Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

INTELIGENCIA DE NEGOCIOS APLICADA DESDE *BIG DATA* COMO HERRAMIENTA PARA ANALIZAR EL PRESUPUESTO NACIONAL DE GUATEMALA DEL EJERCICIO FISCAL 2016

Humberto Rafael Reyes Bermúdez
Asesorado por el Ing. Carlos Antonio Mancio Reyes

Guatemala, marzo de 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

INTELIGENCIA DE NEGOCIOS APLICADA DESDE *BIG DATA* COMO HERRAMIENTA PARA ANALIZAR EL PRESUPUESTO NACIONAL DE GUATEMALA DEL EJERCICIO FISCAL 2016

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

HUMBERTO RAFAEL REYES BERMÚDEZ
ASESORADO POR EL ING. CARLOS ANTONIO MANCIO REYES

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, MARZO DE 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Jurgen Andoni Ramírez Ramírez
VOCAL V	Br. Oscar Humberto Galicia Nuñez
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
EXAMINADOR	Ing. Juan Álvaro Díaz Ardavin
EXAMINADOR	Ing. Manuel Haroldo Castillo Reyna
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

INTELIGENCIA DE NEGOCIOS APLICADA DESDE *BIG DATA* COMO HERRAMIENTA PARA ANALIZAR EL PRESUPUESTO NACIONAL DE GUATEMALA DEL EJERCICIO FISCAL 2016

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería de Ciencias y Sistemas, con fecha 22 de mayo de 2016.

Humberto Rafael Reyes Bermúdez

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS

Ref: ASESOR 02-02

Guatemala 04 de Agosto de 2016

Señores
Comisión de Revisión de Tesis
Carrera de Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala
Guatemala, Ciudad

Respetables Señores:

El motivo de la presente es informarles que como asesor del estudiante Humberto Rafael Reyes Bermúdez he procedido a revisar el trabajo de tesis titulado Inteligencia de Negocios aplicada desde Big Data como herramienta para analizar el presupuesto nacional de Guatemala del ejercicio fiscal 2016 y que de acuerdo a mi criterio el mismo se encuentra concluido y cumple con los objetivos definidos al inicio.

He tenido reuniones periódicas con el estudiante y luego de haber revisado cuidadosamente el trabajo, considero que cumple con los requisitos de calidad y profesionalismo que deben caracterizar a un futuro profesional de la Informática.

Aprovecho para informarle que he leído detenidamente el documento Ref: ASESOR 01-02 y aplicando las recomendaciones que se dan en el mismo procedo a firmar de revisado el trabajo de tesis.

Sin otro particular me suscribo de ustedes,

Atentamente,

Ing. Carlos Antonio Mancio Reyes

[Firma]

CARLOS A. MANCIO REYES
ING. EN CIENCIAS Y SISTEMAS
COLEGIADO No. 9390

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 17 Agosto de 2016

Ingeniero
Marlon Antonio Pérez Türk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **HUMBERTO RAFAEL REYES BERMÚDEZ** con carné **199617129,** titulado: **"INTELIGENCIA DE NEGOCIOS APLICADA DESDE BIG DATA COMO HERRAMIENTA PARA ANALIZAR EL PRESUPUESTO NACIONAL DE GUATEMALA DEL EJERCICIO FISCAL 2016",** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
INGENIERIA EN CIENCIAS Y SISTEMAS
COORDINACION TESIS

ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÌA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS
TEL: 24767644

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación ***INTELIGENCIA DE NEGOCIOS APLICADA DESDE BIG DATA COMO HERRAMIENTA PARA ANALIZAR EL PRESUPUESTO NACIONAL DE GUATEMALA DEL EJERCICIO FISCAL 2016,*** *realizado por el estudiante HUMBERTO RAFAEL REYES BERMÚDEZ aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAD A TODOS"

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION DE INGENIERIA EN CIENCIAS Y SISTEMAS

Ing. Marlon Antonio Pérez Türk
Director
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 23 de febrero de 2017

Universidad de San Carlos
de Guatemala

Facultad de Ingeniería
Decanato

Ref.DTG.D.116.2017

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **INTELIGENCIA DE NEGOCIOS APLICADA DESDE *BIG DATA* COMO HERRAMIENTA PARA ANALIZAR EL PRESUPUESTO NACIONAL DE GUATEMALA DEL EJERCICIO FISCAL 2016** presentado por el estudiante universitario: **Humberto Rafael Reyes Bermúdez,** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.

Ing. Pedro Antonio Aguilar Polanco
Decano

Guatemala, marzo de 2017

DECANO
FACULTAD DE INGENIERIA

/cc

ACTO QUE DEDICO A:

Dios	Por haberme dado todo en la vida.
Mi abuela	Por enseñarme a creer en mí y que no hay ninguna meta imposible de alcanzar.
Mi madre	Por ser siempre el motor que impulsa mis sueños.
Mi hija Isa	Por darme la alegría de ser tu papá.
Mi hijo Rafa	Me veo reflejado en ti, eres lo que siempre soñé.
Mi hermana	Por dejarme ser un ejemplo para ella y siempre demostrarme admiración y respeto.
Mi sobrina Nati	Por llamarme “hijo”.
Mi familia	Porque siempre he sentido amor de ustedes.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

GLOSARIO

CEPAL	Comisión Económica para América Latina.
CRUD	De las siglas en inglés *Create Read Update Delete* describe las operaciones básicas realizadas sobre la información de crear, consultar, actualizar y borrar.
Comprobante Único de Registro (CUR)	Documento que realiza la función de comprobante de una transferencia o pago registrados por las entidades del Estado.
Eurobono	Son bonos de estabilidad europeos que emiten los países de la zona del euro.
Guatenóminas	Sistema de información descentralizado de nóminas y registro de personal del Estado de Guatemala.
Guatecompras	Sistema de información de contrataciones y adquisiciones del Estado de Guatemala.
Hadoop	Proyecto de software abierto que permite el procesamiento distribuido de grandes conjuntos de información.

HDFS	De las siglas en inglés Hadoop *Distributed Files System* que se traduce a Sistema de Archivos Distribuidos de Hadoop en el que se basa el almacenamiento de grandes volúmenes de información en grandes grupos de servidores.
IPA	Interfaz de Programación para Aplicaciones.
JQuery	Marco de trabajo para el desarrollo de aplicaciones web.
Latencia	Se define como la sumatoria de los retardos temporales dentro de una solicitud de información y la obtención de la misma.
Metadata	Es el encabezado de información que describe un conjunto de información de un tipo definido.
Minfin	Ministerio de Finanzas Públicas.
Mspas	Ministerio de Salud Pública y Asistencia Social.
Open Source	Designación que se le da a todo software que se desarrolla en formato de colaboración abierta.
Segeplan	Secretaría de Planificación y Programación de la Presidencia.

SIAF	Sistemas Integrados de Administración Financiera.
Sicoin	Sistema de Contabilidad Integrada.
Sicoin GL	Sistema de Contabilidad Integrada para Gobiernos Locales (Municipalidades).
Siges	Sistema de Gestión.
Usabilidad	Grado de facilidad de uso que ofrece un sistema de información al usuario.

RESUMEN

En el Ministerio de Finanzas Públicas los requerimientos para el manejo de la información han crecido exponencialmente. Requiere de manejar grandes volúmenes de información que crecen día con día a la vez que se espera que los tiempos de respuesta sean más bajos. Las soluciones tradicionales para el manejo de información, como las bases de datos relaciones, se han llevado a su límite, lo que hace imposible contemplarlas como soluciones para empresas que necesitan administrar de manera eficiente la captura de información masiva, así como el posterior análisis de la misma, que en ocasiones llega hacer requerido en tiempo real.

A las necesidades antes expuestas han surgido soluciones de código abierto para solventar las necesidades actuales con nuevos planteamientos en la manera que se debe administrar la captura y la consulta de la información. Una de dichas soluciones se propone a través de Big Data (datos masivos) a través de la implementación de Apache Hadoop. Esta propone hacer uso de un sistema de archivos distribuidos de Hadoop, en el que se divide el problema de la administración de la información en el método que se utiliza para almacenar la misma y la forma posterior que se utilizará para consultarla.

El presente trabajo está orientado a exponer una forma de implementación de Big Data para el trabajo que se realiza en el Ministerio de Finanzas Públicas, para que este pueda plantear un análisis real y oportuno del gasto público, así como también de ser de acceso masivo para la ciudadanía en su constante búsqueda de poder transparentar la gestión del gobierno de Guatemala.

OBJETIVOS

Objetivo general

Generar evidencia documental sobre la aplicación de tecnologías *Open Source* (código abierto) para la arquitectura de Big Data en el análisis del presupuesto nacional.

Objetivos específicos

- Definir la arquitectura de software necesaria para almacenar y analizar la información del presupuesto nacional.

- Definir la arquitectura de hardware mínima y suficiente para las necesidades actuales.

- Proponer el conjunto de herramientas de software de código abierto y de licenciamiento libre que puedan implementar Big Data para el almacenamiento.

- Especificar las herramientas de software de código abierto y de licenciamiento libre para el análisis de la información.

- Definir el flujo del funcionamiento conjunto de las herramientas propuestas para el almacenamiento y el análisis de la información financiera y de ejecución.

INTRODUCCIÓN

Para el año 2016 el Congreso de la República de Guatemala aprobó un presupuesto público por Q 70 797,30 millones de quetzales, de estos se debe llevar un registro exacto y detallado del gasto público, lo que genera alrededor de 15 *Terabytes* de información almacenada en medios electrónicos.

A lo largo del ejercicio fiscal, el cual corresponde a un año calendario, se registran seis tipos de operaciones para llevar el control de los movimientos en el erario público, estas son: formulación presupuestaria, ejecución presupuestaría, flujos de caja, administración de bienes y servicios, administración de pagos y control de inversión pública. Estas operaciones son llevadas a cabo por treinta y tres entidades que representan el gobierno central.

Hoy en día todas estas operaciones se registran en el Sistema Integrado de Administración Financiera (SIAF), a través de sistemas transaccionales con una base de datos relacional, misma que a su vez, subdivide la información en dos grandes esquemas rígidos que almacenan distintos tipos de información. La contable que se lleva en el Sistema de Contabilidad Integrada Nacional (Sicoin) y la de gestión que es llevada por el Sistema Informático de Gestión (Siges).

Ambos sistemas Siges y Sicoin fueron desarrollados como parte de la iniciativa de SIAF que define estas dos grandes agrupaciones de información, tomando como punto inicial el desarrollo del Sicoin, el cual se empezó a utilizar por normativa del Ministerio de Finanzas Públicas (Minfin) en el año del 1998. Años después en el 2000 Minfin, como ente encargado del registro de las compras del Estado, agrega a su normativa de órdenes de compra el uso del Siges. La diferencia de años, diferentes objetivos y una falta de estandarización en el desarrollo de ambos sistemas dio como resultado una escasa interoperabilidad.

Todo esto ha llevado a que se creen islas de información que solo pueden ser interconectadas a través de una base de datos de integración para el análisis (*Data Warehouse*). El reto que marca esta integración es la naturaleza misma de la información que contienen ambos sistemas y la conceptualización en el momento del desarrollo. Todo esto da como producto final la necesidad de crear un sistema robusto para la inteligencia de negocios.

En diferentes períodos gubernamentales y con mayor empeño desde el año 2004, se han creado un sin número de iniciativas para promover la transparencia en el gasto público, esto ha obligado a que la información contable sea observada desde diferentes perspectivas y que la ejecución presupuestaria contemple una visión holística que permita tener un solo punto de referencia.

Una visión general de la información requiere que los entes externos (ejecutores) transporten la información, sobre su ejecución financiera a un solo repositorio de datos, en este caso puntual hacía el Minfin. Esté como ente rector necesita cruzar información de naturaleza financiera, contable y de ejecución presupuestaria con la información de ejecución física, como por ejemplo, la información recolectada como producto de la ejecución por el Ministerio de Salud Pública y Asistencia Social (Mspas), Ministerio de Educación (Mineduc) y otras instituciones de administración central. Esto para la generación de valor a la población en los servicios públicos prestados por las entidades, a manera de poder entender la asignación de recursos como un todo.

Todo esto conlleva estudiar la viabilidad de un sistema de almacenamiento que provea una gran flexibilidad en el resguardo de la información, un sistema de almacenamiento que no tenga la rigidez de necesitar un esquema definido. Esta es una de las primeras iniciativas al crear el concepto de Big Data que se define como “Gran volumen, alta velocidad y/o alta variedad de conjuntos de información que demandan costo vs efectividad, innova en formas de procesar la información que tienen como punto de mira, la toma de decisiones y los proceso de automatización” (Gartner, 2016)

Esta tesis propondrá una arquitectura de sistema que a través del uso de herramientas de código abierto (*Open Source*) y de licenciamiento libre, puedan proveer una solución innovadora haciendo uso de la tecnología de Big Data como medio de almacenamiento que proporcione la base para desarrollar herramientas para el análisis de la ejecución del presupuesto público y que a su vez sea un medio tanto para la toma de decisiones gerenciales por parte del Estado, como para que sea un medio para la auditoría social para la ciudadanía.

1. PRESUPUESTO PÚBLICO NACIONAL DE LA REPÚBLICA DE GUATEMALA

El instrumento más poderoso en la ejecución del Estado es el presupuesto público. Es el que provee de los recursos financieros que son transformados en insumos para que las diferentes entidades ejecutoras del gobierno puedan proveer de un bien o servicio a los ciudadanos de Guatemala.

"Considerando que el presupuesto público, como instrumento de planificación y de política económica, así como de gestión de la administración fiscal del Estado, requiere de una legislación adecuada que armonice en forma integrada con los sistemas de contabilidad gubernamental, tesorería y crédito público, los procesos de producción de bienes y servicios del sector público" (Ley Orgánica del Presupuesto, 1997)

Para el 2016 el Estado de Guatemala tiene un presupuesto aprobado por el Congreso de la República por un monto de Q 70 797,30 millones, lo que representa el cúmulo de recursos monetarios con los que cuentan todas las entidades del gobierno. Estos recursos monetarios serán transformados en insumos materiales y fuerza de trabajo, recursos que permitirán la producción de los bienes y servicios públicos que se prestarán a todos los guatemaltecos.

El número de transacciones de compra que se registran diariamente en los sistemas institucionales son de 2 518 como promedio de los últimos cinco años, 76 609 como promedio mensual y un gran total anual de 919 312[1] de registros por hoja de gasto, teniendo cada uno detalle promedio de 2 operaciones por cada hoja. Este volumen sumado al número de transacciones que se realizan para movimientos, tales como, la requisición de cuota financiera, modificaciones presupuestarias internas, registro de ingresos tributarios, y otras más, da como resultado un registro masivo de información financiera que se almacena en un sistema de base de datos relacional siendo este específicamente el sistema de administración de bases de datos relaciones de la empresa Oracle.

Los grandes volúmenes de información son generados por las cuatro etapas que tiene el presupuesto nacional, son descritas a continuación:

1. Planificación

   En esta etapa son generados los Planes de Operación Anual que deberán detallar la producción de cada entidad de gobierno al nivel de objetivos a cumplir como resultados esperados por la misma entidad. Son validados por la Secretaría de Planificación y Programación de la Presidencia (Segeplan).

[1] Número de transacciones tomados de base de datos Sicoin, incluye todos los registros con todos los diferentes estados que la información pueda registrar.

2. Formulación

   Luego de que el Minfin indique los techos presupuestarios asignados a cada entidad, estas deberán de realizar la formulación presupuestaria de los recursos financieros asignados a un nivel de detalle de programas, subprogramas, proyectos, actividad y obras a realizar, con especificación a nivel de renglón de gasto.

3. Ejecución

   Es la ejecución de la formulación presupuestaria, toma un año calendario en el que las diferentes instituciones del gobierno deberán de traducir los recursos financieros en insumos y recursos para poder alcanzar sus objetivos de producción.

4. Liquidación

   Obedece a reportar la transformación de los recursos financieros a producción de manera detallada y específica. Se debe de reportar todo gasto realizado dentro de la ejecución presupuestaria.

La información generada plantea retos importantes en la generación de herramientas de inteligencia de negocios que brinden acceso a indicadores oportunos y a disposición de todas las entidades del gobierno. Lo que implica varias operaciones sobre la información con un acceso concurrente de un número significativo de usuarios.

Las entidades de gobierno no son las únicas que deben tener acceso a la información presupuestaria. En la de la República en el artículo 237 (Congreso de la República de Guatemala, 1985) El presupuesto público deberá cumplir con tres principios básicos:

1. Anualidad
2. Universalidad
3. Publicidad

Del tercer principio se asume el compromiso que la información del presupuesto deberá publicarse, así como también los análisis que deriven de él, esto con el objetivo de transparentar la información que sirve para la toma de decisiones en las entidades de gobierno. Esto no se cumple en la gran mayoría de instituciones gubernativas.

La necesidad de un repositorio central de información que funcione como fuente de información de gobierno en cuanto a las finanzas del país, es una necesidad latente. Hoy en día el Minfin cumple con esta misión, pero no se tienen los medios de proveer el acceso de la misma de manera oportuna y sencilla para todo el ciudadano a pie.

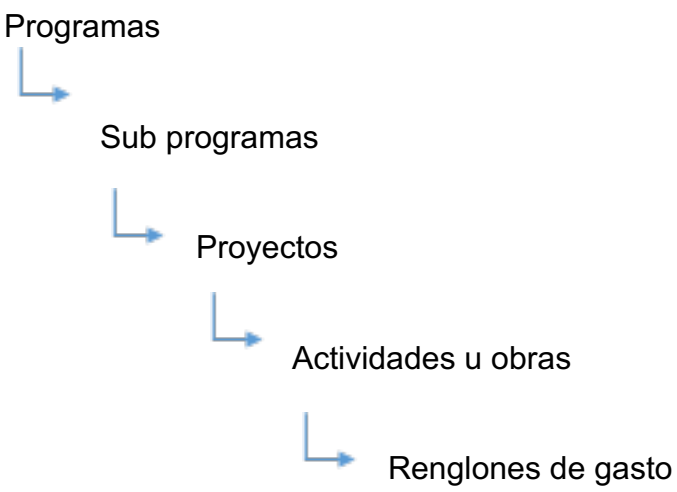
El Sicoin provee de acceso a las entidades de gobierno, así como también de acceso públicos a través de usuarios que son de conocimiento “público”, como por ejemplo, el usuario prensa con contraseña prensa. La problemática de esta vía de publicidad es en cuanto a la usabilidad que este sistema provee para un usuario que no posea un conocimiento profundo sobre la estructura programática del presupuesto, convirtiendo esta vía en más que un medio de acceso en un medio de frustración para el usuario que no es experto.

1.1. Estructura del Presupuesto Público

El instrumento del presupuesto público ha pasado a través de una serie de transformaciones y afinaciones hasta lo que actualmente utilizan las instituciones del estado para formular y ejecutar. En un inicio el presupuesto público era un instrumento que solo ayudaba a resolver los cuestionamientos: ¿Quién gastaba? y ¿Qué se compraba? Dejando por un lado la información sustancial de la ejecución del gasto, ¿Cuál es la razón del gasto?

En la actualidad la estructura del presupuesto parte del concepto de registrar la producción como medio principal para el cálculo del costo y con esto llevar a una formulación presupuestaria basa en programas que se deben de alinear con políticas de gobierno para el cumplimiento de metas de gobierno y estas a metas de Estado.

Los programas son el primer nivel de especificidad en la estructura del presupuesto. A continuación se detallan los cinco niveles que se tienen a nivel de estructura programática:



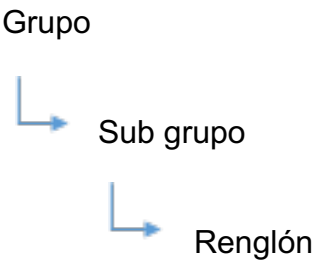
Programas: son agrupaciones de actividades necesarias para que las entidades públicas puedan cumplir con los objetivos planteados en la elaboración de la política de gobierno. Existen dos grandes clasificaciones, los programas sustantivos, que son los que agrupan actividades que prestan directamente un bien o servicio a los ciudadanos y los programas no sustantivos que son agrupaciones de actividades de apoyo a los programas sustantivos.

Sub programas: es un nivel más de especificidad de las agrupaciones de actividades, están contenidos dentro de los programas, sustantivos y no sustantivos.

Proyectos: se caracterizan por desarrollar una parte importante en los proyectos de inversión pública, que básicamente se refieren a los trabajos que se realizan para aumentar la infraestructura física del país.

Actividad u obras: es el mayor grado de especificidad que se tiene de la formulación presupuestaria en cuanto a detalle de ejecución de lo que se realizará. A estas actividades u obras son a las que se les hace la asignación de los recursos. Una actividad u obra es la que deberá describir la producción de las entidades.

Renglones de gasto: son descriptores para grupos de insumos que comparten características entre sí. Estos están organizados en tres niveles:



Estos tres niveles son de uso organizativo para los clasificadores del presupuesto público. Está organización es llamada Clasificación por Objeto de Gasto. En el Anexo A se muestra una tabla con los grupos, subgrupos y renglones de gasto existentes en el Manual de Clasificaciones Presupuestarias.

