



**AUTOMATIZACIÓN DE PRUEBAS PARA EL ASEGURAMIENTO DE LA CALIDAD  
DE ADMINFO VSMART**

Aldodaiyan Romell Rendón Gutiérrez

Informe de práctica para optar al título de Ingeniero de Sistemas

Asesora

Deisy Loaiza Berrío, Ingeniera en Sistemas

Universidad de Antioquia  
Facultad de Ingeniería  
Ingeniería de Sistemas  
El Carmen de Viboral, Antioquia, Colombia  
2022

Cita	Rendón Gutiérrez [1]
<b>Referencia</b> Estilo IEEE (2020)	[1] A.R. Rendón Gutiérrez, “Automatización de pruebas para el aseguramiento de la calidad de Adminfo Vsmart”, Semestre de Industria, Ingeniería de Sistemas, Universidad de Antioquia, El Carmen de Viboral, Antioquia, Colombia, 2022.



Centro de Documentación Ingeniería (CENDOI)

**Repositorio Institucional:** <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - [www.udea.edu.co](http://www.udea.edu.co)

**Rector:** John Jairo Arboleda Céspedes

**Decano/Director:** Jesús Francisco Vargas Bonilla

**Jefe departamento:** Diego José Luis Botía Valderrama

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

## **Dedicatoria**

En memoria de mi tío Orson Uolmar quien desde que yo estaba muy pequeño me ánimo e inspiro con su ejemplo, me mantuvo soñando siempre con sus palabras y consejos cuando iba creciendo poco a poco, estas palabras quedaran ahí para mí siempre, él fue y será una gran inspiración para toda mi vida, a salir adelante, esforzarme cada vez más y ser un gran ser humano y profesional como él lo fue.

## **Agradecimientos**

A Dios, a la vida y a todo el universo por darme tantas Bendiciones y sabiduría, además por permitirme formarme en esta prestigiosa universidad, donde gracias al aprendizaje y al conocimiento impartido por los docentes, logré ingresar a Bancolombia a realizar mis prácticas; también a mi familia, mi madre que siempre me apoyó durante todo este proceso, a mi padre por sus consejos, a mi hermano por sus ánimos y palabras de aliento, finalmente a mi novia por estar ahí para mí incondicionalmente.

## TABLA DE CONTENIDO

RESUMEN.....	8
ABSTRACT .....	9
I. INTRODUCCIÓN .....	10
II. PLANTEAMIENTO DEL PROBLEMA.....	11
A. <i>Antecedentes</i> .....	11
III. JUSTIFICACIÓN.....	12
IV. OBJETIVOS.....	14
A. <i>Objetivo general</i> .....	14
B. <i>Objetivos específicos</i> .....	14
V. MARCO TEÓRICO .....	15
1. <i>Prueba manual</i> .....	15
2. <i>Pruebas automatizadas</i> .....	15
<i>Gherkin</i> .....	15
<i>Calidad del software</i> .....	16
<i>Selenium</i> .....	16
<i>Script</i> .....	16
VI. METODOLOGÍA .....	17
A. <i>Fase de capacitación</i> .....	17
B. <i>Fase de planificación y participación</i> .....	17
C. <i>Fase de ejecución</i> .....	17
VII. RESULTADOS .....	20
A. <i>Fase de planificación y participación</i> .....	20
B. <i>Fase de ejecución</i> .....	21
Diagrama de Bizagi.....	21

Levantamiento de riesgos .....	25
Plan de pruebas .....	26
Automatización .....	28
Grafana.....	31
Refactorización SonarQube .....	32
Resultado del proyecto.....	33
VIII. CONCLUSIONES Y RECOMENDACIONES .....	34
IX. REFERENCIAS .....	35

## LISTA DE FIGURAS

Fig. 1. Gherkin .....	15
Fig. 2. Cronograma de actividades propuesto .....	20
Fig. 3. Diagrama Bizagi de Adminfo .....	21
Fig. 4. Diagrama de los perfiles 16-22-30-68-179 .....	21
Fig. 5. Diagrama de Bizagi perfil 16.....	22
Fig. 6. Diagrama de Bizagi perfil 22 .....	22
Fig. 7. Diagrama de Bizagi perfil 30.....	23
Fig. 8. Diagrama de Bizagi perfil 68.....	23
Fig. 9. Diagrama de Bizagi perfil 179.....	24
Fig. 10. Diagrama de Bizagi Módulo de Mantenimiento.....	24
Fig. 11. Levantamiento de riesgos .....	25
Fig. 12. Esquema lenguaje Gherkin .....	28
Fig. 13. Perfiles para automatizar en lenguaje Gherkin .....	28
Fig. 14. Login Adminfo.....	28
Fig. 15. Login Adminfo Dado.....	29
Fig. 16. Login Adminfo Cuando .....	29
Fig. 17. Login Adminfo Entonces.....	29
Fig. 18. Login Adminfo Xpaths .....	29
Fig. 19. Sincronización repositorio local .....	30
Fig. 20. Configuración de Pipeline Build RM .....	30
Fig. 21. Tablero de Grafana (Antes) .....	31
Fig. 22. Tablero de Grafana (Después) .....	31
Fig. 23. Refactorización SonarQube(Antes) .....	32
Fig. 24. Refactorización SonarQube(Después).....	32
Fig. 25. Resultado Final proyecto .....	33

## SIGLAS, ACRÓNIMOS Y ABREVIATURAS

<b>RM</b>	Release Managment
<b>QA</b>	Quality Assurance
<b>PSR</b>	Grabación de Acciones de Usuario
<b>BRP</b>	Bienes Recibidos en Pago
<b>DSL</b>	Domain-Especific-Lenguaje

## RESUMEN

Bancolombia es un grupo financiero líder, con más de 146 años de experiencia que evoluciona para asumir los desafíos que trae un entorno en constante transformación. Es por esto que se visualizará la automatización de una aplicación llamada Adminfo Vsmart la cual permite la solución integral de la gestión de cobranza, pasando por todas las etapas, como: Cartera Administrativa, Cartera Pre-jurídica, Cartera Jurídica, Cartera Castigada y Bienes recibidos en forma de pago.

Inicialmente se realiza un estudio en el sitio de contenidos del área de tecnología sobre la metodología que se usa para el proceso de pruebas y patrones de diseño, luego se prosiguió con la formulación de un plan de pruebas de acuerdo con el método de la ruta crítica que está constituida por los perfiles 16-22-30-68-179 y el módulo de mantenimiento, esta ruta se construye a través de un diagrama de flujo construido en la aplicación Bizagi. Después de esto, se crea en lenguaje Gherkin, entre 1 y 3 transacciones por perfil, es decir, lo que deberá ejecutar cada perfil; igualmente, se produce el levantamiento de riesgos dando inicio al plan de pruebas, con el que se pretende mejorar la calidad del software. Finalmente, luego de haber avanzado en la ejecución de varios pasos y perfiles se puede comenzar a configurar el Release Managment (RM) y el tablero de Hygieia siguiendo las métricas del Banco, para aumentar los porcentajes de medición, por encima del 80% con la ejecución de la totalidad de las pruebas en el pipeline de Release Managment (RM).

***Palabras clave* — Cucumber, Serenity BDD, ScreenPlay, Selenium, Pipeline, Automatización.**

## ABSTRACT

Bancolombia is a leading financial group, with more than 146 years of experience that evolves to take on the challenges of a constantly changing environment. Therefore, we will visualize the automation of an application called Adminfo Vsmart, which allows the integral solution of the collection management, going through all the stages, such as: Administrative Portfolio, Pre-judicial Portfolio, Legal Portfolio, Write-off Portfolio and Assets received in the form of payment.

Initially, a previous study is made in the Bancolombia wiki about the methodology used for the testing process and design patterns, then it was continued with the formulation of a test plan according to the critical path method that is constituted by the profiles 16-22-30-68-179 and the maintenance module, this path is built through a flowchart built in the Bizagi application. After this, it is created in Gherkin language, between 1 and 3 transactions by profile, namely, what each profile will have to execute; likewise, the risk survey is produced starting the testing plan, with which it is intended to improve the quality of the software. Finally, after having advanced in the execution of several steps and profiles, it is possible to start configuring the RM and the Hygieia dashboard following the Bank's metrics, for increase the measurement percentages, above 80% with the execution of all the tests in the RM pipeline.

***Keywords*** — **Cucumber, Serenity BDD, ScreenPlay, Selenium, Pipeline, Automation.**

## I. INTRODUCCIÓN

El Grupo Bancolombia es un grupo financiero que tiene como propósito promover el desarrollo económico y sostenible para lograr el bienestar de todas las comunidades, este propósito está fundamentado en unos pilares [1]:

- Fortalecimiento de la competitividad del tejido productivo.
- Construcción de ciudad y comunidades más sostenibles.
- La fomentación de la inclusión financiera.
- El desarrollo de la cultura y el talento para la competitividad.

La organización fue fundada el 29 de enero de 1875, varios hechos marcan el camino que ha tenido el banco en los últimos 140 años, entre ellos resalta la fusión BIC Banco de Colombia en el año 1997, además, para el año 2005 se presentó la fusión de Bancolombia Conavi y Corfinsura, que dio origen al grupo Bancolombia [2].

Grupo Bancolombia tiene presencia en Centro América. En el año 2007 se dio la adquisición del Banco Agrícola en El Salvador. En 2013 se adquieren las operaciones del HSBC Pana, Dando origen a Banistmo; el Banco Agromercantil de Guatemala (BAM) se adquiere en el 2013 y en el 2020 se consolidaría como el dueño absoluto del 100% de las acciones del Grupo Agromercantil Holding (GAH) [2].

Las prácticas tuvieron lugar en la Vicepresidencia de Servicios de Tecnología, puntualmente en la Gerencia donde se administra y soporta todo el Ciclo de Crédito, la cual cubre la experiencia de principio a fin del proceso de crédito (originación, mantenimiento y cobranza) para los clientes de todos los segmentos, logrando alineación y sinergia con otras áreas; que genere eficiencia y mejore integralmente la calidad de la cartera.

El foco de la práctica se centró en la solución relacionada con la cobranza, cuyo propósito es la recuperación de cartera participando en el proceso de Soporte y Mantenimiento de la aplicación Adminfo Vsmart, gestionando y usando diferentes tipos de herramientas para lograr su calidad y estabilidad tecnológica, a través del ciclo de vida del software, especialmente en la etapa de pruebas, haciendo uso de procesos automatizados que simulasen la interacción de un usuario con la aplicación y con esto lograr la disminución de incidentes y riesgos en ambientes productivos.

