Actualización de los mensajes de Log en uno de los módulos internos de Bancolombia.

Sebastian Gallego Pulgarín

Reporte de prácticas académicas para optar al título de Ingeniero de Sistemas

Tutor
Jaime Humberto Fonseca Espinal, Especialista en Ciencias Electronicas e Informaticas

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería de Sistemas
Medellín, Antioquia, Colombia
2022

Dedicatoria

A Dios por permitirme alcanzar este objetivo. A mis padres que con esfuerzo me ayudaron a lograr un sueño. A mi pareja por ser una gran motivación, por la fuerza brindada y la paciencia durante todo el proceso. Finalmente quiero dedicar este logro a toda mi familia, amigos, profesores y demás involucrados en cada una de las etapas que este proceso conlleva.

Agradecimientos

Infinitas gracias a todas y cada una de las personas que de una u otra forma fueron parte de este proceso, cada grano de arena aportado por ustedes constituye a este logro.

Tabla de contenido

Lista de figuras

Siglas, acrónimos y abreviaturas

TI	Tecnología de la Información
DevOps	Acronimo en ingles de Development y Operations
E2E	End To End
QA.	Quality Assurance
DEV	Development
CI	Continuous Integration
CD	Continuous Deployment

Resumen

El presente documento da a conocer la realización del proyecto asignado por Bancolombia en la etapa de prácticas académicas correspondidas entre Diciembre de 2021 y Junio de 2022, la forma en cómo se abordó y los resultados obtenidos. Se buscó mejorar los mensajes de log que tiene implementado uno de sus módulos internos por medio de la librería Log4j.

Para la realización del proyecto se rastreó y analizó la documentación propia que posee el banco sobre el módulo a intervenir y los criterios que se deben cumplir para cualquier desarrollo de software dentro de la organización, por lo cual se definen 6 objetivos específicos, que se cumplieron en su totalidad y como fruto final del presente trabajo, se obtuvo una mejora en cuanto a los mensajes de log, los cuales permiten identificar de manera oportuna la situación en la que se encuentra el módulo en cada uno de sus procesos.

Palabras clave: log, mensajes de error, log4j

Abstract

This document discloses the implementation of the project assigned by Bancolombia in the corresponding academic internship stage between December 2021 and June 2022, the way in which it was approached and the results obtained. This project was looking to improve the log messages that one of its internal modules has implemented through the Log4j library.

To carry out the project, the bank's own documentation on the module to be intervened and the criteria that must be met for any software development within the organization were tracked and analyzed, for which 6 specific objectives were defined which were met in its entirety and as a final result of this work, an improvement was obtained in terms of log messages, which allow the situation in which the module is in each of its processes to be identified in a timely manner.

.

*Keywords***:** log, error messages, log4j

Introducción

Bancolombia presenta una necesidad con respecto a los logs generados por uno de sus módulos internos. Esta necesidad surge por la poca claridad que existe en los mensajes de logs que se generan durante las operaciones diarias, especialmente en los momentos de fallas o errores. Como consecuencia se dificulta conocer e identificar los diferentes escenarios o motivos por los que pasó o falló el módulo, retrasando pues, el actuar del equipo de trabajo frente a los incidentes que se generan.

El proyecto *Actualización de los mensajes de log en uno de los módulos internos de Bancolombia*, pretende mostrar el trabajo realizado para cumplir con el proyecto planteado durante el periodo de prácticas correspondiente a las fechas entre Diciembre de 2021 y Junio de 2022 en la entidad financiera Bancolombia.

Se inició con una investigación en la entidad financiera, específicamente en el área de soporte de TI de tesorería, se buscaba identificar la causa raíz que causó que surgiera la necesidad de revisar los mensajes de logs con los que operaba en su momento uno de sus módulos internos.

Posteriormente se analizó la situación actual del módulo a intervenir, se buscó conocer la tecnología en la que estaba desarrollado, su versión, sus dependencias, insumos y documentación existente acerca del funcionamiento que debe tener el módulo.