Existen cuatro clasificaciones por los cuales se organiza el presupuesto público. Se enumeran a continuación y que serán mencionadas a lo largo de este documento:

1. Clasificación Geográfica
2. Clasificación por Finalidad y Función
3. Clasificación por Tipo de Gasto
4. Clasificación por Fuente de Financiamiento

Además de estas existen otros clasificadores que no se estudiarán en esta propuesta por ser clasificaciones que no se relacionan con el uso cotidiano que se tiene de la información del presupuesto público dentro del Minfin. Al lector ávido que tenga alguna duda al respecto, se le recomienda leer el documento de “Clasificador del Presupuesto” publicado en el sitio web del Minfin.

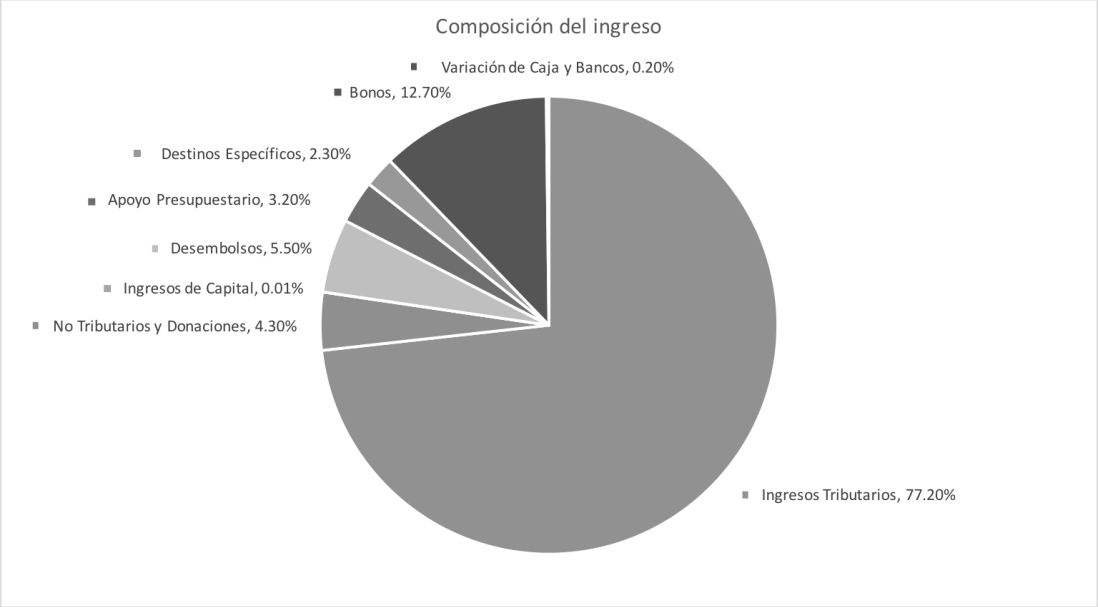
1.2. Rigidez del Presupuesto

El presupuesto público en términos nominales ha mostrado un aumento nominal sostenido en la última década, pero se observa una reducción en su monto respecto al PIB. Esto se debe a que actualmente se plantea el presupuesto con base en cuatro factores:

- Recaudación fiscal
- Deuda pública (interna y externa)
- Fondos propios
- Donaciones

La composición de las fuentes de financiamiento del presupuesto público logra mostrar que la mayor fuente de financiamiento del gasto proviene de las fuentes tributarias, dejando en evidencia la dependencia que se tiene entre poder de gasto y eficiencia en la recaudación. El estimado de la meta que provee la Superintendencia de Administración Tributaria (SAT) es de vital importancia que sea estimada basada en evidencia, ya que una mala estimación generará presión a la caja del Estado.

Figura 1. **Distribución de ingresos**



Fuente: elaboración propia, datos tomados del Sicoin.

En la figura 1 se muestran los ingresos con los que se financia el presupuesto 2016, mostrando que la mayor fuente de financiamiento son los ingresos tributarios, representando el 77,20% de los ingresos, teniendo como segundo rubro importante la colocación de bonos, estos se traducen en deuda externa en el caso de los eurobonos y deuda interna en los bonos del Banco de Guatemala.

Cada uno de los cuatro factores, antes mencionados, influyen directamente en los techos presupuestarios que el Minfin da como límite superior para los presupuestos que deberán elaborar cada entidad de gobierno, esto por ley se debe de efectuar a más tardar el 15 de junio de cada año anterior al ejercicio fiscal que se formula. Luego de esto, un mes después el día 15 de julio del mismo año cada entidad deberá de presentar al Minfin su formulación presupuestaria, la cual deberá de respetar el techo dado previamente.

A simple vista podría parecer que la formulación del presupuesto cuenta desde un inicio con montos de acceso a recursos para ejecución presupuestaria igual o mayor que con los que cuenta en el ejercicio fiscal anterior, siguiendo el comportamiento de aumento que tiene el presupuesto total. Esto no es del todo cierto, en la Constitución de la República existen obligaciones en cuanto a impuestos específicos que deben ser entregados a las instituciones indicadas, dando esto como resultado que las fuentes tributarias (impuestos) ya están previamente repartidas de acuerdo a leyes específicas. Para dar un ejemplo, el aporte que debe ser entregado a la Universidad San Carlos de Guatemala, el cual se especifica en el artículo 84 de la Constitución de la República, "corresponde a la Universidad de San Carlos de Guatemala una asignación privativa no menor del 5 por ciento del Presupuesto General de Ingresos Ordinarios del Estado" (Congreso de la República de Guatemala, 1985). En algunos casos el porcentaje es mucho mayor que este, como es el caso de las municipalidades que deberán recibir el 10% en base a una fórmula que calcula cuanto deberá de recibir cada municipalidad.

De esto podemos observar que antes de iniciar un cálculo de techos[2] para las diferentes entidades de gobierno hay una buena parte del presupuesto que está asignada desde el inicio por ley.

La siguiente tabla muestra los aportes indicados en la Constitución de la República (Ministerio de Finanzas Públicas, 1992):

[2] Un techo presupuestario corresponde al límite superior, asignado por el Minfin a las entidades, que no deberán de sobrepasar las entidades de gobierno al formular su presupuesto anual.

Tabla I. **Aportes obligatorios, Constitución de la República de Guatemala**

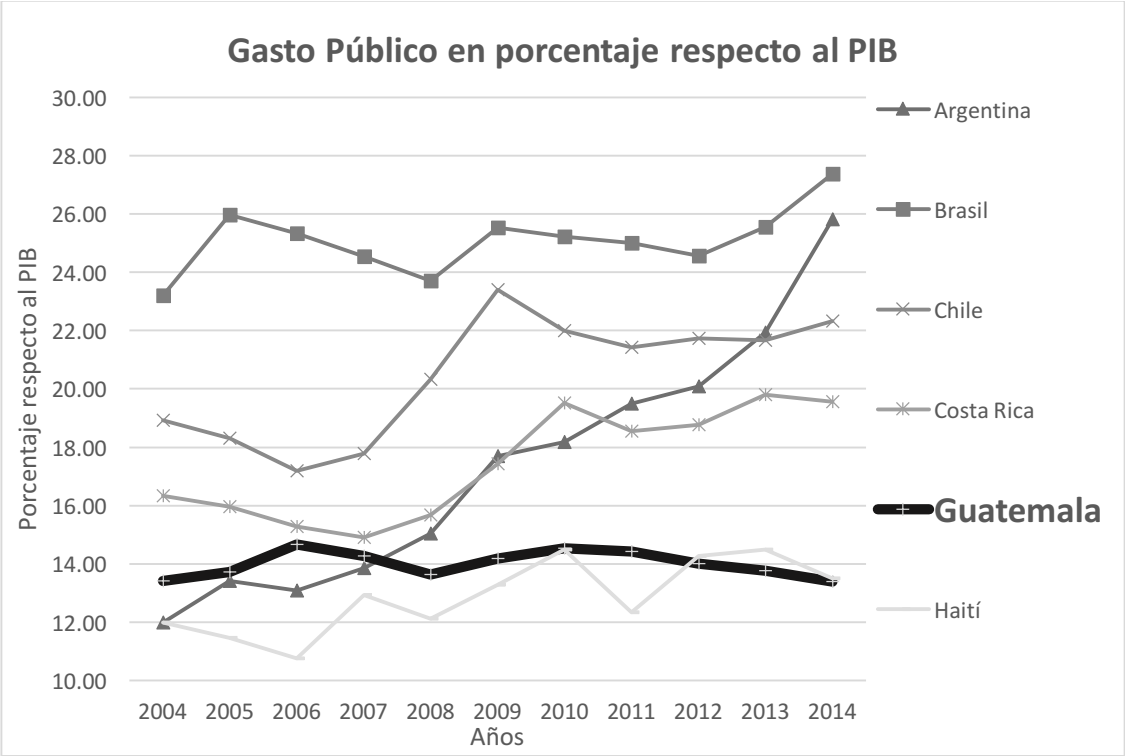
Beneficiario	Forma de Cálculo	Artículo de la Constitución de la República de Guatemala
Escuela Nacional Central de Agricultura	5% del presupuesto del Ministerio de Agricultura	79
Universidad de San Carlos de Guatemala	5% de los ingresos ordinarios del total del presupuesto	84
Confederación Deportiva Autónoma de Guatemala y Comité Olímpico de Guatemala	Se destina el 0.75% de Ingresos ordinarios para el deporte federado	91
Ministerio de Educación y Ministerio de Cultura y Deportes	0.75% de los ingresos ordinarios para educación física, deportes escolares y no federados	91
Consejos de Desarrollo Regionales y Departamentales	Deberán recibir el apoyo financiero necesario para su funcionamiento del gobierno central	229
Municipalidades	10% de los ingresos ordinarios	229

Fuente: elaboración propia.

De la tabla anterior podemos observar que más del 15% de los ingresos ordinarios ya tiene destinos específicos. Esto se agrava más cuando podemos observar que también dentro de los impuestos existen varios que ya tienen destino específico, como por ejemplo, el impuesto al tabaco que se destina al sector salud.

De lo antes expuesto se obtienen como conclusión que la repartición de los recursos financieros a las entidades de gobierno es una tarea ardua y de muchas trabas al querer reaccionar antes las necesidades de un país tan variado como el nuestro. Y más aún cuando el presupuesto público es el más bajo en Latino América como porcentaje respecto al PIB, esto ha sido así desde hace ya varias décadas. En la siguiente gráfica se muestra la evolución de los países de América Latina en su gasto público como porcentaje respecto al PIB, como muestra que va desde el 2004 hasta el 2014.

Figura 2. **Gasto público en porcentaje del PIB, Latinoamérica**



Fuente: elaboración propia, con datos de CEPAL.

Al limitado presupuesto nacional se le deja márgenes muy cortos al momento de destinar recursos para alguna área que se desee priorizar.

Es necesario hacer análisis detallados y específicos para ver los límites y alcances que alguna acción dirigida por los planes de gobierno tenga un impacto significativo o por el contrario el mismo esfuerzo sea diluido por los escasos recursos asignados.

A esto se puede agregar que el índice de alineamiento es algo que no se calcula hoy en día, es decir por índice de alineamiento entendemos el grado de empate que tienen las diferentes entidades de gobierno con las metas o prioridades de gobierno. Esto se puede observar en la asignación de recursos financieros a los programas, subprogramas, proyectos, actividades y obras en la formulación presupuestaria. Es necesario saber si todas la entidades están dirigiendo sus esfuerzos hacia el mismo punto o si por el contrario cada quien va con un rumbo distinto.

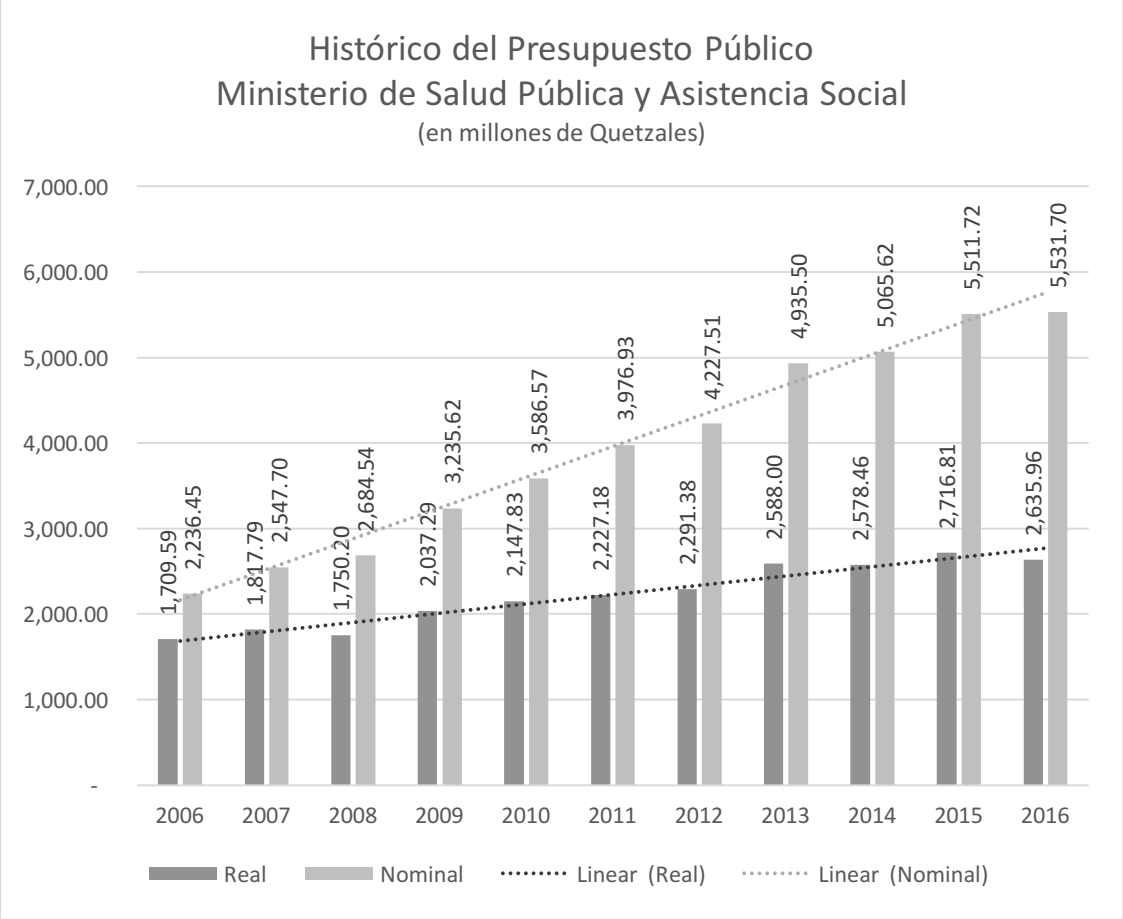
1.3. Histórico del Presupuesto

No podemos saber hacia dónde vamos sin saber de dónde partimos. Es una realidad que el presupuesto público año con año se ve consumido por gastos fijos que van en aumento, como por ejemplo el renglón 011 el cual es directamente para personal fijo, debido a los pactos colectivos. Si podemos observar la evolución de ciertos ministerios que juegan un papel principal en el desarrollo del país, también podremos observar que mientras mayor sea el número de personas necesarias para cumplir con sus funciones mayor será la repercusión de este fenómeno del crecimiento continuo de costos fijos sobre los costos variables de producción. Ejemplos de esto son: Ministerio de Educación que juega el papel fundamental en la educación preescolar, educación primaria, educación secundaria y educación terciaria, también tenemos al Ministerio de Salud Pública y Asistencia Social, ente rector de la salud curativa y preventiva de todos los guatemaltecos.

Cuando se compara los presupuestos anuales de la última década utilizando cifras nominales y reales se puede observar que el crecimiento sostenido realmente tiene un menor crecimiento de lo que se observa utilizando información nominal. Haciendo uso del método de deflactación (Reichler, 2005) con base 2001 = 1, nos permite hacer una comparativa sobre cifras equiparables despreciando el índice de la inflación.

Si a los datos se les traza una línea de tendencia (haciendo uso del método de mínimos cuadrados) (Spiegel, 2010) se podrá visualizar mejor como es que el presupuesto ha sufrido ligeros cambios en la última década.

Figura 3. **Histórico del Presupuesto Público, Mspas**

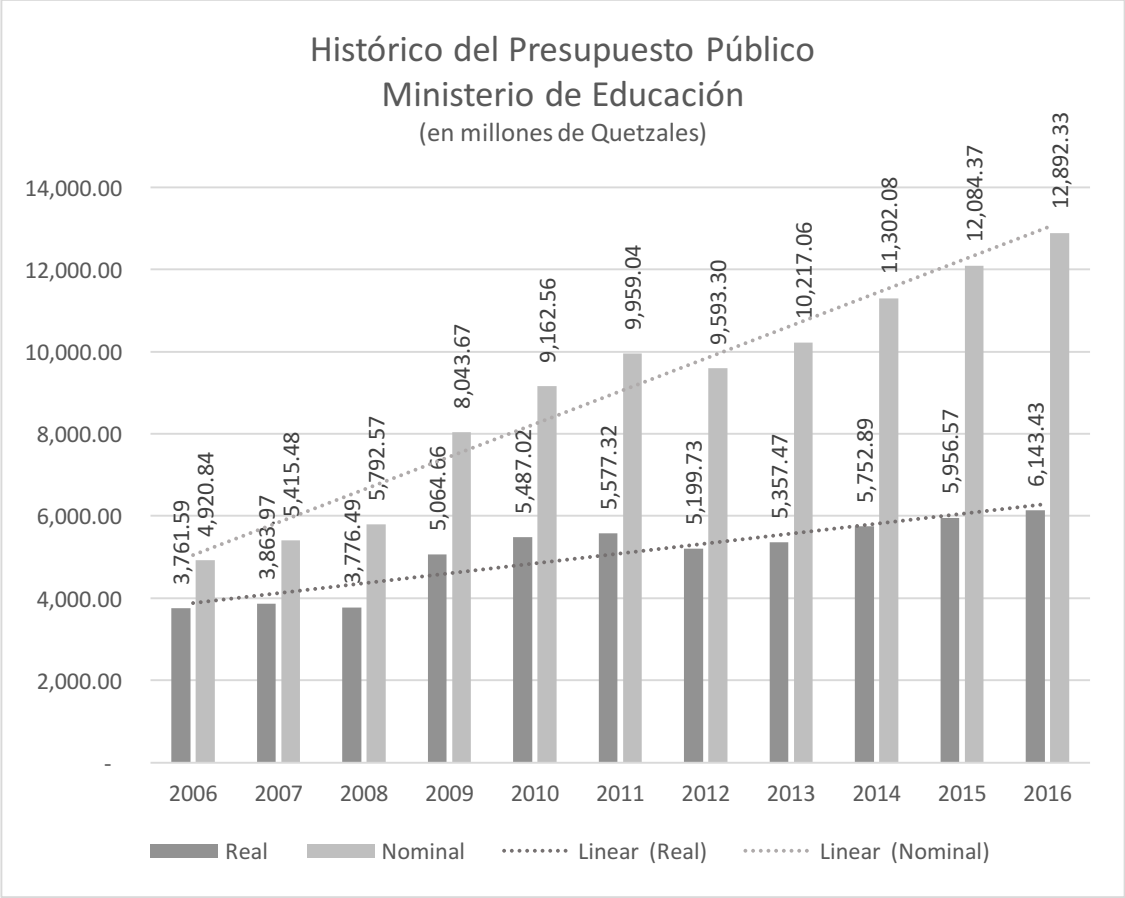


Fuente: elaboración propia, con datos de CEPAL.

Se logra observar que el incremento aparente del que gozan, realmente es un incremento marginal. Así como también que el mayor rubro de crecimiento es en la parte de personal de producción.

Haciendo la misma observación en el caso del Ministerio de Educación, el panorama no es mucho más favorable.