## II. PLANTEAMIENTO DEL PROBLEMA

### A. *Antecedentes*

Previamente en el proyecto de la aplicación Adminfo Vsmart se hacían casos de automatización, pero enfocado al módulo de BRP, este se trabajó durante un tiempo, en el cual, solo se hacían pruebas funcionales. Para los demás módulos se hicieron pruebas manuales y documentadas; dándole prioridad a los requisitos del negocio ya que estos necesitaban salir a producción. También se hacían pruebas manuales al ecosistema de Adminfo: (Informes, plan de pruebas en Word, evidencias, capturas de pantalla o PSR (grabar opciones de usuario), concepto de calidad, carta de aprobación), con la finalidad cumplir ciertas métricas que eran importantes para el banco y la EVC.

Aproximadamente en el año 2017, se dejó a un lado el proyecto por el tema de rotación de personal, además de que, se llegó a tener hasta tres sprint atrasados certificando las historias de usuario, esto se daba por el hecho de que, por ejemplo, había 5 personas desarrollando y 1 solo orientado a las pruebas. En conclusión, las cantidades de historias para certificar eran mayores a las que se podían recibir en un sprint.

### III. JUSTIFICACIÓN

Desde el sitio de contenidos del área de tecnología, en el proyecto Adminfo Vsmart se tenían unas métricas de calidad, las cuales se hacen en un tablero de Grafana que tienen unos lineamientos específicos del banco y se vela por que se cumplan los criterios; estas métricas se encontraron al 56,3%. Se suspendieron las pruebas automatizadas debido al fenómeno del cuello de botella entre los desarrolladores y los Quality Assurance (QA), esto debido a la falta de tiempo y lo complejo que podría ser la construcción del diseño de un Script.

Esto se ve reflejado en expertos que así lo aseguran como es el caso de Fiorella Ichkhanian que afirma: “Algunas empresas ya aplican procesos de QA & Software Testing, cuentan con un área dedicada, pero el equipo no alcanza a cubrir las necesidades de sus proyectos [3]”. Así mismo, un informe de Capgemini y Sogeti, que se basa en una encuesta realizada a 500 máximos responsables del área de TI de sus respectivas empresas, revela que el 62% de los encuestados, le cuesta mucho esfuerzo encontrar profesionales con los conocimientos necesarios para la elaboración de su estrategia de pruebas continuas de software, además, el informe indica que sin la adopción plena de pruebas continuas de software las empresas llegarán a un punto en el que no podrán satisfacer las necesidades de los clientes, con lo que serán vulnerables ante competidores más ágiles y mejor adaptados [4].

En ese sentido a pesar de que automatizar puede consumir mucho tiempo en la creación de los Scripts, estas pruebas automatizadas permiten [5]:

1. **Tener una mayor precisión:** Con las pruebas automatizadas se eliminan los típicos errores causados por los desarrolladores cuando se realizan pruebas manuales. Cada vez que se ejecuta una prueba automatizada se hace de forma precisa, ejecutando siempre las mismas operaciones, lo que ofrece un resultado más preciso y fiable. Ejecutar pruebas de forma automática siempre que se realice un cambio asegura la calidad en el desarrollo del software.

- 
2. **Reducción de tiempo y costo:** Las pruebas automatizadas tienen un tiempo de ejecución menor, pero su coste inicial de implementación es alto. Sin embargo, el ahorro de tiempo amortigua esta inversión inicial consiguiendo que sean mucho más rentables a nivel económico.
  
  3. **Mayor reutilización:** Automatizar las pruebas y poder repetir las constantemente sin costos adicionales de desarrollo permite entregar actualizaciones o nuevas versiones en mucho menos tiempo.
  
  4. **Incrementa la productividad de los equipos de trabajo:** Los equipos de trabajo se benefician de la automatización de pruebas al no tener que ocupar su valioso tiempo en realizar pruebas manuales constantemente.

Es por esto que se vuelve importante aportar al banco un aseguramiento de la calidad y de estabilidad tecnológica. Es así como se optó por llevar a cabo un proyecto que busque crear y diseñar un script de automatización con una ruta crítica planteada de la aplicación Adminfo Vsmart, especialmente en la etapa de pruebas usando procesos automatizados que simulen la interacción de un usuario con la aplicación.

## IV. OBJETIVOS

### *A. Objetivo general*

Implementar buenas prácticas del ciclo de vida del software desarrollando una serie de Scripts de pruebas automatizadas cumpliendo los estándares de Devops y así asegurar la respuesta y la calidad de la aplicación Adminfo Vsmart.

### *B. Objetivos específicos*

- Asegurar que se cumpla el proceso de pruebas definido por la organización para la aplicación Adminfo Vsmart.
- Dar concepto de calidad a la aplicación a través del Script, por medio de un check de las pruebas automatizadas para que este Script pueda pasar a producción.
- Velar que por medio del uso de herramientas como Grafana, Hygieia y SonarQube se cumplan los estándares de calidad de la aplicación con un porcentaje mayor al 80%.
- Explorar sobre las herramientas y lenguajes de programación definidos por la organización, para la implementación de pruebas de software.
- Apoyar en la estrategia y lineamientos del equipo de trabajo a través de las distintas ceremonias del marco de trabajo ágil.

## V. MARCO TEÓRICO

Las pruebas de software podemos dividir las en 2 grandes grupos:

### 1. Prueba manual

Es realizada por un ser humano, sentado frente a una computadora ejecutando cuidadosamente los pasos de prueba, estos pasos están ligados a casos de prueba para satisfacer las reglas de negocios de nuestro clientes o sistema [6].

### 2. Pruebas automatizadas

Estas se realizan mediante una herramienta o software de automatización (herramientas como SoapUI, Selenium, Ranorex, LoadUI, etc.) el cual ejecutará una suite de casos de pruebas. También se puede introducir datos, comparar los resultados esperados y reales, así como generar informes de prueba detallados que permitan responder a preguntas como: ¿efectúo el Test case o los TC's?, ¿falló y en qué momento fallaron? y el ¿por qué fallaron? [6].

### Gherkin

Gherkin es un DSL (Domain-Specific-Lenguaje) o lenguaje específico de dominio, es decir, un lenguaje que esta creado para resolver un problema, tiene una estructura generada por varios elementos, como se observa en la siguiente imagen [7]:

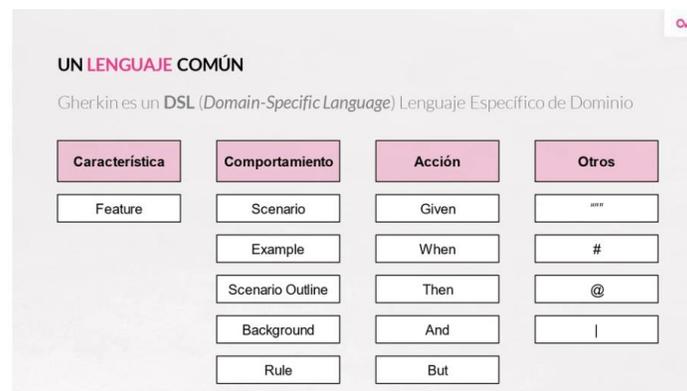


Fig. 1. Gherkin

La Fig.1. presenta como está constituido el lenguaje Gherkin y su planteamiento, las características del lenguaje, así como su comportamiento y las diferentes acciones que se pueden

realizar, además de los símbolos reservados de este lenguaje. Los elementos más utilizados habitualmente son: *Feature, Scenario, Example, Scenario Outline, Given, When, Then y And*.

### *Calidad del software*

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad [8].

La calidad del software es medible y varía de un sistema a otro o de un programa a otro. Un software elaborado para el control de naves espaciales debe ser confiable al nivel de "cero fallas", la calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software [8].

### *Selenium*

Es un entorno de pruebas que se utiliza para comprobar si el software que se está desarrollando funciona correctamente. Esta herramienta permite: grabar, editar y depurar casos de pruebas que se pueden automatizar [9].

### *Script*

Los scripts son fragmentos de código que tienen como objetivo realizar o añadir funciones dentro de una página web o e-commerce. La evolución de los scripts los ha convertido en parte fundamental de la programación web y sin ellos no sería posible realizar muchas de las funciones que las páginas actuales ofrecen a los usuarios [10].

Para crear un script para un sitio web es fundamental definir de forma precisa cuál es la función de dicho programa y qué va a aportar al sitio web. Luego es necesario disponer de los conocimientos necesarios en programación, por ejemplo, en JavaScript, para poder plasmar esa idea en código que pueda ser interpretado y ejecutado en los navegadores de los usuarios [10].

## VI. METODOLOGÍA

### *A. Fase de capacitación*

1. Se hizo una inducción acerca de los procesos de Bancolombia, las herramientas, metodologías y el funcionamiento de cada metodología, en este caso, la metodología Scrum. La organización cuenta con un repositorio grande de información con todos los lineamientos y estándares que se deben seguir en los diferentes proyectos.
2. Se comenzó con la exploración del proyecto que se podría manejar y basado en este se hicieron una serie de capacitaciones en pruebas automatizadas, la instalación y adecuación del equipo de trabajo con las diferentes aplicaciones a usar.

### *B. Fase de planificación y participación*

3. En esta fase se realizaron reuniones periódicamente con el equipo de trabajo y el respectivo líder para la elaboración del cronograma y plantear como se abordaría el proyecto, su alcance y los retos que se tendrían sprint tras sprint, cada vez a más detalle con las tecnologías a usar y leyendo la documentación necesaria para la creación del plan de pruebas, levantamiento de los riesgos, diseño del diagrama de flujo y el modelo de capas del proyecto de automatización.

### *C. Fase de ejecución*

4. Se hizo una contextualización del proyecto Adminfo, que como se ha dicho, consiste en el ciclo de la gestión de la cobranza, la cartera preventiva, la cartera administrativa, la cartera pre-jurídica, la cartera jurídica, la cartera castigada y la administración de bienes y garantías; (bienes recibidos en pago) la cual lo constituyen los dos módulos (perfiles y mantenimiento). Durante las reuniones de la fase de planificación se determinó que para el alcance se trabajasen solo algunos de los más importantes, debido a la gran cantidad de perfiles existentes alrededor de la aplicación, así mismo el detalle de estos, para proceder a diseñar los escenarios y la ruta crítica del proyecto con los módulos a realizar.

