



Modelo de inteligencia de negocios aplicado al proceso de Gestión de Incidentes del departamento de Tecnología, Empresa Grupo de Seguros G.

Sustentada por:

Br. Juárez Antonio Reynoso	2013-0170
Br. Kenji Mukai	2014-2744
Br. Abraham Alcides Ruiz Dominguez	2015-1632

Asesor:

Ing. Santo Navarro

Monografía para optar por el título de:

Ingeniero en Sistemas de Computación,
Ingeniero de Software

Distrito Nacional, República Dominicana

Enero-Abril 2019

Índice General

Portada	i
Índice General	ii
Índice de Figuras y Tablas	v
Dedicatorias y Agradecimientos	x
Resumen	xi
Introducción	xii
Capítulo I: Antecedentes y situación actual	1
1.1 La organización y su tecnología	2
1.2 Proceso de Gestión de Incidentes en la actualidad	3
1.2.1 Aplicativo de Mesa de Servicio	6
1.2.2 Estructura de base de datos para gestión de solicitudes	7
1.3 Implementación actual de Inteligencia de Negocios	11
1.4 El problema con la gestión de incidentes en Grupo de Seguros G	12
1.4.1 Causas de la falla en el manejo de información	13
1.5 Limitaciones de la organización con respecto a la Inteligencia de Negocios	13
1.6 Resumen Capitulo I	15
Capítulo II: Plataforma necesaria para implementar la Inteligencia de Negocios en la Gestión de Incidentes de Grupo de Seguros G	16
2.1 Conceptos Generales	17
2.1.1 Bases de Datos Relacionales	17
2.1.2 Proceso de Extracción, Transformación, Carga (ETL).	19
2.1.3 Data Mart	21
2.1.4 Data Warehouse	23
2.1.5 Cubos de Datos	24
2.1.6 Metodología Kimball	26
2.1.7 Biblioteca de infraestructura de Tecnología de la Información	27
2.1.8 Microsoft Azure	29
2.1.9 Power BI	30

2.1.10 DTU's (Data Transmission Unit)	32
2.1.11 Máquina virtual	32
2.1.12 Azure SQL Database	34
2.1.13 Key Vault	35
2.1.14 Paquetes SSIS	36
2.1.15 Trigger	36
2.1.16 Service Desk	37
2.2 Resumen Capitulo II	40

Capítulo III: Implementación del modelo de Inteligencia de Negocios

Propuesto	41
3.1 Arquitectura de la solución de BI a implementar.	42
3.2 CA Unicenter Service Desk	43
3.3 Base de datos de Origen	45
3.4 Máquina Virtual	47
3.5 Integration Runtime	48
3.6 Creación SQL Server (Logical Server)	51
3.7 Almacenamiento de Datos	53
3.7.1 Arquitectura a utilizar para el almacenamiento de datos	54
3.7.2 Creación de Azure Database – STAGING AREA	55
3.7.3 Creación de Azure Database – DATAMART	57
3.8 Bóveda de Claves criptográficas Key Vault	59
3.8.1 Conectores de base de datos en Microsoft Azure – Secrets	60
3.8.1.1 Formatos Conexiones	62
3.9 Azure Data Factory	63
3.9.1 Estructura Data Factory (ETL)	65
3.9.2 Crear Linked Services	66
3.9.3 Crear Datasets	68
3.9.4 Crear pipelines	70
3.9.4.1 Pipeline: Crear Staging Area	71
3.9.4.1.1 Actividades del Pipeline “Crear Staging Area”	71
3.9.4.2 Pipeline: Copiar Data Origen a Staging	72
3.9.4.2.1 Actividades del Pipeline “Copiar Data Origen a Staging”	73

3.9.4.3 Pipeline: Crear Data mart	74
3.9.4.3.1 Actividades del Pipeline “Crear Data mart”	74
3.9.5 Integration Runtime en el Data Factory	77
3.9.6 Calendarizar la ejecución del Data Factory	82
3.10 Visualización de Datos	83
3.10.1 Configuración Power BI	87
3.10.2 Workspace de Microsoft Power BI	88
3.10.3 Compartir Reportes	89
3.10.4 Conceder permiso de lectura a usuarios	92
3.11 Costo de implementación	94
3.12 Resumen Capítulo III	96
4. Conclusiones	97
5. Recomendaciones	98
6. Bibliografía	99
7. Anexos	102

Índice de Figuras y Tablas

Capítulo I: Antecedentes y situación actual

Figura 1. Diagrama de Flujo de Gestión de Incidentes Grupo de Seguros G.	5
Tabla 1. Nombres y descripciones de tablas de gestión de incidentes.	8
Figura 2. Diagrama de entidad relación de las tablas de gestión de incidentes.	10
Figura 3. Representación visual de donde encaja nuestra propuesta en la implementación actual de Inteligencia de Negocios dentro de la organización.	12
Figura 4. Evolución de los sistemas de información. Big Data Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses.	14
Figura 5. Diagrama de Resumen del Capítulo 1	15

Capítulo II: Plataforma necesaria para implementar la Inteligencia de Negocios en la Gestión de Incidentes de Grupo de Seguros G

Figura 6. Representación de las 5 etapas del ciclo de vida en ITIL.	18
Figura 7. Esta vista previa de Power BI muestra las capacidades de informes y el panel de control que ofrece Power BI.	20
Figura 8. Ejemplo como esta estructurado un Data Warehouse con los Data Marts.	24
Figura 9. Diseño e implementación de un cubo de dato.	26
Figura 10. Representación de las 5 etapas del ciclo de vida en ITIL.	28
Figura 11. Conjunto de plataformas las cuales están conformadas en Azure. Bastante un poco de lo que Azure hace no es selecto para Azure. Amazon, Microsoft y Google están compitiendo. Amazon Web Services, por ejemplo, es el pionero en el campo, frente a las contribuciones de Microsoft y Google.	30
Figura 12. Esta vista previa de Power BI muestra las capacidades de informes y el panel de control que ofrece Power BI.	32
Figura 13. Esquema de los diferentes servicios de SQL Server que existen para Azure.	34
Figura 14. Esquema de los diferentes servicios de SQL Server que existen para Azure.	35

Figura 15. Ejemplo de como crear el Key Vault ID y usarlo como parámetro.	36
Figura 16. Ejemplo de flujo de proceso de un Service Desk	39
Figura 17. Diagrama de resumen capítulo 2.	41
Capítulo III: Implementación del modelo de Inteligencia de Negocios propuesto	
Figura 18. Esquema de arquitectura de BI.	43
Figura 19. Página Principal de la aplicación CA Unicenter Service Desk.	44
Figura 20. Creación de Solicitudes en la aplicación CA Unicenter Service Desk.	44
Figure 21. Resumen de la solicitud una vez cerrada.	45
Figura 22. En el proceso de creación se deben de llenar los campos “Virtual machine name”, “User name”, “Contraseña” y seleccionar el tipo de disco y el sistema operativo.	47
Figura 23. Resumen luego de la creación de la creación de la máquina virtual.	48
Figura 24. Se debe de descargar del Integration Runtime desde la Máquina Virtual utilizando la url indicada.	49
Figura 25. Una vez descargado el archivo se procede a instalar el Integration Runtime en la Máquina Virtual.	50
Figura 26. Así se muestra el Integration Runtime una vez instalada en la Máquina Virtual.	50
Figura 27. Esquema Azure SQL Server.	51
Figura 28. En el proceso de creación se deben de llenar los campos “Server name”, “Server admin login”, “Password”, “Confirm Password”, “Subscription”, “Resource Group” y “Location”.	52
Figura 29. Resumen luego de la creación de la creación del servidor SQL.	53
Figura 30. Esquema de bases de datos, en esta se muestran las bases de datos necesarias para la implementación de esta solución.	54
Figura 31. En el proceso de creación se deben de llenar los campos “Subscription”, “Pricing Tier”, “Database name”, “Collation”, “Server admin login” y “Password”. En estos últimos campos se debe de ingresar las mismas credenciales que se ingresaron durante la creación del servidor SQL.	56
Figura 32. Luego de creada la base de datos se procede configurar la misma a 200 DTU’s. Esto aumentará la capacidad de procesamiento de la base de datos.	56

Figura 33. En el proceso de creación se deben de llenar los campos “Subscription”, “Pricing Tier”, “Database name”, “Collation”, “Server admin login” y “Password”. En estos últimos campos se debe de ingresar las mismas credenciales que se ingresaron durante la creación del servidor SQL.	58
Figura 34. Luego de creada la base de datos se procede configurar la misma a 200 DTU’s. Esto aumentará la capacidad de procesamiento de la base de datos.	58
Figura 35. Esquema de Azure Key Vault. Fuente: Kerkhove, T. (2015). Securing Sensitive Data with Azure Key Vault.	59
Figura 36. Resumen luego de la creación del Key Vault.	60
Figura 37. Se debe de llenar los campos “Name”, “Value” y “Content type”, con el objetivo de crear un secret. En el campo “value” se debe de ingresar un string de conexión con un formato en específico, el mismo se muestra más adelante.	61
Figura 38. Resumen luego de la creación de la creación del secret.	61
Figura 39. Se debe de llenar los campos “Name”, “Subscription”, “Resource Group”, “Versión” y “Location”, con el objetivo de crear un Data Factory en la plataforma de Microsoft Azure.	64
Figura 40. Se muestra el resumen del Data Factory creado.	65
Figura 41. Modelo que presenta la estructura del Data Factory, en esta se muestran todos los componentes necesarios para poder ejecutar un Data Factory.	65
Figura 42. Esquema de actividades del pipeline “Crear Staging Area”, este modelo muestra las actividades y el tipo de actividades que ejecutará el pipeline “Crear Staging Area”.	71
Figura 43. Esquema de actividades del pipeline “Copiar Data Origen a Staging”, este modelo muestra las actividades y el tipo de actividades que ejecutará el pipeline “Copiar Data Origen a Staging”.	73
Figura 44. Esquema de actividades del pipeline “Crear DataMart”, este modelo muestra las actividades y el tipo de actividades que ejecutará el pipeline “Crear Datamart”.	74
Figura 45. En este esquema se muestra la arquitectura del Integration Runtime y cómo se relaciona con la infraestructura local de cualquier organización y con los demás componentes de Microsoft Azure. 1 - El usuario crea un tiempo de ejecución de integración auto hospedado dentro de una fábrica de datos de Azure. 2 - El usuario crea un servicio vinculado para un almacén de datos local. 3 - El nodo de tiempo de ejecución de integración auto hospedado encripta las credenciales. 4 - El	

servicio Data Factory se comunica con el tiempo para la programación y administración de trabajos a través de un canal de control que utiliza una cola compartida de Azure Service Bus. 5 - El tiempo de ejecución de integración auto hospedado copia los datos de un almacén local a un almacenamiento en la nube, o viceversa, según cómo se configure la actividad de copia en la canalización de datos.	78
Figura 46. Se inicia el proceso de creación del Integration Runtime y se selecciona la opción “Perform data movement and dispatch activities to external computes” que se traduce a: Realizar movimientos de datos y despachar actividades a cómputos externos.	79
Figura 47. Se debe de seleccionar el tipo Self-Hosted, esto permitirá ejecutar actividades de copia entre la base de datos de origen y las bases de datos creadas en la plataforma de Azure.	79
Figura 48. Se configura el Integration Runtime y se completan los campos: Nombre y descripción.	80
Figura 49. Se debe de copiar el Key1 de manera que se pueda utilizar en el Integration Runtime instalado en la máquina virtual.	80
Figura 50. En este paso se debe de volver al Integration Runtime de la máquina virtual y se ingresa el Key1 copiado desde el data factory, con el objetivo de relacionar el data Factory con la máquina virtual, esto permitirá el movimiento de los registros almacenados en las bases de datos.	81
Figura 51. En este paso se procede a finalizar el el proceso de configuración del Integration Runtime.	81
Figura 52. Se debe de validar que el Integration Runtime se haya inicializado correctamente.	82
Figura 53. Se procede a crear y configurar el trigger que iniciara la ejecución del datafactory, se completan los campos: nombre del trigger, descripción, fecha de inicio y la recurrencia.	83
Figura 54. Reporte muestra el total de reportes por estado, aplicaciones y antigüedad.	84
Figura 55. Reporte muestra las solicitudes sin asignar y las bloqueadas.	84
Figura 56. Reporte muestra las solicitudes cerradas por día, semana, mes y año.	85
Figura 57. Reporte muestra las solicitudes en estado Test User.	85

Figura 58. Reporte muestra estadística sobre la efectividad del proceso.	86
Figura 59. Reporte muestra todas las solicitudes abiertas.	86
Figura 60. Se debe de configurar los reportes en Power BI, en este pase se debe de ingresar el servidor y base de datos (Datamart) creada en Azure.	87
Figura 61. Se ingresan las mismas credenciales que utiliza el servidor lógico SQL, con el objetivo de que los reportes puedan conectarse a la base de datos “Datamart”.	88
Figura 62. Se procede a colocar el nombre del WorkSpace, se coloca como el Workspace como privado, agregando el grupo que tendrá acceso (Grupo creado para el personal de Operaciones) para Editar el mismo.	89
Figura 63. Se debe de subir los reportes creados desde la máquina virtual al WorkSpace creado, en este paso se selecciona la opción “Local File” y se procede a seleccionar los reportes deseados.	90
Figura 64. Se debe validar que los reportes fueron cargados correctamente en el WorkSpace creado.	90
Figura 65. Se debe configurar el dataset que utilizan los reportes con las credenciales mismas credenciales que se indicaron al crear el servidor lógico SQL	91
Figura 66. Luego de cargado y configurado, se valida que el reporte puede ser actualizado.	91
Figura 67. Luego de seleccionar el dashboard deseado, se debe de hacer click en el botón “Add User”.	92
Figura 68. Se debe de ingresar los correos electrónicos de los usuarios a quienes se les van a compartir el dashboard.	93
Figura 69. Se procede a ingresar algún comentario que podrá ser visualizado por los usuarios indicados en el correo de notificación.	93
Tabla 2. Costo de implementación.	94
Figura 69. Esquema de arquitectura de BI.	96

Dedicatorias y Agradecimientos

Este trabajo se lo dedico a mi familia, la cual siempre ha estado dispuesta a ayudarme y darme soporte con todo lo que me he propuesto. Agradezco especialmente a mis padres, quienes son los principales responsables de que haya llegado tan lejos en mi vida; y por último a todos mis profesores y compañeros quienes me han brindado su apoyo a lo largo de mi carrera universitaria.

Muchas gracias.

- Abraham Alcides Ruiz Dominguez

Dedico este trabajo principalmente a Dios, por haberme permitido llegar a este momento de suma importancia para mi formación personal y profesional. A mi madre Altagracia Salcedo, por ser la que me dio la vida, por ser el pilar más importante y por su apoyo incondicional sin importar cualquier circunstancia. A mi padre Satoshi Mukai, quien me ha dado la mejor educación de mi vida, el que me ha enseñado el valor de la disciplina y responsabilidad, por haberme inculcado los ideales que hoy en día mantengo. A mi pareja Pilar Mireles quien me ha apoyado en todo el proceso educativo de la Universidad y no me permite salir del enfoque hacia el objetivo.

Muchas gracias

- Kenji Mukai Salcedo

Dedico este trabajo a mi familia, en especial a mis padres por haber puesto todo su esfuerzo y dedicación para mostrarme el camino a seguir. Darme su apoyo para nunca dejarme vencer por la dificultades presentadas durante mi carrera universitaria y hacerme ver que puedo llegar a donde me proponga.

A mi hermana, porque ha sabido brindar de su ayuda cuando la he necesitado.

- Juárez Antonio Reynoso

Resumen

Los datos son el recurso con más presencia de toda organización con cierto grado de digitalización en sus procesos. La inteligencia de negocios le da valor potencial a ese recurso en reposo, ayudando el crecimiento de cualquiera de los ámbitos de esta donde es implementada. Cuando hablamos de una organización como Grupo de Seguros G, la falta de implementación de Inteligencia de Negocios en sus procesos es un desperdicio de la plétora de potencial que se encuentra en sus datos. La introducción de la Inteligencia de Negocios a la organización mediante el departamento de tecnología de la información, y sobre uno de los procesos más usados y con más tráfico de datos del departamento, puede servir como una prueba para la alta gerencia del valor que tiene la Inteligencia de Negocios en el futuro de la empresa.

Introducción

El tema del uso de los datos para el análisis y mejoramiento de las organizaciones ha entrado en gran auge en las últimas décadas junto con la evolución de las tecnologías en los sistemas de información, sin embargo, se podría atribuir parte de su aumento exponencial de popularidad al hecho de que la adopción de modelos modernos de gobernanza de TI hacen que sea mucho más fácil proporcionar evidencias del verdadero valor de los recursos de TI dentro de cada organización a la alta gerencia y departamentos no-técnicos. Uno de los recursos más importantes y fáciles de traducir a idioma no-técnico de estos es la data, la base de la Inteligencia de Negocios. La data es siempre presente en todo negocio que cuenta con cierto grado de digitalización. El volumen, la organización, y la relevancia de esta data afecta que tan factible es la implementación de la Inteligencia de Negocios en una organización, sin embargo, en este trabajo presentamos el caso del departamento de Tecnología de la Información de un gigante de los seguros en nuestro país cuya única debilidad aparente es la organización de la data, durante nuestro desarrollo, estaremos descubriendo y explorando las demás debilidades.

Con este trabajo tenemos la intención de descubrir los datos clave a explotar para sacar la mayor cantidad de información posible para soportar la Gestión de Incidentes, al igual que cuáles elementos del departamento de Tecnología de la Información de la organización debemos de observar para asegurar que la producción y manejo de estos datos no resulte en información engañosa que haga más daño que bien en la gestión del departamento. Con esto, queremos diseñar un modelo de Inteligencia de Negocios que ayude a la Gestión de Incidentes de Software en el departamento de Tecnología de la Información del Grupo de Seguros G.

Capítulo I:
Antecedentes y situación actual

1.1 La organización y su tecnología

Grupo de Seguros G es un conglomerado de empresas con más de 50 años de historia que opera a nivel nacional involucrado en una diversa variedad de industrias y áreas de negocios tales como: seguros, seguridad automotriz, administración de fondos de inversiones, venta de repuestos de automóviles y fideicomiso. La organización cuenta con oficinas en Santo Domingo, Baní, Bávaro, Higüey, La Romana, Las Terrenas, Puerto Plata, Punta Cana, San Francisco de Macorís y Santiago. La gran escala geográfica de este grupo organizacional, junto con la diversidad de industrias que toca, y la complejidad de las reglas de negocio de estas industrias, siendo los seguros la principal, coloca una gran carga sobre las áreas de soporte del Grupo G, incluyendo el departamento de Tecnología de la Información.

Este posee numerosos sistemas que manejan las operaciones de cada área de negocios y sus correspondientes estructuras de soporte, un total de 70 aplicaciones son usadas en la organización, tanto de frente al cliente como para uso en el día a día de los empleados. Las 70 aplicaciones se reparten entre las diversas unidades de negocio que componen al Grupo de Seguros G:

- Seguros,
- Administración de Riesgos de Salud,
- Asistencia, la cual incluye servicios de asistencia vehicular, funeral, familiar y del hogar,
- Administración de Fondos de Inversiones,
- Fiduciaria y
- Suplidor de Repuestos Automotrices.

Cada una de estas unidades de negocio cuenta con su aplicación núcleo, o core, y un grupo de aplicaciones de soporte que ayuda a la integración de entre estas, la realización de otras labores del día a día, y la comunicación intraempresarial.

Muchos de los sistemas core de la organización son sistemas de legado, incluyendo su núcleo, el cual, como es de entender, es la parte más difícil de

cambiar de toda organización de gran escala. Las implicaciones del uso de sistemas de legado en una organización de esta magnitud, principalmente, son:

- Brindan retos únicos en cuanto a su actualización,
- El conocimiento colectivo acerca de su tecnología es cada día más escaso,
- La integración de estas aplicaciones con otras toma más esfuerzo, especialmente si se trata de otra aplicación con tecnología más moderna.

Todo esto tiene como consecuencia la llegada de una alta cantidad de incidentes de software al departamento de Tecnología, además del aumento de la dificultad de solución de estos incidentes.

1.2 Proceso de Gestión de Incidentes en la actualidad

Actualmente, el Departamento de TI de Grupo de Seguros G sigue en su mayoría una metodología tradicional de desarrollo de soluciones de software. Esta metodología se extiende al proceso de Gestión de Incidentes de Software, el cual forma parte del día a día de la organización y de numerosas áreas.

Desde enero del 2018 la organización ha estado adaptando sus procesos al modelo ITIL, y a partir de enero de 2019 se han puesto en marcha los procesos de revisión y gestión de la integridad de dichos procesos para asegurar que las operaciones del día a día se apeguen a lo establecido en las nuevas normas del departamento. La Gestión de Incidentes de Software es uno de los procesos que ha pasado la transformación a dicho modelo, sin embargo, el diagrama de estructura de este proceso se desarrolló de forma genérica sobre el mismo diagrama general del proceso establecido como ejemplo en la documentación de ITIL, de forma que fuera fácil traducir el flujo actual del proceso. El flujo establecido dentro de las herramientas de seguimiento es anticuado y obsoleto, en resumen, sigue los siguientes pasos:

- Toma de control, Mantenimiento de Aplicaciones.
 - Responsable: Gerente del área.
 - Descripción: define la llegada de la solicitud a dicha área.
- Desarrollo.

- Responsable: Líder de Proyectos (Desarrollador).
- Descripción: define el proceso de generar la solución de software, inicia cuando la tarea es asignada a su responsable.
- Toma de control, Aseguramiento de Calidad de Software.
 - Responsable: Analista de Calidad.
 - Descripción: define la llegada de la solución a las manos del área.
- Aceptación de usuario:
 - Responsable: Analista de Calidad.
 - Descripción: define las pruebas y evaluación de aceptación del usuario al cambio.
- Toma de control, Ingeniería.
 - Responsable: Gerente del área de Soporte TI.
 - Descripción: define la llegada de la solicitud a dicha área.
- Pase a producción.
 - Responsable: Administrador de Base de Datos o Analista de Centro de Datos, apropiadamente.
 - Descripción: puesta en producción de los nuevos o modificados objetos de base de datos, programas, objetos de aplicaciones, reportes u otros posibles objetos.

Este flujo fue diseñado tomando en cuenta procesos que ya han sido optimizados y es demasiado genérico, este representa diversos riesgos de pérdida de información en el proceso de las solicitudes de software, y ralentiza el trabajo de áreas de TI.

El proceso de Gestión de Incidentes actual es mucho más complejo que lo que se da a conocer en el flujo anteriormente mencionado. Una descripción más fiel del proceso de Gestión de Incidentes de la organización sería el siguiente diagrama,

desarrollado por su área de Calidad de Procesos recientemente

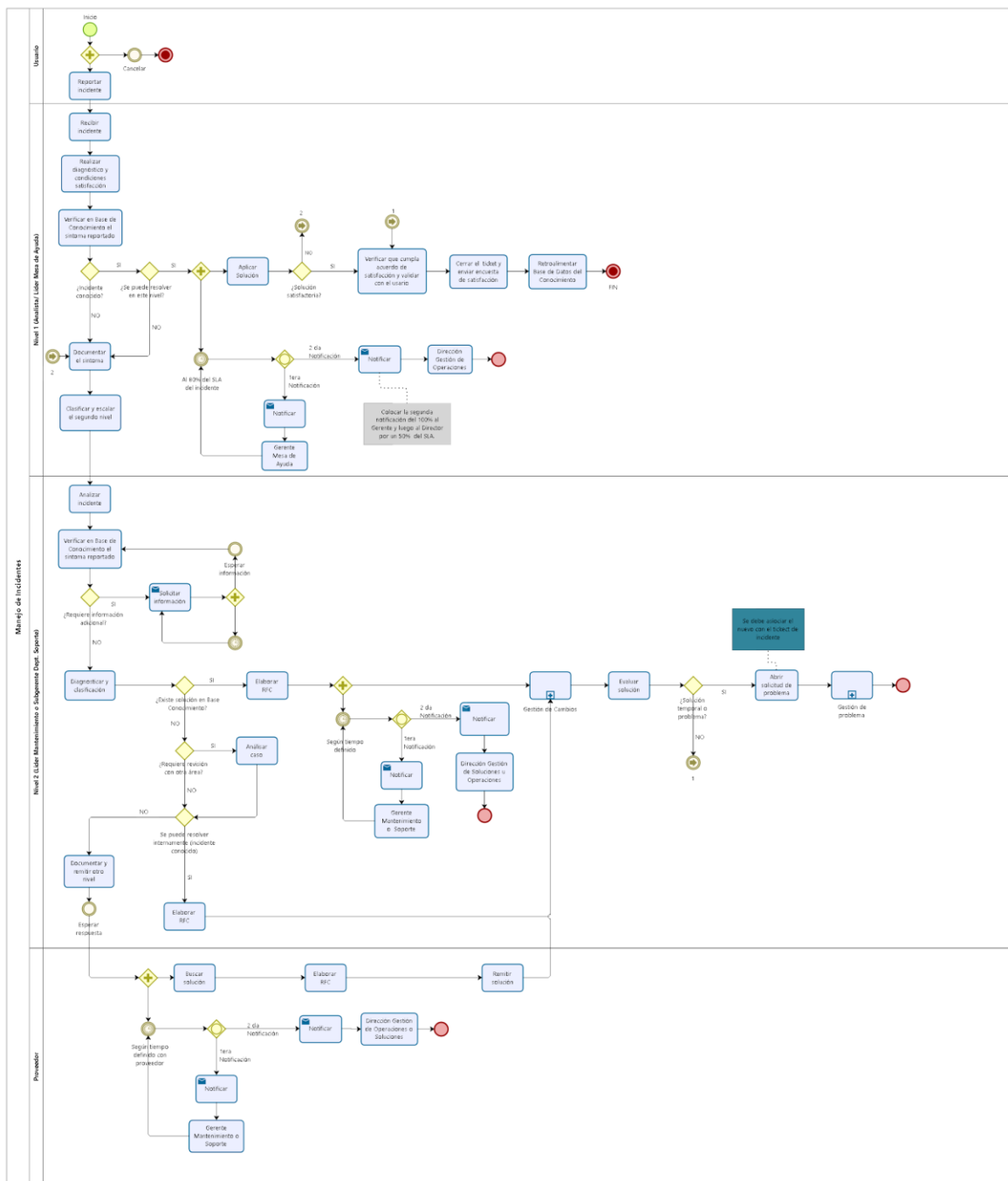


Figura 1. Diagrama de Flujo de Gestión de Incidentes Grupo de Seguros G. Fuente: Departamento de Calidad de Procesos.

Un incidente de software recorre diversas áreas del Departamento de TI (para entendimiento general del área, ver anexo A). Las áreas que recorre un incidente son:

- Mesa de ayuda.

- Mantenimiento de Aplicaciones.
- Aseguramiento de la Calidad de Software.
- Soporte TI.

1.2.1 Aplicativo de Mesa de Servicio

Dentro de las Operaciones de TI, los incidentes de software son tratados como solicitudes a Mesa de Servicio. Las solicitudes son manejadas a través de un sistema llamado “CA Unicenter Service Desk” desarrollado por la empresa CA Technologies basada en Nueva York (ver pantalla de inicio en Anexo B: Inicio Unicenter Service Desk). Esta aplicación es considerada una de los “Sistemas Core” de la organización, esto porque cada uno de los empleados del Grupo G tiene la capacidad de utilizarla, y es la plataforma de la organización para el reporte, asignación, y seguimiento a solicitudes de todo tipo.

De las solicitudes, se almacenan tales informaciones como:

- Área de servicio,
- Responsable,
- Usuario afectado,
- Usuario que reporta,
- Resumen,
- Estado.

Dentro de la aplicación, los cambios en los sistemas de software son manejados con elementos llamados “Órdenes de Cambio”. Las Órdenes de Cambio contienen informaciones parecidas a las solicitudes, pero estas tienen un control del flujo de la solicitud a través de los distintos departamentos de tecnología, la razón de esto es porque no todas las solicitudes de tecnología pasan por más de un departamento, y en su momento, se pensó como apropiada esta forma de separar las solicitudes de software de las demás solicitudes que llegan a TI.

A pesar de que el sistema CA Unicenter Service Desk tiene versiones más recientes, e incluso ha cambiado de nombre a CA Service Desk Manager, el Grupo G tiene en operación una versión del año 2005. Aunque esto hace que la

organización se pierda de las nuevas funcionalidades que pueda tener el software, la usabilidad de este se mantiene fiel a los procesos que el departamento tiene en uso. El problema principal en los procesos tiene raíz en la falta de actualización de los flujos de los procesos, aunque estos siguen siendo útiles en el día a día, no son una representación fiel del trabajo que se hace, esta falta es independiente de la versión del software.

1.2.2 Estructura de base de datos para gestión de solicitudes

A nivel de base de datos, esta aplicación utiliza 7 tablas principales para manejar las solicitudes y solicitudes de cambios (ver detalle de estas con sus campos en Anexo C: Tablas de manejo de solicitudes). Estas tablas son:

Nombre	Descripción
CASOLI	<p>Tabla de solicitudes.</p> <p>Tabla principal del sistema, esta almacena las solicitudes registradas en la aplicación y sus informaciones básicas.</p>
CACAMB	<p>Tabla de ordenes de cambio.</p> <p>Las órdenes de cambio, denominados en el sistema como la palabra en inglés “change order” son utilizados para manejar el flujo de las solicitudes que envuelven cambios en los sistemas informáticos de la organización.</p>
CACONT	<p>Tabla de contactos.</p> <p>En esta tabla se almacenan todos los individuos, grupos, áreas, y departamentos. Se utiliza para registrar todos los recursos humanos del proceso de gestión de solicitudes.</p>
CAPRIO	<p>Tabla de prioridades.</p>
CAESCA	<p>Tabla de estados.</p>
CACATE	<p>Tabla de categorías.</p> <p>Las categorías se utilizan para indicar a qué área corresponde cada una de las solicitudes.</p>
CACARA	<p>Tabla de causa raíz.</p> <p>La causa raíz de una solicitud se utiliza para indicar el sistema de software al que pertenece, si pertenece a alguno. Cuando se trata de incidentes, este es un dato de suma importancia.</p>

Tabla 1. Nombres y descripciones de tablas de gestión de incidentes. Fuente: departamento de Mantenimiento de Aplicaciones.

El departamento de TI no tiene documentación oficial de estas tablas o de cómo se interrelacionan para cumplir con las funciones del sistema, sin embargo, el entendimiento que se tiene de estas dentro del departamento nos indica que su relación es como se muestra en el siguiente diagrama de entidad relación:

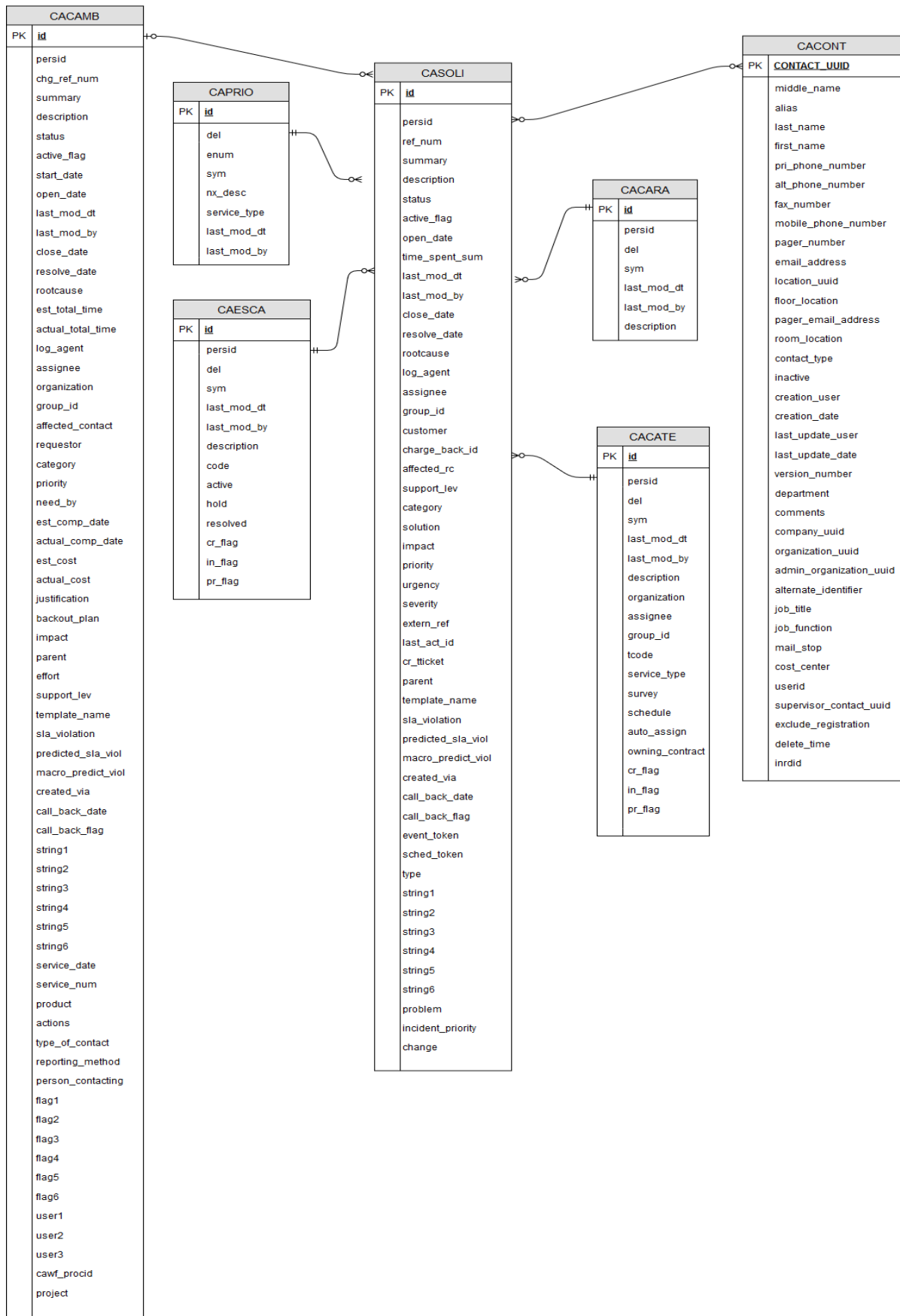


Figura 2. Diagrama de entidad relación de las tablas de gestión de incidentes.