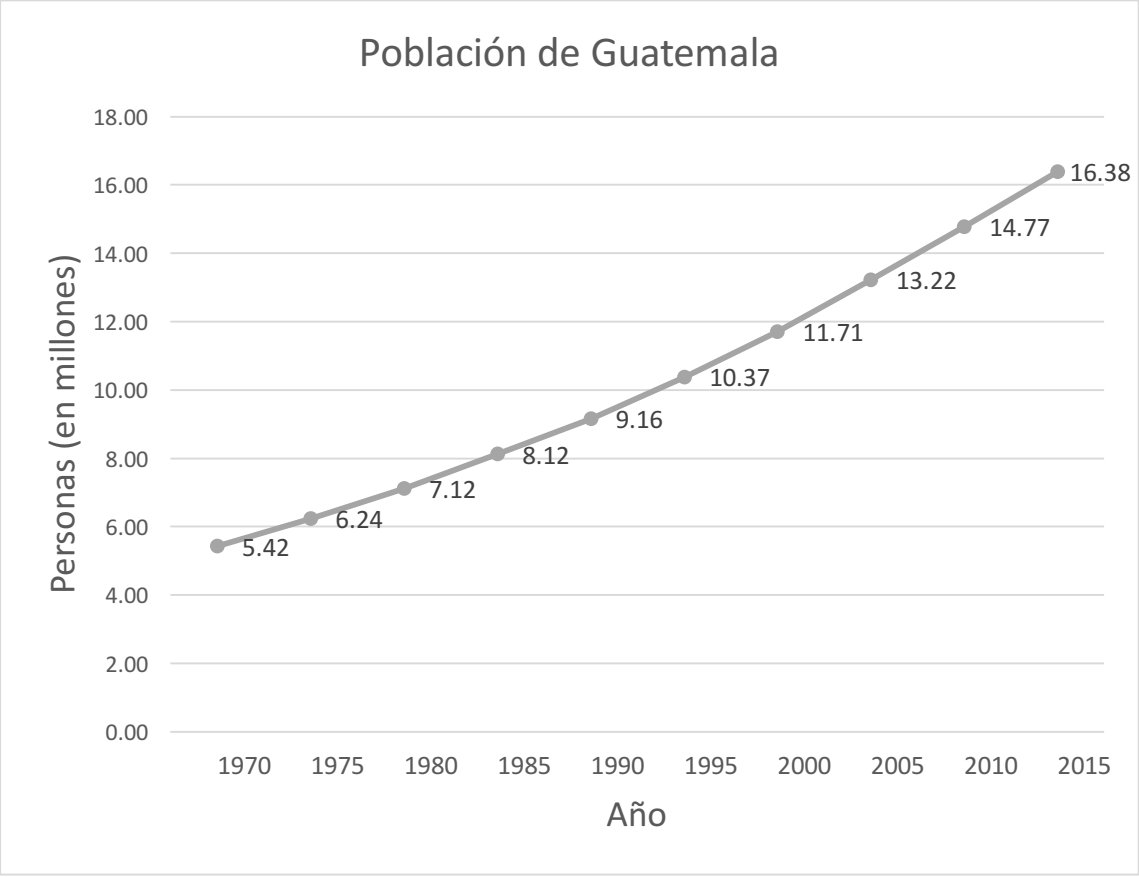
Figura 4. **Histórico del presupuesto público del Ministerio de Educación**



Fuente: elaboración propia, con datos de Sicoin.

La tendencia se repite de manera general en los históricos de los presupuestos públicos de todas las entidades de gobierno.

Figura 5. **Histórico de la población de Guatemala en intervalos quinquenales**



Fuente: elaboración propia, con información de CEPAL.

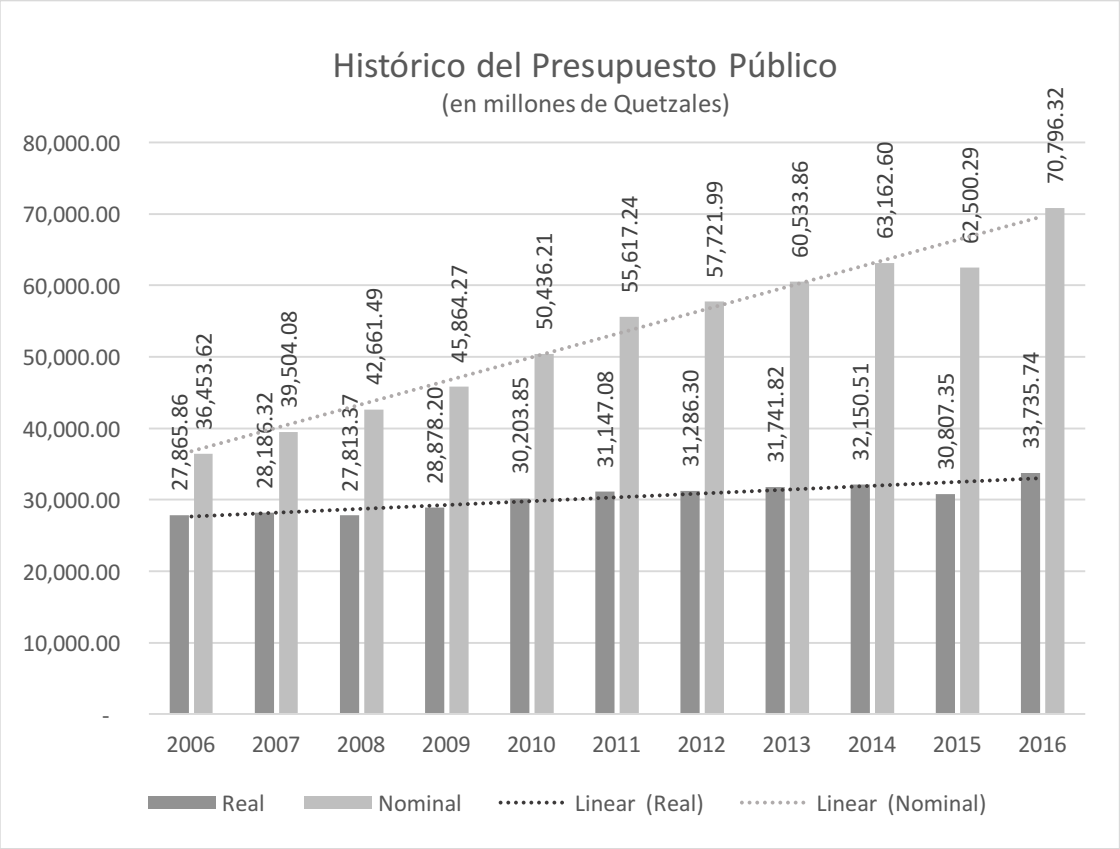
Tomando en cuenta que la población de Guatemala ha crecido aún ritmo de constante, es válido preguntar si nominalmente hablando, la asignación presupuestaria que se tienen en las entidades de gobierno, es suficiente para cubrir las necesidades de un país cada vez mayor.

Por lo antes expuesto se sabe que cada día se hace mayor la necesidad de contar con estrategias de eficiencia y calidad en el gasto público, sin poder olvidar el límite de que nuestro país es el que tiene el presupuesto público más bajo de la región de Latinoamérica. Por lo que se requiere de herramientas cada vez más sofisticadas para poder recopilar volúmenes de información cada día mayor y poder cruzar información de diferente fuente y tipo, para de esta manera tener una visión general de que problemas enfrentar y cómo hacerlo con recursos que en primera instancia ya son insuficientes.

Muchos ciudadanos son adversos a la idea de un incremento en la carga tributaria, pero a su vez exigen cada día más y mejores servicios públicos. Esto en primera instancia se debe al mal manejo de los recursos del Estado por parte de los equipos de gobiernos que han tomado el poder a través de la reciente historia de la democracia en nuestro país. La incredulidad del ciudadano a pie, sobre la transparencia con la que el gobierno ejecuta el presupuesto, aunado a la falta de información y accesibilidad a la misma por parte de las instituciones del Estado ha creado un ambiente en el que la falta de cultura tributaria está a la orden día. Esto expone una nueva dimensión, la auditoría social escasa o casi nula por la falta de información, pobre calidad y la antigüedad de la misma, no son circunstancias ideales para hacer análisis mientras se lleva a cabo la ejecución, esto impide una reacción temprana lo que provoca que no existan soluciones a las problemáticas cotidianas, siempre se tiene un análisis tardío de lo sucedido.

De todo lo antes expuesto se puede observar en la siguiente gráfica que la falta de crecimiento del presupuesto público es un problema de nación.

Figura 6. **Histórico del Presupuesto público de la república de Guatemala**



Fuente: elaboración propia, con información de Sicoin.

2. SISTEMAS ACTUALES

2.1. Definición SIAF

Los Sistemas de Administración Financiera (SIAF) se implementan en América Latina desde mediados de la década de los 80 con el objetivo de gestionar de manera eficaz y eficiente la administración financiera pública. En Guatemala la iniciativa fue tomada con el propósito de controlar la deuda flotante, que por ese entonces, era cada vez mayor.

Los avances en materia de tecnología (hardware, software y telecomunicaciones) así como la actualización conceptual de procesos y procedimientos en la administración financiera pública, ha llevado a una actualización continua de las versiones de los SIAF implementados.

Sobre la base de una definición genérica se puede decir que se conoce como SIAF como "El conjunto de principios, normas, organismos, recursos, sistemas de información y procedimientos que intervienen en las operaciones de programación, gestión y control necesarias para captar los fondos públicos y aplicarlos para la concreción de los objetivos y metas del Estado en la forma más eficiente posible"[3]

[3] Secreataría de Hacienda, Ministerio de de Hacienda y Finanzas Públicas, Argentina, 1991

2.2. Componentes del SIAF

Las funcionalidades del SIAF en su tercera versión, la cual es la versión utilizada actualmente, son determinadas por el alcance de las aplicaciones que lo conforman y las interfaces entre ellas y de otros agentes externos que posibilitan el flujo y la generación de información financiera.

Figura 7. **Relación de los sistemas principales con los subsistemas existentes**

Fuente: elaboración propia.

Como se puede observar en la figura 7, desde el punto de vista de los sistemas informáticos, al SIAF lo conforman estas aplicaciones o subsistemas y los mecanismos de intercambio de información entre ellos, ya sea a través de servicios web, accesos a las bases de datos de las distintas aplicaciones a través de links directos o el uso de vistas materializadas.

De igual manera las aplicaciones que componen el SIAF implementan mecanismos de intercambio de información con sistemas y aplicaciones externas como bancos y otros sistemas de información implementados en la gestión pública.

Es importante destacar que tanto los servicios como los mecanismos empleados a través de vistas entre distintas bases de datos de las aplicaciones componentes del SIAF no implementan lógica de negocio que permitan establecer algún nivel de integración funcional de las aplicaciones o reutilización de funcionalidades. Se trata en todos los casos de procedimientos o servicios de "mensajería" que cumplen con él envío y recepción de datos entre aplicaciones.

2.3. Volumen de información de los sistemas actuales

En el siguiente cuadro se muestra el total de transacciones por sistema componente del SIAF, el total de transacciones contempla actividades dentro de los sistemas que no siempre están asociadas a las funcionalidades de valor para el desarrollo de la gestión financiera, sino que se incluyen en algunos casos transacciones de administración del sistema, consultas, accesos, otros.

A continuación se presenta el total de transacciones mensuales de cada sistema, seleccionadas por módulos de cada aplicación:

Tabla II. **Transacciones realizadas en los diferentes sistemas**

Sistema	Transacciones mensuales
Guatecompras	6 066 945
Sicoin	1 370 189
Siges	997 047
Guatenóminas	1 330 165
Total	9 764 346

Fuente: elaboración propia, con datos proporcionados por la Dirección de Tecnologías de la Información del Minfin.

2.4. Guatenóminas

Guatenóminas es la aplicación utilizada para la liquidación de nómina de los renglones (011, 021, 022 y 029) así como el registro de fichas de personal, Guatenóminas cuenta con 3 194 usuarios nombrados en la base de datos.

2.5. Sistema de Contabilidad Integrada (Sicoin)

El Sicoin, es el sistema contable del SIAF para el ámbito del Gobierno central y empresas públicas del sector público no financiero. Si bien el Sicoin se refiere al sistema contable para las entidades del Gobierno central y empresas públicas, la aplicación también implementa servicios relacionados a tesorería y presupuesto, constituyéndose por tanto en el SIAF para este grupo de instituciones.

Existen dos versiones del Sicoin, que trabajan sobre diferentes esquemas en la misma base de datos, estas versiones son:

- Sicoin Gobierno central (https://sicoin.Minfin.gob.gt/sicoinweb/)
- Sicoin descentralizadas (https://sicoindes.Minfin.gob.gt/sicoinweb/)

El Módulo Presupuesto por Resultado se ha implementado fuera del Sicoin, se creó dentro del alcance del Sistema de Gestión (Siges). A la fecha todas las entidades formulan presupuesto a partir de esta nueva funcionalidad de Siges pero solamente administración central[4] ejecuta el presupuesto a través de esta funcionalidad en Siges.

Los módulos funcionales identificados en el Sicoin y para los cuales se presenta un promedio de transacciones son los siguientes:

Tabla III. **Tabla resumen de las transacciones realizadas en el Sicoin**

Módulos Funcionales	Promedio Mensual de Transacciones
Clasificadores	2 223
Ejecución de Ingresos	26 678
Ejecución de Gastos	346 167
Contabilidad	213 948
Administración y Seguridad	7 732
Tesorería	253 822
Administrativo	121
Convenios y Fideicomisos	9
Fondo Rotativo	61 068
Formulación	383 421
Inventarios	75 000
Total	1 370 189

Fuente: elaboración propia.

[4] Conjunto de 14 Ministerios, Presidencia de la República y Secretarias y otras dependencias del ejecutivo.

2.6. Sistema de Gestión (Siges)

El Siges un es un sistema del SIAF, fue diseñado originalmente para la gestión de compras y contrataciones en toda la Administración Central. Igualmente, se dispone dentro del SIAF de una aplicación para realizar los trámites de compras por Internet (Guatecompras), la cual es de uso obligatorio en toda la administración pública de Guatemala en concreto el sector público no financiero (SPNF).

En el Siges se ha implementado la formulación de presupuesto por resultado y la ejecución del mismo. Ambas funcionalidades implementan mecanismos de actualización de datos en el modelo de datos del Sicoin.

Tabla IV. **Tabla resumen de las transacciones realizadas en el Siges**

Aplicación/Módulo (Nombre del módulo en el sistema)	Promedio mensual
Catálogo de insumos	4 311
Constancia de disponibilidad presupuestaria	55 966
Contratos	5 386
Ejecución de proyectos de inversión (SNIP)	527
Expediente de gasto	29 838
Formulación nómina	18 117
Formulación presupuestos por resultados	1 157
Formulación proyectos de Inversión	51 441
Orden de compra	965
Pagos a municipios	478 811
Pre-Orden de compra	10
Procesos de compra	0
Reportes ley de acceso a la información	346 617
Donaciones	264
Fideicomisos	73
Rendición de cuentas	22
Juntas escolares	247
Promedio general	3 296

Fuente: elaboración propia.

2.7. Guatecompras

El sistema de Guatecompras, sirve para tramitar las compras y publicarlas, pero no interactúa con el SIAF para registrar las distintas etapas presupuestarias y contables.

2.8. Servicios de Gobiernos Locales (Sicoin GL)

Este sistema es usado por los gobiernos locales, en otras palabras municipalidades de los 344 municipios de Guatemala. Estas instituciones generan aproximadamente 3 650 000 transacciones por mes.

3. INTELIGENCIA DE NEGOCIOS

Es la combinación de tecnología, herramientas y procesos que le permiten a una empresa, organización, gobierno u otras hacer una transformación de los datos almacenados en información, esta información se puede transformar en conocimiento y este conocimiento a su vez deberá ser dirigido a un plan o a una estrategia. Esta es la definición que provee *The Data Warehose Insittute*.[5]

La inteligencia de negocios debe ser parte de la estrategia de gobierno, esta le permite optimizar la utilización de los recursos, monitorear el cumplimiento de los objetivos de la política general de gobierno, del plan de gobierno, de las metas presidenciales y de los objetivos específicos de cada entidad de gobierno. Esto ayudará a que exista un mejor índice de alineamiento y podrá generar una visión general de gobierno.

3.1. ¿Por qué Inteligencia de Negocios?

La aplicación de inteligencia de negocios o BI por sus siglas en inglés (*Business Inteligence*) puede verse fácilmente en entornos gubernamentales al describir el siguiente cuestionamiento: ¿Cuáles son los padecimientos más comunes con los que se enfrenta el gobierno hoy en día?

[5] https://tdwi.org/Home.aspx. Consulta: 8 de Agosto de 2016.

3.1.1. Se tienen datos pero se carece de información

Es importante resguardar los datos generados cada día, como por ejemplo, transacciones que registran la compra de bienes o insumos por parte de la entidades de gobierno, registro de nóminas pagadas, beneficiarios del programa de clases pasivas del Estado. Pero si queremos hacer un mejor uso de los recursos públicos se debe de profundizar el nivel de conocimiento para así poder tener la capacidad de encontrar patrones de comportamiento, monitorear, rastrear, entender, administrar y contestar aquellas interrogantes que permiten maximizar el rendimiento de los recursos públicos.

3.1.2. Fragmentación

Los ministerios más importantes del gobierno central cuentan con sistemas propios y generan grandes volúmenes de información pero se carece de una visión global de gobierno. Tal vez por la incapacidad de poder centralizar y almacenar grandes volúmenes de información. Esto limita al gobierno a tomar decisiones importantes sin tener toda la información relevante a la mano. Esta fragmentación conduce a lo que comúnmente se conoce como diferentes versiones de la verdad. Los ministros solicitan informes a las distintas direcciones o departamentos de su institución y reciben como resultado informes que difieren en la información entre departamentos. La tarea ya no es solamente crear el reporte sino también justificarlo y aclarar las condiciones que utilizaron para la generación del mismo. Si el ministro decide agregar una nueva variable o un nuevo filtro de información, recrear este reporte puede conllevar un esfuerzo de horas e incluso días; si acaso es posible generarlo.

3.1.3. Manipulación manual

La necesidad de generar análisis de la información ha llevado a la utilización de herramientas de inteligencia de negocios y/o reportes que no son confiables, como por ejemplo, el uso de aplicaciones como Excel que facilitan a que el usuario pueda analizar la información obligándole a realizar la misma secuencia de pasos para poder actualizar dicho análisis al momento de disponer de nueva información. Esto aumenta la probabilidad de error en cada de las actualizaciones. Esta práctica conlleva exportación de datos a distintas herramientas que resultan en un proceso lento, costoso, duplicación de trabajo, poca confiabilidad en los informes, propenso a errores y sujeto a la interpretación individual y a la discrecionalidad de cada generador de la información.

3.1.4. Poca agilidad

Debido los tres factores antes expuestos, la agilidad que se pueda tener en la toma de decisiones basada en información es poca o casi nula.

Desde un punto de vista pragmático, asociándolo directamente con las tecnologías de información, podemos definir inteligencia de negocios como el conjunto de metodologías, aplicaciones y tecnologías que permiten reunir, depurar y transformar los datos de los sistemas transaccionales y cualquier fuente de información estructurada y no estructurada, en información con sentido dentro del contexto de análisis, para su explotación directa o para su transformación en conocimiento, dando así soporte a la toma de decisiones.

De esta manera se hace notar la necesidad de poder transformar toda la información con la que el gobierno cuenta actualmente en los sistemas Sicoin y Siges, además de pequeños sistemas satélites que sirven de complemento. Estos poseen una información de alrededor de 120 *Terabytes* de tamaño almacenada electrónicamente, la cual resulta ser un escenario ideal para la implementación de inteligencia de negocios por la riqueza de datos e información almacenada.

3.2. Beneficios de la Inteligencia de Negocios

Dentro del marco de beneficios que representa una solución de inteligencia de negocios se pueden destacar las siguientes:

3.2.1. Gestión del conocimiento

El reto para el Minfin es evolucionar, crecer y esto significa “cambio”. ¿Qué tan ágiles pueden ser los procesos para enfrentar los cambios y las necesidades puntuales del país?.

"El nivel más bajo de los hechos conocidos son los datos. Los datos no tienen un significado intrínseco. Deben ser ordenados, agrupados, analizados e interpretados. Cuando los datos son procesados de esta manera, se convierten en información. La información tiene una esencia y un propósito. Cuando la información es utilizada y puesta en el contexto o marco de referencia de una persona, se transforma en conocimiento. El conocimiento es la combinación de información, contexto y experiencia." (Harris, 1996)

3.2.2. Control de costos

El control en el uso de los recursos públicos deberá ser el detonador que forcé el considerar la inteligencia de negocios. Esto se debe, una vez más, al muy bien conocido concepto de la administración, “lo que no se puede medir no se puede controlar”.

3.2.3. Entender mejor la necesidades de país

El Minfin almacena enormes volúmenes de información valiosa relacionada al uso del presupuesto público. El reto es transformar esta información en conocimiento y este conocimiento dirigirlo a una gestión de gobierno que represente un mejor impacto en el abastecimiento de recursos financieros a las diferentes entidades, para que a su vez, estas puedan ejecutar mejor un plan de gobierno, a través de entender mejor la necesidades de país.

3.2.4. Indicadores de gestión

Los indicadores de desempeño que permiten representar medidas enfocadas del trabajo de las entidades de gobierno en cuanto a la ejecución financiera y física de los programas de gobierno deberán de tener la capacidad de representar la estrategia organizacional en objetivos, métricas, iniciativas y tareas dirigidas. Dentro de las capacidades funcionales de los indicadores de gestión podemos mencionar: el monitoreo, análisis y administración.

- Los indicadores deberán monitorear los procesos críticos de gobierno y las actividades utilizando métricas que alerten sobre un problema potencial o alguna gestión que se debe realizar.

- Analizarán la raíz de los problemas explorando la información desde múltiples perspectivas en varios niveles de detalle.

- Administrar los recursos y procesos para dirigir la toma de decisiones, optimizar el desempeño. Esto nos permite tener una visión global del gobierno con la capacidad de dirigir a las entidades en la dirección correcta.

3.3. Big Data

3.3.1. ¿Qué es Big Data?

Históricamente, un alto número de grandes empresas de tecnología que se dedican a búsquedas por internet, publicidad y redes sociales han sido pioneras en generar las innovaciones de software y hardware para Big Data. Por ejemplo, Google analiza los clics, links y el contenido de 1.5 trillones de vistas de páginas por día[6] y los resultados de las búsquedas realizadas y personalizadas son mostrados en milisegundos. Esta es una hazaña remarcable de las ciencias de la computación.