5. Luego de haber planteado la ruta crítica y haberla diseñado en Bizagi, se comenzó con el levantamiento de riesgos, que consiste en el control de calidad para establecer prioridades entre las pruebas y la probabilidad o el impacto de su fallo.
6. Se inició el plan de pruebas, que es un conjunto de transacciones que se encarga de probar las funcionalidades, además es una herramienta general de los profesionales dedicados a QA para que el trabajo sea efectivo en el ciclo de la vida del Software. En este caso, al definir el alcance del proyecto, el grupo de trabajo determinó que se harían las pruebas automatizadas a los perfiles 16, 22, 30, 68, 179 y el módulo de mantenimiento.
7. Basado en la ruta crítica se prosiguió en Intelij a plasmar la realización de los casos de prueba en lenguaje Gherkin para crear entre 1 y 3 transacciones por perfil.
8. Al iniciar con la primera etapa del proyecto, es decir, el Login, se necesitó para funcionar adecuadamente, la instalación y configuración de Intelij, Gradle, Cucumber, y Java JDK 11, igualmente, se procedió a configurar el repositorio donde este sería almacenado, el pipeline de Build RM, el repositorio local con el repositorio en Azure y el tablero de métrica de Grafana, además de hacer los Xpath, es decir, los localizadores de la página web de cada perfil.
9. Para el siguiente paso del proyecto, se comenzó con el perfil 22 de Adminfo, siendo uno de los más grandes, ya que es la parte donde la mayoría de los usuarios hacen uso de este. De acuerdo al cronograma se pudo cumplir las transacciones en su totalidad, por consiguiente se pudo comenzar con el siguiente perfil, es decir el 68, el cual está constituido por la modificación de la fecha de actuación y la consulta de las últimas gestiones dando validación en un archivo de Excel.
10. Luego se comenzó con el perfil 16, se hizo la consulta y la generación de informes pre-jurídicos y el resumen de seguimiento de un cliente activo, localizando correctamente estos informes en pantalla y logrando que estos hayan sido exportados como Excel y CSV.

11. Se continuo con el perfil 30 y se consultó un cliente con gestiones activas y se exportó en Excel las gestiones de informe de seguimiento jurídico de este cliente activo.
12. Más adelante se siguió con la automatización del perfil 179 dividiéndose en dos partes debido a sus extensos pasos; la primera parte de la edición de este estructurador de demandas consistió en tener un cliente con gestiones activas editando su tipo de proceso, *agregando un demandante y un abogado* para ver que se vean reflejados estos cambios. En la segunda parte del estructurador de demandas se debía agregar por una pestaña llamada “Administración de Bienes y garantías” y agregar un tipo de demanda la cual lo constituían: *Seleccionar y agregar un vehículo*, llenando una serie de campos específicos como: Modelo, placa, kilometraje, color, etc y *agregar una vivienda* también llenando una serie de campos obligatorios, para poder guardar esta gestión junto con la del vehículo asignando ambas a un usuario de tal modo que se permita guardar la gestión sobre el usuario y validar que queden guardados correctamente.
13. Después de haber terminado con el perfil 179 y los demás perfiles se dio inicio al módulo de mantenimiento el cual se constituía de 3 partes; la primera parte fue validar y crear un usuario nuevo llenando una serie de campos determinados y obligatorios, para crear este usuario y que este quedará guardado correctamente, se hizo este mismo ejercicio para 2 usuarios. La segunda parte consistió en que estos usuarios ya creados anteriormente se le asignasen unos determinados permisos para diferentes tipos de perfiles validando que el usuario al que se le estaba dando permisos, estuviese activo. Una tercera parte, consistió en que se debía cerrar los usuarios creados anteriormente en la primera etapa quitándole sus permisos, todo esto en una opción que da la aplicación para subir un archivo TXT y que en este archivo estuvieran escritos los usuarios para tomar los usuarios y validar que estos no queden activos después de quitarle los permisos.
14. Finalmente, se abarcó la última etapa del proyecto, la cual consistió en refactorizar SonarQube donde se pretendió optimizar y reducir el código basura y no utilizable de los perfiles 16, 22, 30, 68, 179 y mantenimiento, aumentando las métricas de calidad y bajando el Code Smells (las sugerencias técnicas a 20) y bajar el tiempo de la deuda técnica de 7 horas a 2 horas.



*B. Fase de ejecución*

*Diagrama de Bizagi*

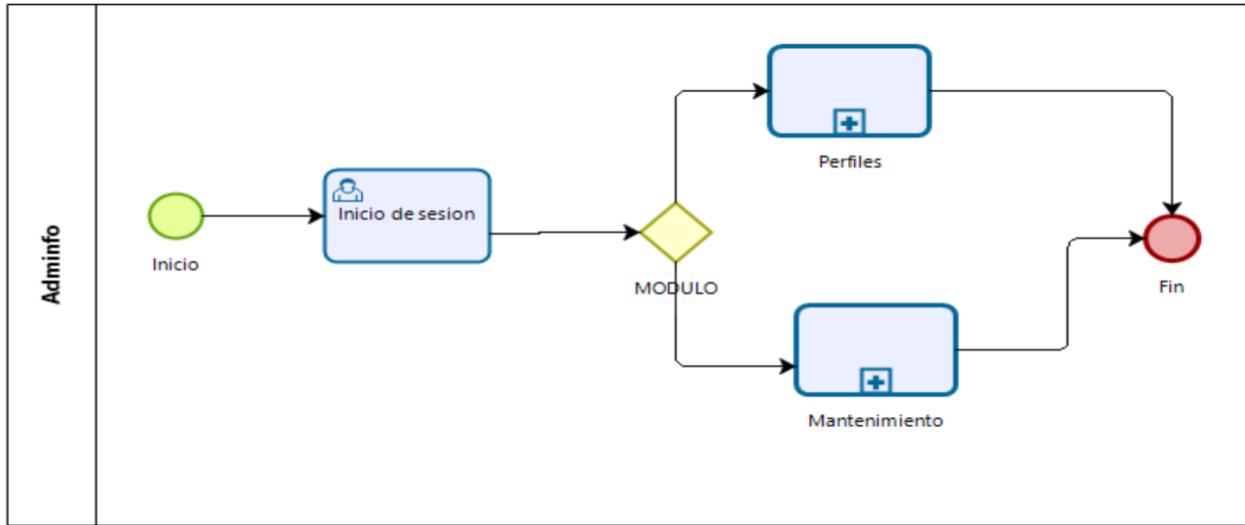


Fig. 3. Diagrama Bizagi de Adminfo

En la Fig.3 se puede ver el camino a seguir en la ruta crítica de Adminfo gráficamente, y el paso a paso de un usuario al iniciar sesión y escoger a qué módulo iría y su fin.

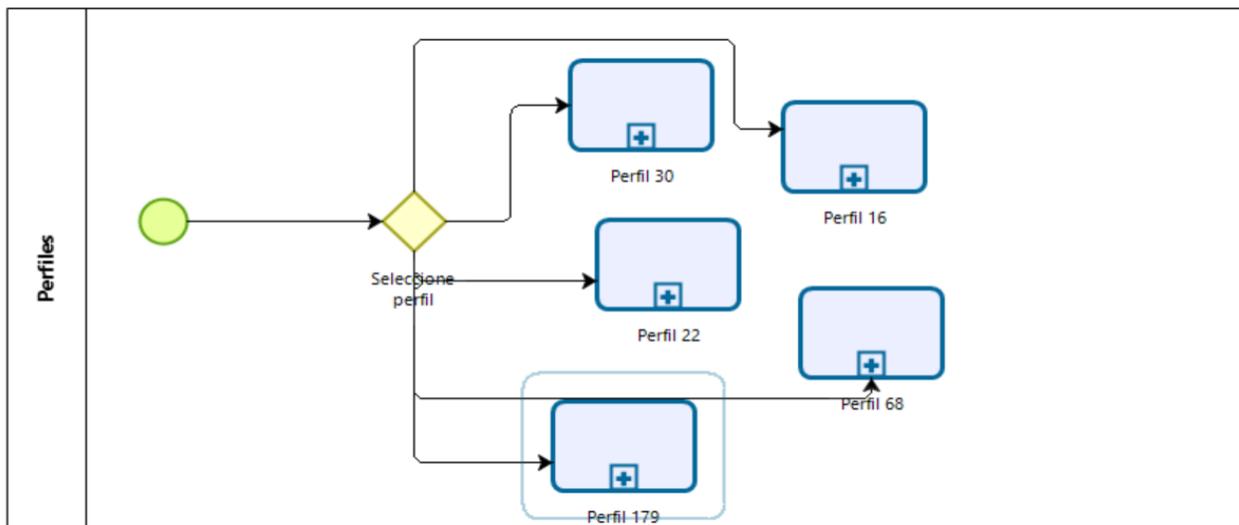


Fig. 4. Diagrama de los perfiles 16-22-30-68-179

Continuando con la explicación de la Fig.3, al ingresar en el módulo de perfiles, se ve representado en la Fig.4 donde se encuentran los perfiles automatizados con su ruta crítica, anteriormente explicada en la aplicación de Adminfo Vsmart.