Fuente: Departamento de Mantenimiento de Aplicaciones.

1.3 Implementación actual de Inteligencia de Negocios

Las etapas evolutivas de la inteligencia de negocios y analíticas, o BI&A, por sus siglas en inglés, pueden dividirse en tres de acuerdo a su nivel de complejidad y sus usos: BI&A 1.0, BI&A 2.0 y BI&A 3.0 (Chen et al., 2012). En el caso del Grupo de Seguros G, debido a su industria, el nivel de las informaciones que maneja y las tecnologías que tienen; se puede hablar de una BI&A 1.0. “El BI&A 1.0 se da cuando la data es mayormente estructurada, esta es recolectada a través de varios sistemas de legado y usualmente almacenada en sistemas de bases de datos relacionales” (Chen et al., 2012, p.1066).

De acuerdo a Chen et al. (2012), las características y capacidades principales de la BI&A 1.0 incluyen: “contenido estructurado y basado en sistemas de manejo de bases de datos, almacenamiento de data en bases de datos relacionales, uso de dashboards y scoreboards, y análisis estadístico y minería de datos” (p. 1169). Esto está alineado con las capacidades que hemos observado dentro de la data que tenemos disponible para el desarrollo de nuestro modelo de inteligencia de negocios.

Actualmente, Grupo de Seguros G utiliza la Inteligencia de Negocio de forma muy reducida. Como parte del plan de la organización de adaptarse a los procesos de ITIL y modernizar sus procesos de TI, han implementado procedimientos de colecta de información general para algunas de las áreas de negocio de la organización, de forma que las consultas y reportes de uso diario sean más accesibles y más fácilmente manejables.

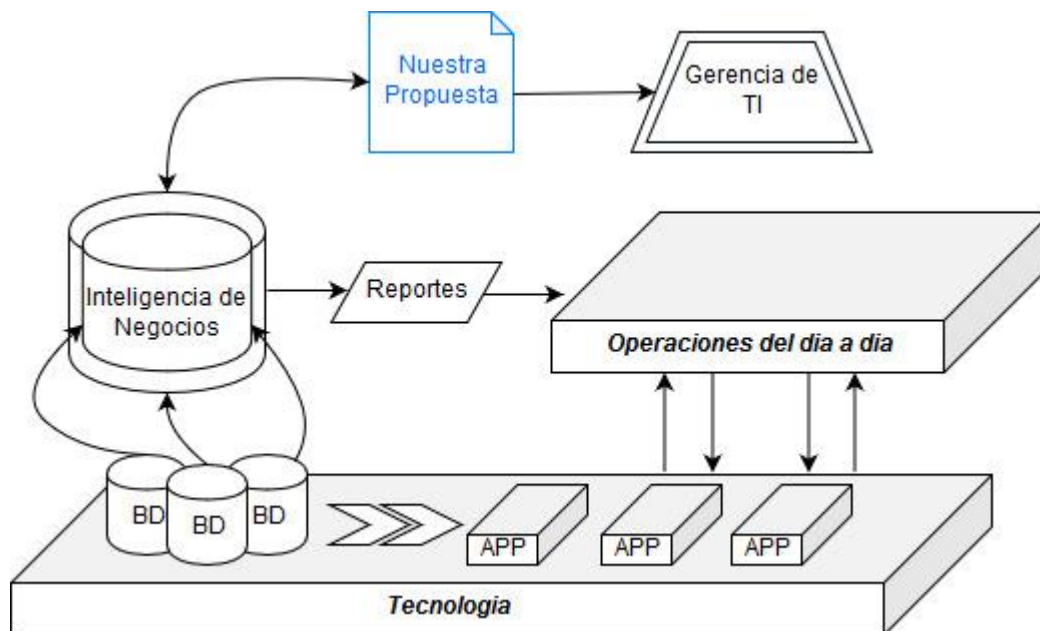


Figura 3. Representación visual de donde encaja nuestra propuesta en la implementación actual de Inteligencia de Negocios dentro de la organización. Fuente: elaboración propia.

1.4 El problema con la gestión de incidentes en Grupo de Seguros G

Hoy en día llegan un promedio de 25 incidentes de software semanalmente al departamento de Tecnología, sin forma de estimar su fecha de entrega fuera de las suposiciones del personal. Una serie de problemas en el proceso de recepción y manejo de información de incidentes hace que los incidentes puedan tardar entre 2 y 5 semanas en ser solucionados, provocando así descontento en los usuarios, ya que no solo llevan un mes esperando, sino que también desconocen qué día se desplegará en producción la solución del incidente reportado.

Uno de los primeros problemas que se tienen en este proceso viene de la dicotomía anteriormente mencionada entre el flujo del proceso que es establecido a través de la aplicación y el flujo de trabajo actual. Esta falla está siendo trabajada por la organización actualmente dentro de su evolución hacia ITIL, sin embargo, no se tienen establecidos procesos de disposición de data histórica y presentación de esta data para identificar las debilidades del proceso, esto fue lo que llevó a que la organización utilizara dicho flujo subóptimo por tantos años sin poder presentar

información o evidencia de que simplemente no estaba funcionando, y, sin lugar a dudas, llevará al uso de otros procesos y tareas subóptimas en un futuro.

1.4.1 Causas de la falla en el manejo de información

Oltra-Bedenes y Roig-Ferriol (2014), comentan que “Uno de los principales cuellos de botella de los departamentos de TI de muchas empresas es el Service Desk.” (p. 218). En el caso del Grupo de Seguros G, este cuello de botella se manifiesta a nivel de la información recibida en la Mesa de Servicio acerca de los incidentes y transmitida hacia los demás departamentos. Este problema en particular se está encarando con la implementación de nuevas normas y procesos sobre dicho departamento, de forma que estos se aseguren de recopilar ciertas informaciones para uso futuro en la gestión de cada incidente. Sin embargo, aún no se utilizan los datos históricos de los incidentes que ya han pasado por el departamento para priorizar y agilizar las gestiones de los nuevos incidentes.

Independientemente de la reducción de este cuello de botella, la causa principal de la falla de manejo de información es que esta no es presentada de forma que pueda ser utilizada para mejoría futura del proceso y la identificación de problemas en las tecnologías usadas en la organización.

1.5 Limitaciones de la organización con respecto a la Inteligencia de Negocios

Minelli, M., et al. (2013), descomponen las tres principales etapas de evolución de los sistemas de información como descrito en el siguiente diagrama:

EVOLUCION DE LOS SISTEMAS DE INFORMACION

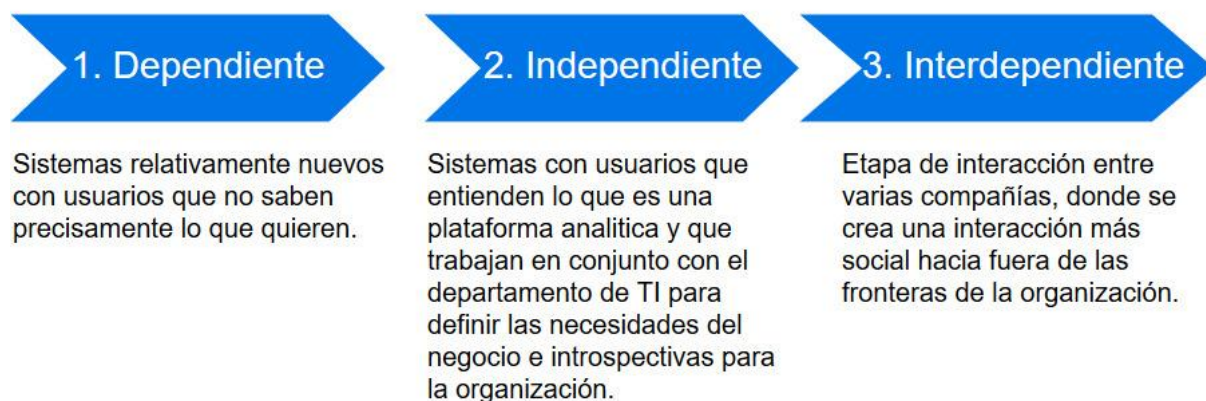


Figura 4. Evolución de los sistemas de información. Fuente: elaboración propia con contenido sacado de artículo Big Data Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses.

Traduciendo esto a la situación actual de la organización, donde el sistema en cuestión es el conjunto de procedimientos y políticas para la gestión de incidentes en Grupo de Seguros G, y los usuarios son todos sus empleados, la principal limitación es como aparentemente la gestión de incidentes está estancada en la etapa número 1 debido a la brecha de información causada por el mal manejo de esta, como la información estadística e histórica es manejada de forma limitada, los usuarios no están en consciencia de estas o sus posibles aplicaciones.

1.6 Resumen Capítulo I

Grupo de Seguros G es un conglomerado de empresas con oficinas en todo el territorio nacional involucrado en una diversa variedad de industrias y áreas de negocios tales como: seguros, seguridad automotriz, administración de fondos de inversiones, venta de repuestos de automóviles y fideicomiso. Para soportar las operaciones de su día a día, la organización cuenta con 70 sistemas y aplicaciones. El gran alcance de áreas de negocio de la organización, combinado con la variedad de las tecnologías utilizadas en sus sistemas, hace que lleguen un promedio de 25 incidentes de software semanalmente.

La organización tiene una débil implementación de inteligencia de negocios, concentrada en sacar informaciones en forma de reportes comúnmente usados por usuarios en producción, y no a la toma de decisiones de gestión sobre procesos internos de la organización.

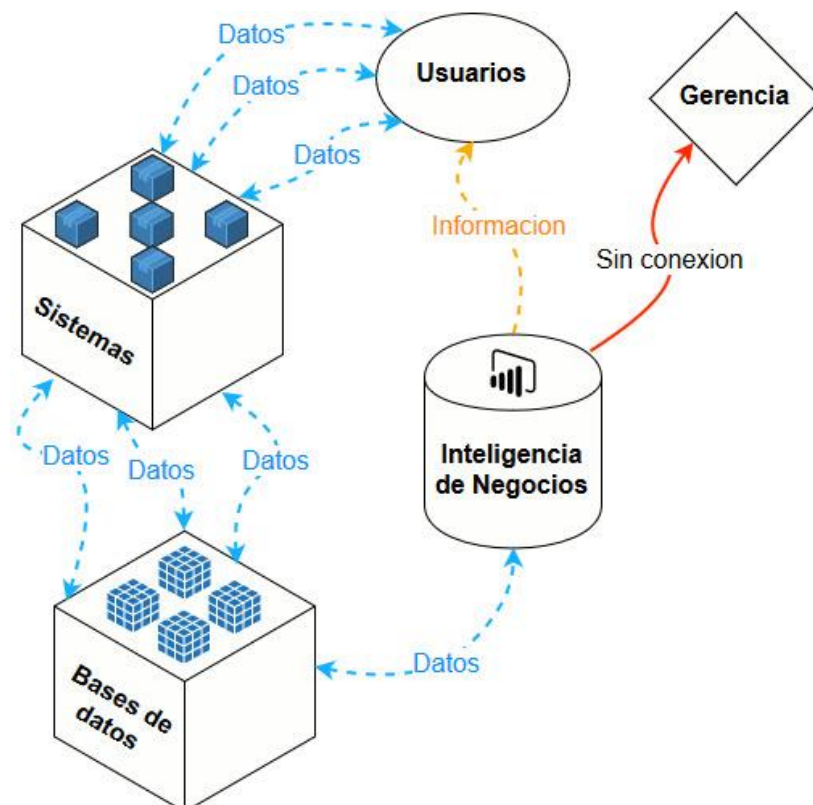


Figura 5. Diagrama de Resumen del Capítulo 1. Fuente: elaboración propia.

Capítulo II:
Plataforma necesaria para implementar la
Inteligencia de Negocios en la Gestión de Incidentes
de Grupo de Seguros G

2.1 Conceptos Generales

2.1.1 Base de Datos Relacionales

Una base de datos relacional es una gran cantidad de tablas representadas formalmente desde las cuales se puede obtener o reensamblar información desde varias perspectivas sin tener que rediseñar las tablas de la base de datos. El cliente estándar y la interfaz de programación de aplicaciones (API) de una base de datos relacional es el lenguaje de consulta estructurado (SQL). Las articulaciones de SQL se utilizan tanto para consultas intuitivas de datos de una base de datos relacional como para información de asuntos sociales para informes.

La base de datos relacional fue diseñada en 1970 por E. F. Codd, en ese momento un joven ingeniero de software en IBM. En su artículo, "Un modelo relacional de datos para grandes bancos de datos compartidos", Codd propuso pasar de guardar información en estructuras progresivas o de navegación a ordenar información en tablas que contienen líneas y secciones.

Cada tabla, que está aquí y allá llamada conexión, en una base de datos relacional contiene al menos una clase de información en segmentos, también llamada rasgos. Cada línea, adicionalmente llamada registro o tupla, contiene un ejemplo único de información, o clave, para las clases caracterizadas por las secciones. Cada tabla tiene una clave esencial excepcional, que distingue los datos en una tabla. La conexión entre las tablas se podría configurar por medio de la utilización de claves externas, un campo en una tabla que se conecta a la clave esencial de otra tabla.

Por ejemplo, una ejecución de la base de datos de la sección de solicitud de negocios del molino incorporaría una tabla que representaba a un cliente con segmentos para nombre, dirección, número de teléfono, etc. Otra tabla representaría una solicitud: artículo, cliente, fecha, costo de las transacciones, etc. Un cliente de una base de datos relacional podría entonces obtener una perspectiva de la base de datos para cumplir con sus requisitos. Por ejemplo, a un administrador de sucursal le puede gustar una vista o dar una cuenta de todos los clientes que

compraron artículos después de una fecha específica. Un administrador de las administraciones relacionadas con el dinero en una organización similar podría, de tablas similares, adquirir una información detallada sobre las cuentas que deberían pagarse.

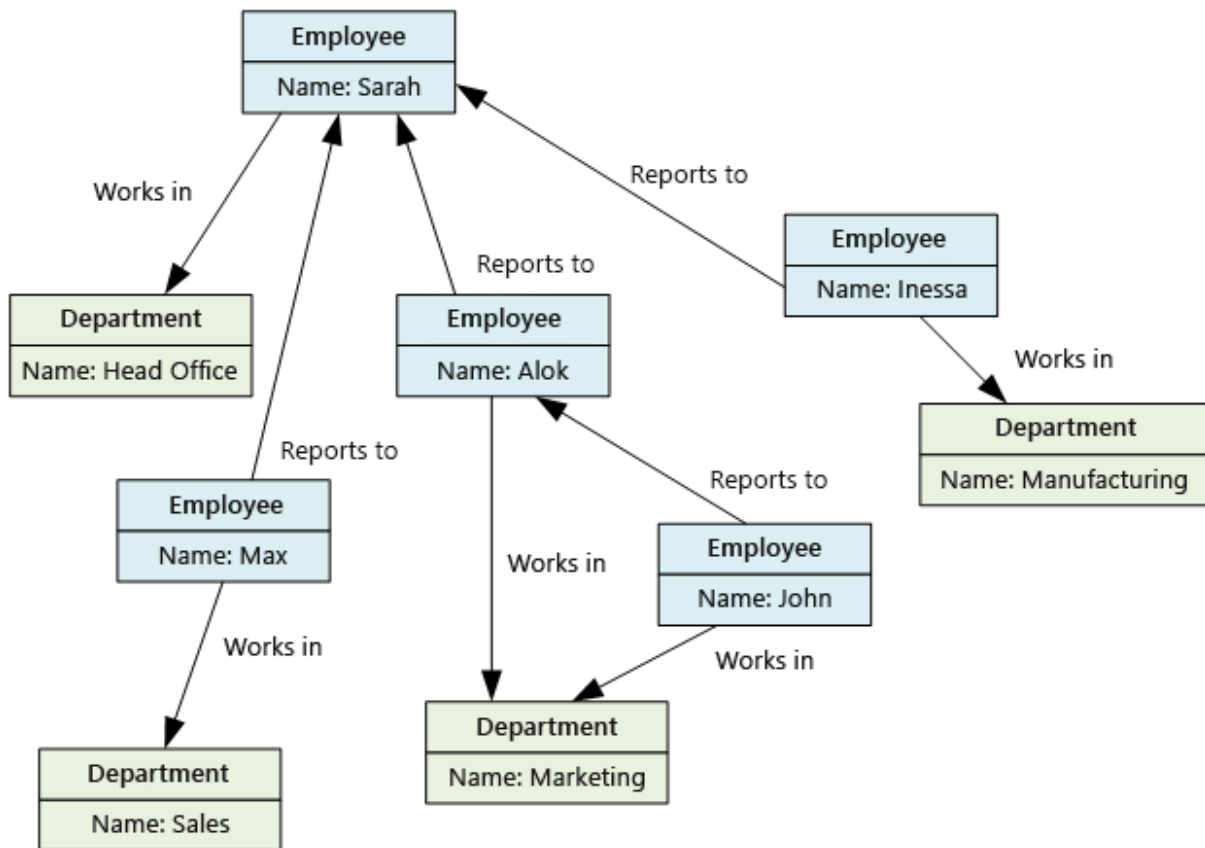


Figura 6. Ejemplo de un esquema de base de datos relacional.

2.1.2 Proceso de Extracción, Transformación, Carga (ETL).

ETL es un tipo de integración de datos que se refiere a los tres pasos (extracción, transformación, carga) utilizados para combinar datos de múltiples fuentes. Se utiliza a menudo para construir un almacén de datos. Durante este proceso, los datos se toman (extraen) de un sistema de origen, se convierten (se transforman) en un formato que se puede analizar y se almacenan (cargan) en un almacén de datos u otro sistema. Extraer, cargar, transformar (ELT) es un enfoque alternativo pero relacionado diseñado para impulsar el procesamiento hacia la base de datos para mejorar el rendimiento.

Las empresas han confiado en el proceso ETL durante muchos años para obtener una visión consolidada de los datos que impulsan mejores decisiones comerciales. Hoy en día, este método de integración de datos de múltiples sistemas y fuentes sigue siendo un componente central de la caja de herramientas de integración de datos de una organización.

Cuando se utiliza con un almacén de datos empresarial (datos en reposo), ETL proporciona un contexto histórico profundo para el negocio.

Al proporcionar una vista consolidada, ETL facilita a los usuarios comerciales analizar e informar sobre datos relevantes para sus iniciativas.

ETL puede mejorar la productividad de los profesionales de datos porque codifica y reutiliza procesos que mueven datos sin requerir habilidades técnicas para escribir código o scripts.

ETL ha evolucionado con el tiempo para admitir los requisitos de integración emergentes para cosas como la transmisión de datos.

Las organizaciones necesitan tanto ETL como ELT para reunir datos, mantener la precisión y proporcionar la auditoría que generalmente se requiere para el almacenamiento de datos, informes y análisis.

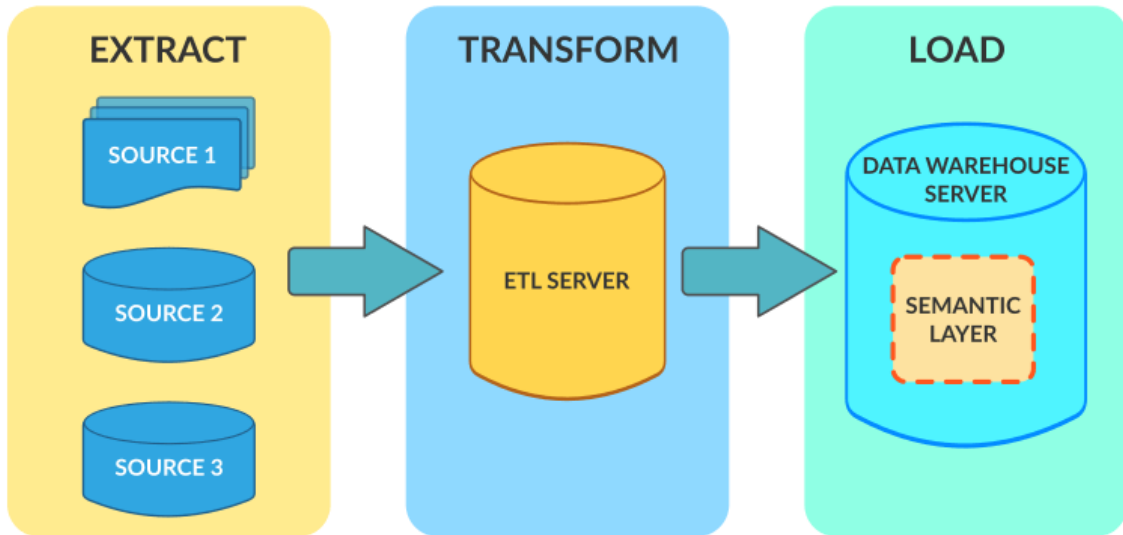


Figura 7. Esquema de funcionamiento de un proceso ETL.

2.1.3 Data Mart

En un mercado regido por una gran cantidad de información y análisis, los data warehouse son una de las claves para cambiar efectivamente los datos en partes de conocimiento. Los centros de distribución de información gestionan regularmente colecciones informativas expansivas, pero la investigación de la información requiere un descubrimiento sencillo y una información de acceso inmediato.

Un data mart es una base de datos situada por el sujeto que con frecuencia es un fragmento parcelado de un centro de distribución de información de la empresa. El subconjunto de información que se guarda en una data mart generalmente se alinea con una unidad especializada específica como ofertas, fondos o publicidad. Los data marts aceleran los formularios comerciales al permitir el acceso a datos importantes en un centro de distribución de información o en un data warehouse operacional en días, en lugar de meses o más. Dado que un data mart solo contiene la información pertinente a un territorio empresarial específico, es un enfoque con conocimientos financieros para aumentar rápidamente notables fragmentos de conocimiento.

Hay tres tipos de data marts: dependientes, autónomos e híbridos. Se organizan en función de su conexión con el centro de distribución de información y las fuentes de información que se utilizan para hacer el marco.

Dependientes

Un data mart necesitado se hace desde un almacén de información de la empresa actual. Es la metodología de arriba hacia abajo que comienza con el almacenamiento de toda la información comercial en un área focal, en ese momento elimina una parte caracterizada inequívocamente de la información cuando se requiere para la investigación.

Para dar forma a un almacén de información, una disposición particular de información se totaliza (enmarcada en un grupo) desde el centro de distribución, se

reconstruye, en ese punto se apila al almacén de información donde se puede cuestionar muy bien. Tiende a ser una vista coherente o un subconjunto físico del centro de distribución de información:

Vista lógica: una tabla virtual / ver que está coherentemente, aunque no físicamente, aislada del almacén de información

Subconjunto físico: extracción de datos que es una base de datos físicamente independiente del centro de distribución de información

La información granular, la dimensión más mínima de la información en el conjunto de objetivos, en el centro de distribución de información se completa como una perspectiva única para cada data mart de datum de sala que se realiza.

Data marts independiente

Un data mart independiente es un marco independiente, hecho sin la utilización de un centro de distribución de información, que se centra en una rama del conocimiento o trabajo empresarial. La información se extrae de fuentes de información internas o externas (o ambas), preparada, en ese punto apilada en el archivo del data mart donde se guarda hasta que se requiere para la investigación comercial.

Los data marts de información gratuitos no son difíciles de planificar y crear. Son útiles para lograr objetivos momentáneos, pero pueden volverse voluminosos para tratar, cada uno con su propio instrumento y razonamiento ETL, a medida que las necesidades comerciales se extiendan y se vuelvan progresivamente complejas.

Data mart híbrido

Un data mart cruzado consolida la información de un almacén de información actual y otros marcos operativos de fuentes. Une la velocidad y el punto focal del cliente final de una metodología de arriba hacia abajo con las ventajas de la conciliación a nivel de empresa de la técnica base.

2.1.4 Data Warehouse

El data warehouse es una innovación que totaliza los datos organizados de al menos una fuente con el objetivo de que puedan analizarse y analizarse para obtener un conocimiento empresarial más importante.

Los data warehouses se utilizan normalmente para relacionar datos comerciales amplios con el fin de brindar un conocimiento oficial más prominente sobre la ejecución corporativa.

Los data warehouses utilizan una estructura alternativa de las bases de datos operacionales estándar. Los últimos se han mejorado para mantener una gran precisión de los datos en el momento actualizando rápidamente los datos en curso. Los almacenes de datos, por otro lado, pretenden dar una perspectiva a largo plazo de los datos después de algún tiempo. Intercambian el volumen de intercambio y más bien tienen experiencia práctica en el total de datos.

Se examinan numerosos tipos de datos comerciales a través de los data warehouses. El requisito de un data warehouse con frecuencia termina cuando los requisitos previos explicativos se cruzan con la ejecución continua de las bases de datos operacionales. La ejecución de una investigación alucinante en una base de datos requiere que la base de datos ingrese en un estado fijo permanente. Esto es regularmente indefendible para las bases de datos basadas en valores. Se utiliza un data warehouse para realizar el trabajo lógico, dejando que la base de datos basada en el valor se concentre en los intercambios.

Las diferentes ventajas de un data warehouse son la capacidad de desglosar los datos de numerosas fuentes y de organizar los patrones de contraste utilizando el procedimiento ETL.

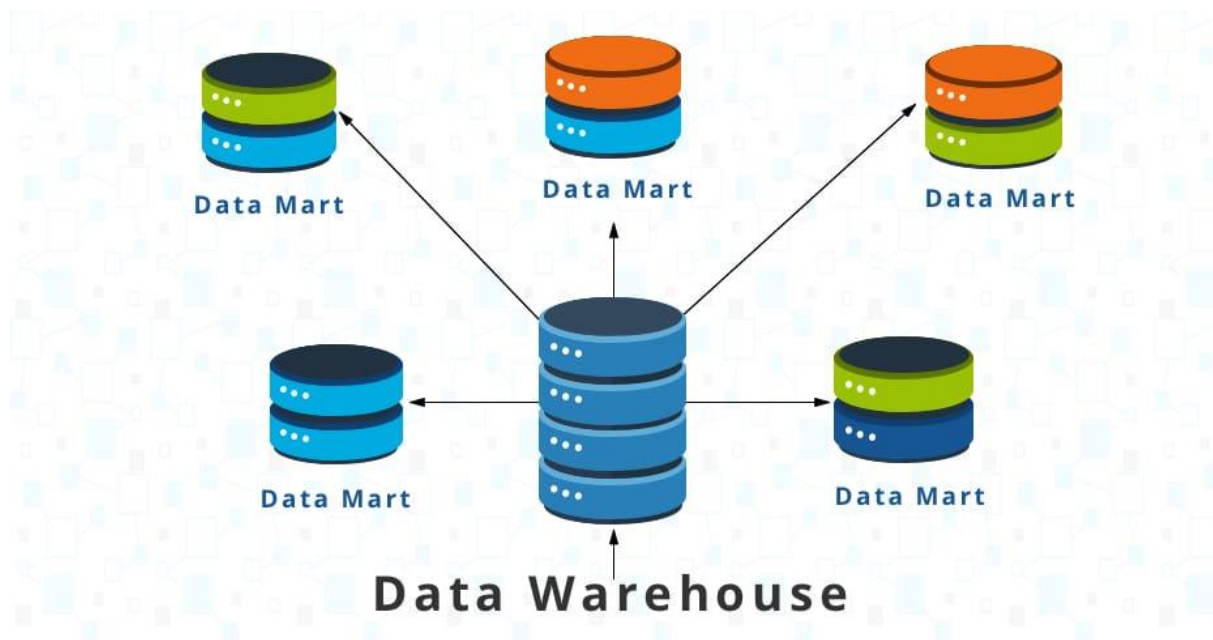


Figura 8. Ejemplo como esta estructurado un Data Warehouse con los Data Marts.

2.1.5 Cubos de Datos

Un cubo de datos alude a un tridimensional (3D) (o superior) de cualidades que se usan comúnmente para aclarar la sucesión de tiempo de los datos de una imagen. Es una reflexión de datos para evaluar los datos acumulados de una variedad de perspectivas. Además, es valioso para la espectroscopia de imagen, ya que una imagen terriblemente asentada se presenta como un volumen 3D.

Un cubo de datos también puede representarse como los aumentos multidimensionales de tablas bidimensionales. Se puede ver muy bien como una acumulación de tablas 2D indistinguibles apiladas unas sobre otras. Los cubos de datos se utilizan para hablar con datos que son demasiado complejos para ser representados de alguna manera por una tabla de segmentos y líneas. Por lo tanto, los cubos de datos pueden ir más allá de 3-D para incorporar muchas más mediciones.

Un cubo de datos se usa comúnmente para traducir datos sin esfuerzo. Es particularmente útil cuando se habla de datos junto con mediciones como proporciones específicas de necesidades comerciales. Cada medida de un cubo habla de cierta normalidad para la base de datos, por ejemplo, cada día, mes a mes o ofertas anuales. Los datos incluidos dentro de un cubo de datos influyen en la

posibilidad de desglosar prácticamente cada una de las cifras para todos los propósitos y propósitos de cualquiera o todos los clientes, ofertas de especialistas, artículos y mucho más. Por lo tanto, un cubo de datos puede configurar patrones e investigar la ejecución.

Los cubos de datos se ordenan principalmente en dos clasificaciones:

Cubo de datos multidimensionales: la mayoría de los elementos OLAP se crean en función de una estructura en la que el cubo está diseñado como un clúster multidimensional. Estos elementos OLAP multidimensionales (MOLAP) generalmente ofrecen una ejecución mejorada cuando se comparan con diferentes metodologías predominantemente a la luz del hecho de que se pueden registrar legítimamente en la estructura del cubo de datos para ensamblar subconjuntos de datos. En el punto en que la cantidad de medidas es más prominente, el cubo termina esparcido. Eso implica que pocas celdas que hablan de mezclas de propiedades específicas no contendrán ningún dato acumulado. Por lo tanto, esto es compatible con las necesidades de capacidad, que pueden alcanzar dimensiones molestas de vez en cuando, haciendo que la disposición MOLAP sea indefendible para conjuntos de datos inmensos con numerosas mediciones. Los sistemas de presión pueden ayudar; Sea como fuere, su utilización puede dañar el ordenamiento común de MOLAP.

OLAP relacional: OLAP relacional hace uso de la base de datos social. El cubo de datos ROLAP se utiliza como un grupo de tablas sociales (aproximadamente el doble de la cantidad de mediciones) en contraste con una exhibición multidimensional. Cada una de estas tablas, conocida como cuboide, connota una vista particular.

