfocadas al desarrollo y operaciones de las empresas, sea cual sea la naturaleza de esta.

DevOps tuvo sus inicios en el año 2007 por Andrew Shafer en la “Agile Conference” realizada en Toronto (aunque en ese entonces aún no se concebía con su nombre actual DevOps), Shafer preparó una charla express con el objetivo de explicar las ventajas y el panorama general de la aplicación de la metodología Agile a la infraestructura; a este evento únicamente se inscribió un asistente, Patrick Devois. Devois en aquel entonces se encontraba liderando un proyecto de migración de un gran Datacenter y sentía que el trabajo podía mejorarse y agilizarse de alguna manera y las teorías presentadas por Shafer eran lo que él buscaba para crear un nuevo marco de trabajo; juntos fundan la “Agile System Administration” en la cual trabajan en esta metodología e intercambian información.

En el año 2009 se celebra la Conferencia de Velocidad de O`Really, durante la charla “10 Despliegues por día” se escucha por primera vez la idea de la fusión de los departamentos de desarrollo y operaciones de TI. Devois, al no poder asistir a esta charla debido a la distancia decide crear su propia conferencia la cual denomina “DevOpsDay” celebrada en Bélgica, a esta asisten desarrolladores, administradores de sistemas, personal de operaciones los cuales comienzan a dar forma a la hoy famosa Cultura DevOps. Es a partir de este punto que el término DevOps comienza a

popularizarse a nivel mundial como sinónimo de automatismo y gestión eficiente de los recursos en TI (*The (Short) History of DevOps*, 2012).

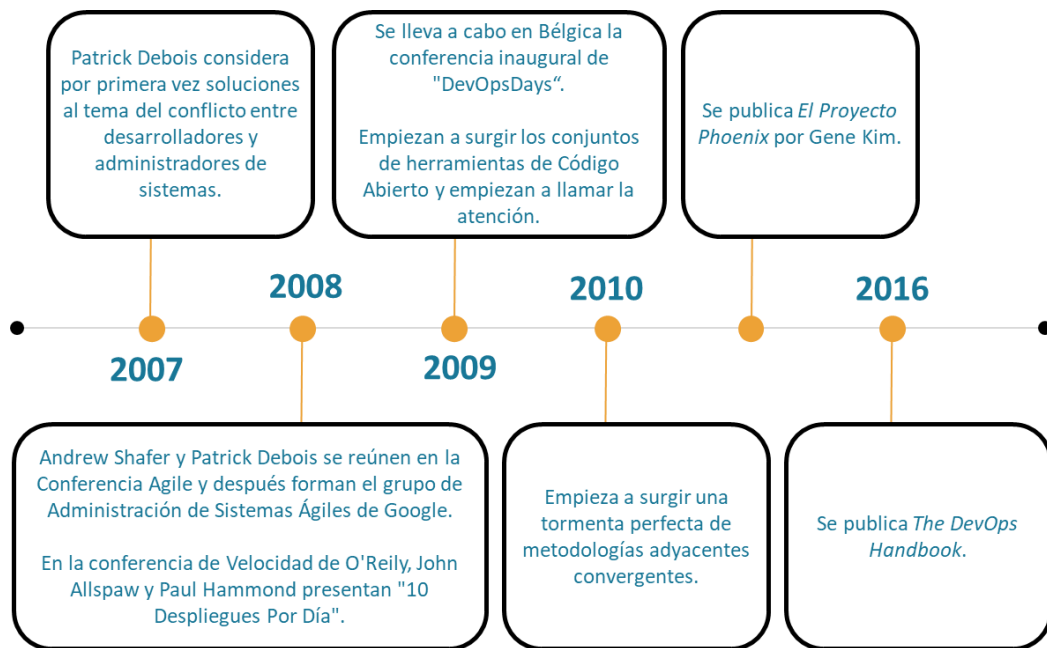


Figura 1 Historia de DevOps

Tomado de "Fundamentos de DevOps", por PeopleCert, 2019

El objetivo principal de DevOps es ayudar al área de TI a entregar valor al negocio de manera más rápida, con menores costos y eficiente a través de la fomentación de una cultura de colaboración y aprendizaje.

En resumen, DevOps es:

- Movimiento, ideología o filosofía
- Trata la relación entre Desarrollo y Operaciones de TI
- Cambio cultural implementado a la organización entera
- Enfocado a la mejora de comunicación y colaboración
- Automatización y nuevas herramientas para la comunicación
- Equilibrio entre velocidad y el cambio requerido por área de Desarrollo con confiabilidad y estabilidad requerida por Operaciones

(PeopleCert, 2019)

Microsoft define a DevOps como una combinación de términos de la lengua inglesa (Development y Operations) cuya traducción literal es

Desarrollo y Operaciones respectivamente, el objetivo es lograr la unión de procesos, personas y tecnología de manera que sea posible la entrega de valor a los clientes constantemente. Dicho de otra manera, DevOps permite que los roles que antes trabajaban de forma aislada y con objetivos individualistas se coordinen y colaboren entre sí para la producción de productos de mejor calidad y más confiables (Microsoft Azure, 2020).

Ambas definiciones son bastante cercanas y dejan en claro la importancia de DevOps para lograr entregas de valor al cliente con mayor rapidez y calidad, no en vano distintas empresas a nivel mundial han adoptado o están en camino a adoptar esta cultura, habiendo ya varios casos de éxito cuyos resultados sobrepasaron las expectativas, entre estas empresas se encuentran los bancos quienes frente a la evolución tecnológica, crecimiento de la demanda y de su competencia buscan nuevas alternativas que les permitan mantenerse competitivos en el mercado financiero, siendo DevOps la opción favorita por excelencia para muchos de ellos.

Así como se define lo que es DevOps es importante señalar qué no es DevOps; existe una gran confusión al concebir a DevOps como un puesto de trabajo o tipo de equipo específico, esta es una cultura que debe ser adoptada por el personal de las áreas de Desarrollo y Operaciones de TI, no solo debe ser conocida por una persona (ingeniero DevOps) o un grupo específico. Otra idea errónea se da al pensar que únicamente es aplicable a nuevas empresas o aplicaciones web, DevOps es una cultura que puede ser aplicada a empresas de todo tamaño sin importar la naturaleza de su negocio, desde instituciones financieras a, empresas de entretenimiento, los pioneros de DevOps han sido Spotify, Netflix y Amazon por mencionar algunos ejemplos, sin embargo negocios que han existido desde hace ya varias décadas son capaces de adoptar esta cultura como: Nasa, Stanford, Walmart, entre otras (PeopleCert, 2019).

Otro concepto bastante errado de DevOps es que busca eliminar las Operaciones de TI, esto es solo un mito ya que, si bien la naturaleza del trabajo del área de operaciones de TI puede cambiar, sigue siendo tan importante como siempre. El cambio que DevOps busca es lograr que

Operaciones de TI colabore con Desarrollo prácticamente desde el inicio del ciclo de vida del desarrollo de las aplicaciones, y que Desarrollo continúe trabajando con Operaciones de TI aún después de la salida a producción del producto (Kim et al., 2016).

En definitiva, DevOps, es un nombre utilizado para describir una evolución de las mejores prácticas de TI existentes de ITIL, Lean y Agile en un enfoque de Desarrollo y Operaciones que promulga la automatización y la entrega continua, y fomenta una cultura de colaboración y aprendizaje para ayudar a TI a ofrecer un mejor valor empresarial, más rápido y con menores costos.

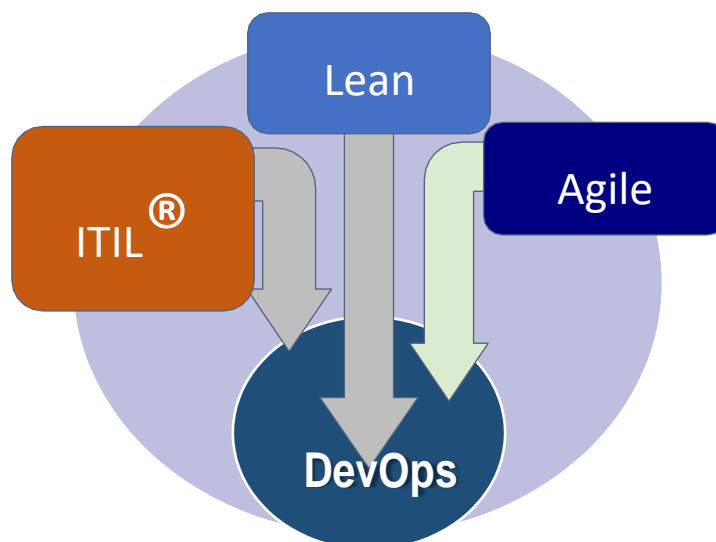


Figura 2 Complementos de DevOps

Tomado de "Fundamentos de DevOps", por PeopleCert, 2019

Los bancos a nivel mundial y por su puesto en Ecuador, han identificado el uso de la tecnología para aumentar su eficiencia y, al mismo tiempo, ofrecer una mejor experiencia a sus usuarios. Según la consultora McKinsey, algunos bancos están experimentando con enfoques de automatización rápida, logrando resultados prometedores. Estas pruebas han demostrado que la automatización de los procesos de extremo a extremo, que solía llevar de 12 a 18 meses o más, es factible en 6 meses, y normalmente requieren la mitad de la inversión (*Automating the Bank's Back Office | McKinsey, 2012*).

En los procesos convencionales que se aplican en algunos casos aún, necesitan de tiempo considerable para planificar el proyecto, relevamiento y validación de los requerimientos, el desarrollo de la aplicación, algunos días para asegurar la calidad (Quality Assurance), otros más para manejar problemas de implementación, mantenimiento y en muchos casos se necesita regresar atrás debido a la falta de coincidencia entre lo que se esperaba y lo que se entrega.

Por lo expuesto, un alto volumen de proyectos considera meses para poder ser entregados o liberados en producción. En este contexto, DevOps ayuda a realizar la implementación de manera rápida. Por lo tanto, el propósito general de DevOps es lograr la rapidez en el despliegue, para ello:

- DevOps recomienda establecer los servicios de TI como una cadena de suministro para el negocio. Es un gran cambio de paradigma desde la entrega del software hasta la prestación de servicios TI.
- Desde el punto de vista arquitectónico, DevOps requiere que las empresas definan un sistema de despliegue automatizado rápido.

DevOps para poder alcanzar su propósito plantea cambios en 3 aristas principales, esto se conoce como el “Paquete Completo de DevOps” que comprende:

- Automatización: Lograr la automatización de procesos y medición mejorada por medio de herramientas y conjunto de herramientas nuevas.
- Prácticas: Integrar elementos de Agile, ITIL, y Lean para la mejora de la eficiencia y el flujo de trabajo.
- Cultura: Crear y fomentar una cultura de colaboración y comunicación.

(PeopleCert, 2019)



Figura 3 Paquete Completo de DevOps

Tomado de "Fundamentos de DevOps", por PeopleCert, 2019

El paquete completo de DevOps es el pilar fundamental que debe alcanzarse para una exitosa implementación de esta cultura en cualquier negocio, incluida la banca; Bill Thomas, presidente de KPMG Internacional, resalta que en la actualidad los CEO deben sentirse cómodos y fomentar un cambio disruptivo a sus modelos de negocio. Esto se refleja en los resultados de una encuesta realizada por la consultora internacional KPMG en la que se evidenció que el 54% de los CEO encuestados en América Latina consideran que trabajar de forma ágil es la manera de mantenerse competitivos en el mercado, ya que si la organización actúa de forma "lenta" frente a la competencia y los cambios puede estar encaminándose a la bancarrota; así también como parte del proceso de cambio a agilidad, se debe invertir en tecnología que les permitan automatizar los procesos claves que aseguren una entrega rápida y continua; esto se evidencia en los resultados de la encuesta la cual concluyó que 61% de los CEO de América Latina optan por la inversión en tecnología sobre el desarrollo de su fuerza laboral (Revista Online Vanguardia Industrial, 2019).

Otro de los aspectos resaltados en la encuesta mencionada, indica que los CEO tienen buenas razones para insistir sobre agilidad y sus beneficios. Las metodologías ágiles realmente pueden acelerar el desarrollo de productos, generar nuevos valores e impulsar el cambio organizacional. Es por eso por lo que la mayoría de las empresas al menos han comenzado algún tipo de transformación ágil, pero el siguiente paso que es llevar un producto al mercado al mismo ritmo en que se desarrolló, está resultando complejo. La mayoría de las empresas simplemente no están preparadas culturalmente para ese tipo de cambio radical.

Justamente DevOps propone los siguientes valores para lograr una verdadera entrega de valor a las áreas de negocio al mismo tiempo que se desarrolla (Five Cultural Changes You Need for DevOps to Work | McKinsey & Company, 2017).

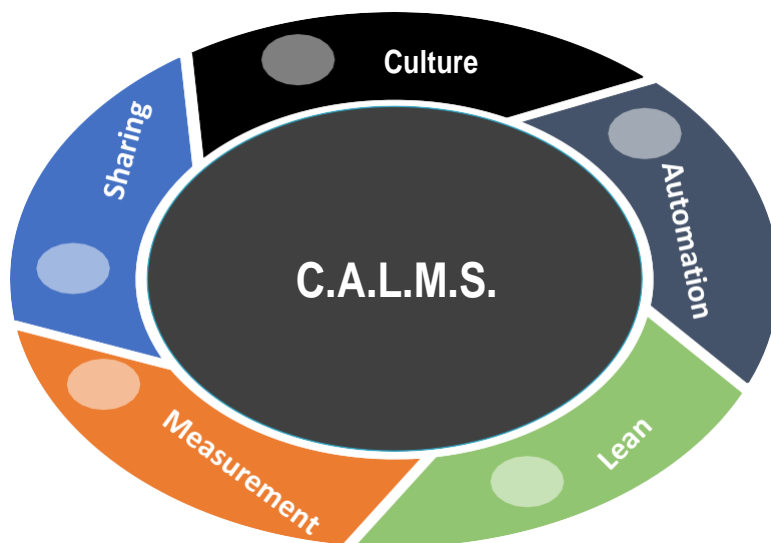


Figura 4 Valores de DevOps

Tomado de "Fundamentos de DevOps", por PeopleCert, 2019

- Cultura: Un entendimiento de que los valores, las creencias y las actitudes que permean una organización y TI son importantes y necesitan cambiar
- Automatización: Una creencia de que cualquier cosa puede ser automatizada y que los procesos manuales deberían automatizarse cada que sea posible

- Lean: Un énfasis en ideas del sistema de valor y la metodología de Lean para reducir la complejidad de los procesos y agilizar el flujo del trabajo
- Medición: Una creencia en la importancia de medir todo y construir procesos que sean visibles y transparentes
- Compartición (Sharing): Un énfasis en la colaboración y la comunicación entre Desarrollo y Operaciones, permitiéndoles integrarse de manera eficaz

(PeopleCert, 2019)

2.2.2 VALOR DEL NEGOCIO

Según Philip Kotler, valor "es la diferencia entre los beneficios que espera recibir del satisfactor y el costo total que debe soportar" Hecha esta aclaración, la diferencia entre los beneficios que brinda tener un producto y su costo es lo que tiene como resultado el valor (ESAN Graduate School of Business, 2017, párr. 3).



Figura 5 Dimensiones del valor de negocio

Tomado de "Fundamentos de DevOps", por PeopleCert, 2019

- Calidad: Garantizar que el producto y los requerimientos, cubrirán las necesidades del cliente
- Costo: Se mide contra la inversión o cuánto debe pagar el cliente
- Velocidad: El producto debe estar listo cuando el cliente lo necesita

Hablando específicamente de TI, el impacto de esta se puede limitar a 3 principales aspectos:

Calidad reducida

- Defectos no detectados: Los defectos pasan de desarrollo a operaciones y producción sin ser identificados
- Mala disponibilidad del Servicio: Los sistemas legados, demasiado complejos o frágiles hacen difícil implementar los cambios
- Incapacidad de integrar datos: Los sistemas de TI desarrollados como pilas de aplicaciones independientes con su propia estructura de datos y base de datos carecen de interoperabilidad con otros sistemas de negocio

Velocidad Reducida

- Baja velocidad para entregar productos: Los sistemas y procesos demasiado complejos dan como resultado defectos y retrabajo incrementados y entregan las cosas más lento
- Falta de prioridades: No existen o son débiles los procesos para establecer acuerdos sobre prioridades de requerimientos de múltiples unidades de negocio
- La constante y creciente demanda de las diferentes unidades de negocio y la limitación de recursos, genera que los pendientes del trabajo sigan creciendo y no sean atendidos

Incremento en costos

- Costos de mantenimiento en aumento: La falta de priorización para eliminar, estandarizar o consolidar procesos y herramientas aumenta los costos de mantenimiento con el tiempo

- Estancamiento del crecimiento: Conforme se eleva el presupuesto para seguir operando, disminuye la capacidad de financiar la innovación y el crecimiento
- Incapacidad de mapear los costos al valor: La falta de visibilidad de los costos reales para entregar servicios limita la inversión (PeopleCert, 2019)

2.2.3 DIFICULTADES DE TI PARA AGREGAR VALOR AL NEGOCIO

Un alto número de instituciones bancarias son instituciones con una gran historia y permanencia sólida en los mercados, por ende han venido adoptando culturas de trabajo y tecnología que soportaba eficientemente su operación en el momento en que fueron adquiridos, la inversión realizada por los bancos en tecnología y aplicaciones, así como el alto grado de personalización (adaptación de los sistemas a los requerimientos propios y específicos de la institución) dificulta el reemplazo en el corto tiempo ya que generalmente estas aplicaciones y/o infraestructura soportan los principales procesos y productos o servicios que brinda a sus clientes (core bancario), siendo por ello muy sensible los cambios o actualizaciones de las mismas.

Estas aplicaciones en el tiempo se convierten en tecnología denominada “legacy” y requieren un alto nivel de especialización para su mantenimiento surgiendo la dependencia de proveedores o de especialista del área de tecnología de las instituciones (F. Suarez, comunicación personal, 2020). Esto a su vez afecta la habilidad de los bancos de ofrecer nuevas experiencias, productos y servicios de calidad a sus clientes de manera veloz y continua (Consultora Cognizant, 2020). Otro gran problema con la tecnología “legacy” es que no siempre soportan la automatización de los procesos y la integración de estos, siendo esta una de las principales preocupaciones de las áreas de tecnología de los bancos para poder cumplir con las nuevas exigencias de los clientes.

Finalmente, uno de los principales problemas que afrontan no solo los bancos sino las empresas con más tiempo en el mercado es la especialización que las áreas han desarrollado en el tiempo, fomentando los llamados silos de información que dificultan la integración y la conformación de equipos

multifuncionales. Se denomina como silo de información a aquellos elementos, departamentos o inclusive personas entre los cuales no existe una transferencia de información y conocimientos; los directores de TIC coinciden que, los silos de información indican que un empresa está anclada aún en el pasado al no ser capaz de trabajar de forma ágil (KORPORATE TECHNOLOGIES GROUP, 2017). Los bancos de la ciudad de Guayaquil no son la excepción, la presencia de tecnología Legacy, así como los silos de información limitan a estas empresas para adoptar una cultura como DevOps (F. Suárez, comunicación personal, 2020).

2.2.4 DEVOPS Y LAS MEJORES PRÁCTICAS DE LA INSDUSTRIA

La visión general de DevOps es conseguir la integración entre las áreas de Desarrollo y Operaciones de TI, de modo que se logre una mejor comunicación, colaboración e integración entre los equipos de ambas áreas, sin embargo, aunque es bastante eficiente como se puede ver en casos de implementación exitosa en varios bancos importantes tanto nacional e internacional, no significa que esta cultura reemplaza todos los demás marcos de trabajo. Las mejores prácticas para gestionar la operación han sido adoptadas en muchas empresas teniendo éxito en su gestión, implementar DevOps no significa reemplazar todo lo que ya funcionaba anteriormente, al contrario DevOps puede complementarse perfectamente con marcos como Agile, ITIL, Lean entre otras potenciando aún más la capacidad de crear valor de las empresas, particularmente los bancos (Andersson, 2017).

DevOps busca romper el esquema de silos y departamentos trabajando de forma independiente, por ello es bastante efectiva esta cultura en empresas que requieren altos niveles de agilidad y flexibilidad como los bancos, que deben ser veloces para ofrecer valor a sus clientes y mantenerse competitivos en el mercado, sin embargo, los bancos por la naturaleza de su negocio necesitan controles por tanto, aun teniendo una cultura de agilidad, necesitan un mínimo de nivel jerárquico; es aquí donde ITIL aporta a DevOps con la administración de operaciones de manera consistente y responsable, con el establecimiento de expectativas y con la definición de servicios que

administran cambios e incidentes, estos son factores universales de ITIL sin los cuales ninguna organización sensata podría sobrevivir (Rae, 2017).

ITIL V4 posee un pieza clave denominada SVS o Sistema de Valor del Servicio, ITIL SVS tiene por objetivo que los distintos componentes y actividades de la organización trabajen en conjunto para facilitar la creación de valor mediante servicios, habilitados por el departamento de tecnología (tecmanag, 2019).



Figura 6 Componentes de ITIL V4

Tomado de "Fundamentos de DevOps", por PeopleCert, 2019

Como lo indica el ingeniero Paúl Vintimilla, consultor certificado ITIL V4, el framework ITIL, no pretende ser una solución para todos los problemas, al contrario el framework recomienda adoptar y adaptar únicamente los componentes que, en base a los servicios soportados y la tecnología de TI, agreguen auténtico valor a sus servicios; esto en conjunto con DevOps potencia en gran medida la productividad del área de Operaciones de TI (P. Vintimilla, comunicación personal, 13 de junio de 2020).

Lean por otra parte es otra de las mejores prácticas que puede complementarse con DevOps. La metodología Lean fue ideada por el fundador de Toyota, Sakichi Toyoda, a finales del siglo XIX y nació con el objetivo de identificar errores en la producción que generaban retrasos en el trabajo así como mala calidad del producto (PROGRESSA LEAN, 2015);

aunque originalmente se pensó para empresas de manufactura, por ello su nombre original de Lean Manufacturing, hoy en día esta metodología es aplicable a todo tipo de empresas.

Los principios del pensamiento Lean son los siguientes:

- **Eliminar el desperdicio:** Es el principal principio de esta metodología, se basa en realizar un análisis exhaustivo de cada etapa identificando cuáles aportan verdadero valor, cuáles ocupan tiempos de espera elevados, con el objetivo de identificar posibles cuellos de botella y puntos de mejora
- **Amplificar el aprendizaje:** Este principio se relaciona directamente con DevOps ya que Lean, como DevOps, se enfoca en crear una cultura de aprendizaje basada en la mejora continua
- **Entrega lo más rápido posible:** Este principio está directamente relacionado con una de las 15 prácticas esenciales de DevOps, con este se busca disminuir los tiempos del ciclo de desarrollo
- **Decidir lo más tarde posible:** Este principio está directamente relacionado con el principio anterior, ya que, si se logra entregas rápidas, es posible disponer de mayor tiempo para el análisis de la información recolectada, evitando la toma de decisiones con información errónea o incompleta
- **Empoderar al equipo:** Este principio de Lean a su vez es un pilar básico de DevOps, dicho de otra manera, la eliminación de silos para poder visualizar al equipo como un todo más que la suma de sus partes individuales
- **Embeber la integridad:** El pensamiento Lean se centra en la calidad del producto, por ello está estrechamente relacionado con testing y aseguramiento de la calidad
- **Considerar el sistema como un todo:** La entrega de valor es un sistema completo, no solo es el resultado de una actividad, de un proceso, de un equipo, del consumidor ni del mercado; es el resultado del impacto que generan cada una de estas áreas en consecuencia de las decisiones tomadas

(Sutil, 2017)

DevOps y Lean son dos pensamientos bastante compatibles entre sí, al analizar a detalle los principios de Lean se puede observar cuando se habla de eliminación de desperdicios visto desde una perspectiva DevOps y el desarrollo de software, desperdicio es aquello que no aporta valor, como código parcialmente terminado o bugs, pero sin duda uno de los mayores desperdicios en el ciclo de vida del software es el movimiento de trabajo entre equipos. En DevOps es esta separación la que también se quiere tratar ya que, si se logra la unión entre Dev y Ops, este desperdicio generado en el paso de información, tareas y/o códigos entre los departamentos de Desarrollo y Operaciones de TI, no se evidenciaría o bajaría sustancialmente; de igual manera los tiempos desperdiciados en toma de decisiones por separado se reducirían drásticamente al trabajar estos 2 departamentos en conjunto.