[6] Estadísticas consultadas del sitio web http://www.alexa.com. Consulta: 5 de Mayo 2016.

Google, Yahoo, Amazon y otras compañías han contribuido con sus propias tecnologías a la comunidad de código abierto, abriendo la puerta a las entidades de ámbito comercial y de sector público interesado en tomar el desafío de hacer trabajos con tecnología de Big Data. Las empresas comerciales han visualizado la tecnología de Big Data ligeramente diferente. En lugar de interpretar la información de manera independiente, ellas ven el valor de agregar la nueva información a los sistemas de bases de datos ya existentes. Esto difiere de la visión inicial de Big Data que tenía como objetivo plantear soluciones totalmente nuevas que no incluyesen los conceptos tradicionales de cómo almacenar y consultar la información.

Entonces, Big Data describe una estrategia para el manejo de información holística, que incluya e integre muchos tipos de información y manejos de información junto con los datos tradicionales. Mientras que muchas de las técnicas para procesar y analizar ese tipo de datos llevan existiendo por algún tiempo, han tenido una proliferación masiva y una baja en el costo del poder procesamiento. Adicionalmente, Big Data ha popularizado dos almacenamientos principales para las técnicas de procesamiento: Apache Hadoop y las bases de datos *NoSQL*[7].

La Big Data también ha sido definida a través del concepto de las cuatro "V": Volumen, Velocidad, Variedad y Valor. Estos conceptos se convierten una prueba razonable para determinar si agregar Big Data a la arquitectura de información.

[7] Bases de datos que difieren de los conceptos tradicionales de bases de datos relaciones, una característica común es que no basan el acceso a la información a través de un lenguaje de consulta estructurado, *SQL* por sus siglas en ingles.

3.3.1.1. Volumen

Se refiere a la cantidad de información. Mientras que el volumen indica más data, este concepto se refiere a la naturaleza de la especificidad de la data y de su unicidad. Los grandes volúmenes de información requiere grandes volúmenes de procesamiento de información de baja densidad, esto es, información de valores desconocidos, como por ejemplo, la información de Twitter, los clics de una página web, el tráfico de red, equipos con sensores que capturan la información a grandes velocidades y muchos más. Esta es la tarea de Big Data que convierte la información de baja densidad en información de alta densidad, esto es, información con valor. Para algunas compañías significa decenas de "terabytes"[8], para otras son muchos cientos de "petabytes"[9].

3.3.1.2. Velocidad

Un rápido flujo de información es recibido y se debe actuar en consecuencia. La alta velocidad de ingreso de la información es cargada directamente en la memoria en contraparte de ser escrita a disco. Algunas aplicaciones que tienen información de salud y políticas de seguridad requieren evaluaciones en tiempo real para tomar acciones inmediatas.

[8] Mil *gigabytes*
[9] Mil *terabytes*

Otros productos inteligentes habilitados para el uso de internet operan en escenarios de tiempo real y muy cerca a tiempo real. Como por ejemplo, el consumo a través de aplicaciones que buscan combinar la tecnología del geo posicionamiento a través de dispositivos móviles con las preferencias personales del usuario para hacerle ofertas más adecuadas a los gustos de los usuarios. Operacionalmente, las aplicaciones móviles son de una cantidad alta de usuarios, incrementa el tráfico de red y tiene la expectativa de respuesta inmediata.

3.3.1.3. Variedad

Existen nuevos tipos de información no estructurada. Los tipos de información no estructurada y la semi-estructurada, como el texto, audio y video requieren procesamiento adicional derivado del soporte de la metadata. Una vez entendida, la información no estructurada tiene los mismos requisitos de la información estructurada, como agregación, que sea auditable y privada. Surge una mayor complejidad cuando los datos de una fuente conocida cambian sin previo aviso. Frecuentemente o en esquemas en tiempo real cambian, estas son una enorme carga para los entornos transaccionales y analíticos.

3.3.1.4. Valor

La información tiene intrínsecamente valor, pero puede ser que aún no haya sido descubierto. Hay una serie de factores cuantitativos y técnicas de investigación para inferir el valor desde la información, descubriendo mejores precios en las compras, diferencias en el costo de proyecto similares o incluso identificando problemas de la ejecución de las entidades de gobierno antes de que estos sean irreparables.

La tecnología avanza, el costo del almacenamiento y procesamiento de la información ha disminuido de manera exponencial, proporcionando así una abundancia de datos. Sin embargo, la búsqueda de valor también requiere nuevos procesos de descubrimiento que involucran análisis inteligentes y perspicaces por parte de los usuarios y autoridades. El verdadero desafío de Big Data es un ser humano que está aprendiendo hacer las preguntas correctas, el reconocimiento de patrones, hacer suposiciones fundamentales y predecir el comportamiento.

3.3.2. La gran pregunta acerca de Big Data

Las buenas noticias es que todos tienen preguntas acerca de Big Data. Tanto las empresas de negocios y las de tecnologías de la información han tomado riesgos y experimentan y este es un camino sano para el aprendizaje. Se debe tomar un enfoque de arquitectura de gobierno que gestiona la información; que Big Data es un activo de país y necesita ser gestionado desde la alineación del plan de gobierno con la gobernabilidad como un eje integral de la arquitectura de gestión de la información. Este es un enfoque práctico que se sabe que mientras se transforma de una prueba de concepto a una escala funcional se tendrán los mismo problemas que los métodos tradicionales de gestión de la información ya tienen, el saber, los requisitos de habilidades, la gobernabilidad, el rendimiento, la escalabilidad, gestión, integración, seguridad y acceso.

3.3.3. ¿Cuál es la diferencia sobre Big Data?

Big Data introduce nueva tecnología, procesos y habilidades a la arquitectura de información y a las personas que la diseñan, operan y usan. Con las nuevas tecnologías, hay una tendencia a separar el nuevo mundo del viejo. Aunque hay excepciones, la expectativa fundamental es que la encontrar patrones en los datos mejora su capacidad para comprender los datos existentes.

A primera vista, las cuatro "V" definen los atributos de grandes volúmenes de datos, pero no son las mejores prácticas en las estrategias de gestión de la información que aseguren el éxito de Big Data. A continuación se presentan algunos puntos clave sobre Big Data:

3.3.4. Un cambio de paradigma en la Arquitectura de la Información

El enfoque de Big Data la estructuración de la información y el análisis son diferentes a los enfoques tradicionales de arquitectura de información. Un enfoque tradicional de *data warehouse* espera la información a través de la estandarización por medio de procesos de extracción, transformación y carga (*ETL* por sus siglas en inglés) y eventualmente será colocada en esquemas predefinidos, esto es conocido como "esquema en la escritura". Una crítica a este enfoque tradicional son los largos procesos requeridos para hacer los cambios a un esquema predefinido. Un aspecto del Big Data es que la información es capturada sin requerir de una estructura "definida".

La estructura se puede inferir de la misma data o a través de procesos algorítmicos, también conocidos como “esquema en la lectura”. Este enfoque es soportado por la nueva arquitectura de hardware y software de bajo costo, en memoria y procesos en paralelo, como por ejemplo *HDFS* de Hadoop y *Spark*.

El concepto principal de este nuevo enfoque se basa en el mejor aprovechamiento de dividir el problema en dos partes, la del como almacenar la información y de cómo procesarla para su consulta. Al no tener que realizar complejas operaciones al momento de almacenarla los procesos se agilizan y se simplifican y se puede almacenar diferentes tipos de información e incluso la información externa puede ser resguardada sin mayor esfuerzo. Al momento de consultarla para convertirla en información, luego en conocimiento se tienen procesos en los que se puede distribuir cargas de procesamiento a través de una infraestructura de red de servidores dedicados analizar pequeños segmentos de todo el universo de información para finalmente solo hacer una agregación de información ya procesada.

A todo esto se pueden hacer procesos bajo demanda, en los que se puede tener una infraestructura de red flexible en la que el número de nodos de información que procesen consultas puede variar en el tiempo, esto en dependencia del poder de procesamiento y del número de usuarios que la necesiten. Por ejemplo, al final de cada mes en el Minfin se tienen una sobre cargad el sistema por el registro de transacciones de compra por parte de los usuarios.

Debido a los grandes volúmenes de información, Big Data también emplea el principio de “tener las capacidades analíticas para la información” en comparación con los procesos tradicionales de “traer los datos a las capacidades analíticas a través de procesos de extracción, transformación y carga”, eliminando así el alto costo de mover grandes volúmenes de información.

3.3.5. La unificación de la información requiere gobernabilidad

La combinación de Big Data con los datos tradicionales añade un contexto adicional y proporciona la oportunidad de ofrecer una mejor percepción de la información. En el siguiente ejemplo se tiene la captura de la información de una compra realizada por una entidad de gobierno que posee un valor, pero cuando se logra cruzar la información de ese mismo tipo de compras por muchas entidades de gobierno se puede inferir un valor promedio de compra del bien o servicio, esto hace que la información tenga mucho más valor.

Por lo tanto se tiene la responsabilidad de alinear los tipos de datos dispares y certificar la calidad de la información, independientemente de su fuente, esto también es conocido como linaje de la información.

3.3.6. Los grandes volúmenes de información continúan creciendo

Una vez alcanzada la implementación de la tecnología de Big Data, es un hecho que el volumen de datos va a seguir creciendo, tal vez incluso de manera exponencial.

En la planificación de rendimiento, más allá del estimación de los conceptos básicos, tales como el almacenamiento de puesta en escena, el movimiento de datos, transformaciones y el procesamiento de análisis, se debe pensar acerca de si las nuevas tecnologías pueden reducir la latencia, como el procesamiento en paralelo, aprendizaje automático, procesamiento en memoria, indexación de columnas y algoritmos especializados.

También es útil distinguir qué datos pueden ser capturados y analizados en un servicio en la nube frente a los almacenados en sitio.

3.3.7. Seguridad de la Big Data

La tecnología de Big Data requiriere de los mismos principios y prácticas como el resto de la arquitectura de información. El manejo de la seguridad busca centralizar accesos, autorizar recursos y prácticas de auditoria. Un punto de inicio en la estrategia de seguridad de Big Data es alinear las prácticas y políticas que ya se tienen establecidas, eliminar las duplicidades de implementaciones y la administración centralizadas a través de todos los ambientes en donde se encuentre almacenada la información.

3.3.8. El proceso de descubrimiento de la Big Data

Se ha comenzado mencionando que el volumen, la velocidad, la variedad y el valor definen lo que es Big Data, pero la característica más sobresaliente de la tecnología de Big Data es el proceso en el cual se descubre el valor de la información.

Big Data es diferente a la inteligencia de negocios convencional, donde el reportar un valor simple conocido revela un hecho, como la suma de la ejecución presupuestaria de una entidad hasta la fecha, esto en Big Data cambia, el objetivo es ser lo suficientemente inteligente como para descubrir patrones, modelo de hipótesis y poner a prueba las predicciones que se pueden inferir de la misma información.

Por ejemplo el valor se descubre a través de una investigación, consulta iterativa y/o proceso modelado, como pedir una pregunta, hacer una hipótesis, elegir fuentes de información, crear modelos estadísticos, visuales o semánticos, evaluar resultados, hacer más preguntas, hacer una nueva hipótesis y luego iniciar de nuevo el proceso.

Expertos en la materia de interpretación de visualizaciones o expertos en realizar consultas basadas en conocimiento interactivo pueden ser auxiliados mediante el desarrollo de "*machine learning*" aprendizaje de las máquinas, que son algoritmos adaptables que pueden descubrir aún más el significado de la información. Si su objetivo es mantenerse al día con el volumen de datos que le rodea, se dará cuenta de que las investigaciones de grandes volúmenes de información son continuas. Y sus descubrimientos pueden resultar en decisiones o pueden a llegar a ser las nuevas prácticas óptimas y se incorporarán en los procesos operativos del gobierno.

El punto de la arquitectura es que los procesos de descubrimiento y de modelado deben ser rápidos e iterativos. Muchas innovaciones tecnológicas recientes permiten estas capacidades y deben ser consideradas, tales como servidores con grandes cantidades de memoria *RAM* para funcionar en modelos de caché, procesamiento, almacenamiento, redes rápidas de optimización, indexación de columnas, visualizaciones, aprendizaje automático, análisis semántico por mencionar unos pocos. El objetivo debe ser descubrir y predecir rápidamente.

3.3.9. Información no estructurada y calidad de la información

Mientras la variedad ofrece flexibilidad, también requiere una atención adicional para entender los datos, posiblemente limpiar y transformar la misma, proporcionar linaje y con el tiempo asegurar que los datos continúan con sentido de lo que se espera que signifiquen. Existen tantos manuales como técnicas para mantener la calidad de los datos no estructurados de forma automática. Ejemplos de archivos no estructurados: un archivo *XML* con un acompañamiento de las declaraciones de esquemas basados en texto, archivos de registro basados en texto, texto independiente, archivos de audio y video, archivos de llave valor (tabla de dos columnas sin semántica predefinida).

Para los casos donde el uso de grandes cantidades de fuentes de datos públicas, ya sean estructuradas, semi estructuradas o no estructuradas, los procesos para asegurar la calidad de la información deben ser automatizados. En la industria de consumo, por ejemplo, los comentarios de medios sociales no sólo provienen de fuentes predecibles como su sitio web y Facebook, sino también de medios como comentarios que son recibidos por vía telefónica. En algunos casos, el "*machine learning*" puede ayudar a mantener vigentes los esquemas de información.

4. ARQUITECTURA DE SOFTWARE

En esta parte se definirán las herramientas, que serán, a través de las cuales se logrará la implementación de la estrategia de Big Data para el manejo de la información presupuestaria del Minfin. Esta arquitectura tendrá como punto focal Hadoop, un producto de la fundación de software Apache, que es el uso del Sistema de Archivos Distribuidos Hadoop (*HDFS* por sus siglas en ingles).

4.1. ¿Qué es Hadoop?

Hadoop es un marco de trabajo basado en código abierto que permite almacenar información y ejecutar aplicaciones en clústeres de hardware. Provee almacenamiento masivo para cualquier tipo de información, enorme poder procesamiento (en dependencia del hardware donde se encuentre instalado) y la habilidad de manejar un número virtualmente infinito de tareas o trabajos concurrentes.

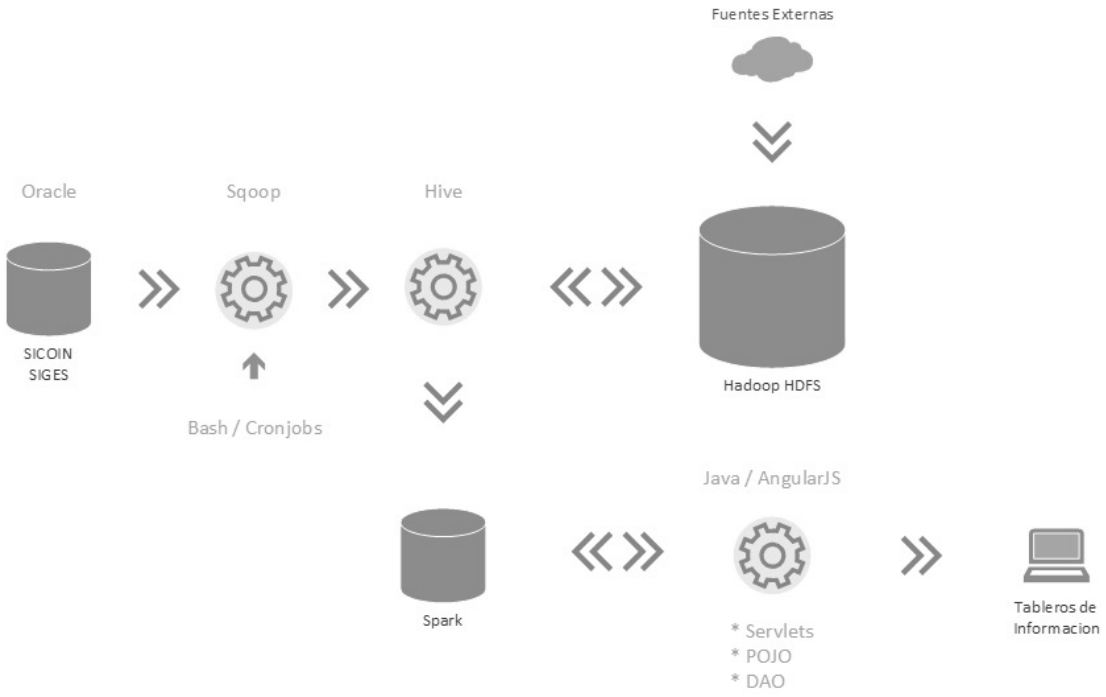
4.2. ¿Qué es HDFS?

De las siglas en inglés, es un sistema de archivos distribuido que está contenido dentro del proyecto Hadoop. Este sistema de archivos basado en *Java* puede proveer escalabilidad y confiablidad en el almacenamiento de la información. Fue diseñado para ser instalado en grandes clústeres de servidores. El *HDFS* ha demostrado un poder de escalabilidad de hasta 200 *Pentabytes* de almacenamiento en un clúster de 4500 servidores, soportando cerca de un billón de archivos y bloques. Cuando esa cantidad y calidad de la información es de gran importancia la opción de *HDFS* es capaz de resolver problemas que las estrategias tradicionales no han logrado resolver con la eficiencia y eficacia esperada.

HDFS es escalable, tolerante a fallos, almacenamiento distribuido que funciona de cerca con una gran variedad de aplicaciones con accesos concurrentes, coordinados por el orquestador de trabajos *YARN*. *HDFS* puede funcionar bajo una gran variedad de circunstancias físicas y sistemáticas. Mediante la distribución del almacenamiento y computación a través de muchos servidores, el recurso de almacenamiento combinado puede crecer linealmente.

El incrementar la capacidad de procesamiento o de almacenamiento es un problema de resolución trivial cuando se emplea *HDFS*, ya que al requerir de más espacio en un sistema distribuido se traduce a incrementar el número de nodos en el clúster, incrementando inmediatamente la disponibilidad de espacio o de poder de procesamiento. El que toda la infraestructura de sistema de archivos distribuidos utilice los nuevos recursos es tan sencillo como hacer de conocimiento de los demás nodos el nuevo nodo en cuestión y automáticamente pasará a ser parte del sistema total.

Figura 8. **Arquitectura de *Software* propuesta para la inteligencia de negocios**



Fuente: elaboración propia.

En las siguientes páginas se definirá cada componente de la arquitectura de software que enumerará un listado de herramientas de Big Data que proveerán el marco de trabajo que será la base para el desarrollo de las herramientas de inteligencia de negocios para la toma de decisiones cumpliendo los fundamentos de las cuatro “V”, volumen, velocidad, variedad y valor. De esta manera se contará con herramientas que proveen un análisis de cómo se ejecuta el gasto del gobierno de Guatemala.

Además se proveerá de la interoperabilidad que tendrán las distintas herramientas propuestas que se encargarán de organizar las dos tareas principales de la estrategia de Big Data, el almacenamiento de la información y la consulta de la misma. Eso se expondrá a la arquitectura adaptable a los sistemas ya existentes dentro del Minfin.

La arquitectura de software deberá ser soportada por una organización de servidores que se detallarán en la arquitectura de hardware. Estos deberán de proveer el medio de comunicación, poder de procesamiento y medio de almacenamiento para el marco de trabajo.

4.3. Arquitectura de Hardware

Al definir la arquitectura ideal que deberá de crearse para la implementación de una arquitectura de información basada en Big Data, se deberán de tomar en cuenta algunas variables que existen o son deseables en la nueva disposición de acceso a la información. A continuación se detallan dichas variables:

4.3.1. Tamaño actual de la información

Actualmente el Minfin tiene un volumen de información de 120 *TB* que está almacenado en un sistema de *Storage Area Network* (*SAN*) que funciona de repositorio de información para los sistemas principales de la iniciativa SIAF, que son los sistemas Sicoin y Siges. Se deberá de partir de una infraestructura que pueda contener y procesar estos volúmenes de información inicial.

Tabla V. **Tamaños de actuales y tasas de crecimientos**

Base de Datos	Tamaño actual	Crecimiento diario/mensual	Observaciones
Sicoin gobierno central	122,51 GB	82,29 MB / 2,41GB	Guarda sólo los últimos tres años de gestión
Sicoin descentralizado	81,73 GB	52,90 MB / 1,55 GB	Guarda sólo los últimos tres años de gestión
Siges	346,59 GB	178,73 MB / 5,24 GB	Por modelo de datos es complicado descargar años de gestión
Replica Sicoin gobierno central	122,51 GB	82,29 MB / 2,41 GB	Para reportería
Replica Sicoin descentralizado	81,73 GB	52,90 MB / 1,55 GB	Para reportería
Replica Siges	346,59 GB	178,73 MB / 5,24 GB	Para reportería
Guatecompras	585,37 GB	254,13 MB / 7,44 GB	
Replica Guatecompras	585,37GB	254,13 MB / 7,44 GB	Para reportería
Guatenóminas	184,03 GB	107,72 MB / 3,16 GB	
Servicios GL	147,19 GB	98,68 MB / 2,89 GB	
Sicoin GL	403,75 GB	244,87 MB / 7,17 GB	
Replica de SicoinGL y servicio GL	550,94 GB	343,55 MB / 10,06 GB	
Histórico	1 750,98 GB	95,78 MB / 2,81 GB	
Servicios (CATSER)	18,32 GB	9,21 MB / 271 MB	
Total	5 327,61 GB		

Fuente: elaboración propia.