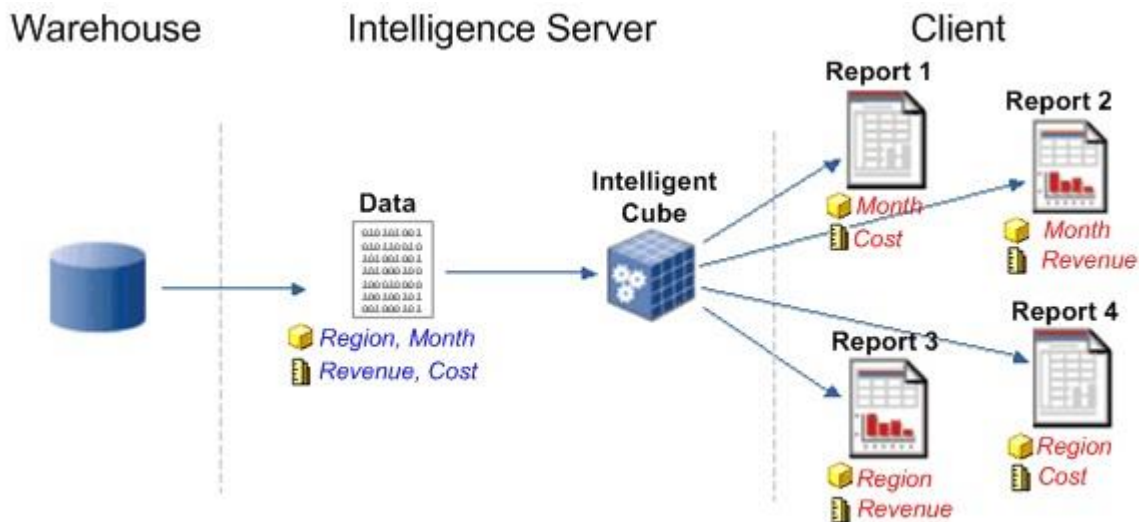


Figura 9. Diseño e implementación de un cubo de dato.

2.1.6 Metodología Kimball

Kimball es un partidario de un enfoque para el diseño del almacén de datos descrito como de abajo hacia arriba, en el cual los mercados de datos dimensionales se crean por primera vez para proporcionar capacidades analíticas y de informes para áreas comerciales específicas como "Ventas" o "Producción". Estos data warehouses se incorporan a largo plazo para hacer un almacén de datos que utiliza un diseño de transporte, que comprende mediciones ajustadas entre cada uno de los data warehouses. Así que el almacén de datos termina siendo dividido en varios bazares de datos inteligentemente independientes y confiables, en lugar de un modelo concentrado mayor y complejo. La confianza empresarial puede devolverse tan rápido como las principales tiendas de datos, y la estrategia se presta a una forma exploratoria e iterativa de tratar con los data warehouses de estructura, por lo que no se requiere una estrategia innovadora.

Kimball es la metodología utilizada con la mayor frecuencia posible, especialmente cuando está utilizando el paquete de Microsoft BI. Es bien sabido que los clientes comerciales pueden ver algunos resultados rápidamente, con el peligro de que pueda hacer copias de los datos o que deba volver a intentar una parte de la estructura porque no hubo una estrategia integral. Con Inmon hay una estrategia innovadora y la mayoría de las veces no tendrá que volver a intentar nada, sin embargo, en el caso de que pase algún tiempo antes de ver alguna

ventaja, el gasto directo es crítico. Además, otro peligro es cuando comienza a producir resultados, los datos de la fuente de negocios han cambiado o hay necesidades cambiadas y es posible que deba volver a intentar un poco de trabajo a cualquier velocidad.

2.1.7 Biblioteca de infraestructura de Tecnología de la Información

La biblioteca de infraestructura de TI (ITIL) es una biblioteca de volúmenes que describe un marco de mejores prácticas para la prestación de servicios de TI. ITIL ha pasado por varias revisiones en su historia y actualmente consta de cinco libros, cada uno de los cuales cubre varios procesos y etapas del ciclo de vida del servicio de TI. El enfoque sistemático de ITIL para la gestión de servicios de TI puede ayudar a las empresas a gestionar el riesgo, fortalecer las relaciones con los clientes, establecer prácticas rentables y crear un entorno de TI estable que permita el crecimiento, la escala y el cambio.

Desarrollado por la Agencia Central de Computadoras y Telecomunicaciones (CCTA) del gobierno británico durante la década de 1980, el ITIL consistió primero en más de 30 libros, desarrollados y publicados a lo largo del tiempo, que codificaron las mejores prácticas en tecnología de la información acumulada de muchas fuentes (incluidas las mejores prácticas de los proveedores).) alrededor del mundo. IBM, por ejemplo, dice que su serie de cuatro volúmenes sobre conceptos de administración de sistemas, Un sistema de administración para sistemas de información, conocidos como los libros amarillos, brindó información vital en los libros originales de ITIL.

ITIL V3 se introdujo en 2011. En esta última iteración, el marco consta de cinco etapas de ciclo de vida. Estas etapas se dividen en múltiples procesos que se basan en principios de servicio, procesos, roles y medidas de rendimiento.

Las cinco etapas del ciclo de vida son:

1. Estrategia de servicio: se centra en definir la estrategia de su organización para atender a los clientes y cómo mantener e implementar esa estrategia. El objetivo de esta etapa del ciclo de vida es hacer que su organización piense y actúe de manera estratégica.
2. Diseño de servicio: se centra en convertir la estrategia de servicio en realidad, diseñando y desarrollando nuevas ofertas de servicio o mejorando las ofertas existentes de su organización.
3. Transición del servicio: se enfoca en reunir todos los activos dentro de un servicio y asegurar que estos estén integrados y probados. También se enfoca en la calidad de un servicio nuevo o modificado antes de que sea operativo.
4. Operación del servicio: se enfoca en asegurar que haya mejores prácticas sólidas que apoyen los servicios de respuesta. Por ejemplo, Incident Management y el servicio de atención al cliente de su organización forman parte de esta etapa.
5. Mejora continua del servicio: se enfoca en mejorar la efectividad y la eficiencia de los procesos y servicios de TI de su organización. Básicamente, esta etapa del ciclo de vida mejora continuamente las otras cuatro etapas.



Figura 10. Representación de las 5 etapas del ciclo de vida en ITIL.

2.1.8 Microsoft Azure

El sitio de Microsoft Azure ofrece un catálogo de varias administraciones distintas que puede utilizar, incluidas máquinas virtuales completas, bases de datos, almacenamiento de registros, refuerzos y administraciones para aplicaciones web y versátiles.

Esta administración fue inicialmente llamada "Windows Azure", sin embargo, progresó a "Microsoft Azure", ya que puede lidiar con mucho más que solo Windows. Puede ejecutar máquinas virtuales de Windows o Linux en Azure, por ejemplo, en lo que quiera.

A través de estas varias administraciones, verá que puede hacer prácticamente cualquier cosa. Además, para cualquier cosa que Azure no ofrezca en una administración simple, puede configurar una máquina virtual de Windows o Linux que tenga cualquier producto que necesite utilizar. Incluso podría alojar un área de trabajo de Windows o Linux en la nube en una máquina virtual e interactuar con ella de forma remota. Es simplemente un enfoque más para utilizar activos de registro remoto.

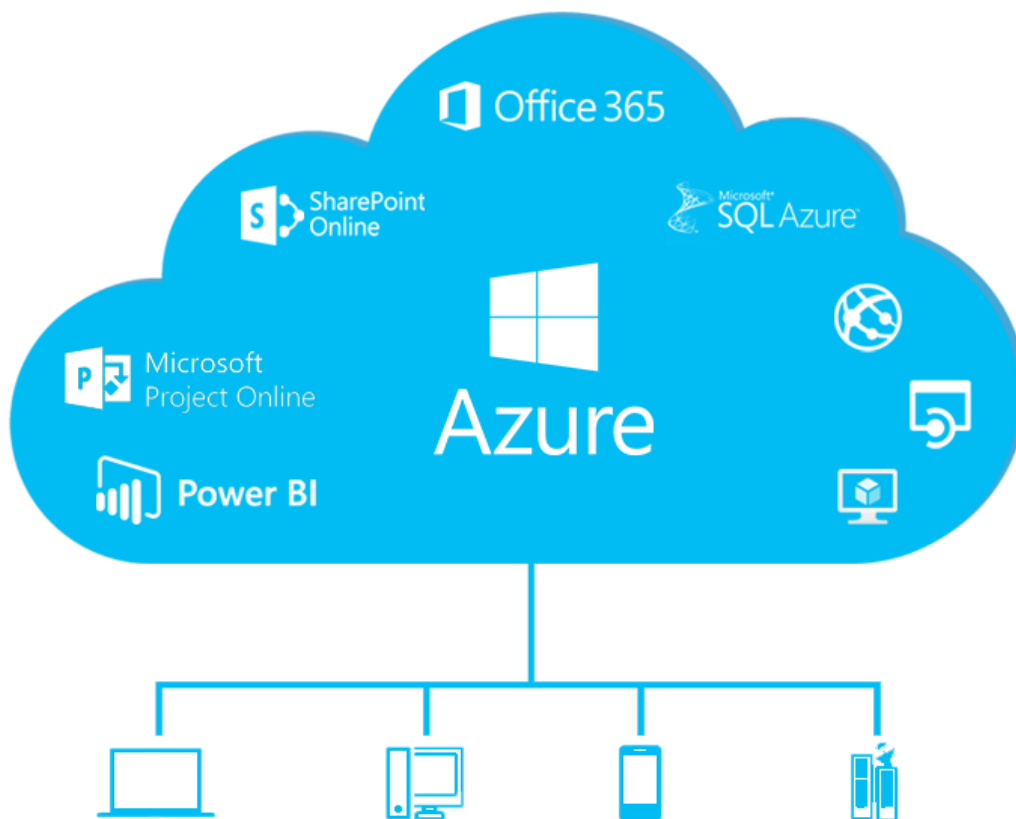


Figura 11. Conjunto de plataformas las cuales están conformadas en Azure. Bastante un poco de lo que Azure hace no es selecto para Azure. Amazon, Microsoft y Google están compitiendo. Amazon Web Services, por ejemplo, es el pionero en el campo, frente a las contribuciones de Microsoft y Google.

2.1.9 Microsoft Power BI

Microsoft Power BI es una etapa de conocimiento de negocios que brinda a los clientes de negocios no técnicos aparatos para conglomerar, diseccionar, visualizar y compartir información. La interfaz de usuario de Power BI es genuinamente instintiva para los clientes familiarizados con Excel y su incorporación profunda con otros elementos de Microsoft lo convierte en un instrumento flexible de autoadministración que requiere una preparación directa y mínima.

Se propone una forma gratuita de Power BI para pequeños y medianos empresarios; Se puede acceder a una versión experta llamada Power BI Plus por un mes a mes. Los clientes pueden descargar una aplicación para Windows 10, llamada Power BI Desktop, y aplicaciones portátiles locales para Windows, Android

y dispositivos iOS. Además, hay un servidor de informes Power BI para organizaciones que deben mantener su información e informes en las instalaciones. Esa forma de Power BI requiere una variante única de la aplicación del área de trabajo, llamada apropiadamente Power BI Desktop para el servidor de informes Power BI.

Microsoft Power BI se utiliza para descubrir partes del conocimiento dentro de la información de una asociación. Power BI puede ayudar a asociar índices de información diferentes, cambiar y limpiar la información en una demostración de información y hacer diagramas o tablas para obtener imágenes de la información. La mayoría de esto se puede impartir a otros clientes de Power BI dentro de la asociación.

Los modelos de información hechos a partir de Power BI se pueden utilizar de diferentes maneras para las asociaciones, incluido el recuento de historias a través de esquemas y representaciones de información y mirando "considerar la posibilidad" de situaciones dentro de la información. Los informes de Power BI también pueden abordar las consultas de manera continua y ayudar a determinar si las oficinas cumplen con las medidas comerciales.

Power BI también puede proporcionar cuadros de mando oficiales a los presidentes o administradores, lo que le da a la administración más información sobre cómo se están llevando las divisiones.

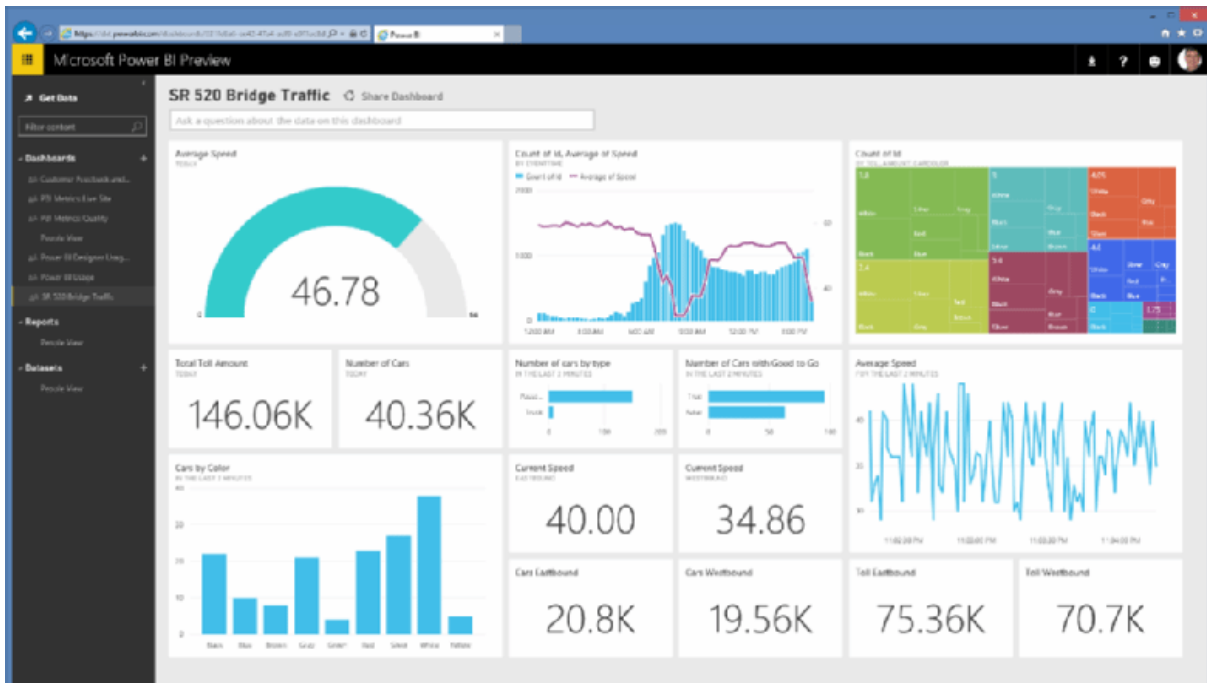


Figura 12. Esta vista previa de Power BI muestra las capacidades de informes y el panel de control que ofrece Power BI.

2.1.10 DTU's (Data Transmission Unit)

DTU es una unidad de medida para la ejecución de un nivel de administración y es un resumen de algunas cualidades de base de datos. Cada nivel de administración tiene un número específico de DTUs relegadas a él como un método simple para pensar en la dimensión de ejecución de un nivel frente a otro.

Las DTU ofrecen un enfoque para representar el límite general de una dimensión de acto de las bases de datos Básica, Estándar y Premium. Las DTU dependen de una proporción mixta de CPU, memoria, peruses y composiciones. A medida que aumentan las DTU, aumenta la potencia ofrecida por los niveles de ejecución. Por ejemplo, un nivel de ejecución con 5 DTU tiene varias veces más poder que un nivel de acción con 1 DTU. Una mayor cantidad de DTU se aplica a cada servidor.

2.1.11 Máquina virtual

Una máquina virtual (VM) es un marco de trabajo (OS) o condición de aplicación que se introduce en la programación, que imita el equipo comprometido.

El cliente final tiene un encuentro similar en una máquina virtual que tendría en un equipo dedicado.

La programación particular, denominada hipervisor, copia la CPU del cliente o servidor de la PC, la memoria, la placa rígida, la disposición y los activos de otros equipos, permitiendo a las máquinas virtuales compartir los activos. El hipervisor puede copiar diferentes etapas de equipos virtuales que se separan entre sí, lo que permite que las máquinas virtuales ejecuten los marcos de trabajo de Linux y Windows Server en el host físico fundamental equivalente. La virtualización limita los costos al disminuir el requisito de estructuras de equipos físicos. Las máquinas virtuales utilizan el equipo con mayor eficacia, lo que reduce la cantidad de equipos y los costos de mantenimiento relacionados, y reduce la demanda de energía y refrigeración. Además, facilitan el uso del tablero porque el equipo virtual no se queda corto. Los gerentes pueden explotar las condiciones virtuales para desenredar los refuerzos, la recuperación de catástrofes, los nuevos arreglos y los mandados de la organización del marco fundamental.

Las máquinas virtuales no requieren un equipo particular, hipervisor explícito. La virtualización sí requiere que se requiera más transmisión de datos, almacenamiento y preparación del límite que un servidor convencional o área de trabajo si el equipo físico va a tener varias máquinas virtuales en ejecución. Las máquinas virtuales pueden, sin mucho esfuerzo, ser duplicadas y reasignadas entre servidores host para mejorar el uso de los activos del equipo. Dado que las máquinas virtuales en un host físico pueden gastar cantidades de activos desiguales, uno puede acumular la capacidad física accesible, mientras que otro almacena casi nada: los expertos en TI deben compensar las máquinas virtuales con activos accesibles.

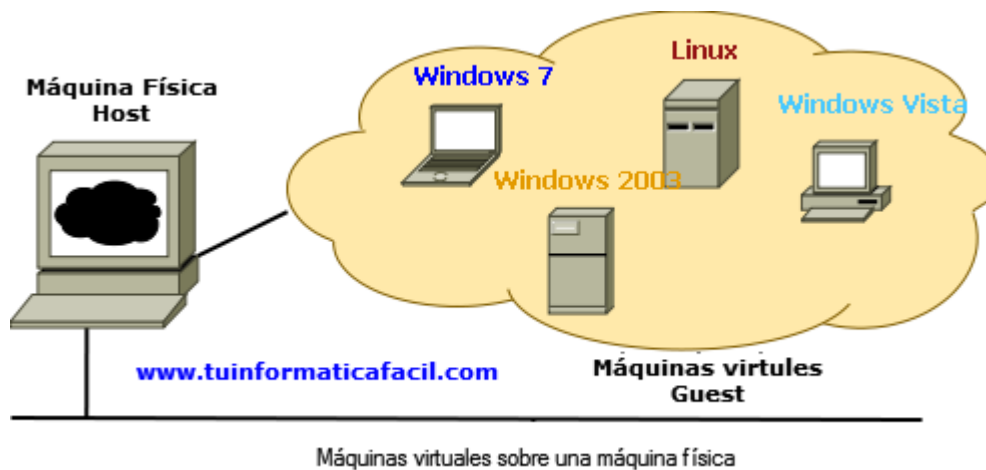


Figura 13. Esquema de máquinas virtuales montadas en una máquina física.

2.1.12 Azure SQL Database

SQL Azure es la administración de la base de datos en la nube de Microsoft. En vista de la innovación de la base de datos de SQL Server y basada en la etapa de computación distribuida de Windows Azure de Microsoft, SQL Azure permite a las asociaciones almacenar información social en la nube y escalar rápidamente el alcance de sus bases de datos a medida que cambian las necesidades empresariales. La información se facilita, supervisa y aprovisiona en las granjas de servidores de Microsoft.

Las asociaciones pueden crear aplicaciones locales y moverlas a SQL Azure o fabricarlas en Windows Azure y mantener la información en la nube. Los respaldos de SQL Azure, el lenguaje de consulta Transact-SQL (T-SQL) de SQL Server, ofrece ayuda para una alta accesibilidad y adaptación a fallos no críticos y tienen en cuenta la información que se debe sincronizar entre SQL Server local y las bases de datos en la nube.

Los investigadores de bases de datos han traído un par de contrastes críticos entre SQL Azure y SQL Server. Por ejemplo, SQL Azure no refuerza todas las direcciones T-SQL, los tipos de información de SQL Server como XML o las capacidades, por ejemplo, las combinaciones de bases de datos cruzadas, y la mejor estimación de la base de datos actualmente es de solo 50 GB. Además, la

descarga actual no realiza los refuerzos planificados y se restablece, sin embargo, Microsoft ha dicho que esto se solucionará en futuras descargas.

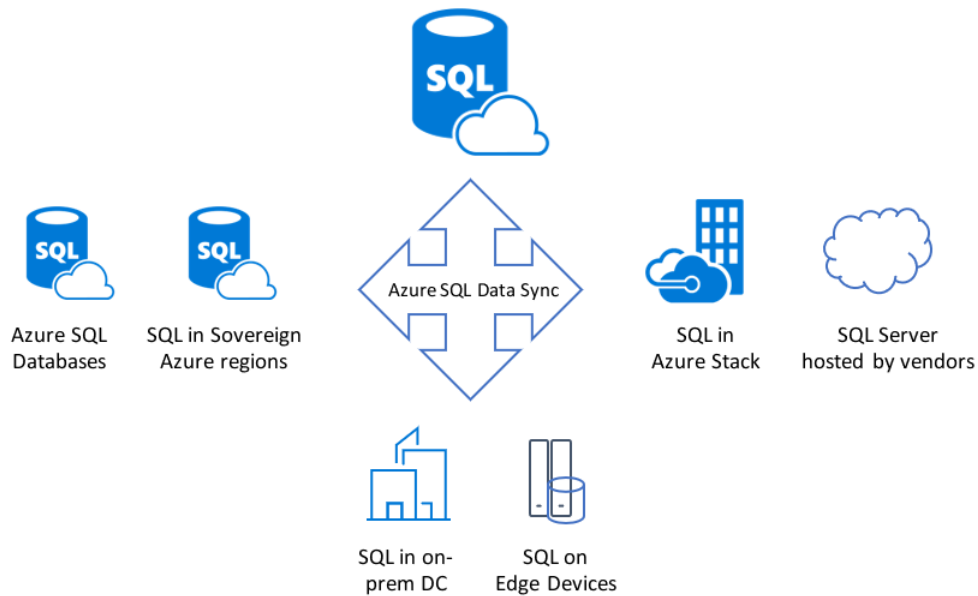


Figura 14. Esquema de los diferentes servicios de SQL Server que existen para Azure.

2.1.13 Key Vault

Microsoft Azure Key Vault es una administración de ejecutivos facilitada por la nube que les permite a los clientes codificar claves y hechos poco privilegiados mediante el uso de claves garantizadas por los módulos de seguridad de equipos (HSM). Pequeños datos internos son información de menos de 10 KB como contraseñas y documentos .PFX. Un HSM es un equipo protegido y seguro que almacena claves criptográficas. Las claves también pueden ser importadas o creadas en HSM que han sido confirmadas a medidas FIPS 140-2 nivel 2.

Ni las aplicaciones ni Microsoft tienen acceso directo a las claves, y los clientes permiten las autorizaciones para que las aplicaciones propias y externas utilicen las claves según sea necesario. Las aplicaciones compuestas con el paquete de ingeniero de programación (SDK) de Azure Storage pueden codificar información de forma natural con una tecla ace en la bóveda de claves. El SDK de almacenamiento de Azure permite a los ingenieros construir aplicaciones de Azure que explotan

activos de computación distribuidos adaptables. Los clientes también pueden transmitir autenticaciones a las máquinas virtuales de Azure y tratar las declaraciones independientemente de la imagen de la máquina virtual.

La cámara acorazada de color azul violáceo que antes estaba accesible como una revisión abierta en enero de 2015 y resultó ser comúnmente accesible en junio de 2015. Está disponible en los niveles de administración Estándar y Premium. No hay gastos de instalación y los clientes pagan por actividades y claves.

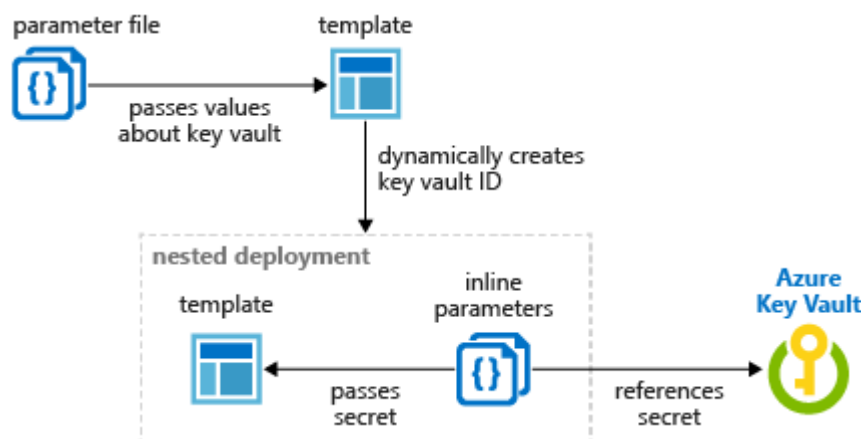


Figura 15. Ejemplo de como crear el Key Vault ID y usarlo como parámetro.

2.1.14 Paquetes SSIS

Un paquete es una recopilación compuesta de asociaciones, componentes de flujo de control, componentes de flujo de información, controladores de ocasión, factores, parámetros y configuraciones, que se recopilan utilizando los dispositivos de plan gráfico que los Servicios de integración de SQL Server proporcionan o fabrican automáticamente. En ese momento, guarda el paquete terminado en SQL Server, el almacén de paquetes SSIS o el marco de registro, o puede enviar la tarea ssISnoverversion al servidor SSIS. El paquete es la unidad de trabajo que se recupera, ejecuta y ahorra.

2.1.15 Trigger

El SQL Server Trigger es un tipo poco común de métodos de almacenamiento que se ejecutan naturalmente cuando ocurre una ocasión en un servidor de base de datos en particular. SQL Server nos proporciona dos tipos

principales de activadores: los activadores DML y los activadores DDL. Los activadores DDL se terminarán debido a varias ocasiones en el lenguaje de definición de datos (DDL), por ejemplo, la ejecución de proclamaciones CREATE, ALTER, DROP, GRANT, DENY y REVOKE T-SQL. El Disparador de DDL puede reaccionar a las actividades de DDL evitando que estas progresiones influyan en la base de datos, realizar otra actividad a la luz de estas actividades de DDL o registrar estas progresiones que se ejecutan contra la base de datos.

El SQL Server DML Trigger es un tipo único de estrategias de guardado destinadas a realizar una sucesión de actividades en una tabla de base de datos, a la que se une el Trigger, cuando ocurre un lenguaje de manipulación de datos (DML), por ejemplo, INSERT, ACTUALIZAR o BORRAR la actividad, salta al alterar la esencia de las tablas o perspectivas de la base de datos, prestando poca atención a si las columnas de la tabla están influenciadas. Los Desencadenadores contrastan con los métodos de eliminación en que los Desencadenadores se terminan naturalmente cuando ocurre una alteración de información predefinida. Los activadores de DML se pueden utilizar para mantener la confiabilidad de la información y mantener las reglas comerciales de la organización, al igual que la utilidad de los imperativos de verificación de tablas y claves remotas, mediante la realización de formularios de revisión y otras actividades posteriores a DML. Puede utilizar los Desencadenadores DML para consultar diferentes tablas y realizar consultas complejas de T-SQL.

En el caso de que se finalice el activador, se utilizará un tipo excepcional de tablas virtuales denominadas tablas insertadas y eliminadas para mantener la información que se estima cuando se realiza la alteración. La articulación del Disparador funcionará en la medida de un intercambio similar que dispare ese Disparador. Esto implica que el intercambio no se enviará totalmente hasta que la explicación de Trigger se complete de manera efectiva. Por otra parte, el intercambio se moverá hacia atrás si la proclamación del Disparador se queda corta.

2.1.16 Service Desk

Una mesa de servicio es un centro de intercambio que proporciona un propósito de contacto (SPOC) solitario entre una organización y sus clientes, trabajadores y colegas. El motivo de una mesa de servicio es garantizar que los clientes reciban la ayuda adecuada de manera oportuna.

Los escritorios de servicio están destinados a lidiar con los dos episodios y las demandas de servicio. En esta circunstancia específica, un episodio es una ocasión que resulta en una perturbación en la accesibilidad o la calidad del servicio. Una demanda de servicio busca ayuda con una asignación normal, por ejemplo, ayudando a un cliente a cambiar una frase secreta u configurando otro cliente en los marcos de trabajo. Los diferentes servicios prestados pueden incluir cambios en la junta directiva, despedir a los ejecutivos y tareas relacionadas con la configuración.

La mesa de servicio es una de las tres alternativas fundamentales para el cliente o, potencialmente, el soporte al cliente. Las otras dos sustancias SPOC son llamadas focales y mesas de ayuda. Existe cierta perplejidad acerca del contraste entre las mesas de servicio y las mesas de ayuda. Como lo indica la definición de ITILv.2, la mesa de servicio era simplemente un término más para la mesa de ayuda. Sea como sea, ITILv.3 se separa entre los dos, diciendo que los escritorios de servicios ofrecen un alcance más amplio de habilidades de servicio, abordan más problemas con menos avances y habilitan la coordinación de formularios comerciales en el marco del servicio de la junta.

La Biblioteca de Infraestructura de Tecnología de la Información (ITIL) caracteriza a la mesa de servicio como un dispositivo fundamental de servicio de innovación de datos (ITSM). ITIL es una acumulación de las mejores prácticas para el servicio de TI de la junta directiva.

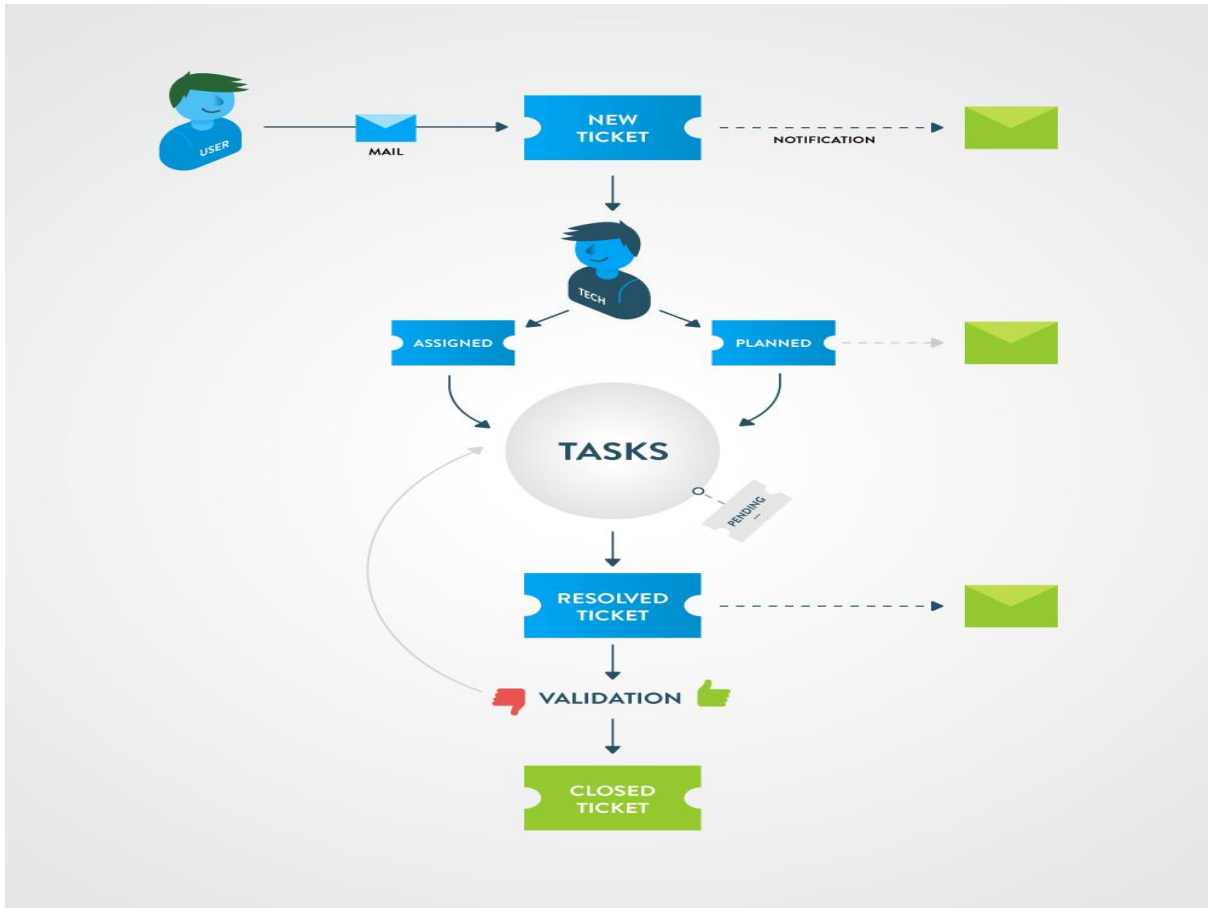


Figura 16. Ejemplo de flujo de proceso de un Service Desk

2.2 Resumen Capitulo II

La mayoría de estas plataformas son creadas por la compañía tecnológica Microsoft ya que ha demostrado tener un conjunto de plataformas sólidas que supervisa el estado de tu empresa mediante un panel activo, obtienes informes interactivos y acceso a los datos sobre la marcha. Estas plataformas cuentan con computación en la nube, es decir las plataformas se sostienen en infraestructuras alternas que permiten la escalabilidad de una manera más rápida y sencilla. Azure, te permite minimizar el presupuesto que tenías destinado para el hardware, lo cual además garantizará un ahorro significativo de energía dentro de la organización. Por otro lado, el servicio de Microsoft Azure únicamente cobra los productos que estás utilizando, de tal forma que se adapta a cualquiera de las necesidades de la empresa y brinda un ahorro significativo en el consumo, mantenimiento y soporte de esta.



