



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Implementación, diseño y consolidación de fuentes para un Data Warehouse

INFORME DE ACTIVIDADES PROFESIONALES

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

Raymundo Baltazar Paéz Naranjo

ASESOR DE INFORME

Ing. Oswaldo Mariscal Bello

Ciudad Universitaria, Cd. Mx., junio 2019



Agradecimientos

A mis padres por todas sus enseñanzas y regaños, por enseñarme que trabajando duro se pueden alcanzar nuestros objetivos, por el resto de mis días les estaré agradecido por su apoyo incondicional.

A mis hermanas a pesar de que no fui el hermano que quizás hubieran deseado, siempre las voy a querer.

A mi pareja que me apoyó en los momentos difíciles y por ayudarme a encaminarme a apoyar a otros seres vivos.

Al Dr. Daniel Trejo por ser mi coach y camarada en el terreno deportivo y profesional.

Al deporte, por recordarme que debemos prepararnos y superarnos cada vez.

Contenido

Introducción.....	1
Definición del problema.....	3
Objetivo.....	4
Objetivos particulares.....	4
La empresa.....	4
Antecedentes.....	5
Inteligencia de negocios.....	5
Proceso de inteligencia de negocios.....	6
Arquitectura.....	6
Base de Datos vs. Data Warehouse.....	8
Desarrollo ágil: SCRUM.....	9
Panorama General.....	10
Principios.....	10
Aspectos.....	12
Procesos.....	14
Marco teórico.....	17
Álgebra.....	17
Computación para Ingenieros.....	20
Sistemas Operativos.....	21
Ingeniería de Software.....	22
Base de Datos.....	23
Administración de Proyectos de Software.....	26
Análisis y metodología.....	28
Inicio.....	28
Primeros pasos.....	28
Arquitectura.....	29
Equipo.....	30
Creación de tareas.....	31

Planeación y estimación	31
Historias de usuario	32
Creación y estimación.....	33
Implementación.....	34
Creación de entregables.....	35
Reuniones diarias y de avance.....	40
Revisión y retroalimentación	41
Validar el Sprint	41
Retroalimentación.....	41
Liberación	42
Liberación de entregables.....	42
Retroalimentación del proyecto	42
Participación profesional	44
Scrum Master.....	44
Inicio	44
Planeación y estimación	45
Implementación	46
Revisión y retroalimentación.....	47
Liberación	48
Desarrollador ETL.....	48
Planeación y estimación	48
Implementación	49
Revisión y retroalimentación.....	50
Liberación	51
Resultados obtenidos.....	53
Conclusiones.....	54
Bibliografía	57

Introducción

El presente informe titulado “Implementación, diseño y consolidación de fuentes para un Data Warehouse” expone la implementación de una plataforma para la extracción y consolidación de datos provenientes de diferentes fuentes, las cuales se depositaron en una arquitectura de Data Warehouse para su explotación y análisis, realizado para una organización de iniciativa privada en el ramo de servicios, a la que me referiré de ahora en adelante por temas de privacidad con el nombre clave “SEDCOD”.

Cabe mencionar que este proyecto pertenece a otro aún más grande, por lo que la dependencia y colaboración con otros equipos se realizó antes, durante y después de la finalización del proyecto; para fines prácticos de este informe se delimitará a la implementación del Data Warehouse.

Con una arquitectura robusta, se otorgó valor de negocio tanto para usuarios finales como personal ejecutivo de la organización SEDCOD, brindando información bajo una nueva perspectiva estratégica de sus propios datos en comparación con los sistemas de operaciones diarias y transaccionales.

Hoy en día las tecnologías que permiten la integración, manejo y análisis de grandes volúmenes de información para que posteriormente ésta sea consultada a través de determinadas aplicaciones permite a las organizaciones contar con plataformas que apoyen y realicen un análisis de los datos en el proceso de toma de decisiones o inclusive brinden información estratégica; estos sistemas son conocidos como Data Warehouse (Inmon, 2002).

Un Data Warehouse juega un papel importante en el proceso de toma de decisiones, permitiendo a los usuarios poseer un ambiente con un enfoque completamente diferente sobre sus datos Este enfoque parte del volumen de información generada hacia las necesidades del usuario (Mendez, Mártire, Britos, & Gracia-Martínez, 2003).

La plataforma de extracción, transformación y carga, por sus siglas en (Anderson, 2019) ETL (extract, transform and load), consolidó datos provenientes de diferentes motores de bases de datos (Oracle, PostgreSQL, SQL Server y MySQL) e inclusive de archivos planos (hojas de cálculo, archivos de texto plano con y sin estructura, además de archivos de texto separados por comas).

Antes de depositar la información en un nuevo motor de base de datos para su consumo y explotación, fue necesario aplicar reglas de negocio establecidas por los usuarios finales dentro de la organización SEDCOD. La plataforma Talend Data Integrator es un

software de ETL permitió una mayor velocidad de carga, actualización periódica y constante de la información, permitiendo garantizar la disponibilidad y fiabilidad de la información.

El proceso de carga de información se depositó en el manejador de bases de datos de procesamiento paralelo Greenplum, creado por Pivotal y de uso libre, el cual fue el motor de base de datos para el Data Warehouse para la generación de entidades. Dicho motor ofrece un proceso para replicar la información, permitiendo garantizar la alta disponibilidad de la información y de la plataforma.

Con facilidades para la optimización de consultas, planes de ejecución y un sistema híbrido de SQL y NoSQL se generó en conjunto con la plataforma ETL una arquitectura robusta. Además, con la información depositada en el Data Warehouse, se permitió el consumo de la misma a los usuarios finales y ejecutivos para su análisis. De tal manera se brindó una solución integral que abarcó desde la extracción de la información hasta su consumo, procurando la constante entrega de valor a la organización SEDCOD.

Definición del problema

La descripción del problema surge tras la evaluación que realicé de las necesidades del cliente y del usuario final en la organización SEDCOD. En reuniones previas con el cliente, como lo dicta la metodología Scrum (SBOK Guide, 2016), identifiqué a los actores involucrados en el proyecto y de entre ellos a los que me brindaron información acerca del entorno tecnológico y áreas de oportunidad.

Obtuve información sobre la situación actual del cliente, de esta manera conocí la problemática de contar con grandes volúmenes de información pertenecientes a la operación diaria de entradas y salidas, que además se encontraba dispersa en diversas fuentes de información, dicha descentralización de la información no permitía la explotación de éstos debido al consumo de horas involucradas en la consolidación de información y armado de reportes por parte de los usuarios finales.

Este proceso se realizaba a través de generar consultas de información a las diferentes instancias de bases de datos o inclusive mediante la generación de reportes manuales en hojas de cálculo. Posteriormente, se debía consolidar la información en hojas de cálculo o en tablas intermedias dependiendo del motor de base de datos para aplicar diversas reglas de negocio generales o particulares.

El proceso se realizaba con una periodicidad diaria, semanal, mensual, trimestral, semestral y anual, dependiendo del tipo de reporte y de información que se les solicitarán a las áreas correspondientes dentro de la organización.

Bajo este contexto se pudo definir el problema:

P: No se tiene una plataforma que consolide la información de diferentes fuentes de información y que aplique determinadas reglas de negocios.

Con este panorama y con los sistemas adecuados, los conceptos de automatización de tareas, reducción en los tiempos de entrega de información y apoyo en el proceso de toma de decisiones permitieron otorgar valor de negocio a la organización.

Objetivo

Implementar una plataforma de extracción, transformación y carga de datos, consolidando diversas fuentes de información para el diseño y creación de un Data Warehouse.

Objetivos particulares

- Implementar una plataforma de ETL (Extract, Transform and Load).
- Construir diversos trabajos que consoliden diversas fuentes de información (bases de datos, repositorios, archivos planos, etc.) aplicando reglas de negocio.
- Depositar la información en un motor de base de datos de procesamiento paralelo.
- Mantener la información consolidada y actualizada, de acuerdo a las necesidades de la organización.
- Diseñar un esquema de Data Warehouse.
- Replicar la información del Data Warehouse para contar con un esquema de alta disponibilidad de bases de datos.

La empresa

La empresa, en la cual fui el Scrum Master de este proyecto, tiene 14 años en el mercado mexicano, certificada en alta madurez de CMMI (Capability Maturity Model Integration) en SVC (servicios) y con oficinas en la Ciudad de México.

Yo reporté durante el proyecto a la oficina de proyectos, que era la encargada de mantener en tiempo y presupuesto el proyecto. Y tuve a mi cargo un equipo de 9 personas directamente.

La empresa fue contratada por su experiencia en esta área de negocios y soluciones informáticas.

Antecedentes

El desarrollo de sistemas que aporten información estratégica para la toma de decisiones y su desarrollo a través de una metodología de trabajo se ha vuelto cada vez más común, pero por otra parte la concordancia entre sistemas que aporten valor a las organizaciones y el desarrollo ágil en los proyectos, es lo que no resulta tan común.

Actualmente las empresas de consultoría buscan elementos técnicos capaces de desarrollar soluciones complejas y eficientes, y que bajo determinados estándares como lo son CMMI, ISO o IEEE, generen confianza a las organizaciones en la realización de sus proyectos, pero sobre todo crean los cimientos técnicos y no técnicos para la entera satisfacción de los objetivos tanto para las consultorías que brindan un servicio, como para las organizaciones que lo reciben.

Inteligencia de negocios

El desarrollo de tecnologías enfocadas a la preparación y análisis de grandes volúmenes de datos permite a las organizaciones convertir esos datos en conocimiento, y este conocimiento toma importancia cuando se utiliza para la toma de decisiones. Surge así una nueva necesidad de desarrollar sistemas robustos para generar este conocimiento que brindará información estratégica tanto para el personal técnico como no técnico.

La interpretación del conocimiento generado depende del uso que se le dé en cada una de las áreas de las organizaciones, por lo que la información generada repercute desde una visualización en vivo de la operación diaria, hasta la toma de decisiones del personal de negocio.

La inteligencia de negocios o BI por su nombre en Inglés (Business Intelligence), se puede definir de acuerdo con el Data Warehouse Institute como: *“La combinación de tecnologías, herramientas y procesos que permiten transformar datos almacenados en conocimiento y este conocimiento se dirige a un plan o estrategia”* (TDWI, 2019).

La inteligencia de negocios debe ser parte de cualquier estrategia que permita la optimización de los recursos, monitorear el cumplimiento de los objetivos de las organizaciones y otorgar capacidad de tomar decisiones de negocio para la obtención de mejores resultados.

Proceso de inteligencia de negocios

El proceso que sigue la inteligencia de negocios (Bernabeu, 2010) cuenta con cinco fases, véase figura 1.

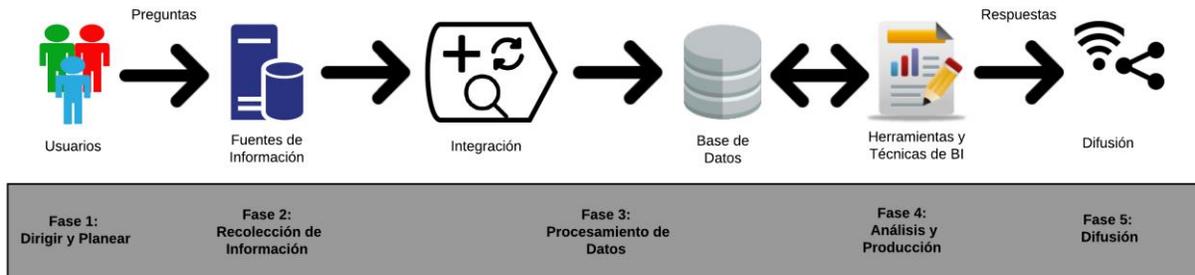


Figura 1 - Fases de inteligencia de negocio. Fuente: adaptación (Bernabeu, 2010).

- Fase 1 Dirigir y planear: En la fase inicial se recolecta la información proveniente de los usuarios para determinar los requerimientos de información específicos y así entender sus necesidades. En conjunto con los usuarios se generan preguntas que ayudarán a alcanzar los objetivos.
- Fase 2 Recolección de información: Fase en donde se realiza el proceso de extracción de información de los sistemas institucionales o fuentes externas, la recolección de información es necesaria para encontrar respuestas a las preguntas de la fase anterior.
- Fase 3 Procesamiento de datos: Los datos obtenidos de la recolección se transforman o se les aplican reglas de negocio para que sean cargados y consolidados en una nueva base de datos.
- Fase 4 Análisis y producción: Utilizando herramientas y técnicas propias de la tecnología BI se realiza el análisis y creación de modelos sobre la información y de esta manera se generan las respuestas a las preguntas de la fase 1.
- Fase 5 Difusión: Se genera una estrategia de liberación de los analíticos y conocimiento generado.

Arquitectura

Para implementar la inteligencia de negocios es necesario un proceso que posibilite consolidar las diversas fuentes de información en un solo destino y posteriormente permitir su análisis. Este proceso se denomina Data Warehouse (Bernabeu, 2010).

El Data Warehouse se encarga de extraer, transformar y consolidar los datos que las organizaciones generan en todos los rubros posibles, permitiendo realizar un análisis y

exploración de la información requerida para brindar información estratégica (Inmon, 2002).

Es importante visualizar los componentes que integran una arquitectura de inteligencia de negocios, la figura 2 representa esta arquitectura.

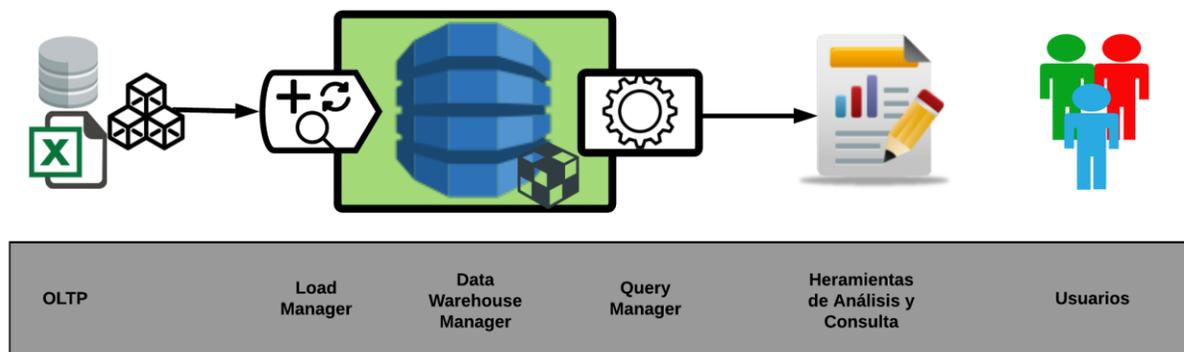


Figura 2 - Arquitectura Data Warehouse. Fuente: adaptación (Bernabeu, 2010).