Además, se tuvo en cuenta el tiempo promedio que le tomaba al área de soporte de TI de tesorería solucionar incidentes que se presentaban en el módulo con los mensajes de logs que contaba en su momento.

Con la información recopilada en los puntos anteriores, fue posible evidenciar que realmente existía una problemática a la hora de afrontar incidentes del módulo con los mensajes de logs con los que se contaba en su momento, ya que estos eran poco específicos, lo cual

dificultaba brindar una solución oportuna ante cualquier incidente que pudiera surgir con el módulo

El contar con buenos mensajes de log, que sean claros y específicos dentro de cualquier sistema de software es primordial para poder brindar un mantenimiento óptimo, ya que, nos ahorra tiempo al no tener que recrear el mismo escenario de error, y además, podemos conocer de primera mano con qué información, en qué lugar y demás datos claves, fue que sucedió alguna falla, incluso, automatizar alertas o soluciones automáticas.

El siguiente trabajo, puede funcionar como motivación para aquellos profesionales u organizaciones que deseen conocer la importancia de tener implementado un buen sistema de log dentro de sus aplicaciones, ya que disminuye el tiempo de respuesta ante escenarios no deseados, permite utilizar sus logs como documentación para validar el correcto funcionamiento de la aplicación tras alguna modificación o actualización, siendo esto, un ahorro en económico.

La motivación para realizar este proyecto surge de la necesidad de brindar un servicio de calidad, con una mayor disponibilidad para los usuarios y disminuir el tiempo que toma abordar un incidente en producción, evitando así afectación a los usuarios del banco y retrasos con la generación de reportes e insumos para el inicio de labores de las demás áreas y compañías dependientes de la información que este módulo procesa.

Es por esto que como objetivo principal de este proyecto se busca modificar los mensajes de logs en uno de sus módulos internos para que sean más descriptivos, y así mejorar el tiempo de entendimiento y respuesta que se le brinda a cada incidente del módulo.

Con el primer objetivo específico se buscó recopilar la información clave que utiliza el módulo en sus diferentes procesos y así conocer sus dependencias

Con el segundo objetivo específico se identificó qué logs se debían modificar, su razón y su nuevo string con el que se reemplazaría, a este nuevo string se le evaluaba la información que

debía tener, fuera modificando las áreas a las que hacía referencia ó añadiendo información clave, como por ejemplo el insumo con el que ocurrió el fallo.

Nuestro tercer objetivo específico buscó validar que nuestro desarrollo no afecta la funcionalidad actual del módulo y que además, ante situaciones de falla, el nuevo mensaje de log se imprime correctamente. Adicionalmente desarrollar pruebas unitarias para cada línea de código modificada o agregada durante el desarrollo, aunque modificar los mensajes de log no interviene en la lógica de la aplicación, estas pruebas se deben realizar principalmente por las políticas dictadas por el área de DevOps y Pruebas Continuas cumpliendo así con el porcentaje de cobertura mínimo en pruebas unitarias que debe tener los desarrollos de software dentro de la entidad y de este modo lograr compilar nuestra aplicación.

Para nuestro cuarto objetivo específico se verificó en los ambientes de QA y certificación que todo continuaba funcionando correctamente, para ello aprovechamos la arquitectura en la nube que utiliza el área de soporte de TI de tesorería para el módulo que se intervino, llevamos a cabo el despliegue del desarrollo en la nube y se ejecutó un plan de certificación que dispone el área, a su vez que se realizaron pruebas E2E del módulo para verificar que todo continuaba funcionando de manera correcta.

Como quinto objetivo específico debimos cumplir con el porcentaje mínimo que exigían las diferentes herramientas de monitoreo que dispone el Banco, a su vez que se cumplía con el flujo de trabajo establecido para las aplicaciones con arquitectura en la nube y para dicho módulo en general, y así logramos encaminar el desarrollo para realizar el paso al ambiente de producción

Como último objetivo específico tuvimos el despliegue del desarrollo en el ambiente productivo, aquí continuamos utilizando la infraestructura en nube que se dispone para el módulo, velamos por el correcto despliegue, dimos soporte y estabilización al cambio realizado.