Figura 17. Diagrama de resumen capítulo 2.

Capítulo III:
**Implementación de Solución de Inteligencia de
Negocios para el proceso de Gestión de Incidentes
de Grupo de Seguros G**

3.1 Arquitectura de la solución de BI a implementar

En la actualidad Seguros G tienen como meta migrar sus sistemas actuales a la nube, esto forma parte del proceso de modernización de su infraestructura tecnológica. La plataforma que han seleccionado y trabajado durante los últimos 9 meses es Microsoft Azure, debido a su escalabilidad y rendimiento, por lo que la solución de Inteligencia de Negocios se implementará en esta plataforma.

Microsoft Azure es una plataforma de servicios integrados en la nube en la cual los profesionales de TI (Tecnología de la Información) pueden en este caso extraer, procesar y almacenar datos en sus bases de datos las cuales se encuentran en su centro de datos y se pueden acceder mediante el internet.

Para la extracción, modelado o transformación y carga de los datos es necesario una máquina virtual la cual deberá contar con una dirección IP y Microsoft Integration Runtime, este último utilizado por el Data Factory para proveer las habilidades de integración a través de las diferentes bases de datos. Este Data Factory será el encargado de extraer, transformar y cargar los datos desde la base de datos de origen a las bases de datos “Staging Area” y “DataMart”. Microsoft Azure permite crear el Data Factory y todos los componentes necesarios en su plataforma y además permite la integración con Power BI, la cual permitirá crear los reportes y finalmente ser visualizados por los usuarios interesados.

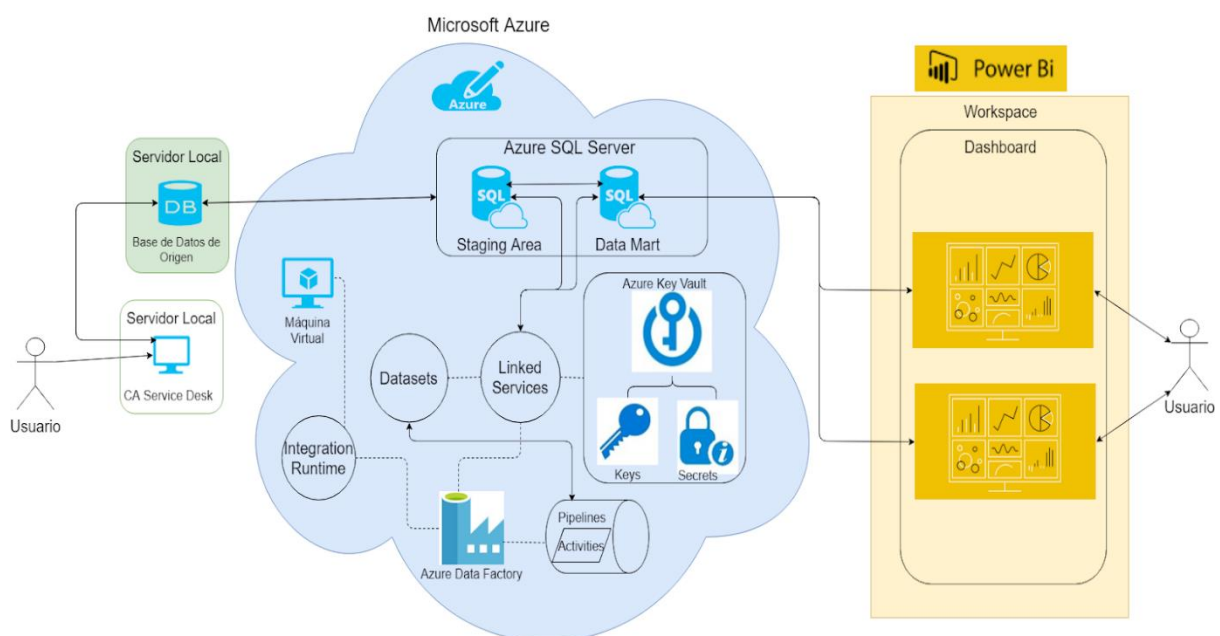


Figura 18. Esquema de arquitectura de BI. **Fuente:** Propia.

3.2 CA Unicenter Service Desk

Es aquella aplicación la cual es utilizada para registrar las solicitudes de incidentes del Grupo de Seguros G, la misma es una aplicación web que se encuentra alojada en un servidor y almacena información en una base de datos que en el transcurso de este trabajo llamaremos “Base de Datos de Origen”.

Una vez creada una solicitud inicia el proceso de gestión de incidentes, esta solicitud es atendida por el analista del área de Mesa de Ayuda el cual deberá determinar si lo reportado por el usuario es en realidad un incidente, en caso de ser un incidente este procederá a enviar la solicitud al área de mantenimiento de aplicaciones, esta área deberá de desarrollar una solución para este incidente y luego enviar la solicitud al área de Calidad de Software, para realizar las pruebas de calidad necesarias y validar que el incidente fue resuelto, una vez validado esto, se envía la solicitud a Gestión de Cambios el cual programa la fecha de despliegue de la solución al ambiente de producción, se cierra la solicitud sea realizado dicho despliegue. (Para entendimiento general del proceso, ver anexo A).

Debajo se muestran imágenes de la aplicación CA Unicenter Service Desk, cabe resaltar que dicha aplicación no será modificada durante la implementación de esta solución.

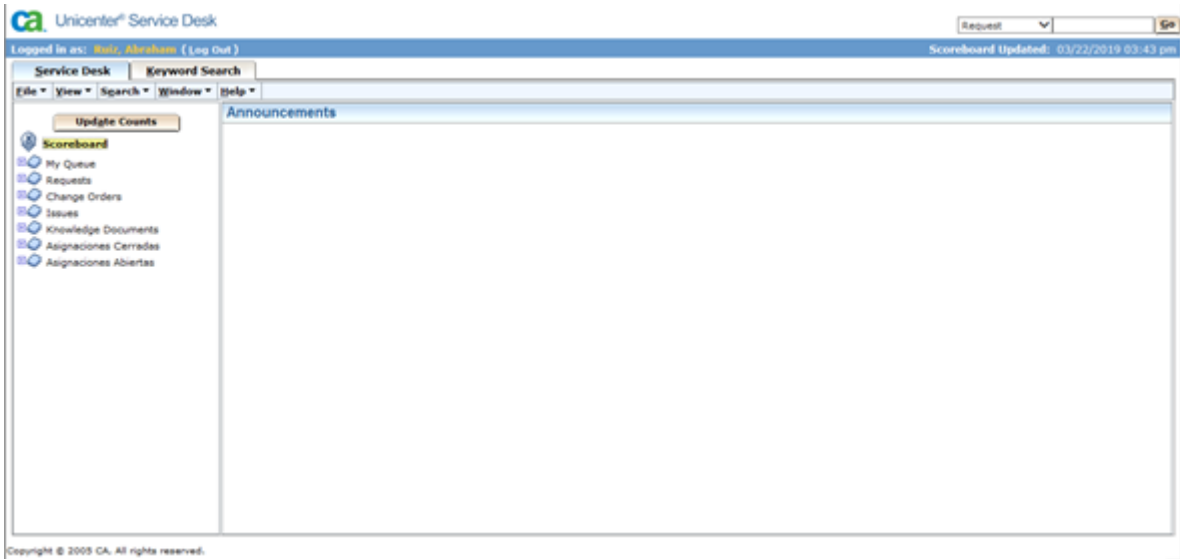


Figura 19. Página Principal de la aplicación CA Unicenter Service Desk. **Fuente:** Propia.

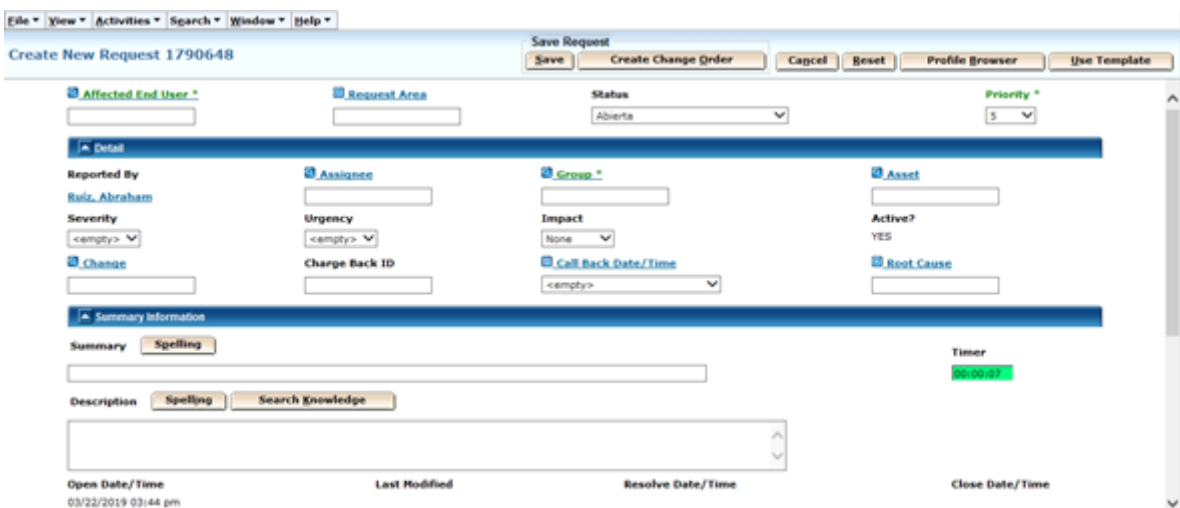


Figura 20. Creación de Solicitudes en la aplicación CA Unicenter Service Desk. **Fuente:** Propia.

1791081 Request Detail

Affected End User: López, Chayra | Request Area: Tecnología, Ingeniería, Base de Datos | Status: Cerrada | Priority: 5

Detail

Reported By: López, Chayra	Assignee: puello, Luis Alfredo	Group: TI, Ingeniería	Asset:
Severity: Urgency	Impact: None	Active?: NO	Root Cause: Ac
Change:	Charge Back ID:	Call Back Date/Time:	

Summary Information

Summary: Refrescar la base de datos px | Total Activity Time: 00:00:05

Description: Favor refrescar la Base de Datos px

Open Date/Time: 03/27/2019 09:28 am	Last Modified: 03/27/2019 10:10 am	Resolve Date/Time: 03/27/2019 10:10 am	Close Date/Time: 03/27/2019 10:10 am
-------------------------------------	------------------------------------	--	--------------------------------------

Request Activity Log List

Created By / Description	On *	Time Spent	Type
puello, Luis Alfredo	03/27/2019 10:10 am	00:00:05	Close
Status changed from 'Abierta' to 'Cerrada'			
System_AHD_generated	03/27/2019 10:10 am	00:00:00	Survey Sent
Survey request ('Enc_Tecnolog') sent to 'López, Chayra'.			
López, Chayra	03/27/2019 09:28 am		Initial
AHD05439: Copied from Request 1790447			

Figure 21. Resumen de la solicitud una vez cerrada. Fuente: Propia.

3.3 Base de datos de Origen

Es una base de datos transaccional donde se aloja actualmente las solicitudes creadas por los usuarios del negocio en el Service desk, a esta base de datos se le deberá agregar una tabla llamada BS_PARAMETRO desde la cual llevaremos control de los registros que se cargan a las demás bases de datos al finalizar el proceso de ETL, además se deberá de agregar una columna llamada CODPARAM a la tabla CODSOLI, con el objetivo de crear una relación entre estas dos tablas.

Script para agregar crear tabla BS_PARAMETRO

```
CREATE TABLE BS_PARAMETRO
(
CODPARAM INT,
VALUEPARAM
DESCRIPCION VARCHAR (255),
FECHAINI DATETIME,
FECHFIN DATETIME,
CODCIAAPLIC INT
);
```

Script para agregar columna CODPARAM a tabla CODSOLI

```
ALTER TABLE CASOLI
ADD FOREIGN KEY (CODPARAM)
REFERENCES BS_PARAMETRO (CODPARAM);
```

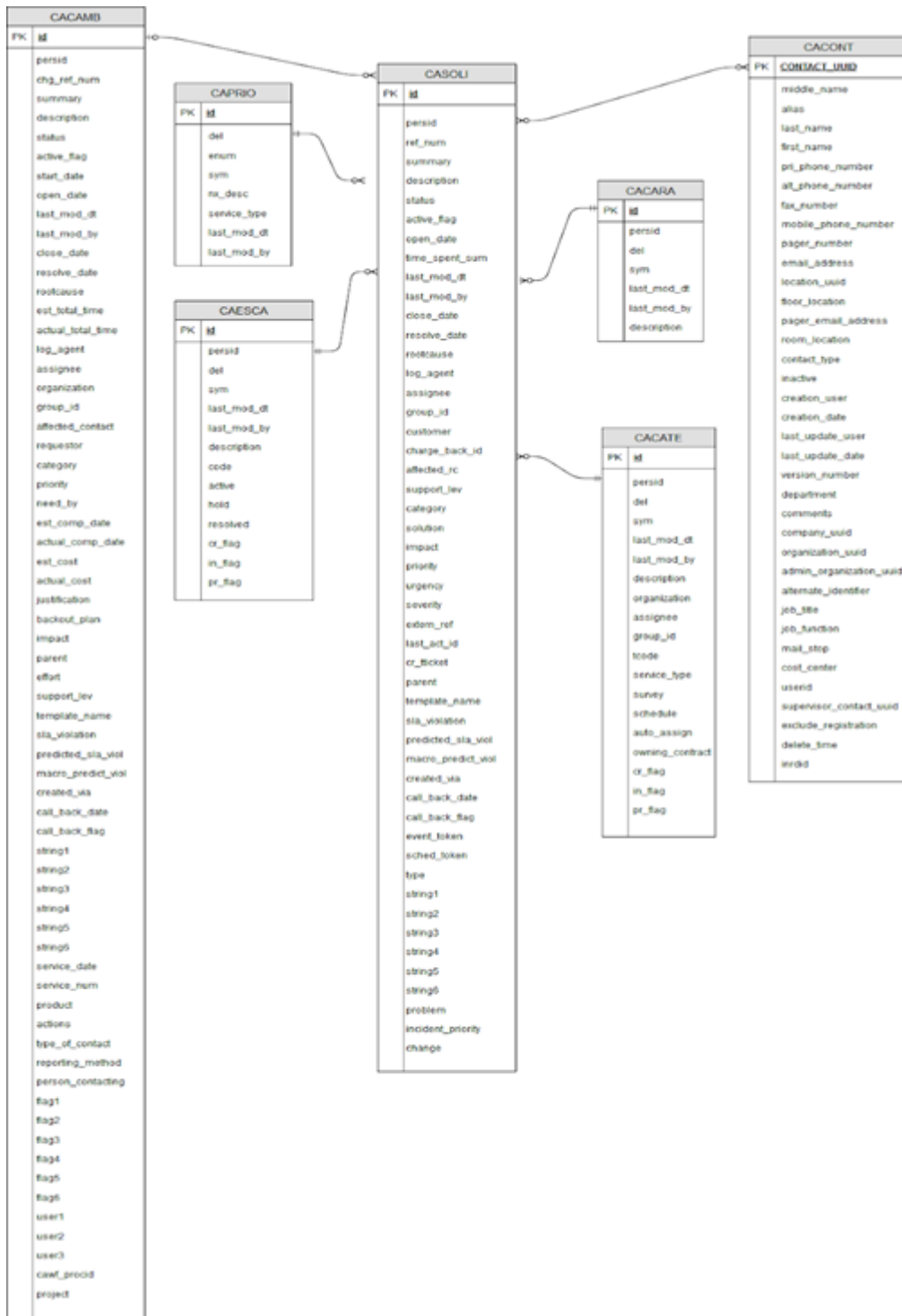


Figura 2. Esquema de la Base de Datos de Origen actual, sin ningún cambio.

Fuente: Propia.

3.4 Máquina Virtual

La máquina virtual es un recurso el cual simula un sistema de computación real y puede ejecutar programas de igual manera, para esta solución en específico es necesaria una máquina virtual debido a que esta será la infraestructura que permitirá realizar la integración de los datos mediante **Integration Runtime**, el cual deberá ser instalado en la máquina virtual.

La plataforma de Microsoft Azure ofrece la posibilidad crear máquinas virtuales en la nube las cuales permitirán realizar todas las funciones necesarias sin necesidad de comprar y mantener un hardware físico para correr dicha máquina virtual.

Se realizó una prueba instalando una máquina virtual con Windows Server 2018. A continuación, se muestra el proceso de creación de la máquina virtual en la nube Microsoft Azure:

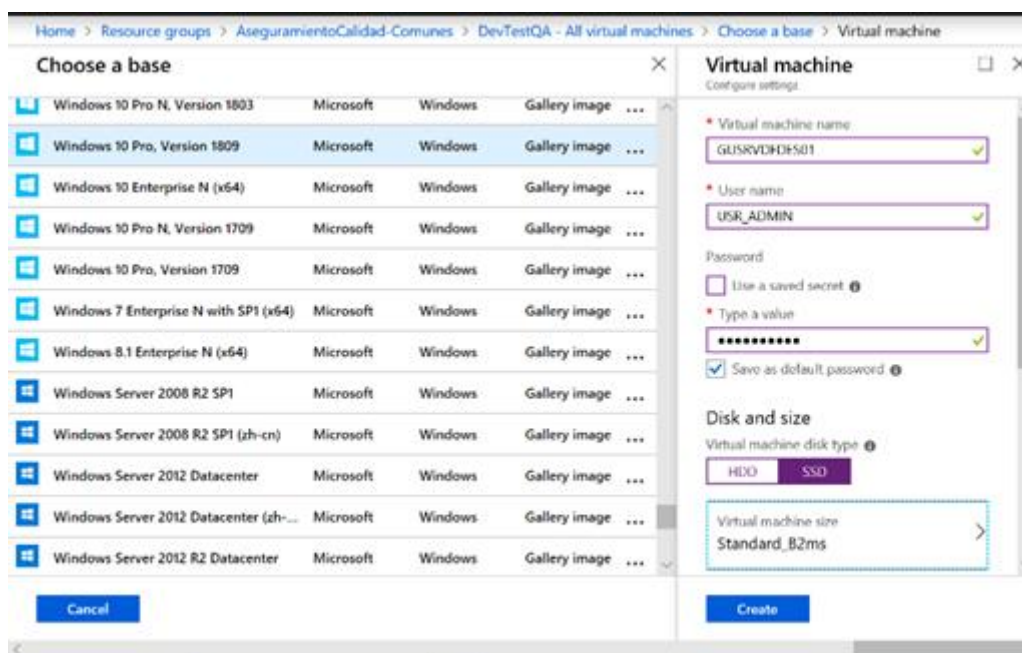


Figura 22. En el proceso de creación se deben de llenar los campos “Virtual machine name”, “User name”, “Contraseña” y seleccionar el tipo de disco y el sistema operativo. **Fuente:** Propia.

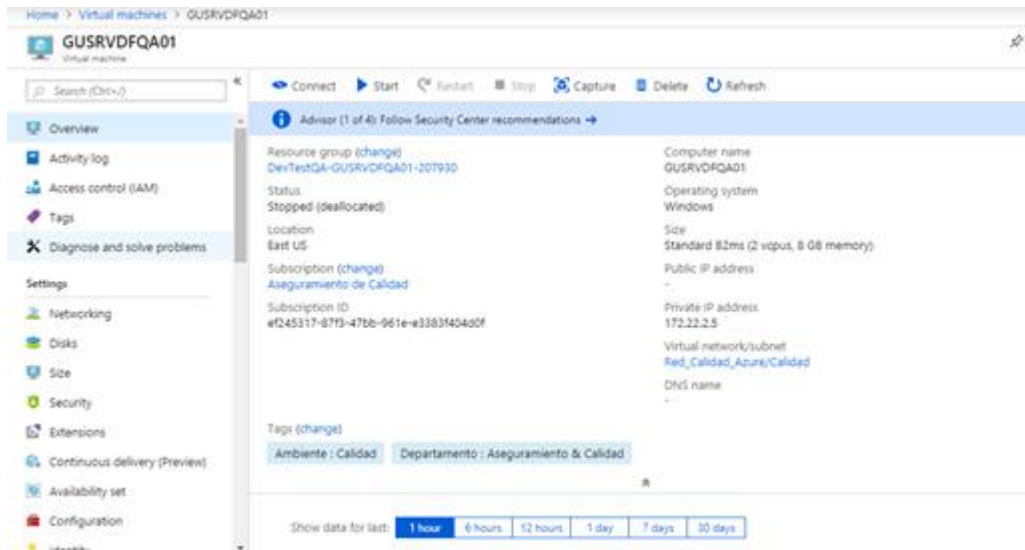


Figura 23. Resumen luego de la creación de la máquina virtual. **Fuente:** Propia.

3.5 Integration Runtime

“Es la infraestructura informática que utiliza Azure Data Factory para proporcionar las siguientes capacidades de integración de datos en diversos entornos de red”. (Microsoft, 2018)

Movimiento de datos: para el movimiento de datos, Integration Runtime mueve los datos entre los almacenes de datos de origen y destino, al tiempo que brinda soporte para conectores integrados, conversión de formato, asignación de columnas y transferencia de datos escalable y de alto rendimiento.

Ejecución de paquetes SSIS: ejecuta de forma nativa paquetes SSIS en un entorno de proceso de Azure administrado. Integration Runtime permite también distribuir y supervisar actividades de transformación que se ejecutan en una gran variedad de servicios de proceso, como: Azure SQL Database, SQL Server, etc.

Para este modelo se utilizará solo una instancia del Integration Runtime para mover y transformar los datos, este se ejecutará en una red privada.

Se realizó una prueba instalando Integration Runtime en la Máquina Virtual creada, a continuación, se muestra el proceso de instalación y configuración de esta en una máquina virtual en Azure.

Se debe instalar el runtime que puede ser descargado desde la siguiente url:

<https://www.microsoft.com/en-us/download/details.aspx?id=39717>

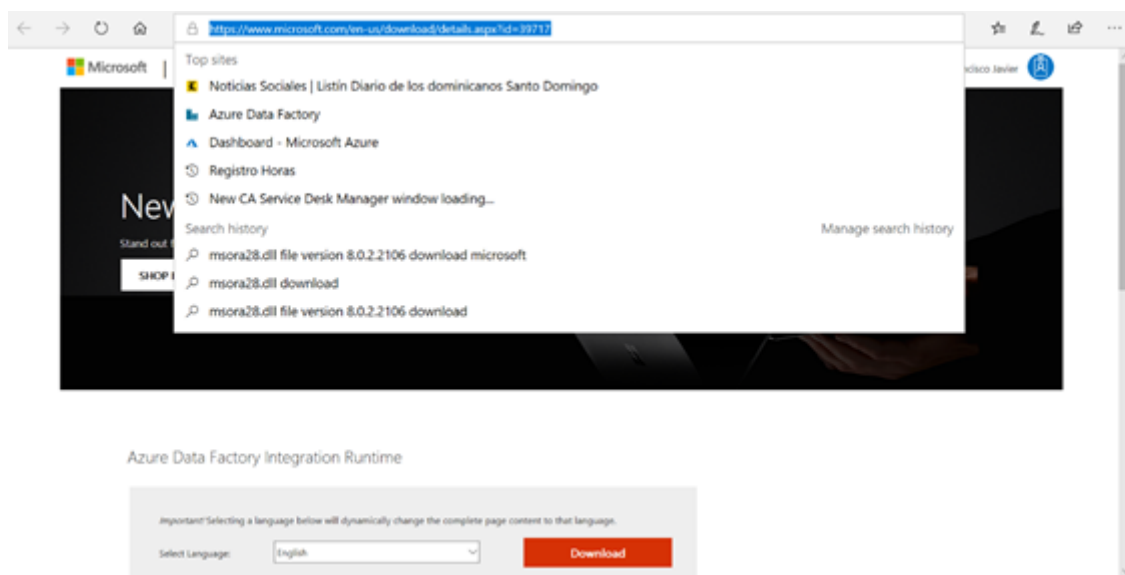


Figura 24. Se debe de descargar del Integration Runtime desde la Máquina Virtual utilizando la url indicada. **Fuente:** Propia.

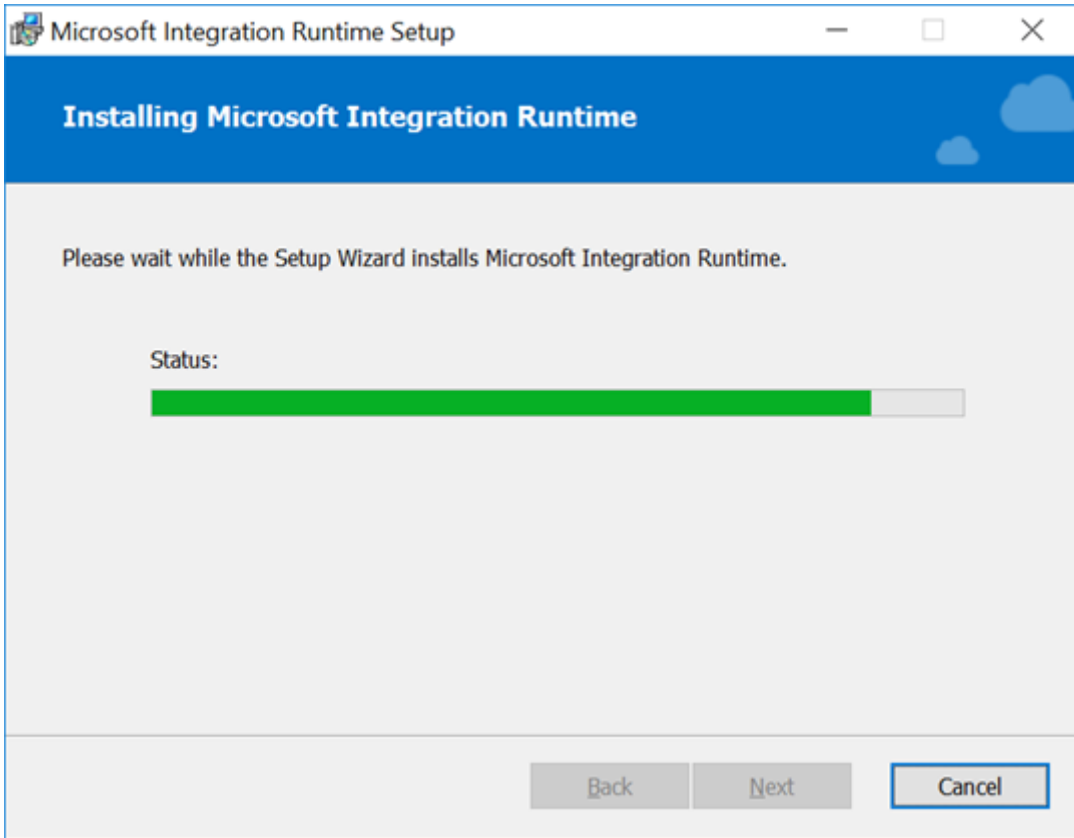


Figura 25. Una vez descargado el archivo se procede a instalar el Integration Runtime en la Máquina Virtual. **Fuente:** Propia.

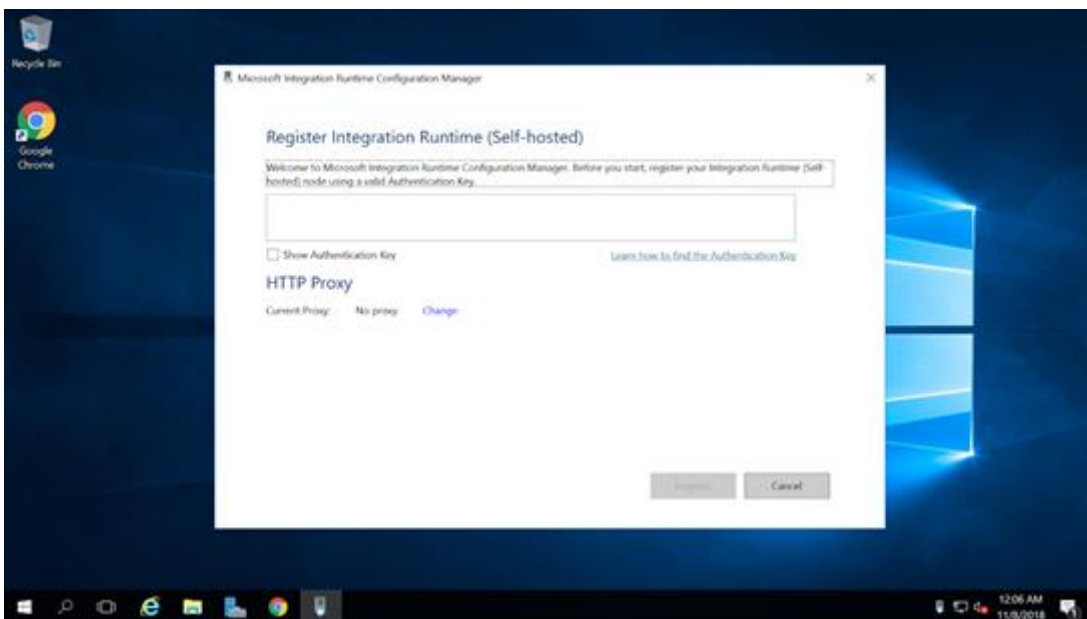


Figura 26. Así se muestra el Integration Runtime una vez instalada en la Máquina Virtual. **Fuente:** Propia.

3.6 Creación SQL Server (Logical Server)

Un servidor de base de datos SQL es una construcción lógica que actúa como un punto de administración central para múltiples bases de datos individuales o agrupadas. Para esta solución se debe de crear un servidor lógico el cual será el encargado de hospedar las bases de datos “Staging Area” y “DataMart. Todas las bases de datos administradas por un servidor de base de datos SQL se crean dentro de la misma región que el servidor de base de datos SQL, la región en la cual se creará este servidor será “East US”.

Cuando se crea un servidor de base de datos SQL, se debe de proporcionar una cuenta de inicio de sesión de servidor y una contraseña que tiene derechos administrativos para las bases de datos alojadas en este servidor, estas credenciales serán necesarias al momento de configurar los reportes en Microsoft Power BI.