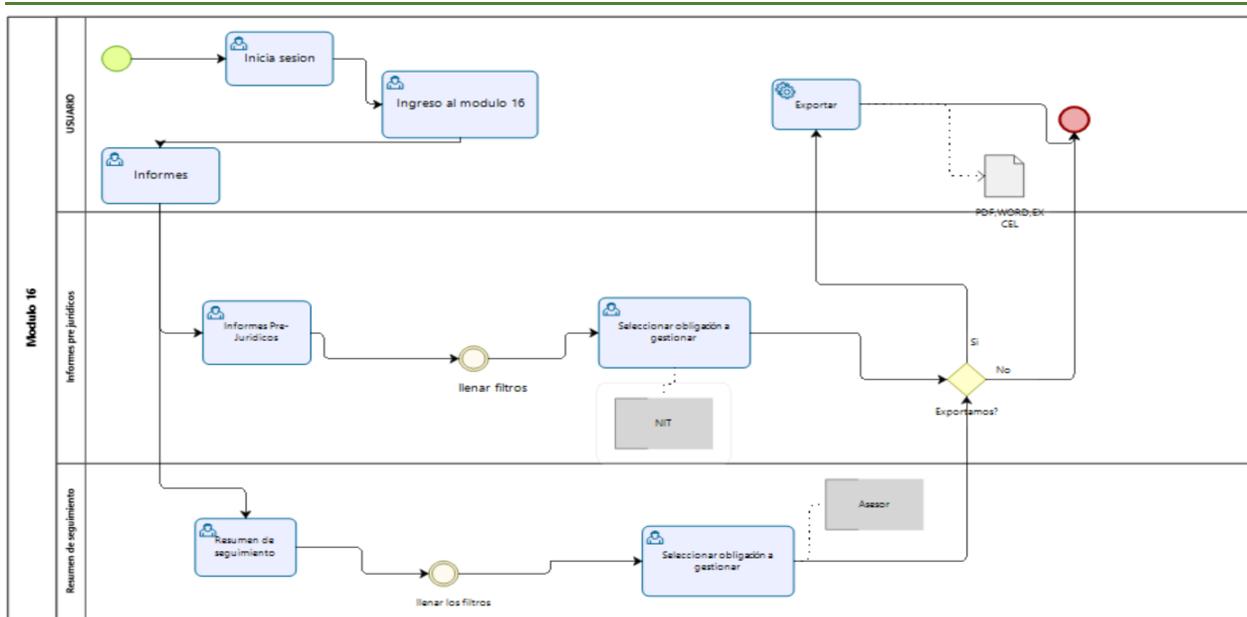


Fig. 5. Diagrama de Bizagi perfil 16

En este diagrama se ve la ruta crítica del perfil 16 y sus transacciones que simulan los pasos que debe hacer una persona y que se ve reflejada en la automatización.

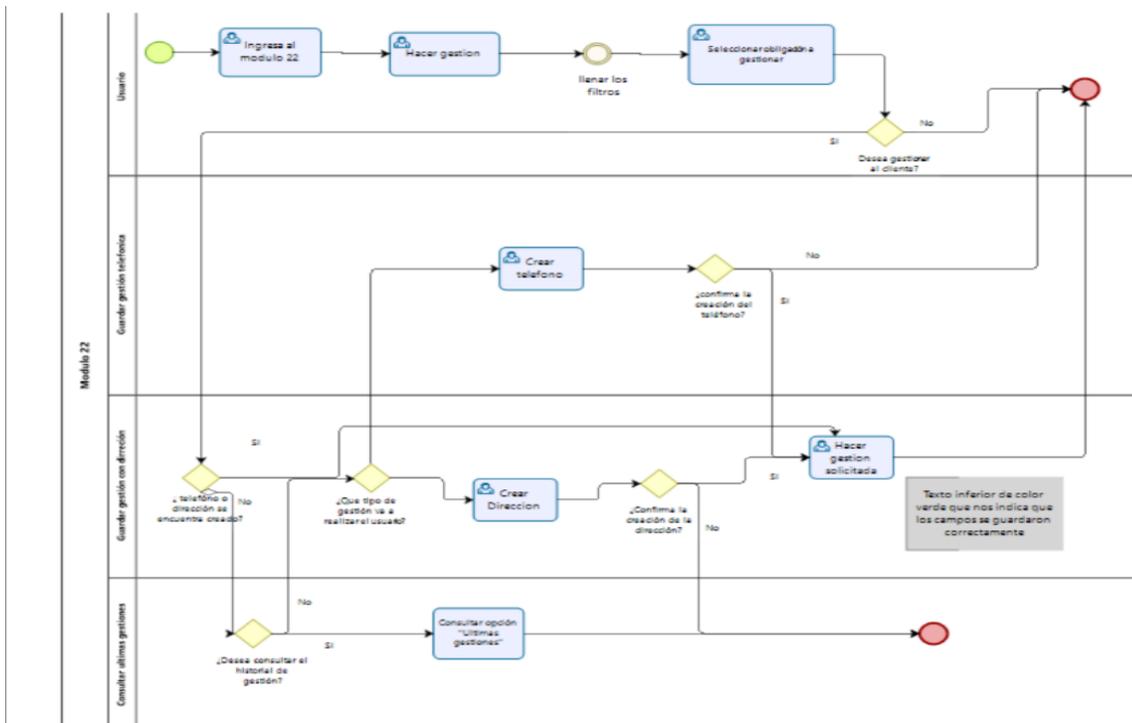


Fig. 6. Diagrama de Bizagi perfil 22

En este diagrama se ve la ruta crítica del perfil 22 y sus transacciones que simulan los pasos que debe hacer una persona y que se ve reflejada en la automatización.

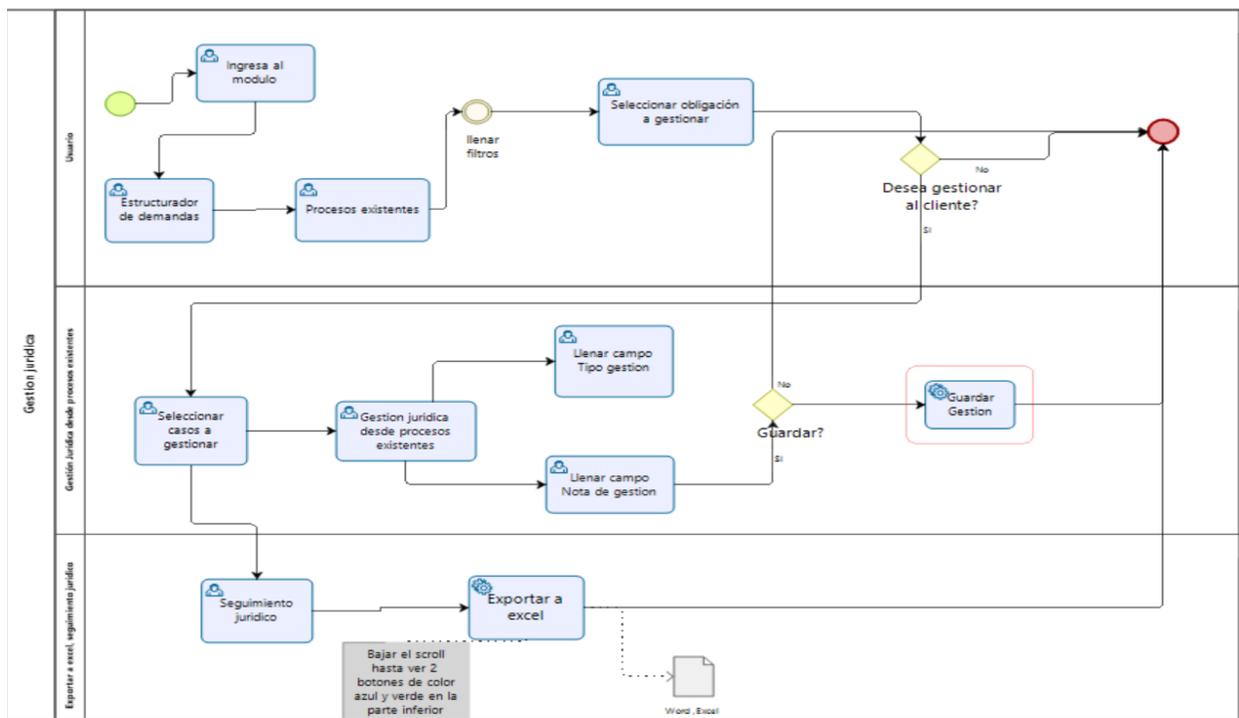


Fig. 7. Diagrama de Bizagi perfil 30

En este diagrama se ve la ruta crítica del perfil 30 y sus transacciones que simulan los pasos que debe hacer una persona y que se ve reflejada en la automatización.

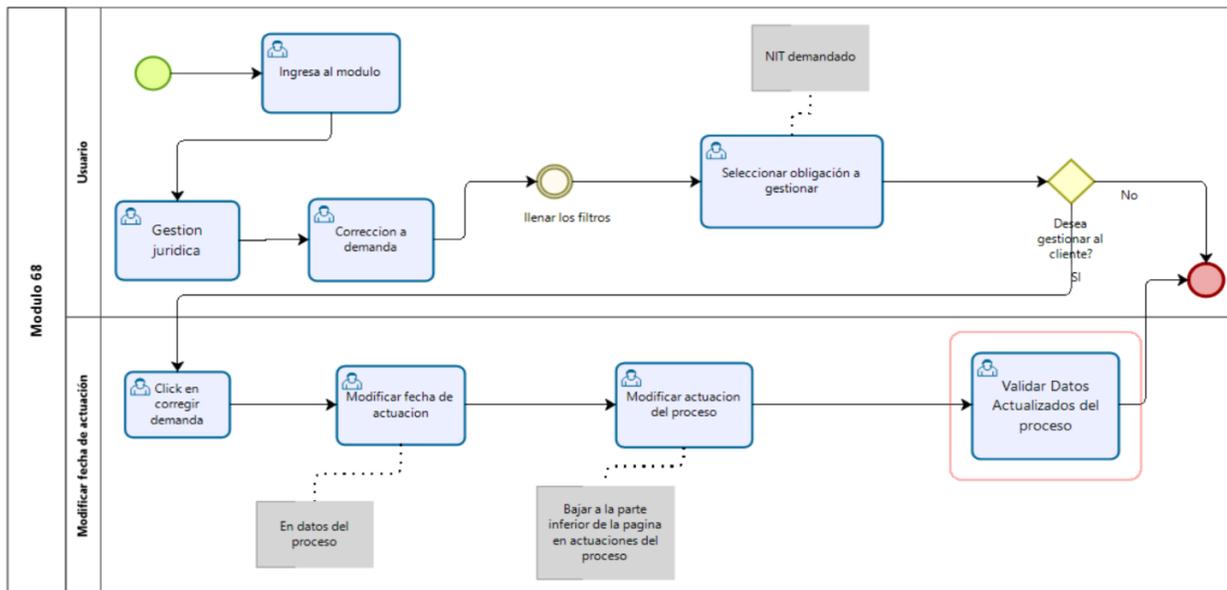


Fig. 8. Diagrama de Bizagi perfil 68

En este diagrama se ve la ruta crítica del perfil 68 y sus transacciones que simulan los pasos que debe hacer una persona y que se ve reflejada en la automatización.