- OLTP. Por sus siglas en Inglés OLTP (On Line Transaction Processing) representa las diversas fuentes de información transaccionales que generan las organizaciones en su operación diaria y fuentes externas que se encuentren disponibles para su consulta. Algunos de los OLTP más comunes entre las organizaciones son: bases de datos transaccionales, archivos planos de texto, hojas de cálculo, etc.
- Load Manager. Corresponde a la integración y mapeo de datos, la cual agrupa una serie de técnicas que se encargan de extraer, transformar, manipular, e integrar los datos en el motor de base de datos del Data Warehouse, labor que realiza la plataforma de ETL (Extract, Transform and Load), consolidando las diversas fuentes del OLT.
- Data Warehouse Manager. Se representa mediante el motor de base de datos que se utiliza para almacenar los datos transformados visualmente en modelos multidimensionales, dimensiones, modelos de negocio y tablas de hecho.
- Query Manager. Permite habilitar componentes, administrar consultas, monitorear procesos, realizar cálculos, generar métricas, etc. Generalmente se realizan las operaciones necesarias para soportar los procesos de gestión y ejecución de consultas relaciones y de consultas propias del análisis de datos como lo son: drill-down, drill-up, pivot, etc.
- Herramientas de análisis y consulta. Interfaz mediante la cual el usuario consultará y analizará la información disponible en el Data Warehouse. Existen diferentes tipos de herramientas de consulta y análisis que, de acuerdo a las

necesidades, pueden ocuparse para reportes, consultas, tableros, minería de datos, etc.

- Usuarios. Son los encargados de tomar decisiones y planificar actividades o estrategias de negocio.

Base de Datos vs. Data Warehouse

Existen diversas soluciones para almacenar y manipular los datos que las organizaciones generan, sin embargo, la finalidad de su uso es donde radica cuál es la mejor solución para las situaciones actuales. El Data Warehousing surge de la necesidad de crear análisis complejos y establecer procesos de toma de decisiones más eficientes, consistentes y de confianza (Boal, 2016), su objetivo va más allá del almacenamiento y la transaccionalidad.

A continuación, en la tabla 1, se presenta un cuadro comparativo entre los motores de base de datos convencionales y un Data Warehouse.

Tabla 1 - Comparación Base de datos vs. Data Warehouse. Fuente: Health Catalyst (Cardon, 2018).

	Base de Datos	Data Warehouse
Definición	Colección de datos organizados para su almacenamiento, accesibilidad y recuperación.	Tipo de base de datos que integra datos de diferentes fuentes de origen y los proporciona para un uso analítico y/o estratégico.
Tipos	Existen diferentes tipos, pero generalmente se aplica a una base de datos de aplicación OLTP como los son bases de datos transaccionales, archivos planos de texto, hojas de cálculo, etc.	Un Data Warehouse es una base de datos OLAP (On Line Analytical Processing), que se encarga de realizar el proceso de analizar volúmenes grandes de datos, con la finalidad de generar un nuevo conocimiento. No todos los OLAPs son iguales, la diferencia radica en como el dato es modelado.
Similitudes	Tanto las bases de datos como un Data Warehouse almacenan y administran los datos en forma de tablas, columnas, índices, vistas y tipos de datos. Se realizan consultas SQL para acceder a los datos.	
Optimización	Optimizado para llevar a cabo operaciones de lectura y escritura de transacciones. Una base de datos debería proporcionar tiempos de respuesta rápidos para pequeños volúmenes de	Optimizado para la consulta de grandes volúmenes de datos. Un Data Warehouse está diseñado para manejar consultas analíticas que recuperen grandes volúmenes de datos haciendo uso de recursos

	datos, realizar consultas con grandes volúmenes puede repercutir en el performance del motor y el tiempo de ejecución de las consultas.	como el CPU y la lectura en los discos.
Organización de los datos	Una base de datos OLTP presenta un modelado relacional y complejo de tablas, debido a que los datos se encuentran normalizados. De esta manera el almacenamiento y procesamiento de datos se vuelve eficaz.	En una base de datos OLAP la estructura de base de datos es organizada específicamente para facilitar la lectura y análisis de los datos. Los datos no están normalizados para mejorar tiempos de respuesta a consultas analíticas proporcionando facilidad de uso para los usuarios de negocio.
Reporte / Análisis	Debido a la combinación de tablas, realizar consultas analíticas resulta ser complejo, por lo que requiere de personas altamente técnicas para la elaboración de reportes, los cuales se mantienen estáticos y aislados a un solo requerimiento.	Con una menor combinación de tablas, las consultas analíticas tienen un mejor performance. Lo que también significa que los usuarios no tan técnicos pueden elaborar sus propios reportes de acuerdo a las necesidades diarias. Un Data Warehouse permite realizar los siguientes análisis: Descriptivo, Diagnóstico, Predictivo y Prescriptivo

Desarrollo ágil: SCRUM

Actualmente en los proyectos relacionados a las Tecnologías de Información se han implementado diversas metodologías que facilitan el desarrollo, sin embargo, las organizaciones buscan hacer eficiente y eficaz el desarrollo de nuevos proyectos. Cada uno de estos proyectos posee una naturaleza y un comportamiento diferente a la hora de su realización.

El éxito del proyecto depende en gran parte de la metodología escogida por el equipo, ya sea de las llamadas *tradicionales* o de las llamadas *ágiles*. Sin importar el tipo de metodología que se emplee, los objetivos deberán ser siempre los mismos: maximizar el

potencial del equipo y propiciar un aumento en la calidad del producto o servicios con los recursos y tiempos establecidos (Figuroa-Diaz, 2007).

Panorama General

Scrum es uno de los marcos más populares para la administración de proyectos ágiles (Lee, 2019). Es una metodología adaptativa, iterativa, rápida, flexible y eficaz, diseñada para ofrecer el máximo valor para el negocio en intervalos cortos de tiempo durante todo el proyecto.

De acuerdo con la guía de conocimiento de Scrum (SBOK Guide, 2016) el marco de la metodología Scrum puede ser entendido a través de principios, procesos y aspectos, cuya relación compone los ejes principales sobre los que la metodología se desarrolla, véase figura 3.

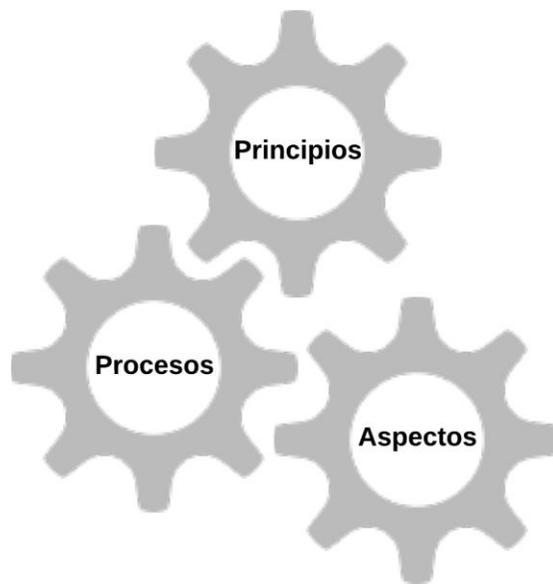


Figura 3 - Componentes de Scrum. Fuente: adaptación (SBOK Guide, 2016).

Principios

Los principios de Scrum son los fundamentos sobre los que se basa el marco de la metodología. Estos principios son obligatorios para la implementación de Scrum en cualquier tipo de proyecto u organización, con el fin de garantizar la aplicación apropiada de la metodología. Los aspectos y procesos pueden adaptarse a los requerimientos del

proyecto u organización, sin embargo los principios no son negociables (SBOK Guide, 2016). Los principios se describen a continuación:

- Control empírico del proceso.

Este principio determina la toma de decisiones basada en la observación y la experimentación, en lugar de la planificación detallada, permitiendo una adaptación a las diversas circunstancias de proyectos u organizaciones. Esto se logra gracias a la base sólida de las ideas de transparencia, inspección y adaptación.

- Autorganización

Este principio se basa en los trabajadores actuales, los cuales entregan un valor significativamente mayor cuando adoptan la forma de trabajar bajo el estilo de autorganización por encima de un estilo de mando, de manera que los trabajadores tienen mucho más conocimiento que ofrecer, que su propia experiencia técnica.

- Colaboración:

Scrum enseña que el desarrollo de un producto y/o servicio es un proceso de creación de valor compartido, en donde los interesados en el desarrollo del proyecto trabajan e interactúan juntos con la finalidad de entregar el más alto valor posible.

- Priorización basada en el valor.

La entrega del mayor valor posible en un periodo corto requiere de priorización, ésta se puede definir como el orden y separación de lo que debe ser hecho a hora y de lo que se necesita hacer después.

- Plazos establecidos.

El tiempo es tratado como una restricción limitante y lo que los plazos establecidos representan es el ritmo en el que todos los interesados trabajan y contribuyen. Los plazos establecidos también proponen determinar una cantidad de tiempo para cada proceso y actividad, considerando la claridad y complejidad de las tareas a realizar.

- Desarrollo iterativo.

El modelo de desarrollo iterativo es un modelo flexible que permite asegurar que las peticiones de cambio realizadas por el cliente puedan ser incluidas en el desarrollo del proyecto. Un principio clave es que los requisitos a largo plazo no se pueden definir completamente al inicio del proyecto, por lo que se debe concentrar en que el equipo sea lo suficientemente flexible para incorporar los cambios en los requerimientos.

Aspectos

Los aspectos de Scrum se deben tomar en cuenta y gestionarse a lo largo de un proyecto, a diferencia de los principios, los aspectos no son necesariamente obligatorios y pueden adaptarse a las situaciones que presente el proyecto u organizaciones (SBOK Guide, 2016). Organización, justificación de negocio, calidad, cambio y riesgo son los aspectos de Scrum y se describen a continuación:

- Organización

Comprender los roles, funciones y responsabilidades dentro de la metodología asegura la implementación exitosa de Scrum. Existen dos tipos de roles: principales y no esenciales, véase tabla 2. Los roles principales son los que intervienen directamente en el desarrollo del producto o servicio, su compromiso en el proyecto es alto y son responsables del éxito del proyecto.

Por otro lado, los roles no esenciales son aquellos que no tienen un rol formal en los equipos de trabajo, pero en determinadas ocasiones se requiere interactuar con ellos, no se pueden hacer responsables del éxito del proyecto, pero su contribución puede ser de diferente manera, como la de aportar recursos a la realización del proyecto.

Tabla 2 - Roles de Scrum. Fuente: elaboración propia.

Roles Principales	Roles No Esenciales
Product Owner	Clientes
Scrum Master	Usuarios
Scrum Team	Patrocinadores
	Vendedores

- Justificación del Negocio

Es importante para las organizaciones llevar a cabo una evaluación apropiada del negocio antes de comenzar un proyecto. Esto ayuda a quienes toman las decisiones a entender la necesidad de cambio o de un nuevo producto o servicio en la organización, al par de comprender la justificación para la realización y viabilidad. La justificación del negocio está basada en el concepto de entrega basada en el valor.

La incertidumbre en los resultados que podría alcanzar el proyecto o servicio juega un papel importante en el éxito del proyecto. Es por eso que Scrum intenta iniciar con la entrega de resultados lo más pronto posible, esta entrega temprana de resultados

proporciona una visión del valor del proyecto a los roles no esenciales de las organizaciones.

La adaptabilidad de Scrum permite que los objetivos y procesos del proyecto cambien si cambia la justificación del negocio.

- Calidad

En Scrum, la calidad se define como la capacidad del producto o del servicio de cumplir con los criterios de aceptación de las organizaciones y de alcanzar el valor de negocio que espera el cliente. Para asegurar que el proyecto cumpla con los requisitos de calidad, Scrum adopta un enfoque de mejora continua.

El equipo aprende de sus experiencias y de la interacción con el cliente y usuarios, generando una actualización constante a la lista de tareas a priorizar. Cualquier cambio en los requisitos debe reflejar cambios en el entorno empresarial, tanto interno, como externo.

Por medio de pruebas repetitivas, Scrum requiere que el trabajo se complete gradualmente a través de periodos conocidos como *sprint* y en donde se llevan a cabo tareas relacionadas con la calidad: desarrollo, pruebas, documentación, validación, etc.

- Cambio

Cada proyecto, independientemente de su método o marco, siempre estará expuesto a cambios. Las organizaciones deben tratar de maximizar los beneficios que se derivan de los cambios y de minimizar los impactos negativos a través de los procesos. Scrum reconoce que es imposible definir todos los requerimientos al inicio del proyecto y que éstos, también pueden cambiar durante el curso del proyecto.

Scrum acepta las peticiones de cambio, mediante el uso de *sprints* cortos y repetitivos en los cuales ya se cuenta con la retroalimentación por parte del cliente. Esto permite que el cliente interactúe con el equipo y obtenga entregables a medida que éstos se encuentren listos y cambie los requerimientos si es necesario antes del siguiente *sprint*.

- Riesgo

Se define como algo incierto o conjunto de eventos que pueden afectar los objetivos de un proyecto y pueden contribuir al éxito o al fracaso. A los riesgos que pueden tener un impacto positivo se les conoce como oportunidades.

La gestión del riesgo debe hacerse de forma preventiva y es un proceso iterativo que debe comenzar al inicio del proyecto y continuar a lo largo del ciclo de vida. El proceso de gestión de riesgos sigue una serie de pasos que van desde la identificación, la

evaluación y la manera en que se responde basado en los factores de probabilidad de ocurrencia y posible impacto en caso de ocurrencia.

Procesos

Los procesos Scrum abordan las actividades y el flujo específico de un proyecto Scrum. En total hay 19 procesos agrupados en 5 fases. Estas fases describen el ciclo de vida de la metodología, véase figura 4.

La primera fase recibe el nombre de inicio en donde dan comienzo las tareas que definen los requerimientos generales y se identifican los roles que existirán a lo largo del desarrollo. Además, se establece al equipo de trabajo, quienes convertirán la visión del proyecto en entregables tangibles y con valor al negocio. También se define un calendario de liberación.

En la fase planeación y estimación, los requerimientos generales se desagregan en tareas más pequeñas y puntuales. Las tareas se priorizan basado en cuanto valor aportan al negocio. Se genera un sprint y la cantidad de trabajo que realizará en determinado lapso, así como, los criterios de aceptación para cada *sprint*.