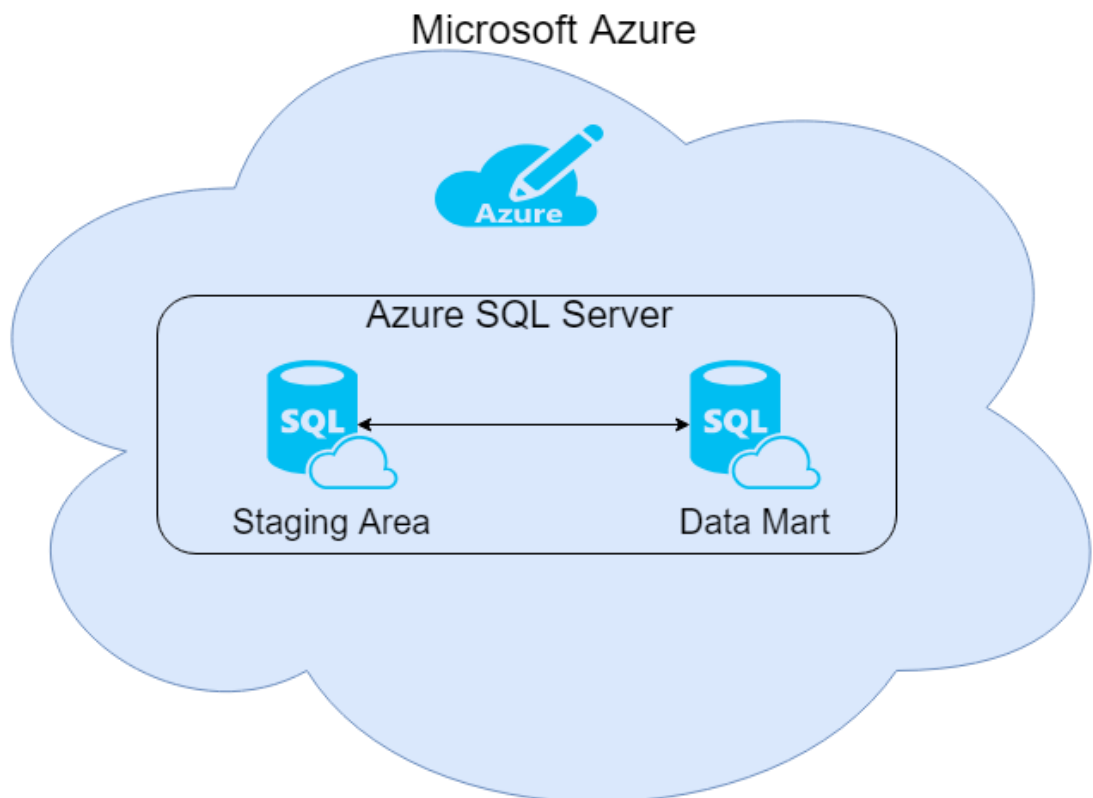
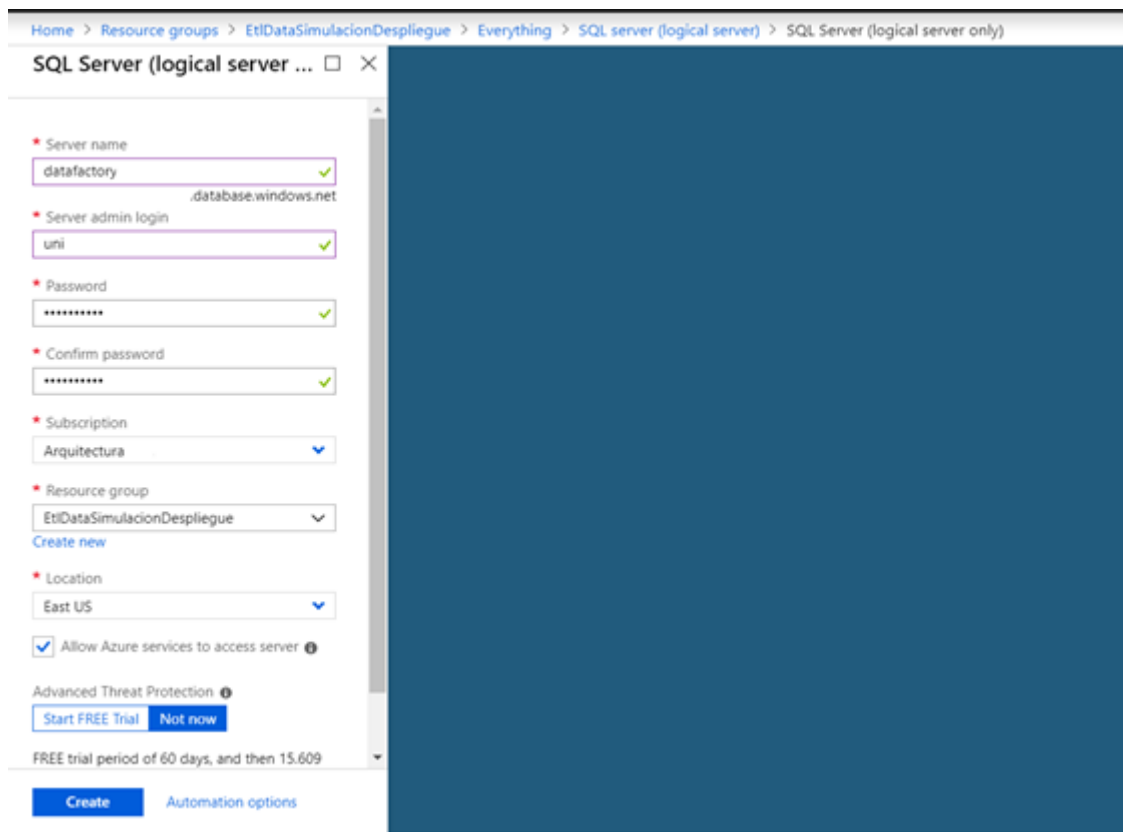


Figura 27. Esquema Azure SQL Server. **Fuente:** Propia

Se realizó una prueba creando un servidor lógico en la plataforma de Microsoft Azure. A continuación, se muestra el proceso.



The screenshot displays the Azure portal interface for creating a logical SQL server. The breadcrumb navigation at the top reads: Home > Resource groups > EtlDataSimulacionDespliegue > Everything > SQL server (logical server) > SQL Server (logical server only). The main form is titled "SQL Server (logical server ...)" and contains the following fields and options:

- Server name:** "datafactory" (with a green checkmark and ".database.windows.net" suffix).
- Server admin login:** "uni" (with a green checkmark).
- Password:** "*****" (with a green checkmark).
- Confirm password:** "*****" (with a green checkmark).
- Subscription:** "Arquitectura" (dropdown menu).
- Resource group:** "EtlDataSimulacionDespliegue" (dropdown menu, with a "Create new" link below it).
- Location:** "East US" (dropdown menu).
- Allow Azure services to access server:** Checked checkbox.
- Advanced Threat Protection:** "Start FREE Trial" and "Not now" buttons.
- Footer:** "FREE trial period of 60 days, and then 15.609" and a "Create" button with a link to "Automation options".

Figura 28. En el proceso de creación se deben de llenar los campos “Server name”, “Server admin login”, “Password”, “Confirm Password”, “Subscription”, “Resource Group” y “Location”. **Fuente:** Propia.

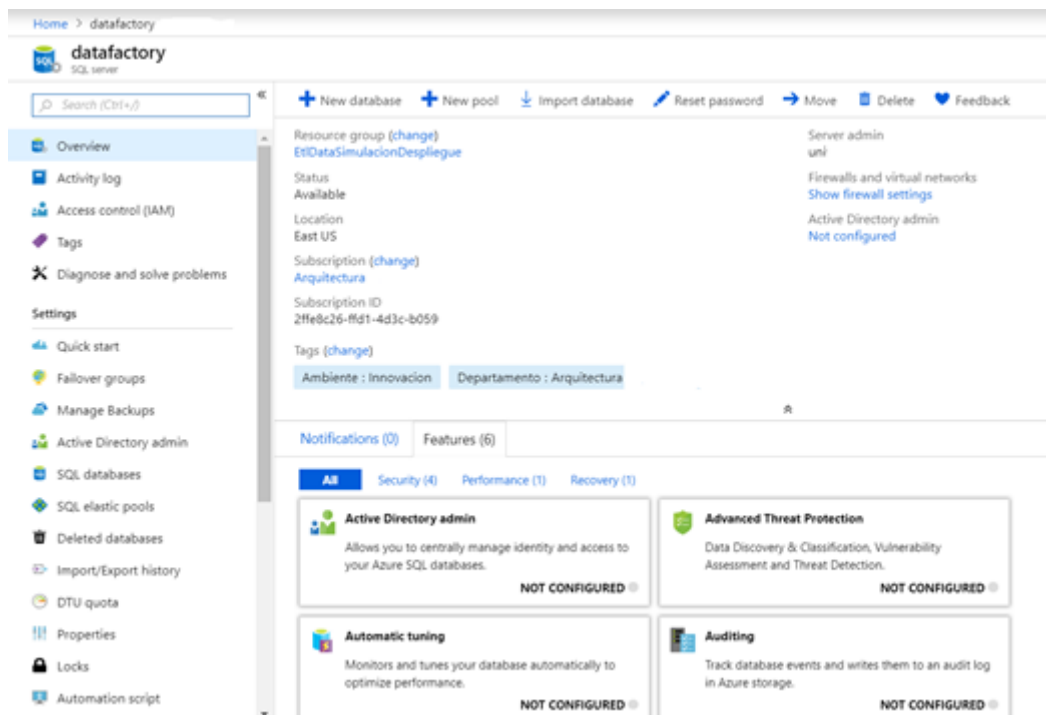


Figura 29. Resumen luego de la creación de la creación del servidor SQL.
Fuente:Propia.

3.7 Almacenamiento de Datos

Para el almacenamiento de datos correspondientes a esta solución se deberán crear dos bases de datos en el la plataforma Azure, esta permite crear bases de datos llamadas Azure Sql Database, esto es un servicio totalmente relacional, gestionadas totalmente desde la nube compatible con el motor SQL Server. El objetivo de estas bases de datos en la nube es la de almacenar los datos transformados modelados y los no modelados, por el Data Factory.

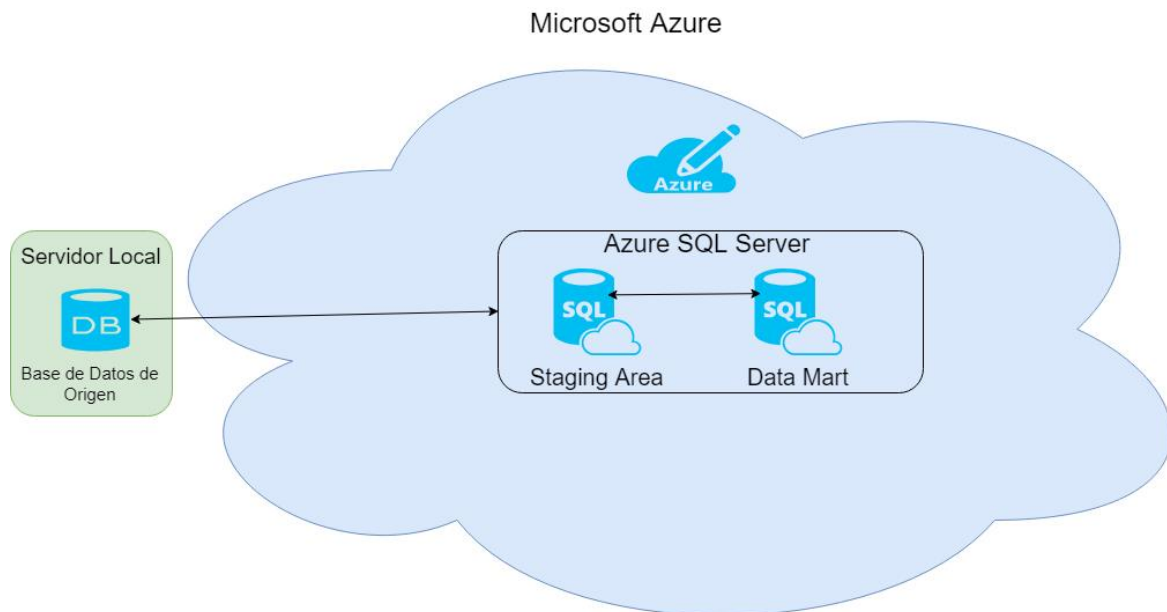


Figura 30. Esquema de bases de datos, en esta se muestran las bases de datos necesarias para la implementación de esta solución. **Fuente:** Propia

3.7.1 Arquitectura a utilizar para el almacenamiento de datos

La metodología a utilizar para la implementación del data warehouse será la metodología Kimball la cual se enfoca en los datos relacionados a una área del negocio denomina el ciclo de vida del negocio.

Justificación

La metodología a utilizarse para la construcción del data warehouse será kimball, debido a que esta se:

- Se enfoca en el negocio y en las necesidades del negocio.
- El esquema de estrella puede ser entendido fácilmente por los usuarios y es fácil para el uso de reportes.
- La mayoría de las herramientas de BI entienden el esquema de estrella el cual es una característica de la metodología kimball.
- Una de las características de esta metodología es que el almacenamiento de datos es pequeño, ósea ocupa menos espacio en la base de datos y hace que la administración del sistema sea fácil.

3.7.2 Creación de Azure Database – STAGING AREA

La base de datos llamada “Staging área” será desde la cual se alimentará el datamart, esta poseerá una copia de todos los campos que contengan las tablas CACARA, CACONT, CAESCA, CASOLI, CACAMB, CAPRIO y CACATE.

Al momento de crear una base de datos en Microsoft Azure no estamos obligados a indicar las especificaciones que debe de tener el servidor, Azure nos da la libertad de indicar la cantidad de DTU's (Data Transmission Unit) que se considera necesaria. Se debe de tener en cuenta que a más DTU's mayor será el costo. Para este caso la cantidad de DTU's a la cual se configurarán las base de datos será de 200 DTU.

Se realizó una prueba creando una base de datos en el ambiente de Microsoft Azure. A continuación, se muestra el proceso de creación y configuración:

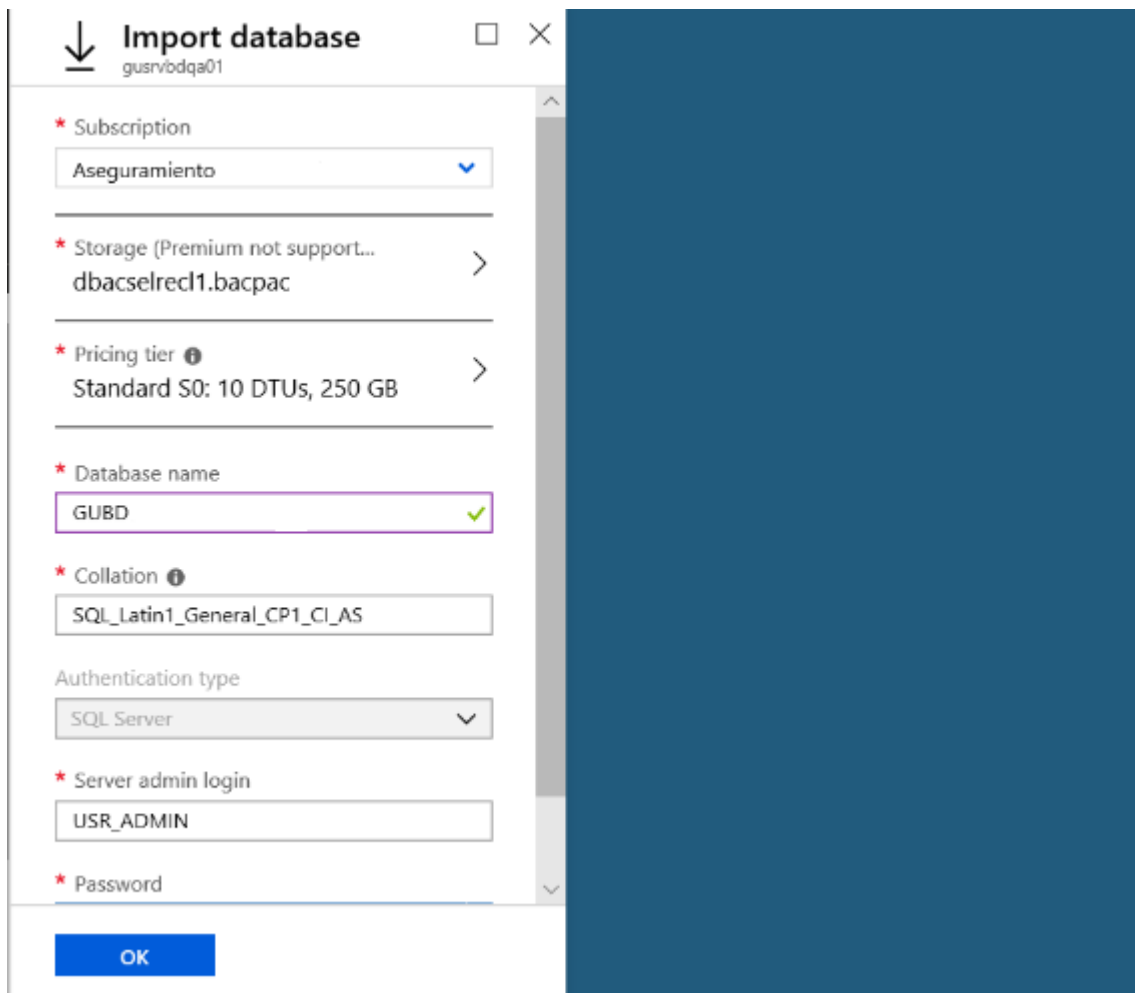


Figura 31. En el proceso de creación se deben de llenar los campos “Subscription”, “Pricing Tier”, “Database name”, “Collation”, “Server admin login” y “Password”. En estos últimos campos se debe de ingresar las mismas credenciales que se ingresaron durante la creación del servidor SQL. **Fuente:** Propia.



Figura 32. Luego de creada la base de datos se procede configurar la misma a 200 DTU's. Esto aumentará la capacidad de procesamiento de la base de datos.
Fuente: Propia.

Una vez finalizado este proceso se deberá proceder a crear la base de datos "Datamart" utilizando el mismo proceso.

3.7.3 Creación de Azure Database - DATAMART

La base de datos llamada "DataMart" a la cual se le cargarán los datos que que serán consumidos por los reportes de BI.

Para la carga de los datos se utilizarán queries los cuales modelan los datos deseados a diferencia de la base de datos "Staging Area" la cual contendrá todos los datos de las tablas de la base de datos de Origen.

Al momento de crear una base de datos en Microsoft Azure no estamos obligados a indicar las especificaciones que debe de tener el servidor, Azure nos da la libertad de indicar la cantidad de DTU's (Data Transmission Unit) que se considera necesaria. Se debe de tener en cuenta que a más DTU's mayor será el costo. Para este caso la cantidad de DTU's a la cual se configurarán las base de datos será de 200 DTU.

Se realizó una prueba creando una base de datos en el ambiente de Microsoft Azure. A continuación, se muestra el proceso de creación y configuración:

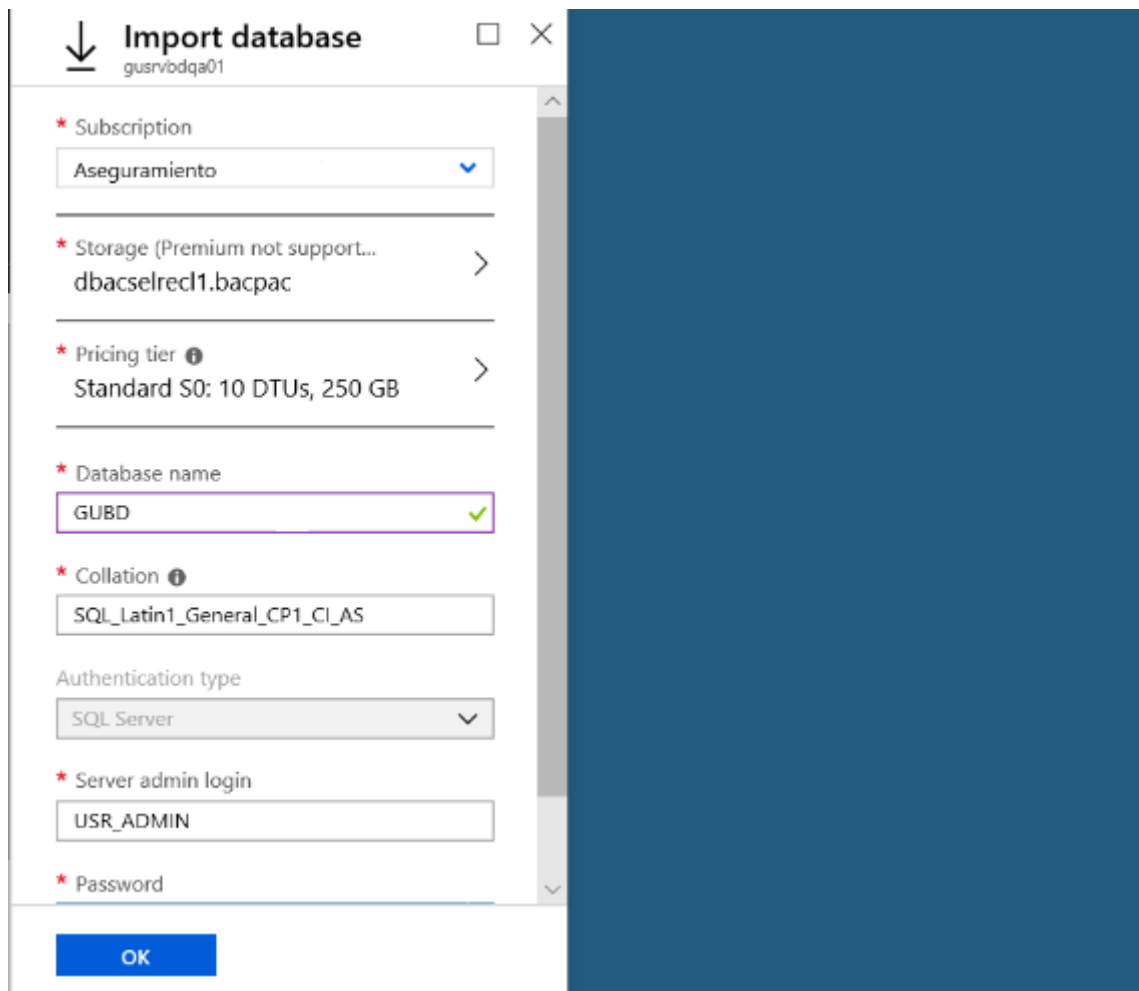


Figura 33. En el proceso de creación se deben de llenar los campos “Subscription”, “Pricing Tier”, “Database name”, “Collation”, “Server admin login” y “Password”. En estos últimos campos se debe de ingresar las mismas credenciales que se ingresaron durante la creación del servidor SQL. **Fuente:** Propia.



Figura 34. Luego de creada la base de datos se procede configurar la misma a 200 DTU's. Esto aumentará la capacidad de procesamiento de la base de datos.
Fuente: Propia.

Una vez finalizada la creación y configuración de las bases de datos se debe de proceder a crear las conexiones necesarias que permitirán la comunicación entre el Data Factory y las bases de datos.

3.8 Bóveda de Claves criptográficas Key Vault

Las aplicaciones y servicios en la nube utilizan claves y secretos criptográficos para ayudar a mantener la información, aquí es donde entra en acción el Key Vault.

Key Vault es una herramienta que se utiliza para almacenar y acceder de forma segura los a secretos criptográficos que serán consumidos por los componentes del data factory.

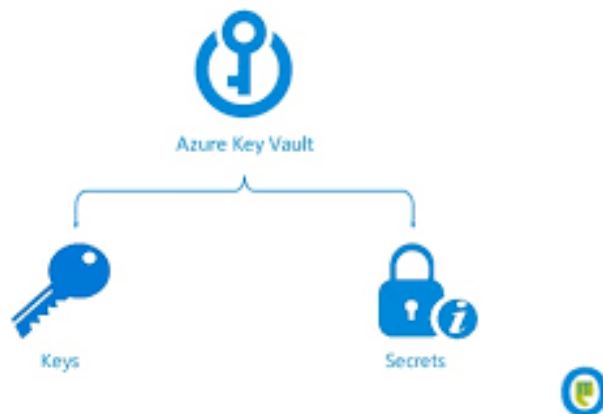


Figura 35. Esquema de Azure Key Vault. **Fuente:** Kerkhove, T. (2015). Securing Sensitive Data with Azure Key Vault.

Se realizó una prueba creando un Key Vault en el ambiente de Microsoft Azure. A continuación, se muestra el proceso de creación.

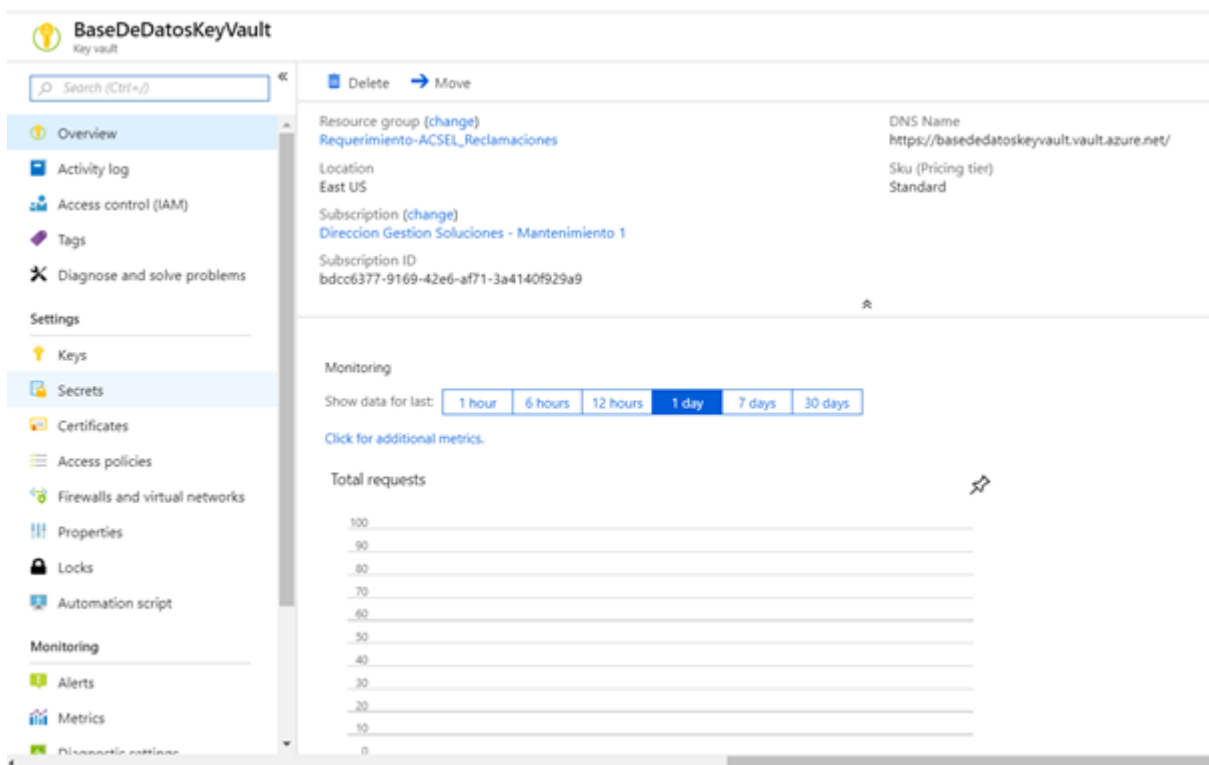


Figura 36. Resumen luego de la creación del Key Vault. **Fuente:** Propia.

3.8.1 Conectores de base de datos en Microsoft Azure - Secrets

Un secret es todo aquellos a lo que se desea controlar de forma estricta el acceso, como las claves de API, string de conexiones, contraseñas, etc. Para la puesta en producción de esta solución de BI será necesario crear tres secrets, los cuales permitirán que los datasets y linked services se puedan conectar a cada una de las bases de datos involucradas en esta solución. Estas son: Base de Datos de Origen (SQL Server), Base de Datos “Staging Area” (Azure SQL) y Base de Datos “DataMart” (Azure SQL).

Se realizó una prueba creando un secret en el ambiente de Microsoft Azure. A continuación se muestra el proceso de creación y configuración:

Home > Requerimiento-ACSEL_Reclamaciones > BaseDeDatosKeyVault - Secrets > Create a secret

Create a secret

Upload options
Manual

* Name ?

* Value
Enter the secret.

Content type (optional)

Set activation date? ?

Set expiration date? ?

Enabled? Yes No

Create

Figura 37. Se debe de llenar los campos “Name”, “Value” y “Content type”, con el objetivo de crear un secret. En el campo “value” se debe de ingresar un string de conexión con un formato en específico, el mismo se muestra más adelante. **Fuente:** Propia.

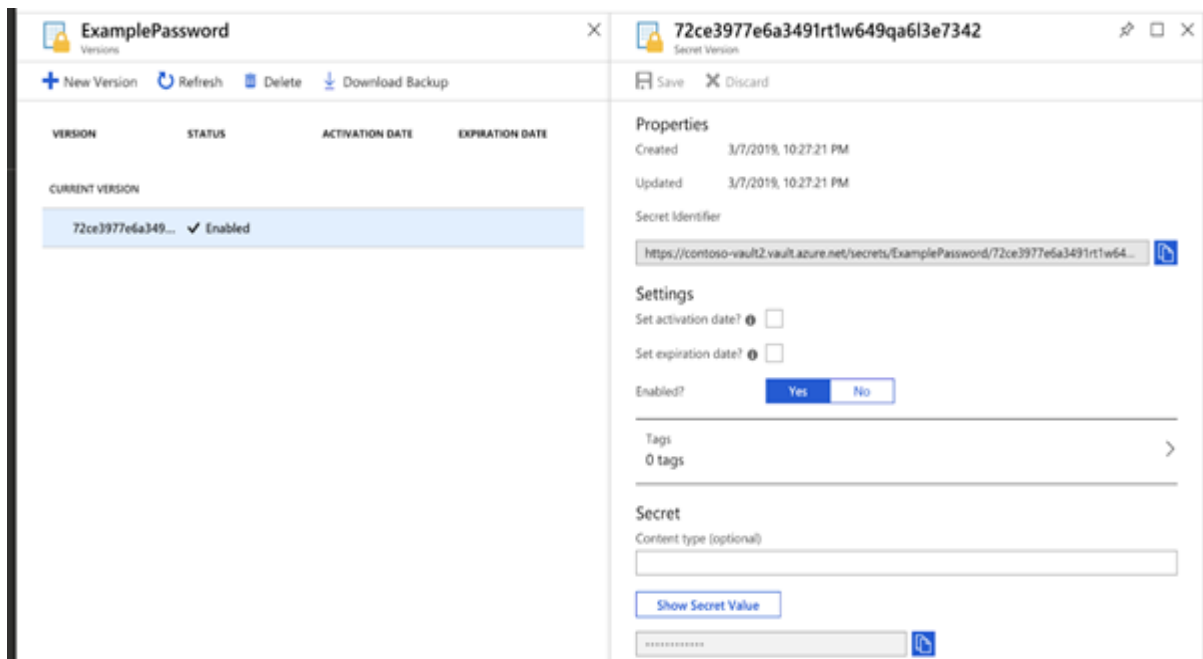


Figura 38. Resumen luego de la creación de la creación del secret. **Fuente:** Propia.

3.8.1.1 Formatos Conexiones

El formato debe tener la siguiente forma:

(Se sustituyen los valores entre las llaves por los valores deseados).

Azure

SQL

```
Server=tcp:{servername}.database.windows.net,1433;Database={databasename};UserID={username}@{servername};Password={password};Trusted_Connection=False;Encrypt=True;ConnectionTimeout=30;
```

Este string de conexión permite la conexión entre el data factory y las bases de datos creadas en la plataforma de Azure mediante los linked services. En este string de conexión se debe de indicar el nombre del servidor, nombre de la base de datos, usuario administrador y su correspondiente contraseña.

SQL Server

```
Data Source={servername}\{instancename(opcional)};Initial Catalog={database};IntegratedSecurity=False;UserID={username};Password={password};
```

Este string de conexión permite la conexión entre el data factory y la base de datos de origen mediante los linked services. En este string de conexión se debe de indicar el nombre del servidor, nombre de la base de datos, usuario administrador y su correspondiente contraseña.

3.9 Azure Data Factory

Azure Data Factory es un servicio basado en la nube que permite la integración de datos, garantizando de esta manera que los flujos de trabajo se integren con los servicios dependientes requeridos en nuestra solución. Este permite que se monitorear las ejecuciones desde un solo panel control.

Para esta solución se necesitará un Data Factory el cual permita el movimiento y transformación de datos de la base de datos de origen a las bases de datos “Staging Area” y “Datamart”, con el objetivo de refinar los datos la base de datos de origen y organizarlo de manera que sea más sencillo el proceso de consumo por parte de los reportes de BI.

Se realizó una prueba creando una un Azure Data Factory en la plataforma de Microsoft Azure. A continuación, se muestra el proceso de creación y configuración:

The image shows a 'New data factory' form with the following fields and values:

- Name:** ACSE (with a green checkmark)
- Subscription:** Arquitectura
- Resource Group:** EtlDataSimulacionDespliegue (with radio buttons for 'Create new' and 'Use existing', where 'Use existing' is selected)
- Version:** V2
- Location:** East US

At the bottom, there is a blue 'Create' button and a link for 'Automation options'.

Figura 39. Se debe de llenar los campos “Name”, “Subscription”, “Resource Group”, “Versión” y “Location”, con el objetivo de crear un Data Factory en la plataforma de Microsoft Azure. **Fuente:** Propia.

