



**UNIVERSIDAD
DE ANTIOQUIA**

**SOPORTE Y ESTABILIZACIÓN DE LOS
APLICATIVOS DE NEGOCIOS FIDUCIARIOS
DEL AREA DE TECNOLOGIA DE
BANCOLOMBIA**

Autor(es)
Santiago Zapata Palacio

Universidad de Antioquia
Ingeniería de Sistemas
Medellín, Colombia
2020



Soporte y estabilización de los aplicativos de Negocios Fiduciarios del área de Tecnología de
Bancolombia

Santiago Zapata Palacio

Informe final de prácticas presentado como requisito parcial para optar al título de:
Ingeniería de Sistemas

Asesores (a):
Jaime Humberto Fonseca Espinal

Universidad de Antioquia
Ingeniería de Sistemas
Medellín, Colombia
2020.

SOPORTE Y ESTABILIZACIÓN DE LOS APLICATIVOS DE NEGOCIOS FIDUCIARIOS DEL AREA DE TECNOLOGIA EN BANCOLOMBIA

Resumen

En el comienzo de esta práctica se tuvo como objetivo lograr dar soporte al área de negocios fiduciarios de Bancolombia, donde se tienen aplicativos con varios fines como administración de fondos, administración de acciones, administración de asambleas y más. Estos aplicativos deben estar disponibles en todo momento porque de estos depende el funcionamiento de la fiduciaria. Para soportar estas aplicaciones es necesario obtener el conocimiento en diversas tecnologías tales como Windows forms, oracle, ASP.NET, sql server, oracle forms, i-series, devops, pruebas unitarias, sonarqube y la debida gestión para lograr la implementación de los cambios en ambientes productivos.

Adicionalmente se debía cumplir con la agenda maestra y los proyectos que llegarán con ella cada mes tales como soportes estructurales, express y proyectos.

Los incidentes deben ser atendidos con prioridad conforme son reportados, buscar la mejor solución a los mismos, negociación con usuarios sobre estos, licitación de requisitos y elaboración de mantenimiento a errores y soportes. Todo a través de la metodología Scrum y con el apoyo de la herramienta Azure DevOps.

Finalmente se logró mediante la experiencia conseguir un nivel de conocimiento superior al requerido para atender los temas soportados, enfrentándose a ellos durante la duración de la práctica desde el inicio hasta el final de la práctica, generando así un mayor promedio de resolución de incidentes por sprint, un porcentaje de cumplimiento de proyectos del 100% y una agilidad a la hora de dar soporte mucho mayor a la inicial antes de la práctica, logrando así ser un ingeniero apto y útil para mantener los aplicativos del área de soporte de tecnología de la fiduciaria de Bancolombia.

Introducción

Accenture es una de las empresas más grandes de tecnología a nivel mundial [1]. Si bien en Colombia lleva pocos años, Accenture ha tenido un gran crecimiento en los últimos años posicionándose cada vez más en el país [2]. Una de las claves principales de su crecimiento viene la mano de los buenos resultados que ha tenido con su cliente principal y más grande en el país, el cual es una de las compañías más exitosas de Colombia, Grupo Bancolombia, el cual es el conglomerado de empresas financieras más grande del país y es la única entidad financiera colombiana que participa en la bolsa de Nueva York [3]. Como parte del grupo Bancolombia se encuentra la Fiduciaria de Bancolombia, la cual se encarga de administrar los fondos de inversión del grupo tales como las populares "Fiducuenta" y

“Fidurenta”, Concesiones que son proyectos de infraestructura y los variados negocios fiduciarios como encargos financieros y patrimonios autónomos [4], es una de las áreas que más dinero aporta al grupo Bancolombia.

Con el propósito de mantener estable el funcionamiento de la Fiduciaria de Bancolombia, el área de tecnología posee un equipo el cual da un soporte dedicado a las herramientas tecnológicas que la Fiduciaria de Bancolombia emplea en el día a día.

Entre las aplicaciones soportadas se encuentran tres (3), las cuales serán el alcance de este trabajo, a continuación, se describen brevemente junto con sus funciones principales:

- **Administración de acciones de diversas empresas:**
 - Compra y venta de acciones
 - Bloqueos y limitaciones
 - Pago de dividendos
 - Entrega de informes

- **Administración de fondos de inversión:**
 - Inversiones en bolsa
 - Certificados tributarios
 - Ingresos y egresos en fondos
 - Bloqueos de saldos
 - Entrega de informes

- **Administración de asambleas de diversas empresas:**
 - Registro de asistentes a las asambleas
 - Impresión de certificados de asistencia
 - Impresión de papeletas de votación
 - Cálculo de Quórum
 - Registro de votaciones
 - Entrega de informes y resultados de las asambleas

La función del área de soporte es mantener el funcionamiento de estas aplicaciones entre otras, solventar problemas que se presenten en ellos, implementar actualizaciones, optimizaciones, cambios, pedidos por la Superintendencia Financiera de Colombia o la Dian etc. Algunos de estos cambios son legales, lo cual implica que tienen un tiempo establecido para su cumplimiento y el no hacerlo implica una sanción para el Grupo Bancolombia a nivel monetario y puede poner en duda la participación del mismo en la bolsa de Nueva York. Para ello el equipo implementa la metodología Scrum con iteraciones de 15 días

Objetivos

General

Brindar soporte a eventos e incidentes que se presenta en los aplicativos en los cuales se manejan temas Windows forms, Oracle, Asp.net, Sql server, Oracle forms, i-series, DevOps, pruebas unitarias, Sonarqube y gestión para lograr la debida solución e implementación de la misma en producción.

Específicos

- Análisis y refactorización del código existente en caso de aportar una mejora.
- Desarrollo de cambios requeridos en las aplicaciones a soportar, sin importar si son optimizaciones, nuevos requerimientos, soluciones temporales.
- Generación de planes de mejora a incidentes recurrentes o que generan indisponibilidad del servicio.
- Solución a incidentes o errores que se generen en producción.
- Implementación de cambios en los ambientes de certificación y finalmente en producción.
- Proceso de análisis y licitación de requerimientos para elaborar soportes y mantenimiento a errores en las aplicaciones.
- Uso de la metodología ágil SCRUM con ciertas variantes que el cliente maneja, para la gestión y seguimiento del proceso.
- Utilizar Azure DevOps para la gestión de cada sprint, repositorios y paso entre ambientes

Marco Teórico

Las metodologías ágiles tienen una gran cantidad de ventajas a la hora de aplicarse en proyectos tecnológicos que han demostrado dar como resultado un mayor porcentaje de éxito frente a metodologías más tradicionales[5] siendo algunas de ellas mayor flexibilidad a los cambios, implicando una solución final más cercana a lo que los usuarios necesitan, un retorno a la inversión temprano generando confianza en el proyecto y hallazgo de errores en etapas iniciales de desarrollo generando así la oportunidad de darles solución antes de afectar el negocio o logrando que estas se replanteen de una manera diferente. Todo esto potenciando la comunicación entre el equipo solucionador y los interesados como un equipo con un mismo fin. Por estas razones, hace ya algunos años Bancolombia se ha volcado a trabajar con metodologías ágiles y cada vez madurando más sobre ellas, donde la base es Scrum [6] pero acoplado a sus necesidades. Teniendo esto en cuenta, en los equipos de tecnología, Bancolombia maneja los siguientes roles para implementar apropiadamente su método de trabajo:

- **Scrum master:** persona encargada de guiar al **Team** en su mejora continua como equipo valioso para la organización, sirve de guía en la mayoría de ceremonias y prepara algunas.
- **Product owner:** Es el usuario o la persona que representa a los usuarios e interesados en el desarrollo de la solución. Debe tener claro los objetivos y funcionalidades a conseguir mediante la solución o proyecto, ya que debe transmitirle al **Team** lo que requiere mediante documentos llamados historias de usuario, dar el visto bueno para la implementación de los cambios y priorizar los temas a trabajar.
- **Team:** Conjunto de personas con conocimientos técnicos, las cuales tienen como propósito entender las historias de usuario generadas por el **Product Owner**, plantear la mejor manera de lograr lo allí plasmado e implementarlo.