Sumado a lo anterior y para brindar mayor contexto al lector, se presentará en el capítulo 1 el planteamiento del problema, donde se aborda el porqué del proyecto, de donde surge y la importancia que este toma. En el capítulo 2 se encontrarán los objetivos anteriormente mencionados, que son el foco principal del proyecto a desarrollar. En el capítulo 3 se podrá encontrar información relevante de diversas fuentes que abordan el tema principal de lo que trata el proyecto. Luego en el capítulo 4 se expondrá la ruta metodológica que se utiliza para abordar el proyecto y lograr alcanzar los objetivos planteados. Los resultados esperados se encuentran en el capítulo 5, allí se encuentra organizado con el mismo hilo conductor de los objetivos específicos con el fin de mostrar los resultados obtenidos de una manera organizada. Finalmente en el capítulo 6 se evidencian las conclusiones obtenidas durante el desarrollo del proyecto de la actualización de los mensajes de log en uno de los módulos internos de Bancolombia.

Adicionalmente vale aclarar que a lo largo de este documento no se dará detalle que pueda considerarse información sensible o reservada de Bancolombia, por lo cual, se tratarán los temas de manera superficial para no comprometer la integridad de la organización.

1. Planteamiento del problema

Para empezar es imprescindible mencionar la importancia que tiene la disponibilidad de servicios en una entidad financiera tan grande como lo es Bancolombia, debido a que cada minuto que alguno de sus servicios se encuentra inactivo se traduce en una gran pérdida económica por operaciones que no se logran concretar, llegando hasta multas por incumplimiento con las normativas que la rigen.

Es por esto que Bancolombia, siendo la entidad financiera más grande de Colombia, requiere efectuar medidas sobre los mensajes de logs de uno de sus módulos internos, ya que, tomando en cuenta el tiempo que toma solucionar incidentes de este módulo, según el histórico de incidentes que se tiene, se identificó que el tiempo promedio de respuesta era algo elevado, estando en varias ocasiones encima de lo tolerable, causando así grandes pérdidas económicas y retrasos en las operaciones diarias.

Una vez identificadas las diferentes causas de los incidentes que han ocurrido con el módulo, la solución y los problemas en el camino para llegar a ella, se verificó la viabilidad de plantear un proyecto donde se obtenga como resultado una disminución en el tiempo que toma la solución de incidentes en los ambientes productivos.

Es por ello que se planteó implementar una mejora en los mensajes de logs del módulo con el fin de lograr que estos mensajes sean mucho más dicientes y se logre dar respuesta a los incidentes en un tiempo óptimo.

2. Objetivos

2.1 Objetivo general

Modificar los mensajes de logs de uno de sus módulos internos para que sean más descriptivos, y así mejorar el tiempo de entendimiento y respuesta que se le brinda a cada incidente del módulo.

2.2 Objetivos específicos

- Identificar los actores e insumos que intervienen en cada uno de los escenarios y operaciones que realiza el módulo.
- Adecuar los String de los logs con la línea de código donde ocurrió el fallo, el trade, la contraparte, el nombre del insumo fuerte con el error y/o demás detalles útiles que se puedan rescatar para cada log en particular.
- Realizar las pruebas unitarias requeridas para el cubrimiento de las nuevas líneas de código tras la modificación de los mensajes de log.
- Verificar el correcto funcionamiento del módulo en ambientes de certificación una vez hecho los cambios de los mensajes de log.
- Cumplir con los criterios y políticas implementadas por DevOps en los flujos de trabajo de las aplicaciones de software.
- Desplegar el módulo con sus correspondientes cambios en el ambiente de producción.