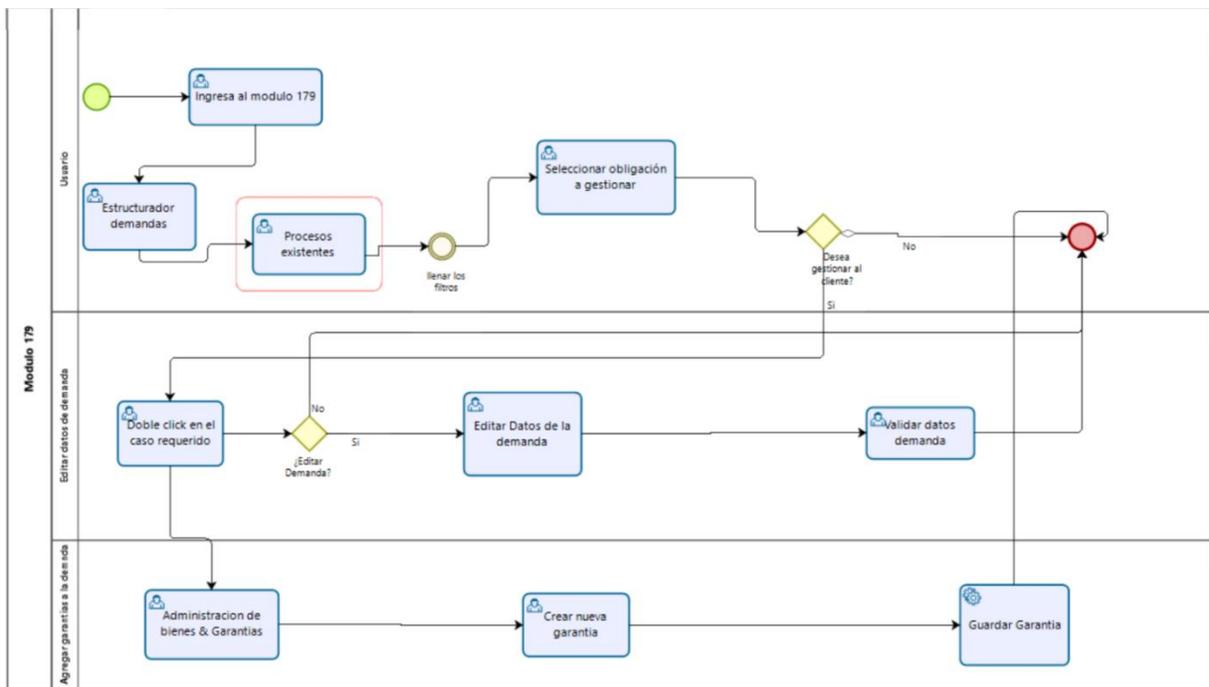


Fig. 9. Diagrama de Bizagi perfil 179

En este diagrama se ve la ruta crítica del perfil 179 y sus transacciones que simulan los pasos que debe hacer una persona y que se ve reflejada en la automatización.

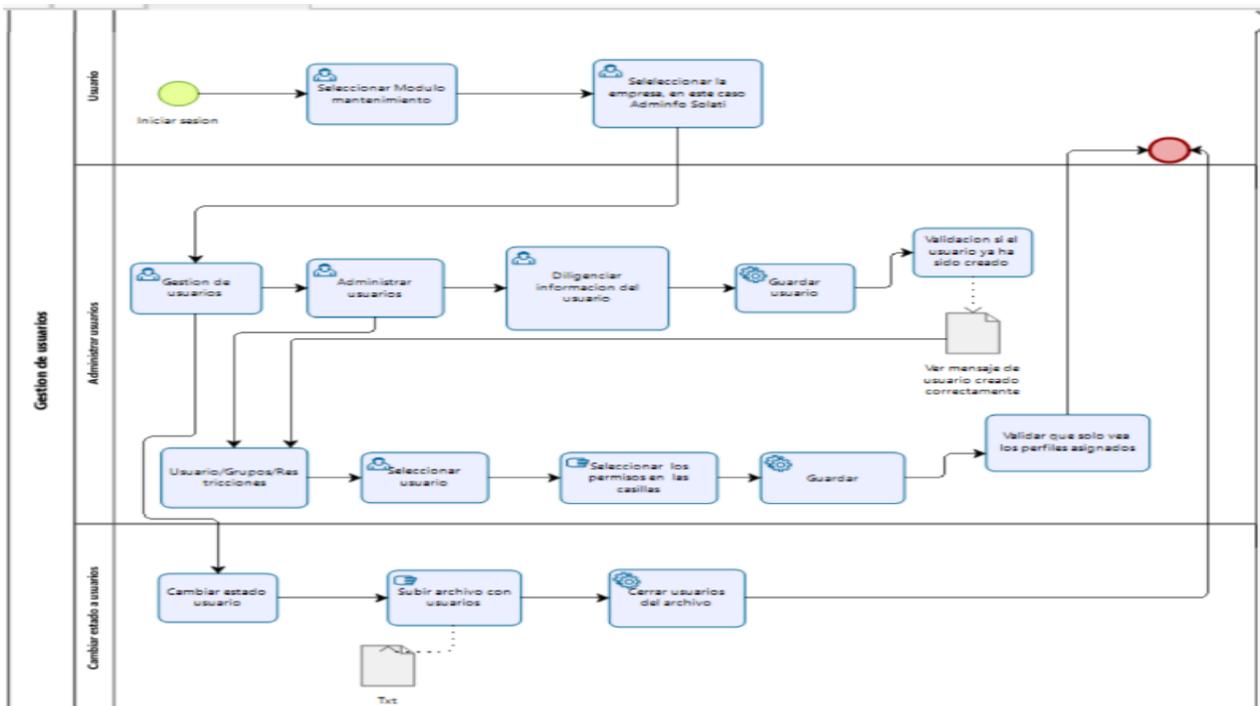


Fig. 10. Diagrama de Bizagi Modulo de Mantenimiento

En este diagrama se ve la ruta crítica del Módulo de Mantenimiento y las transacciones que simulan los pasos que debe hacer una persona y que se ve reflejada en la automatización.

### Levantamiento de riesgos

Consecutivo	R/I/RT	Fecha Identificación del riesgo	Descripción del riesgo	Información adicional	Plan de mitigación	Frente responsable	Nombre del responsable	Fecha del plan	Impacto (1-5)	Probabilidad (25%-100%)	Riesgo	Estado del Riesgo	Fecha Cierre Riesgo
1	R	9/02/2021	Se evidencia la probabilidad de que al momento de generar un informe pre-jurídico el sistema pierda la conexión con la base de datos debido a que la información generada contiene una cantidad de registros superior a la soportada por la aplicación lo que puede ocasionar lentitud en la	Riesgo de producto	Se generara un informe pre-jurídico con una alta volumetria de registros para verificar los recursos de la aplicación no se vean afectadosy así garantizar la fluidez de la aplicación.	Solati			3	25%	0.75	Abierto	
2	R	10/02/2021	Se evidencia la probabilidad de que al momento de generar un informe resumen de seguimiento el boton de exportar como excel, falle o se bloquee y que se pierda la conexión con la base de datos debido a la no respuesta del sistema, lo que pueda ocasionar una dificultad para el usuario al momento de visualizar los datos	Riesgo de producto	Se genera un informe con alta volumetria para verificar la fluidez y que este soporte la exportacion a gran masa y así garantizar la funcion de la aplicación de forma fluida.	Solati			4	25%	1	Abierto	
3	R	10/02/2021	Hay probabilidad de que a la hora de hacer una gestion y buscar obligaciones las ventanas emergentes o las cortapantallas fallen debido a que las cookies bloqueen esta vista, lo que puede ocasionar que el usuario no pueda continuar con la gestión	Riesgo de producto	Se realizan pruebas para verificar registros y funcionalidad correcta de la aplicación y si esto trae complicaciones para los usuarios.	Solati			5	50%	2.5	Abierto	
4	R	10/02/2021	Se evidencia la probabilidad de que al momento de crear un telefono o una direccion no se almacene correctamente debido a un fallo de conexión con la base de datos lo que puede ocasionar una perdida de informacion	Riesgo de producto	Se realizan pruebas de los registros y que estos si queden alojados en la base de datos de forma adecuada garantizando la calidad de la aplicación	Solati			5	25%	1.25	Abierto	
5	R	11/02/2021	Se evidencia la probabilidad que en el momento en que un usuario este verificando las gestiones fechas no queden estas guardadas al este cerrar sesion debido a un problema de conexión con el servidor lo que puede ocasionar retrasos en tiempo	Riesgo de producto	Se genera un informe de seguimiento en estos casos para que se optimice los tiempos de la aplicación y así garantizar la fluidez	Solati			4	25%	1	Abierto	
6	R	11/02/2021	Hay probabilidad de que a la hora de estar gestionando los procesos existentes y quieras agrupar varios casos y exportarlos como zip estos no se descarguen o no se abran adecuadamente debido a que no se este cumpliendo la informacion adecuadamente y esto puede ocasionar problemas en la adecuacion de informacion	Riesgo de producto	Se realizan pruebas a la hora de exportar los archivos para verificar este problema el cual seria habilitar las ventanas emergentes, para poder descargarlo y exportarlo.	Solati			3	25%	0.75	Abierto	
7	R	11/02/2021	Se evidencia la probabilidad a la hora de hacer la gestion juridica y corregir la demanda el boton principal para hacer esta gestion no funcione debido a que el boton en el la interfaz del usuario no conecte adecuadamente a la funcion solicitada ocasionando problemas en la aplicación	Riesgo de producto	Se genera un informe a la parte de desarrollo para que puedan solucionar este problema en ese modulo lo mas pronto posible	Solati			4	25%	1	Abierto	
8	R	11/02/2021	Se evidencia la probabilidad de que a la hora de que se haga una gestion prejuridica no se visualice correctamente el informe debido a problemas de conexión con la base de datos ocasionando poca fluidez en la aplicación y que no se pueda exportar el archivo	Riesgo de producto	Se realizan las pruebas pertinentes para verificar este modulo con su respectiva exportacion de archivo para garantizar la fluidez en la aplicación.	Solati							
9	R	11/02/2021	Se evidencia la probabilidad en el estructurador de demanda y los procesos existentes a la hora de realizar la gestion el boton de las fechas no funcione adecuadamente debido a que el servidor no se conecta con el api de calendario, lo que puede ocasionar retrasos en la gestión de las	Riesgo de producto	Se realizarán más de 3 gestiones con la actualización de la fecha para verificar que sea posible seleccionar una fecha y así garantizar que funcione correctamente la interfaz del calendario.	Solati			2	25%	0.5	Abierto	
10	R	11/02/2021	Se evidencia la probabilidad de algun fallo al tener que darle doble click para poder editar la demanda en los procesos existentes del estructurador de demandas, debido a que un usuario solo tiene este metodo para poder editar lo que puede ocasionar	Riesgo de producto	Se realizan multiples gestiones con diferentes usuarios y se verifica que la funcionalidad del doble click no falle y así garantizar que el usuario pueda realizar sus gestiones diarias	Solati			2	25%	0.5	Abierto	

Fig. 11. Levantamiento de riesgos

En la Fig.11 se ve la herramienta de levantamiento de riesgos la cual consiste en la identificación de un posible riesgo no identificado en la fase de planeación del proyecto que podría tener un impacto en la calidad del software, con esta anticipación de posibles riesgos se puede prever una falla a futuro y así evitar la obstaculización durante todo el ciclo de vida del proyecto.