En resumen:

- El volumen de datos que generan las aplicaciones del SIAF es de aproximadamente 60 *Gigabytes* mensuales.

- La capacidad total de almacenamiento de la infraestructura del centro de datos del Minfin en su *Storage Area Network (SAN)* es de 160 *Terabytes* lo que es equivalente a 163 840 *Gigabytes*.

- De los 160 *Terabytes* disponibles, se destinan aproximadamente 120 *Terabytes* para las aplicaciones en producción del SIAF y 30 *Terabytes* corresponden a el almacenamiento de copias de seguridad (estos discos se encuentran completos y periódicamente se bajan a cinta para liberar nuevo espacio de copias de seguridad).

- Los 120 *Terabytes* que conforman la capacidad de almacenamiento de los sistemas en Producción del SIAF y otras aplicaciones del Ministerio de Finanzas ocupan un 90% de la capacidad establecida, es decir que se cuenta con aproximadamente 12 *Terabytes* disponibles para crecimiento.

4.4. Desafíos de la arquitectura actual

4.4.1. Uso de colas para las transacciones

Todas las aplicaciones que conforman el SIAF interactúan e intercambian información con la aplicación Sicoin que es la responsable de almacenar la estructura presupuestal, el presupuesto formulado y la ejecución del mismo.

El Sicoin se encuentra con muy pocas o nulas modificaciones de su arquitectura base en relación a su concepción original tanto a nivel de modelo de datos como a nivel de aplicaciones, como se demuestra con versiones en producción de módulos que datan del año 2005.

Los equipos de desarrollo han optado por el diseño e implementación de nuevas funcionalidades, principalmente aquellas que buscan darle a los sistemas una orientación a gestión de procesos, en arquitecturas diferentes a la de Sicoin. Esta situación ha llevado a que algunos sistemas, Siges en particular, que es el sistema que ha incorporado funcionalidades que conceptualmente deberían estar alojadas en Sicoin (tomando en consideración que el SIAF vigente no es un sistema orientado a procesos más allá de la intensión del Siges de transformarlo en ello) incorporen nuevas funcionalidades para las que no fueron originalmente diseñados generando un impacto no debidamente estimado en la actualización de datos en la base transaccional del Sicoin.

Lo que se puede determinar es que el Siges es un sistema orientado a procesos en el que toda acción es identificada bajo el concepto de gestión. Toda gestión se identifica a través de un número dentro de su arquitectura funcional y un seguimiento por etapas y actividades de cada una de las gestiones iniciadas. El problema principal es que toda gestión que se realice ha de terminar de una u otra manera en una o más transacciones en Sicoin que es quien ejecuta los procesos finales que afectan el presupuesto público.

Esto particularmente genera un cuello de botella en los requerimientos al Sicoin, porque se ha de tener en cuenta que el Sicoin maneja el concepto de "CUR"[10] y no de gestión, una de las funciones del CUR es la de reservar registros que han de ser afectados (a nivel de Unidad ejecutora), como la regla indica que en el registro secuencial del CUR no puede perderse la secuencia en la generación de un nuevo CUR no se habilita hasta que la transacción anterior haya finalizado. Por esta razón las transacciones se van encolando y frente al riesgo de percepción de "sistema colgado" del usuario final se ha implementado un sistema de colas de requerimientos (*Rabbit*) que permite al usuario seguir operando el sistema mientras que la aplicación *Rabbit* hace el seguimiento de la gestión de su petición al Sicoin. Esto claramente es una mejora en la percepción del usuario final aunque en realidad no se ha instrumentado una solución estable al problema.

El uso de colas por parte de las aplicaciones actuales plantea una problemática a solucionar en las próximas versiones SIAF, en el planteamiento de la arquitectura Big Data no se tomará en cuenta este uso, dado que la carga principal de información será desde las bases de datos transaccionales de los sistemas actuales. Sin embargo es una realidad que esto debe de ser replanteado una vez que se hayan establecido las técnicas a seguir para el uso de grandes volúmenes de información.

[10] CUR Código Único de Registro, que es un número único asignado a cada transacción de la entidad.

4.5. Infraestructura actual

La infraestructura actual de servidores que están ubicados en el Minfin es de 36 servidores virtuales, los cuales dan soporte de procesamiento a los sistemas actuales. Este "*farm*"[11] de servidores los cuales se encuentran con una arquitectura de *Microsoft Windows* y dan soporte a base de datos a través del *RDBMS Oracle*.

A continuación se muestra una tabla de todos los servidores y de que sistema actualmente contienen:

[11] Colección de servidores que sirven para proveer capacidades de servidores como si se tratase de solo una máquina. Los *farm* de servidores son usualmente usado para clústeres de procesamiento.

Tabla VI. **Servidores para los sistemas principales de SIAF**

Sistema	Servidores	Características	Sistema Operativo
Sicoin y Siges	4 servidores de aplicación	Máquina virtual	*Windows Server* 2008 r.2 estándar
	4 servidores de presentación	Máquina virtual	*Windows server* 2003 Standard Edition
	3 servidores para reportes	Máquina virtual	*Windows server* 2003 Standard Edition
Guatenóminas	2 servidores de presentación	Máquina virtual	*Windows server* 2003 Standard Edition
	2 servidores de aplicación	Máquina virtual	*Windows server* 2003 Standard Edition
	1 servidor de reportes	Máquina virtual	*Windows server* 2003 Standard Edition
Sicoin GL	5 servidores	Máquina virtual	*Windows Server* 2008 r.2 estándar
Servicios GL	6 servidores	Máquina virtual	*Windows Server* 2008 r.2 estándar
Guatecompras	2 servidores de presentación	Máquina virtual	*Windows Server* 2008
	2 servidores de aplicación	Máquina virtual	*Windows Server* 2008
	1 servidor de archivos	Máquina virtual	*Windows Server* 2008
	1 servidor de procesos	Máquina virtual	*Windows Server* 2008
	1 servidor espejo de BD	Máquina virtual	*RedHat Enterprise Linux Server* r.5.6
	1 servidor BD transaccional	Máquina virtual	*RedHat Enterprise Linux Server* r.5.6
	1 servidor de control de versiones	Máquina virtual	*Windows Server* 2008

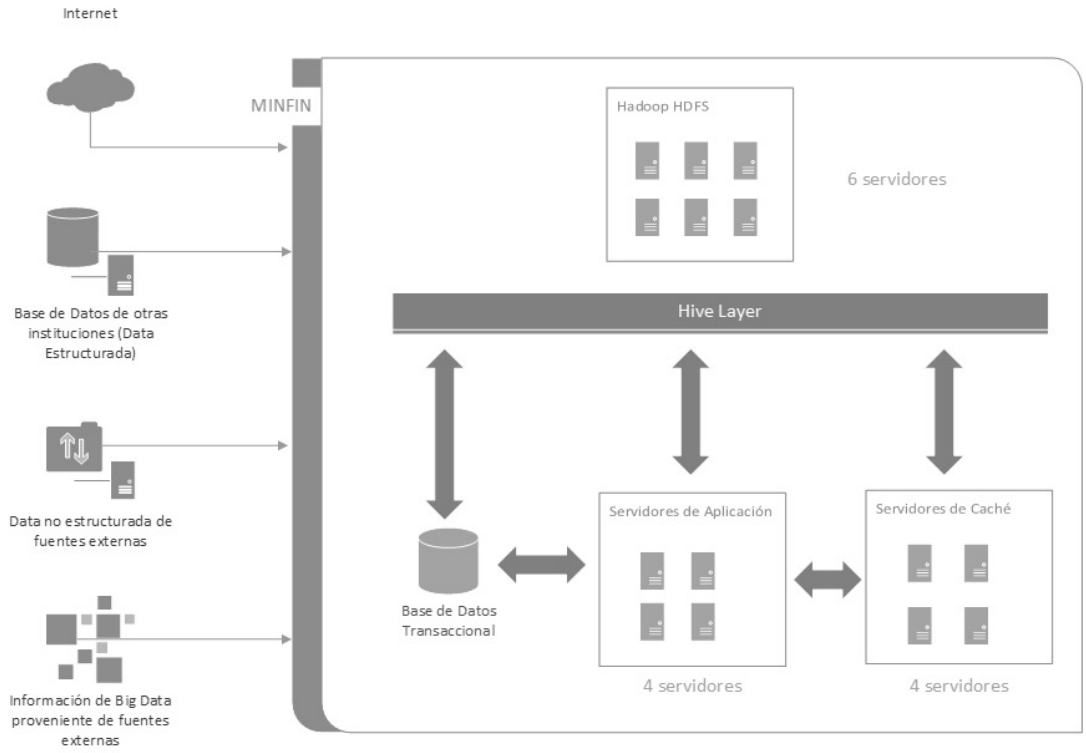
Fuente: elaboración propia.

La tabla anterior muestra la información actual de los sistemas SIAF que son vitales para el manejo de la contabilidad del Estado de Guatemala. Como se puede visualizar es soportada en su gran mayoría pos sistemas operativos Windows.

4.6. Propuesta de Arquitectura de Big Data

Se hará uso de *HDFS* en su versión utilizada por Hadoop para la implementación de la estrategia de Big Data en el Minfin. La arquitectura propuesta será basada en las fortalezas de la distribución de la información a través de un "clúster"[12] de servidores que proveerán tanto el resguardo de la información como el poder de procesamiento.

Figura 9. **Arquitectura de *Hardware* propuesta para la inteligencia de negocios**



Fuente: elaboración propia.

[12] Un clúster es un conjunto de servidores que proveen poder de almacenamiento y/o procesamiento a los sistemas informáticos.

La figura 9 muestra la propuesta de arquitectura de hardware para la implementación de la inteligencia de negocios para el Minfin.

Detallando a continuación los componentes involucrados:

- 6 Servidores de Hadoop *HDFS.*
- 4 Servidores de Aplicación.
- 4 Servidores de Cachés.

4.6.1. Servidores Hadoop *HDFS*

Servidores que tendrán a su cargo las dos tareas principales de la implementación de Big Data, que son el resguardo de la información y el procesamiento de la consulta de la misma. La funcionalidad de estos se detallará de una manera específica en la definición de la arquitectura de software.

4.6.2. Servidores de Aplicación

Se encargarán de ejecutar el sistema de inteligencia de negocios. Servirán de punto de acceso a la información a través de un sistema web de consulta.

4.6.3. Servidores de Caché

Proveerán de un acceso a gran velocidad a la información por parte del servidor de aplicaciones. Estos usarán la información en memoria *RAM* para poder tener tiempos casi instantáneos de acceso a la información.

Este *farm* de servidores será la base de la arquitectura de software del sistema de inteligencia de negocios. Que fue definido en un modelo de Modelo Vista Controlador (*MVC*), en la que el modelo será definido por la estrategia de almacenamiento de información de Big Data y que será consultada vía web.

5. MARCO DE TRABAJO

5.1. Hadoop

Como se ha definido anteriormente en el capítulo 4 en la sección de arquitectura de software, se procederá directamente a definir cuáles son los beneficios que hace seleccionar a esta tecnología como la estrategia a seguir para la implementación de Big Data. A continuación se mencionan los beneficios más sobresalientes que se obtienen en las versiones actuales de Hadoop, (la presente propuesta se basa en la versión 2.7.2).

5.1.1. Beneficios de Hadoop

5.1.1.1. Escalabilidad y Rendimiento

El procesamiento distribuido de la información de forma local en cada nodo dentro de un clúster, habilita a Hadoop para almacenar, administrar, procesar y analizar información a una escala de *Petabytes*.

5.1.1.2. Confiabilidad

Grandes clústeres de poder de procesamiento son propensos a fallas de nodos individuales en el clúster, es decir mientras mayor sea el número de servidores involucrados en el procesamiento de la información, mayor será la probabilidad de que alguno de estos falle. Hadoop es fundamentalmente flexible, cuando un nodo falla en el procesamiento es re direccionado a los nodos restantes en el clúster y la información es automáticamente replicada en preparación a posibles nuevos fallos.

5.1.1.3. Flexibilidad

A diferencia de los sistemas de administración de bases de datos relaciones tradicionales, no se deben de crear esquemas para la información antes de que esta sea almacenada. Se puede almacenar información que carece de formato, incluyendo información parcialmente estructurada o no estructurada, luego se puede aplicar un esquema cuando la información es leída. Esto ofrece un alto desempeño al momento de almacenar grandes volúmenes de información, ya que no se necesita realizar ningún cambio en la información como tarea previa a ser almacenada.

5.1.1.4. Bajo Costo

En comparación con los softwares propietarios, Hadoop es de código abierto lo que implica cero inversión en licenciamiento, así como también se ejecuta en equipo de hardware de bajo costo. No requiere grandes inversiones para implementarlo.

5.1.2. Componentes principales de Hadoop

Hadoop maneja una base principales de componentes que dan soporte a las tareas principales de la tecnología de Big Data. Estos componentes son cuatro y definen el corazón de Hadoop, la forma en la que este almacena la información de forma distribuida, el procesamiento y consulta de la información y finalmente el orquestador del acceso a los recursos que provee Hadoop.

5.1.2.1. Hadoop Common

Es el grupo de librerías y utilidades usadas por otros módulos de Hadoop.

5.1.2.2. Sistema Distribuido de Archivos Hadoop (HDFS)

Sistema de archivos distribuidos basados en *Java* que puede ser escalado a través de múltiples computadoras sin una organización previa.

5.1.2.3. MapReduce

Originalmente presentado en un artículo de la empresa Google en 2004 en el que se definían las dos tareas principales de *Map*() y *Reduce*() para el manejo de grandes volúmenes de información. No fue sino hasta el 2006 que se adoptó por el proyecto Hadoop que ya pertenecía a la fundación de software Apache.

Es un marco de trabajo en sí mismo, diseñado para escribir aplicaciones que tienen que procesar grandes cantidades de información. *MapReduce* es originalmente un marco de trabajo para escribir aplicaciones que requiere procesar grandes cantidad de información estructurada y no estructurada que se encuentre almacenada en el *HDFS* de Hadoop.

MapReduce es muy útil en el procesamiento de información por lotes de terabytes o *petabytes* de tamaño.

Un programa de *MapReduce* se compone de un procedimiento *Map*() que hace la tarea de filtrado y ordenamiento y una método *Reduce*() que realiza la tarea de agrupación y agregación de la información. El sistema del *MapReduce* orquesta el procesamiento por clasificación de los servidores distribuidos que contienen la información, ejecutando varias tareas en paralelo, administrando todas las comunicaciones y transferencias de información entre todas las partes del sistema y proveen redundancia y tolerancia a fallos.

Entre las características principales de *MapReduce* están:

5.1.2.3.1. Simplicidad

Los desarrolladores pueden escribir aplicaciones en su lenguaje de preferencia, como *Java*, *C++* o *Python*, por lo que los trabajos de *MapReduce* son sencillos de ejecutar.

5.1.2.3.2. Escalabilidad

MapReduce puede procesar *petabyates* de información, almacenada en el *HDFS* en un clúster o en varios.

5.1.2.3.3. Velocidad

El procesamiento en paralelo significa que *MapReduce* puede resolver problemas que tradicionalmente pueden tomar días de procesamiento en unas horas o incluso en minutos.

5.1.2.3.4. Recuperación

MapReduce se encarga de la fallas. Si una máquina con una copia de la información no está disponible el trabajo se asignará a otro nodo que contenga una copia de la información que se desea procesar.

5.1.2.3.5. Movimiento mínimo de información

MapReduce mueve los procesos computacionales a donde se encuentra la información dentro del *HDFS* y no al revés. Procesar tareas puede ocurrir físicamente en el nodo donde reside la información. Esto reduce significativamente los patrones de entrada y salida de la red lo que contribuye a la velocidad de procesamiento que tiene Hadoop.

5.1.2.4. Yarn

De sus siglas en ingles *YARN* es el acrónimo para Aún Otro Negociador de Recursos (*Yet Another Resources Negotiator*) (Arun Murthy, 2014). Originalmente descrita por la fundación de software Apache como un rediseño al administrador de recursos, Yarn ahora está caracterizado por ser un sistema de gran escala, para sistema distribuidos para aplicaciones de Big Data.

En 2012, Yarn se convirtió en un sub proyecto de la fundación de software Apache de Hadoop. En ocasiones llamado *MapReduce* 2.0, *YARN* es una nueva escritura de software que desacopla las capacidades de gestión de recursos y la programación de *MapReduce* desde el componente de procesamiento de datos, lo que permite a Hadoop apoyar enfoques de tratamiento más variados y una gama más amplia de aplicaciones. Por ejemplo, los clústeres de Hadoop ahora pueden ejecutar consultas de información interactivas y aplicaciones de toma de información de manera simultánea haciendo trabajos por lotes a través de *MapReduce*.

La idea fundamental de Yarn es separar las dos mayores responsabilidades de un rastreador de trabajo, en otras palabras, administrar y calendarizar/monitorear en dos procesos separados: un controlador de recursos (*RM* por sus siglas en inglés) global y un master de aplicaciones (*AM* por sus siglas en inglés).

Yarn provee de nuevas funcionales para los nuevos componentes del flujo de trabajo de Hadoop. Estos componentes dan un control granular para los usuarios y al mismo tiempo ofrece capacidades más avanzadas el uso de los recursos para las cargas de trabajo, todo esto en el "ecosistema de Hadoop" [13].

[13] Ecosistema de Hadoop se refiere al conjunto de marcos de trabajo, aplicaciones y librerías aplicadas a las distintas fases de almacenamiento y consulta de Hadoop

5.1.2.4.1. Controlador de Recursos

Como se ha mencionado, Yarn es un controlador de recursos basado en la calendarización pura de los trabajos a ejecutar. Está estrictamente limitado a los recursos disponibles en el sistema y las aplicaciones compiten por obtener estos recursos. Está optimizado para ser utilizado en clústeres (mantiene todos los recursos en uso todo el tiempo) a través de varias restricciones como: garantizar la capacidad, asignación equitativa y "*SLA*"[14].

5.1.2.4.2. Master de Aplicaciones

Un concepto nuevo muy importante en Yarn es el master de aplicaciones, este es en efecto una libraría que utiliza una instancia de un marco específico de trabajo para negociar recursos manejados por el controlador de recursos (*RM*).

Adicional a los cuatro componentes que representan el núcleo de Hadoop, se definen una serie de componentes adicionales que se acoplan perfectamente a la implementación de Big Data, dadas las características de los sistemas ya implementados en el Minfin.

[14] Del inglés *Service-Level Agreement*, que es un contrato a nivel de servicio para garantizar que se le asignaran servicios de ejecución

5.2. Hive

Un grupo de programadores de Facebook desarrollaron una estructura de soporte para Hadoop que permitía a cualquier persona que ya tuviese conocimiento fluido de SQL (que es algo muy común para los desarrolladores de bases de datos relacionales) pudiese aprovechar mejor la plataforma de Hadoop sin necesidad de aprender un lenguaje nuevo de consulta de información. Esta creación fue llamada Hive, permite que desarrolladores de SQL puedan escribir declaraciones en el lenguaje de consultas de Hive (*HQL* por sus siglas en ingles) que son muy similares a las declaraciones del SQL estándar. Las declaraciones en *HQL* son descompuestas por el servicio de Hive para ser ejecutadas como trabajos de *MapReduce* para ser ejecutados a través del clúster de Hadoop.

Para cualquier persona con conocimientos previos en SQL o de bases de datos relacionales, el uso de Hive le parecerá muy familiar. Como cualquier sistema de administración de base de datos relaciones, Hive puede ejecutar consultas de muchas formas. Pueden ser ejecutados a través de una línea de comandos (conocido como consola de Hive), desde una conexión *JDBC* o desde una conexión abierta a la base de datos (*ODBC*) todo esto desde el uso de controladores *JDBC/ODBC*, o puede ser utilizado el cliente Hive *Thrift*. El cliente Hive *Thrift* es como cualquier cliente de base de datos que se instala en una máquina cliente y permite el acceso a realizar consultas y modificaciones a la estructura de la información ("*DDL* y *DML*"[15]).

Hive se parece bastante al código tradicional que se utiliza para el acceso con *SQL* a una base de datos. Sin embargo, como Hive está basado en Hadoop se basa en las operaciones de *MapReduce* y estas tienen varias diferencias.

[15] DDL Lenguaje de definición de datos y DML Lenguaje de manipulación de datos, ambos de sus siglas del inglés.

- Hadoop está diseñado para largas y extensas búsquedas secuenciales y por esto las consultas a través de Hive puede llegar a tardar varios minutos. Lo que significa que Hive puede no ser apropiado para aplicaciones que requieren tiempos de respuesta cortos, como los esperados en las bases de datos tradicionales.

- Hive es basado y optimizado para la lectura por lo que no puede ser apropiado para procesar transacciones que típicamente involucran altos porcentajes de operaciones de escritura.

Hive a su vez es soportado por una serie de componentes que permiten su buen funcionamiento, a continuación se describirá el más importante.