3. Marco teórico

Un trace log lo podemos definir según [3] como el registro de eventos que los sistemas de software producen durante el tiempo de ejecución. Son utilizados para comprender de mejor manera su comportamiento dinámico, almacenando información relacionada con el flujo de trabajo que se tiene en cada momento, como por ejemplo: Quien actúa con el sistema, cuándo, dónde, con qué recurso, por qué ocurre un fallo, etc.

Existen diversas formas de implementar estos logs, esto dependerá del tipo de aplicación y las tecnologías con las que se encuentre construida. Una forma de trabajar con estos logs es utilizando la librería llamada Log4j. Esta librería fue publicada a principios del año 1996, pero ha traído diversas actualizaciones a lo largo de los años. Actualmente se encuentra en su versión 2.X siendo la librería referente para implementar manejo de logs en las aplicaciones JAVA que lo requieran [1].

Para utilizar esta librería en su versión 2.X se debe implementar un archivo de configuración que puede ser llamada log4j2.properties, en este archivo se indicará toda la configuración que se necesita para el correcto funcionamiento del servicio de logs. La configuración de este archivo dependerá de la forma en que se quiera visualizar, almacenar y el tipo de escritura que se vaya a utilizar. Algunos conceptos fundamentales son:

3.1 Niveles

La librería log4j permite implementar diferentes niveles de registro. Los niveles son configurables dentro del archivo de configuración de log4j2.properties, lo que permite seleccionar desde qué nivel de mensajes se tomarán en cuenta para el reporte de los logs. Los niveles por defecto que trae la librería son:

- **DEBUG:** Usado para escribir mensajes de depuración.
- **INFO:** Mensajes de estilo verbose. Puramente informativos de determinada acción.
- **WARN:** Para alertar sobre eventos de los que se requiere dejar constancia, pero que no afectan al funcionamiento de la aplicación.
- **ERROR:**Usado para los mensajes de eventos que afectan al programa, pero este puede continuar su funcionamiento.
- **FATAL:** Usado para errores críticos que impiden se siga ejecutando la aplicación.

A su vez se puede crear niveles personalizados para darles un uso particular según sea la necesidad.

3.2 Appenders

Los appenders hacen referencia al tipo de salida que se va implementar y a donde se van a dirigir todos los mensajes que se generan. La librería Log4j por defecto soporta los siguientes appenders:

- **File Appender/RollingFileAppender:** Es una salida dirigida a un fichero llamado <fichero>.log.
- **SocketAppender:** Es una salida hacia un servidor remoto.
- **SMTPAppender:** Se envía la salida hacia un correo electrónico.
- **JDBCAppender:** Se utiliza como salida una base de datos.
- **ConsoleAppender:** Se imprime todo el log por consola.

Dependiendo del tipo de Appender que vayamos a utilizar debemos agregar diferentes atributos a nuestro archivo de configuración Log4j2.properties. Una vez teniendo esto configurado correctamente, ya tendremos funcional nuestro sistema de logs.

4. Metodología

Se utilizó una metodología de trabajo ágil llamada SCRUM, según Kenneth Rubin podemos definirla como "Un marco de trabajo basado en un conjunto de valores, principios y prácticas que suministran los fundamentos para que cada organización le agregue su implementación única." Es una metodología ágil que se centra en la persona y en un conjunto de valores. Las personas forman equipos enfocados en el resultado y que trabajan de forma autodirigida. Una de sus principales características es la adaptación continua para una buena evolución del proyecto.