---

### *Plan de pruebas*

#### **Alcance**

- **Generación de informes pre-jurídicos perfil 16:** Se requiere que el usuario pueda generar un informe básico con detalle en pantalla, además de un informe del resumen de seguimiento y que estos puedan ser descargados en Excel.
- **Gestión pre- jurídica perfil 22:** Se requiere que el usuario pueda realizar diferentes tipos de gestiones como crear teléfono, dirección y referencia con teléfono, además de activar e inactivar los teléfonos de referencia y por último consultar últimas gestiones realizadas.
- **Gestión Jurídica: perfil 30:** Se requiere que el usuario pueda gestionar los procesos jurídicos existentes y exportar a Excel el seguimiento jurídico.
- **Gestión Jurídica: perfil 68:** Se requiere que el usuario pueda gestionar la corrección a una demanda y que este pueda modificar la fecha de actuación y como resultado verificar que esta quede actualizada.
- **Asignación y estructuración de demandas perfil 179:** Se requiere que el usuario pueda editar los datos de la demanda, el tipo de proceso, el abogado, el demandante y el proceso, luego de esto, que pueda validar que los datos se actualicen correctamente, además se debe agregar la garantía a la demanda creando el tipo de garantía.
- **Módulo de mantenimiento administrar perfiles y cambiar estado de usuarios:** Se requiere que al ingresar en el módulo de mantenimiento el usuario pueda gestionar los usuarios creando un usuario, asignándole determinados perfiles y permisos al usuario creado, también validando que este no exista y por último cambiar el estado de usuario.

#### **Estrategia**

- **Módulo de informes pre- jurídicos perfil 16:** Se tomará un cliente con obligaciones activas, a continuación, se podrán ver sus obligaciones y se podrá generar cualquiera de estos casos seleccionándolos y exportando como Word, Excel o PDF.

- **Gestión Jurídica: perfil 30:** Se debe tener un cliente con procesos existentes para poder visualizar las gestiones y proceder a exportar en el seguimiento jurídico.
- **Gestión Jurídica Perfil 68:** Se debe tomar un cliente con gestiones activas, luego de esto se debe permitir la corrección de la demanda con foco en la fecha de actuación y que esta se pueda modificar con un usuario y por último se debe validar y verificar que esta quede actualizada con los cambios aplicados.
- **Asignación y estructuración de demandas perfil 179:** Se debe tener un cliente con una demanda activa para permitir editar los datos de tipo del proceso y con esto permitir gestionar la demanda, modificando el tipo de proceso, se le asigna el abogado y el demandante, además del tipo de proceso, luego de esto se debe validar que estas modificaciones queden aplicadas, finalmente se crea la garantía de tipo vehículo y se guardaran estos cambios para culminar con la asignación de la demanda.
- **Gestión pre- jurídica perfil 22:** Se tomará un cliente con obligaciones activas, posteriormente se le crea un teléfono y dirección de Colombia luego se realizarán 2 gestiones telefónicas y 2 gestiones domiciliarias, se inactivarán y activarán el teléfono, la dirección y por último se validará en la opción de últimas gestiones realizadas y que estas queden guardadas correctamente.
- **Módulo de mantenimiento administrar perfiles y cambiar estado de usuarios:** Se tomará un usuario nuevo para que pueda ingresar al módulo a gestionar usuarios, se deben crear 2 usuarios y asignarles determinados perfiles y permisos, también se debe validar que estos 2 usuarios no estén registrados en la aplicación, por último, se debe cambiar el estado a modo cerrado a 2 usuarios quitándole sus permisos (carga masiva).

## Automatización

```
@test1
Esquema del escenario: Editar demanda fecha de actuacion
  Dado que el usuario requiere editar la demanda en el modulo correccion a demanda <perfil>
  Cuando el usuario modifica la fecha de actuacion <fecha>, <codigo_demanda> y <nota>
  Entonces Se podra visualizar la <fecha> con su actualizacion correcta

Ejemplos:
  |perfil|tipo|numero|codigo_demanda|nota|
  |12|32|4|12950795|Se realiza la gestion de actualización|
```

Fig. 12. Esquema lenguaje Gherkin

En la Fig.12 se presenta el esquema del lenguaje Gherkin antes mencionado con un ejemplo aplicado en el proyecto con forma del esquema que sería el título del escenario. El DADO, el CUÁNDO y el ENTONCES, así mismo un ejemplo de un usuario.

Perfil_16.feature	1K	1 mar
Perfil_179.feature	1K	1 mar
Perfil_22.feature	2K	1 mar
Perfil_30.feature	1K	1 mar
Perfil_68.feature	1K	1 mar
Perfil_Mantenimiento.feature	1K	1 mar

Fig. 13. Perfiles para automatizar en lenguaje Gherkin

En la Fig13. se ven los .feature de los diferentes escenarios en lenguaje Gherkin.

```
Antecedentes: Usuario Ingresa a la pagina de Adminfo
  Dado que el [ ] ingresa a la pagina de Adminfo con contrasena [ ]
  Cuando el ingresa al modulo de gestion perfil 22
  Y que el consulta el cliente con Tipo nit y [ ] a gestionar la obligacion [ ]
```

Fig. 14. Login Adminfo

En la Fig.14 se puede visualizar el ejemplo práctico del proyecto, el inicio de sesión de un usuario X (por temas de control de datos no se puede suministrar esta información), pero deja a la imaginación lo intuitivo que sería y lo que iría en cada espacio resaltado de color rosado.

```
@Dado("^que el (.*) ingresa a la pagina de Adminfo con contraseña (.*)$")
public void queElAromellIngresaALaPaginaDeAdminfoConContrasiaBasico(String usuario, String contraseña){
    OnStage.theActorCalled( requiredActor: "Aldo").attemptsTo(Open.browserOn().the(AdminfoHomePage.class),
        Iniciar.enLaPagina(new Usuario(usuario, contraseña)
    );
}
```

Fig. 15. Login Adminfo Dado

Se empieza con el DADO y se utilizan los respectivos métodos y clases ya planteados en el feature, esto a través del plugin de cucumber.

```
@Cuando("^que el consulta el cliente con Tipo (.*) y (.*) a gestionar la obligacion (.*)$")
public void queElConsultaElClienteConTipoNITYNumeroAGestionarLaObligacionNumeroDeObligacion(String nit, String nu
    OnStage.theActorInTheSpotlight().attemptsTo(BuscarObligacion.aGestionar(nit, numero, obligacion));
}
```

Fig. 16. Login Adminfo Cuando

Se continua con el CUÁNDO que son las acciones que hará el actor también implementando las clases y métodos con el actor en escena y ya planteado esto en el .feature por medio del plugin de cucumber.

```
@Entonces("^El debe ver creada la (.*) en la pantalla de gestion$")
public void apareceMensajeCrewacionCorrecta(String direccion) {
    aldrendo, 12/04/2022 3:31 p. m. * Se realiza la consulta de ultimas gest
    OnStage.theActorInTheSpotlight().should(seeThat((ValidarDireccion.esVisible(direccion)), Matchers.is( value: true)).orComplainWith(Excepto
```

Fig. 17. Login Adminfo Entonces

Y finalizando el último paso del .feature en lenguaje Gherkin, este va a realizar las validaciones para que cumpla de manera adecuada, lo que se solicita en la question.

```
public class AdminfoHomePage extends PageObject {
    public static final Target TXT_LLENAR_USUARIO =Target.the( targetElementName: "LLENAR CAMPO USUARIO").locatedBy( cssOrXPathSelector: "//input[id='us
    public static final Target TXT_LLENAR_CONTRASEÑA =Target.the( targetElementName: "LLENAR EL CAMPO CONTRASEÑA").locatedBy( cssOrXPathSelector: "//inpu
    public static final Target BTN_ENTRAR =Target.the( targetElementName: "BOTON ENTRAR").locatedBy( cssOrXPathSelector: "//button[@class='botonLogin']")
    public static final Target BTN_TIPO_GESTION =Target.the( targetElementName: "BOTON SELECCIONAR GESTIION").locatedBy( cssOrXPathSelector: "//div[id='
    public static final Target BTN_SELECCION_PERFIL =Target.the( targetElementName: "SELECCIONAR PERFIL").locatedBy( cssOrXPathSelector: "//a[contains(@
```

Fig. 18. Login Adminfo Xpaths

En la Fig.18 se pueden visualizar los localizadores en la página por medio de xpath, que serán los elementos de la página web a los cuales “la task” deberá hacer acciones sobre estos.