Adicionalmente Bancolombia también implementa DevOps en sus equipos de trabajo, el cual está enfocado en que los equipos de desarrollo y operaciones colaboren entre sí en la creación, certificación, implementación y monitoreo de aplicaciones con potenciando la velocidad, calidad y control de las mismas en el proceso [7]. Por esta razón Bancolombia también implementa roles en los equipos de trabajo del área de operaciones, tales como:

- **Ingeniero de Plataformas:** Se encarga de administrar la infraestructura de las aplicaciones y tiene comunicación constante con el **Team** para la implementación de cambios o creación de nuevas implementaciones, ya sea realizando las implementaciones el mismo o apoyando en la monitorización de implementaciones automáticas.
- **Administrador de Base de datos:** Se encarga de administrar las bases de datos de las aplicaciones, tiene una comunicación constante con el **Team** para la implementación de cambios en base de datos, creación de nuevos elementos o características, monitorización, ejecución de scripts y más.
- **Ingeniero de Seguridad:** Se encarga de mantener seguras las aplicaciones al momento de su interacción, administran quienes tienen acceso y quienes no a las aplicaciones o sus bases de datos mediante permisos, VPN, administración de reglas de firewall y más. Tiene comunicación con el **Team** a la hora de realizar cambios de accesibilidad a las aplicaciones o sus bases de datos.

Así mismo, Bancolombia implementa ciertas ceremonias o reuniones de los marcos ágiles en búsqueda de lograr cumplir con los propósitos de los equipos de tecnología, las cuales son:

- **Planning:** Reunión en la cual se prioriza que historias de usuario se trabajarán (según el criterio del **Product Owner**) en un periodo de tiempo no muy largo llamado sprint.
- **Daily:** Reunión diaria en la que cada uno de los miembros del **Team** le cuenta a los demás como va en el desarrollo de las historias de usuario

que tiene asignadas y si tiene impedimentos para cumplirlas. El objetivo de la reunión es que en un periodo corto de tiempo se sincroniza el equipo y en caso de que alguno tenga una forma de eliminar los impedimentos del otro puedan hacerlo.

- **Review:** Reunión que se realiza al finalizar el sprint, su objetivo es que las personas del **Team**, le exponen al **Product Owner** el resultado del trabajo realizado en el sprint, basados en las historias de usuario comprometidas en el planning buscando la aprobación de estos por parte del usuario.
- **Retrospectiva:** Reunión con la que se finaliza el sprint, en ella el **Scrum Master** prepara una serie de actividades en las cuales el equipo pueda cambiar un poco de ambiente y divertirse, pero al mismo tiempo buscando que la mejor continua del equipo, que este se autoanalice y revise que cosas hizo bien, qué cosas hizo mal y cómo mejorarlas.

En Bancolombia, los equipos de tecnología pueden trabajar en uno o más proyectos, los cuales están definidos como alguno de los siguientes tipo:

- **Iniciativas:**
 - **Express:** Son proyectos de corta duración y se clasifican en dos de acuerdo a esta de la siguiente manera:
 - **Normal:** Su duración es de 3 sprint (45 días)
 - **Fastrack:** Su duración es de 1 sprint (15 días)
 - **Proyecto:** Son proyectos de gran envergadura y su duración oscila entre 3 meses hasta años.
- **Mantenimientos:**
 - **Soporte Estructural:** Son mejoras que se le hacen a los aplicativos del área por iniciativa del mismo equipo, no por pedido de los usuarios, normalmente se realizan pensando en evitar problemas a futuro
 - **Asesorías:** Estas se realizan cuando el usuario solicita un acompañamiento o aclaramiento de información por parte del equipo de trabajo
 - **Incidentes:** Son los errores que reportan los usuarios de los aplicativos soportados por el área.
 - **Casos problema:** Son incidentes que reporta demasiada gente y muy seguido, por lo cual requieren de mayor atención.
 - **Jubilaciones:** Estas se realizan cuando un aplicativo ya no se usa más o se piensa dejar de usar, consiste en realizar todas las acciones necesarias para esto, como apagar el servidor.Net donde está alojado, eliminar dependencias de otros aplicativos etc.

Las aplicaciones que entran en el alcance manejan las siguientes tecnologías:

- **.Net:** Framework de Microsoft que permite la creación y ejecución de aplicaciones y servicios web de Windows, proporcionando un lenguaje de programación orientado a objetos, entornos de ejecución de código y los mejores estándares de la industria [8].
- **C#:** Lenguaje de programación orientado a objetos que permite crear aplicaciones que se ejecutan bajo el ecosistema del framework **.Net**, muy parecido a C, C++ o Java [9].
- **Windows forms:** Interfaz de programación que hace parte del Framework de **.Net**, es usado para crear aplicaciones stand alone o cliente servidor en sistema operativo Windows, ya que estas no se consumen mediante el navegador, sino que deben instalarse en el equipo que van a ser usadas, el lenguaje de programación es el más básico del framework **C#** [10].
- **Entity Framework:** Framework para realizar la persistencia de datos de una aplicación bajo el framework de **.NET**, permite hacer consultas LINQ, seguimiento de cambios, actualizaciones, mapeo de objetos, migraciones de esquemas, funciona con varios motores de bases de datos tales SQL Server, SQLite, MySQL, PostgreSQL y más [11].
- **SQL Server:** Sistema de administración de bases de datos relacionales de Microsoft, este posee un lenguaje propio derivado de SQL llamado Transact SQL que permite interactuar y administrar las bases de datos. Funciona exclusivamente en ambientes Windows [12].
- **Asp.net:** Es una interfaz de programación que hace parte del framework de **.Net**, es usado para crear aplicaciones web, tiene su propio lenguaje basado en HTML para el Front-End y un code behind que maneja la lógica de cada formulario web en el lenguaje básico del framework **C#** [13].
- **Oracle:** Sistema de administración de base de datos de la empresa Oracle que soporta base de datos relacionales, NoSql y MySql [14].

Para poder lograr el objetivo de dar soporte y mantener estable las aplicaciones y su ecosistema se cuenta con las siguientes herramientas:

- **Visual Studio:** Entorno de desarrollo integrado profesional de Microsoft con el que se puede editar, depurar y compilar código, y luego publicar una aplicación. Incluye compiladores, herramientas de diseñadores gráficos, compatibilidad con gran cantidad de lenguajes e integración con herramientas de versionamiento tales como git y muchas cosas más. Necesita licencia para ser usado en compañías [15].
- **SQL Server Management Studio:** Entorno integrado para administrar cualquier infraestructura SQL. Proporciona herramientas para configurar y administrar las instancias de Bases de datos que posee, integración con Transact-SQL y posibilidad de implementar los componentes de la capa de datos, crear consultas y scripts y mucho más [16].
- **Oracle SQL Developer:** Entorno de desarrollo integrado y gratuito para la gestión de bases de datos Oracle en implementaciones

tradicionales y en la nube. Ofrece la posibilidad de realizar desarrollos en el lenguaje PL/SQL, creación de consultas y scripts, consola para administración de base de datos, generación de informes, modelado de datos y mucho más [17].

- **Azure DevOps:** Plataforma de microsoft que sirve para planificar y organizar los equipos de una organización, permite creación de equipos de trabajo, planeación de sprint ya que permite creación de los ítems de trabajo del mismo tales como historias de usuario o tareas, también permite la implementación de DevOps mediante la implementación de Integración continua y Despliegue continuo, permite obtener y usar repositorios privados e ilimitados de git alojados en la nube permitiendo así una gestión de ambientes, permite realizar despliegues tanto en los ambientes pre productivos como en el de producción de manera automática y semiautomática, permitiendo el uso de tareas de rollback en caso de haber fallos, integración con gran cantidad de herramientas y mucho más[18].

Metodología

Cada mes los líderes del área definen una agenda maestra donde definen y dan prioridad a los objetivos que se trabajarán por el equipo.

Para llegar al cumplimiento de los objetivos, se definieron sprint de 15 días, donde los miércoles cada 15 días se hacía el **planning**, de máximo 2 horas en la mañana, los martes anteriores a dicho día se realizaba tanto el **review** y la **retrospectiva** en la tarde, el daily se realizó diario al comenzar el día.

Primer proyecto

Fue uno de tipo **Express FastTrack**, en el cual se necesitaba generar un informe que permitiera entre un rango de fechas consultar los principales inversionistas para cada día entre ese rango del aplicativo que administra fondos de inversión. Este al ser un proyecto legal, tenía poco tiempo para su resolución y este no era modificable, por lo cual se negociaron tiempos extras. Se llevó a cabo en 2 sprints donde se realizaron los pasos necesarios para finalizar satisfactoriamente la implementación de este cambio en producción **Figura 1**.