5.2.1. HCatalog

Es un sistema de administración de metadata y estructuras de tablas para la plataforma de Hadoop. Permite el almacenamiento de la información en cualquier formato que carezca de estructura. Hadoop puede procesar ambos, los estructurados y los no estructurados y puede almacenar y compartir información sobre la información estructurada en *HCatalog*. Esta capacidad combinada con la naturaleza de Hadoop de “esquema en la lectura” en contra parte de la tradicional de “esquema en la escritura” reduce los ciclos de tiempo para el acceso de la información y la exploración de la misma.

HCatalog intenta habilitar un ecosistema más general para la interacción de la información almacenada en Hadoop a través de *SQL*.

5.3. HBASE

HBase es la base de datos de Hadoop. Si damos un vistazo a la arquitectura de *Bigtable*, podemos concluir que *HBase* es una implementación en código abierto de la tecnología que Google implemento en su concepción de Big Data (George, 2011). Este concepto fue publicado por Google en un artículo publicado en el año 2006.

Es un sistema de administración de base de datos orientado a columnas que se ejecuta sobre *HDFS*. Se ajusta muy bien para los conjuntos de datos dispersos que son comunes en muchos casos de uso de la tecnología de Big Data. A diferencia de los sistemas de bases de datos relacionales, *HBase* no soporta el lenguaje de consulta estructurado *SQL* de hecho, *HBase* no almacena la información físicamente relacionada. Las aplicaciones de *HBase* son escritas en *Java* que son usualmente *MapReduce*.

Un sistema de *HBase* comprende un conjunto de tablas. Cada tabla contiene filas y columnas, al igual que una base de datos tradicional. Cada tabla debe tener un elemento definido como una llave primaria y todos los intentos de acceso a las tablas deben utilizar esta llave primaria, todo es en cuanto a la visión que el usuario tiene de la información, ya que como se mencionó antes la forma en la que realmente almacena la información no es relacional de ninguna manera.

Una columna representa un atributo de un objeto, por ejemplo, si la tabla es el almacenamiento de registros de diagnóstico de servidores de un entorno dado, donde cada fila podría ser una entrada de registro, una columna típica de una tabla de este tipo sería la marca de tiempo de cuando fue escrito el archivo del registro, o tal vez el nombre del servidor donde el dato se originó. De hecho, *HBase* permite que muchos atributos sean agrupados juntos, en lo que se conoce como las familias de columnas, de tal manera que los elementos de una familia de columnas se almacenen juntas. Se debe definir previamente el esquema de tablas y especificar las familias de las columnas. Sin embargo, es muy flexible en cuanto a que nuevas columnas se pueden añadir a las familias en cualquier momento, haciendo el esquema flexible, por lo tanto es capaz de adaptarse a cambios en los requisitos de una aplicación.

5.4. Tez

MapReduce ha hecho un buen trabajo. Durante muchos años ha sido el motor de procesamiento para Hadoop y ha sido la columna vertebral sobre la cual se ha creado una enorme cantidad de las características que ofrece Hadoop en cuanto a proceso de grandes universos de información. A pesar de que está aquí para quedarse, también se necesitan nuevos paradigmas a fin de que Hadoop pueda servir a un número aún mayor de los patrones tradicionales de uso. Un ejemplo clave y emergente es la necesidad de consultas interactivas de la información almacenada, que hoy se ve desafiada por la naturaleza orientada a lotes de información. Un paso clave para que este nuevo mundo ha sido Hive y hoy en día la comunidad propone el siguiente paso, Tez

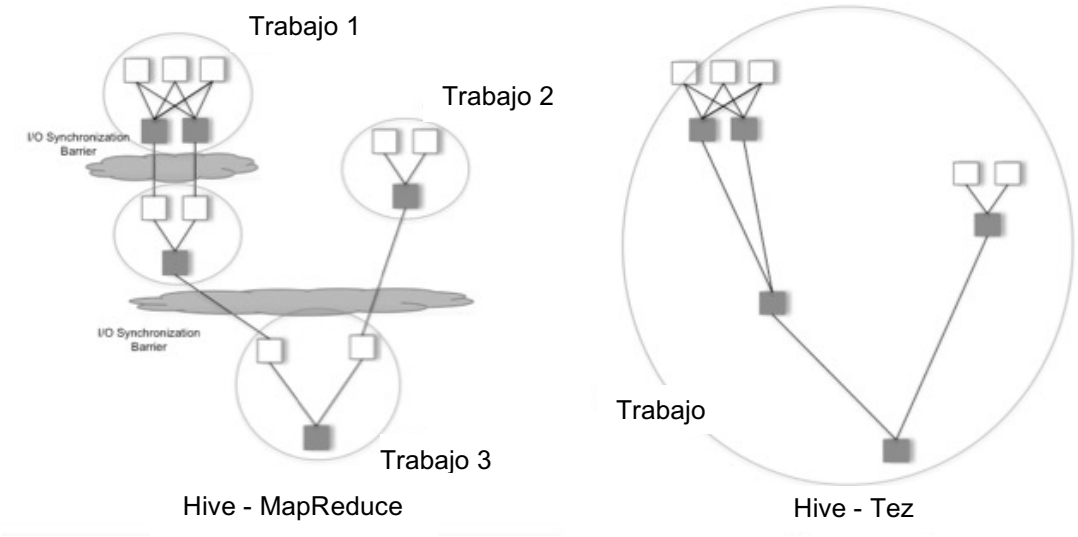
Tez proviene de la palabra en Hindi para velocidad. Provee un propósito general, un marco de trabajo altamente ajustable que crea tareas simples para el procesamiento de la información a través de trabajos a pequeña escala y gran escala en Hadoop. Se generaliza el paradigma de *MapReduce* a un marco de trabajo mucho más poderoso, proporcionando la capacidad de ejecutar un *DAG* complejo (grafo a cíclico dirigido) de tareas para un solo paquete de trabajo para que los proyectos en el ecosistema de Hadoop como Hive, puedan cumplir con los requisitos en tiempo de respuesta interactiva para los usuarios y un rendimiento extremo en escalas de información que alcancen los petabytes (claramente *MapReduce* ha sido un factor clave para lograr esto).

Con la aparición de Yarn como la base de las arquitecturas de procesamiento de datos de próxima generación para Hadoop, existía una fuerte necesidad de una aplicación que pudiese ejecutar un *DAG* de tareas complejas que luego pueden ser compartidos entre Hive y otros. El *DAG* limitado expresado en *MapReduce*, a menudo resultaba en múltiples trabajos de *MapReduce* que perjudicaban la latencia de consultas breves (por encima de lanzamiento de varios grupos de trabajo) y el rendimiento de las consultas a gran escala (demasiado generales para la materialización de los trabajos intermedios de salida al sistema de archivos). Con Tez, se introdujo un *DAG* más expresado en tareas, dentro de una sola aplicación o trabajo, que se ajustan más a la tarea de procesamiento requerida, por lo tanto, cualquier consulta SQL se puede expresar como un solo trabajo utilizando Tez.

Tez es fundamental, un largo camino para ayudar a apoyar a Yarn en los dos tipos de consultas, interactivas y por lotes. Tez proporciona un único marco de referencia base para apoyar tanto la latencia y rendimiento aplicaciones sensibles, no obviando la necesidad de múltiples marcos y sistemas como parte del ecosistema de Hadoop, manteniendo y apoyando, una ventaja clave para racionalizar la arquitectura de datos.

Esencialmente, Tez es el siguiente paso lógico para Hadoop después de Yarn. Con Yarn la comunidad generalizada de *MapReduce* era proporcionar un marco de gestión de recursos de propósito general, donde se convirtió a *MapReduce* en simplemente una de las aplicaciones que puedan procesar los datos en un clúster de Hadoop.

Figura 10. **Comparación entre el plan de consulta de *MapReduce* y Tez**



Fuente: elaboración propia.

Algunos autores destacan que en el resultado de pruebas de rendimiento se observa una mejora de 200% a 300% de disminución en los tiempos de respuesta cuando se consulta la información a través de Hive utilizando los motores de ejecución de trabajo *MapReduce* en contra parte con Tez. La mejora es sustancial y la mejora es tanto en consultas a pequeña y gran escala.

De la figura 10 podemos observar cómo se realiza la simplificación de acceso de los trabajos al ser ejecutados través de *MapReduce* y Tez.

5.5. SQOOP

Sqoop es la manera más eficiente en las transferencias de datos a gran escala entre Hadoop y grandes almacenes de datos estructurados. Transfiere la información de manera eficiente entre sistema de información estructurada, como por ejemplo bases de datos relacionales. Es una aplicación que ayuda a la descarga de ciertas tareas (como el procesamiento *ETL*) de la data *warehouse* de Hadoop para la ejecución eficiente a un costo mucho más bajo en tiempo y uso de recursos de procesamiento.

Además de la tarea de carga de información al *HDFS* de Hadoop, *Sqoop* permite el flujo de información desde Hadoop hacía almacenes de datos estructurados.

Sqoop realiza una serie de acciones cuando hace movimientos de información entre Hadoop y los almacenes de información estructurada. A continuación la lista de acciones:

Tabla VII. **Beneficios del uso de *Sqoop***

Acción	Beneficio
Importación secuencial de los set de datos desde el nodo central	Satisface la necesidad cada vez mayor para mover datos desde el nodo central al *HDFS*
Importación directa de *ORCFiles*	Mejora la compresión y el poco peso de indexación para mejorar el rendimiento en las consultas
Importación de información	Traslada la información desde almacenas externos y data warehouse externos a Hadoop para optimizar el costo-beneficio de la combinación de almacenamiento y procesamiento de la información
Transferencia de información en paralelo	Para un rendimiento mucho mejor en velocidad y la optimización del uso del sistema
Copia de la información rápida	Desde sistemas externos a Hadoop
Análisis de la información eficiente	Mejora la eficiencia del análisis de la información, combinando información estructurada con información no estructurada en un almacén de datos basado en “esquematización en la lectura”
Balance de carga	Mitiga el uso excesivo del almacenamiento y procesamiento utilizando varios nodos dentro del clúster

Fuente: elaboración propia.

5.6. Accumulo

Es un sistema de almacenamiento distribuido de llave-valor con control de acceso a nivel de celda.

Originalmente desarrollado por la Agencia de Seguridad Nacional de Estados Unidos, antes de pasar a ser parte de la fundación de software Apache como un proyecto de incubación. Debido a sus orígenes en la comunidad de inteligencia, provee de acceso extremadamente rápido a información almacenada en tablas masivas, mientras que provee el control de acceso a billones de filas y millones de columnas que bajan a nivel de celdas individuales. Esto es conocido como un control fino-granulado de control de acceso.

El control a nivel de celda es importante para organizaciones con políticas complejas de gobierno de la información de quien puede tener acceso a la misma. Esto permite la mezcla de diferentes conjuntos de datos con las políticas de control de acceso a información sensible. Los usuarios que tienen permiso para ver los datos sensibles pueden trabajar junto a un compañero de trabajo sin esos privilegios. Ambos tipos de usuarios pueden acceder a la información en dependencia de los permisos que posean.

Entre las características principales de Accumulo están:

- Control de acceso a nivel de celda.
- Filas excesivamente grandes no necesitan residir en memoria para su procesamiento.

- Control central contra fallos al utilizar candados con la herramienta Zookeeper.
- Maneja control de archivos de bitácora para recuperación.
- Control maestro de la metadata para hacerlo escalable.
- Ejecución tolerante a fallos.
- Codificación relativa para comprimir llaves consecutivas similares.
- Alto rendimiento para barridos de información a través de hilos corriendo en paralelo.
- Uso de cache para información recientemente consultada.
- Agrupación de columnas en un archivo simple.
- Separación automática de información para realizar balanceo.

5.7. Zookeeper

Zoookeper es coordinador de servicios para aplicaciones distribuidas que es de código abierto. Se expone un conjunto simple de directrices para aplicaciones distribuidas que pueden ser aprovechadas para implementar servicios de nivel superior en su sincronización, mantenimiento de la configuración, grupos y dominios. Está diseñado para ser fácil de programar y utilizar un modelo de datos basado en la estructura de árbol de un sistema de archivos. Se ejecuta en Java y tiene librerías para Java y *C*.

La coordinación de servicios es notoriamente difícil de lograr. Los servicios son propensos a fallar y a errores tales como las condiciones de ejecución y de punto muerto. La motivación detrás de Zookeeper es el aliviar de estos problemas a las aplicaciones distribuidas a cargo de la implementación de servicios de coordinación que se desarrollan desde cero.

De las cuatro cualidades principales de Zookeeper se tiene:

5.7.1. Simplicidad

La coordinación de los procesos entre sí se realiza a través de un espacio de nombres jerárquico común que se organiza de manera similar a un sistema de archivos estándar distribuido. El espacio de nombre se compone de registros de datos, llamados *znodes*, en la jerga Zookeeper y estos son similares a los archivos y directorios. A diferencia de un sistema de archivos típico, que está diseñado para el almacenamiento, los datos Zookeeper se mantienen en memoria, lo que significa Zookeeper puede alcanzar números bajos de latencia, lo que se traduce a un alto rendimiento.

5.7.2. Replicación

Al igual que los procesos distribuidos bajo su coordinación, Zookeeper está replicado a través de un set de servidores llamados conjunto.

Los servidores que componen el servicio Zookeeper todos deben saber el uno del otro. Mantienen una imagen en memoria del estado, junto con los registros de transacciones y las instantáneas en un almacén persistente. Mientras la mayoría de los servidores están disponibles, el servicio Zookeeper estará disponible.

Para los clientes que se conectan a un único servidor Zookeeper, el cliente mantiene una conexión *TCP* a través del cual se envía peticiones, consigue respuestas, consigue ver eventos y envía los latidos del corazón para mantener la comunicación abierta. Si la conexión *TCP* tuviese algún problema, el cliente se conectará a un servidor diferente.

5.7.3. Orden

Se mantiene un registro de cada actualización con un número que refleja el orden de todas las transacciones.

5.7.4. Velocidad

Es especialmente rápido en trabajo “dominantemente de lectura”. Las aplicaciones puede ejecutarse en miles de servidores y el rendimiento será mejor en la lectura que en la escritura, manteniendo relación de 10 a 1, en canto a comparación en la velocidad de operación.

5.8. Spark

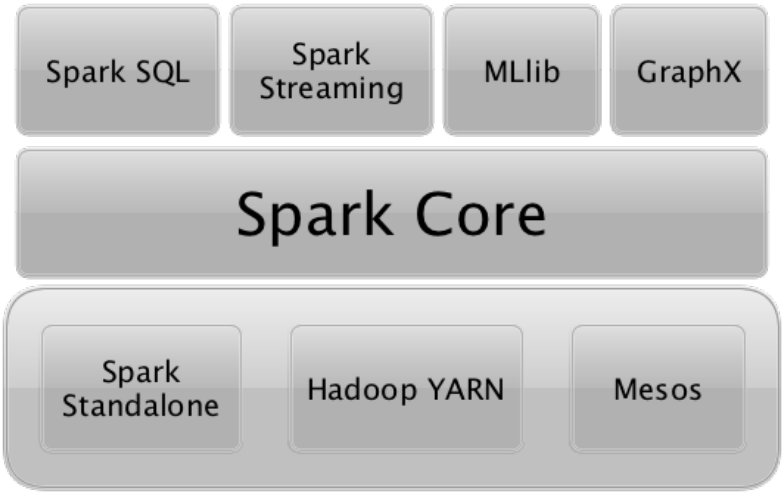
Es un proyecto de código abierto enfocando el desarrollo del motor de procesamiento alrededor de la velocidad de acceso, lectura y escritura de la información, de fácil uso y herramientas sofisticadas para el procesamiento de la información. Originalmente desarrollado en la Universidad de Berkeley en el año 2009.

Desde su lanzamiento, Spark ha visto un rápido crecimiento en su implementación por parte de empresas de un gran número de industrias, como por ejemplo, Netflix, Yahoo, eBay. Las cuales se han dedicado a la implementación en una gran escala, procesando múltiplos de petabytes de información en clústeres de más de 8,000 nodos. Se ha convertido en una extensa comunidad de código abierto enfocada a Big Data, con más de 1 000 contribuyentes a la generación de ideas y de horas de programación que se encuentran en más de 250 organizaciones.

Spark es un marco de trabajo para procesamiento de información y de propósito general, también posee un motor para el procesamiento de información en memoria. Es capaz de hacer tareas como *ETL*, análisis, algoritmos de aprendizaje de computadoras y procesamiento de grandes volúmenes de información en procesos "*batch"*[16] o "procesos en movimiento"[17] que tienen un alto nivel de relación con "IPAs"[18] para lenguajes de programación: Scala, Python, Java, R y SQL.

[16] Procesos en lote, un conjunto de procesos que son ejecutados con la misma prioridad
[17] Procesos de transmisión de información para lectura o escritura
[18] De sus siglas en ingles Interfaz para programación de aplicaciones

Figura 11. **Composición elementos de Spark**



Fuente: elaboración propia.

Se puede describir a Spark como un motor distribuido de procesamiento de información que es capaz de ejecutar procesos por lotes y además hacer movimientos de información con características como consultas en SQL, procesamiento de gráficas y aprendizaje de máquinas.

En contraste de la manera como Hadoop procesa la información en la que se necesitan dos etapas para el proceso de la información en disco, *MapReduce*, Spark ofrece un proceso de la información de etapas múltiples en memoria lo cual provee de un mejor rendimiento. Se ha evidenciado que Spark puede alcanzar hasta 100 veces más velocidad en el acceso a la información de lo que lo puede hacer Hadoop (Xin, 2014).

Tabla VIII. **Comparación de carga de trabajo de Hadoop vs Spark**

	Hadoop *MR*	Spark	Spark
Tamaño	102,5 TB	100 TB	1000 TB
Tiempo de almacenamiento	72 minutos	23 minutos	234 minutos
# de Nodos	2 100	206	190
# de Nucleos	50 400 físicos	6 592 virtualizados	6 080 virtualizados
Velocidad del clúster de discos	3 150 GB/s (estimado)	618 GB/s	570 GB/s
Mide ordenamiento	Si	Si	No
Red	Dedicada, 10 Gbps	Virtualizada (EC2), 10Gbps	Virtualizada (EC2), 10Gbps
Ordenamiento	1,42 TB/min	4,27 TB/min	4,27 TB/min
Ordenamiento/nodo	0,67 GB/min	20,7 GB/min	22,5 GB/min

Fuente *Databriks* Inc.

El objetivo de Spark es apuntar a la velocidad, la facilidad de uso y a realizar análisis interactivos. Además de ofrecer un poder de computación distribuido en clústeres de computadoras funcionando como motor de ejecución.

Spark se basa en una plataforma distribuida de ejecución de aplicaciones de etapas múltiples complejas, en la ejecución de algoritmos de aprendizaje de máquinas y consultas interactivas hechas a la medida. Provee de una abstracción eficiente para uso en memoria de clústeres computacionales llamados conjuntos de datos distribuidos.

Al utilizar el marco de trabajo de Spark se hace una simplificación al acceso del uso de algoritmos de aprendizaje de máquinas y análisis de predicción en forma escalable.

El lenguaje nativo de Spark es "Scala"[19] pero soporta una serie de lenguajes, tales como, Java, Python y R.

Si se tienen grandes cantidades de información y se requiere procesarla en cortas ventanas de tiempo en las que *MapReduce* no puede ofrecerlo, entonces Spark es una alternativa a tomar en cuenta. Las dos principales ventajas son:

- Acceso a cualquier tipo de información a través de cualquier fuente de información.

- Bajos tiempos para el almacenamiento y procesamiento de la información.

[19] Scala es un acrónimo para "Lenguaje Escalable" (por sus siglas en ingles). Es un lenguaje de propósito general. Provee soporte total para la programación funcional.

5.8.1. Hadoop y Spark

Como se ha visto a través de las páginas anteriores, Hadoop es una tecnología de procesamiento para grandes cantidades de información que ha estado presente ya desde hace varios años y ha probado ser la solución elegida para el procesamiento de conjuntos de datos de grandes proporciones. *MapReduce* es una gran solución para procesamientos de una pasada, pero no es muy eficiente en los casos que requieren múltiples pasadas y algoritmos complejos. Cada paso en el procesamiento de la información toma una fase de mapeo (*Map*) y una fase de reducción (*Reduce*) y se necesita convertir, en cualquier caso, a un patrón de *MapReduce* para poder utilizar esta solución.

Los datos de salida de cualquier trabajo, entre cada paso, tienen que ser almacenados en el sistema de archivos distribuido (*HDFS*) antes de que el siguiente paso puede comenzar. Por lo tanto, este enfoque tiende a ser lento debido a la replicación del almacenamiento en disco. Además Hadoop incluye típicamente clústeres que son difíciles de configurar y administrar. También se requiere la integración de varias herramientas para diferentes casos de uso de Big Data.

Si se desea realizar operaciones de consulta de la información que sean complejas, se tiene que hilar una serie de trabajos de *MapReduce* y ejecutarlos en secuencia. Cada uno de estos trabajos es de alta latencia y no pueden iniciar hasta que el trabajo previo haya terminado completamente.

Spark permite a los programadores desarrollos complejos, accesos a la información que requieren varios pasos utilizando grafos a cíclicos directos (*DAG*), permitiendo que diferentes trabajos de consulta puedan trabajar con la misma información.