Teniendo como base las mejores prácticas de la industria (Lean, ITIL, Agile) el DevOps Institute promueve las siguientes 15 prácticas esenciales que son el marco a considerar:

- Alinéate con el valor de negocio definiendo la Voz del Cliente
- Aumenta la colaboración y derriba los silos con la Gestión de Relaciones
- Identifica las restricciones y los cuellos de botella con el Mapeo de Cadenas de Valor
- Elimina el desperdicio en donde sea posible y reduce las entregas con la Optimización Lean de Procesos
- Habilita la compartición del conocimiento y crea una cultura de aprendizaje con la Gestión del Conocimiento
- Crea transparencia en la cadena de valor y haz el trabajo visible con la Gestión Visual
- Trabaja en lotes más pequeños y crea ciclos de retroalimentación con Scrum 'Agile'
- Detecta problemas conforme ocurren y falla más rápido al Mover las Pruebas a la Izquierda

- Logra el equilibrio correcto de flexibilidad y estabilidad con el Control de Cambios
- Incorpora información y recursos donde los necesites, cuando los necesites con la Gestión de la Configuración del Servicio
- Apoya la automatización y crea flujo con la Gestión de Liberaciones y Despliegues
- Resuelve los incidentes proactivamente con la Gestión de Incidentes
- Convierte el conocimiento en mejoras que acerquen la calidad a la fuente con la Gestión de Problemas y Kaizen Lean
- Institucionaliza la mejora diaria con modelos de Mejora Continua
- Acepta el fracaso como una oportunidad de mejora con un enfoque en la Antifragilidad

(PeopleCert, 2019)

El sector financiero ha liderado y sigue liderando la adopción de últimas tecnologías y prácticas modernas de entrega de software. El 84% de los banqueros cree que DevOps y las plataformas modernas en la nube impulsarán la transformación bancaria (Temenos – Company, 2020).

2.2.5 HERRAMIENTAS DE DEVOPS

DevOps fomenta la integración entre los equipos de desarrolladores y los profesionales de Operaciones de TI, tendiendo a lograr un ciclo iterativo que busque siempre la mejora continua y la entrega de valor al negocio. DevOps tiene su propio ciclo de vida en relación con la automatización del proceso de desarrollo de software desde su concepción (idea) hasta su operación (producto en producción) (Villegas, 2019), el cual en general está compuesto por 8 etapas, aunque estas pueden variar, ya sea porque se simplifiquen etapas o se agreguen componentes de cada una de ellas (Quijano, 2018).



Figura 7 Ciclo de vida DevOps

Tomado de "Herramientas DevOps", por Jaivic Villegas, 2019

Entre los principales grupos de herramientas que requiere DevOps se encuentran los siguientes:

Manejo del código fuente

Control de versiones, como también se lo conoce, es una buena práctica de seguimiento de cambios en el código de un software, ya sea para el desarrollo de una aplicación sencilla que requiera el esfuerzo de una sola persona, como para el desarrollo de un proyecto de software a gran escala que involucre el trabajo de un equipo. Los sistemas de administración de código fuente permiten llevar un historial de cambios registrando las diversas versiones de un código desarrollado por varias personas a la vez, e inclusive permite realizar rollback o volver a una versión anterior en caso de errores en el código más actual; una de las funciones más importantes es la posibilidad de trabajar en equipo en el código, ya que al aislar el trabajo asignando la codificación de distintas partes del código a distintos desarrolladores se logra avanzar más rápido con el desarrollo del software (Amazon Web Services, Inc., 2020).

Estas herramientas son especialmente útiles para la cultura DevOps ya que ayudan a optimizar el proceso de desarrollo de software, mediante la mejora en la comunicación entre los miembros involucrados en la codificación con herramientas de mensajería, medición de la eficiencia con la que opera un proyecto de desarrollo de software mediante el monitoreo de indicadores de eficiencia o KPI y la automatización de funciones como pruebas o compilaciones de código (Atlassian, s. f.).

Integración continua

La integración continua de forma simple es la combinación o merge del trabajo individual de varios desarrolladores al código raíz, cada uno trabaja sus distintas partes de su código y una vez terminado lo integra al código principal para ser compilado, con esto cada vez que se combina una nueva parte de código se realiza una serie de pruebas que permiten detectar errores y solucionarlos a tiempo, la entrega continua por otra parte se asegura de lograr que el software esté preparado para su lanzamiento en cualquier momento, finalmente la implementación continua se encarga que los cambios realizados al software se apliquen de forma automática sin afectar el funcionamiento.

La cultura DevOps busca lograr que los ciclos de desarrollo de software sean más rápidos, en consecuencia de esto los ciclos de entrega son cada vez más cortos, por ello la liberación de software que antes se programaba de forma mensual o trimestral ahora se puede dar de forma diaria o inclusive varias veces al día; corregir y revisar código de forma manual con entregas de código constantes se vuelve virtualmente imposible y aún en el caso en que fuera posible hacerlo el tiempo y esfuerzo invertido retrasaría el desarrollo. A causa de esto DevOps recomienda el uso de herramientas que automaticen el despliegue, entrega e integración de código de modo que si hay cambios, la actualización se haga de forma inmediata y transparente para el usuario (Castellanos, 2019).

Virtualización

La integración continua en la cultura DevOps requiere de un testeo regular del código desarrollado día a día, bajo este contexto, la virtualización de servicios resulta bastante necesario ya que proporciona un entorno ininterrumpido para las pruebas (Palacios, 2020), este entorno simula ambientes que cumplan la mismas características y requisitos que los entornos donde será instalado el software, es decir, permite a los desarrolladores y testers una réplica del entorno final, este ambiente permite habilitar todas las fases de prueba previas a las pruebas de usuario final (Israel, 2019).

Cloud

La agilidad y comodidad para gestionar infraestructura y la adaptabilidad de las organizaciones son dos aspectos fundamentales de DevOps, los cuales se pueden alcanzar gracias a Cloud Computing. En un principio Cloud Computing era visto por las empresas como una forma de ajustar el presupuesto ya que se optaba por un proveedor de servicios o IaaS (Infrastructure as Service) en lugar de invertir en infraestructura física, con Cloud únicamente se paga por uso, no hay inversiones iniciales o compromisos contractuales. En el presente ya no solo se considera que el único beneficio de Cloud es un ahorro de carácter económico, su principal beneficio es la facilidad y rapidez con que la Nube aprovisiona y mantiene la infraestructura de TI para ejecución del código fuente y la continuidad de la operación, siendo los beneficios la reducción de costos y el aumento de la competitividad (Arsys, 2016).

La utilización de estas herramientas DevOps habilitan que el desarrollo del software este bajo control garantizando calidad en el resultado de forma mucho más ágil y eficaz, tanto para programadores como para el equipo de Operaciones de TI.

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

Todo trabajo investigativo debe estar acompañado de los debidos métodos y tipos de investigación, los cuales permitan orientar la búsqueda de conocimientos del proyecto, además de las técnicas y herramientas de levantamiento de información pertinentes al problema a resolver.

Por lo expuesto; este capítulo está dedicado explícitamente a la identificación del tipo de investigación, población y herramientas utilizadas para el levantamiento de información. Así también se realizará el análisis de la información obtenida.

3.1 TIPO DE INVESTIGACIÓN

La metodología de la investigación se define como una actividad muchas veces descrita como reflexiva, metódica y sistemática que tiene por objetivo la obtención de conocimientos y soluciones a distintas problemáticas (EcuRed, 2019). Dependiendo de la naturaleza del proyecto, la problemática y demás aspectos del proyecto, el tipo de investigación seleccionado deberá ajustarse a estos aspectos y deberá ser la adecuada para obtener la información necesaria para validar el proyecto de investigación.

Para este trabajo en particular se hará uso del tipo de investigación exploratoria, ya que esta representa el primer nivel de investigación que es lo que se busca con este trabajo, trazar los primeros pasos con una guía la cual, sujeta a posteriores investigaciones puede ser mejorada, además la naturaleza propia de esta metodología de investigación plantea la recolección de información por parte de expertos de la materia, así como de los beneficiarios de esta investigación, por lo que este es el método de recolección de información que se plantea usar (E-nquest, 2018).

Por otra parte la investigación empleada será de dos tipos, en primer lugar está la investigación documental, ya que la realización de las guías se apoya en el sustento teórico de DevOps hallado en la bibliografía de DevOps, y sitios oficiales como el DevOps Institute, el segundo tipo de investigación es la de

campo ya que, además de estar apoyada en documentación la guía será moldeada en base a la información obtenida de expertos del área con su experiencia en proyectos de implementación de esta cultura en bancos nacionales e internacionales.

3.2 POBLACIÓN Y MUESTRA

3.2.1 POBLACIÓN

El presente trabajo de investigación está orientado a las matrices bancarias de la ciudad de Guayaquil, la recopilación del listado de bancos obtenido de la Superintendencia de Bancos es el siguiente (SUPERINTENDENCIA DE BANCOS, s. f.):

- Banco de Amazonas
- Banco de Guayaquil
- Banco del Pacífico
- Banco Bolivariano
- Banco del Bank
- Banco Coopnacional
- Banco D-Miro
- Banco Litoral

De estos 8 bancos se entrevistan a perfiles del departamento de TI de cada banco con el fin de obtener información que sustente el problema, así como el nivel de aceptación de la solución. Los perfiles entrevistados varían entre desarrolladores, oficiales de arquitectura, personal involucrado con pruebas del software, gerentes de proyectos y ejecutivos de operaciones y Desarrollo; estos distintos perfiles permiten recopilar información desde los distintos puntos de vista de cada área que está involucrada en el desarrollo y despliegue de software, obteniendo así información de los departamentos en los que se enfoca la cultura DevOps, que son desarrollo y operaciones, además de un punto de vista más global gracias a los gerentes de proyectos y Gerentes de áreas.

Adicional, aunque no forman parte de la población se entrevistó a 3 consultores con el objetivo de obtener información útil para el desarrollo de las guías basado en la experiencia de estos expertos DevOps.

Debido al tamaño de la población, para la aplicación de las encuestas se considera la totalidad de la población para ser encuestada.

3.3 INSTRUMENTOS DE RECOLECCIÓN DE DATOS

La obtención de información es una de las partes críticas de un proyecto de investigación, ya que la confiabilidad y validez del estudio dependerá de la información levantada y el respectivo análisis de esta. Los datos obtenidos deben ser confiables y suficientes para sustentar el trabajo investigativo, por ello se debe seleccionar cuidadosamente los instrumentos de levantamiento de información a emplear (*ITSON*, s. f.).

Varios autores indican que la selección de la herramienta o técnica de recolección de datos dependerá de la información a obtener, así como de la metodología de la investigación empleada. Existen varias técnicas de recolección de información a disposición, algunas son la entrevista y la encuesta; la entrevista se considera como técnica que emplea el contacto directo con las personas consideradas fuente de información primaria, la encuesta es una de las técnicas más usadas, la cual consiste en una serie de preguntas, abiertas, cerradas o semiabiertas, preparadas con el propósito de obtener información de las personas encuestadas (Ly & Siesquén, 2012).

La observación directa se ha convertido en una herramienta que cada vez cobra mayor relevancia y credibilidad debido a que está orientada a obtener información directa del contexto investigado. Existen más herramientas útiles para levantar información, pero para el presente trabajo investigativo se han empleado 2 herramientas en particular las cuales se describen a continuación.

La encuesta está dirigida a los funcionarios de las áreas de tecnología de las matrices bancarias de la ciudad de Guayaquil, está compuesta por 14 preguntas de opción múltiple, algunas de múltiples respuestas y otras de respuesta única, el formato de esta se encuentra adjunta como Anexo 1.

Debido a que la información compartida por los funcionarios de los distintos bancos es sensible, se ha considerado necesario el mantener la confidencialidad de la información, por lo que no se puede revelar el nombre del banco en el que cada funcionario encuestado labora.

El objetivo perseguido con esta herramienta es la evaluación de los siguientes aspectos:

- Ciclo de entrega de los productos o servicios tecnológicos hacia las áreas del negocio de los bancos
- Acercamiento de los bancos con los marcos ágiles
- Familiarización de los bancos con la cultura DevOps
- Razones por las que los bancos consideran importante la adopción de DevOps
- Valor que representa para los bancos el contar con unas guías para su proceso de autoevaluación con el objetivo de la adopción de DevOps

Por otra parte la entrevista, cuya tabulación se encuentra adjunta como Anexo 2, está orientada a consultores que han formado parte de proyectos de adopción de DevOps, tanto a nivel local como internacional y los aspectos a determinar con esta herramienta son los siguientes:

- Conocer si los consultores entrevistados cuentan con experiencia en proyectos de implementación de DevOps en bancos de Ecuador y/o a nivel Internacional
- La importancia de la adopción de DevOps por las instituciones bancarias
- ¿Cómo deben prepararse los bancos para iniciar el proceso de adopción de esta cultura?
- Mecanismos que pudieran ayudar a los bancos a iniciar por su cuenta el proceso de adopción
- Posibles factores que impiden a los bancos la adopción de la cultura DevOps
- Evaluación de los niveles de madurez en el proceso de adopción de DevOps en base a la experiencia de los consultores

- Conocimiento de las herramientas para la automatización en los diferentes estados de madurez de la adopción de DevOps
- Evaluación de los aspectos organizativos y conformación de equipos
- ¿Cómo cambiar el enfoque vertical (silos) de los bancos a un entorno de mayor colaboración de pares y equipos de trabajo?
- Identificación de indicadores que permitan la evaluación del proceso de adopción de DevOps

3.4 SELECCIÓN DE LOS ENCUESTADOS

La selección de los encuestados se realizó en base a los siguientes criterios:

- Que pertenezcan a áreas de desarrollo y operaciones de TI, áreas claves para la cultura DevOps
- Los roles variados considerando líderes, jefaturas, gerencias, desarrolladores y técnicos de áreas de desarrollo y operaciones de TI
- Años de experiencia desempeñando su rol en el mercado financiero (bancos)

Los roles encuestados fueron:

- Analista de gestión de pruebas
- CIO (Chief Information Officer)
- Desarrollador de aplicaciones
- Gerente de producción
- Gerente de desarrollo
- Líder de sistemas
- Oficial Senior Arquitectura e Innovación
- Arquitecto de Datos
- Ejecutivos (Jefatura en área de desarrollo y operaciones)

La variedad de roles encuestados permite tener las distintas perspectivas desde cada una de las funciones que cumplen los encuestados en los diferentes niveles estructurales del área de TI de los bancos, la información entregada genera alto valor para la elaboración y sustentación del resultado de este trabajo investigativo.

Otro factor clave considerado fueron los años de experiencia de cada uno de los encuestados, los roles gerenciales tienen un promedio de experiencia entre 15 y 20 años, los roles de mandos medios tienen experiencia entre 10 y 15 años, y los roles más operativos tienen experiencia mayor a 5 años, adicionalmente al ser personal que ha desempeñado su rol en distintos bancos del Ecuador conocen la forma en la que los diferentes bancos del país han trabajado y su evolución, de esta manera se respalda la confiabilidad de los resultados obtenidos.

Respecto a los consultores entrevistados se contó con la gran ayuda y apertura de la empresa Maint S.A, una de las más reconocidas en el país y con 35 años de presencia en el mercado ecuatoriano, se tuvo acceso a un Gerente de proyectos, Arquitecto DevOps, Arquitecto de soluciones de negocios, todos ellos con experiencia mayor a 10 años en proyectos de implementación de la cultura DevOps en empresas a nivel nacional. Adicionalmente, a través de los consultores de Maint S.A. se tuvo acceso a un partner muy reconocido en España que acompaña a Maint S.A en las implementaciones de proyectos DevOps.

3.5 ANÁLISIS DE RESULTADOS

El ciclo de entrega de valor de las áreas de tecnología es uno de los aspectos importantes a ser evaluados; de los resultados obtenidos se evidencia que el 41,7% de los encuestados indican que sus tiempos se encuentran entre 6 y 12 meses.

¿Cuál es el tiempo promedio de entrega de proyectos de desarrollo en su institución bancaria?
12 respuestas

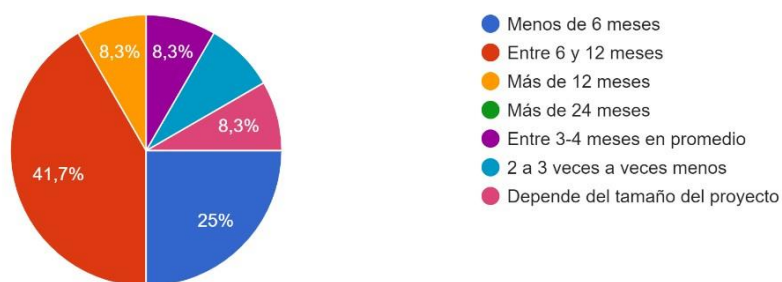


Figura 8 Tiempo promedio de entrega de proyectos de desarrollo

En coherencia con este resultado se observa que el 75% indican que sus procesos de pruebas y despliegue es manual, el 66,7% indican que dependen de terceros en relación con sus departamentos y el 8,3% evidencia que su proceso testing emplea altos tiempos en correcciones de errores, lo que podría estar impactando en el alto tiempo que la mayoría de los bancos indican transcurre entre la concepción de la idea y la entrega de valor a las áreas de negocio.

El proceso de testing de software en su institución bancaria es:

12 respuestas

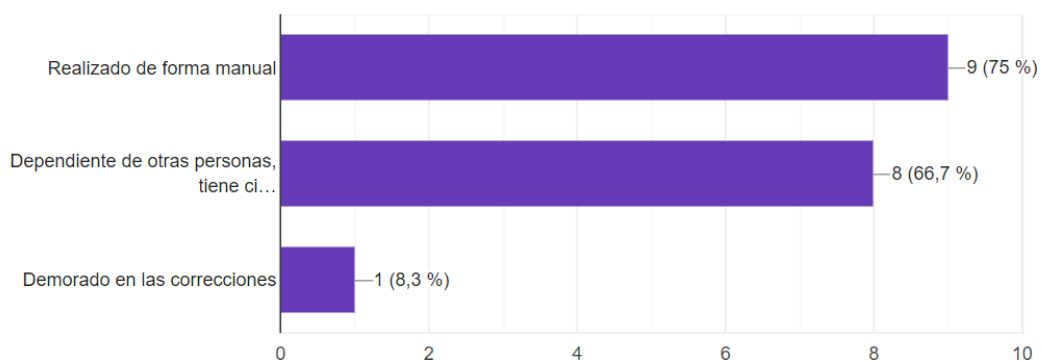


Figura 9 Características del Testing de software

Referente a la competencia una de las preguntas estaba orientada a conocer, comparado con la competencia, si los tiempos de entrega de valor a las áreas de negocio son iguales, mayores o menores; en este contexto se evidenció que la mayoría con un 41,67% desconoce esta información, de los encuestados que conocían esta información se observa que tanto los que dicen que tienen entregar valor al mismo tiempo que su competencia como los que indican que su competencia entrega valor más rápido coinciden con un 25%, finalmente el 8,33% indican que su competencia entrega en mayor tiempo que su institución. Se puede observar que la mayoría de los encuestados considera que no son más ágiles que su competencia, a pesar de contar con metodologías ágiles para el desarrollo de sus productos.

Tabla 1 Tiempos de entrega de productos y servicios tecnológicos

Tiempos de entrega	Porcentaje (%)
Igual tiempo que su institución	25.00%
Mayor tiempo que usted	8.33%
Menos tiempo que su institución	25.00%
No tengo conocimiento	41.67%
Grand Total	100.00%

En cuanto a la evaluación respecto a los objetivos de los departamentos de Desarrollo y Operaciones de TI, el 66,7% indica que los objetivos de ambos departamentos no son compartidos, por lo que se podría entender que puede afectar al objetivo común que debería ser la entrega de valor en el menor tiempo posible.

¿Los objetivos a cumplir de sus departamentos de desarrollo y operaciones de TI son compartidos?
12 respuestas

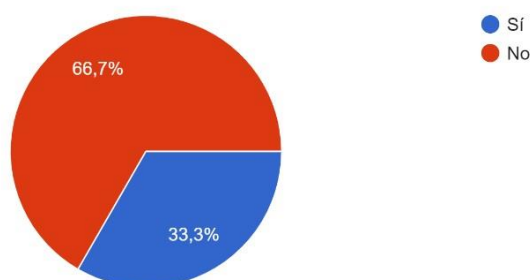


Figura 10 Objetivos de los departamentos de Desarrollo y Operaciones de TI de los bancos

Un aspecto positivo que se evidenció es que el 91,67% de los encuestados indicaron que utilizan metodología o marcos ágiles en sus desarrollos, lo cual representa un avance para la adopción de la cultura DevOps; únicamente una encuesta evidencio que no utilizan metodologías ágiles, solo tradicional.

Tabla 2 Metodologías de Desarrollo Ágiles

Metodologías de desarrollo	Porcentaje (%)	
Extreme programing, Scrum, Tradicional Cascada	8.33%	
Scrum	16.67%	8.33%
Scrum, Agile inception	8.33%	16.67%
Scrum, Agile inception, Tradicional Cascada	8.33%	8.33%
Scrum, Kanban, Tradicional Cascada	16.67%	8.33%
Scrum, Tradicional Cascada	33.33%	16.67%
Tradicional Cascada	8.33%	<u>33.33%</u>
Grand Total	100.00%	<u>91.67%</u>

Por otro lado, el 66,6% indican que utilizan tanto metodologías ágiles como tradicionales (cascada) en sus proyectos.

Tabla 3 Metodologías de desarrollo Tradicionales y Ágiles

Metodologías de desarrollo	Porcentaje (%)	
Extreme programing, Scrum, Tradicional Cascada	8.33%	
Scrum	16.67%	
Scrum, Agile inception	8.33%	8.33%
Scrum, Agile inception, Tradicional Cascada	8.33%	8.33%
Scrum, Kanban, Tradicional Cascada	16.67%	16.67%
Scrum, Tradicional Cascada	33.33%	<u>33.33%</u>
Tradicional Cascada	8.33%	<u>66.67%</u>
Grand Total	100.00%	

Otro aspecto positivo, es que el 100% de los encuestados están familiarizados con la cultura DevOps, aunque solo un 37,5% se encuentra en un proceso de adopción de la misma; con esta información se puede entender que los bancos están buscando la adopción de la cultura DevOps con el objetivo de ser más ágiles.

¿En la institución bancaria en la que usted labora se ha implementado esta cultura o esta en proceso de implementación?

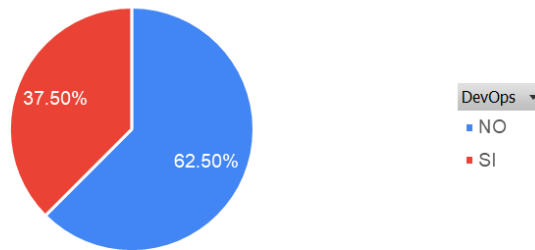


Figura 11 Adopción de cultura DevOps en bancos

Entre las principales razones que consideran los bancos para la adopción de la cultura DevOps, se resalta que en su mayoría el 80% de los encuestados, que ya están implementando esta cultura, optaron por hacerlo debido a que buscan mejorar la velocidad con la que se desarrolla y despliega el software; por otro lado, el 60% lo hizo para cambiar su cultura actual por una más colaborativa, mientras que el 40% buscó DevOps como opción para poder adaptar nuevas tecnologías a su negocio y finalmente un 40% evidencia que el proceso de desarrollo de software afectaba el cumplimiento de las metas planteadas por el negocio. De forma general las instituciones bancarias necesitan mejorar los tiempos de entrega de valor hacia el negocio.