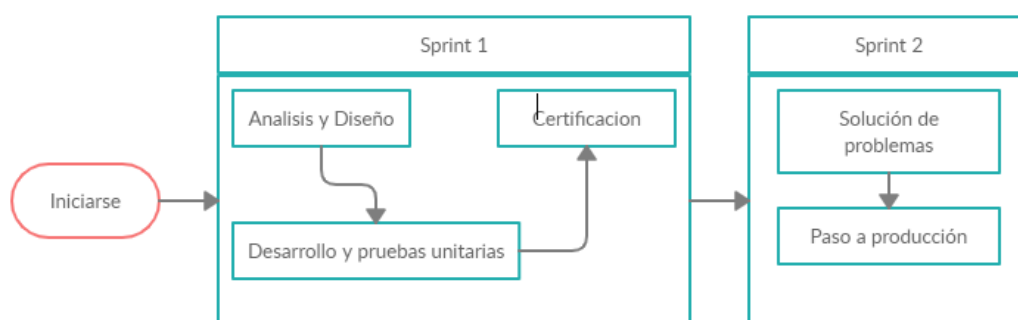


Figura 1. Fases del proyecto Principales inversionistas del aplicativo de fondos de inversión

Análisis y Diseño: En esta fase se recibió una explicación corta de la necesidad por la premura del proyecto. Primero se analizó que este aplicativo está principalmente bajo el framework de **.Net** y posee un modelo multicapa. Para el desarrollo solo se necesitó el uso de 4 capas. La capa Web donde se encuentra la interfaz gráfica del aplicativo en **Asp.net** como el code behind para la lógica de la interfaz desarrollada en **C#**, la capa Negocio, donde se encuentra la lógica de negocio desarrollada en **C#**, la capa de Datos desarrollada en **C#** que hace la persistencia de los mismos mediante librerías propias de Bancolombia a una Datos **Oracle** y la capa Test donde van las pruebas unitarias desarrolladas en **C#** y usando la librería **NUnit** para las mismas. La comunicación entre las capas es como puede verse en la **Figura 2**.

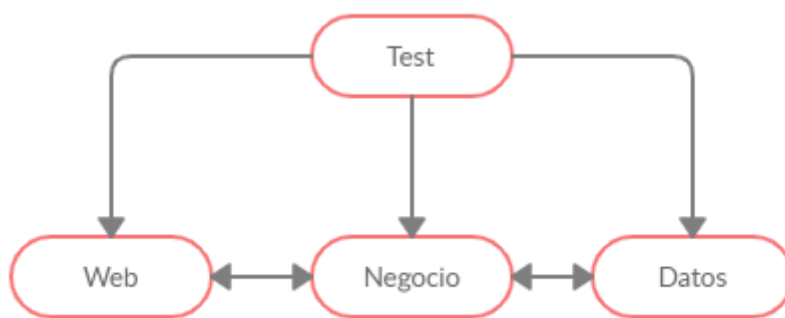


Figura 2. Capas de aplicativo que administra fondos de inversión.

Con el **Product Owner** se llegó a la conclusión que debía desarrollarse una nueva pantalla en el aplicativo que permitiera ingresar la fecha inicial del rango, la fecha final y el fondo a consultar, finalmente estos datos de entrada se llevarían por las capas de la aplicación hasta la de persistencia donde debía desarrollarse un procedimiento almacenado que realizará las consultas y cálculos necesarios y al finalizar devolviera un dataset con el resultado de la consulta para ser exportado a un archivo de texto en la ubicación que el usuario escogiera.

Desarrollo y pruebas unitarias: En esta fase se comenzó primero con el desarrollo del procedimiento almacenado mediante la herramienta **SQL Developer** en el lenguaje PL/SQL, ya que la base de datos es **Oracle**, y se consultaron con los expertos del aplicativo que datos de prueba usar, así como los estándares que debían usarse, entre los cuales tienen un estándar para retorno de mensajes de error mediante códigos numéricos y también una tabla log para guardarlos. Al estar esta parte finalizada se comenzó con el desarrollo de la interfaz web que permitiera ingresar los rangos de fechas y el fondo a consultar mediante la herramienta **Visual Studio**, esta se creó con los estándares del aplicativo, el cual no usa ningún framework visual solo **Asp.net**, se obtuvo acceso al repositorio de los fuentes mediante **Azure DevOps**, se creó una nueva rama para trabajar basada en desarrollo, se clonó localmente para trabajarla con **Visual Studio**, en dicho momento se

descubrió que era necesario listar los fondos disponibles para consulta en una lista desplegable, por lo cual se creó un procedimiento almacenado que listara los fondos y los retornara al aplicativo y así listarlos mediante los controles de **AspNet**. Para las fechas se usaron controles del **AspNet** que permitieran desplegar una interfaz para elegir la fecha y un botón para generar el informe, el cual envía la información desde la capa web hasta la de datos así consumiendo el procedimiento almacenado y retornando la información en un dataset el cual mediante una utilidad que ya se tenía en el aplicativo, se pasaba a archivo de texto para la posterior descarga del mismo por parte de los usuarios. Finalmente las herramientas para paso entre ambientes de Bancolombia para este aplicativo exigen que mínimamente el 50% del código nuevo esté cubierto por pruebas unitarias, porcentaje que se valida mediante **sonarqube** al momento de intentar hacer pull request desde nuestra rama a la rama de desarrollo mediante **Azure Devop**, por la premura se desarrollaron una pequeña cantidad de pruebas que incluían validaciones de fechas, retorno correcto de fondos y validaciones en consulta de principales inversionistas, con lo cual se cubrió un 60% más que suficiente para pasar a certificación.

Certificación: En esta fase luego de pasar a desarrollo y desplegar correctamente en el ambiente de desarrollo, se realizaron pruebas de desarrollo y finalmente se pasó al ambiente de certificación el cual está compuesto por las siguientes tareas de pipeline **Figura 3**.

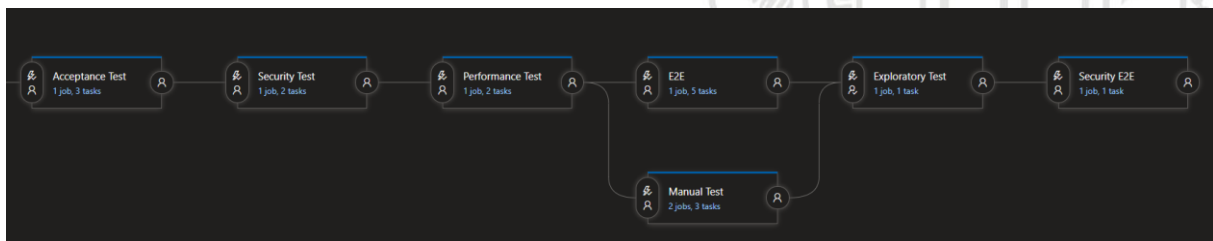


Figura 3. Pipeline de certificación, aplicativo que administra fondos.

Si bien la primera tarea se despliega en el ambiente de certificación y se pudo realizar las pruebas junto a la persona encargada de certificación las cuales fueron exitosas, en este punto hubo varios problemas que se describen un poco más a continuación.

Solución de problemas: Hubo bastantes problemas en la integración continua ya que se encontraba mal configurada por unos cambios que habían realizado, esto nos hizo perder mucho tiempo y fue resuelto por el área de DevOps, luego en el stage de pruebas de aceptación también falló porque no detectaba las pruebas, problema también resuelto por el área de DevOps y paso lo mismo con las pruebas de performance. En las pruebas E2E estaban realizando unos cambios que también nos afectaron, todo esto resultando en tiempo extra de parte nuestra de las personas que estaban realizando los cambios.

Paso a producción: Finalmente pudo realizarse paso a producción donde es necesario usar unas herramientas de Bancolombia para la implementación de Base de datos, debe gestionarse la ayuda de un **Ingeniero de Base de Datos** para la implementación del query y **Azure Devops** para la instalación de aplicación, el **Ingeniero de Plataformas** para el visto bueno, backup y apoyo en caso de fallas para la parte aplicativa. Debe gestionarse unas llamadas “Órdenes de cambio” donde es registrado el cambio que se va a realizar, a qué horas, cuánto dura, qué impacto tiene, si falla qué plan hay para arreglarlo, manual de implementación y otros elementos para asegurar que este no falle y estar preparados para una posible visita de auditoría. Finalmente, el paso se realizó correcta y satisfactoriamente logrando entregar los informes a la superintendencia financiera.