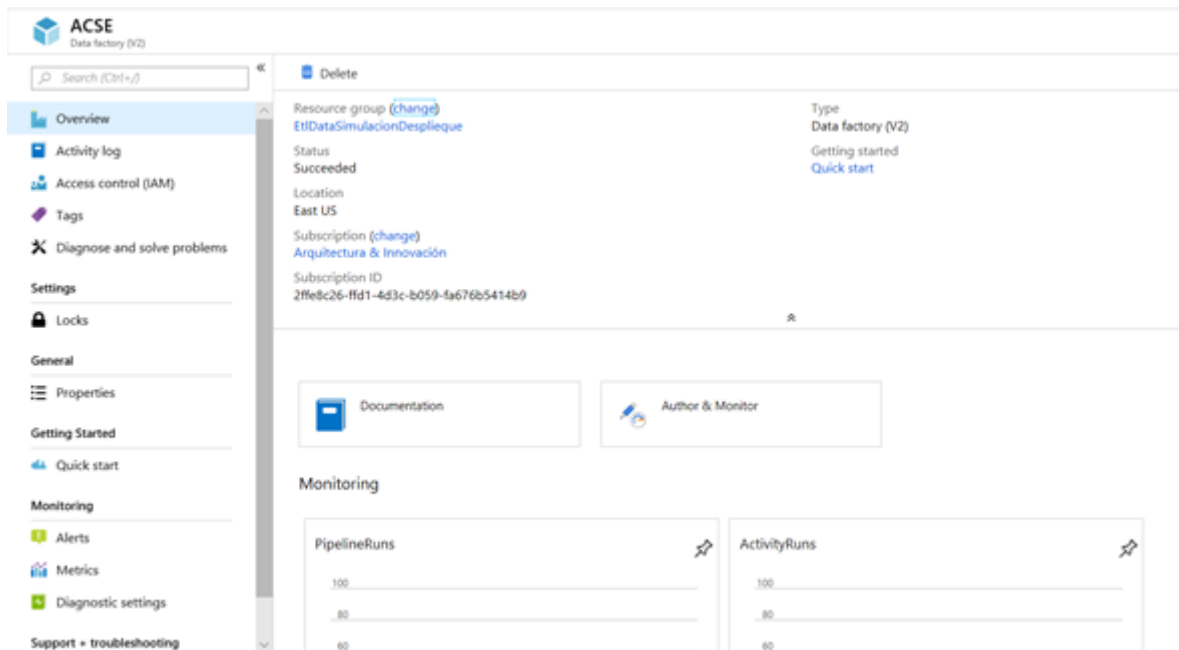


Figura 40. Se muestra el resumen del Data Factory creado. **Fuente:** Propia.

3.9.1 Estructura Data Factory (ETL)

La estructura del data factory está compuesta por pipelines los cuales agrupan de manera lógica las actividades que manipulan los diferentes datasets que representan los datos en los linked services, los cuales que permiten la conexión con las bases de datos y al mismo tiempo consume un Integration Runtime el cual debe relacionarse a una máquina virtual.

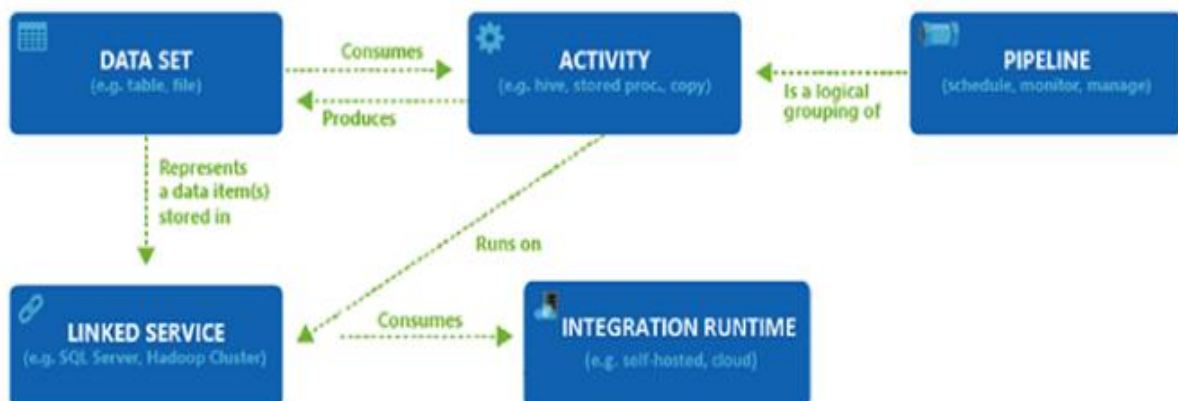


Figura 41. Modelo que presenta la estructura del Data Factory, en esta se muestran todos los componentes necesarios para poder ejecutar un Data Factory. **Fuente:** (“Lo S. (2018). Datasets and linked services in Azure data Factory”).

3.9.2 Crear Linked Services

Los Linked Services son un conjunto de datos que representa la estructura de las bases de datos vinculados, y el servicio vinculado define la conexión a la fuente de datos, este también define la información de la conexión necesaria para que Data Factory se conecte a recursos externos.

Debajo se muestran los scripts necesarios para crear los Linked Services necesarios para la implementación del Azure Data Factory, para este script se utilizará el lenguaje JSON.

Staging Area

```
{
  "name": "StagingAreaLinkedService",
  "type": "Microsoft.DataFactory/factories/linkedservices",
  "properties": {
    "type": "AzureSqlDatabase",
    "typeProperties": {
      "connectionString": {
        "type": "AzureKeyVaultSecret",
        "store": {
          "referenceName": "BaseDeDatosKeyVault"
          "type": "LinkedServiceReference"
        },
        "secretName": "StagingConnectionString"
      }
    },
    "connectVia": {
      "referenceName": "DfIntegrationRuntime",
      "type": "IntegrationRuntimeReference"
    }
  }
}
```

DataMart

```
{
  "name": "DataMartLinkedService",
  "type": "Microsoft.DataFactory/factories/linkedservices",
  "properties": {
    "type": "AzureSqlDatabase",
    "typeProperties": {
      "connectionString": {
        "type": "AzureKeyVaultSecret",
        "store": {
          "referenceName": "BaseDeDatosKeyVault"
          "type": "LinkedServiceReference"
        },
        "secretName": "DataMartConnectionString"
      }
    },
    "connectVia": {
      "referenceName": "DfIntegrationRuntime",
      "type": "IntegrationRuntimeReference"
    }
  }
}
```

Base de Datos Origen

```
{
  "name": "BdOrigenLinkedService",
  "type": "Microsoft.DataFactory/factories/linkedservices",
  "properties": {
    "type": "SqlServer",
    "typeProperties": {
      "connectionString": {
        "type": "AzureKeyVaultSecret",
        "store": {
          "referenceName": "BaseDeDatosKeyVault"

```

```

        "type": "LinkedServiceReference"
      },
      "secretName": "OrigenConnectionString"
    }
  },
  "connectVia": {
    "referenceName": "DfIntegrationRuntime",
    "type": "IntegrationRuntimeReference"
  }
}
}

```

3.9.3 Crear Datasets

“Un Dataset es una vista con nombre de datos que simplemente apunta o hace referencia a los datos que desea utilizar en sus actividades como entradas y salidas. Los conjuntos de datos identifican datos dentro de diferentes almacenes de datos, como tablas, archivos, carpetas y documentos.” (Microsoft, 2018).

Para nuestra solución se deben de crear tres datasets los cuales representarán los datos correspondientes a las bases de datos creadas mediante los linked services de cara al Data Factory.

Debajo se muestran los scripts necesarios para crear los Datasets necesarios para la implementación del Azure Data Factory, para este script se utilizará el lenguaje JSON.

Base de Datos de Origen

```

{
  "name": "OrigenDataset",
  "properties": {
    "type": "SqlServerTable",
    "linkedServiceName": {
      "referenceName": "BdOrigenLinkedService",
      "type": "LinkedServiceReference"
    }
  }
}

```



```

    },
    "typeProperties": {
        "tableName": "CACARA",
        "tableName": "CACONT",
        "tableName": "CAESCA",
        "tableName": "CASOLI",
        "tableName": "CACAMB",
        "tableName": "CAPRIO",
        "tableName": "CACATE",
    }
}
}

```

AZURE - STAGING AREA

```

{
    "name": "StagingDataset",
    "properties":
    {
        "type": "AzureSql",
        "linkedServiceName": {
            "referenceName": "StagingAreaLinkedService",
            "type": "LinkedServiceReference"
        },
        "typeProperties": {
            "tableName": "CACARA",
            "tableName": "CACONT",
            "tableName": "CAESCA",
            "tableName": "CASOLI",
            "tableName": "CACAMB",
            "tableName": "CAPRIO",
            "tableName": "CACATE",
        }
    }
}

```

```
    }  
  }  
}
```

AZURE - DATA MART

```
{  
  "name": "DataMartDataset",  
  "properties":  
  {  
    "type": "AzureSql",  
    "linkedServiceName": {  
      "referenceName": "DataMartLinkedService",  
      "type": "LinkedServiceReference"  
    },  
    "typeProperties": {  
      "tableName": "CACARA",  
      "tableName": "CACONT",  
      "tableName": "CAESCA",  
      "tableName": "CASOLI",  
      "tableName": "CACAMB",  
      "tableName": "CAPRIO",  
      "tableName": "CACATE",  
    }  
  }  
}
```

3.9.4 Crear pipelines

Es una agrupación lógica de actividades que realiza una unidad de trabajo. Juntas, las actividades en un pipeline realizan una tarea.

El beneficio de los pipelines radica en la capacidad de administrar las actividades como un conjunto en lugar de administrar cada una individualmente. Las actividades en un pipeline se pueden encadenar para operar secuencialmente, o pueden operar independientemente en paralelo.

3.9.4.1 Pipeline: Crear Staging Area

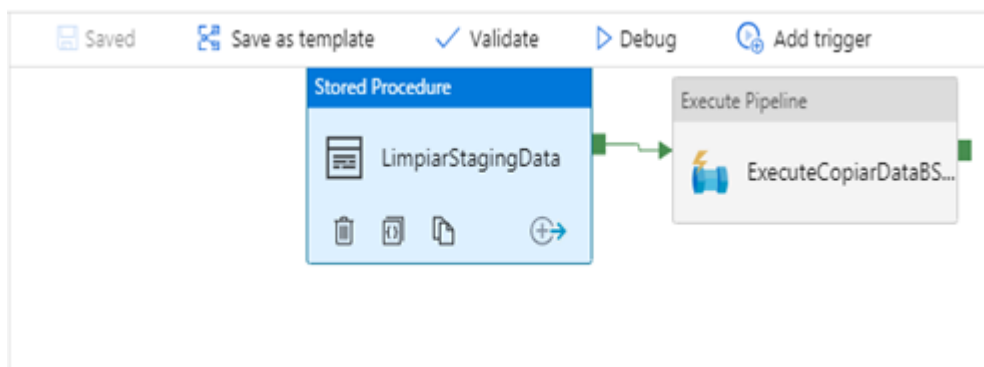


Figura 42. Esquema de actividades del pipeline “Crear StagingArea”, este modelo muestra las actividades y el tipo de actividades que ejecutará el pipeline “Crear StagingArea”. **Fuente:** Propia.

3.9.4.1.1 Actividades del Pipeline “Crear Staging Area”

Limpiar Staging: Esta actividad se encargará de limpiar la base de datos llamada “Staging Area”, esto se realizará cada vez se inicie la ejecución del Data Factory. El objetivo de esto es no tener datos repetidos en nuestra base de datos.

Debajo se muestra el script que se utilizará seleccionar los registros deseados.

```
TRUNCATE table CACARA;  
TRUNCATE table CACONT;  
TRUNCATE table CAESCA;  
TRUNCATE table CASOLI;  
TRUNCATE table CACAMB;  
TRUNCATE table CAPRIO;  
TRUNCATE table CACATE;
```

Execute pipeline:

Esta actividad da inicio al proceso de copiado de los datos que se encuentra en la base de datos de Origen a la base de datos “Staging Area”.

3.9.4.2 Pipeline: Copiar Data Origen a Staging

El objetivo de este pipeline es el de copiar las tablas deseadas desde la base de datos de origen a la base de datos llamada "Staging Area".

Código para crear el pipeline

```
{
  "name": "ExecuteCopyPipeline",
  "properties": {
    "description": "Copiar datos desde Base de Datos Origen
(SQL) a Staging Area (Azure SQL)",
    "activities": [
      {
        "name": "CopyFromOrigenToAzure",
        "type": "Copy",
        "inputs": [
          {
            "name": "InputDataset"
          }
        ],
        "outputs": [
          {
            "name": "OutputDataset"
          }
        ],
        "typeProperties": {
          "source": {
            "type": "SQL"
          },
          "sink": {
            "type": "SqlSink",
            "writeBatchSize": 10000,
            "writeBatchTimeout": "60:00:00"
          }
        }
      }
    ]
  }
}
```

```
    },  
    "policy": {  
      "retry": 2,  
      "timeout": "01:00:00"  
    }  
  }  
]  
}  
}
```

3.9.4.2.1 Actividades del Pipeline “Copiar Data Origen a Staging”

CopyData

El siguiente esquema muestra las tablas que se copiarán desde la base de datos de Origen a la base de datos “Staging Area” mediante la actividad “Copy Data”, esta actividad es propia de Microsoft Azure y permite la copia de todos los registros que contiene una tabla a otra.

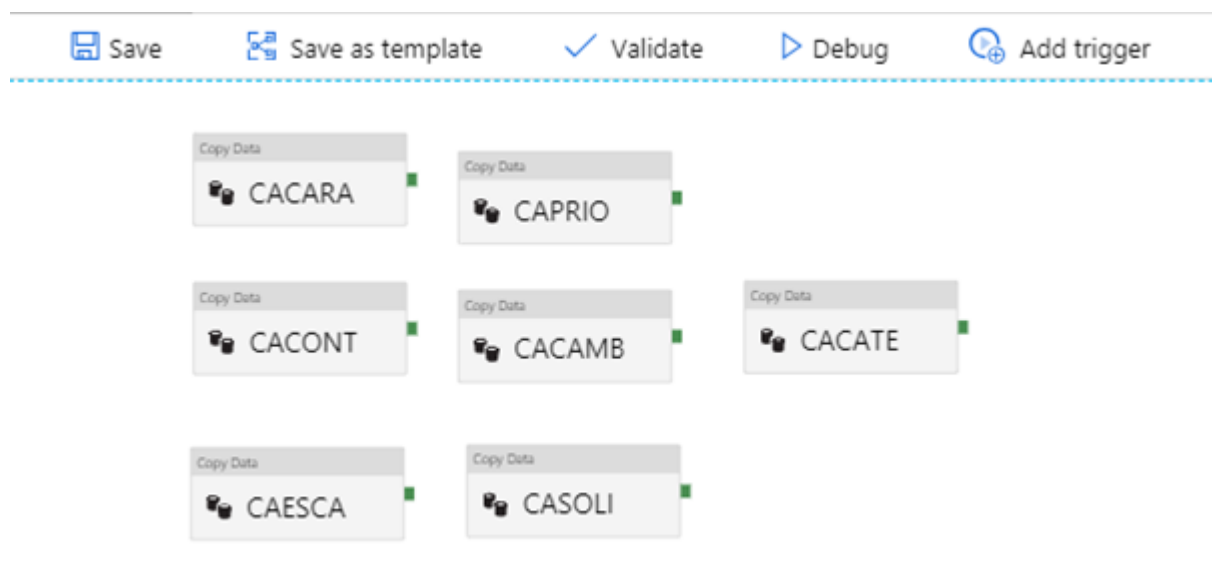


Figura 43. Esquema de actividades del pipeline “Copiar Data Origen a Staging”, este modelo muestra las actividades y el tipo de actividades que ejecutará el pipeline “Copiar Data Origen a Staging”. **Fuente:** Propia.

3.9.4.3 Pipeline: Crear Data mart

El siguiente esquema muestra las actividades que formarán parte de este pipeline, el cual tiene como objetivo transformar y cargar los datos desde la base de datos “Staging Area” a la base de datos “DataMart”. Una vez modelados estos datos, estarán listos para ser consumidos por los reportes de BI.

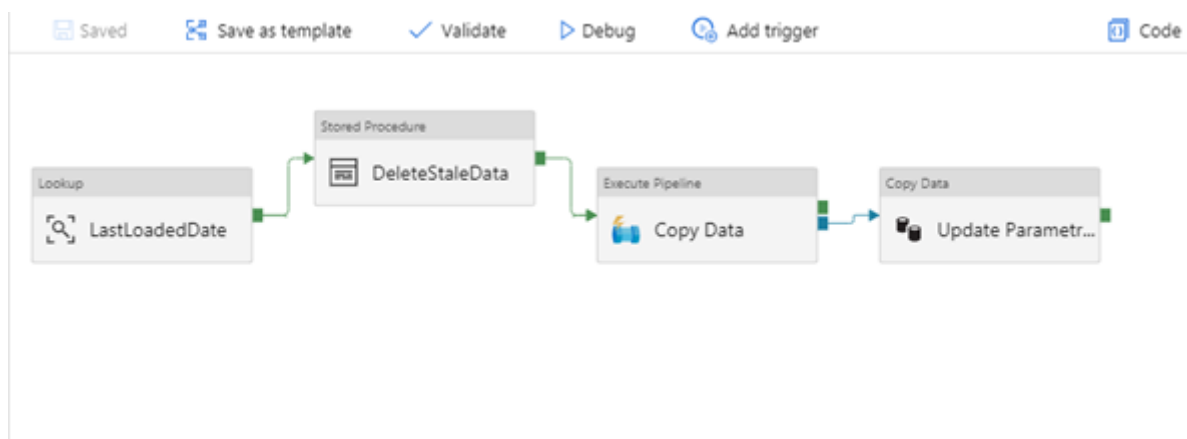


Figura 44. Esquema de actividades del pipeline “Crear DataMart”, este modelo muestra las actividades y el tipo de actividades que ejecutará el pipeline “Crear DataMart”. **Fuente:** Propia.

3.9.4.3.1 Actividades del Pipeline “Crear Data mart”

Actividad LastLoadedDate:

La función de este proceso es el de identificar los registros que fueron insertados por última vez en la base de datos “Staging Area”, esto con el objetivo de agilizar el proceso de carga que se realiza desde la base de datos “Staging Area” a “DataMart”.

Debajo se muestra el script que se utilizará seleccionar los registros deseados.

```

SELECT
CODPARAM, DESCRIPCION,
        FECHAINI, FECHAFIN,
        VALUEPARAM, CODCIAAPLIC
FROM BS_PARAMETRO WHERE descripcion = 'BI_LOAD_SA';

```

Actividad DeleteStaleData

Stale data es información la cual se encuentra en una base de datos y no refleja la realidad, esto supone un problema a la hora de ejecutar el data factory debido a que podría afectar las operaciones que se desea realizar debido a la falta de integridad de los datos.

El siguiente script valida los registros que fueron cargados en la última ejecución del data factory:

```

@activity('LastLoadedDate').output.firstRow.FECHAINI
@activity('LastLoadedDate').output.firstRow.FECHAFIN

```

Actividad CopyData

Este proceso modela mediante un script los registros que se cargarán desde la base de datos “Staging Area” a “DataMart”. Debajo se muestra un modelo del script:

```

SELECT
SOL.REF_NUM as Solicitud,
SOL.SUMMARY as Subject,
SOL.DESCRPTION as Descripcion,
DateAdd(SECOND,OPEN_DATE,convert(datetime,          '1969-12-31
20:00:00.000', 121)) AS Fecha_Inicial,
DateAdd(SECOND,(CLOSE_DATE),convert(datetime,       '1969-12-31
20:00:00.000', 121)) AS Fecha_Termino,
GR.LAST_NAME AS Grupos,
pr.sym as Impacto,

```

```

ISNULL(ASIG.FIRST_NAME+ ' ' + ISNULL(ASIG.middle_NAME, ' ')+
' ' + ASIG.LAST_NAME,'SIN ASIGNAR') AS Usu_Asignado,
ISNULL(UF.FIRST_NAME+ ' ' + ISNULL(UF.middle_NAME, ' ')+ ' '
+ UF.LAST_NAME,'SIN ASIGNAR') AS Usu_Final,
ISNULL(REP.FIRST_NAME+ ' ' + ISNULL(REP.middle_NAME, ' ') +
' ' + REP.LAST_NAME,'SIN ASIGNAR') AS Usu_Reporto,
ISNULL(CAT.sym,'SIN ASIGNAR') AS Request_Area,
ST.SYM AS Status,
ISNULL(root.sym,' ') AS Root,
(SELECT chg.chg_ref_num FROM CACAMB CHG WHERE SOL.CHANGE=
CHG.ID) AS Change_Order,
ISNULL(SOL.SEVERITY,0) AS Complejidad,
ISNULL(SOL.Urgency,0) AS Prioridad,
SOL.ACTIVE_FLAG as Indicador_Activo
FROM (((((((
CASOLI      SOL
LEFT OUTER JOIN CACATE  PRO  ON SOL.CATEGORY = CAT.PERSID)
LEFT  OUTER  JOIN  CACONT  UF      ON  SOL.CUSTOMER      =
UF.CONTACT_UUID)
LEFT  OUTER  JOIN  CACONT  REP      ON  SOL.LOG_AGENT     =
REP.CONTACT_UUID)
LEFT  OUTER  JOIN  CACONT  ASIG     ON  SOL.ASSIGNEE     =
ASIG.CONTACT_UUID)
LEFT  OUTER  JOIN  CACONT  GR      ON  SOL.group_id      =
GR.CONTACT_UUID)
LEFT OUTER JOIN CACARA ROOT ON SOL.rootcause = ROOT.ID)
LEFT OUTER JOIN CAPRIO      PR  ON SOL.priority = PR.ENUM )
LEFT OUTER JOIN CAESCA  ST  ON SOL.status = ST.code)
WHERE
year(DateAdd(SECOND,OPEN_DATE,convert(datetime, '1969-12-31
20:00:00.000', 121))) >= year(getdate ())

```

Actividad Update Parametro adic

Se actualiza este campo con la fecha de carga de los registros, para que en la siguiente corrida se pueda validar cuales fueron los registros insertados por última vez.

```
update parametro adic
update BS_PARAMETRO
    set FECHAINI = FECHAFIN - VALUEPARAM,
        FECHAFIN = trunc(sysdate)
where codparam = 'BI_LOAD_SA';
```

3.9.5 Integration Runtime en el Data Factory

Integration Runtime auto hospedado puede ejecutar actividades de copia entre un almacén de datos en la nube y un almacén de datos en una red privada, y puede enviar actividades de transformación contra recursos informáticos en una red local o en una red virtual de Azure.

Es necesario crear un Integration Runtime en el data factory debido a que este permite el cálculo nativo utilizado por ADF para ejecutar o enviar las actividades definidas en el data factory. Este Integration Runtime debe ser de tipo Realizar actividades de movimiento y despacho de datos a cómputos externos.

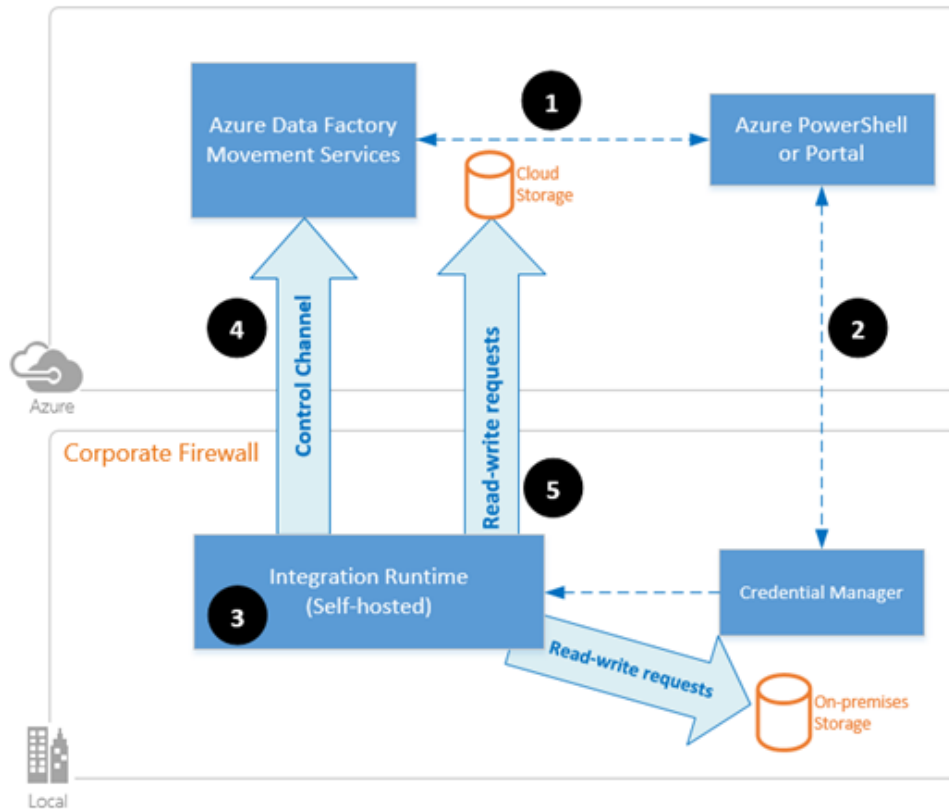


Figura 45. En este esquema se muestra la arquitectura del Integration Runtime y cómo se relaciona con la infraestructura local de cualquier organización y con los demás componentes de Microsoft Azure. 1 - El usuario crea un tiempo de ejecución de integración auto hospedado dentro de una fábrica de datos de Azure. 2 - El usuario crea un servicio vinculado para un almacén de datos local. 3 - El nodo de tiempo de ejecución de integración auto hospedado encripta las credenciales. 4 - El servicio Data Factory se comunica con el tiempo para la programación y administración de trabajos a través de un canal de control que utiliza una cola compartida de Azure Service Bus. 5 - El tiempo de ejecución de integración auto hospedado copia los datos de un almacén local a un almacenamiento en la nube, o viceversa, según cómo se configure la actividad de copia en la canalización de datos. **Fuente:** Propia.

Se realizó una prueba creando una un Integration Runtime en el Data Factory. A continuación, se muestra el proceso de creación y configuración:

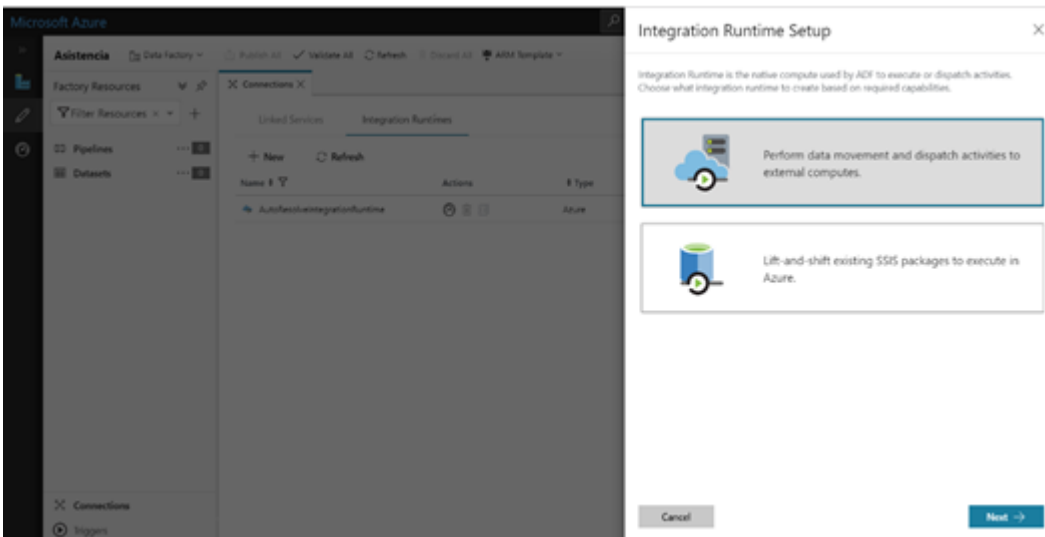


Figura 46. Se inicia el proceso de creación del Integration Runtime y se selecciona la opción “Perform data movement and dispatch activities to external computes” que se traduce a: Realizar movimientos de datos y despachar actividades a cómputos externos. **Fuente:** Propia.

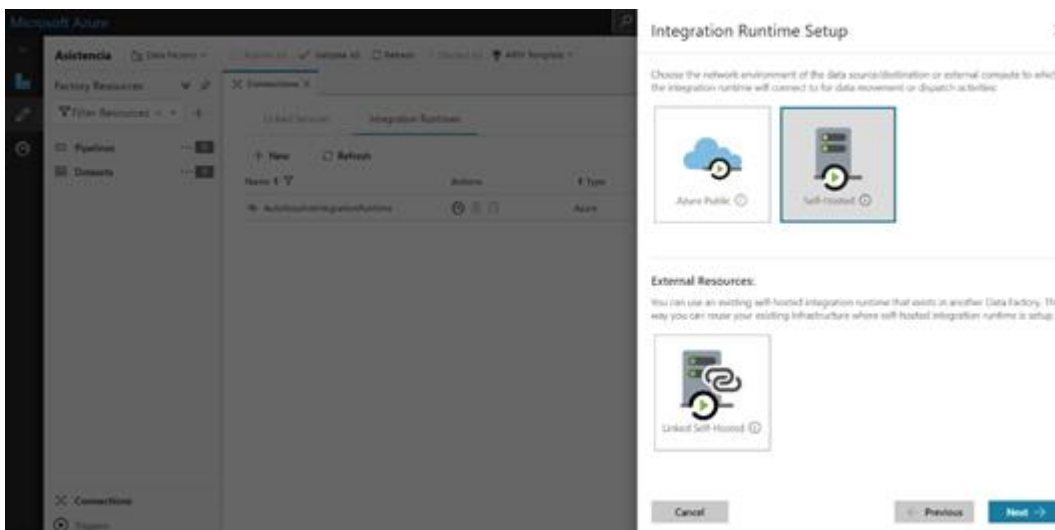


Figura 47. Se debe de seleccionar el tipo Self-Hosted, esto permitirá ejecutar actividades de copia entre la base de datos de origen y las bases de datos creadas en la plataforma de Azure. **Fuente:** Propia.

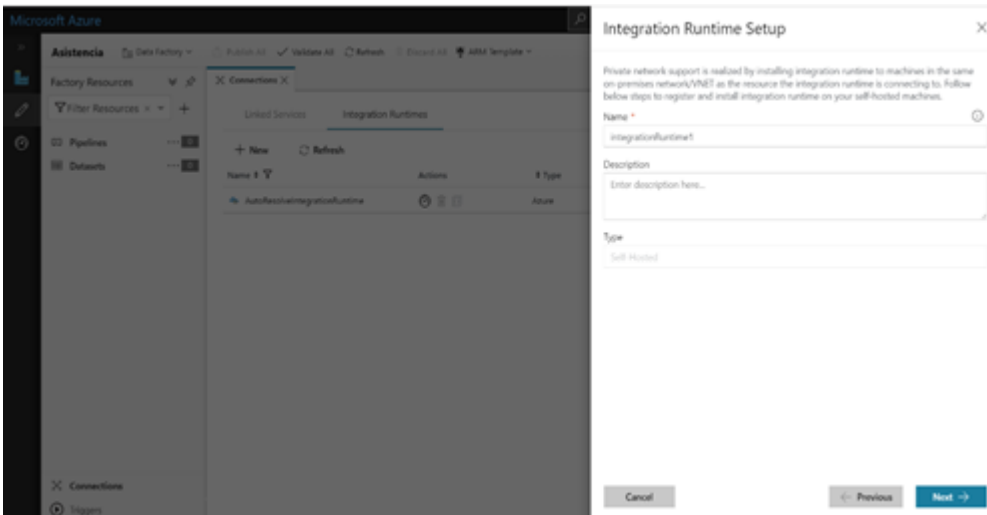


Figura 48. Se configura el Integration Runtime y se completan los campos: Nombre y descripción. **Fuente:** Propia.