Si respondió sí, ¿qué razón o razones lo llevaron a hacerlo?



10 respuestas

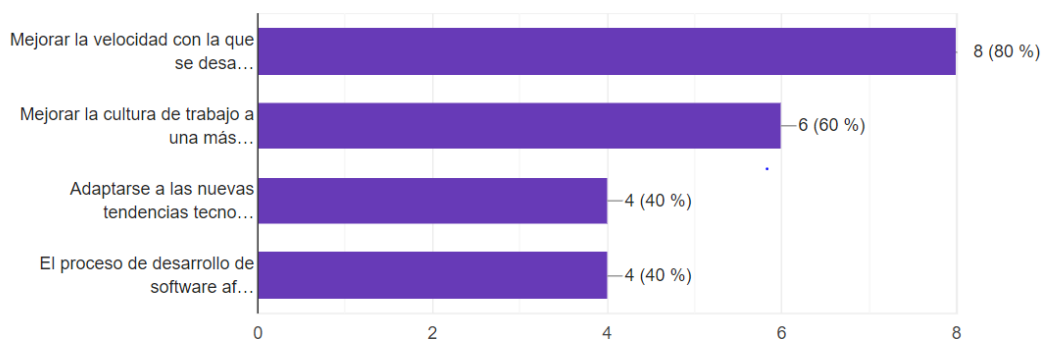


Figura 12 Razones para adoptar DevOps

De los bancos que no han adoptado DevOps, el 50% indican que no cuentan con personal especializado en la cultura y adicional, no cuentan con el tiempo para dedicarse a un proyecto de esta naturaleza, por otra parte el 25% indica que desconoce que es DevOps razón por la cual no ha sido

sugerido como proyecto a ser implementado, finalmente un 25% indica que se encuentra en planeación para la implementación de esta cultura.

Después del análisis de esta información, se observa que la principal razón, radica en que no disponen de personal con conocimiento y experiencia por lo que consideran complicada la adopción de esta cultura.



Figura 13 Limitaciones para adopción de DevOps

Concluyendo con la encuesta, el 92.3% de los encuestados aseguran que contar con una guía aceleraría su decisión de adopción de esta cultura y consideran que les permitiría iniciar por su cuenta su proceso de autoevaluación para iniciar el proceso de adopción de DevOps, siendo la única respuesta negativa la proporcionada por un banco que no ha implementado DevOps e indica que la razón de no haberlo hecho, ha sido el desconocimiento de esta cultura.

A continuación, se prosigue con los resultados de la segunda herramienta de recolección de información empleada.

Cómo resultado de las entrevistas realizadas se obtiene el siguiente análisis:

Todos los consultores entrevistados han formado parte de equipos de proyectos de implementación en varias entidades financieras y de otros ámbitos que han conseguido implantar DevOps con éxito, esta experiencia respalda lo que a continuación se analiza en torno a la cultura DevOps, y el desarrollo de la guía general para el Assessment de autoevaluación objeto de este trabajo.

En base a su amplia experiencia, confirman que DevOps es una metodología que extiende los frameworks de desarrollo ágil a los departamentos de operaciones, eliminando el trabajo en silos lo que hace que se mejore de manera significativa el tiempo de entrega de valor. Combina elementos de distintos frameworks de desarrollo como son XP, Lean Sw Development, Continuous Delivery que lo hacen muy efectivo para la entrega de valor.

Todos coinciden que el inicio del proceso de adopción de esta cultura es el Assessment para la evaluación los procesos de desarrollo y operaciones dentro de la organización y su estructuración, de tal modo que, partiendo del conocimiento cierto del estado actual, se pueda definir a donde se quiere llegar en el mediano o largo plazo, con que aspectos se cuenta para ello y en que se debería realizar los esfuerzos.

Los aspectos que deben ser evaluados dentro del Assesment son los que para DevOps son sus pilares, estos son:

- Tecnología
- Procesos
- Personas

Todos recomiendan que es necesario evaluar el proceso de entrega de valor de la organización, las tecnologías utilizadas para soportar dicho proceso, la estructura organizativa para detectar posibles problemas de comunicación u objetivos independientes que afecten el proceso de entrega continua, incrementando los tiempos en la entrega de los productos desde la concepción de la idea hasta su entrega al negocio; las metodologías o frameworks de agilidad utilizados en el proceso de desarrollo, los perfiles tanto a nivel de soft skills como hard skills, finalmente evaluar su orientación al cambio.

En lo relacionado al objetivo de este trabajo de desarrollo de una guía general, consideran adecuado y de utilidad, para la autoevaluación, algunos indican que debe considerarse que el resultado de esta auto evaluación no será la ruta definitiva, pues este es un proceso continuo, que debe ser medido y evaluado y de requerirse, realizar los ajustes a la ruta que la organización

se plantee como objetivo; así también otra recomendación es que al momento de la implementación se pueda requerir como parte del equipo, roles que actúen objetivamente y puedan llevar a la organización de acuerdo a los hallazgos encontrados y el plan que cada organización considere debe llevar a cabo, si la empresa no tiene este perfil, en la etapa de implementación, debería contar con ayuda externa.

A pesar de no encontrar un consenso total, respecto a los niveles de madurez, la mayoría de los consultores entrevistados, recomiendan que el modelo de CMMI de madurez, es adecuado para poder realizar un plan de ruta o road map para el proceso de adopción de la organización.

De la mano con las diferentes etapas o modelos de madurez en el proceso de adopción de DevOps, es imprescindible el uso de herramientas para lograr un nivel adecuado de automatización y lograr los beneficios esperados.

En el enfoque organizativo, el aspecto fundamental es conseguir establecer un objetivo común que sea transversal al equipo dejando en segundo plano los objetivos parciales de los departamentos. Otro aspecto importante es el empoderamiento de todo el equipo en la entrega de valor. No ser responsable solo de tu parte, sino de todo el proceso.

Finalmente, todos coinciden en la importancia de la medición, aunque hay diferencias entre los indicadores que consideran utilizar, las coincidencias están en los siguientes:

- Tiempo de ciclo
- Frecuencia de despliegue
- Tiempo medio de restauración del servicio
- Porcentaje de fallos en los cambios.

En base al análisis realizado tanto a nivel de las encuestas a los diferentes cargos de los bancos de Guayaquil, y a los consultores, se puede concluir que la elaboración de la guía generará valor y es factible poderla realizar.

CAPÍTULO IV

PROPUESTA METODOLÓGICA

Como todo proyecto requiere un marco referencial a seguir, de acuerdo a la información obtenida con la entrevista a los consultores, y de acuerdo a la documentación investigada respecto a la cultura DevOps, se plantea el siguiente proceso metodológico a seguir:



Figura 14 Modelo propuesta de adopción de cultura DevOps

Tomado de "Entrevista con Maint S.A DevOps", por Jorge Gonzalez, Gerente de Proyectos

Se establecen 5 pasos a seguir para completar de forma exitosa la adopción de DevOps, sin embargo en este proyecto de investigación únicamente se abarca hasta el punto 3 en el que se sugiere un road map general a seguir ya que la ejecución de este no forma parte de los entregables. Como primer punto se tiene la situación inicial, es decir, conocer como se encuentra el banco en relación a los principales aspectos que considera la cultura DevOps que son:

- Gente y cultura
- Procesos y prácticas
- Tencología y automatización

Estos aspectos se conocen como el paquete completo de DevOps.

Las fuentes investigadas para realizar este trabajo son:

- Matriz de madurez por Mediatek Consulting Hub (*Mediatek Consulting Hub*, s. f.)
- Libro *Assessing DevOps For Information Technology, Business and Industry* (Manas Shome & Raghubir Bose, 2018)
- Entrevistas y revisiones con consultores Maint S.A

4.1 GUÍA GENERAL PARA EL LEVANTAMIENTO DE INFORMACIÓN Y OBTENCIÓN DEL NIVEL DE MADUREZ

Para el levantamiento de información se ha desarrollado y se pone a disposición, en el Anexo 3, la matriz de entrevistas que es una guía general que contiene:

- Nivel funcional y/o cargo sugerido al cual se realizará el levantamiento de información
- Objetivos que se buscan cumplir en relación a la información obtenida
- Documentación que se sugiere solicitar al entrevistado
- Mecanismos sugeridos para recolectar la información

Es importante indicar que la persona o grupo de personas que vayan a realizar el levantamiento de información utilicen este formato de matriz para la elaboración de su propia matriz ajustada al banco en el cual se ejecutará; en base a la matriz realizada en detalle se debe gestionar un cronograma de entrevistas considerando fechas y horas para la ejecución de las mismas.

La información que se debe conocer cubre los siguientes aspectos:

- Gente y cultura
 - Comunicación
 - Transparencia
 - Colaboración
 - Actitudes
- Procesos y prácticas
 - Prácticas ágiles
 - Gestión del servicio

- Gestión de la demanda y requerimientos
- Gestión del conocimiento
- Tencología y automatización
 - Control de versiones
 - Ambientes
 - Integración y entrega continua
 - Despliegue continuo
 - Calidad y seguridad del código
 - Pruebas automatizadas
 - Monitoreo

La información que se busca obtener, se describe a continuación, las preguntas específicas por cada aspecto para el levantamiento de la información se encuentran en la matriz de valoración adjunta como Anexo 4.

4.1.1 ASPECTO CULTURA

La cultura se debe entender como la forma de trabajar de todos los equipos, así como la relación interpersonal de toda el área de TI. Este aspecto es de suma importancia ya que determinará el éxito de los cambios tecnológicos y de procesos que requiera realizar la organizaicón durante el proceso de adopción del marco DevOps. Para ello se deben analizar los siguientes puntos:

1. Comunicación

- Evaluar si existe la participación y debate entre todos los involucrados en el desarrollo del software
- Validar la frecuencia con que un miembro del equipo de desarrollo se comunica con un miembro del equipo de operaciones o QA, sin pasar por otros niveles jerárquicos
- Evaluar si la información como métricas, objetivos, requerimientos están a disposición de todas las áreas involucradas
- Evaluar si existen procesos eficientes de difución de la información

2. Transparencia

- Validar que miembros del equipo de calidad y seguridad estén presentes en forma temprana y durante todo el ciclo del desarrollo del software
- Validar que el trabajo realizado por desarrollo sea de conocimiento de operaciones
- Validar que las novedades encontradas por operaciones sean comunicadas a desarrollo

3. Colaboración

- Validar que los miembros de las distintas áreas involucradas en el ciclo de vida del software y operaciones trabajen bajo el mismo objetivo de entrega de valor de la solución en desarrollo

4. Actitudes

- Evaluar si se cuestiona el trabajo de los miembros del equipo de una manera proactiva
- Evaluar si se escuchan las opiniones de todos los miembros del equipo
- Conocer si las evaluaciones son individuales o al equipo
- Evaluar si existe aprendizaje a través de los errores y la mejora continua

4.1.2 ASPECTO PROCESOS Y PRÁCTICAS

En el aspecto de procesos/prácticas se abarca todo el ciclo desarrollo del proyecto, validando su integración con las prácticas ágiles, calidad (Lean) e ITIL (gestión del servicio) con el objetivo de mejorar la eficiencia y el flujo del trabajo.

1. Prácticas Ágiles

Las metodologías ágiles han sido una gran ayuda para mejorar las entregas de software como valor, tanto en tiempos, calidad, costo; muchas empresas han optado por adoptar estos marcos de trabajo en conjunto con metodologías tradicionales y los bancos no son la excepción.

DevOps no es una metodología que busca ser única solución, adoptarla no implica dejar de usar otros marcos con los que la organización ya trabaja y ha tenido buenos resultados, al contrario, DevOps se ve potenciado al apoyarse de metodologías ágiles como Scrum, Lean, Kanban, entre otras y es primordial que un banco ya cuente con metodologías ágiles previo a la adopción de DevOps.

El análisis de esta información debe estar orientado a:

- Identificar si la organización emplea prácticas ágiles
- Identificar si hay personas o equipos experimentadas encargadas de difundir buenas prácticas ágiles al resto de la organización
- Existen herramientas para la automatización del proceso ágil del punto a punto

2. Gestión de la demanda y requerimientos

TI recibe del negocio diferentes requerimientos de soluciones para generar valor al cliente o por cumplimiento de entes de control. Lo más importante de este punto es conocer de qué manera el banco administra los distintos requerimientos de software hacia TI, sean estos productos nuevos, características nuevas de productos en operación, errores reportados, entre otros; es imperativo conocer si el banco posee control sobre este proceso, es decir, validar si la información está a disposición de las personas interesadas, la factibilidad de poder realizar el seguimiento desde el registro hasta el cierre pasando por los diferentes estados y áreas que intervienen en el proceso completo del desarrollo del requerimiento.

Las preguntas a realizar deben estar orientadas a conocer:

- Si los requerimiento/proyectos que llegan a TI son centralizados a través de alguna herramienta o proceso
- La valoración con la que se priorizan los distintos requerimientos que llegan a TI desde el negocio

El análisis de la información levantada debe orientarse principalmente a la identificación de las herramientas en la gestión de la demanda de requerimientos validando si estas están integradas con las herramientas empleadas en el ciclo de vida del software y de ser así, verificar que exista

trazabilidad a través de todo el proceso. Este análisis permite definir el nivel de automatización que existe en esta etapa al levantar las herramientas empleadas, así como su integración con las herramientas del resto de etapas.

3. Gestión del servicio

La mesa de servicios es una parte clave para la cultura DevOps, todo es susceptible a fallos y la tecnología no es la excepción, por ello contar con un equipo de trabajo que sea punto de contacto entre los usuarios del banco y las tecnologías para la recepción de incidencias y requerimientos y su respectiva resolución o seguimiento es fundamental en un banco cuyas actividades no pueden cesarse por un fallo tecnológico.

En este punto es importante identificar el área, departamento o equipo de trabajo que provee a los usuarios del banco la resolución y/o canalización de las necesidades relacionadas al uso de recursos y servicios de plataformas, y productos o servicios que TI presta a las distintas áreas de las instituciones financieras.

El análisis de esta información debe estar orientado a:

- Validar si existen notificaciones automatizadas de incidencias y requerimientos desde monitoreo
- Validar si existe integración de herramientas de mesa de servicios con herramientas de proyectos, de modo que sea posible identificar número de incidencias por proyecto

4. Gestión del conocimiento

Dentro de DevOps es esencial la práctica de integrar y hacer accesible el conocimiento a todas las áreas que intervienen en el proceso de desarrollo, despliegue del software y posterior monitoreo y análisis en la operación. La información debe estar disponible en el lugar y momento adecuado para que la toma de decisiones este soportada en esta; mejorar la eficiencia compartiendo el conocimiento y evitando redescubrir el mismo. En definitiva, compartir ideas, experiencias e información.

El análisis de esta información debe estar orientado a:

- Identificar si se usan tecnologías de software colaborativo

- Identificar si existen bases de conocimiento
- Identificar si existen mentores para la ejecución de las tareas
- Validar que existan diversos conjuntos de habilidades en equipos multifuncionales
- Validar si se comparte el conocimiento y aprendizaje a través reuniones grupales
- Validar si existen herramientas de gestión visual

4.1.3 ASPECTO TECNOLOGÍA Y AUTOMATIZACIÓN

En el aspecto de tecnología es necesario identificar las herramientas usadas en las distintas etapas del desarrollo, despliegue y monitoreo del software.

1. Control de Versiones

La gestión de versiones es un aspecto esencial en la implementación de DevOps, por lo que en la fase de diagnóstico es imprescindible identificar dentro de la organización esta gestión.

La gestión de versiones permite llevar un control de todos los objetos generados durante el proceso de desarrollo del software, como código fuente, archivos de configuración, casos de prueba, entre otros, durante el proceso de desarrollo es muy común los cambios continuos de los objetos, por lo que se requiere saber quién es el responsable de cada cambio, permitiendo el trabajo conjunto e integrado, o inclusive poder realizar un retorno a una versión anterior de dichos objetos.

El análisis de esta información debe estar orientado a:

- Validar si es posible que varias personas puedan trabajar en el mismo producto o componente que se está desarrollando
- Identificar si es factible, regresar a una versión anterior estable en caso de que la liberación de un producto o componente presente fallos
- Acelerar la solución de defectos mediante la rápida identificación del componente causa raíz del problema

2. Ambientes

Los ambientes son esenciales para el ciclo de vida de desarrollo del software, la entrega de software no es una entrega directa a producción una vez terminado el desarrollo, durante el ciclo de desarrollo existen entregas de software a múltiples entornos en los cuales debe pasar una serie de pruebas antes de considerarse listo para un ambiente de producción, lo que se conoce como línea de entrega o cadena de suministros. Para estas pruebas es importante contar con ambientes dedicados que emulen el hardware y el software en el cual el producto se ejecutará una vez implementado.

Algunos ambientes usados en la banca son Dev, Build, QA, SIT, UAT, Producción; la cantidad de ambientes empleados dependerá del negocio.

DevOps propone ciclos de entrega más cortos por ello es requerida la simplificación de la provisión de ambientes a la velocidad requerida por el equipo de desarrollo sin sacrificar la estabilidad requerida por operaciones.

El análisis de esta información debe estar orientado a:

- Verificar si el aprovisionamiento de ambientes se realiza de forma manual o si existe automatización
- Si hay automatización identificar el nivel de esta

3. Integración Continua, Entrega Continua, Pruebas continuas y Despliegue Continuo

DevOps considera imprescindible para lograr una mayor velocidad y calidad del desarrollo y despliegue de software el contar con 4 prácticas importantes:

El desarrollo del software como proyecto es un proceso complejo que involucra diferentes departamentos y personal trabajando diferentes funciones sobre el mismo código, por ello surge la necesidad de la práctica para integrar el trabajo realizado de forma individual a un código raíz (repositorio o rama principal), de esta manera es posible paralelizar distintas tareas y realizar pruebas a medida que nuevo código es integrado, permitiendo la detección y solución de errores en etapas tempranas.

La entrega continua, por otro lado, tiene por objetivo asegurar que el código este siempre listo para ser desplegado a producción de forma segura

y rápida, esto solo se puede lograr a través de pruebas automatizadas realizadas con ejecutables del software en ambientes de preproducción.

Las pruebas continuas consisten en pruebas que se deben realizar en cada etapa del ciclo de desarrollo del software hasta su despliegue, estas pruebas deben llegar a ser automatizadas con el fin de mantener el código en un estado desplegable.

El despliegue continuo es una extensión del concepto de entrega continua, como su nombre lo indica el despliegue continuo realiza el envío a producción de todos los cambios que hayan superado las pruebas automatizadas. Lo deseable es que el despliegue del software a producción sea automatizado pero muchas veces este paso se realiza de forma manual ya que por temas regulatorios se requieren aprobaciones previas, más aún en los bancos en los que se debe ser muy cuidadoso con el software implementado a producción.

El análisis de esta información debe estar orientado a:

- Validar el grado de automatización de los procesos involucrados en el ciclo de vida del desarrollo del software
- Detectar cuellos de botella que generan demoras en el proceso y son fuente probable de errores

4. Calidad y seguridad del código

La búsqueda de DevOps es la entrega de software de una forma rápida, por lo que la seguridad y calidad del software es de relevante importancia en la entrega de valor al negocio y mantener la satisfacción de los clientes internos y externos; para ello es necesario nuevamente la automatización que permita la obtención de informes sobre el cumplimiento de estándares de seguridad y calidad según las definiciones de la organización.

El análisis de esta información debe estar orientado a:

- Verificar que las validaciones de seguridad se realicen desde las etapas más tempranas del ciclo de desarrollo
- Verificar que los objetivos de calidad y seguridad sea de responsabilidad de los equipos de desarrollo y funcionarios de operaciones

- Verificar que las validaciones sean realizadas a lo largo de todo el ciclo de desarrollo

5. Pruebas automatizadas

Dentro de la cultura DevOps es imprescindible la detección de posibles bugs o fallos de seguridad en cualquier paso del ciclo de vida del desarrollo del software y en cualquier ambiente, bajo esta premisa las pruebas automatizadas son de valor ya que reduce tiempos desperdiciados en pruebas manuales, así como detección de errores de forma inmediata; reduciendo significativamente los tiempos empleados en el desarrollo y test del código.

El análisis de esta información debe estar orientado a:

- Determinar el nivel de automatización en las pruebas de código a lo largo de todo el ciclo de vida del desarrollo del software, de existir

6. Monitoreo y retroalimentación

El objetivo del monitoreo es acumular, auditar, trazar y registrar todos los datos posibles de una máquina, sistema, proceso, base de datos o producto, de tal forma que sea posible generar informes que permitan identificar con facilidad posibles errores brindando una retroalimentación del comportamiento del producto final de manera temprana y automatizada.

El análisis de esta información debe estar orientado a:

- Determinar si existe el monitoreo de los productos en producción para prevenir pérdidas del servicio que afecten a la organización y sus clientes

4.1.4 VALORACIÓN Y ANÁLISIS DE MADUREZ

Para obtener el nivel de madurez en el que se encuentra el banco en relación a las mejores prácticas de DevOps, es necesario realizar una valoración cualitativa para lo que se ha desarrollado la matriz de valoración basándose en el modelo de madurez más reconocido a nivel de desarrollo de software, CMMI (Capability Maturity Model Integration). Este fue el modelo recomendado por los consultados entrevistados ya que no existe un modelo de madurez establecido para DevOps específicamente.

El modelo de madurez CMMI considera cinco niveles designado por los números del 1 al 5, estos son:

- Inicial
- Repetible
- Definido
- Medido
- Optimizado

Aplicando el modelo de CMMI a DevOps se han elaborado las siguientes tablas de valoración que son parte de la matriz de valoración que se encuentran adjuntas como Anexo 5. La presentada a continuación corresponde al aspecto cultural.

Tabla 4 Tabla de valoración-aspecto cultural

#	NIVEL	DESCRIPCIÓN
1	Inicial	<ul style="list-style-type: none"> -Organización trabaja en un esquema netamente funcional (silos) -No existe o poco trabajo colaborativo entre las áreas que participan en el ciclo de vida del software -Información solo es compartida entre miembros de las mismas áreas -Herramientas para la comunicación no existen o si existen solo son utilizadas dentro de cada área
2	Repetible	<ul style="list-style-type: none"> -Se establecen procesos de comunicación y colaboración entre áreas -Comunicación limitada entre miembros de Dev y Ops -Existen equipos trabajando bajo metodología ágil, más colaborativo -Se comparte el conocimiento de forma limitada -Existen herramientas estandarizadas para la comunicación y colaboración entre áreas
3	Definido	<ul style="list-style-type: none"> -Dev y Ops se comunican de forma activa y directa sin pasar por una estructura jerárquica (aprobaciones) -Existe un canal de comunicación establecido entre Dev y Ops -Existen equipos ágiles sincronizados que mantienen un ritmo de trabajo similar -Equipos multifuncionales (todos los perfiles requeridos) - Se comparte el conocimiento a través de herramientas entre Dev y Ops -Herramientas de comunicación integradas con otros procesos y herramientas

4	Medido	<ul style="list-style-type: none"> -Comunicación entre Dev y Ops es un proceso maduro (hábito) -Dev y Ops trabajan bajo el mismo fin / prioridades -Existe un proceso para fomentar el aprendizaje compartiendo conocimiento y habilidades entre los miembros de los diferentes equipos
5	Optimizado	<ul style="list-style-type: none"> -Existe un alto nivel de comunicación y colaboración entre los equipos y las áreas del ciclo de vida del software -Herramientas estandarizadas sirviendo a la automatización de los procesos -Mejora continua

La matriz de valoración está compuesta por varias pestañas, cada una correspondiente a los aspectos evaluados, dentro de cada una de ellas se encuentran las preguntas correspondientes a la encuesta; el evaluador deberá registrar la valoración que asigne, en función de su análisis; esta matriz también contiene la pestaña de las 3 tablas de valoración correspondiente a cada aspecto y la pestaña correspondiente a los resultados finales.