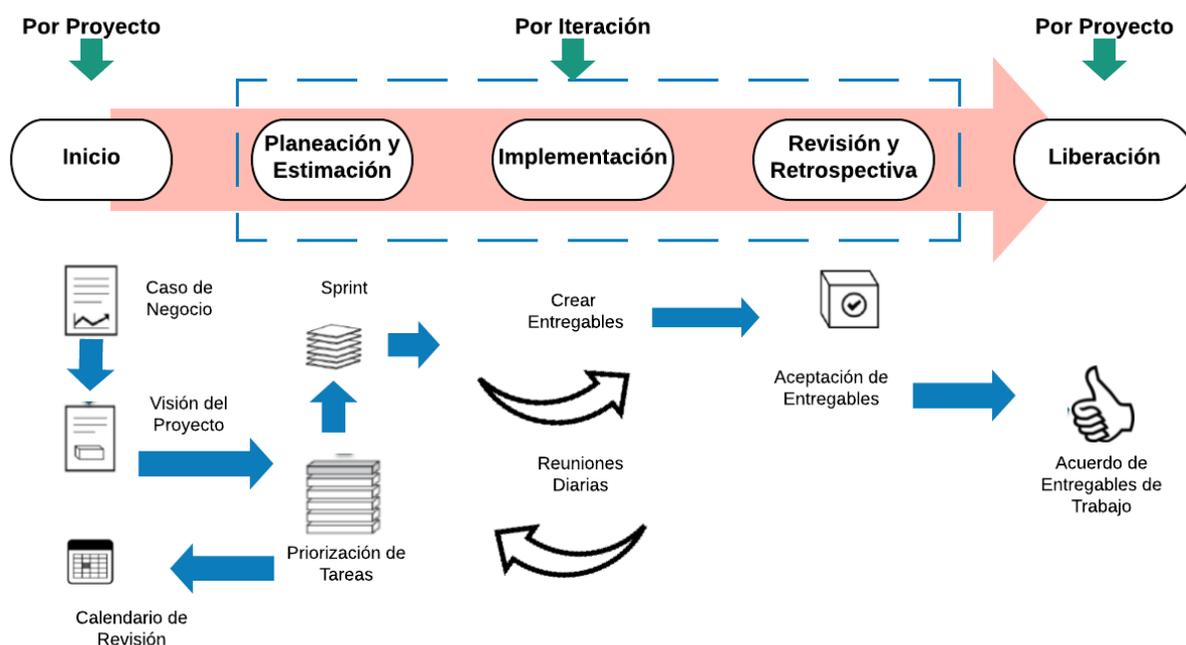


Figura 4 - Flujo de Scrum. Fuente: adaptación (SBOK Guide, 2016).

La etapa de desarrollo corresponde a la fase de implementación, aquí da comienzo el sprint, las tareas propuestas en la fase anterior se materializan y conforme se van

concluyendo, existe una actualización constante sobre la prioridad de las tareas, lo cual permite adaptación y flexibilidad para los equipos de trabajo en conjunto con los interesados de las organizaciones.

Al término de cada *sprint* se hacen reuniones de retroalimentación para la mejora continua en los equipos de trabajo y optimizar el rendimiento de cada integrante. Los resultados de las reuniones se consideran para el inicio del siguiente sprint o proyecto. A través del criterio de aceptación los entregables son revisados y validados.

Finalmente, en la fase de liberación, se llevan a cabo acciones en conjunto con las organizaciones para la puesta en marcha de las entregas, al igual que la fase de revisión y retroalimentación se realizan reuniones para obtener una retroalimentación de todo el proyecto.

Los procesos que detallan los pasos en cada una de las fases, se muestran en la tabla 3. La utilización de cada uno de ellos depende de la forma en la que se ejecuten y del tamaño del proyecto, Scrum fue desarrollado para cualquier tipo de proyecto.

Tabla 3 - Procesos de Scrum. Fuente: adaptación (SBOK Guide).

Inicio	Planeación y Estimación	Implementación	Revisión y Retroalimentación	Liberación
<ul style="list-style-type: none"> - Crear la visión del proyecto. - Identificar al Scrum Master y los Stakeholders (Accionistas). - Formar el equipo. Desarrollar las épicas. - Crear la lista priorizada. - Realizar la planeación de liberación. 	<ul style="list-style-type: none"> - Crear las historias de usuario. - Aprobar y estimar las historias de usuario. - Crear las tareas. - Estimar las tareas. - Crear la lista priorizada del sprint. 	<ul style="list-style-type: none"> - Crear los entregables. - Realizar la reunión diaria. - Mantener la lista priorizada actualizada. 	<ul style="list-style-type: none"> - Convocar el Scrum de Scrum. - Demostrar y validar el sprint. - Realizar la retroalimentación del sprint. 	<ul style="list-style-type: none"> - Liberar las entregas. - Realizar la retroalimentación del proyecto.

Con un panorama general de la metodología de desarrollo Scrum, las organizaciones pueden llevar a cabo proyectos de cualquier índole y tamaño mediante las entregas basadas en el valor y desarrolladas por equipos multifuncionales con enfoques colaborativos y serviciales.

Esta metodología requiere de una madurez que se irá generando conforme el proyecto se vaya desarrollando. El énfasis está orientado sobre las personas y no sobre los procesos, permite la participación de todos los involucrados durante el ciclo de vida del proyecto y el empoderamiento de los participantes.

Marco teórico

El marco teórico constituye la base de conocimientos científicos y técnicos sobre los que se basa el desarrollo de este informe, el desarrollo teórico me permitió poseer las capacidades de análisis e interpretación necesarias para el desarrollo de este proyecto.

De acuerdo con el plan de estudios del año 2010 de la Carrera de Ingeniería en Computación (Facultad de Ingeniería, 2019) considero que las materias más relevantes para el desarrollo de este informe son las siguientes:

Álgebra

La utilización de la teoría de conjuntos facilitó la comprensión y armado de consultas específicas para la extracción de información.

De acuerdo con la definición “*un conjunto es una agrupación, colección o clase de objetos bien definidos que cumplen una propiedad determinada*” (Sullivan, 2006), las tablas que conforman cada esquema de base datos, archivos de texto o separado por algún carácter son nuestros conjuntos de información, con los que podemos realizar las mismas operaciones del álgebra de conjuntos (Lipschutz, 1991):

- Unión

Con los datos de las diversas fuentes de información, ya sea en su propio ambiente (UNION por sentencia SQL) o a través de la plataforma ETL, se pudo llevar a cabo la operación de unión de diferentes tablas y/o archivos, véase figura 5.

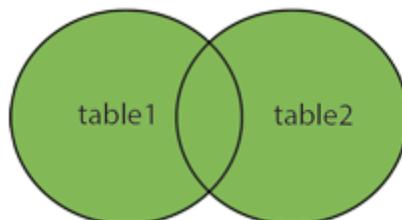


Figura 5 - Diagrama de Venn unión. Fuente: (W3school, 2019).

Consulta SQL:

```
SELECT <campos>  
FROM table_a
```

```
UNION
SELECT <campos>
FROM table_b
```

- Intersección

La operación de intersección se realizó a través de la sentencia SQL INNER JOIN en los motores de base de datos, la intersección con los archivos planos se realizó con la ayuda de la plataforma ETL. Con lo que sólo se obtuvo la información que coincidiera con los criterios que tuvieran en común, véase figura 6.

Diagrama de Venn:

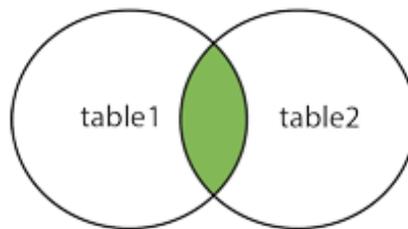


Figura 6 - Diagrama de Venn intersección. Fuente: (W3schools, 2019).

Consulta SQL:

```
SELECT <campos>
FROM table_a a
INNER JOIN table_b b
ON a.key = b.key
```

- Diferencia

La operación diferencia puede representarse de dos maneras, al menos en el nivel SQL. En el lenguaje SQL cuenta con dos sentencias: LEFT JOIN y RIGHT JOIN, con las cuales obtenemos parcialmente la operación de diferencia, lo que hace falta para obtener la diferencia es especificar una cláusula WHERE dónde se especifique una propiedad en común de los conjuntos, la cual deberá ser nula, véase figura 7 y 8.

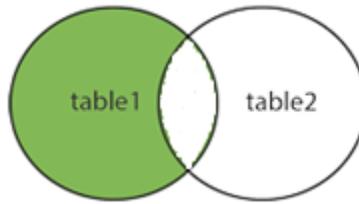


Figura 7 - Diagrama de Venn diferencia izquierda. Fuente: (W3schools, 2019).

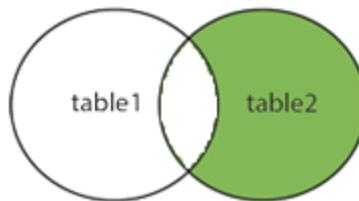


Figura 8 - Diagrama de Venn diferencia derecha. Fuente: (W3schools, 2019).

Consulta SQL:

LEFT JOIN DIFFERENCE

```
SELECT <campos>
FROM table_a a
LEFT JOIN table_b b
ON a.key = b.key
WHERE b.key = IS NULL
```

RIGHT JOIN DIFFERENCE

```
SELECT <campos>
FROM table_a a
RIGHT JOIN table_b b
ON a.key = b.key
WHERE b.key = IS NULL
```

- Producto cartesiano

Finalmente, el producto cartesiano se representa con la sentencia SQL FULL OUTER JOIN, la cual se conforma con las propiedades de cada conjunto, es decir, las coincidencias de ambas tablas, véase figura 9.

Diagrama de Venn:

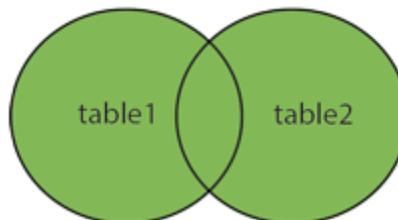


Figura 9 - Diagrama de Venn producto cartesiano. Fuente: (W3schools, 2019).

Consulta SQL:

```
SELECT <campos>
FROM table_a a
FULL OUTER JOIN table_b b
ON a.key = b.key
```

Computación para Ingenieros

El primer acercamiento con los fundamentos de la computación aplicada a la ingeniería fueron comprendidos en esta asignatura.

El conocimiento de la estructura física y lógica de las computadoras sumadas a las especificaciones de las aplicaciones me permitió considerar un panorama general sobre el tipo de infraestructura, adecuada que se requirió, para la plataforma de ETL, cuyo uso se centraría en la extracción de información y aplicación de reglas de negocio se necesitó de infraestructura que soportará y mantuviera información en memoria principal.

En el caso del motor de base de datos, se requirió del procesamiento de consultas que serán solo de lectura, sin embargo, durante el proceso de actualización de los datos se precisó del procesamiento por parte de la infraestructura.

De manera general presento en la figura 10 el diagrama de la infraestructura utilizada:

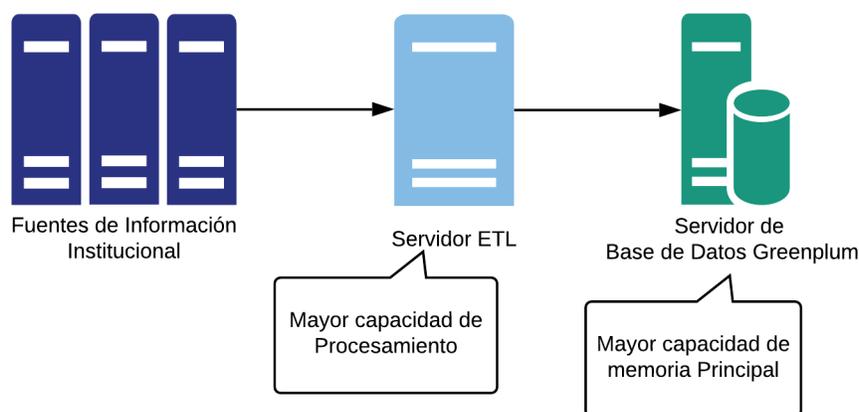


Figura 10 - Diagrama general arquitectura. Fuente: elaboración propia.

Tener un panorama general de la infraestructura me permitió la elaboración del plan de desarrollo y metodología para la administración, la cual se siguió a lo largo del proyecto, cabe mencionar que los primeros acercamientos con infraestructuras más complejas de

cómputo requirieron del conocimiento de bases sólidas de los componentes computacionales.

Sistemas Operativos

El consumo de recursos, tanto de la plataforma ETL, como del motor de bases de datos Greenplum sobre la infraestructura destinada a cada una de las plataformas entra en lo denominado administración de procesos y memoria. El óptimo funcionamiento de las plataformas depende del conocimiento técnico sobre las mismas, pero sobre todo de su comportamiento en el sistema operativo donde son huéspedes.

Cada uno de los procesos que se ejecutan en el sistema operativo genera un consumo de recursos, los cuales son finitos. El sistema operativo mantiene por cada proceso una serie de estructuras de información (Carretero Pérez, 2001) que permite identificar las características de éste, así como los recursos asignados.

Dentro de los procesos existe una clasificación que depende del sistema operativo en donde se estén ejecutando. Esta clasificación cobra importancia por el número de usuarios que vayan a acceder a las diferentes plataformas, punto importante a considerar para un óptimo desempeño, véase figura 11.

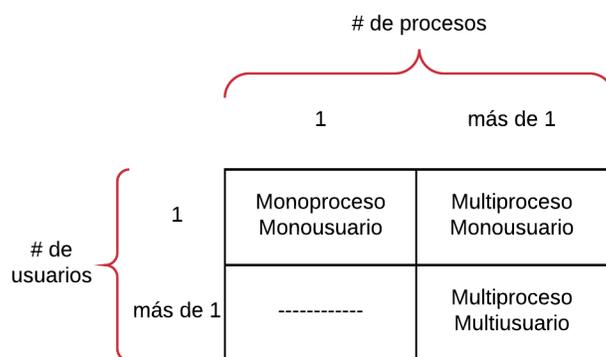


Figura 11 - Relación proceso y usuario. Fuente: adaptación (Carretero Pérez, 2001).

La memoria principal es uno de los recursos más importantes de los equipos utilizados para este proyecto. El responsable de su administración corresponde a la labor que realiza el gestor de memoria de cada uno de los sistemas operativos. Cada sistema operativo ha desarrollado su propio gestor de memoria y aunque cada gestor es completamente diferente uno del otro, es importante cumplir con los siguientes objetivos:

- Ofrecer a cada proceso un espacio lógico.
- Proporcionar protección entre los procesos.
- Permitir que los procesos compartan memoria.

- Maximizar el rendimiento del sistema.

En lo correspondiente al sistema de archivos, éste nos permite organizar la información dentro de los dispositivos de almacenamiento destinados a ello. Durante la instalación de los sistemas operativos, los dispositivos de almacenamiento se encuentran vacíos por lo que previamente se dividió física y lógicamente los discos en particiones o volúmenes para poder inicializar su uso. En los ambientes Windows y Linux utilizados en este proyecto, se realizó la distribución que se muestra en la figura 12.

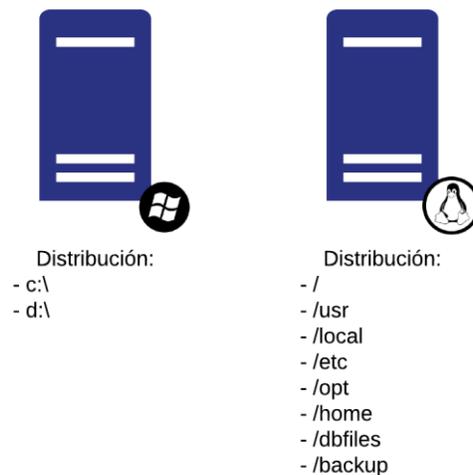


Figura 12 - Distribución de particiones. Fuente: elaboración propia.