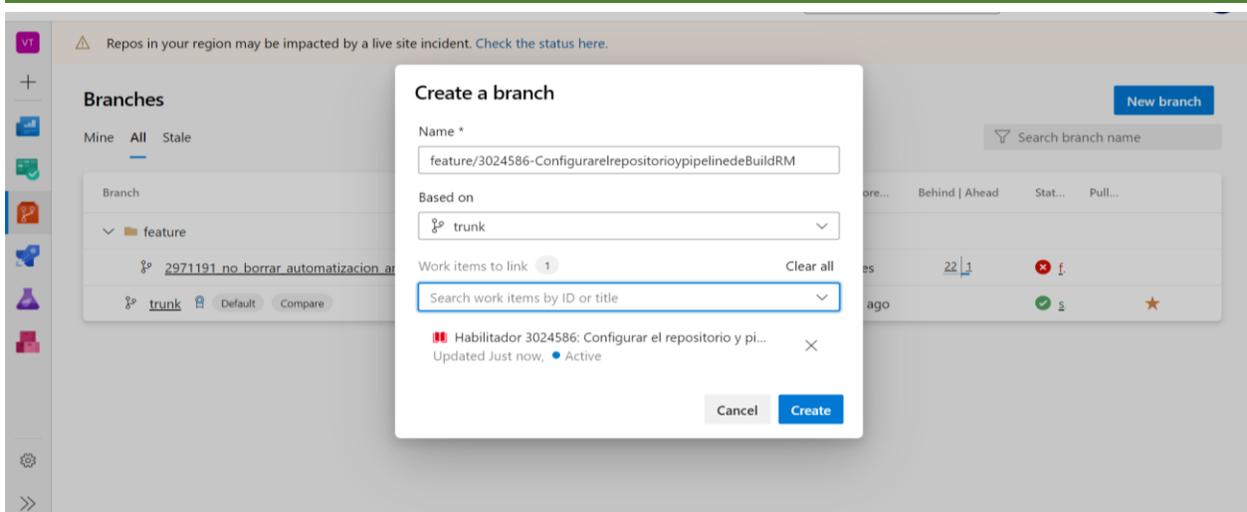


Fig. 19. Sincronización repositorio local

En la Fig.19, se ve la creación de una rama, para hacer la correcta sincronización con el repositorio local para el tema del versionamiento en nube.

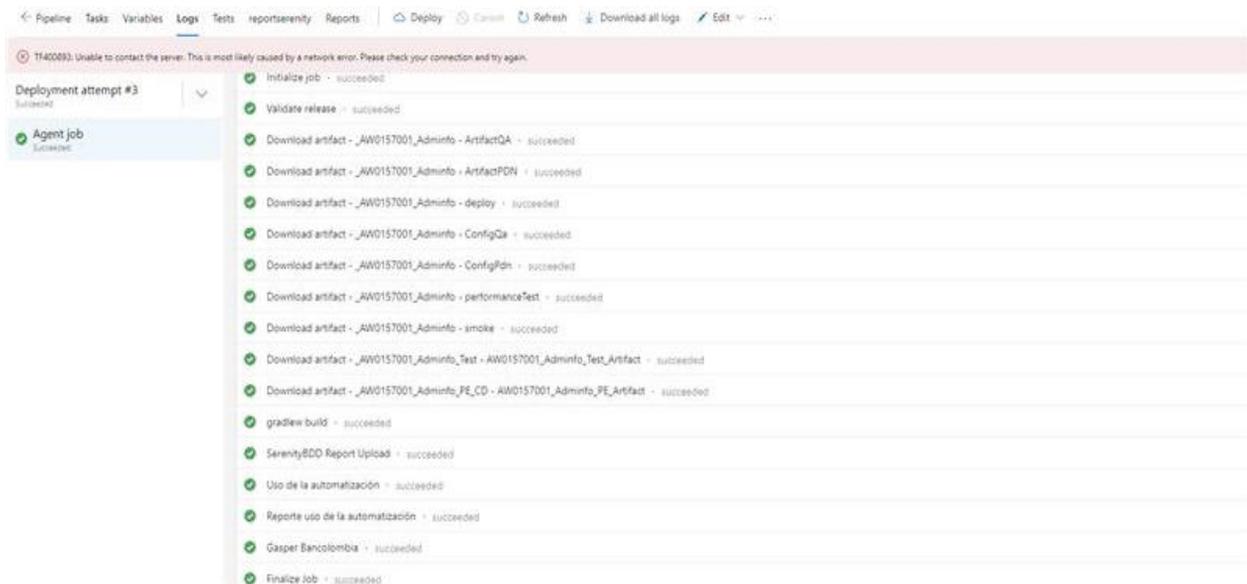


Fig. 20. Configuración de Pipeline Build RM

En la Fig.20 se puede ver la configuración del agente y el despliegue de este, por medio del pipeline de Build RM para la correcta construcción por medio del repositorio y correr las pruebas en la nube del proyecto de automatización.

## Grafana



Fig. 21. Tablero de Grafana (Antes)

Fuente: Bancolombia

En la Fig.21 se puede ver un tablero de métricas de calidad de Bancolombia, en el cual se tomó el proyecto con el cumplimiento del 56.3%.

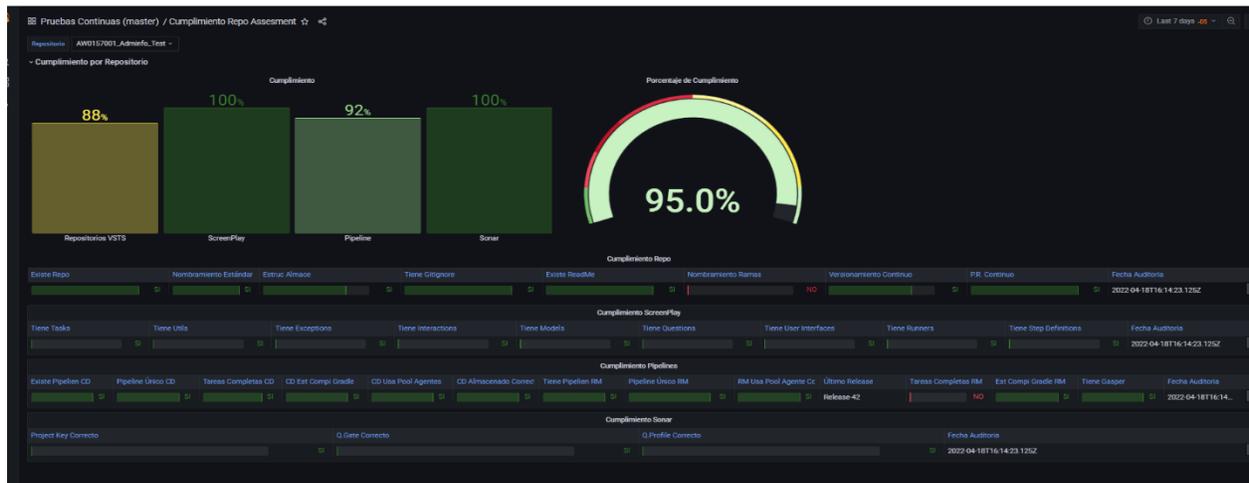


Fig. 22. Tablero de Grafana (Después)

En la Fig.22, se ve un aumento considerable del cumplimiento de las métricas de calidad establecidas por Bancolombia, en el que se estuvo trabajando a lo largo del proyecto logrando subir este porcentaje al 95%.

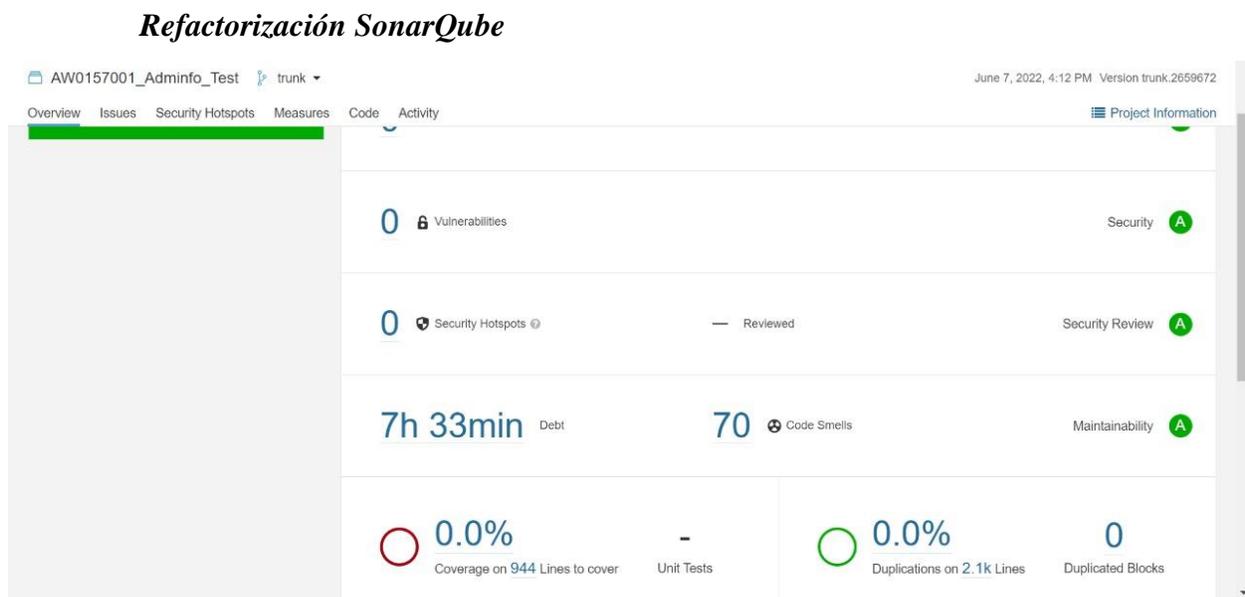


Fig. 23. Refactorización SonarQube(Antes)

En la Fig.23, se tiene que la refactorización global del proyecto muestra a través de la herramienta de SonarQube el total de deuda técnica que se tiene en todo el código del proyecto, por ejemplo, cosas que no se están usando como: Variables, clases no utilizadas y métodos.

Como resultado de este, SonarQube presenta un tiempo de 7 horas y 33 minutos para corregir los 70 Code Smells (sugerencias técnicas).