Spark se ejecuta sobre una infraestructura de sistema de archivos distribuidos de Hadoop, en la que ayuda a proveerle de funciones adicionales y mejoras a las existentes. Provee de soporte para implementar aplicaciones de Spark en un clúster ya instalado de Hadoop, con Spark dentro de *MapReduce* (*SIMR* por sus siglas en ingles).

Actualmente se debe visualizar a Spark como una alternativa al componente *MapReduce* de Hadoop, más que como un reemplazo a todo el ecosistema de Hadoop. No intenta reemplazar Hadoop, intenta ser parte de, para proveer de una solución que pueda administrar diferentes casos de uso y requerimientos de Big Data.

5.8.2. Características de Spark

Spark lleva a *MapReduce* al siguiente nivel con cambios menos costos en el procesamiento de la información y con capacidades parecidas al manejo de la información en memoria y muy cerca del procesamiento en tiempo real, dando esto como resultado que Spark tenga rendimiento muchas veces mejor que otras tecnologías que implementan Big Data.

Spark también soporta la evaluación de llamadas a demanda de consultas de grandes cantidades de información, lo que ayuda a optimizar los diferentes pasos involucrados en los flujos de trabajo para el procesamiento. También provee un alto elevado de interfaces de programación para aplicaciones, lo que hace una mejora en la productividad del programador y esto genera un modelo de arquitectura más consistente con las soluciones de Big Data. (Laskowski, 2015)

Una de sus características más relevantes es que puede mantener los resultados en memoria sin tener necesidad de escribirlos a disco, lo cual es muy útil cuando se necesita realizar muchas operaciones sobre el mismo set de información. Es importante resaltar que Spark es capaz de realizar operaciones en disco cuando la información no se puede ajustar a la memoria "*RAM*"[20]. Está diseñado para ser un motor de ejecución que trabaja tanto en memoria *RAM* como en disco. Spark también puede ser utilizado para procesar grandes conjuntos de información que se convierten en un proceso de agregación en memoria a través de todo el clúster de Hadoop.

Spark tiene preferencia por almacenar la información en memoria *RAM* y luego desbordar a disco la que ya no logre colocar en *RAM*. Puede almacenar parte del conjunto de información en memoria y el restante a disco, dando como resultado que se tiene que tener la consideración de los requerimientos de memoria para los casos donde se utilice. Cuando Spark utiliza el almacenamiento en memoria es cuando realmente demuestra las ventajas que brinda sobre otras tecnologías conocidas, como por ejemplo, *MapReduce* o Tez.

[20] Dado que hoy en día se tienen grandes capacidades de memoria *RAM* en la mayoría de sistemas esto es poco frecuente pero cabe mencionar que Spark posee también esta flexibilidad.

Otras características que Spark brinda son:

- Ofrece un mayor número de funciones más allá de solo dos funciones *Map* y *Reduce*.

- Optimiza operadores gráficos.

- Evaluaciones a demanda en todo procesamiento de información lo que mejora mucho el proceso del flujo de trabajo.

- Provee de librerías con las mismas herramientas para lenguajes como Scala, Java y Python.

- Ofrece una consola interactiva para Scala y Python. Aún no está disponible la consola para *Java* que se encuentra en desarrollo.

5.8.3. El Ecosistema de Spark

"Spark posee un núcleo pero necesita de librerías adicionales para poder brindar las características antes mencionadas, a esto se le llama el ecosistema de Spark. Todo este conjunto de software es el que nos provee de las cualidades de análisis en Big Data y de áreas de aprendizaje de máquinas".[21]

A continuación un resumen de las cuatro librerías más importantes:

[21] KARAU, Holden & KONWINSKI, Andy & WENDELL, Patrick & ZAHARIA, Matei *Learning* Spark - *Lightning-Fast Data Analysis.* Sebastopol : O'Reilly Media, Inc., 2015. Página 213.

5.8.3.1. Spark Streaming

Puede ser usada para el procesamiento en tiempo real de la información que fluye directamente a *Spark*.

5.8.3.2. Spark SQL

Provee a Spark de la capacidad de exponer la información a través de conexiones tipo "*JDBC*"[22] y permite realizar consultas a través del Lenguaje Estructurado de Consultas (SQL por sus siglas en ingles). Esto también le permite ejecutar procesos de extracción, transformación y carga de información de diferentes formatos.

5.8.3.3. Spark MLib

Es una librería escalable para proveer servicios de aprendizaje de máquinas que consisten en algoritmos de aprendizaje y utilidades, clasificación, regresión, agrupamientos, filtros, reducción de la dimensión, así como de optimizaciones primitivas.

5.8.3.4. Spark GraphX

Es la librería que provee del cálculo de grafos y en paralelo. *GraphX* unifica los procesos de *ETL*, análisis exploratorio y grafos iterativos en un solo sistema.

[22] Conexión a Base de Datos de Java (*JDBC* por sus siglas en ingles)

5.8.4. Arquitectura de Spark

La arquitectura de Spark incluye los siguientes tres componentes principales:

- Almacenamiento de la información.
- Interfaz para programación de aplicaciones (*API* por sus siglas en ingles).
- Administración de marco de trabajo.

A continuación una descripción de cada uno de estos componentes:

Almacenamiento de la Información

Hace uso del sistema de archivos distribuidos de Hadoop, para propósitos de almacenamiento. Funciona con cualquier fuente de datos compatible con Hadoop, como por ejemplo, *HDFS*, *HBase*, Cassandra, y otras más.

Interfaz para programación de aplicaciones

Esta interfaz es la que provee a los programadores la capacidad de poder crear aplicaciones basadas en Spark utilizando los estándares definidos en esta interfaz. Spark provee de esta interfaz a los siguientes lenguajes de programación:

- Scala
- Java
- Python

Administrador de recursos

Spark puede ser implementado en modo Solitario o en modo distribuido como por ejemplo Yarn.

Figura 12. **Componentes principales de Spark**

Fuente: elaboración propia.

5.9. Interacción de todos los componentes del ecosistema de Hadoop

Como se ha podido ver en el presente capítulo, existen distintas librerías, componentes, sistemas, que se interconectan entre sí para dar soporte a la arquitectura de Big Data, implementada desde la propuesta Hadoop. Esta interacción es compleja desde la concepción de la división del trabajo, que como hemos podido observar, es de responsabilidad exclusiva de algunos de estos componentes tareas que son vitales para el cumplimiento del almacenamiento, consulta y exploración de la información.

Se dividirán los componentes involucrados en tres grandes grupos: componentes del núcleo de Hadoop, componentes principales y componentes de soporte a componentes principales. Se describirán estos tres grupos a continuación:

Componentes del núcleo

Son todos aquellos que definen a la propuesta Hadoop, como por ejemplo el propio Hadoop y máquina virtual de Java.

Componentes Principales

En esta clasificación están todos los componentes que agregan funcionalidad a Hadoop. Estos se han ido agregando al ecosistema de Hadoop para proveer de funcionalidades que aportan algún tipo de valor agregado como por ejemplo el componente Spark, que al tratarse de otra propuesta completa de Big Data, provee de más de una funcionalidad, como lo es *MLib* que agrega algoritmos inteligentes para la búsqueda de patrones, aprendizaje de máquinas, análisis gráfico y otras funcionalidades.

Componentes de Soporte

Estos son todos aquellos que son necesarios para el buen funcionamiento de los componentes principales. En muchos casos serán componentes esenciales de los principales y en otros serán requisitos para la instalación de estos últimos.

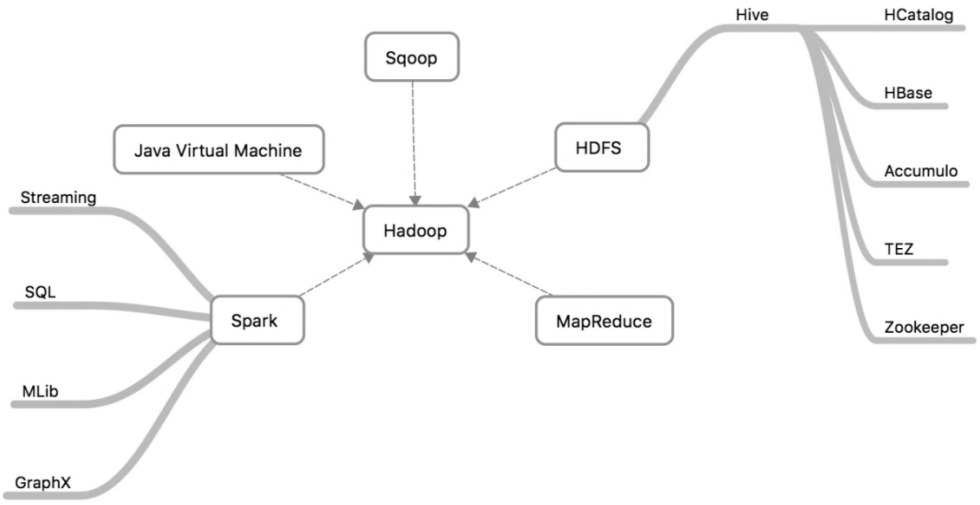
Tabla IX. **Componentes utilizados en el ecosistema de Hadoop propuesto**

Elemento	Tipo
Hadoop *HDFS*	Núcleo
MapReduce	Núcleo
Java *JVM*	Núcleo
Yarn	Núcleo
Hive	Principal
HCatalog	Soporte
HBase	Soporte
Tez	Principal
Sqoop	Principal
Accumulo	Soporte
Zookeeper	Soporte
Spark	Principal
Scala	Soporte
Spark *Streaming*	Soporte
Spark SQL	Soporte
Spark *MLib*	Soporte
Spark *GraphX*	Soporte

Fuente: elaboración propia.

A continuación se presenta en la figura donde se muestra la relación de todos los elementos involucrados en la arquitectura de Big Data propuesta.

Figura 13. **Elementos propuesta Hadoop de Big Data y su relación**



Fuente: elaboración propia.

La interrelación que existe entre los elementos del ecosistema de Hadoop provee de la funcionalidad completa de una solución de Big Data, en la cual se evidencia, a través de la funcionalidad, los roles que juegan cada uno de estos elementos en las tres tareas más relevantes en cuanto administración de la información. En la siguiente tabla se muestra la organización de estos elementos agrupados por su funcionalidad.

Tabla X. **Componentes ecosistema de Hadoop por funcionalidad**

Elemento	Funcionalidad
Hadoop *HDFS*	Almacenamiento
MapReduce	Consulta
Yarn	Consulta
Hive	Almacenamiento, Consulta y IPA
HCatalog	Almacenamiento
HBase	Almacenamiento
Tez	Consulta
Sqoop	Almacenamiento
Accumulo	Almacenamiento
Zookeeper	Administración
Spark	Almacenamiento y Consulta
Spark *Streaming*	Consulta
Spark SQL	Consulta
Spark *MLib*	IPA
Spark *GraphX*	IPA

Fuente: elaboración propia.

6. HERRAMIENTAS DE DESARROLLO

El modelo propuesto para el desarrollo del sistema que debería de dar acceso a la información en Big Data, es el modelo de desarrollo de tres capas, Modelo, Vista y Controlador. Este modelo proveerá de la capa de acceso a la información de manera abstracta y permitirá de una mejor integración con la capa de acceso a la información de Hadoop.

La parte de acceso a datos las proveerán dos componentes principales, Hive a través de su librería *JDBC* y Spark a través de su librería de conexión *IPA* para *Java*. De esto último surge definir *Java* como el lenguaje sobre el que se debería desarrollar al controlador.

El controlador conectará los datos con el usuario del sistema a través de la vista, la cual se propone ser desarrollada sobre HTML5 con soporte de AngularJS para proveer al usuario de una experiencia interactiva de última generación.

6.1. HTML5

6.1.1. La filosofía detrás del *HTML5*

Detrás de *HTML5* existe una serie de principios de diseño. Esto se traduce en tres grandes objetivos de *HTML5*:

- Especificar los estándares detrás de los exploradores web actuales y define los principios de la interoperabilidad.

- Define por primera vez el manejo de errores.

- Evoluciona el lenguaje para hacerlo mucho más robusto en el desarrollo de aplicaciones web.

Muchos de los métodos actuales de desarrollo de sitios y aplicaciones web no están documentados o al menos no especificados y los exploradores agreguen con frecuencia cualidades que no llegan a describirse como estándares. Por ejemplo los *XMLHttp-Request* el poder detrás de muchos sitios que implementan *AJAX*[23]. Fue inventado por *Microsoft*, luego fue utilizada ingeniería inversa por el resto de corporaciones para ser implementado en los otros exploradores, pero nunca fue especificado como un estándar.

Era necesario romper con la ambigüedad que definía como los exploradores y otros agentes lidiaban con las etiquetas inválidas. Esto no era un problema en el mundo del *XML*. Ellos simplemente los especificaban como un "manejo de error draconiano"[24] en el que el explorador detenía toda la renderización de una página web al primer error encontrado. "En nuestra opinión una de las razones para el éxito de la *Web* es que un mal código tiene un buena oportunidad de empezar hacer guiado por algún explorador o por todos ellos" (*HTML5*, 2011). La barrera para publicare un sitio *Web* ha decrecido, pero cada explorador era libre de decir cómo manejar el mal código. Como ejemplo:

Hola mundo

[23] *AJAX* es un tecnología de comunicación asíncrona entre el usuario y un servidor web, permite una interacción que simula ser de dos vías y no como la tradicional comunicación *HTTP* en la que se espera una solicitud del usuario al servidor web.
[24] Es una técnica para manejar errores en el que el primer error encontrado detiene toda la ejecución o en el caso del *XML* la interpretación. Su nombre es tomado de un legislador griego y se aplica "una ley, una providencia o una medida sanguinaria o excesivamente severa"

Nótese el error al cerrar las etiquetas, el orden esta intercambiado. Este código genera diferentes resultados en exploradores diferentes. La diferencia de los documentos generados luego de la renderización puede producir que la aplicación de hojas de estilo generen visualizaciones diferentes en los diferentes exploradores. Una generación consistente en la visualización es importante para el diseño del *HTML5* dado que el lenguaje por sí mismo está definido en el documento que se ha generado de la renderización.

En el interés de una gran interoperabilidad, es vital que el manejo de errores sea igual a través de todos los exploradores, esto asegura que se generará un documento "*DOM*"[25] cuando se confronta con código *HTML* que está mal codificado. En orden para que esto suceda, el *HTML5* detalla casi 900 páginas de un extenso documento de especificaciones, pero solamente cerca de 300 son relevantes para los autores de páginas web, el resto del documento es para los implementadores de exploradores web, diciéndoles como exactamente deber ser pareadas las etiquetas, incluso las etiquetas que están mal codificadas.

6.1.2. Aplicaciones Web

Se ha incrementado el número de sitios en la red que son llamados aplicaciones web, esto es para poder compararles en características con las aplicaciones en las computadoras de escritorio y que las diferencia de las páginas web de antaño que solo exponían texto estático, imágenes y enlaces a otros documentos que en su mayoría eran igual de estáticos.

[25] Del inglés *Document Objetc Modal*, este es el resultado del proceso de renderización que realizan los exploradores web para mostrar una página web.

Ejemplos de lo que suele ser llamado una aplicación web son: procesadores de texto en línea, herramientas para edición de fotografías, sitios de mapeo y demás usos. Altamente potenciados por *JavaScript*, que impulso en gran medida todas las características que tuvo el *HTML 4*.

HTML5 especifica las nuevas interfaces de programación para aplicaciones (*API* por sus siglas en inglés) para características como: sujetar y dejar, eventos de envió desde el servidor, dibujos y videos, por mencionar algunos. Estas nuevas interfaces de las paginas *HTML* exponen que dar acceso a *JavaScript* a estos objetos en el *DOM* hacen mucho más fácil escribir este tipo de aplicaciones utilizando estándares especificados en lugar de hacer uso de técnicas no documentadas.

El *HTML5* se está moviendo muy rápido hoy en día, desde que se publicará su primera especificación en octubre del 2009, los exploradores han ido implementando el soporte del *HTML5* (particularmente alrededor de los *IPA*) luego de esta fecha.

Entre las nuevas características que ofrece el *HTML5* están:

Nuevo tipo de documento

Se ha asociado un nuevo tipo de documento a la estandarización del *HTML5*, no se debe de especificar el tipo de documento. Una mejora sustancial en cuanto a ya no tener que diferenciar entre un tipo de página y otra, todas pertenecen al mismo tipo.

La nueva etiqueta “*figure*”

Permite hacer una asociación directa entre una imagen y un texto que se visualizará asociado directamente a ella.

La etiqueta “*small*” se redefine

El elemento *small* ahora hace alusión a la impresión del texto.

No más tipos para las etiquetas “*link*” y las “*script*”

La especificación del tipo de las etiquetas *link* y *script* ya no es necesario, fue simplificado de tal manera que cada etiqueta podrá saber el tipo que se ha asociado a la fuente.

Las comillas de los atributos

El ya no incluir la especificación *XHTML* para el *HTML5*, hace que ya no sea necesario agregar comillas a los valores de los atributos, ahora se deja como opcional el colocarlas o no.

Contenido editable

El contenido dentro de las etiquetas puede ser editado por el usuario.

Tipo de campo de ingreso *Email*

El explorador se encargará de validar que el texto ingresado tenga el formato de válido de un correo electrónico.

Valores sugeridos

Nombrados en inglés como *placeholders*, es un texto que sugiere el valor o da un tipo de instrucción de la forma correcta del ingreso del valor esperado. Es un texto que desaparece en el momento de que el usuario inicie con el ingreso del valor.

Campos obligatorios

El atributo requerido ahora está disponible para indicar que un campo dentro del formulario debe ser ingresado por el usuario.

Atributo autofocus

Este atributo, antes no disponible, provee de la funcionalidad de indicar que campo dentro del formulario tendrá el carrete seleccionado para hacer ingreso de información.

Expresiones regulares

Los campos de ingreso permiten indicar una expresión regular para hacer la validación cuando el usuario ingrese la información.

Soporte para Audio y Video

Ya no es necesario recurrir a soluciones complejas para lograr hacer las publicaciones de archivos de audio y video, *HTML5* provee ya de las etiquetas que permiten hacer uso de este tipo de información.

Cabe mencionar que a todas las características mencionadas anteriormente se suman los componentes adicionales que pueden acompañar al *HTML5* y que enriquecen la experiencia del usuario, tal es el caso de las hojas de estilo, que permiten que de manera rápida y ordena dar un diseño visual específico por cada página contenido dentro de un sitio web. Por eta razón se recomienda hacer uso de estas para la implementación del sistema.

Se mencionará brevemente un componente de diseño que al día de hoy es a considerar en cualquier proyecto que involucre el objetivo de brindar una experiencia de enriquecida de usabilidad y sea visualmente agradable, este es bootstrap, que se ampliará a continuación.

6.2. Bootstrap

Es un marco de trabajo de hojas de estilo en cascada (*CSS* por sus siglas en ingles) desarrollado por Twitter alrededor del año 2011, que permite mediante archivos *CSS* y con ayuda de "JQuery"[26] proveer de un conjunto de tipografías, botones, paneles, menús y otros elementos que pueden ser utilizados en cualquier sitio web.

Aun cuando el desarrollo de Botstrap fue iniciado por una empresa privada, fue liberado bajo "licencia *MIT*"[27] en el año 2011 y su desarrollo se volvió parte de la comunidad a través de su publicación en el sitio web especializado de "GitHub"[28].

[26] JQuery es un marco de trabajo basado en JavaScript que provee de funcionalidad adicional.
[27] Licencia *MIT* definida así por el Instituto Tecnológico de Massachusetts y es una licencia de software permisiva lo que significa que impone muy pocas limitaciones en la reutilización.
[28] GitHub es una plataforma que permite el desarrollo en colaboración y permite alojar proyectos utilizando un sistema de control de versiones Git.

Este marco de trabajo es una excelente herramienta para dar apoyo al *HTML5* permitiéndole crear interfaces para usuario limpias y totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea el tamaño del dispositivo desde donde se visualice. Además de todo esto ofrece las herramientas necesarias para crear cualquier tipo de sitio web utilizando los elementos en sus librerías. Cabe mencionar que todos los elementos que se pueden apreciar en el sitio web de Twitter están disponibles en Bootstrap.

Desde la aparición de Bootstrap en la versión 3, el marco de trabajo ha evolucionado en la compatibilidad con el nuevo objetivo de desarrollo "*responsive*" que busca la unificación de los sitios web en sus diferentes versiones para adaptarse a diferentes dispositivos, por ejemplo, anteriormente se llevaba el desarrollo en paralelo de dos sitios web, en los cuales uno estaba dedicado a los dispositivos tradicionales y el segundo estaba dedicado a los dispositivos móviles. La problemática de esto es que adicional al esfuerzo extra que esto requería, la experiencia para el usuario era diferente, esto generaba problemas de usabilidad.

La compatibilidad actual de Bootstrap *3* respecto a los exploradores web que existen actualmente es:

Tabla XI. **Compatibilidad de Bootsrap con los exploradores más usados**

Explorador *Web*	Plataforma
Chrome	Todas
Safari	MacOS y iOS
Firefox	MacOS y Windows
Internet Explorer	Windows y Windows *Phone*
Opera	Windows y MacOS

Fuente: elaboración propia.

6.3. AngularJS

6.3.1. ¿Qué es Angular?