Segundo proyecto

Proyecto de tipo **Express Fastrack** para cambio en informe de rentabilidades para operaciones de cancelación de fondo también en el aplicativo que administra fondos, este proyecto no estaba planeado en la agenda maestra, pero entro con una prioridad alta ya que era otro tema legal por lo cual debió ser atendido inmediatamente y se trabajó en 2 sprint como se ve en **Figura 4**.

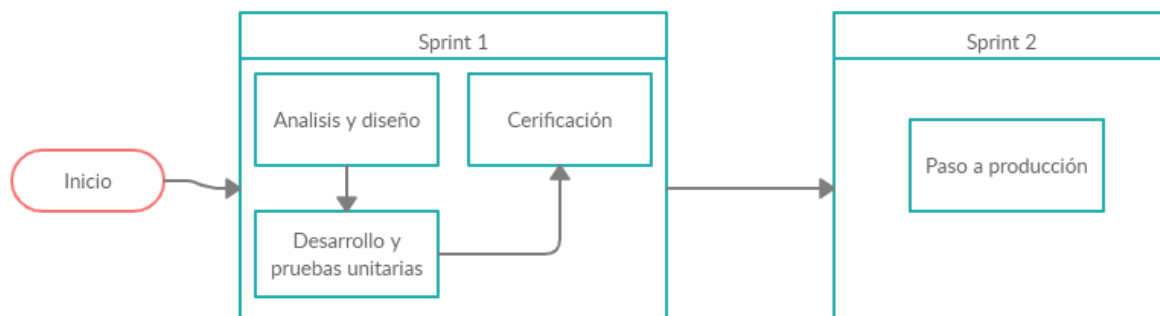


Figura 4. Fases del proyecto Cambio en informe de rentabilidades para operaciones de cancelación de fondo.

Análisis y Diseño: En esta fase se recibió la contextualización del problema, el cual era que, al cancelar fondos, se estaba cobrando dos veces un impuesto por cancelación, adicionalmente se estaban pidiendo unos cambios de ortografía en el informe. Se realizó el análisis y se detectó que el problema se encontraba en un query que realizaba el cobro duplicado, adicionalmente se revisó el código que llenaba el informe para detectar dónde estaban los errores de ortografía.

Desarrollo y pruebas unitarias: Luego de los resultados de la fase de análisis el desarrollo fue bastante sencillo, solo fue necesario borrar la parte del query que generaba el cobro doble, esto sucedía en un procedimiento almacenado en la base de datos **Oracle**, por lo que el cambio se realizó con la herramienta **SQL Developer**. Finalmente se creó una nueva rama de trabajo basada en el ambiente de desarrollo del aplicativo mediante la

herramienta **Azure DevOps** y se clonó localmente para poder trabajar en ella mediante la herramienta **Visual Studio** y se realizaron los cambios de ortografía que eran pocos. Cabe anotar que ya que los cambios no generaron código nuevo no fue necesario desarrollar pruebas unitarias. Se realizó el paso a desarrollo, donde se hicieron las pruebas en dicho ambiente antes de pasar a certificación.

Certificación: Las pruebas se realizaron rápidamente por parte de la persona encargada de la certificación donde no hubo ningún problema en ellas, adicionalmente gracias a los problemas resueltos en el proyecto anterior no hubo problemas en integración continua, pruebas de aceptación, pruebas de performance ni en pruebas E2E.

Paso a producción: El paso a producción no tuvo complicaciones, se usó la herramienta de Bancolombia para pasar el cambio en el query de base de datos, el **Ingeniero de Base de Datos** para la implementación del query y mediante **Azure Devops** se realizaron los cambios de aplicación, el **Ingeniero de Plataformas** para el visto bueno, backup y apoyo en caso de fallas para la parte aplicativa. Se generaron las llamadas "Órdenes de cambio" para registrar todos los temas del paso, el cual se llevó a cabo satisfactoriamente y el paso no duró más de 1 hora.

El tercer proyecto

Fue un **Soporte Estructural**, para arreglar problemas en informe de Balance Global del aplicativo que administra fondos, este nació ya que encontraron descuadres en la información generada por el informe y se tenía la sospecha de que fueron errores de cálculo. Las fases del proyecto se llevaron a cabo como se ve en la **Figura 5**.

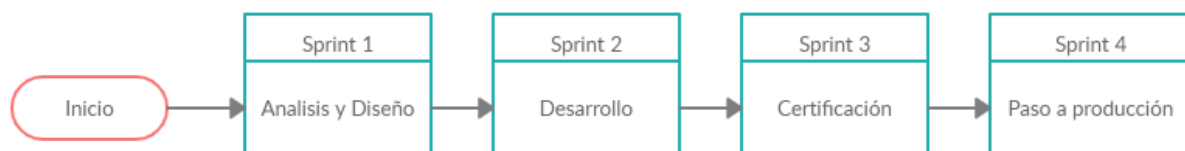


Figura 5. Fases de proyecto descuadre en el informe de Balance Global.

Análisis y Diseño: En esta fase se recibió contextualización del problema, donde se evidenciaba que había un descuadre en el informe que era consultado una vez al mes. Este análisis fue largo y tomó un sprint completo donde dio como resultado que el descuadre no era un error de cálculo, sino que no estaban siendo retornados en la consulta los registros de inversionistas que no poseían asesor, lo cual no debería ser un impedimento, se detectó que el problema estaba en la consulta de un procedimiento almacenado en la base de datos **Oracle**.

Desarrollo: En esta fase se comenzaron a realizar pruebas en búsqueda de solucionar el problema mediante la herramienta **SQL Developer**, ya que en la

fase de análisis se detectó que el problema era en un inner join que realiza la conexión entre dos tablas de base de datos cuando coinciden en un campo que hace de clave principal en una y en la otra de clave foránea, en este caso una tabla de inversionista con la tabla de asesores, el problema radicaba en que algunos inversionistas al no tener asesor, no tenían registro en la tabla de asesores por lo cual la consulta los descartaba, esto se solucionó mediante un left join, el cual sin importar que no hubiera un registro que hiciera match en la tabla de asesores con uno de la tabla de inversionistas, todos los inversionistas serían tomados en cuenta. En esta fase se realizaron bastantes pruebas en ambientes de desarrollo ya que la consulta debe trabajar con muchos registros. Finalmente se realizó el paso al ambiente de certificación mediante la herramienta de Bancolombia para realizar versionamiento de objetos de Base de datos.

Certificación: En esta fase se apoyó a la persona de certificación para que realizara las pruebas y cálculos pertinentes para probar el correcto funcionamiento del cambio, las cuales eran bastantes por lo que se llevó un sprint completo.

Paso a producción: En esta fase se realizó el paso a producción sin complicaciones, se usó el aplicativo de Bancolombia para el versionamiento de objetos de bases de datos, se gestionó el acompañamiento del **Ingeniero de Base de Datos** para la implementación del cambio y se creó la respectiva "Orden de Cambio" para legalizar y llevar a cabo la instalación en producción.

Proyecto principal

Finalmente, el proyecto en que se trabajó la mayoría del tiempo de prácticas y principal fue el **Soporte Estructural** del aplicativo que administra acciones donde se buscaba darle solución a diversos temas encontrados por los usuarios el año pasado. Este soporte duro todo el tiempo de las prácticas y se detenía cuando llegaban proyectos de más urgencia o incidentes. Los temas trabajados fueron 3 y se describen a continuación:

- 1. Problemas para unificar accionistas que tuvieron bloqueos:** En el aplicativo existe una funcionalidad que permite unificar en un nuevo accionista la información de uno o más accionistas, esto generalmente usado para accionistas extranjeros que se nacionalizaron o accionistas que estaban como menores de edad y ya han cumplido la mayoría de edad, el error era que al tener bloqueos no se puede ser parte de una unificación, pero luego al quitarle el bloqueo seguía diciendo que tenía y no lo dejaba unificarse. el proyecto se desarrolló en 2 sprints como se ve en la **Figura 6**.

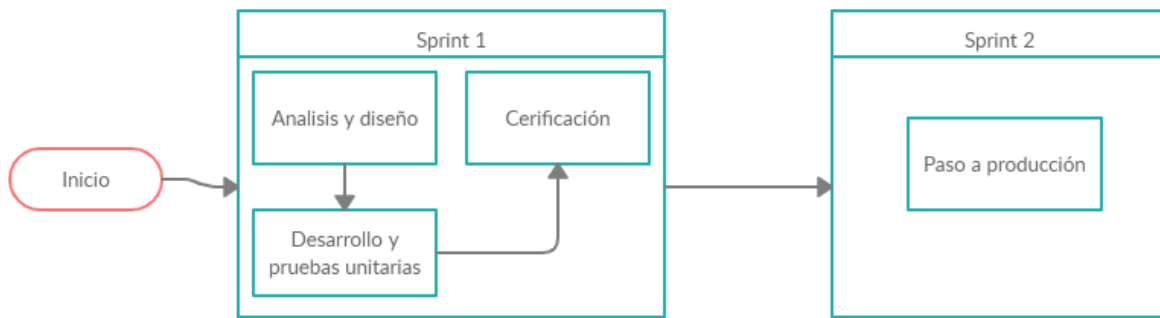


Figura 6. Fases para resolución de tema de problemas para unificar accionistas que tuvieron bloqueos.