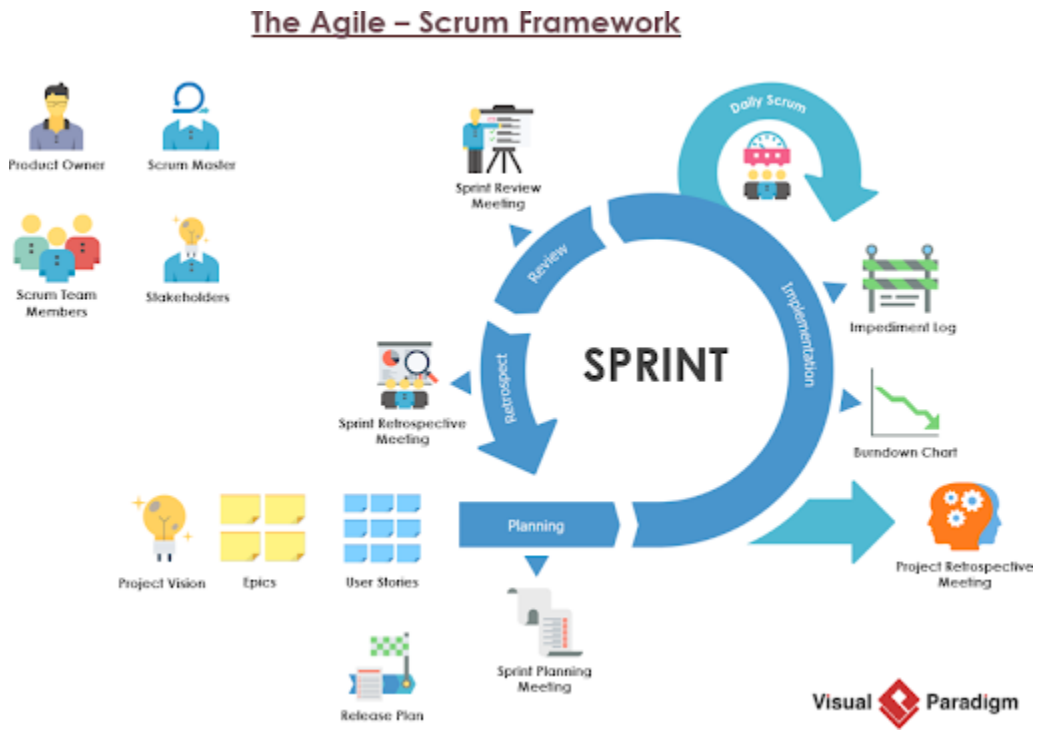

Figura 1. Metodología Scrum

Se llevaron a cabo capacitaciones, guías y reuniones de apoyo sobre el funcionamiento y adecuación en ambiente local del proyecto por parte de los ingenieros de Software encargados del módulo dentro del área. Dichas actividades se realizaron a demanda, sin exceder los 15 días calendario entre una actividad y otra.

Se realizó un previo análisis de la documentación que se tiene acerca del funcionamiento del módulo. Esto permitió identificar cada uno de los escenarios por los cuales el módulo trabaja, además de los insumos necesarios para su ejecución y su correspondiente archivo ejecutable que desencadena cada proceso interno del módulo.

Posteriormente se analizó, de manera independiente, cada uno de los escenarios que presenta el módulo con su correspondiente insumo (si lo requería), esto se realizó con el fin de conocer la estructura del insumo con el que trabaja cada escenario y el flujo de ejecución que este conlleva. Luego se realizó un Debug en un ambiente local del escenario en cuestión con el fin de identificar las clases que intervienen en la ejecución, a su vez, identificar todos los mensajes de Log actuales por los que pasaba el módulo. Una vez se reconoció cada String de Log, se procedió a analizar y proponer un nuevo String que tuviera información relevante del estado en el cual se encuentra el módulo y la razón por cual está fallando el proceso (si aplica).

Una vez modificados todos los Logs que se identificaron que debían ser intervenidos, se procedió a desplegar estos cambios en los ambientes intermedios (Desarrollo y certificación) que se tienen disponibles, allí se ejecutaron y finalizaron de manera exitosa todas y cada una de las pruebas que se dispusieron para verificar el correcto funcionamiento del módulo. Una vez se cumplió con lo anterior, se procedió a realizar el paso al ambiente de producción. Estos ambientes se encuentran en una arquitectura 100% Cloud.

5. Resultados

Como resultado final luego de haber corrido 26 ciclos de sprint se logró cumplir con los objetivos planteados para este proyecto en su totalidad. Cada uno de los objetivos planteados fue fundamental para alcanzar el éxito del proyecto.