Ingeniería de Software

Llevar a cabo la iniciación, planeación, estimación, implementación y finalmente la puesta en marcha (SBOK Guide, 2016) es sólo una parte de la ingeniería de software. Como lo define la IEEE, la ingeniería de software es *“La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software”* (Pressman, 2010) y en lo que debemos basarnos para el desarrollo del proyecto.

Si bien este proyecto no se relaciona con el desarrollo de software, si se genera un desarrollo de información acompañado de sistemas críticos por lo que el desarrollo de este proyecto abarcó la serie de capas que compone a la ingeniería de software, véase figura 13.



Figura 13 – Capas de la Ingeniería de Software. Fuente: adaptación (Pressman, 2010).

Estas capas representan lo que compone a la ingeniería de software, pero no definen los pasos a seguir para hacer ingeniería en el software, la cual se lleva a cabo con el apoyo de metodologías de desarrollo:

- Cascada
- Incremental
- Prototipo
- Evolutivo
- Rapid Application Development (RAD)
- Extreme Programming (XP)
- Rational Unified Process (RUP)

La elección adecuada de las metodologías depende del tipo de proyecto a la cual se quiera implementar, las herramientas con las que se cuente, los recursos materiales disponibles, pero sobre todo del personal que se involucre.

Base de Datos

Uno de los temas fundamentales para el desarrollo de este proyecto es el tema de base de datos y que el desarrollo del proyecto utilice diversos conceptos de base de datos aplicados a un enfoque práctico y laboral.

Una base de datos por concepto se define como un conjunto de datos lógicamente coherentes, relacionados entre sí y perteneciente a un mismo contexto (Elmasri & Navathe, 2003); sin embargo, este concepto general en un ambiente laboral se vuelve más complejo. Hay que considerar una serie de aspectos: qué tipo de base de datos, el almacenamiento físico de los datos, los mecanismos que se utilizan para su creación, estructura y acceso, las aplicaciones que harán uso y, sobretodo, la seguridad.

Hoy en día existen dos tipos generales de bases de datos: SQL y NoSQL. El uso que se le dé a los datos y la clase de aplicaciones utilizadas serán determinantes a la hora de elegir algún tipo o inclusive, una combinación de ambos.

Para las bases de datos SQL, también conocidas como bases de datos relacionales, se definen y manipulan los datos a través de un lenguaje de consulta estructurado que por sus siglas en inglés se conoce como SQL (Structured Query Language). Las bases relacionales son las más utilizadas y útiles para el manejo de datos estructurados que organizan elementos de datos y se estandariza como se relacionan entre sí (Anderson, 2019).

Por otra parte las bases de datos NoSQL, también conocidas como bases de datos no relacionales, permiten almacenar y recuperar datos no estructurados usando esquemas dinámicos. En este tipo de bases no relacionales se crean estructuras únicas y pueden ser documentos, gráficos, columnas o inclusive llaves organizadas como estructuras de datos como lo son pilas, grafos, etc (Anderson, 2019).

Un sistema de gestión de base de datos (DBMS por sus siglas en inglés) es una capa de software que permite crear, manipular y recuperar información desde una base de datos, algunas de las tareas fundamentales de un DBMS hacen referencia a la seguridad de acceso a los datos, al mantenimiento de la integridad de los datos, mecanismos de restauración y recuperación parcial o total de la información, manejo de la concurrencia y eficiencia en la respuesta a la hora de consultar los datos (Millán, 2017).

Mediante un par de lenguajes de bases de datos el usuario puede llevar a cabo una serie de actividades. El lenguaje de definición de datos (DDL por sus siglas en inglés) es el lenguaje mediante el cual se procesa las definiciones y almacena las descripciones de los esquemas en el catálogo del DBMS. El lenguaje de manipulación de datos (DML por sus siglas en inglés) permite realizar la consulta o manipulación de los datos, por último el lenguaje de control de datos (DCL por sus siglas en inglés) se encarga del control y seguridad de los datos (Elmasri & Navathe, 2003).

Finalmente, la arquitectura de un BDMS se basa en tres niveles, véase figura 14: (externo, conceptual e interno) en donde se trata de separar la forma en que los usuarios visualizan los datos, de los detalles de almacenamiento físico colocando una abstracción de los datos en medio.

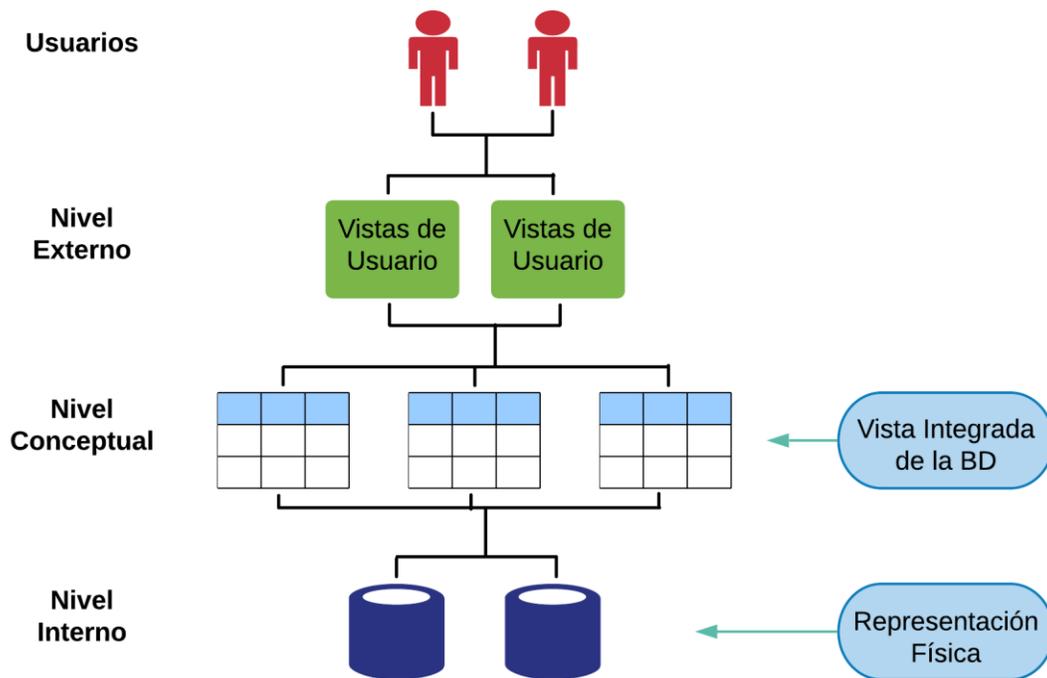


Figura 14 - Arquitectura DBMS. Fuente: adaptación (Millán, 2017).

El nivel interno describe la estructura física del almacenamiento de la base de datos, dentro de este nivel físico se trata con los mecanismos de almacenamiento físico que el sistema operativo utiliza.

El nivel conceptual corresponde a la descripción de los datos y las relaciones entre estos, ocultando los detalles de las estructuras físicas de almacenamiento, se concentra en describir entidades, tipos de datos, relaciones, restricciones, etc.

Por último el nivel externo representa un esquema externo que describe la parte de la base de datos que interesa a determinados usuarios y oculta a ese grupo el resto de la información (Millán, 2017).

El modelo entidad-relación es una herramienta que permite representar cómo se organiza la información en una base de datos a través de las interrelaciones y propiedades de los datos (Pérez, 2017).

El modelo entidad-relación tiene tres elementos principales (Pérez, 2017):

- Entidades: representa un objeto o concepto del mundo real.
- Atributos: describen las propiedades que posee cada entidad.
- Relaciones: conjunto de vínculos entre entidades.

Estos conceptos representaron un breve panorama sobre las bases de datos y los pilares de conocimientos técnicos sobre el desarrollo de proyecto.

Administración de Proyectos de Software

Dentro de la materia de administración de proyectos de software obtuve un panorama general de lo que representa el desarrollo de un proyecto en el área de tecnologías de la información, el concepto de la administración de proyectos se puede definir como la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades que representan la elaboración de un proyecto con la única finalidad de satisfacer los objetivos del mismo (Carcaño, 2014).

Muchas de las técnicas de la administración de proyectos en general se pueden aplicar a un proyecto de software, pero que de acuerdo con el autor Hughes, los proyectos de software cuentan con características de los diferencian de cualquier otro proyecto y la manera de percibir los proyectos de software es el proceso de hacer visible lo que es invisible (Hughes & Cotterrell, 2001). Las características de un proyecto de software son las siguientes:

- Invisibilidad

El avance en proyectos generalmente es visible, sin embargo, en los proyectos de software no es inmediatamente visible hasta determinada etapa.

- Complejidad

Los proyectos de software contienen mayor complejidad que otros tipos de proyectos respecto a los recursos utilizados.

- Flexibilidad

El software puede cambiar fácilmente en comparación con otros productos.

Un rol muy importante es la del administrador de proyectos, ya que dentro de sus funciones destacan entre otros, la identificación de requerimientos, establecimiento de objetivos claros y posibles de realizar, equilibrio de las demandas concurrentes de calidad, alcance, tiempo y costos, adaptación de las especificaciones, planeación y el enfoque de las diversas exceptivas de los interesados, etc. Cualesquiera que sean la funciones del administrador este debe tener siempre presente los siguientes conceptos, véase figura 15 (Carcaño, 2014).



Figura 15 - Pilares de la administración de proyectos de software. Fuente: adaptación (Carcaño, 2014).

- Alcance. Engloba todo lo que se necesita hacer para cumplir los objetivos del proyecto, incluyendo funciones y características propias de cada proyecto.
- Costo. Cantidad limitada de recursos materiales y financieros que se asignan a la realización del proyecto.
- Tiempo. Cantidad de tiempo establecida para la duración del desarrollo del proyecto.
- Calidad. Conjunto de características medibles que deben cumplir los componentes del proyecto y en apego al cumplimiento con los requerimientos.

Como administrador de proyectos se debe encontrar la concordancia entre estos factores para el desarrollo satisfactorio del proyecto y donde se puede hacer uso de diversas metodologías de desarrollo para su cumplimiento.

Análisis y metodología

El desarrollo del proyecto se realizó bajo la metodología de desarrollo ágil Scrum durante las cinco fases en las que se desenvuelve: Inicio, Planeación y Estimación, Implementación, Revisión y Retroalimentación, Liberación (SBOK Guide, 2016). La metodología permitió un desarrollo adaptable en diversas etapas y la entrega constante de valor a la organización SEDCOD. A continuación, se presenta el análisis y desarrollo bajo el enfoque de la metodología SCRUM.

Inicio

En la etapa inicial del proyecto se llevaron a cabo reuniones con los interesados de la organización SEDCOD en donde se realizaron pláticas sobre los recursos y necesidades dentro de la organización.

Primeros pasos

En las primeras reuniones realicé un levantamiento de requerimientos para establecer los objetivos para ambas entidades y comenzar con el desarrollo del proyecto. Estas reuniones en un primer intento fueron con el personal encargado del proyecto por parte de la organización SEDCOD. Si bien no se trató del personal que se encarga de la operación diaria, sí me brindó un panorama general de las necesidades. También a este personal se le asignó el rol de *Product Owner*.

El resultado de las reuniones arrojó una serie de requerimientos generales, dentro de los cuales destacan los siguientes:

- Diseño de arquitectura.
- Selección, limpieza y consolidación de datos.
- Generar información estratégica de entradas y salidas de productos.
- Implementación de mecanismos para actualizar la información.
- Verificar horarios de consulta a sistemas institucionales.
- Resguardar la información obtenida en mecanismos para su restauración.

Estos requerimientos generales fueron el comienzo operacional del proyecto. Como resultado se pudo establecer la visión general del proyecto y se determinaron los objetivos internos como consultoría, los cuales fueron los bastiones a seguir, logrando así el cumplimiento como equipo y empresa.

Arquitectura

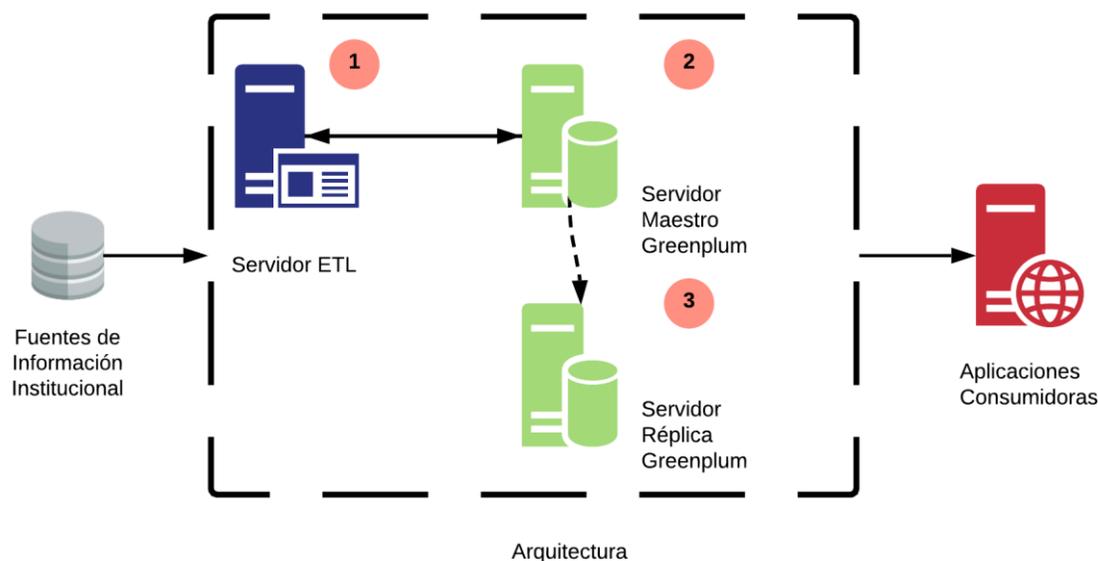


Figura 16 - Diagrama de la arquitectura. Fuente: elaboración propia.

En la figura 16 se presenta la arquitectura que se propuso y fue aceptada por la organización SEDCOD. Esta presenta una serie de tres servidores que se encargaron de consolidar y almacenar la información. El primero de los servidores es el servidor de ETL en el cual se instaló la herramienta de Talend Data Integrator, la cual proporcionó los servicios de ETL, incluyendo una diversa y amplia variedad de conectores de base de datos y sobre todo de la personalización de componentes.

El segundo y tercero de los servidores representan el Data Warehouse, en ellos se instaló el motor de base de datos de Pivotal llamado Greenplum. Se consideraron dos servidores debido a que se planteó una replicación de la información generada en el servidor “Maestro” hacia un servidor “Réplica”, generando así una arquitectura de alta disponibilidad a nivel de infraestructura y base de datos.