Análisis y diseño: En esta fase se habló con el **Product Owner** donde se llegó y se mostró en el ambiente de producción el problema, por lo cual se intentó replicar en el ambiente de desarrollo y donde pudo verse que si sucedía y que el problema era en la capa de persistencia de datos. El aplicativo está desarrollado bajo el framework **.Net** bajo un modelo multicapa. Para la capa de interfaz gráfica llamada “presentación” se usó **Windows Forms** ya que es un aplicativo cliente servidor donde se tiene toda la lógica de interfaz en **C#**, tiene una capa de negocio, en ella se encuentra todas las reglas de negocio, se encuentra desarrollada en **C#**, tiene una capa de modelos donde se encuentran los objetos de negocio que hacen el símil con la realidad en el aplicativo y en ellos se mapea la información de las tablas de base de datos para ser interactuada en el aplicativo también desarrollada esta capa en **C#**, finalmente la capa de datos que hace persistencia de datos con la base de datos **SQL Server** mediante el framework de microsoft **Entity Framework** con el cual se mapean los registros de las tablas directamente a objetos de la capa de modelos, la capa está desarrollada en **C#** y finalmente la capa test en la que se desarrollan las pruebas unitarias en **C#** usando la librería NUnit. La interacción entre las capas puede verse en la **Figura 7**.

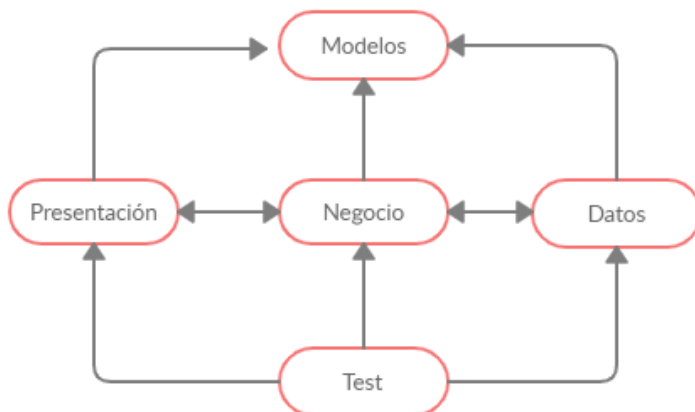


Figura 7. Capas de aplicativo que administra acciones.

Finalmente se llegó a la conclusión que el problema se encontraba en una consulta a la base de datos que se hacía en el aplicativo mediante el **Entity Framework** ya que esta tomaba los bloqueos inactivos o deshabilitados que habían tenido los accionistas como vigentes.

Desarrollo y pruebas unitarias: Primero se accedió mediante la herramienta **Azure DevOps** al repositorio del aplicativo, se creó una rama basada en la de desarrollo para trabajar y se clonó localmente para trabajar sobre ella mediante la herramienta **Visual Studio**, allí en la capa de datos se identificó el método donde se consultaba si un accionista era apto para una unificación o no y se detectó que no se revisaba el campo que indicaba si el bloqueo estaba activo o no, solo se validaba que tuviera un registro de la tabla de bloqueos, se le agregó esta condición en el where y se realizó el paso al ambiente de desarrollo, se hicieron las pruebas pertinentes y se procedió a pasar los cambios al ambiente de certificación.

Certificación: En esta fase se apoyó a la persona de certificación en las pruebas donde se probó en accionistas sin bloqueos, con bloqueos, con bloqueos inactivos, con bloqueos activos en otras empresas y bloqueos inactivos en otras empresas, donde hubo un resultado satisfactorio en todas.

Paso a producción: El paso a producción ya que solo involucró la parte de aplicación se coordinó con el Ingeniero de plataformas su apoyo para dar su visto bueno, realizar backup y se realizó el paso a través de la herramienta Azure DevOps, se gestionó la Orden de Cambio respectiva para el cambio y no hubo ningún tipo de complicaciones.

- 2. Accionistas que tienen el dividendo inicial pendiente en estado cancelado:** En el aplicativo se realizan pagos de dividendos a los accionistas de cada empresa, en este caso el problema era que había accionistas con dividendos pendientes, que salían como cancelados por lo cual no se podían pagar. Este problema es un llamado "Cambio de data en producción" para el cual en Bancolombia se tiene un flujo diferente por temas de auditoría y se manejó como puede verse en la **Figura 8**.

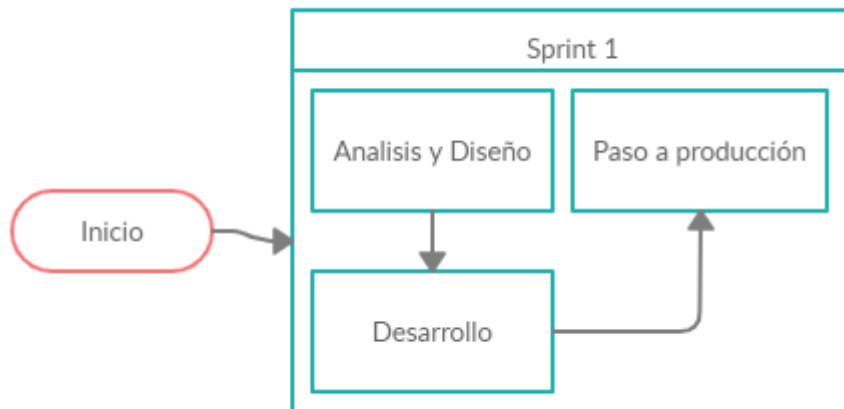


Figura 8. Fases para resolución de tema de accionistas con dividendo inicial pendiente en estado cancelado.

Análisis y Diseño: En esta fase como siempre se tuvo una contextualización con los usuarios donde se nos explicó que un dividendo inicial, el cual es con el que el accionista se crea por primera vez en el aplicativo, no puede estar cancelado, solo si se ha muerto por lo cual debe tener una operación de sucesión asociada, ya que se cancela el dividendo inicial y se crea una copia para el heredero, pero había accionistas que no había sido parte de una operación de sucesión y aun así tenían este dividendo cancelado, se realizó un análisis exhaustivo pero no se pudo encontrar la causa raíz, por lo cual se llegó a la conclusión que solo debía arreglarse los datos en los que se presentaba actualmente, lo cual es llamado un “Cambio de datos en producción”.

Desarrollo: Para el desarrollo, tuvo que solicitar acceso solo de consulta a las bases de datos de producción para acceder mediante la herramienta **SQL Server Management Studio**. De esta manera se desarrolló un script que permitiera buscar todas las personas que tuvieran un dividendo inicial cancelado y no hayan sido parte de una sucesión, así cambiándoles el estado de cancelado a pendiente.

Paso a producción: Para los cambios de data de este tipo no se realiza certificación, ya que los ambientes no tienen los mismos registros y datos por lo cual no sería una prueba fiable. En cambio, se debe preparar un script que permita volver al estado anterior en caso de haber problemas en la implementación, además debe gestionarse con el área dueña de los datos (en nuestro caso el área de fiduciaria de Bancolombia) que den un visto bueno mediante correo electrónico el cual debe adjuntarse en los documentos de preparación para pasar a producción. Se utilizó la herramienta de Bancolombia para el versionamiento de objetos de base de datos y se coordinó con el **Ingeniero de Base de Datos** la implementación del script y se realizó el cambio sin complicaciones.

3. Ajuste en traslado de dividendos luego de una sucesión: En el aplicativo, luego de que se realiza una operación de sucesión donde la persona que murió le entrega sus acciones a sus herederos, existe una funcionalidad adicional que permite también trasladar los dividendos pendientes a los herederos. En este caso dicha funcionalidad había generado un error, trasladando a los herederos de la sucesión el dinero de todos los demás, lo cual generó un descuadre en toda la operación. Este tema se trabajó en 6 sprint y las fases se pueden observar en la **Figura 9**.