Cada elemento que compone los 3 aspectos principales de DevOps se ha agrupado por colores siendo verde los elementos del aspecto cultural, rojo los elementos del aspecto de procesos y prácticas, y azul los elementos que conforman el aspecto de tecnología y automatización. Los pasos para llenar la matriz de valoración se describen a continuación (se ha tomado como ejemplo el aspecto cultural):

1. Ir a cada pestaña del aspecto a evaluar (en este caso el elemento comunicación del aspecto cultural) y colocar una “x” en cada uno de los niveles que han sido cubierto con la pregunta valorada. Si la pregunta se ubica en un nivel 2 se deberá coloca una “x” tanto en el nivel 1 y 2 como se presenta en el siguiente ejemplo:

Tabla 5 Calificación elemento transparencia
















Transparencia							
#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Se involucran a miembros de áreas de seguridad y calidad del software desde las fases tempranas en el ciclo de desarrollo del software?	3	x	x	x		
2	¿El área de desarrollo comunica el trabajo que está realizando al área de operaciones?	1	x				
3	¿El área de operaciones comunica las novedades encontradas al área de desarrollo?	2	x	x			
4	¿Existe una herramienta donde queda registrada la información transmitida entre estas áreas?	4	x	x	x	x	




Resultado	2
------------------	----------

2. La columna valor calcula en base a fórmula el nivel máximo asignado por pregunta, es decir, como muestra la gráfica, para la primera pregunta el valor resultante es 2, ya que está marcado el nivel 1 y nivel 2; la pregunta 2 tiene un valor de 3, contabilizando las marcas de los niveles 1, 2 y 3; la pregunta 3 tiene un valor de 4, contabilizando las marcas en los niveles 1, 2, 3, y 4, finalmente la pregunta 4, tiene un valor de 1 ya que solo ha sido marcado el nivel 1.
3. El valor final de dicho aspecto, en este caso comunicación, consiste en el promedio de los niveles de madurez de cada pregunta, este promedio ha sido redondeado al entero menor, es decir, si el promedio da como resultado 2.5 el nivel de madurez final será 2 ya que como indica CMMI, para pasar a un siguiente nivel de madurez se debe cumplir con todo lo especificado en el nivel previo.
4. Se debe repetir el proceso de valoración para cada elemento de los 3 aspectos a evaluar.

Una vez valorados todos los elementos, la pestaña “RESULTADOS” mostrará el nivel de madurez por cada elemento y el nivel de madurez de cada uno de los 3 aspectos. El resultado de madurez de cada aspecto se obtiene al calcular el promedio de cada elemento que compone el aspecto, por ejemplo: para el aspecto cultural, se calcula el promedio de los niveles de madurez de los elementos comunicación, transparencia, colaboración y actitudes.

Tabla 6 Resultado Assessment

Aspecto	Elemento Evaluado	Estado As Is	Estado To Be
Cultura	Comunicación	 1	5
Cultura	Transparencia	 2	5
Cultura	Colaboración	 1	5
Cultura	Actitudes	 1	5
Procesos	Prácticas ágiles	 2	5
Procesos	Gestion de la demanda y requerimientos	 1	5
Procesos	Gestión del Servicio	 1	5
Procesos	Gestión de Conocimiento	 1	5
Tecnológico	Control de Versiones	 1	5
Tecnológico	Ambientes	 1	5
Tecnológico	Integración Continua y Entrega Continua	 1	5
Tecnológico	Calidad y Seguridad de Código	 1	5
Tecnológico	Testing Automatizado	 1	5
Tecnológico	Monitoreo	 1	5
Tecnológico	Despliegue Continuo	 1	5

Cultura		2
Procesos		1
Tecnología		1

En base al estudio de la teoría de DevOps y con las entrevistas realizadas se ha generado la siguiente matriz de madurez que abarca los aspectos del paquete completo de DevOps:

Tabla 7 Matriz de niveles de madurez DevOps

Nivel	Cultura Organizacional	Gente(equipo)	Prácticas y Procesos	Construcción e Integración continua	Entrega continua/Ambientes	Testing
5	-Cultura de colaboración perfectamente adoptada -Dev y Ops comparten metas y prioridades -Mejora Continua	-Mejora continua -Validación de niveles de satisfacción de los equipos	-Mejora Continua -Mejora de tiempos de time to market -Mejora de tiempos de servicio	-Mejora continua del proceso -Rápida retroalimentación	-Mejora continua -Análisis continuo del tiempo de generación de ambientes y No. de defectos	-Test totalmente automatizado -Número de errores tendientes a cero
4	-Existen hábitos de comunicación y colaboración (cultura DevOps) -Formación de comunidades de práctica -Todas las áreas de TI trabajan en equipos con enfoque en la entrega de valor al negocio	-Colaboración de equipos -Equipos son autónomos y empoderados	-Creación de varias cadenas de valor -Base del conocimiento implementada -Generación automática de incidencias en la mesa de servicio -Trazabilidad establecida -Monitorización integral de aplicaciones/servicios -Recolección de información para Kpi's -Procesos estandarizados	-Métricas relacionadas a la construcción del código -Lecciones aprendidas -Procesos mejorados -Integración de pipelines	-Despliegue automático de todo lo liberado -Entornos generados y aprovisionados automáticamente -Cero tiempo de espera para el deployment	-Métricas de calidad establecidas
3	-Eliminación de silos	-Varios equipos -Comunicación integrada -Se comparte conocimiento a través de herramientas	-Equipos sincronizados, ritmo de trabajo similar, organizados como cadena de valor -Se registra información como datos organizados o estructurados -Integración de la mesa de servicio con otras herramientas de gestión de proyectos y requerimientos Definición de Kpi's	-Integración continua -Test integrado, cada código terminado es testeado y agregado al código raíz -Existen prácticas de revisión de código para calidad y seguridad -Creación de pipelines	-Entrega automática de código en sprint cortos pero más frecuentes -Se extiende el aprovisionamiento automático a mas entornos	-Testing automatizado
2	-Inicio del cambio cultural Cultura ágil (colaborativa)	-Colaboración entre Dev y Ops más ágil -Se utilizan herramientas para la comunicación y colaboración	-Equipos independientes haciendo uso de prácticas ágiles -Base de conocimiento al menos con registro de datos como observaciones, hechos -Mesa de servicios con buenas practicas -Existe una mínima trazabilidad -Gestión de la demanda centralizada, herramientas para la gestión de proyectos.	-Existe configuración del software -Control de versiones y generación de branches para desarrollo en paralelo	-Proceso definido para la entrega automática en ciertos entornos -Herramienta para gestión de entregables	-Testing unitario automatizado -Testing integrado en equipos (no hay fase de test)
1	-Existen Silos -Herramientas no estandarizadas -Cultura no colaborativa: objetivos, prioridades y métricas distintas entre Dev y Ops -Se premia o evalúa negativamente el trabajo individual más que el grupal	-Comunicación limitada dentro de cada área -No se comparte el conocimiento -No se trabaja en equipo	-Gestión de proyectos tradicional, (sin prácticas ágiles) -No existe base de conocimiento implementada a nivel de TI -Existe una mesa de servicio con funciones básicas -No existe un proceso claro o centralizado de la gestión de la demanda y/o requerimientos	-Integración de código manual -No existe herramienta de control de versiones -No se aplican las prácticas de seguridad y Calidad del código dentro del ciclo de desarrollo	-No existen entregas continuas de software -Dependiente de disponibilidad de entornos	-Pruebas manuales después de desarrollo -Requiere personal específico y tiempos para pruebas

Con la valoración obtenida, utilizando un esquema de semáforo se puede documentar el estado del banco evaluado utilizando el color rojo para los niveles no alcanzados, amarillo para los alcanzados en cierto nivel, pero no en su totalidad y verde aquellos que se ha cubierto en su totalidad.

Siguiendo el ejemplo anterior la matriz de madurez documentada quedaría de la siguiente manera:

Tabla 8 Matriz de madurez con esquema de semáforo

Nivel	Cultura Organizacional	Gente(equipo)	Prácticas y Procesos	Construcción e Integración continua	Entrega continua/Ambientes	Testing
5	-Cultura de colaboración perfectamente adoptada -Dev y Ops comparten metas y prioridades -Mejora Continua	-Mejora continua -Validación de niveles de satisfacción de los equipos	-Mejora Continua -Mejora de tiempos de time to market -Mejora de tiempos de servicio	-Mejora continua del proceso -Rápida retroalimentación	-Mejora continua -Análisis continuo del tiempo de generación de ambientes y No. de defectos	-Test totalmente automatizado -Número de errores tendientes a cero
4	-Existen hábitos de comunicación y colaboración (cultura DevOps) -Formación de comunidades de práctica -Todas las áreas de TI trabajan en equipos con enfoque en la entrega de valor al negocio	-Colaboración de equipos -Equipos son autónomos y empoderados	-Creación de varias cadenas de valor -Base del conocimiento implementada -Generación automática de incidencias en la mesa de servicio -Trazabilidad establecida -Monitorización integral de aplicaciones/servicios -Recolección de información para Kpi's -Procesos estandarizados	-Métricas relacionadas a la construcción del código -Lecciones aprendidas -Procesos mejorados -Integración de pipelines	-Despliegue automático de todo lo liberado -Entornos generados y aprovisionados automáticamente -Cero tiempo de espera para el deployment	-Métricas de calidad establecidas
3	-Eliminación de silos	-Varios equipos multifuncionales por cadena de valor -Comunicación integrada -Se comparte conocimiento a través de herramientas	-Equipos sincronizados, ritmo de trabajo similar, organizados como cadena de valor -Se registra información como datos organizados o estructurados -Integración de la mesa de servicio con otras herramientas de gestión de proyectos y requerimientos Definición de Kpi's	-Integración continua -Test integrado, cada código terminado es testeado y aareado al código raíz -Existen prácticas de revisión de código para calidad y seguridad -Creación de pipelines	-Entrega automática de código en sprint cortos pero más frecuentes -Se extiende el aprovisionamiento automático a mas entornos	-Testing automatizado
2	-Inicio del cambio cultural Cultura ágil (colaborativa)	-Colaboración entre Dev y Ops más ágil -Se utilizan herramientas para la comunicación y colaboración	-Equipos independientes haciendo uso de prácticas ágiles -Base de conocimiento al menos con registro de datos como observaciones, hechos -Mesa de servicios con buenas practicas -Existe una mínima trazabilidad -Gestión de la demanda centralizada, herramientas para la gestión de proyectos.	-Existe configuración del software -Control de versiones y generación de branches para desarrollo en paralelo	-Proceso definido para la entrega automática en ciertos entornos -Herramienta para gestión de entregables	-Testing unitario automatizado -Testing integrado en equipos (no hay fase de test)
1	-Existen Silos -Herramientas no estandarizadas -Cultura no colaborativa: objetivos, prioridades y métricas distintas entre Dev y Ops -Se premia o evalúa negativamente el trabajo individual más que el grupal	-Comunicación limitada dentro de cada área -No se comparte el conocimiento -No se trabaja en equipo	-Gestión de proyectos tradicional, (sin prácticas ágiles) -No existe base de conocimiento implementada a nivel de TI -Existe una mesa de servicio con funciones básicas -No existe un proceso claro o centralizado de la gestión de la demanda v/o requerimientos	-Integración de código manual -No existe herramienta de control de versiones -No se aplican las prácticas de seguridad y Calidad del código dentro del ciclo de desarrollo	-No existen entregas continuas de software -Dependiente de disponibilidad de entornos	-Pruebas manuales después de desarrollo -Requiere personal específico y tiempos para pruebas

4.1.5 ANÁLISIS DE APLICACIONES

El análisis de aplicaciones que soportan el servicio de TI a clientes internos y externos es necesario para identificar las acciones y el esfuerzo que la organización debe realizar para iniciar su proceso de adopción de la cultura DevOps. Reconocer qué aplicaciones son prioridad para ser consideradas bajo el marco DevOps y cuáles deben permanecer sin ser alteradas es el esfuerzo que se busca en este análisis.

El análisis de esta información debe estar orientado a:

- Identificar las tecnologías en las que están desarrolladas las aplicaciones
- Identificar el tamaño de la aplicación
- Identificar la complejidad de la arquitectura
- Identificar el valor que entrega al negocio
- Identificar las aplicaciones consideradas para ser migradas

A continuación, se incluye una tabla en la cual se puede documentar el inventario de aplicaciones del banco con la información de interés. Una vez completado es necesario realizar un análisis para establecer las prioridades de las aplicaciones a ser usadas como piloto con la metodología DevOps.

Tabla 9 Inventario de aplicaciones

Descripción de la Aplicación	Tecnología	Complejidad en su arquitectura de explotación	Tamaño de la aplicación	No. De Servicios	Valor Funcional que entrega al negocio	Existe plan de migración

El objetivo de este análisis es identificar y definir el aplicativo que será considerado como piloto para la adopción de la cultura DevOps, así como también la prioridad de las aplicaciones que pudieran ir adoptando el modelo DevOps de acuerdo con el avance del proceso de madurez del banco.

4.2 ROAD MAP GENERAL

Una vez que se ha realizado el levantamiento de información, análisis de la misma y se ha obtenido el resultado del nivel de madurez en el que se encuentra la organización en relación a las mejores prácticas de la cultura DevOps, se puede tener como referencia el siguiente modelo de fases, cabe recalcar que este siempre será solo una guía, cada organización en base a los conceptos y mejores prácticas de DevOps y el conocimiento profundo de su organización y sus capacidades en relación a recursos materiales y de gente, podrá definir su propio mapa ruta para la adopción de esta cultura.

En relación con el modelo de madurez CMMI que ha sido utilizado, se propone cinco fases. En cada una de ellas se seguirán los enfoques de la matriz de madurez:

- **Enfoque Cultural:** Se establece el cambio en el comportamiento de los equipos de trabajo y los objetivos.
- **Enfoque Procesos:** Se especifican las buenas prácticas que deben ser adoptadas.
- **Enfoque Tecnológico:** Las diferentes herramientas y tecnologías que apoyarán la implementación de los enfoques de cultura y procesos.

4.2.1 FASE 1

En función de las experiencias de los consultores entrevistados, y considerando que una empresa requiere cambiar su forma tradicional de trabajar, es decir su cultura, se considera una de las fases más importantes por lo que su duración se estima entre 12 y 18 meses dependiendo de los recursos que la institución requiera asignar para el proceso.

4.2.1.1 Enfoque cultural - organizativo

Como se ha mencionado en los capítulos anteriores, uno de los principales factores para tener en cuenta en una organización a la hora de adoptar el modelo DevOps es el cambio cultural, este es la base que permite realizar los cambios tecnológicos y funcionales que la organización debe realizar en las diferentes fases del proceso de adopción. Se propone centrar

los esfuerzos del cambio cultural inicial en los pilares básicos de la comunicación y la colaboración, de tal manera que se genere un ambiente de trabajo en equipo, se buscan las oportunidades de mejora en la revisión y mejora continua de los procesos.

Como lo indica el modelo de madurez CMMI, deben existir métricas de tal manera que se soporte la toma de decisiones hasta lograr tener los procesos y herramientas perfectamente implementados, ejecutados y se aplique la mejora continua, para ello desde que existen procesos definidos se deben emplear métricas.

En la Fase 1 significa que la organización en el aspecto organizativo no tiene implementada agilidad, por lo que requiere iniciar su adopción. En este punto, en base a la investigación realizada con los consultores, se sugiere que la organización cree un grupo pionero, conformado por la siguiente estructura básica:

- Una unidad organizacional o grupo que sea el pionero en la implementación de la agilidad, dentro de este centro o unidad debería haber un rol de Agile Coach, que sea el que socialice el framework ágil escogido (SCRUM, KANBAN, LEAN, etc.) y realice el acompañamiento a los diferentes equipos ágiles que se vayan formando en la organización
- Un equipo ágil, siguiendo las recomendaciones, cada equipo deberá estar conformado de 4 a 9 personas e integrarán todos los perfiles necesarios para el producto que desarrollarán

4.2.1.2 Enfoque procesos y prácticas

- El equipo debe de tener acceso a información que le permita conocer como están respondiendo los productos, es decir, debe haber comunicación básica establecida entre Dev y Ops
- Definir un proceso para la Gestión del conocimiento, donde se inicie el registro de observaciones o hechos, por ejemplo, experiencias positivas o negativas que experimenta el equipo de agilidad

- Establecer procedimientos para entregas continuas iniciando con entornos piloto, pruebas unitarias automatizadas, integrar el test dentro del equipo de desarrollo
- Promover a la mejora de la colaboración entre equipos a través del uso de herramientas compartidas como Teams, Slack o grupos de WhatsApp
- Definir el proceso para gestionar configuración del software
- Implementar el proceso de control de versiones, para todo el código fuente incluido el código PLSQL si lo hubiere, proceso debe incluir la generación de branches para viabilizar el desarrollo en paralelo
- Implementación de una herramienta para la gestión de los proyectos ágiles, de tal forma que se pueda realizar el seguimiento de los mismos y asegurar la transparencia
- Mejorar la gestión del servicio o mesa del servicio, alineando los procesos a las mejores prácticas
- Definir un proceso para el manejo centralizado de la gestión de la demanda
- Empezar a tener trazabilidad, como por ejemplo, a través de la herramienta de gestión de proyectos ágiles

4.2.1.3 Enfoque tecnológico

- Herramienta de la gestión del servicio (mesa de servicio)
- Herramienta de comunicación, que permita la comunicación continua con cualquier persona de la organización, sin pasar por la creación de tickets, correos electrónicos o llamadas telefónicas
- Herramienta para la gestión del conocimiento, para centralizar documentación, manuales, requerimientos, y cualquier detalle que se considere apropiado compartir con el resto de las personas de un equipo, o de toda la organización
- Herramienta para la gestión de la demanda
- Herramienta de gestión de proyectos ágiles

4.2.2 FASE 2

En la fase 2 la institución ha iniciado el cambio y con ello se han definido procesos y herramientas que deben tender a ser usados por toda el área de TI, para alcanzar el estado definido; esta fase es muy importante en el proceso de adopción y también requiere alto esfuerzo por lo que se estima un tiempo de duración entre 8 y 12 meses.

4.2.2.1 Enfoque cultural - organizativo

En esta fase se busca la estandarización y normalización de los procesos y herramientas que la organización tiene, tendiente siempre a lograr la colaboración y comunicación de punto a punto, para lograr el trabajo y comunicación integrados entre todas los equipos y áreas o departamentos que intervienen en el ciclo de vida del software, compartiendo las mismas prioridades y metas y siendo medidos por un mismo objetivo que es la entrega continua del valor hacia el negocio.

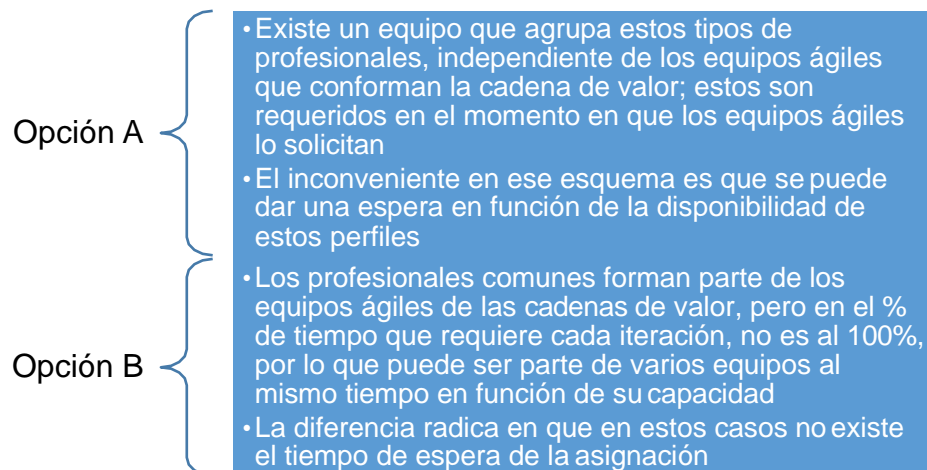
Al haber adoptado ya los frameworks ágiles seleccionados, durante los procesos de ciclo de vida de desarrollo del software, se producen entregas pequeñas, pero con mayor frecuencia, lo que permite evaluar el valor que se está entregando al negocio de forma temprana.

Los equipos que practican agilidad normalmente ven resultados que los motiva, y contagian este positivismo al resto de equipos de la organización, de tal modo que se facilita la adopción completa del nuevo modelo DevOps.

En esta fase, la organización en el aspecto organizativo realiza prácticas de agilidad con equipos aislados. En este punto y en base a las entrevistas realizadas con los consultores dentro del capítulo 3, se sugiere que la organización cree un grupo pionero, conformado por la siguiente estructura básica:

- Primera Cadena de Valor compuesta por más de un equipo ágil
- Equipos organizados según el esquema de roles y capacidades que corresponda

- En el caso de los recursos que brindan ámbitos profesionales comunes como: especialistas en base de datos, arquitectos, especialistas en seguridad de código, especialista en seguridad, etc; se sugieren dos opciones:



- Un Líder de Cadena de Valor / Product Manager responsable de todos los equipos a su cargo, que promueva la comunicación y la colaboración no solo entre los equipos de su Cadena de Valor

Adicionalmente los consultores recomiendan disponer de los siguientes apoyos sin que estos sean parte de los equipos que conforman la cadena de valor:

- Un evangelizador de Calidad, que asesore y transmita al equipo las buenas prácticas necesarias para garantizar una integración total de las especialidades de su departamento dentro del ciclo de vida del desarrollo de un producto
- El mismo caso para un Evangelizador de Seguridad
- Por último, se repite lo mismo para un Evangelizador de Cumplimiento Normativo
- Si se detecta algún otro tipo de Evangelizador que se desee incorporar al esquema organizativo (aspectos legales, contabilidad, arquitectura, etc) puede agregarse

4.2.2.2 Enfoque procesos y prácticas

En esta fase el objetivo es incrementar las funcionalidades de los procesos y herramientas ya existentes e incluir nuevas prácticas que lleven al siguiente nivel de madurez de la organización en la adopción de DevOps:

- Disponibilidad del código fuente para cualquier miembro del equipo (ya sea Desarrollo u Operaciones), pero también para otros equipos de distintas áreas, de acuerdo con los correspondientes roles y permisos asignados; esto incrementa la comunicación integrada, colaboración y compartición del conocimiento
- Estandarización de sistemas operativos. Cuanta más variedad exista, más complejo, difícil y costoso se convierte la gestión de TI
- Integrar el proceso de configuración del software con la herramienta de control de versiones
- Las pruebas pasan a ser automáticas, dentro de lo posible. Con el apoyo de los evangelizadores de calidad, se pueden crear tests automatizados de todo tipo
- Implementar el proceso de QA o calidad para garantizar que el código creado este bien escrito, refactorizado, documentado y bien estructurado, para garantizar el futuro mantenimiento
- Implementar el proceso de código seguro, el o los evangelizadores de Seguridad deben definir el proceso que garantice que la arquitectura de las aplicaciones como el código que se va escribiendo debe implementar de forma implícita la seguridad
- Se implementa integración continua. Una vez que la gran mayoría de requisitos se han cumplido hasta esta fase, la integración continua debe ser implementada
- Se utilizan sprints cortos (small batches) para aumentar el número de entregas y reducir el tiempo entre las mismas
- Se debe mejorar el proceso de gestión del conocimiento, de tal manera que los datos ingresados sean estructurados
- Se deberá iniciar con la definición de Kpi's que permitan medir el grado de adopción del modelo DevOps en el banco, esos indicadores deben estar orientados a la obtención de objetivos del negocio para el banco.