Al contrario del motor de base de datos Greenplum, el software del ETL no contó con una versión de código abierto por lo que con el licenciamiento de la herramienta no se pudo realizar una réplica para mantener siempre disponible la plataforma de ETL, sin embargo, se generó una estrategia de respaldo de la aplicación y de los trabajos ETL generados para su pronta recuperación en algún otro servidor.

A continuación, en la tabla 4 se presentan algunas características técnicas de los servidores utilizados en el proyecto y en cumplimiento de los requisitos de sistema de los fabricantes a nivel ETL y base de datos.

Tabla 4 - Características de la infraestructura. Fuente: elaboración propia.

Característica	Servidor ETL	Servidor Maestro Greenplum	Servidor Réplica Greenplum
Sistema Operativo	Windows Server 2012	CentOS 7.0	CentOS 7.0
Marca	Dell	Dell	Dell
Procesadores	2	2	1
Cores Totales	16	16	8
Memoria Principal	256 GB	512 GB	256 GB
Almacenamiento	1 TB	4 TB	4 TB
Red	2 tarjetas, 10 GB	2 tarjetas, 10 GB	2 tarjetas, 10 GB

La implementación del software en los servidores se realizó basada en los manuales de los fabricantes, “*Instalación del Integrador de Datos*” para el caso de la plataforma ETL (Talend, 2019) y “*Configuración e Instalación de Greenplum*” para el motor de base de datos (Pivotal, 2019).

Equipo

La conformación del equipo de trabajo requirió de un equipo con perfil orientado a bases de datos y otro perfil que apoyará en la generación de documentación tanto interna como externa, pero no sólo con capacidades técnicas, sino que contarán con habilidades para la constante comunicación con usuarios como con el equipo. El equipo se conformó de un Scrum Master, dos elementos de base de datos y un documentador.

Con el equipo conformado para esta parte del proyecto, realicé una reunión como Scrum Master del equipo para comunicar los resultados obtenidos en las primeras reuniones con el personal de la organización SEDCOD. En esta reunión se comentó ampliamente la visión del proyecto, la cual nos ayudó al equipo y a mí a elaborar una estrategia para abordar el reto que representó la construcción de un Data Warehouse.

Aprovechando los recursos que nos brindó la metodología Scrum, establecimos las reuniones diarias antes de comenzar nuestra jornada laboral, para poner en sintonía a todo el equipo, así como mostrar el trabajo realizado y por realizar en las jornadas de

trabajo, además de comunicar los incidentes y las posibles soluciones de manera que todo el equipo fuese participe e inclusive para poder planear mantenimientos cuando detectábamos la oportunidad correcta.

Creación de tareas

Con los primeros pasos y el equipo conformado, la siguiente etapa fue la reunión con los usuarios finales, el personal que lleva a cabo las tareas y operación de la organización SEDCOD. Estas reuniones se llevaron en primera instancia para dar a conocer al equipo, pero sobre todo para presentar la estrategia de trabajo ya que en este punto los usuarios debían involucrarse en el desarrollo del proyecto.

En las reuniones con los usuarios realicé junto con el equipo un segundo levantamiento de requerimientos más detallado, en el que nuestro principal objetivo fue la generación de épicas lo que nos sirvió para acotar el alcance del proyecto y refinar los objetivos que persiguió la organización. Algunas de las épicas más destacables fueron las siguientes:

- Consolidar información en una sola interfaz.
- Realizar análisis sobre información histórica.
- Comparativos de los movimientos de entrada y salida.
- Generación de reportes personalizables.

Una vez obtenidas épicas, me reuní con los involucrados de proyecto o como se conoce en Scrum, el *Product Owner*. El objetivo fue presentar las épicas para que conocieran el detalle y alcance de las tareas que se llevarían a cabo. Además, se realizó en conjunto la priorización de estas épicas. Con el apoyo que me brindó mi equipo pude sugerir las tareas de las cuales tuvimos la oportunidad de consolidar lo más rápido posible, generando así una entrega de valor desde el inicio del proyecto.

La priorización de las épicas no fue algo estático, al contrario de lo que podría pensarse, esta es una tarea que se volvió dinámica por los constantes cambios en el valor de las tareas. Conforme fue avanzando el proyecto, estas adquirirían diferentes valores y de acuerdo a la organización vio los avances por lo que procure mantener la concordancia con el alcance y objetivos propuestos.

Planeación y estimación

En la siguiente fase se realizó la planeación y estimación de tareas, las cuales se desarrollaron en la etapa de implementación. En este punto la duración de nuestro primer *sprint* fue un poco más larga de lo normal y es que en mi experiencia, en esta fase el equipo apenas se estaba familiarizando con los componentes, reglas de negocio y particularidades de la operación dentro de la organización, sin embargo, conforme se

desarrollan los *sprint* subsecuentes el equipo de trabajo fue adquiriendo madurez y habilidades para incrementar el ritmo de trabajo.

Historias de usuario

Como resultado de las reuniones con los usuarios obtuvimos las épicas, las cuales desagregamos en historias de usuario más manejables y realizables, en las cuales se plasmó la funcionalidad del Data Warehouse y cuya implementación aporta valor al cliente y a la organización.

A continuación, se presenta una historia de usuario realizada en el proyecto véase figura 17 y la manera en la que se plasmó dentro de la documentación generada.

Historia de Usuario	
Número: 7	Usuario: José Cárdenas (Área de entradas)
Nombre: Importes de entrada y salida.	
Prioridad: Alta	Riesgo: Medio
Puntos mínimos estimados: 7	Puntos máximos estimados: 12
Desarrollador responsable: Julio Romero	
Descripción: Quiero obtener los importes de entradas, salidas y una combinación de ellos, basado en los movimientos que los componen.	
Validación: El usuario puede obtener el importe de entradas, salidas mediante la aplicación de filtros a los movimientos que lo componen, además de poder combinar en un solo reporte ambos movimientos.	

Figura 17 - Historia de usuario. Fuente: elaboración propia.

En la imagen de arriba se muestra una historia de usuario que se compone del nombre del usuario a quién se le asigna dicha historia, se determina una prioridad basada en el valor que aportó esta historia a las actividades del usuario y se estima un riesgo.

Una parte variable dentro de las historias es la estimación para concluir esta tarea. Dicha estimación se generó basada en las capacidades del desarrollador en turno, es decir, las capacidades técnicas de cada integrante del equipo definió la duración de cada tarea. No todos los integrantes poseían el mismo ritmo de trabajo, así que el tiempo estimado para una tarea varía en función de la capacidad técnica y no técnica de cada integrante de realizarla, cada elemento del equipo adquirió los conocimientos técnicos y madurez del

negocio, por el lado contrario si incrementan, determinadas habilidades no se han desarrollado adecuadamente.

Finalmente se agregó un apartado de descripción en forma de diálogo del usuario describiendo la funcionalidad que deseo realizar y un apartado de validación que funcionó como un criterio de aceptación para dicha tarea.

El objetivo de haber generado historias de usuario es el haber convertido las épicas en tareas más manejables, menos complejas y más realizables en intervalos de tiempo más congruentes con dichas tareas.

Creación y estimación

En las historias de usuario se comentó la sección de puntos de estimación, que compone una aproximación al tiempo de finalización de la tarea, la cual desarrollaron integrantes de equipo bajo determinadas circunstancias. Los puntos asignados en las historias de usuario varían de acuerdo con los puntos que puede ofrecer el miembro del equipo asignado a dicha labor y de la complejidad de la tarea. Sin embargo, estos puntos fueron disminuyendo conforme el equipo fue conociendo la operación y particularidades de la organización.

Este conocimiento se convirtió en mejoras en el desarrollo permitiéndonos incrementar el ritmo de trabajo de todo el equipo, aunque también existieron situaciones que afectaron el rendimiento del equipo, y, por ende, en los tiempos de entrega del desarrollo. Por lo que tuve que realizar ajustes en equipo de trabajo para cumplir con los objetivos.

Las historias de usuario se descomponen en tareas todavía aún más puntuales. Esto se realizó con mecanismos que nos brindó Scrum en dónde junto con mi equipo se desglosaron las historias de usuario en acciones concretas para su realización y las representamos como lo muestra la tabla 5.

Tabla 5 - Tareas durante un sprint. Fuente: elaboración propia.

Sprint #1				
Historias	Tareas			
	Por hacer	En progreso	En prueba	Realizadas
7.- Importes entrada y salida	<ul style="list-style-type: none"> - Obtener información institucional. - Diseñar y desarrollar trabajo ETL. - Aplicar regla de negocio. - Cargar información. - Validar Información con usuario. 			

La tabla de anterior es una representación de lo que trabajamos como tablero de Scrum, en la que se listaron todas las historias de usuario existentes y cada una de ellas se desglosó en tareas más puntuales. Cada una de las nuevas tareas atravesó por cuatro etapas: al inicio se estipularon las tareas que comprenden la historia de usuario, estas son las tareas por hacer, cuando dio comienzo su realización se movió a la etapa de progreso, la validación y verificación de las tareas se realizó en la etapa de pruebas y finalmente cuando culminaba la tarea la situó en la etapa de tareas realizadas.

Cabe mencionar que se realizó un tablero de Scrum por cada uno de los *sprint* que se conformó del proyecto. En conjunto se determinó que cada *sprint* tuviera una duración de cuatro semanas teniendo la realización de este proyecto una duración de diez *sprint*.

Implementación

Con las condiciones y plan de trabajo establecidos fue el turno de pasar a la fase de implementación, en dónde se puso en marcha el desarrollo de los entregables que conforme se desarrollaron, fueron sufriendo cambios que coloque en la lista de priorización de tareas que fue constantemente actualizada. También surgieron nuevos requerimientos que se trataron con la organización para medir su impacto en el proyecto.

Creación de entregables

Para convertir la planeación en entregables tangibles se puso en marcha el desarrollo del proyecto. En primera instancia nos acercamos al personal encargado de administrar los sistemas de información institucional y solicitamos el acceso a los sistemas para conocer el tipo de información que genera la organización SEDCOD.

En este primer acercamiento con los sistemas institucionales se generó la tabla 6, referente a las diversas fuentes de información.

Tabla 6 - Fuentes de datos institucionales. Fuente: elaboración propia.

Fuente de Datos	Sistema
Oracle	<ul style="list-style-type: none">• Entradas• Salidas
SQL Server	<ul style="list-style-type: none">• Almacenaje
MySQL	<ul style="list-style-type: none">• Catálogos Institucionales
Hojas de Cálculo	<ul style="list-style-type: none">• Información no Sistematizada

El reconocimiento de los sistemas institucionales nos permitió conocer los tipos de motores que se utilizaban y saber a qué tipo de datos nos enfrentamos debido a que cada motor tiene sus propios tipos de datos. También dentro de este acercamiento con el personal a cargo de los sistemas institucionales solicité el diagrama de entidad relación y el diccionario de datos, sin embargo, esta información no se encontraba totalmente disponible, no por cuestiones de privacidad si no que no se había requerido hasta ahora, lo que representó un detalle a considerar.

El modelado de datos que se propuso para el Data Warehouse fue un esquema del tipo constelación (Bernabeu, 2010) que también se le conoce como la unión de un par de esquemas de tipo estrella. El esquema de constelación cuenta con una serie de tablas de hechos que yacen en el centro del modelo y están relacionadas con sus respectivas tablas de dimensiones, véase figura 18.

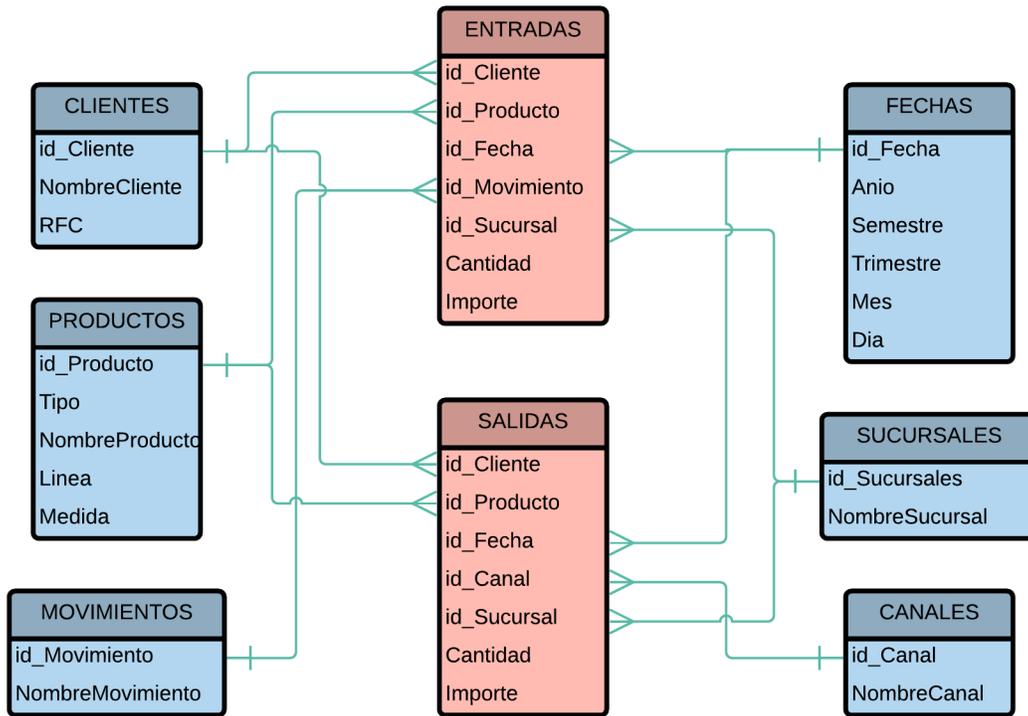


Figura 18 - Diagrama entidad relación de entradas y salidas. Fuente: elaboración propia.

Dentro de la plataforma ETL se generaron las conexiones hacia los sistemas institucionales, este proceso se realizó una sola vez ya que dicha conexión se reutilizó dentro de la plataforma de Talend, véase figura 19.

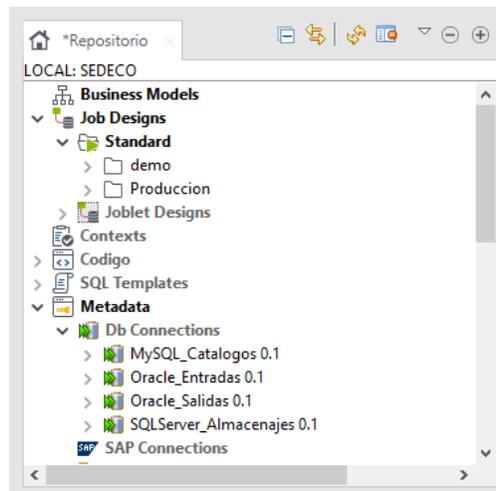


Figura 19 - Panel de repositorio. Fuente: Talend Data Integrator.