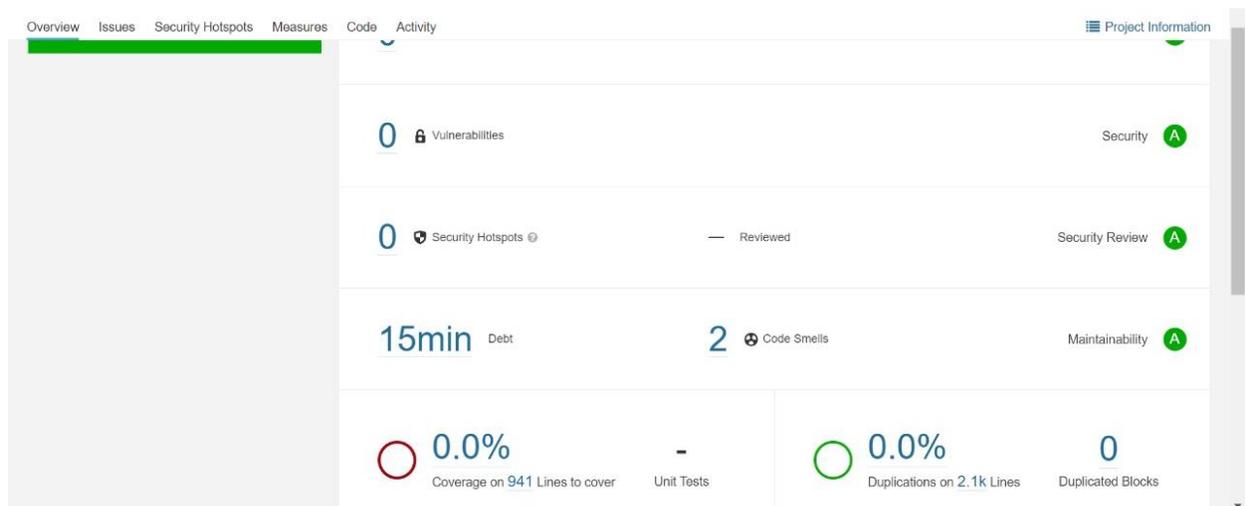


Fig. 24. Refactorización SonarQube(Después)

Después de trabajar en la deuda técnica que arrojó SonarQube se pueden ver estos resultados en la Fig. 24, en la parte de la optimización del código llegando a tener un tiempo de 15

minutos con 2 Code Smells (sugerencias técnicas) logrando así y de manera satisfactoria refactorizar el proyecto con lo sugerido de SonarQube cerca de su máxima optimización.

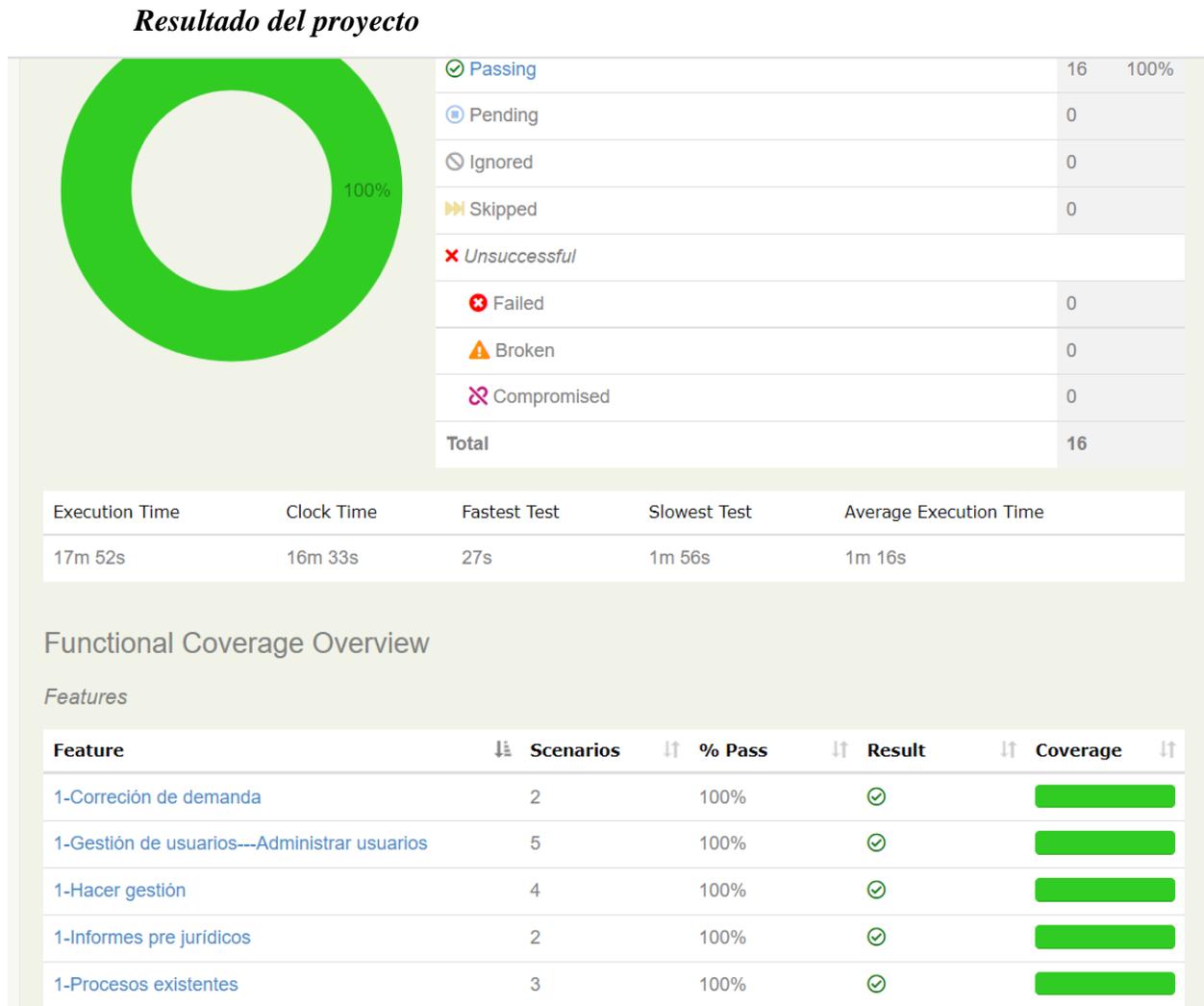


Fig. 25. Resultado Final proyecto

En la Fig. 25 se puede visualizar el proyecto compilado correctamente y funcionando en su totalidad con todas las transacciones planteadas, las pruebas automatizadas y los escenarios funcionando al 100%, todo eso a través de la herramienta de SonarQube para generar y ver el reporte de la automatización global, permitiendo el aseguramiento de la ruta crítica planteada para los perfiles 16-22-30-68-179 y el módulo de mantenimiento de Adminfo Vsmart.

## VIII. CONCLUSIONES Y RECOMENDACIONES

- En el presente proyecto se muestra la metodología para el diseño e implementación de pruebas automatizadas, teniendo claros los requerimientos de la misma, el proceso o flujo del software, los resultados esperados de cada ejecución y el paso a paso de dichas pruebas de manera adecuada.
- Gracias a los pipelines y a los servicios en nube como AWS se facilita grandemente la implementación y ejecución de los equipos de trabajo para que esta pueda pasar a producción obviando lo engorroso y difícil que es hacer la integración de todos los equipos y sistemas en tierra.
- Gracias a la herramienta de SonarQube se pueden generar reportes de cada ejecución de las pruebas automatizadas con todas las transacciones que se implementaron en la ruta crítica (Ver Fig.25), además de detectar errores dentro de las funcionalidades desarrolladas, también se muestra la información detallada del paso a paso permitiendo determinar dónde está el error o falla a través de unas imágenes evidenciando el problema de forma gráfica para poder solucionarlo.
- Se evidenció en el Tablero de Métrica de Grafana una mejora sustancial en el repositorio destinado por Bancolombia para las pruebas automatizadas, aplicando las métricas del banco se logró incrementar el porcentaje de cumplimiento de los estándares de calidad en el uso del repositorio implantación del patrón Screenplay, uso de los Pipelines y Sonar de un 56% (Ver Fig. 21) a un porcentaje del 95% (Ver Fig.22).
- La automatización busca reducir el esfuerzo humano debido a que automatizar pruebas de software busca simplificar el trabajo dispendioso, repetitivo y complejo haciéndolo más efectivo y productivo, ahorrando costos y tiempo.

## IX. REFERENCIAS

- [1] Grupo Bancolombia, « "Nuestro Propósito | Grupo Bancolombia",» 2022. [En línea]. Available: <https://www.grupobancolombia.com/corporativo/conocenos/nuestro-proposito>. [Último acceso: 18 enero 2022].
- [2] Grupo Bancolombia, «Historia y Evolución,» 2022. [En línea]. Available: <https://www.grupobancolombia.com/corporativo/conocenos/historia>. [Último acceso: 18 enero 2022].
- [3] F. Ichkhanian, «3 dificultades para abordar Testing & QA y cómo resolverlas,» Arbusta, 10 03 2022. [En línea]. Available: <https://arbusta.net/blog-es/3-dificultades-para-abordar-testing-qa-2022/>. [Último acceso: 2022].
- [4] Computing, «Las empresas deben incorporar procesos de QA al desarrollo Agile,» 03 04 2020. [En línea]. Available: <https://www.computing.es/infraestructuras/informes/1117807001801/empresas-deben-incorporar-procesos-de-qa-al-desarrollo-agile.1.html> . [Último acceso: 2022].
- [5] Ilimit, «Beneficios de la Automatización de Pruebas,» 19 02 2021. [En línea]. Available: <https://www.ilimit.com/blog/beneficioss-automatizacion-pruebas/#:~:text=Las%20pruebas%20automatizadas%20permiten%20un,automatizaci%C3%B3n%20de%20procesos%20y%20pruebas..> [Último acceso: Mayo 2022].
- [6] A. A. P. Rodriguez, «Pruebas Manuales VS Pruebas Automatizadas ¿Dónde Comenzar?,» 15 04 2020. [En línea]. Available: <https://blog.nearsoftjobs.com/pruebas-manuales-vs-pruebas-automatizadas-d%C3%B3nde-comenzar-9218140edf3b> . [Último acceso: Mayo 2022].
- [7] M. A. E. Ordoñez, «Qué es Gherkin,» 10 11 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-Gherkin/#:~:text=Gherkin%20es%20un%20DSL%20o,vemos%20en%20la%20siguiente%20imagen..> [Último acceso: 2022].
- [8] D. G. L. y A. B. B. Oscar M. Fernández Carrasco, «Un enfoque actual sobre la calidad del software.,» sep.-dic. 1995. [En línea]. Available:

---

[http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1024-94351995000300005](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94351995000300005).

[Último acceso: 2022].

- [9] S. G. SOTOMAYOR, «Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa,» 09 12 2021. [En línea]. Available: <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>. [Último acceso: 2022].
- [10] E. RODRÍGUEZ, «Qué es un script: cómo funciona, cómo crearlo o eliminarlo,» 19 10 2020. [En línea]. Available: <https://www.seoestudios.es/blog/que-es-un-script/>. [Último acceso: 2022].