Los consultores encuestados sugieren los siguientes Kpi's más usados en el mundo DevOps::

- Tiempo de entrega: tiempo que transcurre desde que llega un requerimiento desde el área de negocio hasta que este requerimiento es puesto en producción cumpliendo con la entrega de valor requerida
- Frecuencia de despliegue: número de veces que se pone software en producción en un periodo de tiempo
- Porcentaje de fallos en cambios: porcentaje de afectación que tuvo el servicio, ya sea fallo total o degradación del servicio, ocasionado por cambio(s)
- Tiempo medio para restauración del servicio: promedio del tiempo que toma reestablecer el servicio a su proceso normal
- Iniciar con la creación de Pipelines para la automatización de las fases del desarrollo de software

4.2.2.3 Enfoque tecnológico

- Herramienta de integración continua, que permita realizar el despliegue correspondiente a los distintos ambientes
- Herramientas para pruebas automatizadas durante el ciclo de vida de desarrollo de software
- Herramienta de revisión de código estático y dinámico para seguridad y calidad
- Herramienta para la gestión de entregables, que permita almacenar y gestionar los desplegados a los distintos ambientes de la organización
- Integrar la mesa de servicios con las herramientas implementadas de Gestión de proyectos
- Herramientas para creación de Pipelines

4.2.3 FASE 3

En esta fase el Banco ha alcanzado un nivel de madurez DevOps avanzado, por lo que se requiere establecer puntos de control. Si los resultados no son los esperados, se deberá revisar las causas raíz mediante un nuevo ciclo de assessment para aplicar las medidas correctivas.

En esta fase los procesos están definidos y las herramientas implementadas, por lo que los esfuerzos para la automatización se estiman entre 6 y 12 meses.

4.2.3.1 Enfoque cultural - organizativo

En esta fase se continuará con la implementación de las mejores prácticas para destruir por completo los silos y muros de confusión, la organización adopta las prácticas DevOps como hábitos y todas las áreas que intervienen en el ciclo de vida del software se enfocan en la generación de valor para el negocio, se continúa trabajando en el cambio cultural positivo y de la integración de todos los elementos del ciclo de vida del desarrollo del software.

Se puede continuar con los siguientes cambios incrementales:

- Creación de nuevas Cadenas de Valor, con igual estructura que la creada en la fase anterior
- Expansión en el número de equipos en prácticas ágiles
- Iniciar con la creación de la Comunidad de Práctica (CoP), esta debe estar compuesta por un conjunto de profesionales de la organización de todos los equipos existentes, el objetivo de esta es ser un espacio de interacción para compartir conocimientos, experiencias, resolver posibles problemas, es decir seguir aprendiendo en función de la compartición del conocimiento

4.2.3.2 Enfoque procesos y prácticas

Se continúan incrementando nuevas prácticas y mejorando las existentes:

- En la fase anterior se recomendó la estandarización de sistemas operativos, en esta fase, aunque se sabe es complejo, debería tenderse a tener un sistema operativo estándar, de tal modo que los despliegues sean en un único sistema operativo, si no es factible por lo menos se debe tender a tener al menos una familia de sistemas operativos. El objetivo es reducir la complejidad al máximo, para el tiempo que se ahorre de esta eliminación, dedicarla a continuar con la automatización e innovación

- Implementar la utilización de archivos de configuración para los distintos sistemas y ambientes de los productos, y almacenarlos utilizando el control de versiones
- Analizar la implementación del despliegue continuo si bien es uno de los objetivos a alcanzar dentro del modelo de DevOps, la entidad bancaria depende de importantes organismos de control por lo que esto quizá sea una práctica no conveniente de adoptar
- Estandarizar la gestión de las bases de datos
- En desarrollo implementar la creación de funciones para crear código modular reutilizable
- Extender la automatización del aprovisionamiento a todos los entornos

4.2.3.3 Enfoque tecnológico

- Implementar la gestión de ambientes
- Implementar la automatización de bases de datos, es decir, realizar la creación de las bases de datos utilizando archivos de código fuente versionables, esto para alcanzar la automatización de despliegues y cambios en las arquitecturas de los productos
- Incrementar la entrega continua

4.2.4 FASE 4

En esta fase los esfuerzos están dirigidos a la medición de los procesos para optimizarlos, se estima una duración entre 6 y 8 meses.

4.2.4.1 Enfoque cultural - organizativo

En esta fase los equipos de Dev y Ops deben estar trabajando completamente integrados y con el objetivo común de entregar valor al negocio. Los equipos ya formados y con empoderamiento, están generando buenos resultados, ya no debe existir cuellos de botella por esperas respecto a tomas de decisiones fuera del equipo o por espera de tareas de otras áreas.

Como resultado de las implementaciones realizadas, los equipos deben haber ganado velocidad en la ejecución, por lo que se hace factible integrar sostenidamente a este modelo a los integrantes que la organización requiera.

- Integrar los equipos según los requerimientos de las Cadenas de Valor existentes
- Crecimiento de los líderes de Cadena de Valor / Product Manager, a líderes de agilidad
- Crecer en el número de Evangelizadores de las distintas áreas de trabajo para cubrir la demanda de la organización
- Acelerar la difusión de DevOps al resto de la organización mediante la combinación de equipos, es decir, rotar los integrantes entre los distintos equipos de tal modo que se conformen con personal experimentado y con nuevos integrantes, con el objetivo de acelerar o bajar la curva de aprendizaje de los recursos menos expertos.

4.2.4.2 Enfoque procesos y prácticas

Se continúa profundizando con nuevas prácticas:

- Asegurar la alta disponibilidad, por defecto, en todos los productos y aplicaciones
- Analizar la adición de KPI's que permitan la continua optimización de los procesos implementados

4.2.4.3 Enfoque tecnológico

Evaluar en lo posible la utilización de la siguiente tecnología:

- Herramienta de computación en la nube, tanto públicas como privadas
- Herramienta de contenedores de aplicaciones y microservicios
- Herramienta de orquestación
- Herramienta de despliegue continuo

La migración de productos legacy o antiguos a las tecnologías y herramientas propuestas no siempre es factible y en el caso de serlo puede resultar en un alto costo.

Al momento de definir la adquisición de herramientas para su implementación, se recomienda considerar las distintas alternativas que existen en el mercado como:

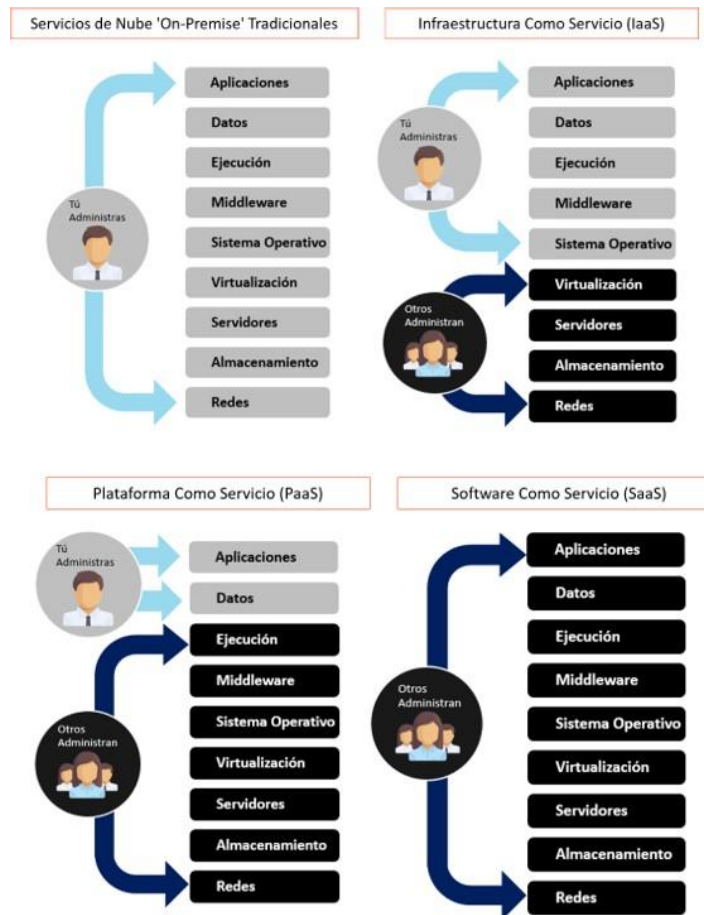


Figura 15 Modelos de servicio de cómputo en la nube

Tomado de "Fundamentos de DevOps", por PeopleCert, 2019

4.2.5 FASE 5

Esta fase corresponde a la mejora continua y se vuelve parte de la operación del área de TI de la institución.

4.2.5.1 Enfoque cultural - organizativo

Según el modelo de madurez CMMI que es en el que basa este assessment y este capítulo, esta fase se centra en la mejora continua. Todo debe ser analizado, ya sea mediante retrospectivas, KPIs, o métricas para la continua evaluación de los procesos y poder realizar las acciones para la mejora continua.

Escenarios que se pueden plantear para análisis son los siguientes:

- Optimización de los equipos de trabajo ágiles

- Potenciar los perfiles multifuncionales
- Incrementar la estandarización de herramientas
- Evaluación de las estrategias definidas para incrementar la velocidad de aprendizaje
- Velocidad de entrega de valor

Se debe partir del concepto que todo es factible de mejora, existen varias herramientas que ayudan a la mejora continua, el modelo PDCA (plan-check-do-act) es uno de los más conocidos según las entrevistas, la organización podrá utilizar esta herramienta o cualquier otra que le permita cumplir el objetivo.

4.2.5.2 Enfoque procesos y prácticas

En este punto no debe interpretarse que absolutamente todo TI debe estar bajo el modelo de DevOps, ciertas aplicaciones por tecnología o por otros aspectos relevantes para la organización deberán seguir bajo sus metodologías tradicionales, sin embargo, siempre se benefician de los cambios culturales y tecnológicos que ha realizado el Banco.

En un nivel de madurez 5 se entiende que todo aquello que ha podido ser estandarizado/automatizado ha sido cubierto, sin embargo, como la propia filosofía DevOps lo indica: “todo es susceptible de ser automatizado” (PeopleCert, 2019), por ello se deben encaminar los esfuerzos a mediciones y reconocer oportunidades de mejora y automatización de los procesos y/o herramientas que califiquen para estas mejoras.

4.2.5.3 Enfoque tecnológico

En función de las oportunidades de mejora identificadas la organización deberá evaluar las distintas herramientas que se consideren puedan automatizar los procesos que califiquen para el cambio.

4.3 CLASIFICACIÓN DE HERRAMIENTAS DEVOPS

Como se mencionó en el punto 2.2.5 del Marco Teórico, DevOps define su propio ciclo de vida en relación con la automatización del ciclo del desarrollo

del software desde su concepción como idea, hasta su operación en producción. Las etapas del ciclo establecidas se resumen en el gráfico:



Figura 16 Ciclo de vida DevOps

Tomado de "Herramientas DevOps", por Jaivic Villegas, 2019

En base a este ciclo de vida y tomando en cuenta los niveles de madurez sugeridos en el roadmap general, se ha realizado una investigación y recopilación de las principales herramientas recomendadas. La selección de estas herramientas se realizó tomando en consideración las siguientes fuentes que califican herramientas de software, como son:

- Gartner peer insights
- Software Advice
- Tabla periódica de herramientas DevOps 2020 creada por profesionales certificados DevOps (practitioners) (digital.ai, 2020)
- Libros

La matriz está compuesta por los siguientes campos:

- Etapas del ciclo de vida DevOps: Indica la etapa propia del ciclo DevOps a la que pertenecen las herramientas descritas, así como también el nivel de madurez en el que debe ser implementada la herramienta
- Descripción de la etapa: una breve descripción del propósito que deben cumplir las herramientas en la etapa especificada
- Nombre de herramienta
- Costos disponibles con corte agosto 2020
- Evaluación por Gartner peer insights
- Evaluación por Software Advice
- Descripción de herramienta

- Características más destacables de la herramienta

Tabla 10 Matriz de herramientas recomendadas

Etapa	Detalle	Herramienta	Costos ¹	Evaluación Gartner peer insights ²	Evaluación Software Advice ³	Descripción	Características
Plan	Punto de partida para crear software, se definen responsables y estiman tiempo. Debe ser posible gestionar la planificación de un proyecto de software. Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 2	Jira	-Versión gratuita -Standard \$7 por usuario/mes -Premium \$14 por usuario/mes	4.3/5	4.36/5	Herramienta web configurable en metodología de software ágil enfocada en el seguimiento de tareas e identificación de errores	-Configuración y seguridad por roles y permisos de usuario -Registro de historias de usuario -Sistema de notificaciones -Permite la compartición y manipulación de archivos
		Trello	-Versión gratuita -Business Class \$9.99 por usuario/mes -Enterprise \$17.50 por usuario/mes	sin calificar	4.51/5	Herramienta web con versión gratuita, orientada a concepto Kanban simple y visual	-Configurable en 20 idiomas diferentes -Versión gratuita sin límite de usuarios -Integración con herramientas como GitHub, Box, Dropbox, Slack, Harvest y Toggle -Variedad de plugins gratis y pagados -Integración con Jira
		Asana	-Basic -Premium \$10.99 por usuario/mes -Business \$24.99 por usuario/mes -Enterprise cotización con vendedores	4.5/5	4.42/5	Herramienta web para la gestión de proyectos que enfocada en la identificación de riesgos y eliminación de obstáculos.	-Permite rastreo de tareas y agregar seguidores -Sistema de recordatorio y notificaciones -No incluye funciones para llevar control del aspecto financiero de los proyectos -Creación de gráficos de Gantt -Tableros Kanban
		Wrike	-Free -Professional \$9.80 por usuario/mes -Business \$24.80 por usuario/mes -Enterprise cotización con vendedores	4.6/5	4.24/5	Herramienta web altamente personalizable con uso nativo de diagramas Gantt.	-Configurable con metodología Scrum -Generación automática de informes del estado y progreso de los proyectos (personalizable) -Etiquetas para mejor control de proyectos, carpetas y tareas -Soporte nativo widget para asociarse con Google drive, Jira, Microsoft OneDrive, Slack, Outlook, Github, entre otros -API para personalizar integración
		Azure DevOps Boards	-Básico \$6 por usuario/mes -Básico+Test Plans \$52 por usuario/mes	4.4/5	sin calificar	Herramienta de gestión de proyectos de Microsoft que combina la planeación de sprints y el seguimiento flexible de los elementos de trabajo	-Seguimiento del progreso del proyecto en cada etapa del proceso de desarrollo -Paneles de Scrum integrados -Herramientas de análisis y widgets -Integración con Github -Permite personalizar paneles de datos -Compatible e integrable con Microsoft Teams y Slack

¹ Los costos de las herramientas fueron tomados de los sitios oficiales de cada una de ellas

² Las evaluaciones por parte de Gartner fueron tomadas del sitio (Gartner Inc, 2020)

³ Las evaluaciones por parte de Software Advice fueron tomadas del sitio (Software Advice, 2020)

Etapa	Detalle	Herramienta	Costos ¹	Evaluación Gartner peer insights ²	Evaluación Software Advice ³	Descripción	Características
							-Acceso a más de 1000 extensión y posibilidad de creación de extensiones propias
		VersionOne	-Versión gratuita -Enterprise \$29 por usuario/mes -Ultimate \$39 por usuario/mes	4.6/5	4.12/5	Herramienta web que soporta Scrum, Kanban, XP, metodologías de desarrollo híbridas y que facilita la planificación y seguimiento de las tareas realizadas por los diferentes equipos involucrados en los proyectos de software	-Mapea el flujo de trabajo en todo el ciclo de vida del software, permitiendo la planificación estratégica hasta la integración continua -Configurable con metodologías ágiles como Scrum, Kanban, XP o metodologías híbridas de herramientas y gestión de proyectos -Vistas de progreso y rendimiento personalizadas -Informes en todos los niveles de la organización -Asignaciones presupuestarias, seguimiento del gasto y pronóstico del presupuesto final
Código	Fase que comprende las herramientas que permiten la gestión y respaldo para el desarrollo del código. Las herramientas por excelencia son el control de versiones. Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 2	Github	-Versión gratuita -Team \$4 por usuario/mes -Enterprise \$21 por usuario/mes -Github One cotización con vendedores	4.8/5	4.76/5	Red social basado en Git que actúa como sistema de control de versiones	-Admite flujos de trabajo no lineales distribuidos -Escalable (número de usuarios) -Permite trabajos de desarrollo en paralelo -Compatible con comandos Git -Herramientas CI/CD para facilitar la integración de código -Sistema de notificaciones webhook -Admite rollback -Usa SHA1 para la identificación de objetos en el repositorio
		Bitbucket	-Versión gratuita -Standard \$3 por usuario/mes -Premium \$6 por usuario/mes	4.4/5	4.56/5	Sistema de control de versiones con plataforma web de Git con integración con Jira, Trello, Bamboo y Confluencer	-Multiplataforma -Compatible con comandos Git -Permite integración con herramientas de CI/CD como Jenkins -Admite wikis por proyecto -Permite feedback de los pull request y cambios por parte de usuarios -Rastreo de cambios -Sistema de notificación webhook -Integraciones personalizadas a través de REST API

Etapa	Detalle	Herramienta	Costos ¹	Evaluación Gartner peer insights ²	Evaluación Software Advice ³	Descripción	Características
		Gitlab	-Versión gratuita -Starter \$4 por usuario/mes -Premium \$19 por usuario/mes -Ultimate \$99 por usuario/mes	4.5/5	4.58/5	Suite de DevOps con sistema de usuario, repositorios públicos y privados e integración de herramientas que agilizan las etapas del ciclo de vida de desarrollo de software	-Multiplataforma -Compatible con comandos Git -Creación de wikis por proyecto -Integración con herramientas de la suite de DevOps -REST API para integraciones personalizadas -Notificaciones mediante sistema webhook -Suite de DevOps enfocado para trabajo en Dockers
		Subversion	Software Libre	sin calificar	sin calificar	Sistema de control de versiones de Apache abierto	-Sistema básico -Bloqueo de archivos -Repositorio espejo de solo lectura -Nativo cliente/servidor -API para enlaces a varios lenguajes de programación
Build	Fase en la que se compila el código para generar un artefacto que pasará a las siguientes fases, comprende: -Obtención del código -Integración de dependencias -Compilación -Generador de artefacto Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 3	Maven	Software Libre	sin calificar	sin calificar	Aplicación de software libre para gestión y construcción de software basado en java enfocada en el manejo de dependencias	-Repositorio central con gran variedad de librerías -Permite realizar secuencias de pruebas unitarias y de integración -Usa estándares de la industria -Permite integración con herramientas de CI/CD -Generación de reportes de pruebas -Generación de artefactos en diversos formatos (.jar/.war) -Integrada a IDE de java -Multiplataforma
		Gradle	-Trial de 30 días -Enterprise cotización con vendedores	sin calificar	sin calificar	Software para automatización de build que maneja dependencias. Es el sistema de compilación oficial de Android	-Depuración compartida para mayor velocidad en resolución de problemas -Compilación inteligente incremental -Soporta java, groovy, Kotlin -Es integrable con IDE de java mediante plugin -Estructura de carpetas y archivos para los proyectos -Genera artefactos en formatos JAR, WAR, EAR -Herramientas para empaquetamiento de código basado en Java Virtual Machine -Publicación de artefactos en repositorios Ivy y Maven -Caché de dependencias de terceros para evitar tráfico de red

Etapa	Detalle	Herramienta	Costos ¹	Evaluación Gartner peer insights ²	Evaluación Software Advice ³	Descripción	Características
		Jenkins	Software Libre	sin calificar	sin calificar	<p>Servidor de automatización de tareas escrito en java. Es de código abierto y cuenta con 2 cualidades importantes:</p> <ul style="list-style-type: none"> -Creación de Pipeline o cadenas de ensamblaje -Disponibilidad de gran cantidad de plugin para agregar funcionalidades <p>Su versión comercial es Cloudbees Jenkins con calificación de 4.5/5 por Gartner</p>	<ul style="list-style-type: none"> -Se puede integrar con herramientas de control de versiones como GitHub, bitbucket o gitlab -Se puede integrar con plataformas de nube (AWS, GPC Oracle Cloud) -Herramienta web -Escalamiento automático -Fácil mantenimiento gracias a herramientas incorporadas para actualizaciones -Alrededor de 1000 complementos -Multiplataforma -Concurrencia gracias a la distribución de tareas en múltiples máquinas
		Docker	<ul style="list-style-type: none"> -Free (individual) -Pro \$5 por mes (individual) -Team \$7 por usuario/mes 	4.6/5	sin calificar	<p>Herramienta que empaqueta software y sus respectivas dependencias en un contenedor virtual, se considera como una de las estrategias actuales de compilación y despliegue de software a nivel global</p>	<ul style="list-style-type: none"> -Portable -Compatible con diversos lenguajes de programación -Comunidad activa con aportes diarios -Permite integración con herramientas de CI/CD
		Azure DevOps	<ul style="list-style-type: none"> -Básico \$6 por usuario/mes -Básico+Test Plans \$52 por usuario/mes 	4.4/5	sin calificar	<p>Suite de DevOps de Microsoft que comprende varias herramientas integradas entre sí:</p> <ul style="list-style-type: none"> -Azure Boards -Azure repos -Azure Pipeline -Azure Test Plans -Azure Artifacts 	<ul style="list-style-type: none"> -Herramienta web -Multiplataforma de modo web -Compatible con comandos Git -Alta integración con toda la suite de Microsoft Azure -Permisos de usuario y rol robustos - Permite compilación de código en cualquier lenguaje -Cuenta con un gran número de plugin para la asociación con múltiples herramientas del mercado
Test	Fase de pruebas en la que se verifican que cambios aplicados no hayan afectado la funcionalidad del	Junit	Software Libre	sin calificar	sin calificar	<p>Framework especializado en pruebas unitarias</p>	<ul style="list-style-type: none"> -Numerosa documentación disponible -Integración a múltiples componentes y aplicativos de java -Generación de reportes -IDE de java genera pruebas automáticas a través de plugin