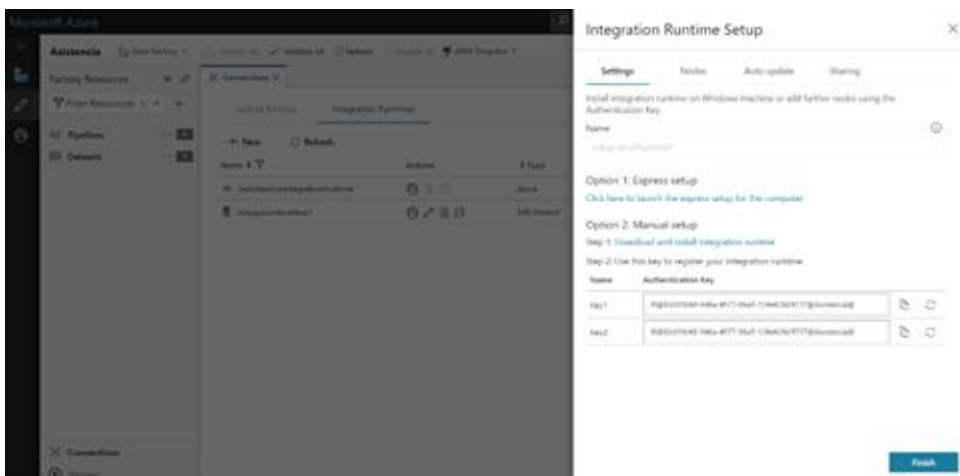



Figura 49. Se debe de copiar el Key1 de manera que se pueda utilizar en el Integration Runtime instalado en la máquina virtual. **Fuente:** Propia.

Register Integration Runtime (Self-hosted)

Welcome to Microsoft Integration Runtime Configuration Manager. Before you start, register your Integration Runtime (Self-hosted) node using a valid Authentication Key.

Show Authentication Key

[Learn how to find the Authentication Key](#)

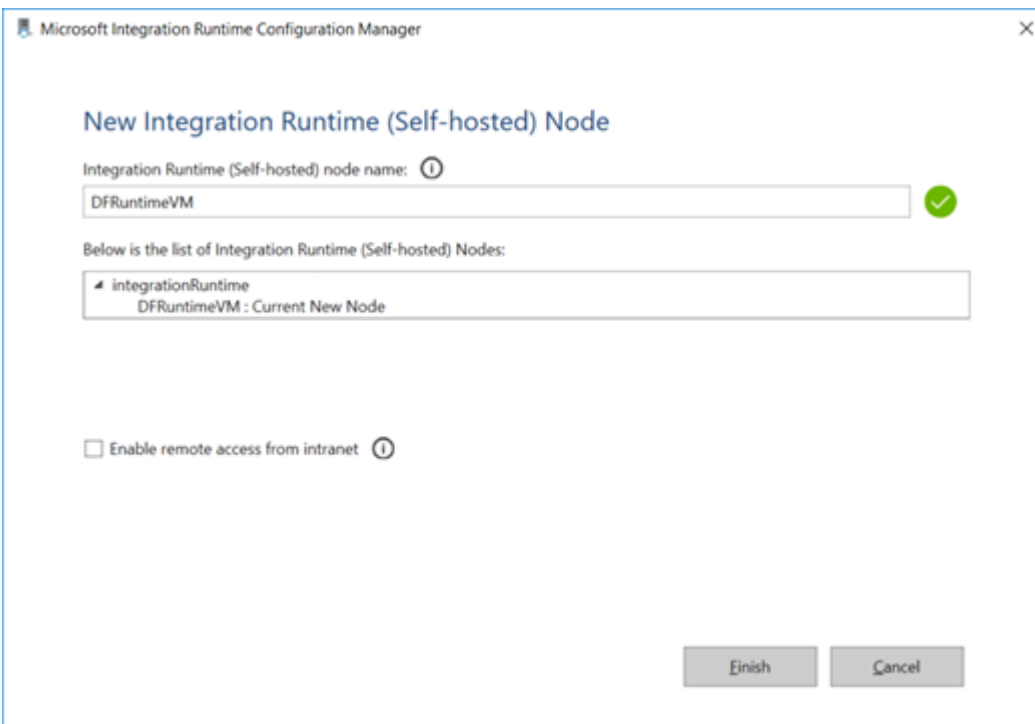
HTTP Proxy

Current Proxy: No proxy [Change](#)

Register

Cancel


Figura 50. En este paso se debe de volver al Integration Runtime de la máquina virtual y se ingresa el Key1 copiado desde el data factory, con el objetivo de relacionar el data Factory con la máquina virtual, esto permitirá el movimiento de los registros almacenados en las bases de datos. **Fuente:** Propia.



Microsoft Integration Runtime Configuration Manager X

New Integration Runtime (Self-hosted) Node

Integration Runtime (Self-hosted) node name: ⓘ

Below is the list of Integration Runtime (Self-hosted) Nodes:

- integrationRuntime
 - DFRuntimeVM : Current New Node

Enable remote access from intranet ⓘ

Finish Cancel

Figura 51. En este paso se procede a finalizar el el proceso de configuración del Integration Runtime. **Fuente:** Propia.

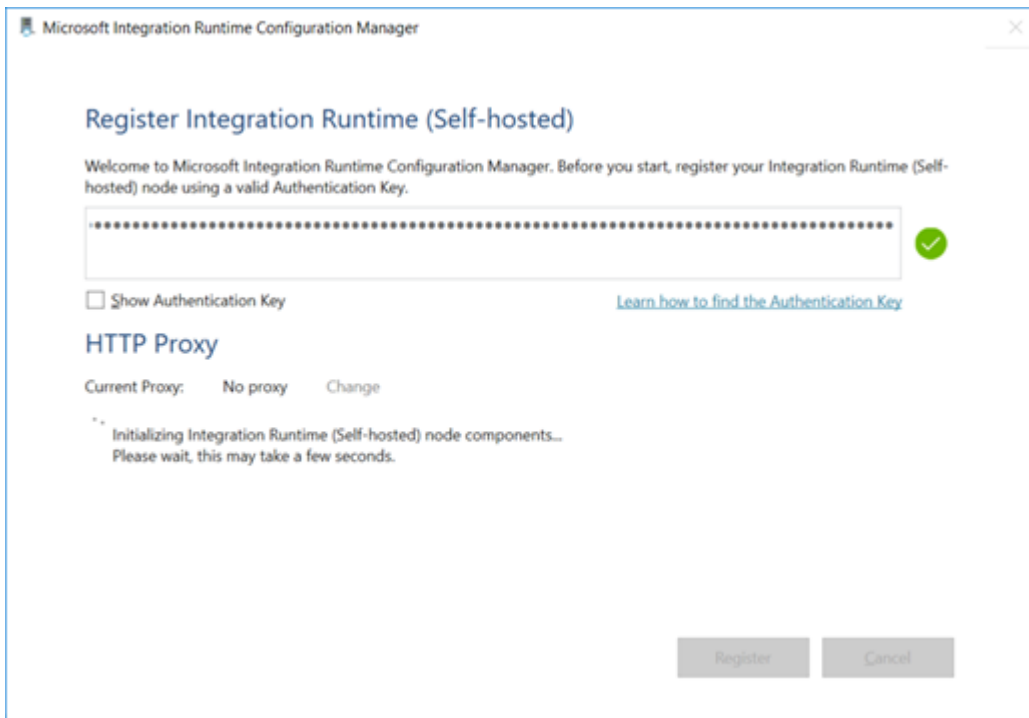


Figura 52. Se debe de validar que el Integration Runtime se haya inicializado correctamente. **Fuente:** Propia.

3.9.6 Calendarizar la ejecución del Data Factory

Se deben de calendarizar las ejecuciones del Data Factory, esto se realiza mediante un trigger el cual iniciará las ejecuciones. Este deberá ser configurado de manera que se las ejecuciones se realicen tres veces al día, todos los días.

A continuación, se muestra la creación y configuración de un trigger en el ambiente de Microsoft Azure:

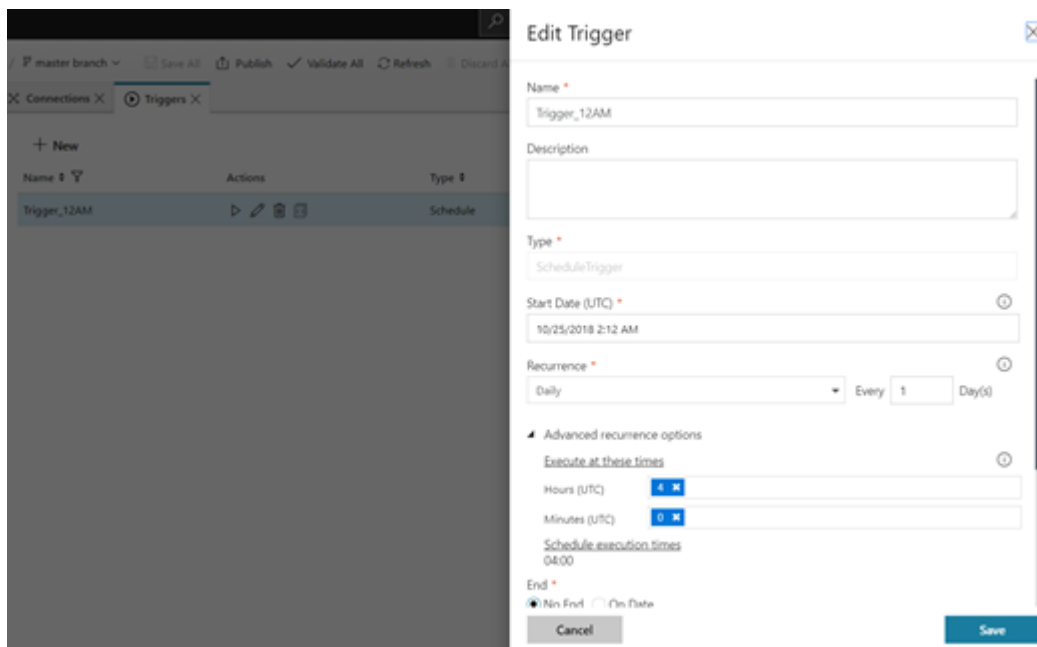


Figura 53. Se procede a crear y configurar el trigger que iniciara la ejecución del datafactory, se completan los campos: nombre del trigger, descripción, fecha de inicio y la recurrencia. **Fuente:** Propia.

3.10 Visualización de Datos

El objetivo final de una solución de inteligencia de negocios es la de mostrar información deseada por el usuario o en este caso área, por lo que se debe de crear reportes para este fin. La herramienta de visualización que se utilizará será Power BI.

Power BI es una solución de análisis empresarial que permite visualizar los datos y compartir información con toda la organización, o insertarla en su aplicación o sitio web. (Microsoft, 2016).

Las razones por la cual se eligió Power BI es debido a la integración existente entre microsoft azure y microsoft power bi, además de esto permite compartir dashboards y la visualización de los mismos desde la web, sin necesidad de instalar la versión de escritorio de la aplicación en las computadoras de los usuarios.

Para esta solución se crearon varios reportes los cuales sirven para mostrar la información modelada por el ETL. Los mismos se muestran debajo:

A continuación se muestran los reportes:

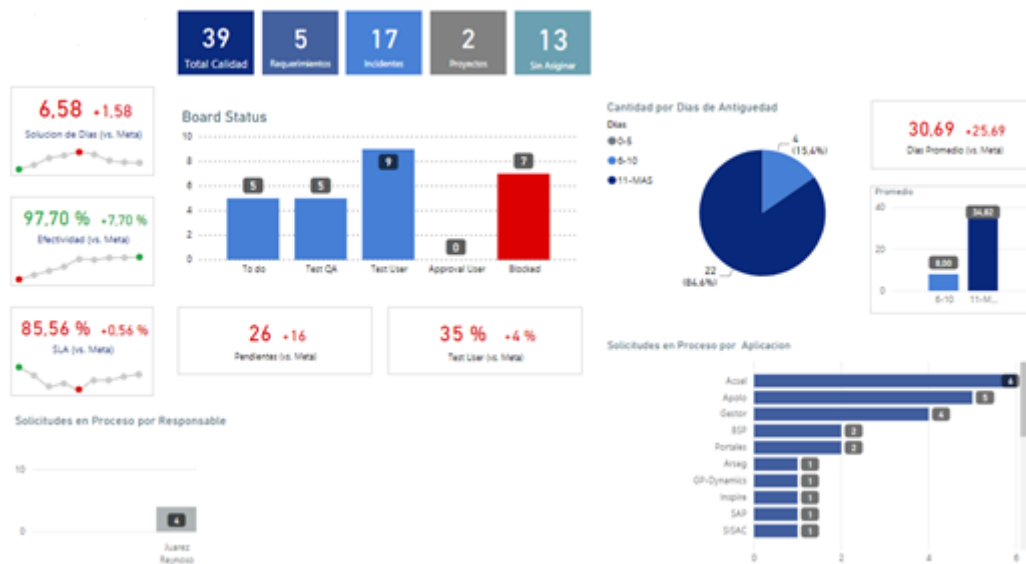


Figura 54. Reporte muestra el total de reportes por estado, aplicaciones y antigüedad. **Fuente:** Propia.



Figura 55. Reporte muestra las solicitudes sin asignar y las bloqueadas. **Fuente:** Propia.

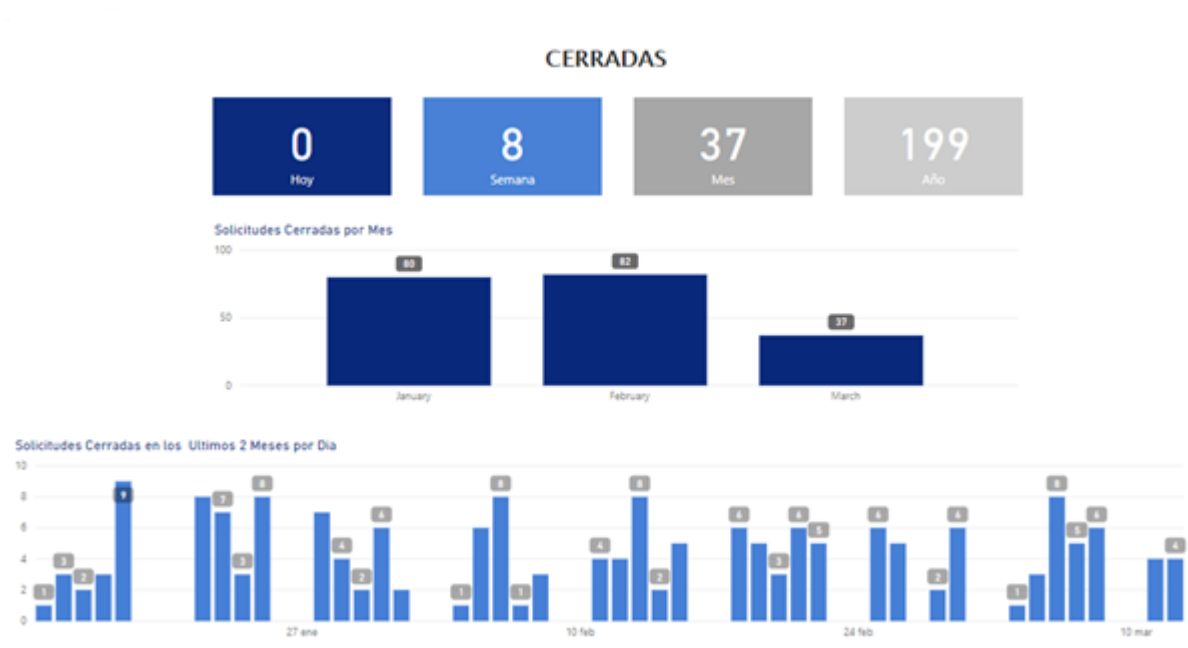


Figura 56. Reporte muestra las solicitudes cerradas por día, semana, mes y año. **Fuente:** Propia.

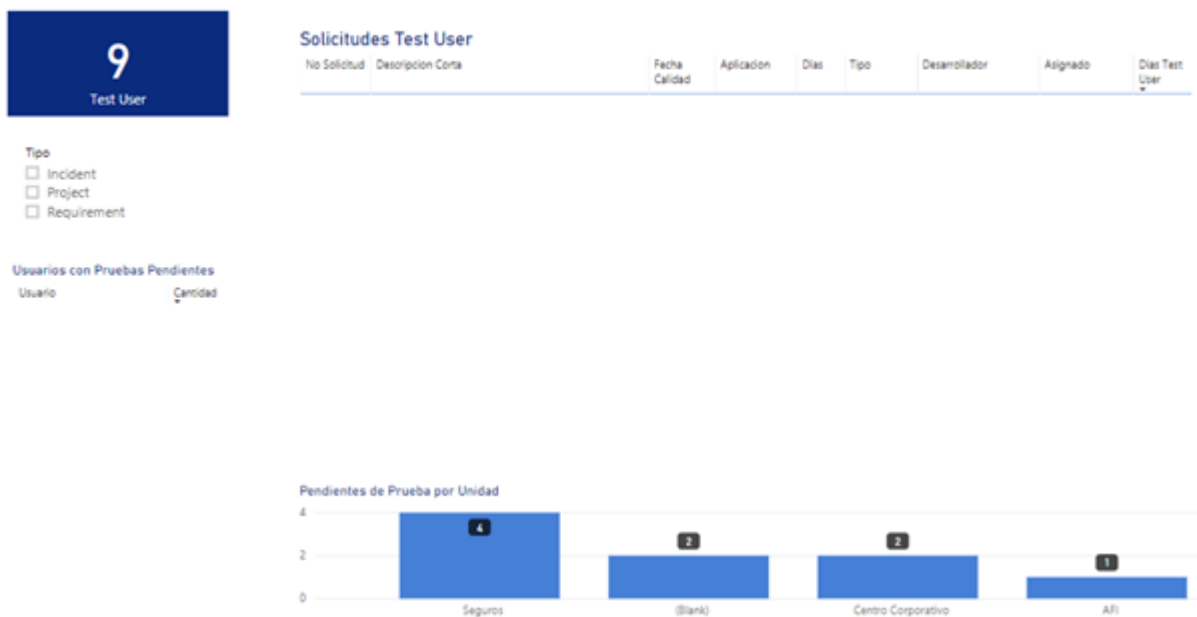


Figura 57. Reporte muestra las solicitudes en estado Test User. **Fuente:** Propia.

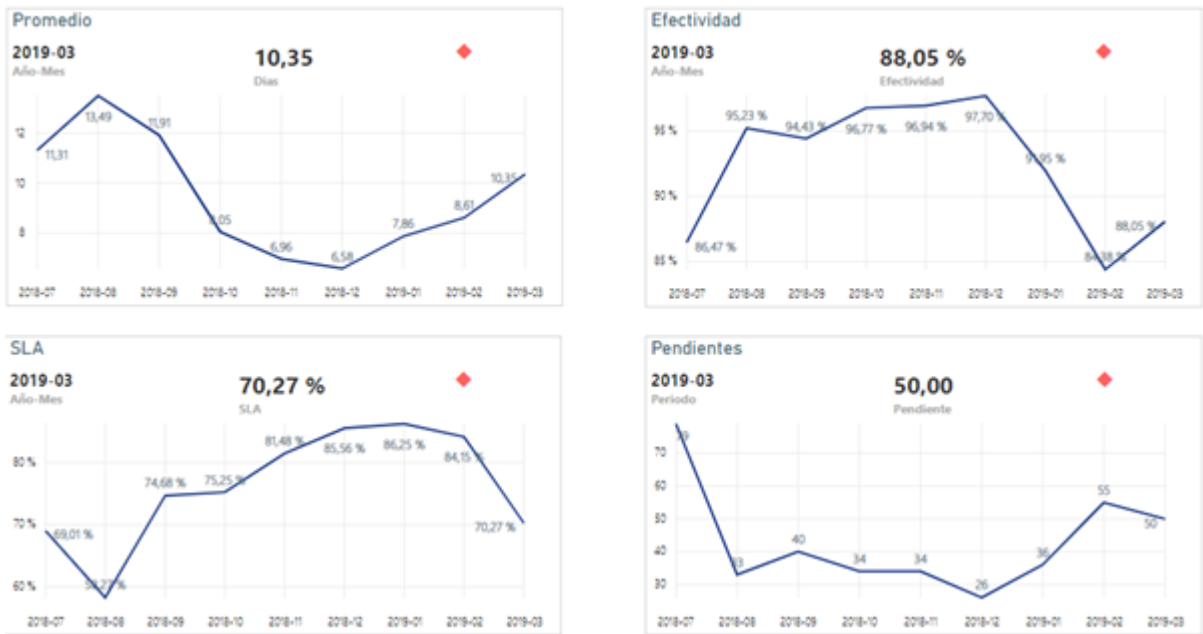


Figura 58. Reporte muestra estadística sobre la efectividad del proceso. Fuente: Propia.

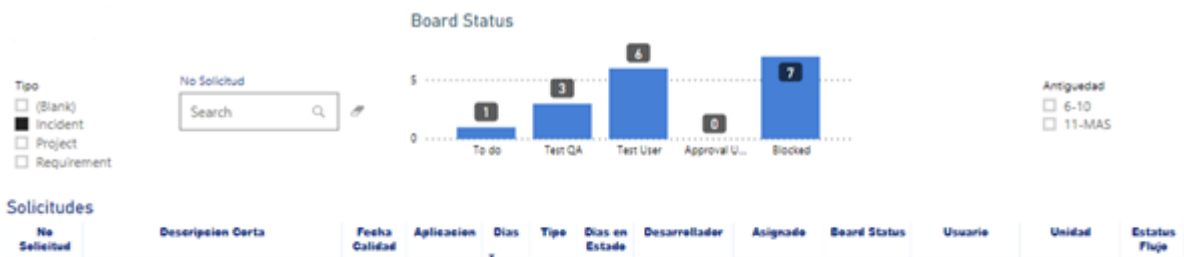


Figura 59. Reporte muestra todas las solicitudes abiertas. Fuente: Propia.

3.10.1 Configuración Power BI

Una vez creados los reportes en la herramienta Power BI, a estos archivos de formato (.pbix) se les deberá configurar la fuente de datos con el objetivo de que estos puedan consumir la información almacenada en la base de datos “Datamart”, la razón por la cual se debe de relacionar los reportes con esta base de datos en especifica es que en esta se almacenan los datos ya modelados y transformados, de manera de que puedan ser consumidos de manera eficiente por los reportes.

A continuación se muestra el proceso de configuración de los reportes:

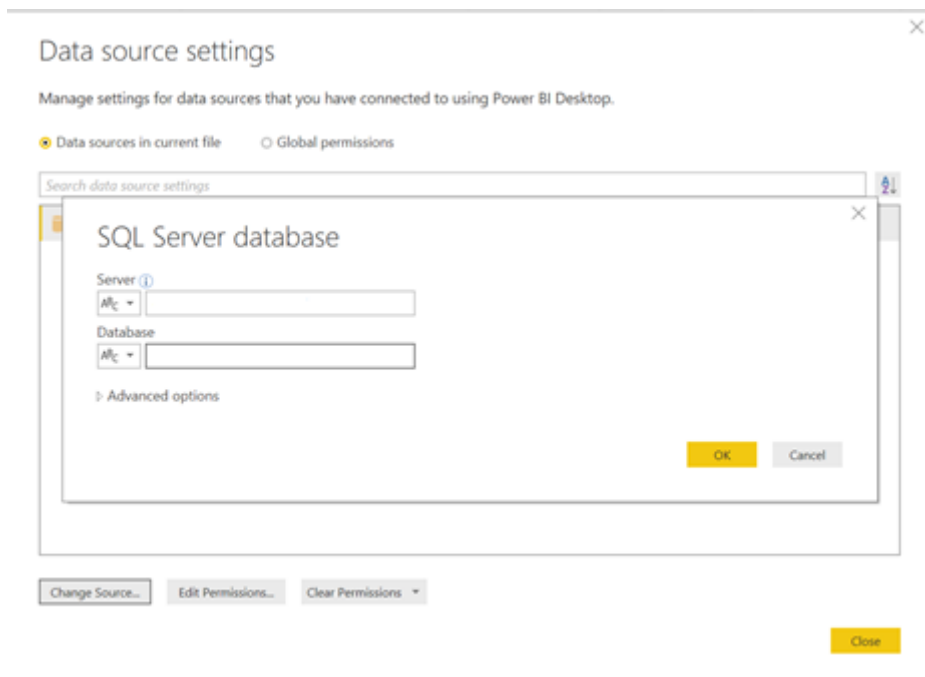


Figura 60. Se debe de configurar los reportes en Power BI, en este pase se debe de ingresar el servidor y base de datos (Datamart) creada en Azure. **Fuente:** Propia.

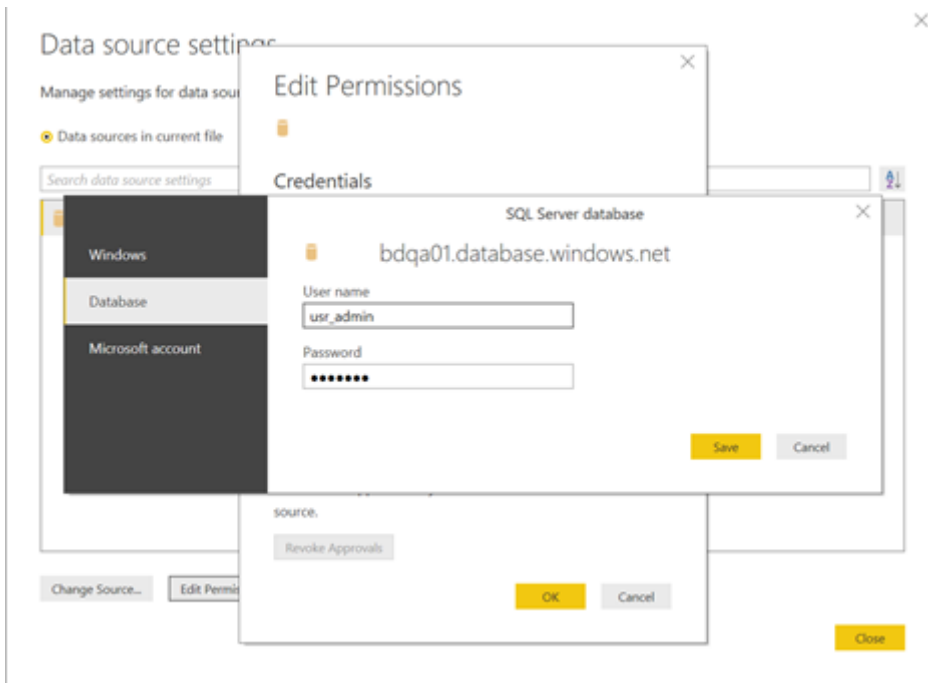


Figura 61. Se ingresan las mismas credenciales que utiliza el servidor lógico SQL, con el objetivo de que los reportes puedan conectarse a la base de datos “Datamart”. **Fuente:** Propia.

3.10.2 Workspace de Microsoft Power BI

Los Workspace son espacios de trabajo donde se publican los reportes y dashboard de Power BI. Dentro de estos espacios se configuran los orígenes de datos, por esto podemos considerar que son los diferentes ambientes que utiliza la empresa para el desarrollo de esta herramienta.

Se debe crear un Workspace BI Gestión de Incidentes. El Objetivo es contener los reportes oficiales respaldados bajo el sistema de versiones.

Para el acceso a los Workspace, se recomienda crear grupo de Usuarios, donde estas personas serán las autorizadas para acceder a los diferentes espacios de trabajo

A continuación, se muestra el proceso necesario para la creación de un Workspace en la plataforma de Power BI:

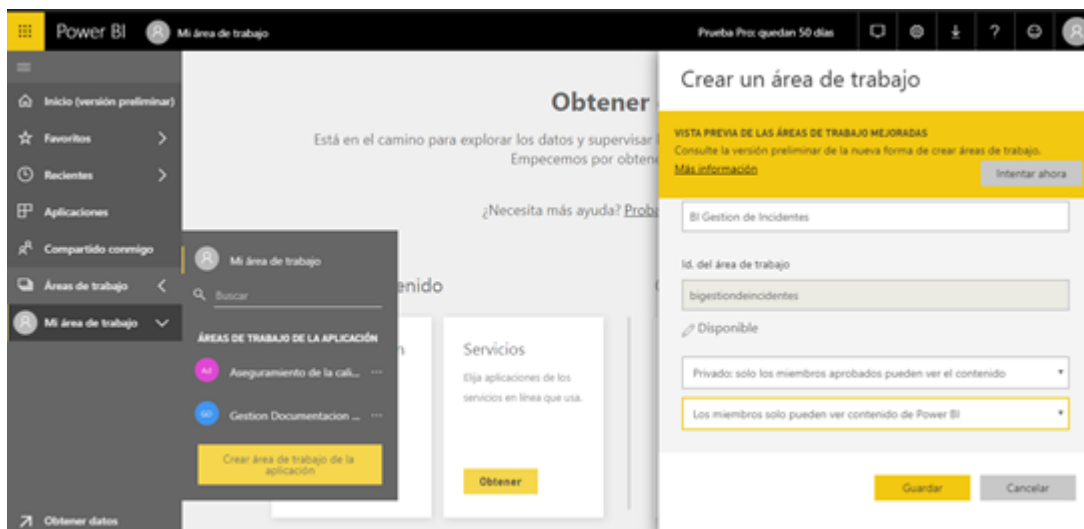


Figura 62. Se procede a colocar el nombre del WorkSpace, se coloca como el Workspace como privado, agregando el grupo que tendrá acceso (Grupo creado para el personal de Operaciones) para Editar el mismo. **Fuente:** Propia.

Se procede a colocar el nombre del WorkSpace, se coloca como el Workspace como privado, agregando el grupo que tendrá acceso (Grupo creado para el personal de Operaciones) para Editar el mismo. Los usuarios con permisos sobre este espacio, tendrán acceso a modificar Data Sources y publicar reportes.

3.10.3 Compartir Reportes

El último paso de una implementación de inteligencia de negocios que los interesados puedan visualizar la información deseada con el fin de tomar decisiones en base a esta y tener un histórico en este caso para poder tener una trazabilidad de los incidentes que llegaron al área de tecnología de Grupo Seguros G.

Para poder compartir estos reportes con los usuarios interesados se debe de realizar el siguiente proceso:

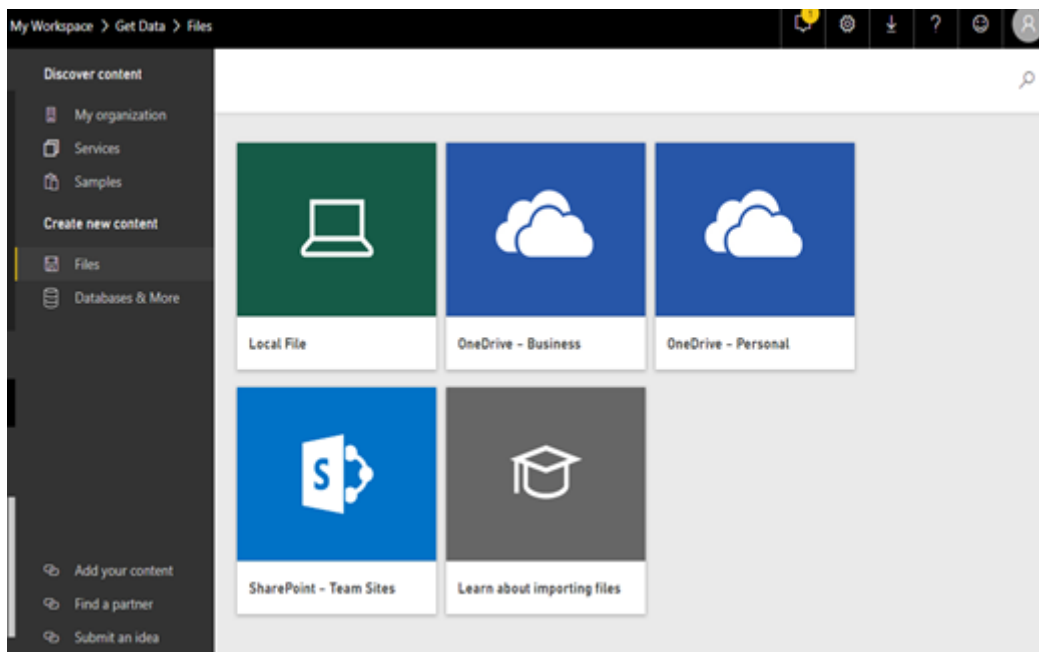


Figura 63. Se debe de subir los reportes creados desde la máquina virtual al WorkSpace creado, en este paso se selecciona la opción “Local File” y se procede a seleccionar los reportes deseados. **Fuente:** Propia.