Una de las acciones concretas para depositar la información en el motor Greenplum provenientes de los sistemas institucionales fue la creación de los trabajos ETL, estos trabajos se componen de elementos que nos brindó la aplicación de Talend, una interfaz gráfica en la que fue posible diseñar los trabajos ETL mediante la unión e interacción de componentes, véase figura 20.

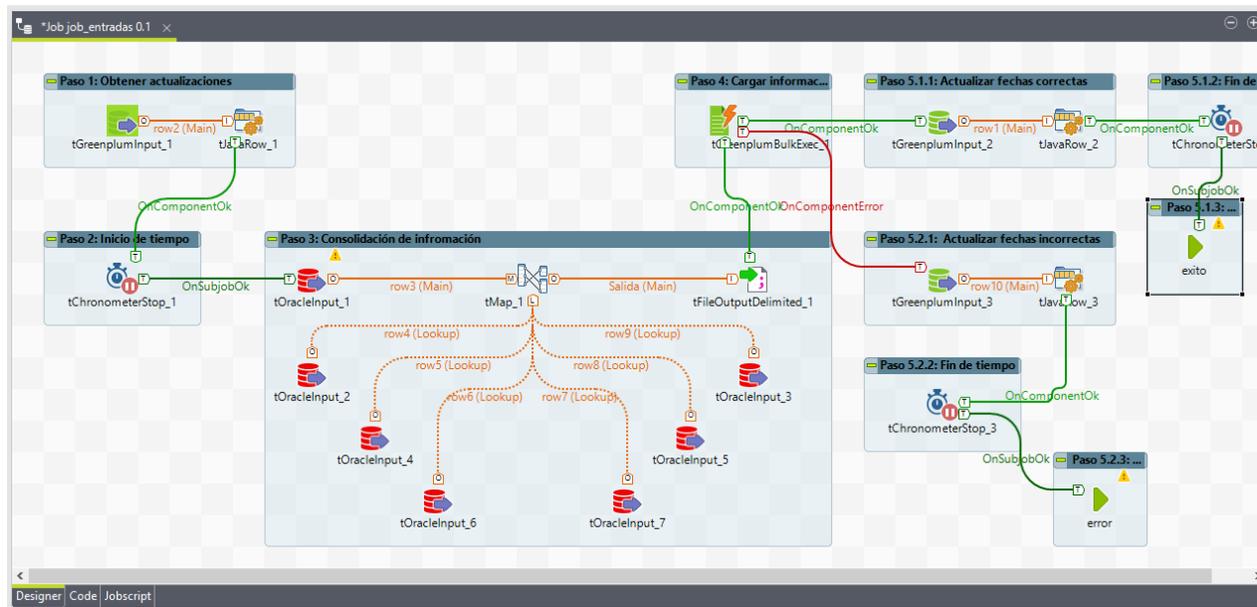


Figura 20 - Panel de trabajo. Fuente: Talend Data Integrator.

La imagen anterior representa uno de los trabajos ETL desarrollados, lo que realicé en este trabajo ETL fue la carga de una tabla de hechos que se llevó a cabo bajo una serie de pasos que se enumeran a continuación.

- Paso 1: Obtener actualizaciones

El trabajo ETL inicio con la obtención de fechas que se actualizaron durante la ejecución del trabajo, estas fechas se obtuvieron mediante el desarrollo de una función en la base de datos Greenplum, la cual recibe como parámetro el nombre de la tabla que se va a actualizar y devuelve las fechas que se actualizarán. Esto se logró almacenar en variables del lenguaje Java con el uso de los componentes que permitieron el uso del lenguaje de programación.

- Paso 2: Inicio de tiempo

En este paso se inició un cronómetro con la finalidad de tener información de la duración de los trabajos ETL, esto se realiza de manera adicional a la oferta que brinda la herramienta.

- Paso 3: Consolidación de información

La consolidación de información es el paso en donde se realizan las consultas hacia las bases de datos institucionales de la organización. Mediante el uso de los componentes ETL se realizó la consulta SQL a los motores de bases de datos y los resultados de las consultas se consolidaron en un componente llamado “tMap”, véase figura 21, dentro de este componente se realiza la unión de los resultados a través de los campos que se establecieron como llaves y se aplicaron las reglas de negocio correspondientes.

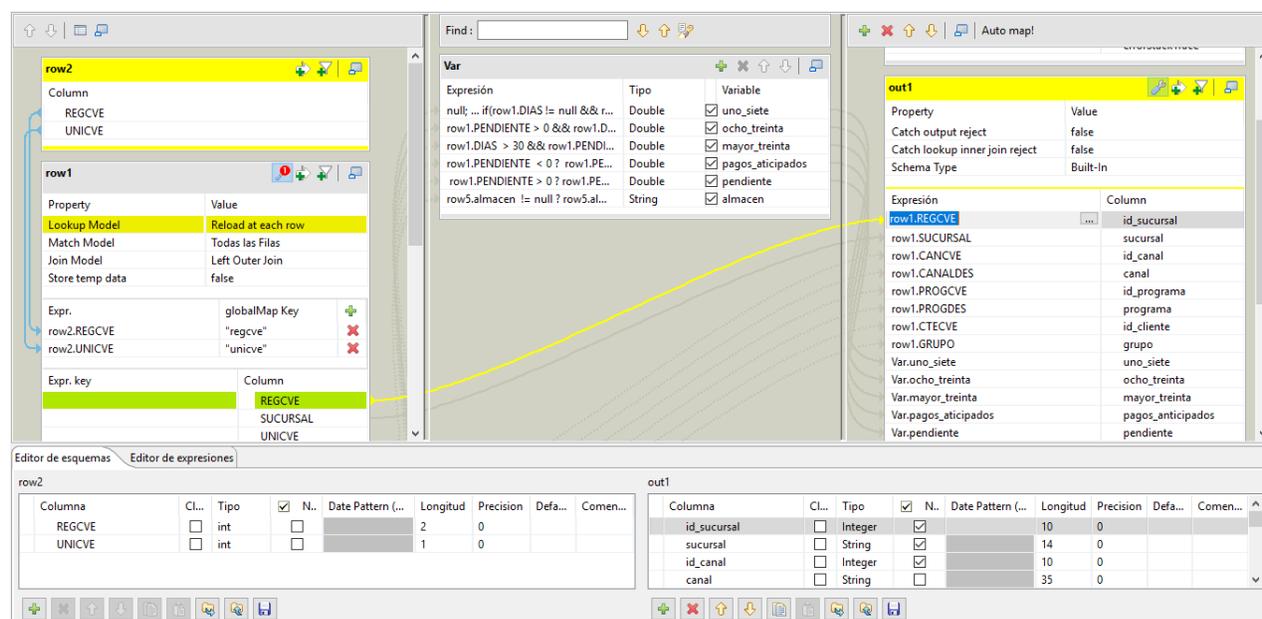


Figura 21 - Panel "tMap". Fuente: Talend Data Integrator.

También dentro de este componente se llevó a cabo el mapeo de las columnas hacia la tabla que se designó en el Data Warehouse para su posterior carga, adicionalmente dentro de este componente se realizaron tareas de limpieza de datos, debido a los diversos formatos en los tipos de datos e inclusive una inspección general a la ortografía del catálogo.

Físicamente los datos son depositados en un archivo binario temporal que fue utilizado en el proceso de carga además de que brindó la oportunidad de visualizar la información en el proceso de *debug* de los trabajos ETL.

- Paso 4: Cargar información

La carga de la información correspondió a un componente del tipo “BulkExec” que realiza un volcado de los datos usando métodos de carga que varían conforme al motor de base de datos. Para el caso particular de Greenplum el componente de ETL se encargó de ejecutar una sentencia SQL definida en el motor como COPY que se encargó de mover los datos del sistema de archivos a las tablas del Data Warehouse.

Dentro de la lógica que utilicé en la creación de trabajos ETL, dentro del componente de carga se generó una condición que determina si la carga se había realizado de manera exitosa o por el contrario se generó un error en la carga. Estos mecanismos me ayudaron en el proceso de *debugeo* de los trabajos ETL, sin embargo, su uso también abarcó el monitoreo ya en ambientes productivos. Dependiendo si fue exitosa la carga o no, el flujo sigue una serie de pasos posteriores.

- Paso 5.1.1 y 5.2.1: Actualizar fechas correctas e incorrectas

Como se visualiza en la imagen al inicio de la descripción del trabajo ETL, los pasos posteriores a la carga de la información se abre una condición de éxito y error. Los componentes utilizados a la salida del componente de carga son los mismos para ambos casos de la condición y se encargaron de generar metadatos del estatus del trabajo ETL permitiéndome contar con información sobre la ejecución de los trabajos ETL en ambientes productivos.

- Paso 5.1.2 y 5.2.2: Fin de tiempo

Posteriormente de la generación de metadatos se agregó un componente que pusiera fin al tiempo de ejecución que inicio en el paso 2. En este punto la información institucional ya se encontraba depositada en las tablas del motor Greenplum sin embargo, a determinados trabajos ETL les asigne una dependencia como parte del flujo de trabajo por lo que este flujo podría seguir ejecutando otros trabajos.

- Paso 5.1.3 y 5.2.3: Notificar

Finalmente, una vez concluido el flujo del trabajo ejecutado satisfactoriamente o no, generé una notificación de éxito y de error a los interesados en la información generada con la finalidad de informar la disponibilidad de la información o notificar un error en el proceso de carga de la información al equipo de desarrollo, esto último nos ayudó al equipo y a mí a la temprana detección de mantenimiento preventivo.

De esta manera fue como desarrollé algunos de los trabajos ETL que consolidaron la información institucional. Se realizaron alrededor de 90 trabajos divididos en los módulos del proyecto. La siguiente fase corresponde a generar el conocimiento con la información que se recabó, este trabajo se realizó en la base de datos Greenplum, que a través de diversos framework nos permitió el uso avanzado de funciones para el Data Warehouse.

El framework utilizado fue el creado por la fundación Apache conocido como MADLib, cuya utilización es de código abierto. Con el uso de las funciones de MADLib fue posible realizar transformaciones a los datos, generación de modelos predictivos, etc. A continuación, se presenta un ejemplo de consulta de datos.

```
%psql.sql SELECT (MADLib.one_way_anova (  
CASE WHEN carrier = 'US' THEN 1  
WHEN carrier = 'DL' THEN 2  
ELSE NULL  
END,  
Arrdelayminutes  
)).* FROM faa.otp_r;
```

Con ayuda del framework se pueden hacer uso de las funciones como *one_way_anova* para la generación de información.

Dentro del proceso de creación de entregables desarrolle estas actividades directamente involucradas en la operación del proyecto, este proceso fue iterativo y se fue desarrollando conforme a las tareas previstas en cada sprint.

Reuniones diarias y de avance

Conforme al marco de la metodología Scrum, participe en diversas reuniones: al inicio de proyecto las reuniones de requerimientos, entre otras. Si bien éstas aportaron interés a la realización del proyecto, también existieron otro tipo de reuniones que tuvieron un impacto mayor: las reuniones diarias.

Las reuniones diarias representaron un pilar para el desarrollo del proyecto y del equipo de trabajo. Estas reuniones se llevaron a cabo al inicio de la jornada laboral diaria, cada integrante del equipo se encargó de comentar una serie de aspectos.

Primero comentábamos el trabajo realizado hasta el momento de tal manera conocíamos el avance real del proceso y se detectaron retrasos en las tareas asignadas a cada integrante del equipo.

El siguiente aspecto que se comentaban eran las tareas que se realizarían en el transcurso de la jornada de trabajo, así como los posibles inconvenientes que tuviéramos en su realización. Aquí es cuando pude realizar ajustes, porque si un compañero se atrasaba, otros podríamos apoyarlo siempre y cuando no afectará considerablemente el ritmo de trabajo de los demás integrantes.

Otro punto que tratamos el equipo y yo en las reuniones, fue si la dificultad de la tarea era mayor a la que se estimó y de esta manera se realizaba un ajuste en el calendario de actividades. En este punto actualicé la lista de tareas basado en el valor que entregaron en la realización de cada una de ellas.

De esta manera se llevaron a cabo las reuniones diarias con el equipo, las cuales representaron un escenario de comunicación abierta entre los miembros del equipo.

Revisión y retroalimentación

La etapa de revisión y retroalimentación corresponde a todas aquellas actividades que realicé una vez concluidos los entregables de cada sprint. Estas actividades se basaron en dos principios: validar el sprint y la retroalimentación del sprint. Esto se tradujo en la validación de las tareas realizadas en el desarrollo de los trabajos ETL y su concordancia con los criterios de aceptación y la retroalimentación de estos procesos.

Validar el Sprint

La validación de cada uno de los sprint corresponde a la evaluación de la calidad de los entregables, esto se refirió a la función que realizó cada uno de los entregables concuerde con los requerimientos de los usuarios finales, a través de los criterios de aceptación.

Durante la toma de requerimientos con los usuarios e interesados en el proyecto se estipularon los criterios de aceptación para las historias de usuario generadas en la etapa temprana del proyecto. Con la finalización del desarrollo de los entregables se debían cumplir los objetivos de cada una de las historias, este proceso de validación se llevó a cabo con el usuario final en el término de cada sprint.

Es así como se realizó el empoderamiento de los usuarios finales durante todo el desarrollo del proyecto. Los usuarios participaron en diversas actividades lo que provocó que también conocieran el estatus del proyecto conforme este se fue desarrollando. Es por eso que se determinó generar esta responsabilidad de validación en conjunto.

Retroalimentación

Al igual que el objetivo de las reuniones diarias, la retroalimentación del equipo de trabajo al término de cada sprint presentó una oportunidad para realizar los ajustes pertinentes en consecución de nuestros objetivos en el proyecto.

Al término de cada sprint, lleve a cabo una reunión con la finalidad de conocer los puntos de vista de cada uno de los integrantes sobre el trabajo que habíamos realizado. Estas opiniones abarcaron el buen trabajo que se realizó y que se conservó en sprint subsecuentes. Otro aspecto fue sobre las cosas que debimos mejorar y que hicimos mal. Finalmente analizamos las actividades que debían ser agregadas para complementar lo bien hecho, incluyendo capacitaciones y/o herramientas de software.

Liberación

La etapa final del proyecto corresponde a la liberación de los entregables y una retroalimentación final del proyecto. En esta etapa final se realizó la entrega del trabajo realizado y la puesta en marcha del proyecto, cabe destacar que esto es simbólico, ya que la entrega constante de valor permitió a la organización contar con elementos tangibles del proyecto lo más pronto posible.

Liberación de entregables

Para la liberación de los entregables el equipo de trabajo y yo nos coordinamos con el área de TI de la organización SEDCOD para establecer el calendario de liberación ya que desde el término del primer sprint se comenzó con la liberación de las actividades correspondientes.

En estas reuniones se tocaron temas de índole técnica y administrativa, debido a que sólo en determinados horarios establecidos por la organización se pudo acceder a la información institucional por lo que requirió que realizará una planeación sobre cuando pudimos acceder a la información y cuándo notificar en qué momento se encontraría disponible en el Data Warehouse.