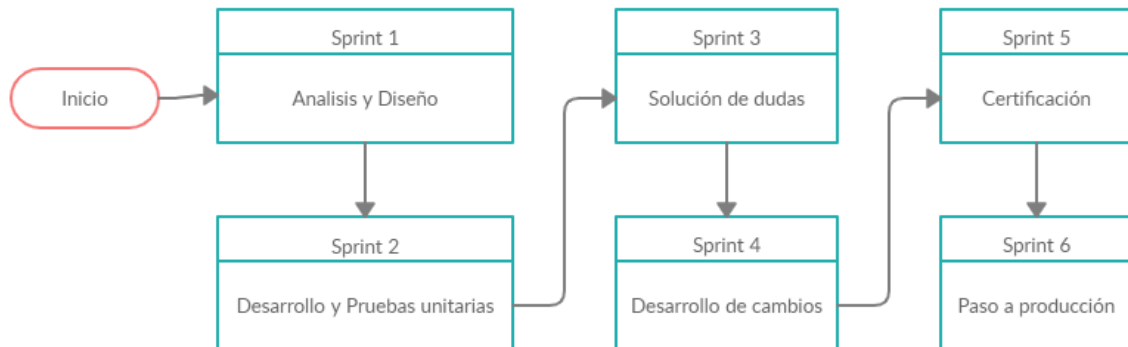


Figura 9. Fases para resolución de tema ajuste en traslado de dividendos luego de una sucesión.

Análisis y Diseño: En esta fase se tuvo la contextualización del usuario en cuanto al problema presentado que se presentó en dos sucesiones, se replicó en el ambiente de desarrollo y se determinó que el problema se encontraba al momento de repartir los dividendos en la funcionalidad ya que al primer heredero le estaba dando los dividendos de todos los demás.

Desarrollo y Pruebas unitarias: En esta fase se accedió al repositorio del aplicativo a través de **Azure DevOps** se creó una rama basada en desarrollo y se clonó localmente para trabajar con ella a través de la herramienta **Visual Studio**. Se realizó el cambio ya que al calcular los dividendos a trasladar se sacaba un porcentaje de acuerdo a las acciones heredadas sobre las totales del fallecido para elegir el valor de dividendos que le tocaba a cada heredero, pero se estaba calculando erróneamente, se realizó el cambio a nivel de aplicación. Se crearon las pruebas unitarias que validaron que se calculará correctamente el valor de los dividendos y se procedió a implementar los cambios en el ambiente de certificación usando **Azure DevOps**.

Solución de dudas: En las primeras revisiones en el ambiente de certificación se generaron dos dudas, la primera era si una persona al fallecer tenía dos sucesiones no debería de darse todo el pendiente en la primera sucesión y la segunda que pasaría con los datos que ya se

encontraban con errores en producción. Para resolverlas se habló con el **Product Owner** con el que se llegó a la conclusión de que debía realizarse un cambio adicional y era lograr que los usuarios pudieran elegir los dividendos que se iban a trasladar en cada operación de traslado. Para el tema de los datos ya afectados en producción se llegó a la conclusión que debía hacerse un cambio de data que reconstruyera los dividendos de los accionistas afectados.

Desarrollo de cambios: En esta fase se desarrolló usando el **Visual Studio** en la funcionalidad de traslado de dividendos un grid donde se visualizan los dividendos pendientes con un check que permite elegir cuales se van a trasladar y cuáles no, para la parte de corregir los datos se pidieron permisos de consulta a producción para acceder a través de la herramienta **SQL Server Management Studio** y se generó un script que repartiera correctamente los dividendos a los accionistas afectados. Se crearon las pruebas unitarias que permiten validar que se listen en el grid los dividendos pendientes del fallecido y se realizó el paso a certificación usando la herramienta **Azure DevOps**.

Certificación: En esta fase se apoyó a la persona de certificación en las pruebas, donde se probó con sucesión única y con dos o más sucesiones, trasladando todo el pendiente y trasladando solo un poco. Finalmente validando que el funcionamiento era el correcto.

Paso a producción: Para el paso a producción en la parte de aplicación se realizó mediante Azure DevOps con la ayuda de un Ingeniero de Plataformas que como siempre ayuda dando el visto bueno para el paso, realizando backup y apoyando en caso de que haya algún problema. Para la parte de Base de datos como era un cambio de data se gestionó con el área de fiduciaria de Bancolombia el visto bueno para realizar el cambio de data en producción y se usó el aplicativo de Bancolombia para versionamiento de objetos de base de datos para pasar el script y se gestionó la ayuda de un ingeniero de Base de Datos para la implementación del cambio en producción donde todo salió según lo esperado.

Resultados y análisis

Se obtuvo un gran conocimiento del área de negocios fiduciarios y de los aplicativos que allí se usan. Como área de tecnología se aprendió bastante de las tecnologías que allí manejan:

- Se aprendió sobre el funcionamiento aplicativo que administra fondos, creación de pantallas en **Asp.Net**, por ejemplo, en el proyecto donde se creó una pantalla para consultar los inversionistas principales (Ver **Figura 10**), donde a pesar de la sencillez del formulario se aprendió sobre los estándares manejados en el aplicativo, sus capas, así como su usabilidad. Lo cual implicó que en los demás proyectos fuera más sencilla la curva de aprendizaje

VALORES BANCOLOMBIA S.A

Consulta de Principales Inversionistas

Fondo: RENTA LIQUIDEZ VALORES BANCOLOMBIA ▾

Fecha Inicio: 04/10/2018

Fecha Fin: 04/11/2018

Figura 10. Pantalla para consultar los principales inversionistas de un fondo en un rango de fechas.

- Se desarrollaron conocimientos sobre el uso de la herramienta **Sonarqube**, como revisar la calidad del código y así generar planes para la refactorización del mismo para aportar mejoras, donde si bien en la actualidad el aplicativo no es el más limpio ni tiene las mejores prácticas ya que lleva varios años de desarrollado (**Ver Figura 11**), poco a poco se pueden arreglar los problemas en la misma teniendo conciencia de ello gracias a el análisis de código estático que nos brinda esta herramienta.

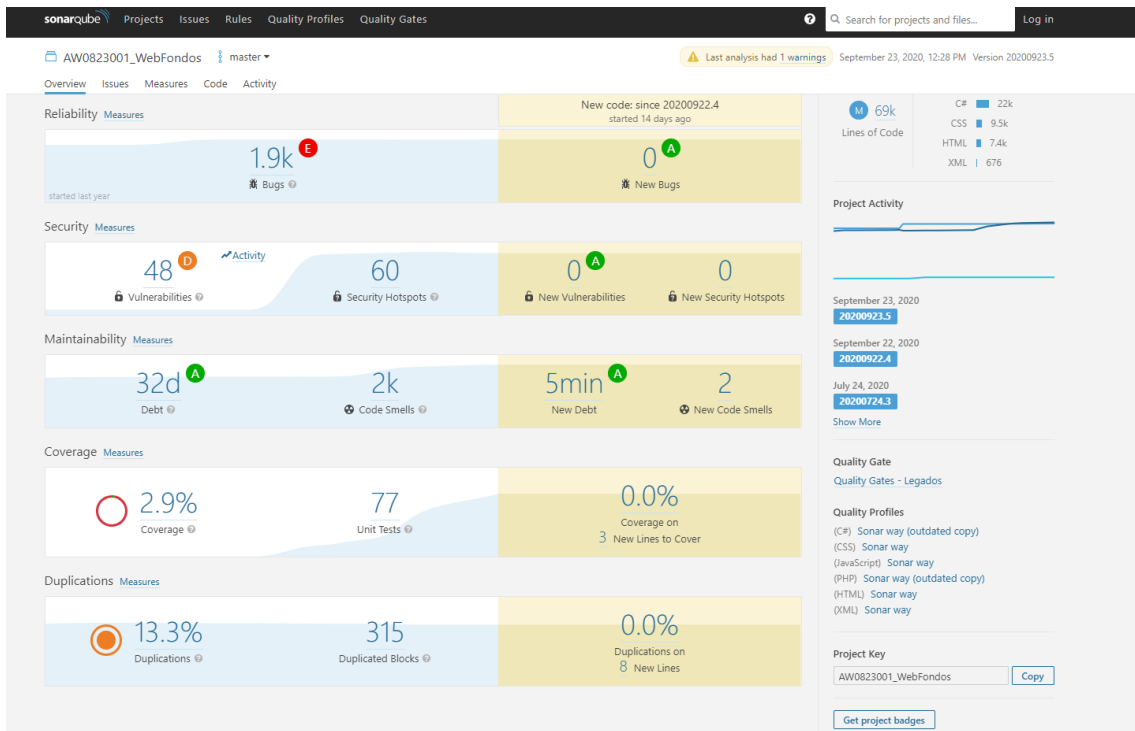


Figura 11. Análisis de código estático aplicativo que administra fondos Bancolombia