Etapa	Detalle	Herramienta	Costos ¹	Evaluación Gartner peer insights ²	Evaluación Software Advice ³	Descripción	Características
	software. Esta etapa se compone por diferentes pruebas que pueden ser automatizadas. Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 2 y 3	Selenium	Software Libre	4.5/5	sin calificar	Conjunto de artefactos informáticos que ejecuta pruebas sobre navegadores web simulando el comportamiento de un usuario frente a una página web	<ul style="list-style-type: none"> -Permite la creación, modificación ejecución y grabación/reproducción de resultados de pruebas en aplicaciones web -Comandos Selenium orientados para la ejecución de acciones sobre objetos de un navegador -Manejo de puntos de verificación (breakpoints) facilitando la depuración -Especializado en pruebas de regresión -Genera reportes a través de videos simulando el comportamiento de un usuario frente a una aplicación web -Compatible con múltiples lenguajes de desarrollo
		SauceLabs	<ul style="list-style-type: none"> -Live Testing con 3 opciones: <ul style="list-style-type: none"> * \$19/mes en máquinas virtuales * \$49/mes máquinas físicas * \$68/mes híbrido -Virtual cloud \$149/mes -Real Device Cloud \$399/mes 	4.0/5	sin calificar	Framework de pruebas basado en Selenium que posee una plataforma web como servidor de pruebas remotas	<ul style="list-style-type: none"> -Versatilidad en las pruebas (combinaciones de sistemas operativos, navegadores web, resolución de pantalla y otras características) -Comandos Selenium orientados para la ejecución de acciones sobre objetos de un navegador -Manejo de puntos de verificación (breakpoints) facilitando la depuración -Especializado en pruebas de regresión -Integración con herramientas para CI/CD -Genera reportes a través de videos simulando el comportamiento de un usuario frente a una aplicación web -Compatible con múltiples lenguajes de desarrollo
		Device Farm AWS	<ul style="list-style-type: none"> -Pago por uso \$0.17/minuto de uso -Pruebas ilimitadas desde \$250/mes -Dispositivos dedicados desde \$200/mes 	sin calificar	sin calificar	Entorno de pruebas para aplicaciones web y móviles	<ul style="list-style-type: none"> -Permite pruebas en diferentes dispositivos en paralelo -Permite realizar pruebas de aplicaciones nativas e híbridas para Android, iOS, y FireOS -Permite acceso remoto de aplicaciones Android y iOS para pruebas interactivas -Generación de reportes y grabaciones de calidad e informativos -Pruebas móviles y web sobre dispositivos reales -Posee un API y SDK compatibles con múltiples lenguajes
Release	Fase en la cual ya existe un artefacto o ejecutable producto de las fases anteriores, las herramientas empleadas deben enforzarse en 3	Jfrog Artifactory	<ul style="list-style-type: none"> -Free -Pro Team \$98/mes -Enterprise \$699/mes -Enterprise+ cotización con vendedores 	sin calificar	sin calificar	Repositorio universal integrable con cualquier ecosistema existente, que permite rastrear artefactos desde las fases de desarrollo hasta producción	<ul style="list-style-type: none"> -Admite paquetes de software creados en cualquier lenguaje de programación o tecnología -Admite gestión binaria de extremo a extremo -Plataforma web -Gestión de registros de dockers con alta disponibilidad -Permite la automatización y control de todas las fases del ciclo de vida de DevOps

Etapa	Detalle	Herramienta	Costos ¹	Evaluación Gartner peer insights ²	Evaluación Software Advice ³	Descripción	Características
	aspectos clave: -Seguridad de acceso -Seguridad de permanencia -Alta disponibilidad Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 2 y 3	Sonatype Nexus	-Free -Starter \$120 por usuario/año -Basic \$48,000/año (100 usuarios) -Premium \$85,500/año (100 usuarios) -Enterprise (basado en volumen)	sin calificar	sin calificar	Gestiona componentes desde el desarrollo hasta la entrega: binarios, contenedores, ensamblajes y productos terminados	-Soporte avanzado para el ecosistema Java Virtual Machine (JVM) -Compatible con herramientas como Eclipse, IntelliJ, Hudson, Jenkins, entre otros -Autenticación SAML / SSO para una seguridad mejorada y una experiencia de inicio de sesión único
		Google Container Registry	Cálculo en función de varios aspectos	4.5/5	sin calificar	Manejador de registro de contenedor de Google	-Análisis automático de vulnerabilidades de imágenes -Posee una base de datos con registros de paquetes y sistemas operativos con vulnerabilidades, usado para analizar imágenes Docker automáticamente -Control de acceso a repositorios e imágenes mediante políticas -Permite la inserción y extracción de imágenes Docker en repositorios para su uso en sistemas de desarrollo propios o en definiciones dentro del servicio Google Kubernetes Engine -Al ser un servicio no se administra ningún servidor
		Docker Hub	-Free -Pro \$5/mes (individuo) -Team \$7 por usuario/mes (organizaciones)	sin calificar	sin calificar	Registro de contenedor docker básico con alta disponibilidad de imágenes que pueden ser usadas como base para aplicaciones	-Control de acceso a repositorio e imágenes mediante usuarios -Permite la inserción y extracción de imágenes Docker en repositorios para su uso en sistemas de desarrollo propios o en definiciones de cualquier entorno o servicios -Al ser un servicio no se administra ningún servidor -Alta disponibilidad de recursos por altas tasas de transferencia
		Google Cloud Storage	Cálculo en función de varios aspectos	4.8/5	sin calificar	Servicio de almacenamiento de objetos de alta escalabilidad y alcance mundial similar a Amazon Simple Storage Service	-Encriptación de datos -Control de versiones -Sistema de notificaciones configurables -Logs de auditoría de acceso -Sistema de seguridad de acceso por autenticación de roles y/o ACL -Ciclo de vida configurable

Etapa	Detalle	Herramienta	Costos ¹	Evaluación Gartner peer insights ²	Evaluación Software Advice ³	Descripción	Características
Deploy	<p>Fase de implementación de la nueva versión obtenida en fases previas, herramientas deben permitir el pase automático a producción considerando sistemas de respaldo y rollback</p> <p>Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 3 y 4</p>	AWS CodeDeploy	Cálculo en función de varios aspectos	sin calificar	sin calificar	<p>Servicio de implementación de componentes capaz de ejecutar tareas que permitan la implementación de aplicaciones en diferentes servicios informáticos</p>	<ul style="list-style-type: none"> -Ideal para implementación y actualización de aplicaciones no dockerizadas, código, funciones AWS Lambda -Consola de administración permite la gestión de las implementaciones mostrando el estado de todas las acciones en un mismo sitio -Es posible detener y revertir implementaciones en caso de errores -Permite aumento de capacidad dinámica para las implementaciones que lo requieran -Permite la ejecución de implementaciones en PC locales de la organización -SDK y API de AWS lo vuelven compatible con herramientas de terceros -Sistema de notificaciones push para el monitoreo del estado de las implementaciones
		Ansible	Cuenta con 2 versiones: standard y premium, y se requiere contactar a equipo de ventas para cotización	4.7/5	sin calificar	<p>Software de automatización que permite el aprovisionamiento de infraestructura, gestión de configuración y construcción y despliegue de aplicaciones</p>	<ul style="list-style-type: none"> -Admite aprovisionamiento en la nube -Gran volumen de documentación disponible, así como una comunidad activa -Disponible para varias distribuciones de Linux -No necesita agentes desplegados en instancias finales (solo python en nodos)
		Kubernetes	Software Libre	sin calificar	sin calificar	<p>Plataforma de administración y orquestación de servicios y aplicaciones, similar a un sistema de cluster que permite la administración de redes, almacenamiento, escalamiento, balance de carga y despliegue y destrucción de aplicaciones</p>	<ul style="list-style-type: none"> -Permite el despliegue de todo tipo de aplicación siempre y cuando esté en docker -Despliega Docker y permite la integración con herramientas de CI/CD para la obtención de configuraciones de despliegue -No provee ni obliga a usar sistema o lenguaje de configuración -Soportado por plataformas Cloud

Etapa	Detalle	Herramienta	Costos ¹	Evaluación Gartner peer insights ²	Evaluación Software Advice ³	Descripción	Características
		Elastic container Services (ECS)	Cálculo en función de varios aspectos	4.6	sin calificar	Servicio de administración de contenedores compatible con múltiples servicios de AWS, su fuerte es la escalabilidad, fácil configuración y adaptabilidad de herramientas de terceros	<ul style="list-style-type: none"> -Herramienta a cargo de la administración e servidores -Herramientas de AWS se encargan de auto escalamiento y balanceo de carga -Logs y métricas centralizadas en cloudWatch -Sistema de alertas y notificaciones configurable -Integración con herramientas de CI/CD -Proporciona el ambiente idóneo para que una cualquier número de una misma aplicación se pueda ejecutar
Operate	Fase enfocada a los recursos en los que se ejecuta el software. Las herramientas y servicios para esta fase deben facilitar el aprovisionamiento de infraestructura y monitoreo de componentes. Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 3 y 4	Vagrant	<ul style="list-style-type: none"> -Free -Personal \$5/mes/box privada -Organization \$25/mes por box privada 	sin calificar	sin calificar	Herramienta de construcción y administración de Infraestructura centrado en la automatización	<ul style="list-style-type: none"> -Integración con herramientas de administración de configuración como Chef, Puppet, Ansible y Salt -Multiplataforma (Mac, Linux, Windows) -Provisiona un entorno de desarrollo ideal para equipos DevOps
		CloudWatch	<ul style="list-style-type: none"> -Versión o capa gratuita -Versión o capa pagada se calcula en función de ciertas métricas 	4.6/5	sin calificar	Servicio de monitoreo para recursos en la nube de AWS y aplicaciones. Permite la visualización de todo el sistema respecto al uso de recursos, rendimiento de aplicaciones, estado operativo	<ul style="list-style-type: none"> -Configurado para integrarse de forma inmediata con Elastic Compute Cloud -Tableros de control de rendimiento para instancias RDS (Servicio de Bases de Datos relacionales) -Monitoreo de latencia de lectura/escritura y mediciones similares a través de Amazon Elastic block store -Monitoreo de métricas como espacio de almacenamiento o memoria libre de instancias de bases de datos RDS -Monitoreo de mensajes enviados/recibidos y métricas similares en colas de mensajes (Amazon SQS) -Monitoreo de mensajes publicados/entregados y métricas comunes en Servicios de notificación de Amazon (AWS SNS)
		Chef	Cotización con equipo de ventas	4.2/5	sin calificar	Automatiza la configuración y mantenimiento de servidores de una organización, y es integrable con plataformas basadas en la nube	<ul style="list-style-type: none"> -Fomenta la implementación de la Infraestructura como código -Acelera la adopción de Cloud -Automatiza el proceso de gestión de configuraciones -Actualizaciones de hardware y aplicaciones aplicadas de forma dinámica

Etapa	Detalle	Herramienta	Costos ¹	Evaluación Gartner peer insights ²	Evaluación Software Advice ³	Descripción	Características
		Terraform	-Free hasta 5 usuarios -Team & Governance \$20 por usuario/mes -Business cotización con ventas	4.3/5	5/5	Software de aprovisionamiento de infraestructura que soporta varias plataformas Cloud	-Permite la administración de recursos y monitoreo del versionamiento de estos -Adaptable a los recursos de la plataforma Cloud donde se ejecuta -Guarda el estado de los recursos administrados -Permite el cálculo de cambios que se efectuarán sobre la plataforma previo a su ejecución -Fomenta la implementación de la Infraestructura como código
		Prometheus	Cotización con equipo de ventas	sin calificar	4.44/5	Base de datos diseñada para para el monitoreo de objetivos, servidores, bases de datos, máquinas virtuales, entre otros	-Modelo de datos simple pero potente -Lenguaje de consulta que permite el análisis del rendimiento de aplicaciones e infraestructura -Almacena series temporales en memoria y en disco local -Bibliotecas personalizadas fáciles de implementar
Monitoreo	Fase en la cual se realiza el seguimiento del comportamiento de la aplicación. El monitoreo debe contemplar la validación de componentes o características de las aplicaciones, comportamiento de los componentes en producción, procesamiento de datos generados por los usuarios y modo de uso de la aplicación por parte de los usuarios. Estas herramientas son las sugeridas para seleccionar y	New Relic	Cálculo en función de varios aspectos	4.7/5	sin calificar	Servicio de rendimiento de aplicaciones web que proporciona un monitoreo de servidor flexible y dinámico permitiendo a los equipos operativos tomar decisiones respecto a sistemas complejos	-Visualización de datos de rendimiento de las aplicaciones como tiempo de respuesta del navegador po geografía, transacciones web realizadas en tiempo real, entre otras -Compatible con todos los lenguajes de desarrollo web -Posee un tablero de instrumentos flexibles -Correlaciona la experiencia del usuario final con el rendimiento de las aplicaciones -Conecta el rendimiento de la infraestructura y la aplicación -Herramientas de diagnóstica permiten el análisis de errores en tiempo real
		Elasticsearch	-Estándar \$16/mes -Oro \$19/mes -Platino \$22/mes -Enterprise cotización con equipo de ventas	4.4/5	sin calificar	Base de datos diseñada para almacenar, recuperar y administrar datos orientados a documentos o semiestructurados. Usado por Wikipedia, StackOverflow, Github, entre otros	-Datos son almacenados en formato JSON -Es escalable hasta petabytes de datos estructurados y no estructurados -Puede ser usado como reemplazo de almacenes de documentos -Mejora del rendimiento de búsqueda a través de la desnormalización - Código abierto -Funciones expuestas como REST API

Etapa	Detalle	Herramienta	Costos ¹	Evaluación Gartner peer insights ²	Evaluación Software Advice ³	Descripción	Características
	evaluar su implementación en el nivel de madurez 3 y 4	Kibana	Software Libre	sin calificar	sin calificar	Herramienta de código abierto para la exploración y visualización de datos que permite el análisis de registros, monitoreo de aplicaciones y casos de uso de inteligencia operativa	<ul style="list-style-type: none"> -Generación de histogramas, gráficos de líneas, gráficos circulares, mapas térmicos y soporte geoespacial -Integración con Elasticsearch -Permite la restricción de datos lo que facilita el intercambio seguro de estos entre paneles -Contiene varios métodos de búsqueda en los datos de registro -Generación de informes
		Sentry	<ul style="list-style-type: none"> -Developer \$0 -Team \$26/mes -Business \$80/mes -Enterprise cotización con equipo de ventas 	4.7/5	sin calificar	Plataforma para el monitoreo de errores, permite descubrir, clasificar y priorizar errores en tiempo real	<ul style="list-style-type: none"> -Mejora de la supervisión del rendimiento de la aplicación con trazas de pila -Diferencia errores identificados con anterioridad de los errores nuevos -Monitoreo de datos en tiempo real
		Datadog	Cálculo en función de varios aspectos	4.6/5	4.57/5	Herramienta de monitoreo y análisis que permite determinar métricas de rendimiento a través de monitoreo de eventos de infraestructura y servicios en la nube	<ul style="list-style-type: none"> -Permite el monitoreo de servidores, bases de datos y diversas herramientas -Software disponible para implementación on-premise o como servicio (SaaS) -Multiplataforma (Windows/Linux/Mac) -Le brinda al equipo de TI/DevOps una vista única de su infraestructura -Tableros son personalizables -Alertas basadas en problemas críticos -Recopilación y análisis automático de registros, tasas de error, latencias, entre otras

CONCLUSIONES

Para la elaboración de la guía general de autoevaluación del estado actual de las instituciones financieras se realizó el estudio del marco DevOps a través del material del curso oficial para la certificación de DevOps Fundamentals, así como también se realizaron entrevistas a consultores con experiencia en proyectos de implementación de este marco de trabajo en instituciones financieras nacionales e internacionales.

Para esta guía se consideraron las áreas de enfoque que son: aspecto cultural, de prácticas y procesos, y tecnología y automatización que son los aspectos fundamentales conocidos como el paquete completo de DevOps; la guía cuenta con una matriz de entrevistas hacia las diferentes áreas que deben ser analizadas dentro de una institución financiera y adicionalmente se sugieren cuestionarios que son herramientas para el relevamiento y análisis de la información obtenida.

Adicionalmente la guía cuenta con una matriz de madurez en la cual, en base al análisis realizado por el evaluador, se podrá determinar el nivel de madurez de cada aspecto

Se elaboró un road map de 5 fases apegado al modelo de madurez de CMMI, en cada fase se especifican los objetivos respecto a los aspectos cultural, procesos y prácticas, y tecnología y automatización que deben ser alcanzados por la institución durante el proceso de adopción de DevOps. Una vez que el evaluador concluya el levantamiento y análisis de la información, es decir, obtenga el nivel de madurez de la institución que está siendo evaluada, podrá utilizar este road map general para desarrollar el road map específico ajustado a la realidad de la organización que busca implementar DevOps.

Para la elaboración de la matriz de herramientas se realizó la investigación utilizando material bibliográfico de DevOps, información propuesta por las marcas más reconocidos a nivel mundial como Amazon, Microsoft, Google, entre otros; y se consideraron opciones de software libre. Las herramientas recopiladas en esta matriz fueron clasificadas de acuerdo

con las etapas del ciclo de vida de las aplicaciones y su integración con operación, conocido como el ciclo de vida de DevOps, así como también de acuerdo con la fase del nivel de madurez del road map general para la adopción de DevOps. La matriz contiene una descripción breve de cada herramienta, así como sus características clave, sus costos y calificaciones provistas por Gartner peer insights y Software Advice.

El romper los silos de comunicación existentes en las instituciones bancarias es uno de los objetivos de DevOps y uno de los que representa mayor esfuerzo, ya que implica el cambio de la cultura de trabajo que han mantenido las instituciones durante muchos años. Como se menciona en el road map se debe iniciar con el cambio en la comunicación y colaboración a través de herramientas y procesos, y sobre todo esto se logra a través de la conformación de equipos multifuncionales con los diferentes roles y perfiles necesarios participando durante todo el ciclo del desarrollo del software de forma colaborativa. El cambio cultural requiere una constante evangelización (couching) por lo que este proceso es transversal en todas las fases de la implementación de DevOps, inclusive en la mejora continua.

Al cumplir los objetivos propuestos en este trabajo de investigación se puede responder a la pregunta de investigación planteada de forma afirmativa, es decir, fue posible el desarrollo de unas guías generales que pueden ser utilizadas para la autoevaluación de una institución financiera y la elaboración de un posible road map para la adopción de la cultura DevOps.

RECOMENDACIONES

Se recomienda que los funcionarios que realicen la autoevaluación tengan conocimiento y algunos años de experiencia dentro de la institución financiera, además que la evaluación sea realizada bajo un criterio objetivo de la realidad del estado actual de la institución financiera en relación con el marco de cultura DevOps.

Es necesario que las guías sean revisadas y en base a esta, realizar la planificación del levantamiento de información preparando las entrevistas a realizar a cada una de las áreas sugeridas.

De acuerdo con la realidad de cada empresa se pudieran adicionar preguntas formuladas de tal manera que la información obtenida pueda ser ubicada en uno de los niveles de madurez.

El road map general planteado contiene tiempos estimados para su cumplimiento, sin embargo, cada organización deberá planificar su road map específico de acuerdo a los recursos disponibles y a la velocidad con la que la institución decida realizar el proceso de adopción de DevOps.

Una vez que la institución inicie la ejecución de la implementación, podría ser necesario repetir el proceso de autoevaluación para validar si las acciones tomadas han llevado a las empresas a un siguiente nivel de madurez, y en función de los resultados confirmar o realizar los ajustes necesarios al road map definido.

Dado que el ambiente tecnológico es muy cambiante, es posible que surjan herramientas adicionales a las que se sugieren en la matriz de herramientas, en este aspecto es necesario que la organización investigue las herramientas disponibles en el mercado en el momento en el que se requiera implementar.

REFERENCIAS BIBLIOGRAFICAS

05_lectura_Tecnicas_e_Instrumentos.pdf. (s. f.). Recuperado 1 de julio de 2020, de http://brd.unid.edu.mx/recursos/Taller%20de%20Creatividad%20Publicitaria/TC03/lecturas%20PDF/05_lectura_Tecnicas_e_Instrumentos.pdf

Amazon Web Services, Inc. (2020). *¿Qué es el control de código fuente? - Amazon Web Services*. Amazon Web Services, Inc. <https://aws.amazon.com/es/devops/source-control/>

Andersson, L. (2017). *ITIL and DevOps – better together*. AXELOS. <https://www.axelos.com/news/blogs/september-2017/itil-and-devops-better-together>

Arsys. (2016, febrero 25). La estrecha relación entre DevOps y Cloud. *Blog de arsys.es*. <https://www.arsys.es/blog/soluciones/cloud/cloud-computing/la-estrecha-relacion-entre-devops-y-cloud/>

Atlassian. (s. f.). *Software de control de versiones: Descripción general*. Bitbucket. Recuperado 20 de junio de 2020, de <https://bitbucket.org/product/es/version-control-software>

Automating the bank's back office / McKinsey. (2012). <https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/automating-the-banks-back-office>

Avendaño Carbellido, O. (2018). Los retos de la banca digital en México. *Revista IUS*, 12(41), 87-108.

- BBVA. (2015, enero 14). Qué es el «fintech» y cómo las «startups» quieren innovar en los servicios financieros. *BBVA NOTICIAS*. <https://www.bbva.com/es/que-es-el-fintech/>
- BBVA. (2019, septiembre 17). Neobancos: Qué son, cómo operan y cuáles son sus características. *BBVA NOTICIAS*. <https://www.bbva.com/es/neobancos-que-son-y-como-operan/>
- Brown, T. (2020). *DevOps vs Agile: What's the difference?* Opensource.Com. <https://opensource.com/article/20/2/devops-vs-agile>
- Cambio-de-Cultura-DevOps-en-Banca-1.pdf*. (s. f.). Recuperado 13 de junio de 2020, de <https://insights.nisumlatam.com/wp-content/uploads/2017/11/Cambio-de-Cultura-DevOps-en-Banca-1.pdf>
- Cap1.pdf*. (s. f.). Recuperado 10 de junio de 2020, de <http://www.economia.unam.mx/secss/docs/tesisfe/GuerraMC/cap1.pdf>
- Castellanos, R. (2019, marzo 21). *Mejores Herramientas de Integración Continua para desarrollo de aplicaciones móviles*. Medium. https://medium.com/@ricardocastellanos_13596/mejores-herramientas-de-integraci%C3%B3n-continua-para-desarrollo-de-aplicaciones-m%C3%B3viles-46b88a52a2df
- Consultora Cognizant. (s. f.). *DevOps automatiza software de banca—Caso práctico / Cognizant España*. Recuperado 13 de junio de 2020, de <https://www.cognizant.com/es-es/case-studies/devops-strategy-change-optimization>

Consultora Cognizant. (2020). *What Are Legacy Core Banking Systems?* | Cognizant.
<https://www.cognizant.com/glossary/legacy-banking-systems>

Cordova, A. (2018, enero 17). Banco del Pacífico lanzó BDP Wallet, la primera billetera digital de Ecuador. *Blog de Prensa Banco del Pacífico*.
<https://bancopacificoprensa.ec/banco-del-pacifico-lanzo-bdp-wallet-la-primer-billetera-digital-de-ecuador/>

Das, L., Lau, L., & Smith, C. (2017, febrero 26). *Five cultural changes you need for DevOps to work* | McKinsey & Company.
<https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/digital-blog/five-cultural-changes-you-need-for-devops-to-work>

DevopsMaturityAssessment.pdf. (s. f.). Recuperado 24 de agosto de 2020, de
<http://www.mediatekhub.com/docs/DevopsMaturityAssessment.pdf>

digital.ai. (2020). *Periodic Table of DevOps Tools* | Digital.ai.
<https://digital.ai/periodic-table-of-devops-tools>

El Telégrafo-Noticias del Ecuador y del mundo. (2019, abril 20). *La banca se une a la transformación digital para ser más competitiva*. El Telégrafo - Noticias del Ecuador y del mundo.
<https://www.eltelegrafo.com.ec/noticias/economia/4/banca-transformacion-digital-competitiva>

E-nquest, M. (2018, septiembre 25). ▷ ¿Por qué es importante una investigación exploratoria? 【 E-nquest 】 . *e-nquest*. <https://www.e-nquest.com/por-que-es-importante-investigacion-exploratoria/>

- ESAN Graduate School of Business. (2017). *Marketing relacional: Entre el valor y el precio*. <https://www.esan.edu.pe/apuntes-empresariales/2017/03/marketing-relacional-entre-el-valor-y-el-precio/>
- Gartner Inc. (2020). *Enterprise IT Software Reviews | Gartner Peer Insights*. Gartner. <https://www.gartner.com/market/enterprise-agile-planning-tools/vendor/microsoft/product/azure-devops-formerly-visual-studio-team-services>
- González, C. (2019). *La transformación digital como motor de la banca latinoamericana*. La transformación digital como motor de la banca latinoamericana. <https://www.bbva.com/es/la-transformacion-digital-como-motor-de-la-banca-latinoamericana/>
- Heredia, R. (2015). *Por qué pensar en IT de dos velocidades | Ebanking News*. <https://www.ebankingnews.com/noticias/por-que-pensar-en-it-de-dos-velocidades-0034061>
- Israel. (2019, enero 16). *¿Cuáles son las herramientas DevOps?* Viewnext. <https://www.viewnext.com/cuales-son-las-herramientas-devops/>
- Jorge Gonzalez, Gerente de proyectos Maint S.A. (2020). *Proyecto de implementación de DevOps en bancos de Guayaquil* [Comunicación personal].
- Kim, G., Debois, P., Willis, J., Humble, J., & Allspaw, J. (2016). *The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations* (First edition). IT Revolution Press, LLC.