Adicionalmente el equipo y yo nos encargamos de realizar una pequeña capacitación al personal de TI, con la finalidad de que se familiarizaran con las nuevas herramientas para su monitoreo y posibles incidentes.

En cuanto a la dependencia de desarrollo de nuestros sprint con otros sprint de otros equipos de trabajo, se comentó al inicio de este informe, que este trabajo representó una parte de un proyecto aún más grande. En coordinación con los Scrum Master de los equipos colaboradores, notifiqué oportunamente la culminación de cada uno de los sprint y apoyé con los protocolos de comunicación para que la aplicación que consumió la información tuviera todos los parámetros para su configuración.

Retroalimentación del proyecto

La reunión final de retroalimentación del proyecto siguió el mismo principio de las reuniones diarias y las reuniones realizadas al término de cada sprint: comunicación abierta para la mejora continua. Participé en las reuniones finales del proyecto y es que éstas se dividieron en dos, la primera reunión se llevó a cabo con los dirigentes e interesados en el proyecto por parte de la organización SEDCOD y la segunda reunión se realizó con el equipo de trabajo.

En la reunión con los dirigentes de la organización SEDCOD, se comunicaron los aspectos que se tendrían que considerar una vez concluido el desarrollo del proyecto como lo fue el mantenimiento a los servidores y a nivel de base de datos dentro del Data Warehouse.

Otro aspecto importante comentado en la reunión fue un informe de mejoras que realizamos como resultados de las áreas de oportunidad detectadas en las reuniones de término de los *sprint*. Estas mejoras comprenden aspectos en la operación interna de la organización SEDCOD y es que mi equipo y yo detectamos posibles mejoras que bajo un análisis más exhaustivo generaría beneficios para la organización.

En la reunión interna con el equipo de trabajo seguí la dinámica de las reuniones diarias y de término de sprint, las actividades que nos llevaron a la consecución de nuestros objetivos dentro del proyecto que debemos aplicar para futuros proyectos, estas buenas prácticas se deben preservar para el éxito de nosotros como empresa y también en el ámbito personal. Nuestras áreas de oportunidad para pulir esos aspectos que sigan incrementando nuestro ritmo de trabajo, permitiendo que el equipo adquiera madurez tanto técnica como de desarrollo personal.

El desarrollo de habilidades y técnicas nos permitió al equipo y a mí realizar una retroalimentación de la cantidad de actividades que fuimos capaces de realizar en cada uno de los sprint, comparando los puntos que esfuerzo que ofrecimos en la realización del primer sprint con el esfuerzo realizado en la etapa final del proyecto.

Es así como a través de las cinco fases que constituyen una parte del marco de la metodología Scrum se desarrolló el proyecto “Implementación, Diseño y Consolidación de Fuentes para un Data Warehouse”, adoptando una forma de trabajo ágil y que permite la entrega constante de valor a las organizaciones. Un desarrollo adaptable e iterativo que rompe con el paradigma de las metodologías tradicionales.

Participación profesional

Mi participación profesional en el proyecto “Implementación, Diseño y Consolidación de Fuentes para un Data Warehouse” fue un módulo que perteneció a un proyecto aún más grande, que reunió a un gran número de equipos de trabajo y que bajo la metodología antes descrita se logró la entera satisfacción del cliente. Así mismo, en este informe sólo se presenta lo correspondiente al módulo en el que me tocó colaborar. Esta participación estuvo dividida en los dos roles que desempeñé a lo largo de la duración del proyecto, los cuales se describirán a continuación.

Scrum Master

Mi rol como Scrum Master, como lo menciona la metodología Scrum, fue la de facilitador de tareas y recursos para mi equipo de trabajo. Bajo este rol estuve a cargo de un equipo de trabajo conformado por Desarrolladores ETL y Documentadores. Pero como mencioné anteriormente, este proyecto fue un módulo de un proyecto aún más grande por lo que tuve la oportunidad de colaborar con otros equipos durante el desarrollo de este proyecto.

Así que la constante comunicación con otros integrantes fue la clave del éxito para cumplir con el objetivo, tanto para nuestro equipo como los demás. En este rol desarrolle las siguientes actividades, las cuales describiré en las cinco fases de SCRUM (SBOK Guide, 2016).

Inicio

- Objetivos y alcance

En la toma de requerimientos generales al inicio del proyecto participé en reuniones con todos los equipos involucrados en el proyecto, para establecer en conjunto los objetivos generales y particulares del módulo de Data Warehouse, así como el alcance que tuvo. De esta manera se identificaron las dependencias entre módulos, dando como resultado la visión del proyecto, es decir el panorama general del proyecto, el cual comuniqué hacia todo el equipo.

- Identificación de roles

Durante los primeros acercamientos con la organización SEDCOD, fue de suma importancia identificar a las personas involucradas en el mismo, de esta manera garantizamos que todas las personas que participaron en el desarrollo de los entregables generaran cada una de ellas un sentimiento empoderamiento tanto de los usuarios finales como los responsables dentro de la organización.

Por ello fue necesario identificar a los *Patrocinadores* del proyecto, quienes proveen los recursos materiales y financieros, el rol de *Product Owner* quien fue el enlace con la organización para dar seguimiento tanto en la operación como al avance del proyecto, además de brindarnos las necesidades del usuario y de esta manera generar una priorización de las necesidades del usuario como de la organización. Finalmente se identificó a los *Usuarios Finales*, quienes nos apoyaron en diversas etapas de toma de requerimientos, verificación y validación.

- Formar el Equipo

La elección de las personas para la conformación del equipo, no fue decisión mía, sin embargo, con el conocimiento previo de los objetivos y el alcance, fue necesario sugerir el tipo de perfiles que necesitaba para poder lograr dichos objetivos. Los perfiles propuestos fueron de personas que tuvieran afinidad a las bases de datos y conocimiento de diversos motores de bases de datos y del manejo de la plataforma ETL, adicionalmente se requirió de un perfil que apoyará en la generación de la documentación técnica del trabajo que se realizó.

Si bien algunas de estas características son técnicas, es necesario que estos perfiles también cuenten con otro tipo de habilidades conocidas como *soft skills*, y es que en este proyecto el conocimiento técnico no es suficiente para lograr el éxito.

Tener personal multifuncional, permitió al equipo de trabajo contar con la flexibilidad de que cualquier integrante podría cumplir con las tareas del otro integrante, de esta manera se logró mantener el ritmo de trabajo, ya que cualquiera de los desarrolladores ETL y/o documentador, podría llevar a cabo reuniones con los usuarios finales con el objetivo de tomar requerimientos, comprender reglas de negocio, informar avances, etc.

Planeación y estimación

- Historias de Usuario

La definición de los requerimientos de la organización como de los usuarios finales, la plasmé en forma de *historias de usuario* las cuales nos permitieron a mi equipo y a mí comprender las tareas a realizar De igual manera establecimos el criterio de aceptación para cada historia, es decir, qué parámetros se consideraría para que cada historia

cumpliera con lo especificado por el usuario y así asegurar la calidad de las actividades realizadas.

Al final dentro de esta etapa generamos una organización de estas historias de usuario, procurando iniciar con las que más aportarán valor a la organización y al usuario final.

- Creación y estimación de tareas

Con las historias de usuario organizadas por jerarquía, el siguiente paso fue dividir el requerimiento en tareas a realizar, con ayuda de mi equipo se logró establecer el plan de tareas para cada una de las historias.

Esta etapa tuvo una importancia vital en del desarrollo del proyecto, debido a que en este punto se generó el plan de trabajo para llevar a cabo las tareas técnicas, con las metodologías tradicionales de desarrollo, este plan no habría tenido la capacidad de realizar un ajuste en medio del desarrollo, es por eso que el uso de la metodología Scrum permite ajustes en la creación y estimación de tareas.

Cada tarea creada, recibe puntos de esfuerzo que no es más que un método para definir la complejidad de las tareas. Entre más compleja, más puntos de esfuerzo recibirá la tarea. Este mecanismo me apoyó en la estimación de las tareas.

Para afrontar la realización de cada una de ellas, fue necesario conocer el esfuerzo que cada integrante del equipo puede aportar, de esta manera obtuve un tiempo aproximado para la realización de cada tarea. Con el tiempo aproximado ya establecido pude generar la cantidad de tareas que seríamos capaces de resolver en determinado lapso, a esto se le llama *sprint*, este proyecto tuvo diez sprint.

Implementación

- Creación de los entregables

Esta es la etapa de desarrollo del proyecto, mi rol como Scrum Master fue la de facilitar recursos y materiales para ayudar a los integrantes del equipo de trabajo a cumplir con cada una de las tareas asignadas y objetivos establecidos anteriormente. Asegurarme que todo el equipo las conociera y que fuesen claras para cada integrante a lo largo de cada uno de los módulos desarrollados, fue otra de mis labores en esta etapa, además de la resolución de conflictos técnicos y del personal.

- Reuniones diarias y de avance

Informar el avance, áreas de oportunidad, tareas a realizar, incidentes, etc. Fueron cubiertos en reuniones diarias con el equipo de trabajo y reuniones de avance con los responsables del proyecto dentro de la organización del cliente.

Con el equipo de trabajo se realizaba una reunión diaria para conocer el trabajo que se llevaría a cabo ese mismo día, los posibles inconvenientes y/o áreas de oportunidad para mejorar. Este tipo de reuniones me permitió resolver diversas problemáticas, que en un inicio no fueron consideradas, como un cambio de usuario final, por ejemplo. También nos permitió conocer en qué momento era necesario realizar un mantenimiento preventivo a la plataforma.

En general, estas reuniones diarias con el equipo de trabajo ofrecían un avance real del proyecto y daban a conocer si había que realizar algún tipo de ajuste, permitiéndonos ser proactivos y no reactivos.

Con la organización del cliente, las reuniones se realizaban para conocer el estatus y avance del proyecto. Como en las reuniones diarias con el equipo de trabajo esta información ya me era proporcionada y me correspondía dar un informe ejecutivo con la información técnica que obtenía de las reuniones diarias.

La frecuencia con las que se realizaban estas reuniones eran semanales sin embargo muchas ocasiones se realizaban bajo demanda de los interesados del proyecto en la organización del cliente, en donde los interesados veían continuamente los entregables tangibles del proyecto, de esta manera cumplimos con la entrega constante de valor.

Revisión y retroalimentación

- Validar el Sprint

Antes de dar por terminado cada uno de los *sprint* del proyecto, se realizó la validación de las tareas que lo conformaron, asegurándome que cumpliera con los criterios de aceptación estipulados en la etapa de creación de historias de usuario con el responsable del proyecto en la organización SEDCOD.

- Retroalimentación

Al finalizar cada sprint, llevé a cabo una reunión con todos los integrantes del equipo de trabajo para conocer su opinión sobre el trabajo realizado en el lapso de duración de cada sprint. El objetivo fue conocer las áreas de oportunidad y lo que podíamos mejorar, esta etapa, como en otras, me permitía realizar ajustes tanto técnicos como en el ámbito personal con cada uno de los integrantes de mi equipo de trabajo.

Estos ajustes podían realizarse en esta etapa justo después de terminar una carga importante de trabajo y antes de comenzar una nueva. Así mismo recibí de cada uno de los integrantes del equipo la retroalimentación de mi desempeño como Scrum Master.

Esta retroalimentación más allá de conocer lo que se ha hecho bien o lo que se podía mejorar, permitieron a los integrantes del equipo de trabajo tener un empoderamiento

sobre el proyecto lo que también autogeneraba compromiso por parte de ellos. Este empoderamiento y compromiso iba acompañado de beneficios para la organización del cliente, debido a que el equipo mejoraba técnicas y habilidad gradualmente durante el desarrollo del proyecto, minimizando los riesgos y el tiempo de cada sprint.

Liberación

- Liberación de entregables

Finalmente, la puesta en marcha en ambientes productivos de cada uno de los sprint. En esta etapa se coordinó con personal de la organización la liberación del sprint generando. Así se generó la entrega constante de valor al cliente, en consecuencia, los usuarios finales lograron visualizar el trabajo realizado por el equipo de manera gradual y constante. Adicionalmente las actividades que realicé con el personal de TI de la organización fue el desarrollo de un calendario para acceder a la información institucional.

Además de cada uno de los entregables realizados también se informó a los otros equipos de trabajos de los otros módulos dicha liberación para continuar con los flujos de trabajo establecidos en la fase de inicio del proyecto. Además, se realizó en conjunto con el integrante del equipo encargado de la documentación una revisión de los entregables físicos del proyecto.

- Retroalimentación del proyecto

Al igual que en la etapa de retroalimentación del sprint, llevé a cabo una retroalimentación con el equipo de trabajo, pero acerca del proyecto. Generamos una reunión con las lecciones aprendidas a lo largo del ciclo de vida del proyecto, sugerencias y/o solicitudes sobre capacitaciones o inclusive cosas que debían mantenerse para así establecer una guía para futuros proyectos y que como profesionistas busquemos la mejora continua. Esta retroalimentación no sólo se llevó con el equipo a mi cargo, sino con los demás equipos una vez concluido el proyecto.

Desarrollador ETL

Si bien uno de mis roles fue el de Scrum Master, la metodología Scrum indica que deben conformar el equipo de trabajo personas multifuncionales, de esta manera también desempeñé un rol como desarrollador ETL, el cual se desarrolló en las siguientes fases.

Planeación y estimación

- Historias de usuario

Realicé una reunión con los usuarios finales con el objetivo de establecer los requerimientos para la obtención de información. Estas reuniones sirvieron para esclarecer las necesidades del usuario final lo cual nos permitió delimitar la tarea correspondiente a determinado requerimiento, dar a conocer el alcance al usuario final y conocer la operación interna de la organización.

Plasmé en historias de usuario estos requerimientos y/o necesidades para que el Scrum Master y los demás compañeros del equipo mantengamos la realización de las tareas enfocadas a dichas historias.

- Creación y estimación de tareas

Con los requerimientos y/o necesidades establecidas en las historias de usuario se realizó el análisis para establecer la cantidad de esfuerzo y tiempo que requiere del equipo de trabajo la realización de cada una de las historias, las cuales deben ser divididas en tareas dependiendo de la complejidad de cada una y si es posible subdividir esta tarea en otras de menor complejidad, esta desagregación de las tareas me permitió entregar valor inmediatamente conforme se fueron realizando cada una de las tareas.

Las habilidades técnicas de cada uno de los miembros del equipo influyeron en el tiempo de realización: mientras algunas tareas de extracción de información las realicé con consultas simples SQL, como seleccionar todo de una tabla, otras requirieron la elaboración de complejos trabajos ETL y de análisis de negocio interno para consolidar las fuentes de información. Por lo que los puntos de esfuerzo determinaron un aproximado de la duración de las tareas.