- Se conoció la herramienta **Azure DevOps** la cual fue usada tanto para gestionar los sprint (**Ver Figura 12**), como para manejo de repositorios (**Ver Figura 13**) y despliegue en los ambientes (**Ver Figura 14**)

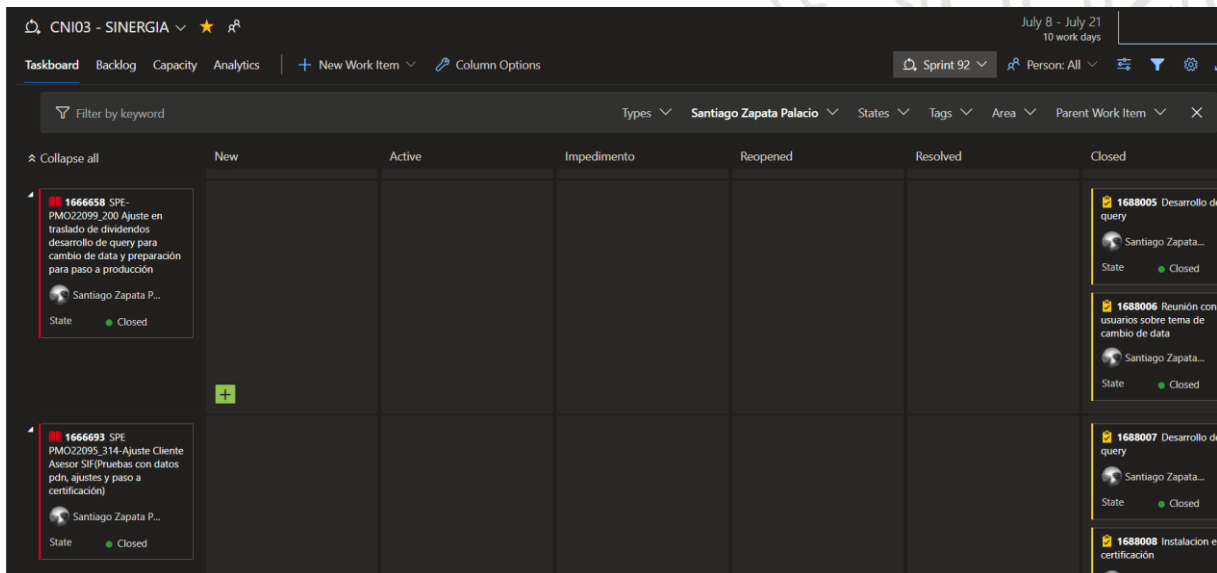


Figura 12. Gestión de historias de un sprint a través de Azure DevOps

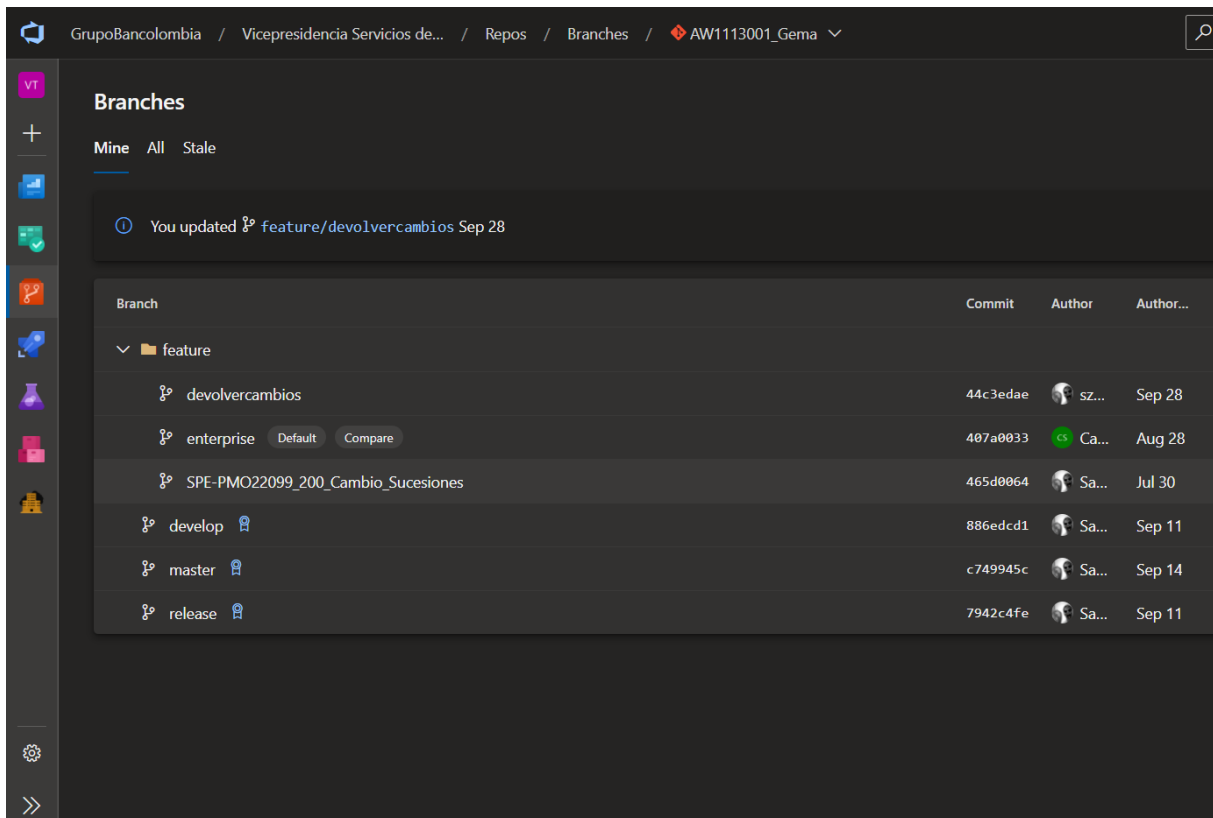


Figura 13. Gestión de repositorios y ramas a través de **Azure DevOps**

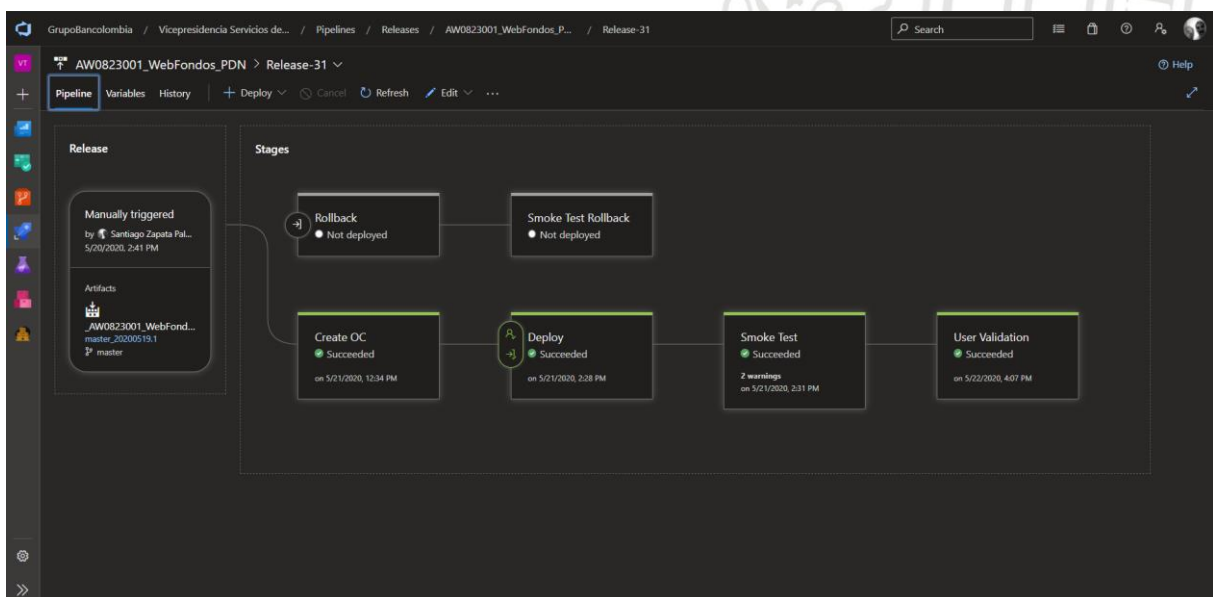


Figura 14. Despliegue exitoso en producción del aplicativo que administra fondos a través de **Azure DevOps**.