Se ejecutó la metodología mencionada en el este documento, además de incluir pasos adicionales con el fin de cumplir con todas las políticas y requisitos dictados por el área de DevOps y Pruebas Continuas para los desarrollos de software dentro de la organización. Esto nos dió una línea temporal de las actividades requeridas para alcanzar los objetivos de manera exitosa cubriendo también estos nuevos requisitos.

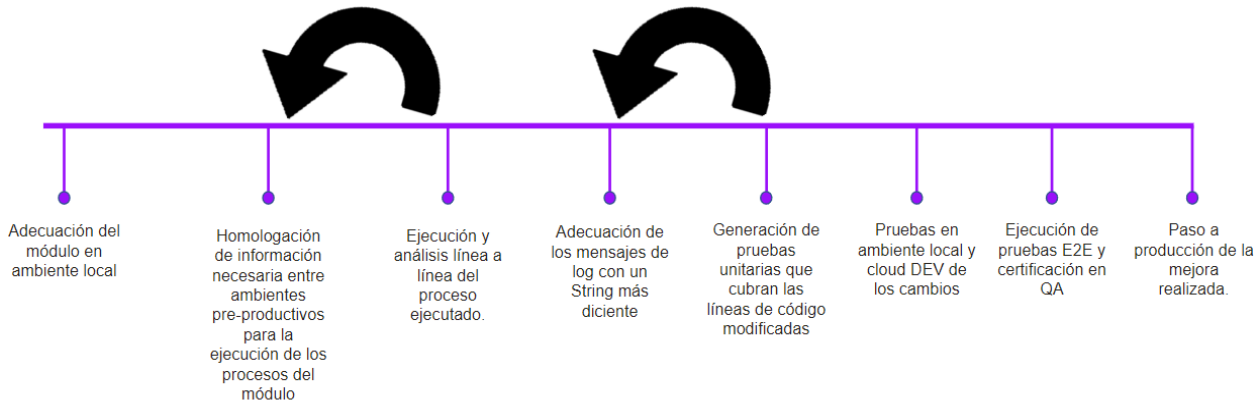

Figura 2. Ejecución de la metodología

Se comenzó configurando el ambiente local de desarrollo donde se ejecutaría el módulo correspondiente a intervenir, esto hace referencia a la instalación del lenguaje de programación con el que está construido el módulo, su versión, la correspondiente base de datos y herramientas adicionales que se requieren para una ejecución exitosa del módulo en el ambiente local.

Posteriormente fue necesario homologar cierta información entre los ambientes pre-productivos QA y DEV con el fin de corregir errores que surgían durante la ejecución del

módulo por falta de información crítica en las bases de datos de desarrollo e insumos con estructura de datos desactualizada.

Una vez verificado que la ejecución del proceso del módulo que se iba a trabajar finalizó correctamente, se procedió a realizar un debug linea a linea del escenario en cuestión identificando los mensajes de log que debían ser modificados, en nuestro caso se modificaron únicamente los mensajes de log de tipo error que tuvieran ciertas características que debían ser modificadas, dichas características fueron brindadas por el área de Soporte de TI, y/o que directamente fueran mensaje de logs poco dicientes con respecto a la situación actual en la cual se encontraba el módulo.. En caso de ocurrir un fallo durante la ejecución del módulo por falta de información en el ambiente de desarrollo, se ejecutaba nuevamente el paso anterior.

Luego de tener el contexto completo de lo que hace el módulo en cada uno de sus escenarios e identificar los logs necesarios a modificar, se planteó un nuevo String de log para reemplazar en cada situación encontrada. Adicionalmente, para cumplir con las políticas de DevOps y Pruebas Continuas para los despliegues automatizados CI/CD, se tuvo que realizar una prueba unitaria que pudiera verificar el correcto funcionamiento del cambio realizado. Para esta situación obtuvimos un cambio en más de 200 mensajes de log, siendo requeridas 101 pruebas unitarias, las cuales corresponden a más de 2.000 líneas de código.