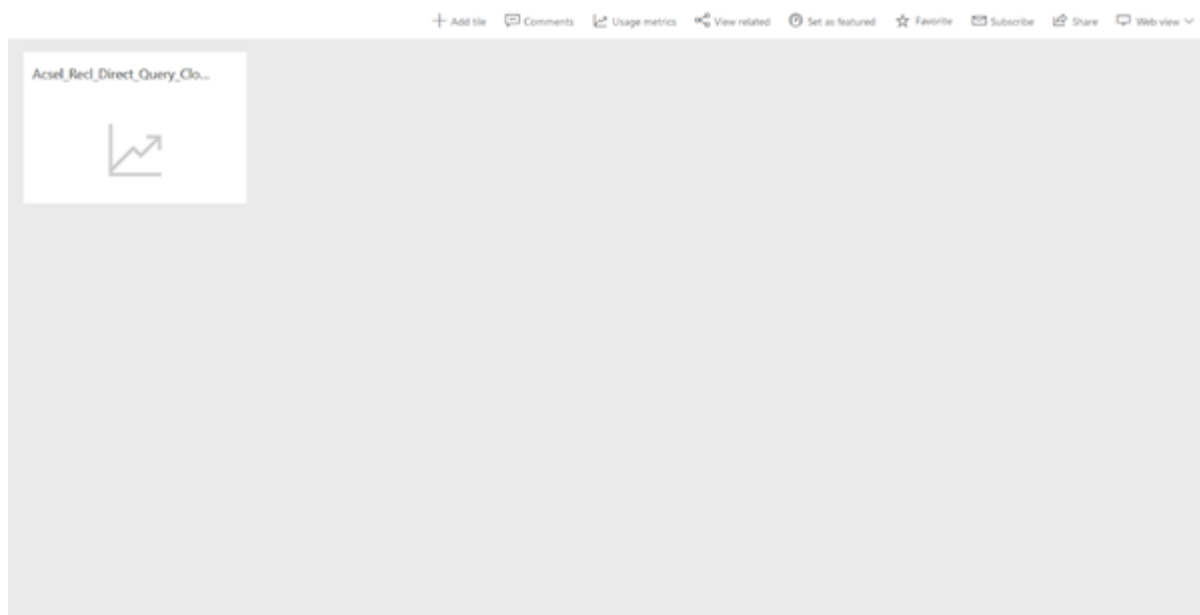


Figura 64. Se debe validar que los reportes fueron cargados correctamente en el WorkSpace creado. **Fuente:** Propia.

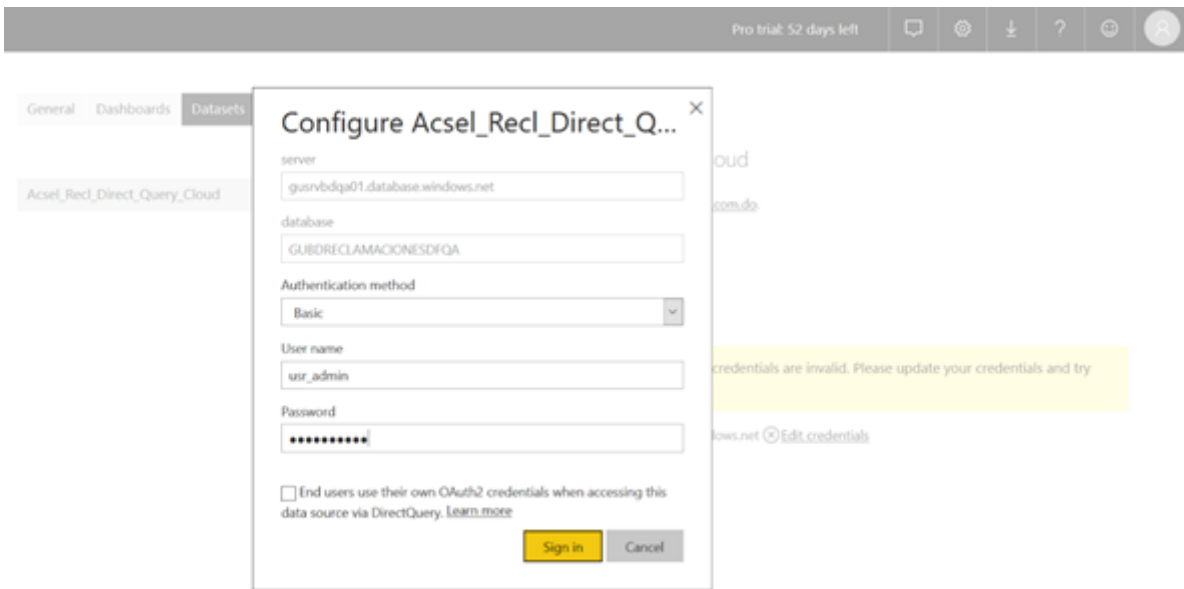


Figura 65. Se debe configurar el dataset que utilizan los reportes con las credenciales mismas credenciales que se indicaron al crear el servidor lógico SQL. **Fuente:** Propia.

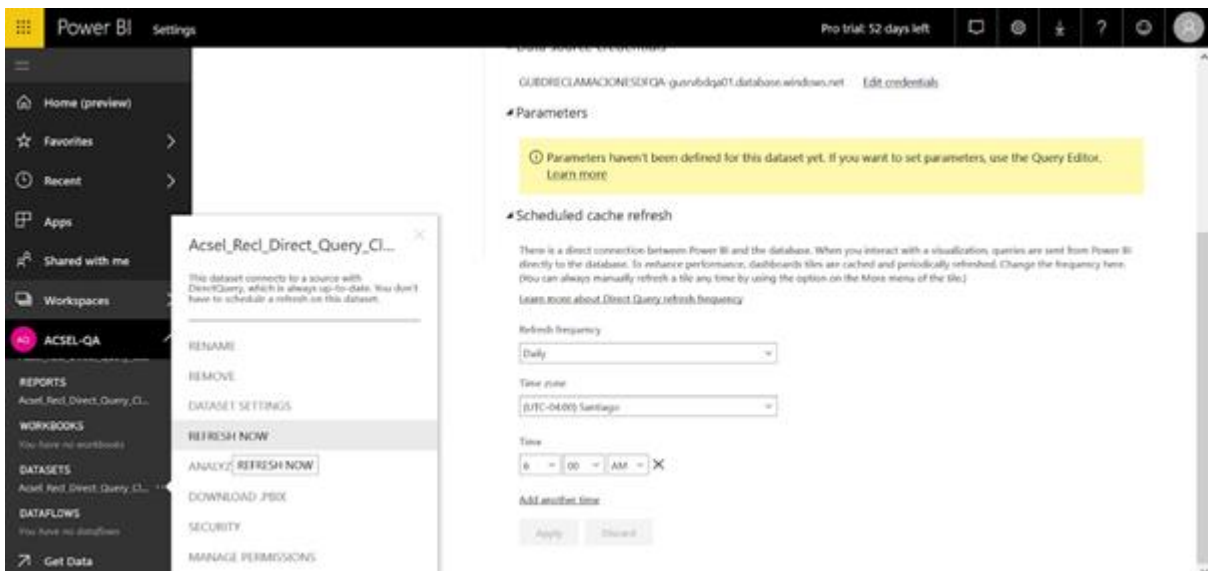


Figura 66. Luego de cargado y configurado, se valida que el reporte puede ser actualizado. **Fuente:** Propia.

3.10.4 Conceder permiso de lectura a usuarios

Para que los usuarios interesados puedan visualizar los reportes cargados a Power BI App, se debe de compartir el dashboard que contendrá cada uno de los reportes. Para poder compartir este dashboard se necesita poseer una licencia de Power BI Pro, aquellos a quienes se le comparta este dashboard también deberán de poseer licencias de Power BI Pro. Una vez compartido los usuarios podrán ver e interactuar con estos con la salvedad de que no podrán editarlo.

A continuación, se muestra el proceso a realizar para compartir el dashboard en la plataforma Power BI:

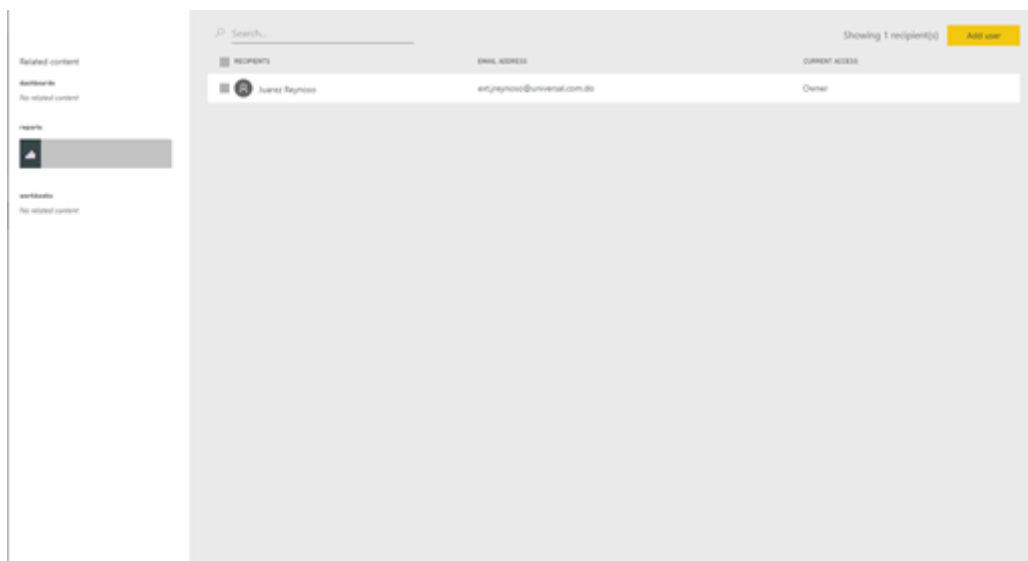


Figura 67. Luego de seleccionar el dashboard deseado, se debe de hacer click en el botón “Add User”. **Fuente:** Propia.

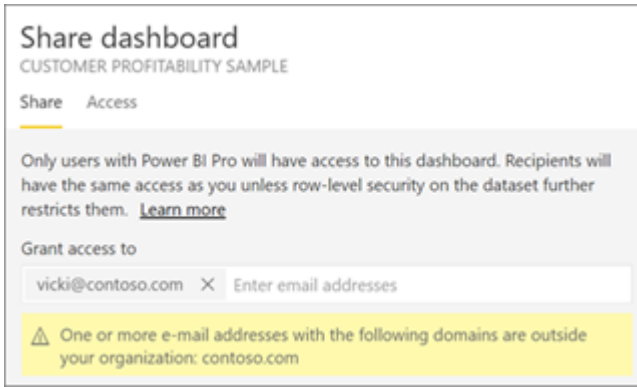


Figura 68. Se debe de ingresar los correos electrónicos de los usuarios a quienes se les van a compartir el dashboard. **Fuente:** Propia.

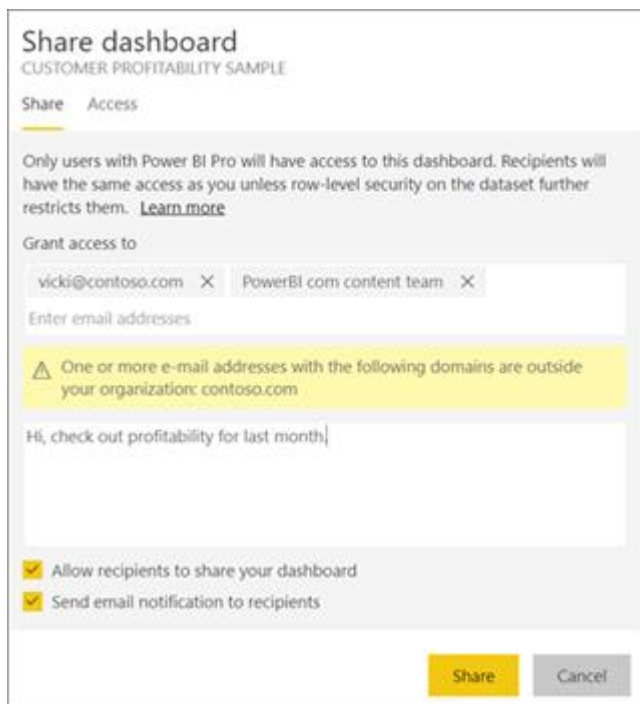


Figura 69. Se procede a ingresar algún comentario que podrá ser visualizado por los usuarios indicados en el correo de notificación. **Fuente:** Propia.

3.11 Costo de implementación

A continuación se muestra el costo de cada uno de los componentes necesarios para implementar esta solución de BI.

Componente/Servicio	Descripción	Costo anual (\$ US)
Máquina Virtual	1 B2S (2 vCPU(s), 4 GB RAM); Windows – (OS Only); 1 year reserved; 0 managed OS disks – E4, 100 transaction units	\$326.52
IP Addresses	1 Dynamic IP Addresses, 1 Static IP Addresses	\$259.20
Azure SQL Database	Single Database, DTU Purchase Model, Premium Tier, P1: 200 DTUs, 250 GB included storage per DB, 1 Database(s) x 24 Hours, 5 GB Retention	\$3,739.44
Azure SQL Database	Single Database, DTU Purchase Model, Premium Tier, P1: 200 DTUs, 250 GB included storage per DB, 1 Database(s) x 24 Hours, 5 GB Retention	\$3,739.44
Data Factory	Azure Data Factory V2 Type, Data Pipeline Service Type, 17 Read/Write operation(s), 1 Monitoring operation(s), Azure Integration Runtime: 1 Activity Run(s), 4 Data movement unit(s), 4 Pipeline activities, 0 Pipeline activities – External; Self-hosted Integration Runtime: 1 Activity Run(s), 4 Data movement unit(s), 4 Pipeline activities, 4 Pipeline activities – External	\$18.259.20

Key Vault	3 operations/mo, 0 advanced operations/mo, 0 renewals/mo, 0 protected keys/mo 0 advanced protected keys/mo	\$0.36
Licencia Power BI Pro por usuario	30 Licencia de Power BI Pro	\$3,596.4
Total		\$29,920.56

Tabla 2. Costo de implementación.

3.12 Resumen Capítulo III

Grupo de Seguros G en busca de un mejor proceso de gestión de incidentes deberá implementar una solución de BI la cual permita conocer el estado de cada una de las solicitudes creadas, para así gestionar de la mejor manera el proceso de gestión de incidentes y así cumplir con los estándares definidos por ITIL.

En este capítulo se indican cuales son los componentes necesarios para la implementación de una solución de BI que satisfaga las necesidades existentes en el area de Tecnologia de Grupo de Seguros G, mientras se utilizan tecnologías y plataformas tecnológicas fáciles de utilizar, a un bajo costo y que permitan escalar la solución de BI en un futuro pensando en las futuras necesidades del área.

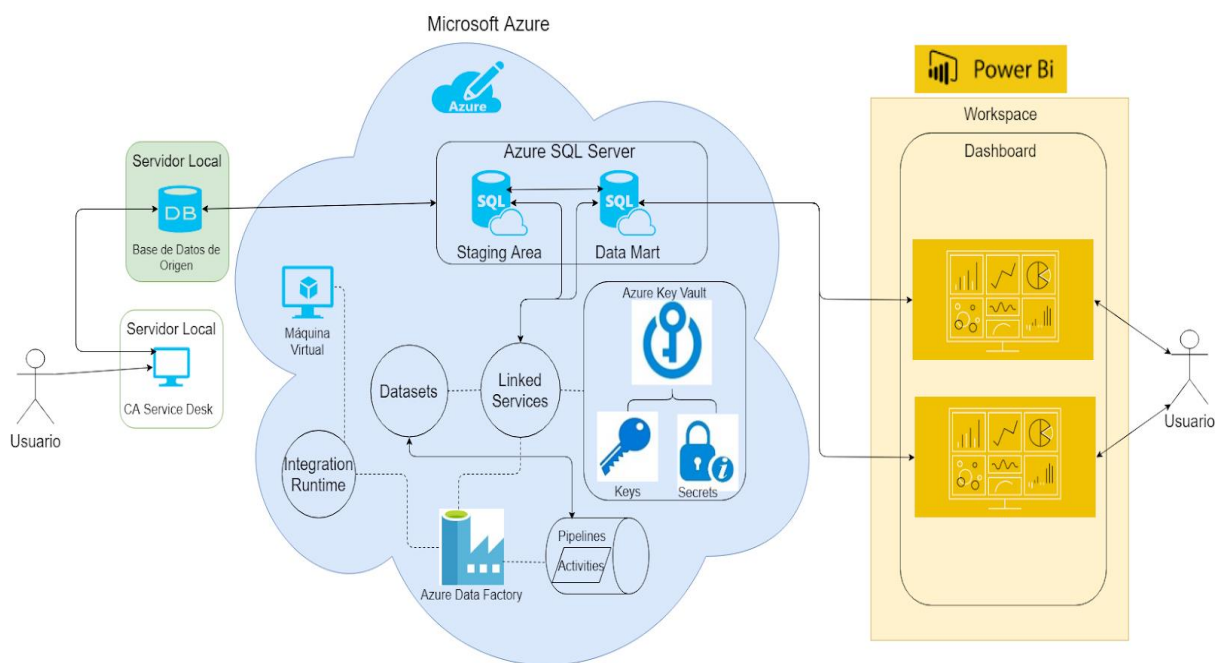


Figura 69. Esquema de arquitectura de BI. Fuente: Propia.

4. Conclusiones

El Grupo de Seguros G tiene deficiencias en sus procesos, políticas, metodologías y organización de manejo de información en su proceso de Gestión de Incidentes, sin embargo, si hay algo que aprender de la Inteligencia de Negocios, es que si hay datos, hay información, y si hay información, hay valor. Con el correcto acercamiento y la tecnología apropiada, se puede disponer de informaciones de alto valor estratégico para Gestión de Incidentes, y la alta criticidad de este proceso hace que cada paso que tomamos hacia su evolución se refleje en toda la organización.

Con una implementación de Inteligencia de Negocios que utilice un modelo relativamente simple como el de Kimball y que opere sobre herramientas de uso común como Microsoft Power BI, lo cual le permite a la organización apoyarse en el conocimiento colectivo que se ha desarrollado durante los años de uso de la herramienta alrededor del mundo, se puede crear una plataforma que facilite la extracción de información de las tablas de Gestión de Incidentes al mismo tiempo que sobrevuela las debilidades tecnológicas y operacionales del departamento de Tecnología de Información de la Organización.

Al momento de diseñar la implementación de Inteligencia de Negocios en una organización, se debe tener en cuenta de donde salen los datos que queremos explotar, y las debilidades que tiene la organización al momento de manipular estos datos. Es crucial observar la integridad de los datos con respecto a los procesos organizacionales que representan, de forma que la información que se genera a través de la Inteligencia de Negocios no sea dañada por las malas prácticas de la organización.

5. Recomendaciones

A partir de la información que recopilamos durante la redacción de este trabajo, identificamos debilidades en varios elementos de la Gestión de Incidentes de Grupo de Seguros G que afectan la calidad de la información que se puede extraer de los datos del proceso. Para solucionar estos problemas, se consideran las siguientes recomendaciones fuera de lo propuesto anteriormente en este trabajo:

- Cambiar el flujo de los procesos representado en el sistema de manejo de solicitudes para que se apegue a la realidad. Si esto se logra, será posible implementar este flujo de forma más integrada a la información extraída de las solicitudes.
- Crear plan de actualización de flujo de trabajo para la Gestión de Incidentes que contemple la actualización no sólo de las documentaciones y diagramas de flujo sino también de todas las representaciones de dicho flujo, en los sistemas y en los reportes de Inteligencia de Negocios.
- Documentar la estructura de base de datos de la herramienta de manejo de solicitudes.

6. Bibliografía

Chen, H., Chiang, R. y Storey, B. (2012). Business Intelligence and Analytics: from Big Data to Big Impact. *MIS Quarterly*, Vol. 36 No. 4, pag. 1165-1188. Recuperado de:

https://s3.amazonaws.com/academia.edu.documents/32970305/FROM_BIG_DATA_TO_BIG_IMPACT.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1553445618&Signature=A%2BkGYK%2BohZbPw2u9EFgol8l%2BK%2F8%3D&response-content-disposition=inline%3B%20filename%3DSPECIAL_ISSUE_BUSINESS_INTELLIGENCE_RESE.pdf

Ericson, G. (2018). What is Azure Machine Learning service?. Recuperado de:

<https://docs.microsoft.com/en-us/azure/machine-learning/service/overview-what-is-azure-ml>

Hoffman, C. (2018). What Is Microsoft Azure, Anyway?. Recuperado de:

<https://www.howtogeek.com/337961/what-is-microsoft-azure/>

Lo S. (2018). Datasets and linked services in Azure data Factory

Recuperado de:

<https://docs.microsoft.com/es-es/azure/data-factory/concepts-datasets-linked-services>

Minelli, M., Chambers, M., Dhiraj A. (2013). Big Data Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses. ISBN: ISBN 978-1-118-23915-5. Publicado por John Wiley & Sons, Inc., Hoboken, New Jersey, Estados Unidos de América. Recuperado de:

<https://books.google.es/books?id=Mg3WvT8uHV4C&printsec=frontcover&hl=es#v=onepage&q&f=false>

Neira, B. (2019). What is Azure Key Vault?

Recuperado de:

<https://docs.microsoft.com/en-us/azure/key-vault/key-vault-what-is>

Nottingham, C. (2018). Overview of Windows virtual machines in Azure

Recuperado de:

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/overview>

Oltra-Badenes, R., y Roig-Ferriol, J. (2014). Herramienta para la Evaluación de la Adecuación de Software al Proceso de Gestión de Incidentes de ITIL. *3 Ciencias*, Vol. 3 no. 11, pág. 213-227. Recuperado de: <http://ojs.3ciencias.com/index.php/3c-tic/article/view/209>

Rabeler, C. (2019). Choose the right SQL server option in Azure

Recuperado de:

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-paas-vs-sql-server-iaas>

Rouse, M. (2015). Microsoft Power BI. Recuperado de:

<https://searchcontentmanagement.techtarget.com/definition/Microsoft-Power-BI>

Rouse, M. (2016). SQL Azure (SQL). Recuperado de:

<https://searchsqlserver.techtarget.com/definition/SQL-Azure>

Rouse, M. (2016). Virtual Machine (VM). Recuperado de:

<https://searchservervirtualization.techtarget.com/definition/virtual-machine>

S Rawat, A Narain (2019). Understanding Azure Data Factory

Recuperado de:

<https://www.scribd.com/document/396148849/Understanding-Azure>

Yaseen, A. (2018). SQL Server Triggers: Understanding and Alternatives.

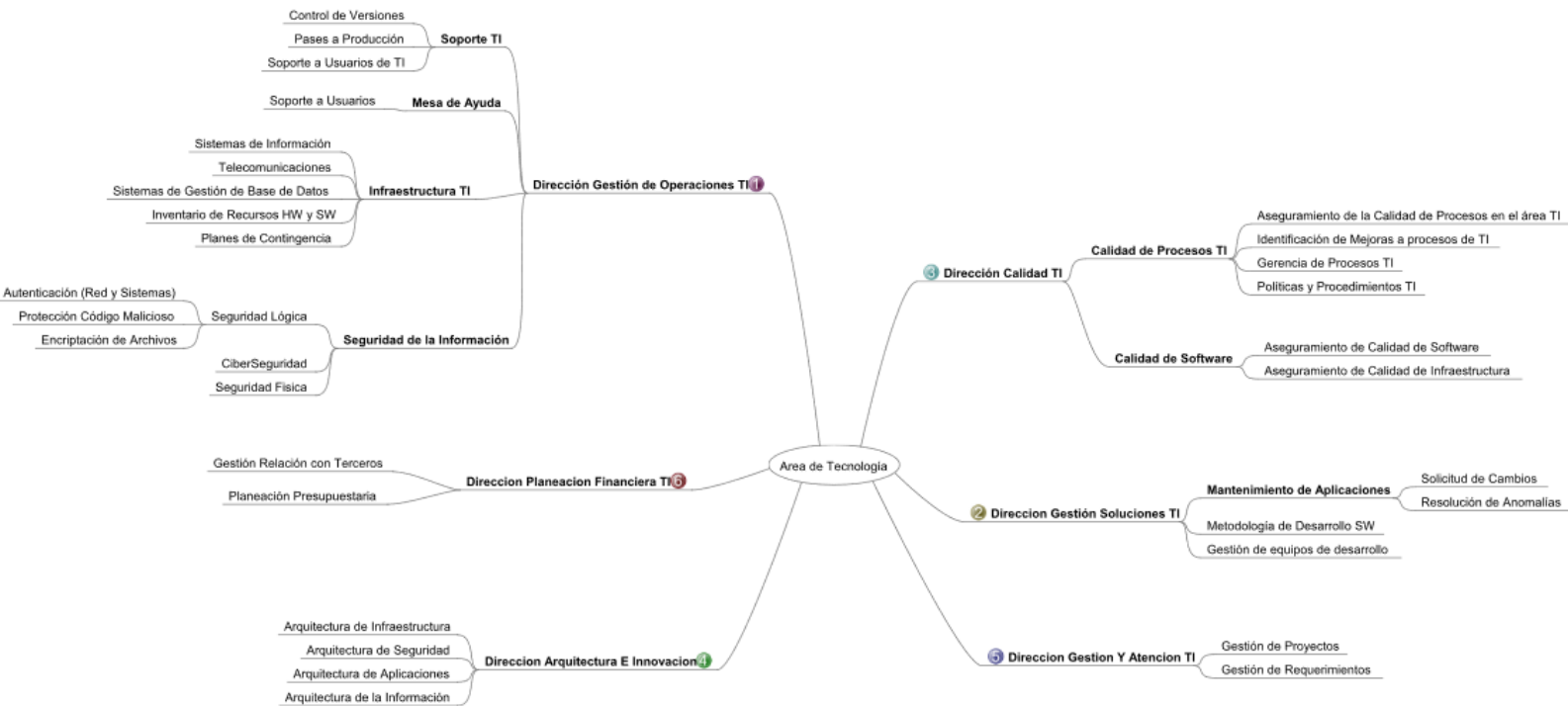
Recuperado de: <https://codingsight.com/sql-server-triggers-understanding-alternatives/>

Zhang, J. (2017). Integration Services (SSIS) Packages. Recuperado de:

<https://docs.microsoft.com/en-us/sql/integration-services/integration-services-ssis-packages?view=sql-server-2017>

Anexos

Anexo A: Diagrama de Entendimiento de Departamento Tecnología de la Información de Grupo de Seguros G



Anexo B: Inicio Unicenter Service Desk

The screenshot displays the Unicenter Service Desk interface. At the top left, the logo and text "Unicenter® Service Desk" are visible. To the right of the header, there is a "Request" dropdown menu and a "Go" button. Below the header, a blue bar indicates the user is logged in and provides a "(Log Out)" link. On the right side of this bar, it states "Scoreboard Updated: 03/22/2019 03:43 pm".

The main interface is divided into two primary sections. On the left is a navigation sidebar with a "Scoreboard" header and a list of menu items: "My Queue", "Requests", "Change Orders", "Issues", "Knowledge Documents", "Asignaciones Cerradas", and "Asignaciones Abiertas". Above this list is an "Update Counts" button. On the right is a large, empty area titled "Announcements".

At the bottom left of the interface, a small copyright notice reads: "Copyright © 2005 CA. All rights reserved."

Anexo C: Tablas de manejo de solicitudes

Tabla	CASOLI		
Campos	Nombre	Tipo	Tamaño
	id	Long Integer	4
	persid	Short Text	30
	ref_num	Short Text	30
	summary	Short Text	240
	description	Long Text	-
	status	Short Text	12
	active_flag	Long Integer	4
	open_date	Long Integer	4
	time_spent_sum	Long Integer	4
	last_mod_dt	Long Integer	4
	last_mod_by	Binary	16
	close_date	Long Integer	4
	resolve_date	Long Integer	4
	rootcause	Long Integer	4
	log_agent	Binary	16
	assignee	Binary	16
	group_id	Binary	16
	customer	Binary	16
	charge_back_id	Short Text	12
	affected_rc	Binary	16
	support_lev	Short Text	30
	category	Short Text	30
	solution	Short Text	30
	impact	Long Integer	4
	priority	Long Integer	4
	urgency	Long Integer	4
	severity	Long Integer	4
	extern_ref	Short Text	30
	last_act_id	Short Text	12
	cr_ticket	Long Integer	4
	parent	Short Text	30
	template_name	Short Text	30
	sla_violation	Long Integer	4
	predicted_sla_viol	Long Integer	4
	macro_predict_viol	Long Integer	4
	created_via	Long Integer	4
	call_back_date	Long Integer	4
	call_back_flag	Long Integer	4
	event_token	Short Text	30
	sched_token	Short Text	128
	type	Short Text	10
	string1	Short Text	40
	string2	Short Text	40
	string3	Short Text	40
	string4	Short Text	40
	string5	Short Text	40
	string6	Short Text	40
	problem	Short Text	30
	incident_priority	Long Integer	4
	change	Long Integer	4

Tabla	CACAMB		
Campos	Nombre	Tipo	Tamaño
	id	Long Integer	4
	persid	Short Text	30
	chg_ref_num	Short Text	30
	summary	Short Text	240
	description	Long Text	-
	status	Short Text	12
	active_flag	Long Integer	4
	start_date	Long Integer	4
	open_date	Long Integer	4
	last_mod_dt	Long Integer	4
	last_mod_by	Binary	16
	close_date	Long Integer	4
	resolve_date	Long Integer	4
	rootcause	Long Integer	4
	est_total_time	Long Integer	4
	actual_total_time	Long Integer	4
	log_agent	Binary	16
	assignee	Binary	16
	organization	Binary	16
	group_id	Binary	16
	affected_contact	Binary	16
	requestor	Binary	16
	category	Short Text	12
	priority	Long Integer	4
	need_by	Long Integer	4
	est_comp_date	Long Integer	4
	actual_comp_date	Long Integer	4
	est_cost	Long Integer	4
	actual_cost	Long Integer	4
	justification	Long Text	-
	backout_plan	Long Text	-
	impact	Long Integer	4
	parent	Long Integer	4
	effort	Short Text	240
	support_lev	Short Text	30
	template_name	Short Text	30
	sla_violation	Long Integer	4
	predicted_sla_viol	Long Integer	4
	macro_predict_viol	Long Integer	4
	created_via	Long Integer	4
	call_back_date	Long Integer	4
	call_back_flag	Long Integer	4
	string1	Short Text	40
	string2	Short Text	40
	string3	Short Text	40
	string4	Short Text	40
	string5	Short Text	40
	string6	Short Text	40
	service_date	Long Integer	4
	service_num	Short Text	30
	product	Long Integer	4
	actions	Long Text	-
	type_of_contact	Long Integer	4
	reporting_method	Long Integer	4
	person_contacting	Long Integer	4
	flag1	Long Integer	4
	flag2	Long Integer	4
	flag3	Long Integer	4
	flag4	Long Integer	4
	flag5	Long Integer	4
	flag6	Long Integer	4
	user1	Short Text	100
	user2	Short Text	100
	user3	Short Text	100
	cawf_procid	Short Text	40
	project	Binary	16

Tabla	CAESCA		
Campos	Nombre	Tipo	Tamaño
	id	Long Integer	4
	persid	Short Text	30
	del	Long Integer	4
	sym	Short Text	30
	last_mod_dt	Long Integer	4
	last_mod_by	Binary	16
	description	Long Text	-
	code	Short Text	12
	active	Long Integer	4
	hold	Long Integer	4
	resolved	Long Integer	4
	cr_flag	Long Integer	4
	in_flag	Long Integer	4
	pr_flag	Long Integer	4

Tabla	CAPRIO		
Campos	Nombre	Tipo	Tamaño
	id	Long Integer	4
	del	Long Integer	4
	enum	Long Integer	4
	sym	Short Text	12
	nx_desc	Short Text	40
	service_type	Short Text	30
	last_mod_dt	Long Integer	4
	last_mod_by	Binary	16

Tabla	CACATE		
Campos	Nombre	Tipo	Tamaño
	id	Long Integer	4
	persid	Short Text	30
	del	Long Integer	4
	sym	Short Text	128
	last_mod_dt	Long Integer	4
	last_mod_by	Binary	16
	description	Long Text	-
	organization	Binary	16
	assignee	Binary	16
	group_id	Binary	16
	tcode	Long Integer	4
	service_type	Short Text	30
	survey	Long Integer	4
	schedule	Long Integer	4
	auto_assign	Long Integer	4
	owning_contract	Long Integer	4
	cr_flag	Long Integer	4
	in_flag	Long Integer	4
	pr_flag	Long Integer	4

Tabla	CACARA		
Campos	Nombre	Tipo	Tamaño
	id	Long Integer	4
	persid	Short Text	30
	del	Long Integer	4
	sym	Short Text	60
	last_mod_dt	Long Integer	4
	last_mod_by	Binary	16
	description	Short Text	240