KORPORATE TECHNOLOGIES GROUP. (2017, septiembre 3). Cómo eliminar los silos de información. *Grupo Korporate*. <https://grupokorporate.com/como-eliminar-los-silos-de-informacion/>

¿Las fintech son más competitivas que los bancos? (s. f.). Recuperado 29 de mayo de 2020, de <https://www.americaeconomia.com/analisis-opinion/las-fintech-son-mas-competitivas-que-los-bancos>

Leclercq, L. (2019, octubre 4). Los bancos deben salir de la zona de confort de silos para crear un entorno de trabajo más eficiente. *Capgemini México*. <https://www.capgemini.com/mx-es/2019/10/los-bancos-deben-salir-de-la-zona-de-confort-de-silos-para-crear-un-entorno-de-trabajo-mas-eficiente/>

Ly, L. C. T., & Siesquén, L. I. S. (2012). *TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS*. 37.

Manas Shome, & Raghubir Bose. (2018). *Assessing DevOps: Assess DevOps Quantitatively for your IT, Business and Industry*. Independently published.

Martínez, S. (2017, julio 3). Evolución histórica de la Transformación Digital | Zemsania. *Zemsania Global Group*. <https://zemsaniaglobalgroup.com/evolucion-historica-transformacion-digital-economias-europeas/>

Microsoft Azure. (2020). *¿Qué es DevOps? Explicación de DevOps | Microsoft Azure*. <https://azure.microsoft.com/es-es/overview/what-is-devops/>

Novoseltseva, E. (2018, abril 3). Los beneficios y herramientas DevOps. *Apiumhub*. <https://apiumhub.com/es/tech-blog-barcelona/beneficios-herramientas-devops/>

NUEVO FINANCIERO. (2018, diciembre 14). 100 años de la digitalización financiera. La evolución de la tecnología en las finanzas. *Nuevo Financiero*.
<https://nuevofinanciero.com/digitalizacion-financiera-evolucion/>

Palacios, J. (2020). *Virtualización de Servicios como habilitador DevOps | Deloitte España*. Deloitte Spain.
<https://www2.deloitte.com/es/es/pages/technology/articles/virtualizacion-de-servicios-como-habilitador-devops.html>

PeopleCert. (2019). *PeopleCert Fundamentos de DevOps El Paquete Completo de Cultura, Prácticas y Automatización*.

PROGRESSA LEAN. (2015, mayo 22). *ORIGEN Y EVOLUCIÓN DEL LEAN MANUFACTURING*. Progressa Lean. <https://www.progressalean.com/origen-y-evolucion-del-lean-manufacturing/>

Quijano, J. (2018, abril 1). *El ciclo de DevOps, una guía para iniciarse en las fases que lo componen*. Genbeta. <https://www.genbeta.com/desarrollo/el-ciclo-de-devops-una-guia-para-iniciarse-en-las-fases-que-lo-componen>

Rae, B. (2017, diciembre 1). *ITIL, DevOps and the search for silver bullets*. AXELOS.
<https://www.axelos.com/news/blogs/december-2017/itil-devops-and-the-search-for-silver-bullets>

Rebaza, J. C. V. (2007). *Metodologías Ágiles*. 37.

RedHat. (2017). *¿Quiere saber qué es la transformación digital? Obtenga información acerca de qué es, cómo incorporarla, cómo influye en los mercados y por qué es mucho mejor con open source*.

<https://www.redhat.com/es/topics/digital-transformation/what-is-digital-transformation>

Revista Online Vanguardía Industrial. (2019, junio 11). “Un CEO exitoso necesita ser un CEO ágil”: KPMG. *Vanguardia Industrial*. <https://www.vanguardia-industrial.net/un-ceo-exitoso-necesita-ser-un-ceo-agil-kpmg/>

Software Advice. (2020). *Business Software Reviews from Software Advice®*. <https://www.softwareadvice.com/>

Suarez, F. (2020). *Dependencia de Empresas a proveedores por tecnología Legacy* [Comunicación personal].

Suárez, F. (2020). *DIFICULTADES DE TI PARA AGREGAR VALOR AL NEGOCIO* [Comunicación personal].

SUPERINTENDENCIA DE BANCOS. (s. f.). *Enlaces – Superintendencia de Bancos*. Recuperado 30 de junio de 2020, de <https://www.superbancos.gob.ec/bancos/enlaces/>

Sutil, E. (2017, diciembre 12). *El pensamiento Lean y su influencia en el movimiento DevOps*. Xeridia. <https://www.xeridia.com/blog/el-pensamiento-lean-y-su-influencia-en-el-movimiento-devops>

tecmanag. (2019, mayo 19). *Certificados Internacionales ITIL 4 | PMP | DEVOPS | COBIT | SCRUM y Más—Certificado Internacional. Certificados Internacionales ITIL 4 | PMP | DEVOPS | COBIT | SCRUM y Más*. <https://tecmanagement.org/itil-4-y-el-sistema-de-valor-del-servicio/>

Temenos – Company. (2020). *El número de ejecutivos bancarios afirmando que las nuevas tecnologías impulsaran la banca global aumenta a 57% en un año, revela el reporte de the Economist Intelligence Unit lanzado por Temenos— Temenos Latin America*. <https://www.temenos.com/es/news/2020/06/09/el-numero-de-ejecutivos-bancarios-afirmando-que-las-nuevas-tecnologias-impulsaran-la-banca-global-aumenta-a-57-por-ciento-en-un-ano/>

The (Short) History of DevOps. (2012). <https://www.youtube.com/watch?v=o7-luYS0iSE>

Villegas, J. (2019). *HERRAMIENTAS DEVOPS*.

Vintimilla, P. (2020, junio 13). *¿Cómo se integran DevOps e ITIL?* [Llamada por teléfono].

Zauzich, I. (2015). *El método Agile y sus ventajas para el sector financiero*. <https://blog.cobiscorp.com/ventajas-agile-para-el-sector-financiero>

ANEXOS

Anexo 1:

Formato de encuesta

Encuesta cultura DevOps

Estimados, como parte de mi proyecto de titulación me es necesario conocer información general de los bancos en los que laboran respecto a la cultura DevOps.

¿En qué banco labora? *

Texto de respuesta corta

.....

¿Cuál es su cargo? *

Texto de respuesta corta

.....

⋮

¿Cuál es el tiempo promedio de entrega de proyectos de desarrollo en su institución bancaria? *

Menos de 6 meses

Entre 6 y 12 meses

Más de 12 meses

Más de 24 meses

Otra...

⋮

¿Qué metodologías de desarrollo utiliza? Elija todas las que apliquen. *

Extreme programming

Scrum

Kanban

Agile inception

Tradicional Cascada

Otra...

El proceso de testing de software en su institución bancaria es: *

- Realizado de forma manual
- Dependiente de otras personas, tiene ciclos de espera
- Demorado en las correcciones

¿Cómo es su proceso para el pase de software entre entornos? *

- Manual
- Automático

Si es manual, ¿la documentación del proceso es extensa?

- Sí
- No

¿Los objetivos a cumplir de sus departamentos de desarrollo y operaciones de TI son compartidos? *

- Sí
- No

⋮

Las otras empresas que forman parte de su competencia entregan nuevos productos al mercado basados en soluciones informáticas en: *

- Menos tiempo que su institución
- Igual tiempo que su institución
- Mayor tiempo que su institución
- No tengo conocimiento

¿Ha escuchado sobre la cultura DevOps? *

- Sí
- No

¿En la institución bancaria en la que usted labora se ha implementado esta cultura o esta en proceso de implementación? *

- Sí
- No

Si respondió sí, ¿qué razón o razones lo llevaron a hacerlo?

- Mejorar la velocidad con la que se desarrolla y despliega software
- Mejorar la cultura de trabajo a una más colaborativa
- Adaptarse a las nuevas tendencias tecnológicas
- El proceso de desarrollo de software afectaba con el cumplimiento las metas planteadas por el negocio
- Otra...

⋮

¿Cuál fue el mecanismo que uso para la implementación de esta cultura?

- Contratación de consultoría externa
- Con personal interno del banco
- Combinación entre consultor externo y personal del banco

Si respondió no, ¿qué razón o razones lo llevaron a no hacerlo?

- Desconozco que es DevOps
- No se como empezar con la adopción de esta cultura
- No se disponen de perfiles con el conocimiento o experiencia para la implementación
- No se cuenta con el tiempo del personal para iniciar un proyecto de esta naturaleza
- No es de su interés
- Otra...

¿Consideraría útil contar con guías para realizar una autoevaluación de su organización y establecer su nivel de madurez con relación a la cultura DevOps? *

- Sí
- No

¿Consideraría que el contar con estas guías podría acelerar su decisión de optar por la cultura DevOps? *

- Sí
- No

Anexo 2:

Tabulación y análisis de encuestas

	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 6	Pregunta 11
	¿Ha participado en algún caso de implementación de DevOps en bancos la cual haya resultado exitosa? O en su defecto ¿Conoce de algún o algunos casos de éxito de implementación de DevOps en bancos?	¿Considera que DevOps se ha posicionado como una de las mejores metodologías para mejorar el proceso de desarrollo y despliegue de software? ¿Qué lo lleva a pensar esto?	A su criterio ¿es necesario que los bancos realicen un assestment o proceso de evaluación de su estado actual con el objetivo de prepararse para la adopción de DevOps?	Si el banco contara con un proceso guía, ¿podría iniciar la evaluación de su estado en relación a DevOps y llegar a definir una ruta de implementación?	Otro de los ejes principales de DevOps son las herramientas para la automatización, ¿cree usted que debe considerarse un mínimo de implementación de herramientas para la automatización encada nivel de madurez?
Arquitecto Certificado DevOps, Consultora MAINT	Si	Si	Si	Si	Si
Gerente de Proyecto, Consultora MAINT	Si	SI	SI	Si	Si
Consultor Xeridia (España)	Si	SI	SI	Si	Si

	Pregunta 4		
	¿Cuáles serían todos los aspectos que deben ser evaluados en el assestment?		
	Tecnología	Procesos	Personas
Arquitecto Certificado DevOps, Consultora MAINT	SI	SI	SI
Gerente de Proyecto, Consultora MAINT	SI	SI	SI
Consultor Xeridia (España)	SI	SI	SI

	Pregunta 5						
	¿Qué elementos de evaluación debe contener el assestment?						
	Inventario de aplicaciones y funcionalidad a alto nivel	Inventario de tecnologías utilizadas	Metodologías utilizadas	Evaluación de las soft y hard skills de los equipos de trabajo	Capacidad del Cambio	Tamaño de las aplicaciones, complejidad, importancia del valor	Enfoque cultural: conocer la forma actual de trabajo en cuanto a valores, objetivos, prácticas, actitudes, silos, perfiles
Arquitecto Certificado DevOps, Consultora MAINT	x	x	x	x			
Gerente de Proyecto, Consultora MAINT	x	x		x	x	x	x
Consultor Xeridia (España)	x	x		x	x		

Pregunta 7	En su experiencia, ¿cuáles piensa que son los problemas que evitan que un banco arranque un proyecto de adopción de esta cultura?
Arquitecto Certificado DevOps, Consultora MAINT	La principal razón es el cambio de paradigma al momento de armar los equipos y empoderarlos para que sean autosuficientes y autoorganizados, algo que colisiona con las muy tradicionales estructuras departamentales
Gerente de Proyecto, Consultora MAINT	La estructura organizativa suele ser el principal problema. La existencia de silos y departamentos. Esto genera serios problemas de comunicación y limitan enormemente el trabajo en equipo
	Hay otros elementos tecnológicos y de metodología como uso de tecnología muy antigua y monolítica y uso de procesos waterfall que limitan mucho la adopción de DevOps
Consultor Xeridia (España)	La estructura organizativa suele ser el principal problema. La existencia de silos y departamentos. Esto genera serios problemas de comunicación y limitan enormemente el trabajo en equipo
	Hay otros elementos tecnológicos y de metodología como uso de tecnología muy antigua y monolítica y uso de procesos waterfall que limitan mucho la adopción de DevOps

Pregunta 8	A su criterio, ¿considera que los bancos que no puedan invertir en consultoría y que no disponen de personal con experiencia en DevOps, si dispone de una guía metodológica basada en los conceptos DevOps pudieran ejecutar su proceso de autoevaluación?	
Arquitecto Certificado DevOps, Consultora MAINT	SI	Sin duda, contar con una guía es importante para una autoevaluación
Gerente de Proyecto, Consultora MAINT	SI	El tema de costo de contratación de una consultoría es uno de los principales problemas que van en relación con la falta de experiencia en DevOps del personal, en caso de tener una metodología base, se podría iniciar un proceso de autoevaluación y conocer que camino deben seguir para la adopción del Modelo Devops
Consultor Xeridia (España)	SI	Siempre es conveniente una visión externa para determinar los problemas de la organización. Se puede hacer una evaluación inicial genérica, para el proceso de implementación, considero necesario la participación una persona experta, la transformación integral de la organización es una tarea compleja

Pregunta 9	¿Considera usted que esta guía puede ser diseñada en base a la documentación oficial DevOps e investigación sobre casos de implementación exitosa?	
Arquitecto Certificado Devops Consultora MAINT	SI	DevOps se nutre de varias fuentes y los casos de implementación exitosas pueden marcar muchas pautas si en la investigación se incluyen casos de implementación de organizaciones del mismo sector o región
Gerente de Proyecto Consultora MAINT	SI	Si lo considero, ya que pueden ser utilizados como referencia para el camino a tomar para la adopción del Modelo
Consultor Xeridia (España)	SI	Una Guía general para la auto evaluación, sin embargo no debería entenderse como una guía definitiva de transformación

Pregunta 10	De su experiencia en los proyectos de implementación de DevOps ¿cuáles son los niveles de madurez de DevOps que deben ser considerados durante el proceso de adopción? ¿Cuáles son los indicadores que me permiten encasillar un banco en cada nivel?
Arquitecto Certificado Devops Consultora MAINT	Recomienda basarse en el modelo de madurez CMMI; Inicial, Repetible, Definido, Medido, Optimizado
Gerente de Proyecto Consultora MAINT	De mi experiencia existen 5 niveles de madurez dentro de la adopción: Inicial, Repetible, Definido, Medido y Optimizado. Para determinar el nivel de madurez se deben evaluar los siguientes aspectos: Organización (Cambio de cultura), Equipos (ruptura de silos), Construcción (automatización), Entrega continua (automatización), Gestión del ciclo de vida (trazabilidad, planificación, priorización, monitorización), testing (automatización), Gestión de datos y seguridad (automatización)
Consultor Xeridia (España)	Considera no existe un modelo consensuado

Pregunta 12	En el enfoque organizativo, ¿cuáles serían los principales aspectos para considerar para la adopción de DevOps? ¿Cómo deberían estar configurados los equipos y sus roles?
Arquitecto Certificado Devops Consultora MAINT	El equipo debe de estar conformado por todos los roles necesarios para poder entregar valor desde que negocio transmite la Idea al equipo hasta que esté la pone en producción. Esta es la premisa principal que se podrá conseguir en cierto grado dependiendo de la organización. Para la configuración del Equipo, la propuesta inicial es el modelo descrito por Spotify, pero esta no es una solución prescriptiva ya que debe de evolucionar según el contexto de la organización y aunque existen muchos detractores, es al día de hoy una forma de comenzar
Gerente de Proyecto Consultora MAINT	La conformación de un equipo multifuncional, en el que se considere los roles de desarrollo y operaciones como lo indica su propio nombre DevOps. Este equipo debe considerar los roles de desarrolladores, calidad, seguridad, testers, operaciones (servidores, bases de datos, infraestructura), producción. Debe ser autosuficiente y será el responsable de todo el ciclo de vida de la aplicación, no sólo por una parte de él, para esto lo principal es la ruptura del trabajo en silos y de manera tradicional, y llevarlo a un trabajo en equipo con un marco ágil de trabajo.
Consultor Xeridia (España)	Hay que conseguir construir un equipo multifuncional alineado con el Value Stream y que sea capaz de entregar valor end to end. Se podrían tener otros equipos que dan soporte a este equipo principal, pero el fundamental es ese. El libro Team Topologies es una buena referencia para construir estos equipos y sus interacciones.

Pregunta 14	¿Cuáles son los KPIs para medir el éxito del cambio cultural?
Arquitecto Certificado Devops Consultora MAINT	La cultura es algo difícil de medir mediante una cifra ya que esta sujeta a la percepción de la persona que la observa, es por eso es más útil emplear OKRs, ya que estos buscan medir si algo se ha cumplido o no y los aspectos que se deben de medir pueden variar según el contexto, pero unos básicos que se pueden analizar es que la cultura sea:
	Centrada en las necesidades del cliente
	Multifuncional (sin silos)
	Pensamiento en equipo ("nosotros")
	Orientada a Resultados
	Aceptar nuevas ideas
	Colaboración
	Proactividad
Los problemas y los errores son una oportunidad de aprender	
Gerente de Proyecto Consultora MAINT	Entre los KPI's importantes están:
	Tiempo de respuesta a cambios solicitados
	Frecuencia de despliegues
	Automatización de casos de prueba
	Tiempo promedio de recuperación
	Entre los KPI's que son explícitamente para medir el impacto cultural se pueden considerar:
	Felicidad de los equipos: mediante encuestas de satisfacción y otras métricas de participación.
	Eficiencia de reunión: reuniones breves, puntuales y en menor frecuencia en función de la alineación a través de Devops de equipos de desarrollo y operaciones.
Aprendizaje: mediante la exposición y enseñanza a otros miembros de la organización sobre lo aprendido a través de demostraciones, discusiones, almuerzos de aprendizaje. La frecuencia con la que se dan estos eventos constituye el KPI de aprendizaje.	
Consultor Xeridia (España)	Realmente en DevOps solo se deberían controlar 4 KPI's. La medida del cambio cultural se hace de manera indirecta a través de estos KPI:
	- Tiempo de ciclo
	- Frecuencia de despliegue
	- Tiempo medio de restauración del servicio
	- Porcentaje de fallos en los cambios.
En el libro Accelerate se establece una correlación entre estos KPI y una organización exitosa en la implementación de DevOps.	

Anexo 3:

Matriz de entrevistas

Fase de descubrimiento (Levantamiento de la información)

Posición funcional con quien se puede obtener la información	Objetivo	Documentación	Sugerencia de mecanismo para obtener la información
Nivel Funcional más alto dentro de Tecnología Como Gerencia de Tecnología	<ul style="list-style-type: none">-Conocer la estructura organizativa-Entendimiento del proceso de Desarrollo y Operación y personas claves- Información sobre servicio actual- Visión y necesidades en el ámbito de Integración de las áreas de Desarrollo y Operaciones orientado a la agilidad	<ul style="list-style-type: none">-Organigrama de área de TI	El mecanismo recomendado sería: <ul style="list-style-type: none">- Entrevista- Revisión de documentación existente

<p>Nivel Funcional más alto del área encargada del control de los proyectos</p> <p>Podría ser la Gerencia de la oficina de proyectos (PMO) si lo hubiere</p>	<ul style="list-style-type: none"> - Conocer la estructura de los distintos grupos de proyectos y personas claves - Conocimiento con cierto detalle del portafolio de proyectos tecnológicos en curso - Conocimiento de las distintas metodologías y procedimientos utilizados en el desarrollo y soporte de proyectos - Conocimiento de la gestión de la demanda del banco 	<p>-Documentación de procesos de gestión de proyectos disponibles</p>	<p>El mecanismo recomendado sería:</p> <ul style="list-style-type: none"> - Entrevista - Revisión de documentación existente
<p>Responsables de Operación</p>	<ul style="list-style-type: none"> -Conocer el proceso para la entrega de desarrollo a operación -Conocer el proceso de entrega al negocio 	<p>- Documentación de los procesos</p>	<p>El mecanismo recomendado sería:</p> <ul style="list-style-type: none"> - Entrevista - Revisión de documentación existente -Observación del proceso, si es factible

<p>Responsable con visión Global de Aplicaciones y su entrega al negocio Podría ser el Gerente de Desarrollo</p>	<p>-Conocer el inventario de aplicaciones detallando su arquitectura, tamaño, tecnología, etc -¿Cómo y cuáles son los requerimientos desde el negocio? -¿Cómo y cuáles son los requerimientos de las áreas de control interno como Seguridad, Riesgos, etc? -¿Cómo se reciben y cuáles son los requerimientos de las instituciones de control externo?</p>	<p>- Organización del departamento de desarrollo - Inventario de aplicaciones del banco</p>	<p>El mecanismo recomendado sería: - Entrevista - Revisión de documentación existente -Observación del proceso, si es factible</p>
<p>Responsable área o unidad de Agilidad (si lo hubiere)</p>	<p>-Conocer los frameworks de agilidad implementados -Conocer el nivel de madurez de agilidad implementado</p>	<p>Documentación disponible</p>	<p>El mecanismo recomendado sería: - Entrevista - Revisión de documentación existente</p>

			-Observación del proceso, si es factible
Responsable área de Seguridad	<ul style="list-style-type: none"> -Documentación de los procesos de seguridad e integración con los procesos de desarrollo -Requerimientos de entidades externas de control como la Super Intendencia de Bancos 	-Normas de seguridad requeridas por entidades de control externas y/o internas	<p>El mecanismo recomendado sería:</p> <ul style="list-style-type: none"> - Entrevista - Revisión de documentación existente
Responsable del área Calidad	<ul style="list-style-type: none"> -Conocer cómo se realiza el testing en sus diferentes ambientes -Conocer los procesos para la revisión de calidad del código 	Documentación disponible	<p>El mecanismo recomendado sería:</p> <ul style="list-style-type: none"> - Entrevista - Revisión de documentación existente

			-Observación del proceso, si es factible
Responsable de Infraestructura	<ul style="list-style-type: none"> -Conocimiento de la infraestructura actual del banco -Conocer el proceso para la gestión de la configuración 	-Documentación de la infraestructura y del proceso de configuración	<p>El mecanismo recomendado sería:</p> <ul style="list-style-type: none"> - Entrevista - Revisión de documentación existente -Observación del proceso, si es factible
Responsable Monitoreo	<ul style="list-style-type: none"> - Conocer los distintos sistemas de monitorización - Herramientas utilizadas - Detalle de la gestión de alertas y problemas 	<ul style="list-style-type: none"> -Documentación de los procesos de monitoreo -Informes generados 	<p>El mecanismo recomendado sería:</p> <ul style="list-style-type: none"> - Entrevista - Revisión de documentación

			<p>existente</p> <p>-Observación del proceso, si es factible</p>
Responsable Mesa de Servicio	<ul style="list-style-type: none"> - Conocimiento del sistema de gestión de incidencias - Soluciones y herramientas utilizadas 	<ul style="list-style-type: none"> - Informes generados - Evaluaciones de la mesa de servicio 	<p>El mecanismo recomendado sería:</p> <ul style="list-style-type: none"> - Entrevista - Revisión de documentación existente - Observación del proceso, si es factible

Anexo 4:

Matriz de valoración

Comunicación							
#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Existe alguna herramienta o procedimiento en donde las áreas involucradas en el ciclo del software pueden consultar los objetivos, requerimientos, métricas, etc.?	0					
2	¿Existen políticas de comunicación continua en los procesos o marcos de trabajo orientado hacia los distintos miembros del área de TI?	0					
3	¿Se cuenta con herramientas que permitan la comunicación entre distintas áreas?	0					
3.1	¿Estas herramientas son de uso estandarizado por todas las áreas o cada área usa herramientas de comunicación distintas?	0					

4	¿Se involucra a operaciones en el ciclo de vida del desarrollo del software desde el inicio?	0					
5	¿Se realizan reuniones en las que participen todos los involucrados en el desarrollo del software para revisar y debatir las ideas que ayuden para la creación del producto?	0					
6	¿Es frecuente la comunicación directa de miembros del equipo de desarrollo con miembros de equipos de operaciones o QA sin tener que pasar por una estructura jerárquica (aprobaciones de jefaturas)?	0					

Resultado	0
------------------	----------

Comunicación							
#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Existe alguna herramienta o procedimiento en donde las áreas involucradas en el ciclo del software pueden consultar los objetivos, requerimientos, métricas, etc.?	0					
2	¿Existen políticas de comunicación continua en los procesos o marcos de trabajo orientado hacia los distintos miembros del área de TI?	0					
3	¿Se cuenta con herramientas que permitan la comunicación entre distintas áreas?	0					
3.1	¿Estas herramientas son de uso estandarizado por todas las áreas o cada área usa herramientas de comunicación distintas?	0					
4	¿Se involucra a operaciones en el ciclo de vida del desarrollo del software desde el inicio?	0					
5	¿Se realizan reuniones en las que participen todos los involucrados en el desarrollo del software para revisar y debatir las ideas que ayuden para la creación del producto?	0					

6	¿Es frecuente la comunicación directa de miembros del equipo de desarrollo con miembros de equipos de operaciones o QA sin tener que pasar por una estructura jerárquica (aprobaciones de jefaturas)?	0					
---	---	---	--	--	--	--	--

Resultado	0
------------------	----------

Transparencia

#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Se involucran a miembros de áreas de seguridad y calidad del software desde las fases tempranas en el ciclo de desarrollo del software?	0					
2	¿El área de desarrollo comunica el trabajo que está realizando al área de operaciones?	0					
3	¿El área de operaciones comunica las novedades encontradas al área de desarrollo?	0					
4	¿Existe una herramienta donde queda registrada la información transmitida entre estas áreas?	0					

Resultado	0
------------------	----------

Colaboración							
#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿La planificación de trabajo de las áreas de desarrollo y operaciones comparten las mismas prioridades?	0					
2	¿Existen metas compartidas entre el departamento de desarrollo y operaciones?	0					
3	¿Se busca fomentar el reconocimiento del equipo en la consecución de los objetivos sobre el individual?						
4	¿Existe retroalimentación entre desarrollo y operaciones?	0					
5	¿Las evaluaciones y análisis de proyectos que resulten en éxito y/o fracaso son compartidos de igual manera por los departamentos de desarrollo y operaciones?	0					
6	¿Existen herramientas que apoyen la colaboración entre Dev y Ops?	0					
7	¿Existe una delegación del trabajo de forma equitativa entre los miembros de los diferentes equipos?						