- Se cultivaron habilidades para gestionar incidentes, resolverlos, negociarlos, diagnosticarlos, así como Órdenes de Cambio para legalizar cambios en producción, lo cual es indispensable en un área de soporte. Dando como resultado un aumento en el número de incidentes y temas que se resuelven por sprint como se puede observar en la **Tabla 1**. Donde se ve que con el pasar de los sprint se lograron

atender una mayor cantidad de incidentes. Si bien el número de incidentes que llegan cada Sprint depende de los usuarios y los problemas que encuentren la capacidad de atender más cada Sprint es de la persona del equipo de tecnología que los resuelve.

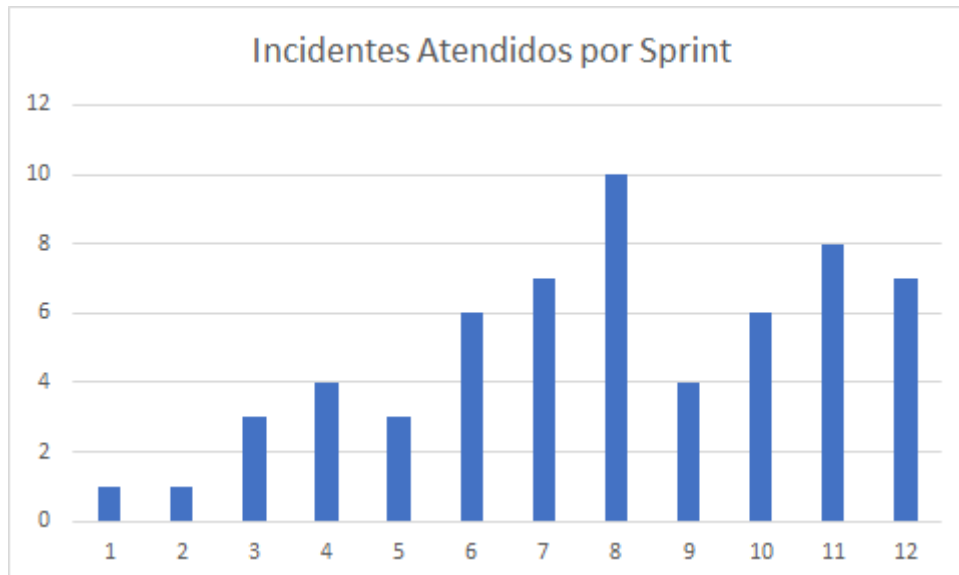


Tabla 1. Incidentes atendidos por Sprint.

Conclusiones

Al finalizar el desarrollo de la práctica se logró:

- Cumplimiento de los objetivos específicos, por ende, con el objetivo general.
- Obtención de conocimiento en herramientas para análisis de código estático como **Sonarqube** y usarlas para refactorizar código y mejorarlo.
- Implementación cambios en las aplicaciones mediante soportes estructurales, proyectos express e incidentes donde se buscó la solución raíz a los mismos aprendiendo a usar las herramientas **Visual Studio, SQL Server Management Studio, SQL Developer** y desarrollando skills en el framework de **.Net**.
- Desarrollo de conocimientos de gestión para realizar cambios entre ambientes, a gestionar las tareas y elementos de cada sprint y a gestionar los repositorios usando la herramienta **Azure DevOps**.
- Se realizaron análisis previos a los proyectos para definir los alcances de los mismos y se aplicaron los principios básicos de la metodología Scrum.
- También se demostró que el cumplimiento de todos los objetivos generó objetivos habilidades tanto blandas como técnicas que permitieron que cada Sprint logrará de forma más natural la atención

de incidentes, sin dejar de lado los proyectos planteados, logrando así un aumento en el número de incidentes atendidos por Sprint.

Como trabajo futuro queda:

- Seguir apoyando la herramienta **Sonarqube** para lograr un código y arquitectura más limpia en los aplicativos y así mejorando la calidad y el mantenimiento del mismo.
- Enfrentarse a proyectos como líder técnico, para fortalecer competencias de liderazgo y de gestión desde un lado más administrativo.
- Desarrollar y proponer espacios de transmisión de conocimiento donde se enseñe lo aprendido, especialmente a los nuevos integrantes de soporte de tecnología.
- Desarrollo de habilidades de comunicación ya que, si bien la gestión de los proyectos trabajados se llevó de una buena manera, con habilidades de mejor comunicación fluyen más los proyectos.

Referencias Bibliográficas

[1] Economipedia.com. (2020). Empresas más grandes del mundo 2020. [En Línea]. Disponible en: <https://economipedia.com/ranking/empresas-mas-grandes-del-mundo-2020.html> [Consultado 3 Apr. 2020].

[2] Informativoindustrial.com. (2017). Accenture se renueva para continuar creciendo en Colombia. [En Línea]. Disponible en: <http://www.informativoindustrial.com/industria/accenture-se-renueva-para-continuar-creciendo-en-colombia/> [Consultado 3 Apr. 2020].

[3] Rankia.co. (2020). Mejores empresas colombianas para octubre 2020. [En Línea]. Disponible en: <https://www.rankia.co/blog/mejores-opiniones-colombia/3153489-mejores-empresas-colombianas-para-octubre-2020> [Consultado 20 Jun. 2020].

[4] Fiduciaria.grupobancolombia.com. (2020). Fiduciaria. [En Línea]. Disponible en: <https://fiduciaria.grupobancolombia.com/wps/portal/fiduciaria> [Consultado 25 Jun. 2020].

[5] McCormick, Mike (2012). Waterfall vs. Agile Methodology. [En Línea]. Disponible en: http://www.mccormickpcs.com/images/Waterfall_vs_Agile_Methodology.pdf [Consultado 3 Apr. 2020].

[6] Trigas Gallego, Manuel. Metodología Scrum. [En Línea]. Disponible en: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTF0612memoria.pdf> [Consultado 3 Apr 2020].

- [7] ibm.com. (2020). ¿Qué es DevOps?. [En Línea]. Disponible en: <https://www.ibm.com/ar-es/cloud/devops> [Consultado 15 Jul. 2020].
- [8] docs.microsoft.com. (2017). Overview of .Net Framework [En Línea] Disponible en: <https://docs.microsoft.com/en-us/dotnet/framework/get-started/overview> [Consultado 15 Jul. 2020].
- [9] docs.microsoft.com. (2015). C# language. [En Línea]. Disponible en: <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework> [Consultado 15 Jul. 2020].
- [10] docs.microsoft.com. (2020). Windows Forms for .NET 5 and .NET Core 3.1. [En Línea]. Disponible en: <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-5.0> [Consultado 15 Jul. 2020].
- [11] docs.microsoft.com. (2020). Entity Framework Documentation. [En Línea]. Disponible en: <https://docs.microsoft.com/en-us/ef/> [Consultado 15 Jul. 2020].
- [12] sqlservertutorial.net. What is SQL Server. [En Línea]. Disponible en: <https://www.sqlservertutorial.net/getting-started/what-is-sql-server/> [Consultado 15 Jul. 2020].
- [13] dotnet.microsoft.com. [2020]. What is ASP.NET?. [En Línea]. Disponible en: <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet> [Consultado 8 Aug. 2020].
- [14] oracle.com. [2020]. Oracle Database. [En Línea]. Disponible en: <https://www.oracle.com/database/> [Consultado 8 Aug. 2020].
- [15] docs.microsoft.com. [2019]. Welcome to the visual studio ide. [En Línea]. Disponible en: <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019> [Consultado 8 Aug. 2020]
- [16] docs.microsoft.com. [2020]. SQL Server Management Studio(SSMS). [En Línea]. Disponible en: <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15> [Consultado 8 Aug. 2020].
- [17] oracle.com. [2020]. Oracle SQL Developer. [En Línea]. Disponible en: <https://www.oracle.com/database/technologies/appdev/sqldeveloper-landing.html> [Consultado 10 Sep. 2020].
- [18] Azure.microsoft.com. [2020]. Azure DevOps. [En Línea]. Disponible en: <https://azure.microsoft.com/en-us/services/devops/> [Consultado 10 Sep. 2020]