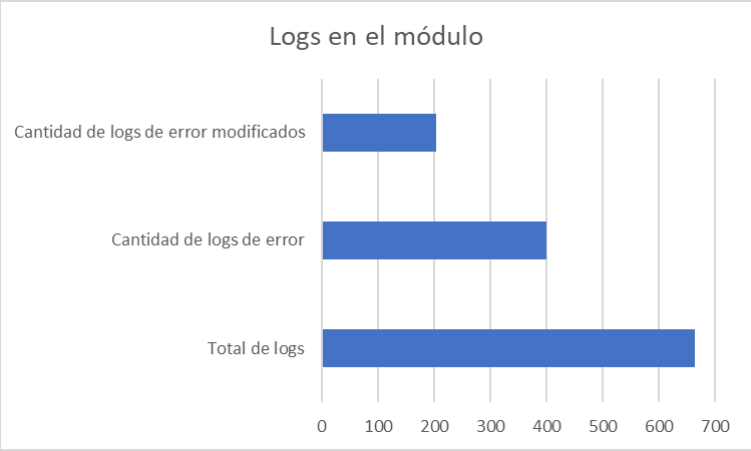

Figura 3. Reporte de logs, resultados.

Adicionalmente a lo anterior, la creación de dicha cantidad de pruebas unitarias, permitió aumentar en un 40% el código total del módulo cubierto por estas, lo cual permitirá validar y verificar que posibles cambios que se realicen a futuro en el código del módulo no impactan ni afectan el funcionamiento de la aplicación, incluso, algunas modificaciones que pueda requerir el código de este módulo se podrán realizar de una manera más ágil y segura, ya qué, no en todos los casos será necesario construir nuevas pruebas. Este porcentaje del código total del módulo se traduce en un 82% del código con logs de error cubierto por pruebas unitarias.

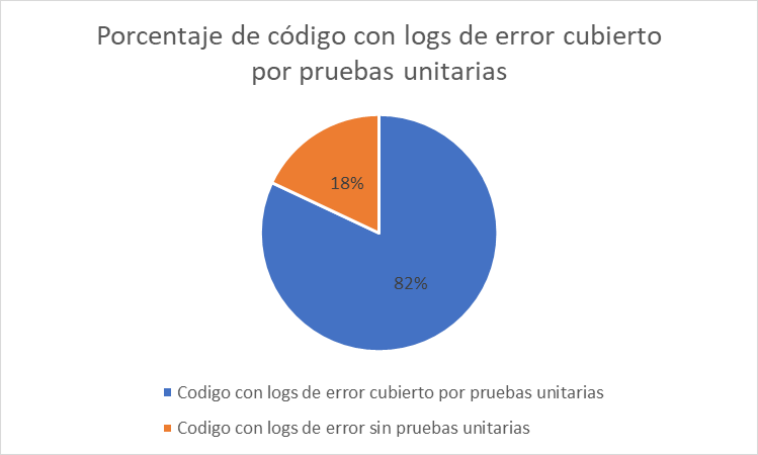

Figura 4. Código con logs de error cubierto por pruebas unitarias.

Una vez cubiertos los cambios realizados con sus correspondientes pruebas unitarias, se procedió a llevar el desarrollo al ambiente cloud dispuesto para este módulo, allí buscamos probar que todo lo realizado de manera local continuaba funcionando correctamente en la nube para posteriormente realizar la prueba de certificación en el ambiente de pruebas QA.

Este proceso de certificación, correspondió ya no sólo probar que el desarrollo funcionaba correctamente, sino que también, continuaba funcionando el módulo en su totalidad, esto quiere decir, funcionando correctamente con las conexiones a los demás módulos, rutas, bases de datos, etc. Para esto se ejecutó el plan de pruebas E2E que se encontraba ya planeado por los

encargados de la certificación en el área. Dichas pruebas finalizaron de manera exitosa, lo cual nos permitió realizar el paso a producción del desarrollo luego de generar la documentación necesaria para registrar los cambios realizados.