Implementación

- Creación de entregables

En la etapa de desarrollo del proyecto las funciones que desempeñé fueron el diseño y creación de tablas en la base de datos de Greenplum, además de creación de objetos como vistas materializadas, secuencias, índices y mantenimientos a la base de datos.

Otra de las funciones fue la creación de los trabajos ETL, los cuales también contaban con un diseño previo y se generó un flujo por donde se realizaba la conexión a las bases de datos institucionales para extraer la información. Si esta información no necesitaba la aplicación de reglas de negocio con el uso de la plataforma ETL se depositaba en el Data Warehouse. Por el contrario, si se tenía que aplicar una regla de negocio, se hacía uso de la plataforma ETL y sus diversos componentes para aplicar la regla de negocio; estas acciones correspondían a realizar una operación aritmética por un índice de inflación dictaminado la organización, o agrupar diversas partidas presupuestales, asignar determinado valor de acuerdo a una clasificación, etc.

La optimización del funcionamiento del Data Warehouse se realizó bajo la generación de estadísticas a los objetos de base de datos Greenplum, la creación de índices y optimización de consultas mediante el análisis de los planes de ejecución de las consultas SQL, finalmente la creación y programación de respaldos de metadatos y datos.

- Reuniones Diarias y de Avance

En las reuniones diarias mi participación fue la de notificar las tareas realizadas y las que realizaría ese mismo día, Esto ayudó a mantener un estatus del proyecto casi tiempo real, notificando cuando las cosas no salieron como originalmente se planearon, lo que permitió al Scrum Master ajustar el plan de trabajo que tuve en ese momento. También brindé mi apoyo a los demás miembros del equipo cuando teníamos dificultades.

Revisión y retroalimentación

- Validar el Sprint

Al finalizar las tareas y subtareas que comprendieron las historias de usuario asignadas a cada integrante del equipo, me correspondía asegurar el cumplimiento de los criterios de aceptación que se había estipulado a cada una de las historias de usuario. Este cumplimiento de los criterios de aceptación se da en el marco de cumplimiento de calidad, en la que nos asegurábamos de que las necesidades del usuario final y de los interesados dentro de la organización se cumpliera coincidiera con el trabajo realizado.

De acuerdo al criterio, se realizó a comparativa de cifras de validación en donde comparábamos con reportes equivalentes la información generada en el Data Warehouse. Con información donde no había reportes de comparación se generó una matriz de pruebas tomando una muestra representativa de la información o inclusive evaluación manual con apoyo de los usuarios finales. Este proceso se realizaba al término de cada tarea por lo que no había que esperarse al término del sprint para realizar la validación con el usuario final.

- Retroalimentación

Al finalizar cada sprint, se realizó una reunión con el Scrum Master y el equipo de trabajo donde exponíamos tres puntos importantes: las áreas de oportunidad por mejorar, lo que debía seguir manteniéndose y lo que debía agregarse para fortalecer el ritmo de trabajo del equipo. Determinadas situaciones no podían solucionarse por su propia naturaleza o por factores externos como lo fue estipular un horario de reuniones con algunos de los usuarios finales, que desafortunadamente no nos pudieron asignar un lapso por sus condiciones de trabajo en la operación diaria.

Por otro lado, otras fueron solucionadas satisfactoriamente en donde el trabajo lo realizábamos en equipos de escritorio en los cuales perdíamos el progreso de nuestras tareas cuando estando en las instalaciones del cliente el funcionamiento del sistema eléctrico era irregular, esto se solucionó con sistemas de alimentación ininterrumpida.

Liberación

- Liberación de entregables

Tras la validación de las tareas y subtareas realizadas por el equipo de trabajo, la puesta en marcha consistió de otorgar permisos a los usuarios a la información puesto que está ya se encontraba en el Data Warehouse solo faltaban los permisos de acceso para que la capa de aplicación y los propios usuarios empezarán a consumir la información. Además, creé un plan para los mantenimientos que se requiriera en el Data Warehouse para su correcto funcionamiento a lo largo de la vida del proyecto, la cual fue compartida con el área de TI de la organización SEDCOD.

- Retroalimentación del Proyecto

Al término del proyecto, se realizó nuevamente una reunión con el Scrum Master y los integrantes del equipo, la cual consistió en algo similar de lo realizado en el cierre de cada sprint, en donde se comentaron los puntos favorables como las áreas de oportunidad sin embargo al referirse al final del proyecto nos sirvió para evaluarnos a nosotros como equipo y como empresa donde los comentarios de cada uno de los integrantes del equipo realizamos una serie de lecciones aprendidas del proyecto.

También realicé una autoevaluación y calificamos tanto al Scrum Master como al resto del equipo. Esta dinámica de retroalimentación me permitió darme cuenta de errores técnicos que cometí a la hora de crear los trabajos de ETL en dónde no estaba optimizando el uso del espacio en el Data Warehouse, por mencionar alguno de ellos. De igual manera sugerí las posibles capacitaciones que en el futuro posiblemente nos permitan abordar temas que técnicamente desconocemos y a las cuales nos enfrentamos en el desarrollo del proyecto pero que invertimos tiempo en su solución.

De esta manera fue como desarrollé mi participación profesional en el proyecto “Implementación, Diseño y Consolidación de Fuentes para un Data Warehouse”, en el cual me involucré en dos roles distintos. El rol como desarrollador ETL fungí como el generador y desarrollador de los entregables del proyecto a un nivel que mayoritariamente fue muy técnico, enfocado a las bases de datos, sin embargo, también me permitió ser participe en cuestiones no tan técnicas, pero con el mismo enfoque de cumplir los objetivos.

El rol como Scrum Master desarrollado por primera vez me dio a conocer que la administración de proyectos es algo que requiere de diversas habilidades principalmente trasladar las necesidades del cliente hacia lo técnico, es decir lo tangible que resultará el proyecto. Pero no sólo fue eso, sino la administración de un equipo de trabajo donde no existe un libro o una guía de cómo hacer que un equipo rinda a lo largo de la duración de un proyecto pero que con una metodología que trabajo muy madura en el equipo permitió el éxito para cada una de las entidades.

Resultados obtenidos

Tras la implementación y desarrollo del proyecto “Implementación, Diseño y Consolidación de Fuentes para un Data Warehouse”, logramos desarrollar trabajos ETL para la eficiente extracción de información para cumplir con las necesidades tanto de los directivos como los usuarios finales, brindando y generando información estratégica para la toma de decisiones en la organización SEDCOD. A continuación, se listan los resultados:

- Se implementó satisfactoriamente una plataforma de ETL con trabajos de extracción autónomos siendo un total de 95 trabajos ETL desarrollados para la consolidación de diversas fuentes de información institucional.
- Fueron un total de 7 sistemas institucionales que se consolidaron en el motor de base de datos del Data Warehouse, teniendo como principales fuentes manejadores de bases de datos Oracle, SQL Server y MySql. Adicionalmente se trataron archivos planos como hojas de cálculo para la información no sistematizada.
- Se implementó y se diseñó una estructura de Data Warehouse que represento para la organización SEDCOD contar con un sistema de toma de decisiones estratégico y táctico. Esto se tradujo en contar con información homogénea proveniente de diversos sistemas generadores de información, resultando en una disminución considerable en los tiempos de obtención de datos y de acceso a los sistemas transaccionales de la operación diaria.
- Se logró una adaptación de la metodología de trabajo Scrum a la operación de la organización SEDCOD, la cual representó una integración y colaboración de diversos roles dentro de la organización con un equipo de trabajo multifuncional y se generó una entrega continua de valor hacia la organización desde las etapas tempranas del proyecto hasta la culminación de éste.

Conclusiones

Se cumplieron los objetivos general y particulares del proyecto, con diversas áreas de oportunidad para cada uno de los integrantes del equipo, pero con un ligero retraso de casi un mes a lo planeado, pero cumpliendo la expectativas de la organización y de su personal.

Ser consultor externo y Scrum Master de un proyecto de Data Warehouse cuya duración se extendió a poco más de un año en su totalidad, representó un reto profesional que exigió el desarrollo constante de técnicas y habilidades aplicadas a diversos rubros técnicos, como no técnicos. A lo largo del desarrollo del proyecto, también implicó que las actividades como administrador de proyectos van más allá de la planeación y desarrollo del mismo, y que involucrarán encontrar el equilibrio con los factores internos, externos, así como tratar con particularidades de los proyectos de tecnologías de información

La existencia de múltiples plataformas para la generación estratégica de información es amplia y diversa, no hace falta saber que los gigantes de la industria como lo son Oracle, IBM y Microsoft por comentar algunos, han desarrollado mecanismos robustos para satisfacer las necesidades de un creciente y más demandante mercado, sin embargo, también han surgido organizaciones capaces de ofrecer soluciones más rentables con las necesidades de las organizaciones.

Eso implica que las empresas y/o consultorías que ofrecen soluciones con productos innovadores y adaptables a las necesidades, seamos capaces de ofrecer modelos para el desarrollo exitoso de los proyectos y otorgar confianza a las organizaciones.

Con el apoyo de la metodología Scrum logré adoptar junto a mis compañeros una filosofía de trabajo de autocompromiso y empoderamiento como profesionales, si bien no apliqué los principios, aspectos y procesos al pie de la letra, es porque la metodología misma permite adaptarla a múltiples escenarios, siempre bajo la directriz de cumplir los objetivos internos y externos.

Es posible que dicha adaptación no vuelva a repetirse en proyectos futuros, debido a que quizás no cuente con los mismos recursos, ni con el mismo equipo de trabajo, sin embargo este tipo de proyectos me permitió comprender que el éxito obtenido en la realización, es resultado del esfuerzo en conjunto tanto de la organización SEDCOD como de mis compañeros de equipo y es en este punto donde la sinergia de todos los involucrados debe encaminarse hacia el beneficio colectivo de ambas organizaciones como del personal que los integra.

Durante el desarrollo de este proyecto cometí diversos errores tanto técnicos como administrativos. Sin embargo, la retroalimentación constante fue la herramienta que me permitió darme cuenta de que tenía que realizar ajustes, de lo contrario el impacto de mis acciones provocaría incertidumbre en lo que fue desarrollo y muy probablemente en proyectos futuros donde tuviera la oportunidad de participar.

Actualmente el desarrollo de estos proyectos relacionados a sistemas que contienen información crítica y sensible para las organizaciones, demanda de una madurez y responsabilidad que los equipos que los desarrollarán deben tener. Este proceso no se logró de la noche a la mañana, fue a través de las experiencias en determinadas situaciones que se fue logrando, también contar con un ambiente que propicie obtener el potencial de cada profesionista.

Finalmente puedo comentar que como Scrum Master quebranté ciertas reglas de la sabiduría convencional, no trate a todos los miembros de mi equipo por igual, asigné las tareas más complejas a mis compañeros más talentosos, lo cual me causo ciertos problemas, pero al final de cada sprint sabíamos que el equipo podía responder al ritmo y complejidad del trabajo. Siempre busqué tener un complemento, sabía que en determinadas situaciones yo tenía la última palabra, pero no toda la solución, así que me apoyé en un integrante de mi equipo para evaluar la mejor solución.

Es relevante mencionar que la Facultad de Ingeniería me ayudó a tener los conceptos básicos para mi carrera profesional, en algunas asignaturas con buena aportación.

Bibliografía

- Anderson, K. (2019). SQL Vs. NoSQL, Top Databases. Retrieved March 25, 2019, from <http://highscalability.com/blog/2019/3/6/2019-database-trends-sql-vs-nosql-top-databases-single-vs-mu.html>
- Bernabeu, R. D. (2010). *Data Warehousing: Investigación y Sistematización de Conceptos* (Segunda Ed). Argentina.
- Boal, P. (2016). The Big Data Juggling Act. Retrieved March 10, 2018, from <https://tdwi.org/articles/chapters/st-louis/march-4-2016.aspx>
- Carcaño, S. (2014). Notas clase de Administración de Proyectos de Software. México: FI UNAM.
- Carretero Pérez, J., García Carballeira, F., De Miguel Anasagasti, P., & Pérez Costoya, F. (2001). *Sistemas Operativos Una Visión Aplicada*. (C. F. Madrid, Ed.) (Primera Ed). Madrid, España: McGrawHill.
- Elmasri, R., & Navathe, S. (2003). *Fundamentos de Sistemas de Bases de Datos*. Pearson Prentice Hall.
- Facultad de Ingeniería. (2019). Mapa Curricular 2010. Retrieved February 20, 2018, from http://www.ingenieria.unam.mx/programas_academicos/licenciatura/computacion_plan2010.php
- Figueroa-Díaz, R. (2007). *Metodologías Tradicionales vs. Metodologías Ágiles*. Loja. Retrieved from https://www.researchgate.net/publication/299506242_METODOLOGIAS_TRADICIONALES_VS_METODOLOGIAS_AGILES
- Hughes, B., & Cotterrell, M. (2001). *Software Project Management* (Segunda Ed). Londres: McGrawHill.
- Inmon, W. H. (2002). *Building the Data Warehouse*. (Robert Elliot, Ed.) (Third). Robert Ipsen.
- Lee, J. (2019). Agile vs. Acrum: A Detailed Comparison. Retrieved March 12, 2019, from <https://learn.g2crowd.com/agile-vs-scrum>
- Lipschutz, S. (1991). *Teoría de Conjuntos y Temas Afines*. (McGrawHill, Ed.). McGrawHill.
- Mendez, A., Mártire, A., Britos, P., & Gracia-Martínez, R. (2003). *Fundamentos de Data Warehouse*. Buenos Aires.
- Millán, M. (2017). *Fundamentos de Bases de Datos Notas de Referencia*. (P. Editorial, Ed.) (Primera Ed). Colombia: Universidad del Valle.

- Pérez, L. H. (2017). Modelo entidad relación: descripción y aplicaciones. Retrieved March 27, 2019, from <https://www.icemd.com/digital-knowledge/articulos/modelo-entidad-relacion-descripcion-aplicaciones/>
- Pivotal. (2019). Configuring Your Systems and Installing Greenplum. Retrieved March 20, 2019, from https://gpdb.docs.pivotal.io/540/install_guide/prep_os_install_gpdb.html
- Pressman, R. S. (2010). *Ingeniería de Software un Enfoque Práctico*. (P. R. Vázquez, Ed.) (Séptima Ed). México: McGrawHill.
- SBOK Guide. (2016). *A Guide to the SCRUM BODY OF KNOWLEDGE*. (SCRUMstudy, Ed.) (Edición 20). SCRUMstudy.
- Sullivan, M. (2006). *Álgebra y Trigonometría*. (Pearson, Ed.), *Álgebra y Trigonometría* (Septima Ed). Pearson.
- Talend. (2019). Talend Data Integration Installation Guide for Windows. Retrieved March 19, 2019, from https://help.talend.com/reader/oxVioc0CMfvyU9s9iJbw_w/_JsLBxhdK86B0wMK76BkGA
- TDWI. (2019). Research & Resources. Retrieved March 10, 2019, from <https://tdwi.org/articles/2010/11/01/enterprise-business-intelligence-defined.aspx>