Resultado	0
------------------	----------

Actitudes							
#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Se cuestiona de manera proactiva el trabajo de los diferentes miembros del equipo?	0					
2	¿Son escuchadas y tomadas en cuenta las opiniones de los diferentes miembros de los equipos de trabajo?	0					
3	Cuando se analiza una causa raíz con un problema, ¿se evita buscar culpas individuales?	0					
4	¿Se realiza el aprendizaje a través de los errores y la mejora continua?	0					

Resultado	0
------------------	----------

Prácticas Ágiles

#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Existen implementadas prácticas ágiles como Scrum, Kanban u otras?	0					
2	¿Existen equipos ágiles conformados y trabajando de forma sincronizada?	0					
3	¿Los perfiles especializados de Dev y Ops forman parte de los equipos ágiles?	0					
4	Las decisiones que requieren los equipos ágiles, ¿son tomadas por el propio equipo o por las gerencias/jefaturas?	0					

Resultado	0
------------------	----------

Gestión de la demanda y Requerimientos

#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Existen herramientas o procesos que permitan centralizar y gestionar los requerimientos, proyectos y/o resolución de incidentes que llegan a TI?	0					
2	¿Cada uno de estos requerimientos y/o proyectos tienen una valoración o alguna característica que permita realizar una priorización?	0					
3	¿Existe la integración de las herramientas donde se registra la demanda con herramientas de gestión de proyectos y/o requerimientos?	0					

Resultado	0
------------------	----------

Gestión del servicio

#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Hay integración de las herramientas de la mesa de servicios con las herramientas de gestión de demanda (requerimientos del negocio)?	0					
2	¿Las herramientas de monitoreo pueden generar incidencias de forma automática en las herramientas de gestión de la mesa de servicios?	0					

Resultado	0
------------------	----------

Gestión del conocimiento

#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Existe tecnología de software colaborativo para compartir información como Slack o Wiki interna?	0					
1.2	¿Estas herramientas son de uso estandarizado o generalizado en la organización?	0					
2	¿Existe una base del conocimiento compartido por las distintas áreas involucradas en el ciclo del software?	0					
3	¿En caso de presentarse problemas existen mentores con conocimiento que asistan o capaciten al personal?	0					
4	¿Existen equipos multifuncionales?	0					
5	¿Se realizan reuniones informales que estén enfocadas a impartir conocimiento, como ferias del conocimiento?	0					

Resultado	0
------------------	----------

Control de versiones

#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Están aplicadas tecnologías que permitan el control de versiones del código?	0					
2	¿Existen archivos de configuración versionados para diferentes máquinas y ambientes?	0					
3	¿Se usan procedimientos comunes y estandarizados para crear ramas y tags?	0					
4	El sistema permite regresar a una versión estable anterior en caso de presentarse errores con la versión más actual?	0					
5	¿El código fuente está a disposición de los equipos de Dev y Ops?	0					

Resultado	0
------------------	----------

Ambientes							
#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿El aprovisionamiento de ambientes para las distintas fases se realiza de forma manual o automatizada?	0					
2	¿El aprovisionamiento de ambientes está estandarizado?	0					
3	¿Los ambientes para el ciclo de vida de desarrollo de software son físico o virtuales?	0					
4	Para los ambientes virtuales, ¿se utiliza la infraestructura como código?	0					

Resultado	0
------------------	----------

Integración y entrega continua

#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Utilizan herramientas para la integración continua?	0					
2	Si se trabaja con scrum o prácticas ágiles, ¿se pasa a producción al final de cada sprint?	0					
2.1	¿Se trabaja con Sprint pequeños?	0					
3	Se automatizan las pruebas?	0					
4	¿Están definidas métricas que evalúen el desempeño de la construcción del código?	0					
5	¿Se realizan sesiones de lecciones aprendidas por cada entrega de software?	0					
6	¿Están definidos pipeline en el proceso de desarrollo de software?	0					

Resultado	0
------------------	----------

Despliegue Continuo

#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Existe una línea de despliegue para proyectos desarrollados con marcos ágiles?	0					
3	¿Existe una estandarización para los pases a los distinta ambientes?	0					
4	¿La ejecución de los pases a los distintos ambientes está automatizada?	0					
5	¿La organización ha estandarizado los sistemas operativos usados?	0					

Resultado	0
------------------	----------

Calidad y seguridad del código

#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Existen herramientas que ayuden a verificar la calidad del código?	0					
1.2	¿Estas herramientas son utilizadas en todo el ciclo de vida del software y en todos los ambientes?	0					
2	¿Se realiza el análisis de código en el ámbito de seguridad durante todo el ciclo de vida del producto?	0					
3	¿Hay herramientas automatizadas de calidad de código?	0					
4	¿Los perfiles de seguridad y calidad son integrados en los equipos ágiles?	0					

Resultado	0
------------------	----------

Pruebas automatizadas

#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Se hacen test unitarios del código?	0					
1.2	¿Se hacen test funcionales del código?	0					
2	¿Hay testing de integración automatizado?	0					
3	¿Hay testing de aceptación automatizado?	0					

Resultado	0
------------------	----------

Monitoreo							
#	Aspecto	Valor	1 - Inicial	2 - Repetible	3 - Definido	4 - Gestionado	5 - Optimizado
1	¿Se usan herramientas para el monitoreo de las aplicaciones?	0					
2	¿El monitoreo está integrado con otros sistemas como la mesa de servicios?	0					
3	¿El monitoreo abarca todos los posibles ambientes desde desarrollo hasta producción?	0					
4	¿Existen herramientas que proporcionen feedback a los desarrolladores?	0					

Resultado	0
------------------	----------

Anexo 5:

Tabla de valoración aspecto cultural

#	NIVEL	DESCRIPCIÓN
1	Inicial	<ul style="list-style-type: none"> -Organización trabaja en un esquema netamente funcional (silos) -No existe o poco trabajo colaborativo entre las áreas que participan en el ciclo de vida del software -Información solo es compartida entre miembros de las mismas áreas -Herramientas para la comunicación no existen o si existen solo son utilizadas dentro de cada área -Evaluaciones individuales -No existe un enfoque de distribución equitativa de trabajo -Cultura de culpas
2	Repetible	<ul style="list-style-type: none"> -Se establecen procesos de comunicación y se comparte información. -Existe colaboración entre áreas -Comunicación limitada entre miembros de Dev y Ops -Existen equipos trabajando bajo metodología ágil, más colaborativo -Se comparte el conocimiento de forma limitada -Herramientas de comunicación implementadas en las diferentes áreas -Existe evaluación grupal pero prevalecen las evaluaciones individuales -Con agilidad se empieza a utilizar una distribución de trabajo más equitativa (equipos ágiles) -Se buscan los errores a niveles de procesos más que a nivel de personas
3	Definido	<ul style="list-style-type: none"> -Dev y Ops se comunican de forma activa y directa sin pasar por una estructura jerárquica (aprobaciones) -Existe un canal de comunicación establecido entre Dev y Ops y se comparte habitualmente información -Existen equipos ágiles sincronizados que mantienen un ritmo de trabajo similar -Equipos multifuncionales (todos los perfiles requeridos) - Se comparte el conocimiento a través de herramientas entre Dev y Ops -Existen herramientas estandarizadas para la comunicación y colaboración entre áreas -Evaluaciones grupales -Mejora en la distribución equitativa (incremento de agilidad) -Se estandarizan los procesos de análisis de causa raíz de los problemas o errores suscitados
4	Gestionado	<ul style="list-style-type: none"> -Comunicación entre Dev y Ops es un proceso maduro (hábito) -Se comparte la información, Dev y Ops trabajan bajo el mismo fin / prioridades -Existe un proceso para fomentar el aprendizaje compartiendo conocimiento y habilidades entre los miembros de los diferentes equipos -Herramientas de comunicación y colaboración integradas con otros procesos y herramientas con mejores prácticas -Métricas grupales con el objetivo de entrega de valor al negocio -Distribución equitativa del trabajo -Análisis causa raíz establecido
5	Optimizado	<ul style="list-style-type: none"> -Existe un alto nivel de comunicación y colaboración entre los equipos y las áreas del ciclo de vida del software -Herramientas estandarizadas sirviendo a la automatización de los procesos -Mejora continua

Tabla de valoración aspecto prácticas y procesos

#	NIVEL	DESCRIPCIÓN
1	Inicial	<ul style="list-style-type: none"> -No se han implementado prácticas ágiles -Gestión de la demanda no gestionada de forma centralizada -Mesa de servicio no implementada o existe con funciones básicas -No se maneja una base del conocimiento o se maneja a nivel de cada grupo o área
2	Repetible	<ul style="list-style-type: none"> -Existen equipos ágiles que trabajan de forma independiente -Existe una gestión de la demanda centralizada y con priorización de proyectos -Existe una mesa de servicio con buenas prácticas implementadas -Existe una base de conocimientos con registros de información que se comparte entre áreas -Herramientas para compartir el conocimiento -Mínima trazabilidad
3	Definido	<ul style="list-style-type: none"> -Equipos ágiles organizados en cadenas de valor -Gestión de la demanda integrada a las herramientas de proyectos y mesa de servicio -Equipos multifuncionales (todos los perfiles requeridos) -Existe herramienta para la gestión de la base del conocimiento estandarizada -Integración de mesa de servicios con herramientas de gestión de proyectos y/o requerimientos
4	Gestionado	<ul style="list-style-type: none"> -Varias cadenas de valor compuestas por varios equipos ágiles -Proceso de gestión de la demanda en control (medido) -Base de conocimiento apoya la toma de decisiones -Equipos ágiles empoderados -Existen mentores para apoyar al equipo en la transmisión del conocimiento -Se agiliza el aprendizaje a través de reuniones informales de grupos ágiles -Monitoreo genera alertas automáticas en mesa de servicios y existe trazabilidad
5	Optimizado	<ul style="list-style-type: none"> -Mejora de tiempo de time-to-market -Mejora de tiempo de servicio -Mejora continua

Tabla de valoración aspecto tecnología y automatización

#	NIVEL	DESCRIPCIÓN
1	Inicial	<ul style="list-style-type: none"> -No existe herramienta de control de versiones -Integración del código manual -Aprovisionamiento y creación de entornos manual en entornos físicos y/o virtuales -Pruebas manuales una vez concluido el desarrollo -Despliegue manual al final del ciclo de desarrollo del software -Análisis de calidad y seguridad del código no se aplica durante el desarrollo del software -Monitoreo sin integración con otras herramientas -No existe o es muy eventual la revisión del desempeño de los proyectos
2	Repetible	<ul style="list-style-type: none"> -Control de versiones y generación de branches para desarrollo en paralelo -Integración del código automatizado -Creación y aprovisionamiento de ambientes manual en entornos físicos y/o virtuales, pero ya existen procesos definidos -Entrega y despliegue automático de software en ciertos entornos -No todos los sprint generan entregas a producción -Frecuencia de entrega de código baja -Pruebas unitarias automatizadas -Análisis de calidad y seguridad no se aplica durante el desarrollo del software, existen herramientas de revisión de código estático dinámico -Monitoreo sin integración con otras herramientas, proceso definido -Se define o establece un proceso de evaluación de los proyectos de desarrollo (retrospectiva)
3	Definido	<ul style="list-style-type: none"> -Control de versiones con mejores prácticas -Entrega automática de código en sprint cortos (más frecuente) -Se establecen métricas para evaluar el desempeño de la construcción del código -Mejoras al proceso de integración continua -Definición de pipelines para el proceso de desarrollo -Pruebas automatizadas (unitaria, funcionales, aceptación, integración) -Generación de infraestructura de ambientes automática, carga de software base manual, ya sea en entornos físicos o virtuales -Se incrementa el aprovisionamiento automático en más entornos, se estandariza el proceso de aprovisionamiento y generación de entornos -Análisis de calidad y seguridad integrado en el proceso del ciclo de vida del desarrollo -Monitoreo integrado con herramienta de mesa de servicios -Sistemas operativos estandarizados -Entrega continua y despliegue continuo se implementa progresivamente en nuevos entornos -Se define como estándar proceso de evaluación de los proyectos
4	Gestionado	<ul style="list-style-type: none"> -Control de versiones en control -Cada Srint genera entrega de valor al negocio -Integración continua completamente automatizado e integrado con otras herramientas -Métrica relacionadas a la construcción del código, lecciones aprendidas completamente establecidas y entregando información para análisis -Integración de pipeline -Ambientes generados y aprovisionados automáticamente en ambientes físicos, virtuales y/o en la nube -Se definen métricas para la calidad de software (pruebas, seguridad, calidad del código) -Monitoreo genera tickets de forma automática en la mesa de servicios -Monitoreo se extiende a todos los ambientes dando feedback en las diferentes etapas del proceso -Entrega continua y despliegue continuo en todos los entornos implementados
5	Optimizado	<ul style="list-style-type: none"> -Mejora continua del procesos -Rápida retroalimentación

Matriz de Madurez DevOps

Nivel	Cultura Organizacional	Gente(equipo)	Prácticas y Procesos	Construcción e Integración continua	Entrega continua/Ambientes	Testing
5	<ul style="list-style-type: none"> -Cultura de colaboración perfectamente adoptada -Dev y Ops comparten metas y prioridades -Mejora Continua 	<ul style="list-style-type: none"> -Mejora continua -Validación de niveles de satisfacción de los equipos 	<ul style="list-style-type: none"> -Mejora Continua -Mejora de tiempos de time to market -Mejora de tiempos de servicio 	<ul style="list-style-type: none"> -Mejora continua del proceso -Rápida retroalimentación 	<ul style="list-style-type: none"> -Mejora continua -Análisis continuo del tiempo de generación de ambientes y No. de defectos 	<ul style="list-style-type: none"> -Test totalmente automatizado -Número de errores tendientes a cero
4	<ul style="list-style-type: none"> -Existen hábitos de comunicación y colaboración (cultura DevOps) -Formación de comunidades de práctica -Todas las áreas de TI trabajan en equipos con enfoque en la entrega de valor al negocio 	<ul style="list-style-type: none"> -Colaboración de equipos -Equipos son autónomos y empoderados 	<ul style="list-style-type: none"> -Creación de varias cadenas de valor -Base del conocimiento implementada -Generación automática de incidencias en la mesa de servicio -Trazabilidad establecida -Monitorización integral de aplicaciones/servicios -Recolección de información para KPI's -Procesos estandarizados 	<ul style="list-style-type: none"> -Métricas relacionadas a la construcción del código -Lecciones aprendidas -Procesos mejorados -Integración de pipelines 	<ul style="list-style-type: none"> -Despliegue automático de todo lo liberado -Entornos generados y aprovisionados automáticamente -Cero tiempos de espera para el deployment 	<ul style="list-style-type: none"> -Métricas de calidad establecidas
3	<ul style="list-style-type: none"> -Eliminación de silos 	<ul style="list-style-type: none"> -Varios equipos multifuncionales por cadena de valor -Comunicación integrada -Se comparte conocimiento a través de herramientas 	<ul style="list-style-type: none"> -Equipos sincronizados, ritmo de trabajo similar, organizados como cadena de valor -Se registra información como datos organizados o estructurados -Integración de la mesa de servicio con otras herramientas de gestión de proyectos y requerimientos Definición de KPI's 	<ul style="list-style-type: none"> -Integración continua -Test integrado, cada código terminado es testeado y agregado al código raíz -Existen prácticas de revisión de código para calidad y seguridad -Configuración de software integrado a control de versiones -Creación de pipelines 	<ul style="list-style-type: none"> -Entrega automática de código en sprint cortos, pero más frecuentes -Se extiende el aprovisionamiento automático a más entornos -Inicio de estandarización de sistemas operativos -Herramientas para gestionar y almacenar los entregables a ser liberados en los distintos ambientes 	<ul style="list-style-type: none"> -Testing automatizado

2	<ul style="list-style-type: none"> -Inicio del cambio cultural Cultura ágil (colaborativa) 	<ul style="list-style-type: none"> -Colaboración entre Dev y Ops más ágil -Se utilizan herramientas para la comunicación y colaboración 	<ul style="list-style-type: none"> -Equipos independientes haciendo uso de prácticas ágiles -Base de conocimiento al menos con registro de datos como observaciones, hechos -Mesa de servicios con buenas prácticas -Existe una mínima trazabilidad en el ciclo de vida del software -Gestión de la demanda centralizada -Herramientas para la gestión de proyectos ágiles 	<ul style="list-style-type: none"> -Existe configuración del software -Control de versiones y generación de branches para desarrollo en paralelo 	<ul style="list-style-type: none"> -Proceso definido para la entrega automática en ciertos entornos 	<ul style="list-style-type: none"> -Testing unitario automatizado -Testing integrado en equipos (no hay fase de test)
1	<ul style="list-style-type: none"> -Existen Silos -Herramientas no estandarizadas -Cultura no colaborativa: objetivos, prioridades y métricas distintas entre Dev y Ops -Se premia o evalúa negativamente el trabajo individual más que el grupal 	<ul style="list-style-type: none"> -Comunicación limitada dentro de cada área -No se comparte el conocimiento -No se trabaja en equipo 	<ul style="list-style-type: none"> -Gestión de proyectos tradicional, (sin prácticas ágiles) -No existe base de conocimiento implementada a nivel de TI -Existe una mesa de servicio con funciones básicas -No existe un proceso claro o centralizado de la gestión de la demanda y/o requerimientos 	<ul style="list-style-type: none"> -Integración de código manual -No existe herramienta de control de versiones -No se aplican las prácticas de Seguridad y Calidad del código dentro del ciclo de desarrollo 	<ul style="list-style-type: none"> -No existen entregas continuas de software -Dependiente de disponibilidad de entornos 	<ul style="list-style-type: none"> -Pruebas manuales después de desarrollo -Requiere personal específico y tiempos para pruebas



**Presidencia
de la República
del Ecuador**



**Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes**



SENESCYT

Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

DECLARACION Y AUTORIZACION

Yo, **Terán Villafuerte Karen Denisse**, con C.C: # 0930838271 autor/a del trabajo de titulación: **Análisis y Formulación de estrategia general para preparar la adopción de la cultura DevOps en matrices bancarias de la ciudad de Guayaquil** previo a la obtención del título de **Ingeniero en Sistemas Computacionales** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 10 de septiembre de 2020

f. 

Terán Villafuerte Karen Denisse

C.C: 0930838271

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TEMA Y SUBTEMA:	Análisis y Formulación de estrategia general para preparar la adopción de la cultura DevOps en matrices bancarias de la ciudad de Guayaquil		
AUTOR(ES)	Karen Denisse Terán Villafuerte		
REVISOR(ES)/TUTOR(ES)	César Adriano Salazar Tovar		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Ingeniería		
CARRERA:	Ingeniería en Sistemas computacionales		
TÍTULO OBTENIDO:	Ingeniero en Sistemas computacionales		
FECHA DE PUBLICACIÓN:	10 de septiembre de 2020	No. DE PÁGINAS:	137
ÁREAS TEMÁTICAS:	Matrices Bancarias de Guayaquil, Cultura DevOps		
PALABRAS CLAVES/KEYWORDS:	DevOps, Instituciones bancaria de Guayaquil, nivel de madurez, CMMI, Plan de acción general		
RESUMEN/ABSTRACT:	<p>El presente trabajo de titulación contiene unas guías generales a ser usadas como referencia para preparar la adopción de la cultura DevOps en bancos con casa matriz en la ciudad de Guayaquil. La investigación utilizada fue de tipo exploratoria, ya que se hizo uso de encuestas a funcionarios de bancos para conocer su estado actual frente al tema de creación y entrega de software a sus clientes, y se realizaron entrevistas a consultores (dos nacionales y uno internacional) para que, en base a su experiencia, obtener las pautas y recomendaciones para la formulación de las guías. Las guías cuentan con 3 matrices y un road map general, iniciando con una matriz guía de entrevistas para el levantamiento de información las cuales contienen: roles/áreas a entrevistar, sugerencia de documentación a obtener, sugerencias de mecanismos para obtener la información; para la evaluación y cálculo del nivel de madurez de los bancos, se formuló la matriz de valoración la cual contiene una serie de preguntas para evaluar los 3 aspectos principales de DevOps: cultura y gente, prácticas y procesos, y tecnología y automatización, así como una matriz general de madurez DevOps, ambas matrices se crearon usando los 5 niveles del enfoque CMMI recomendado por los consultores entrevistados e investigación de diferentes fuentes de consultora como MediaTek Consulting Hub que han puesto a disposición su trabajo. Para el road map general se propone cinco fases y en cada una de ellas se seguirán los 3 aspectos evaluados de la matriz de madurez. Finalmente se propone una matriz de herramientas recomendadas por DevOps las cuales fueron clasificadas acorde a las fases del ciclo de vida de esta cultura, conteniendo costos, valoraciones y características clave de cada herramienta propuesta.</p>		
ADJUNTO PDF:	SI	NO	
CONTACTO CON AUTORES:	Teléfono: 593-939'793624	E-mail: karen.terun01@cu.ucsg.edu.ec	
CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE):	Nombre: Ing. Edison José Toala Quimí		
	Teléfono: +593 99-097-6776		
	E-mail: edison.toala@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			