Cuando se tuvo la oportunidad de realizar el paso a producción, se realizó de manera exitosa a través de los pipelines del módulo para automatizar estos procesos, luego de unos minutos se visualizó que los pipeline finalizaron su ejecución de manera exitosa, y tras una validación manual de que los cambios realizados quedaron efectivamente arriba en el ambiente de producción, dábamos por finalizado el alcance del proyecto planteado.

Por otra parte, durante el desarrollo de este proyecto se presentaron varios inconvenientes de alta criticidad, por lo cual se tuvieron que suspender las actividades correspondientes a este proyecto y volcar todos los esfuerzos hacia los inconvenientes críticos que surgieron durante el tiempo establecido. Se presentaron varias vulnerabilidades de seguridad en las aplicaciones que debían ser solucionadas lo antes posible, estas soluciones fueron principalmente actualizar la versión de ciertas herramientas que utilizaban varias de las aplicaciones del área. Adicionalmente se tuvo que planear y ejecutar un plan de contingencia y solución hacia una problemática de constantes alertamientos que estaban llegando al área por retraso en algunas operaciones core.. Finalmente fue necesario brindar reuniones de asesoría a miembros de otras áreas internas al banco, el objetivo de estas asesorías era aclarar los productos que ofrece el área, validar la viabilidad de ciertas peticiones y explicar el funcionamiento de algunas herramientas propias del área

6. Conclusiones

- El marco de trabajo ágil SCRUM que se utiliza dentro de la compañía permitió una constante comunicación con el equipo, lo que permitió superar problemáticas del camino y optimizar el tiempo de acoplamiento a los procesos y políticas de la organización.

- Los mensajes de log son una herramienta fundamental que permite reducir tiempo y costo a la hora de afrontar cualquier incidente que se pueda presentar en las aplicaciones, ya que obtenemos información a tiempo real de lo que está sucediendo sin necesidad de buscar replicar un error en ambientes de desarrollo sin tener gran contexto de lo sucedido.

- Una buena documentación potencia la curva de aprendizaje sobre una herramienta y facilita la interacción con ella, como pueden ser posibles modificaciones.

- Las pruebas unitarias dentro de nuestras aplicaciones son fundamentales, ya que nos permiten validar que nuestros cambios realizados no afectan la funcionalidad en algún otro módulo del aplicativo.

7. Referencias

- Apache Software Foundation. Logging Service Log4j. Recuperado de https://logging.apache.org/log4j/2.x/
- B. Molina Montero, H. Vite Cevallos, and J. Dávila Cuesta, “Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,” Espirales Rev. Multidiscip. Investig., vol. 2, no. 17, pp. 114–121, 2018, [Online]. Available: http://revistaespirales.com/index.php/es/article/view/269/225.
- G. Rong, S. Gu, H. Zhang, D. Shao and W. Liu, "How Is Logging Practice Implemented in Open Source Software Projects? A Preliminary Exploration," 2018 25th Australasian Software Engineering Conference (ASWEC), 2018, pp. 171-180, doi: 10.1109/ASWEC.2018.00031.
- H. Li, H. Zhang, S. Wang and A.E.Hassan, "Studying the Practices of Logging Exception Stack Traces in Open-Source Software Projects," in IEEE Transactions on Software Engineering, doi: 10.1109/TSE.2021.3129688.
- Kemp, D., & Mollett, J. (2011, June). 5.4. 1 Would the real systems engineer please stand up?. In *INCOSE International Symposium* (Vol. 21, No. 1, pp. 610-628).
- M. Trigas, “Metodología Scrum,” 2012.
- Scrum permite el trabajo colaborativo entre equipos. Metodología agile. (n.d.). Retrieved June 19, 2019, from gefes website: https://gefes.es/metodologias/scrum/.