Es un marco de trabajo estructural para desarrollar páginas web dinámicas. Permite el uso de *HTML* como plantilla de lenguaje y permite extender las funcionalidades de la sintaxis de *HTML* para escribir los componentes de las aplicaciones de forma clara y sucintamente. En Angular el llenado de la información y la dependencia de la inyección eliminan en gran parte la necesidad de escribir muchas líneas de código que de otra manera se deberían de codificar. Y todo lo antes mencionado sucede dentro del explorador, haciendo que Angular sea un compañero ideal de cualquier tecnología.

Es el complemento perfecto para el *HTML* para poder desarrollar aplicaciones de un grado elevado de complejidad. El *HTML* es un gran lenguaje de etiquetas para documentos estáticos, en su versión 5 ha mejorado pero aún no provee de toda la interacción que si tiene las aplicaciones de escritorio. Esto da como resultado que el desarrollador tenga que encontrar mecanismo de cómo engañar al explorador para que haga lo que el desarrollador quiere.

La brecha entre aplicaciones web dinámicas y documentos estáticos se puede resolver mediante las siguientes dos opciones:

- Librerías: una colección de funciones que pueden ser útiles para construir aplicaciones web. Este código está a cargo y puede llamar a otras librerías que se ajusten. Como por ejemplo, JQuery.

- Marcos de trabajo: son una implementación particular de una aplicación web, donde el código generado está lleno de detalles. El marco de trabajo está a cargo y las llamadas dentro del código necesitan ser específicas a la aplicación. Por ejemplo, druandal, ember, nodejs.

Angular toma otro acercamiento para resolver el problema. Intenta minimizar la brecha entre documento estático *HTML* y lo que una aplicación necesita para ser creada en nuevas construcciones basada en *HTML*. Angular enseña al explorador nuevas sintaxis dentro de la construcción de llamadas a directivas. Algunos ejemplos son:

- Enlace de datos, esta sintaxis es {{ }}.

- Estructuras de control *DOM* para generar ciclos, mostrar y ocultar fragmentos del *DOM.*

- Soporte para formularios y validación de los mismos.

- Agrega nuevos comportamientos para los elementos de *DOM*, como el manejo de eventos.

- Agrupamiento de *HTML* para reutilización de componentes.

Angular no es una simple pieza dentro de todo el ecosistema de la construcción de una aplicación web cliente – servidor. Se hace cargo de todo el *DOM* y de las llamadas *AJAX*, funciona como una facilitador que una vez escrito genera una estructura bien definida entre la información y la presentación. Esto hace que las opciones acerca de cómo realiza operaciones de lectura, altas, bajas y cambios dentro de la aplicación sean un proceso natural dentro del diseño. También trata de asegurar que esto sea un punto de partida que pueda ser cambiado fácilmente en la evolución natural que tiene todo sistema. Angular viene con una forma innovadora que hace pensar fuera de la caja. Entre las relaciones más notorias entre la forma convencional de realizar las operaciones y la solución que ofrece angular están:

- Todo lo que se necesita es construir las altas, lecturas, cambios y bajas ("*CRUD*"[29] por sus siglas en inglés) en una aplicación son análogas a: enlace a datos, plantillas básica de directivas, validación de formularios, rutas, enlace profundo, componentes reutilizables e inyección de dependencias.

- Historias de testeo: pruebas unitarias, pruebas de fin a fin y simulación de objetos.

- Alto rendimiento de las aplicaciones con una capa de directorio y scripts de test como puntos de inicio.

[29] *CRUD Crate Read Update Delete,* es el acrónimo en inglés para designar las operaciones sobre la información

Angular simplifica el desarrollo de aplicaciones por medio de la presentación a un alto nivel de abstracción para el desarrollador. Como una abstracción que viene a costa de flexibilidad. En otras palabras, no toda aplicación se ajusta bien a Angular.

Angular está construida y pensada para aplicaciones que implican operaciones *CRUD*. Por suerte las aplicaciones que conllevan operaciones *CRUD* representan la gran mayoría de aplicaciones web. Para entender para qué es bueno Angular, se debe de entender para que no es bueno.

Los juegos y las aplicaciones que conlleven mucho procesamiento gráfico con uso intenso y manipulación de las estructuras de los *DOM*, son aplicaciones que requieren una metodología de desarrollo diferente a las que aplicaciones con uso extenso de operaciones *CRUD* y esto da como resultado que este tipo de aplicaciones no se ajusten bien a Angular. En este caso es mejor hacer uso de librerías con un nivel bajo de abstracción, como es el caso de JQuery.

6.3.2. La parte fundamental de Angular

"Angular está construido alrededor de la creencia de que el código declarativo es mejor que el imperativo. Cuando es requerida la construcción de interfaces de usuario y la unión con los componentes de software. Mientras que el código imperativo es excelente para expresar lógicas de negocio". [30]

Angular se basa en las experiencias del pasado para hacer proposiciones sobre cómo se debe hacer el desarrollo de aplicaciones web. Entre las más relevantes están:

[30] KURZ, JOSH. *Mastering AngularJS Directives.* Birmingham : Packt Publishing, 2014. Páginas consultadas 116-118.

- Es una buena idea separar la manipulación de los elementos de presentación de la capa de lógica. Esto mejora dramáticamente las pruebas que se han de aplicar a las aplicaciones web.

- Es una mejor idea el dar la misma importancia a la fase de pruebas de la aplicación tal cual importante es la fase de codificación. Las dificultades de las pruebas es drásticamente afectada por la forma en la que el código es estructurado.

- Separar el código del lado del cliente y el código del lado del servidor es de vital importancia. Esto permite que el trabajo de desarrollo pueda progresar en paralelo y permite poder hacer reutilización de código por parte de ambos lados.

- Es muy útil si el marco de trabajo puede guiar a los desarrolladores a través de todo el proceso de construcción de la aplicación: Desde el diseño de la interfaz de usuario (presentación), la capa de lógica de negocios para terminar finalmente en la fase de pruebas.

- Siempre es buena idea realizar pruebas de tareas comunes como de tareas difíciles que son posibles que se presenten.

Angular permite olvidarse de una serie de tareas que en las metodologías tradicionales de desarrollo eran necesarias de ser codificadas por parte de los desarrolladores. Entre las más comunes que ya no son necesarias realizar están:

- Registro de llamadas de retorno: Registrar las llamadas de retorno dentro del código, siempre es difícil ver el bosque desde la perspectiva de unos cuantos árboles. Reducir la codificación de estas llamadas reduce la cantidad de código de *JavaScript* que es necesario para encargarse de las mismas.

- Manipular el *HTML* desde la programación: Manipular los elementos *HTML* es siempre una necesidad cuando se hace uso de aplicaciones que utilizan *AJAX*, pero esto siempre es incómodo y propenso a errores. El uso de directivas permite tener un código libre de tareas que realicen manipulaciones a los elementos *HTML*.

- Clasificando la información desde y para la interfaz gráfica: las operaciones *CRUD* se realizan en su mayoría desde tareas *AJAX*. El flujo de la clasificación de la información desde el server hacia el objeto interno del formulario *HTML*, permite que los usuarios modifiquen el formulario, lo validen, desplegué errores y todo esto sea regresado a un objeto interno y luego sea enviado al servidor. *Angular* elimina mucho de estas tareas, deja que el código describa el flujo general de la aplicación en lugar de implementar un sin fin de detalles para su manipulación.

6.3.3. Directivas

Angular ofrece a nueva perspectiva de desarrollo de aplicaciones web que está haciendo cambiar más y más opiniones cada día. La razón para que las personas estén sumándose a la dirección que ha tomado Angular es por vistas ortogonales en la encapsulación y separación que ofrece de todas la partes del desarrollo. La separación de la lógica en ámbitos definidos estructuralmente es la especialidad de Angular y con esto se obtiene un mayor enfoque en la lógica.

Las directivas ofrecen la mayor forma de encapsulamiento dentro de las aplicaciones de Angular. Esto es verdad dado que el enfoque está en la separación de la vista del modelo. Por años los desarrolladores han tenido que combinar diferentes tipos de lógicas del lado del cliente que no están obligadas a tener un enlace específico con la lógica de negocios. La separación entre la vista y el modelo ha tenido un efecto en las aplicaciones web modernas y las directivas de Angular tienen esto como objetivo principal.

Se cree que las directivas son la parte más difícil de aprender de Angular, se cree eso porque las directivas toman una perspectiva de las convenciones tradicionales de JavaScript, las cuales no se habían hecho anteriormente. No hay muchas librerías que se enfoquen en un acercamiento declarativo para poder manejar la relación entre *HTML* y JavaScript. Estos nuevos conceptos parecen difíciles a primera vista, pero una vez que se logra entender la lógica, las cosas empiezan a parecer que están en su lugar muy rápidamente.

Muchos de los problemas habituales son resueltos de forma simple por una directiva o por un conjunto de estas que trabajan al unísono una con otra. Una vez que las directivas son entendidas hay muchos casos que se acoplan perfectamente a ellas.

En conclusión AngularJS ofrece muchas diferentes facetas de la tecnología que pueden ser usadas para cumplir con diferentes tareas de forma eficaz y eficiente. En Angular no existe una implementación más específica o poderosa que una directiva. Una directiva puede ser definida como una instrucción oficial o autoritativa. Este es un término moderno no técnico de una directiva. En Angular las directivas aún siguen la definición sin embargo son una descripción mucho más técnica que puede contener una serie de instrucciones, la meta principal es leer o escribir código *HTML*.

6.4. Java

6.4.1. Servlets

El entorno de los Servlets se extiende más allá de la necesidad de un apoyo básico de Java al ámbito web. Cualquier computadora que ejecute Servlets necesita tener un contenedor. Un contenedor es una pieza de software responsable por la carga, ejecución y descarga de los Servlets. Las razones para esto están largamente relacionadas con la historia del desarrollo web del lado del servidor. Una rápida revisión de los inicios de las más prominente solución de contenidos dinámicos es la Interfaz de entrada común (*CGI* por sus siglas en inglés) y las diferencias entre esta y los Servlets es de mucha ayuda para entender porque los Servlets requieren de un contenedor.

En el mundo de Java los Servlets fueron diseñados para resolver los problemas más comunes que tenía *CGI*, entre los que cabe mencionar, el acceso a recursos compartidos, el pobre ciclo de vida de los procesos relacionados, las dificultades de comunicación entre los programas externos y el servidor web. Los Servlets fueron creados para crear entornos robustos del lado del servidor para los desarrolladores web. Similares a *CGI* los Servlets permiten peticiones que serán procesadas por un programa externo y permite que el mismo programa produzca la respuesta del contenido dinámico. Los Servlets adicionalmente definen un eficiente ciclo de vida que incluye la posibilidad de usar procesos simples para atender todas las respuestas. Esto elimina la necesidad de múltiples procesos de *CGI* con lo que se logra que el proceso principal comparta los recursos entre múltiples Servlets y múltiples respuestas.

Una de las mayores características de los Servlets es que permiten que exista un proceso principal desde el que se ejecutan todos los procesos de respuesta, lo que es una de las mayores claves de eficiencia. Con la eficiencia como una de las metas principales y el soporte que ofrece *Java* con su soporte multiplataforma los Servlets han resuelto varios problemas y esto los ha convertido en una solución popular para la funcionalidad de contenido dinámico del lado del servidor. Ya hace varios años que los Servlets aparecieron en el entorno del desarrollo web y aún siguen siendo, junto a otras tecnologías, la solución a muchos problemas cotidianos en el desarrollo de aplicaciones web. Esto se puede observar que junto a las páginas *javaserver* (*JSP* por sus siglas en ingles) se combinan para forma la capa web para estándar Java *2 Enterprise Edition J2EEE*.

Los Servlets son una interface definida según estándares en *Java* a través del paquete *javax.servlet*. Este declara tres métodos esenciales para el ciclo de vida de los Servlets: *init*(), *service*() y *destroy*(). Estas tres funciones principales son invocadas en tiempos específicos por el servidor.

Init(): este método es llamado en la fase de inicialización en el ciclo de vida del *Servlet*. Permite el acceso del Servlet a los parámetros de inicialización desde la aplicación web.

Service() / *doPost*() / *doGet*(): este método es invocado en el momento de que una petición es realizada. Cada petición es atendida por su propio hilo dentro del proceso principal. El contenedor llama al método *service*() por cada petición. Este método determina el tipo de petición y despacha el método apropiado que se encargará de la respuesta. El tipo de método que se llame dependerá del contexto de su uso.

Destroy(): este método es llamado cuando el objeto del Servlet tiene que ser destruido. Se encarga de liberar los recursos previamente utilizados por el proceso *Service*().

En el ciclo de vida del Servlet, hemos podido observar que las clases dentro del server son cargadas por el contenedor de forma dinámica. Cada petición tiene su propio hilo dentro del proceso principal y un objeto del Servlet puede ser usado por múltiples hilos al mismo tiempo.

6.4.1.1. Ciclo de vida

- La clase del Servlet es cargada

El cargador de clase es el responsable de cargar la clase del Servlet. La clase del Servlet es cargada con la primera petición recibida para el Servlet por parte del contenedor.

- La instancia del Servlet es creada

El contenedor crea la instancia del Servlet luego de que es carga la clase. La instancia del Servlet se crea una sola vez dentro del ciclo de vida del Servlet.

- El método *Init* es invocado

El contenedor llama al método *init* solamente una vez luego de que es creada la instancia del Servlet. El método *init* es usado para inicializar el Servlet.

- El método *service* o *doget* o *dopost* es invocado

El contenedor llama al método *service* o *doget* o *dopost* cada ocasión en la que existe una petición al Servlet es recibida. Si el Servlet no está inicializado, se da la secuencia de los tres primeros pasos. Si el Servlet ya se encuentra inicializado entonces pasa a la llamada de este paso directamente.

- El método *destroy* es invocado

El contenedor llama al método *destroy* antes de remover la instancia del servicio. Esto permite que el Servlet tenga la oportunidad de limpiar cualquier recurso utilizado, por ejemplo, memoria, hilos, accesos hacia archivos.

El uso de Servlets para la comunicación entre la capa de presentación y la capa de datos son ampliamente utilizados por la comunidad de código abierto. Siendo *Java* el lenguaje principal en muchos esfuerzos de la fundación Apache, resulta una estrategia vital el desarrollar la capa de lógica en el lenguaje nativo para Hadoop como la tecnología de Big Data que se desea utilizar.

6.4.1.2. Contenedores

El rendimiento de los Servlets se puede atribuir directamente al contenedor del mismo.

Un contenedor de Servlet, también llamado contenedor o contenedor web, es una pieza de software que administra todo el ciclo de vida de un Servlet. El software del contenedor es el responsable de interactuar con el servidor web para procesar una petición al Servlet para obtener una respuesta del mismo. La definición oficial de un contenedor es descrita de forma total por las especificaciones del Servlet. A diferencia de la mayoría de tecnologías propietarias, las especificaciones solamente definen el estándar para la funcionalidad que deberá de implementar el contenedor. Existen muchas implementaciones de contenedores de Servlets de diferentes vendedores con diferentes precios, rendimiento y características. Esto deja a los desarrolladores con muchas opciones para el desarrollo del software.

7. TOCANDO LA GRAN SINFONÍA

A lo largo de los capítulos anteriores se ha evidenciado como la propuesta de Big Data que actualmente es una realidad para el manejo de información masiva, también se ha podido dejar muestra que se trata de una tecnología con una madurez de desarrollo en la que muchas veces supera tecnologías tradicionales para el análisis de grandes bases de información. Con todo esto se ha pretendido dejar claro que los retos que se tenían tradicionalmente de almacenar y consultar información cuando esta última llega a grandes cantidades, *gigabytes* o incluso *terabytes*, empiezan hacer no solucionables con sistema tradicionales de bases de datos relacionales.

Si a todo esto agregamos la dimensión de querer poder realizar análisis de la "fotografía completa", es decir análisis integrales que conlleven no solo procesar información interna del Ministerio de Finanzas Públicas, sino también de entes externos que son los ejecutores principales de la producción pública del país, el uso de las tecnologías tradicionales como los sistemas de administración de bases de datos relacionales (*RDBMS* por sus siglas en ingles) en sus versiones "*OLTP* y *OLAP*"[31], carecen de los mecanismos necesarios para hacer uso de estas en análisis oportunos y de gran importancia para el país.

[31] Procesamiento de transacciones en línea por sus siglas en ingles *OLTP*, Procesamiento analítico en línea por sus siglas en ingles *OLAP*.

De la manera tradicional en la que se enfocaba el problema, era realizar un procesamiento completo de la información, desde la concepción inicial de la misma hasta el procesamiento y consulta. Se estructuraba la información que daba pie a un formato específico de la información basada en diagramación de la misma, que a su vez hacia sentido al negocio.

La estructura de la información es una cualidad de la captura y de la consulta, pero ¿qué sucede cuando no tenemos un control sobre la fuente de la información? Nos vemos obligados a generar grandes procesos de transformación antes de que esta información pueda pasar hacer parte de nuestro universo de información analizable.

Partiendo del principio, de que el análisis de partes de la información pueda darnos soluciones parciales o de poco impacto para grandes problemas, es de vital importancia contar con herramientas de información que puedan proveernos de una visión holística de la información. De esto viene la propuesta del uso de Big Data. Como ya hemos estudiado anteriormente la tecnología de Big Data propone el concepto de estrategia militar muy conocido de los tiempo en las que el emperador Julio Cesar la materializo en su frase muy famosa: “Divide et impera” que viene del latín y tienen una traducción al castellano de “Divide y vencerás”, es decir dividiremos el problema del almacenamiento de la información al problema de la consulta de la información.

Lo anterior crea la necesidad de poder tener un solo sistema que se encargue del manejo de la información, ciertamente, pero apoyado de varios subsistemas que se encarguen de partes específicas de la gran solución. Esto genera la obligación de dar una ubicación específica a cada uno de estos subsistemas en un mapa general y definir cuáles serán los límites y los alcances de sus responsabilidades y aportes de cada uno a la solución general.

7.1. Mapa general de la solución

A continuación se hace un listado resumen de todos los sistemas que colaborarán entre sí para proveer la solución de inteligencia de negocios a través de la tecnología de Big Data. Se agruparán por tipo de responsabilidad:

- Captura de la información derivada de procesos transaccionales.
- Almacenamiento estructurado de la información generada por el Minfin.
- Traslado de la información a un sistema de archivos distribuido carente de esquema.
- Almacenamiento de la información en un sistema de archivos distribuido, información carente de esquema y con esquema.
- Procesamiento de la información por lotes y acceso a la misma en tiempo real.
- Dar acceso al usuario a la información procesada y analizada.

A continuación se muestra la tabla resumen:

Tabla XII. **Componentes y responsabilidades dentro de la solución propuesta**

Componente	Responsabilidad
Sicoin	Captura de la información de los procesos transaccionales
Oracle RDBMS	Almacenamiento de la información estructurada generada por Minfin
Sqoop	Traslado de la información estructurada de Oracle hacia el sistema de archivos distribuidos
Hadoop	Se encarga del almacenamiento de la información en su propio sistema de archivos distribuidos de Hadoop
Hive	Se encarga de brindar una capa de estructura a la información proveniente de *Oracle*
Spark	Se encarga de procesar la información almacenada en el *HDFS*
Java Servlets	Se encarga de la capa de lógica del negocio del sistema de inteligencia de negocios
AngularJS	Provee de la capa intermedia entre la capa de presentación y la capa de acceso a datos y lógica del negocio
HTML5	Se encarga de la capa de la presentación de datos

Fuente: elaboración propia.

7.1.1. Flujo de comunicación entre los subsistemas

Tan importante es saber dónde se ubicará cada uno de los componentes de la solución propuesta, como también los es el flujo de la información a través de los componentes hasta el momento que es consultado por el usuario.

La información es generada de manera descentralizada, por parte de todos los usuarios que ejecutan gasto a través del sistema de contabilidad integrada. Esta información generada es almacenada por un sistema de administración de bases de datos relacionales, en este caso específico es Oracle. Este se encarga de almacenar la información estructurada.

Luego la información es traslada por la aplicación Sqoop que se encarga de consultarla desde Oracle y la almacena en el sistema de archivos distribuidos de Hadoop. Este se auxilia de Hive para mantener un esquema lógico de la información tal y como se encuentra en *Oracle*. Cabe resaltar que aun cuando Hive puede reflejar la información como que si esta fuera estructurada, por principio Hadoop almacena la información sin estructura. Estas tablas de Hive son almacenadas bajo los principios de Hadoop y se almacena de forma separada la lógica del esquema que tiene dentro de los sistemas transaccionales.

Acto seguido Hive es capaz de proveer el acceso a la información mediante *HQL* que es un lenguaje estructurado de consulta que, como lo hemos visto anteriormente, es capaz de generar trabajos de *MapReduce* para hacer la ejecución de las consultas a trabajos en lote, que son trasladados a Yarn para su ejecución.

Spark provee de la rapidez necesaria para la manipulación de la información al realizar cálculos intensivos sobre la información almacenada en Hadoop, este tipo de datos pueden generar cálculos sobre la misma data en una serie de tiempo, por ejemplo el cálculo del gasto de una entidad de forma mensual acumulada, es decir el gasto acumulado mensual en un ejercicio de tiempo determinado. Acá es donde Spark provee un mejor rendimiento ya que hace uso de la información en memoria de manera mucho más eficiente.

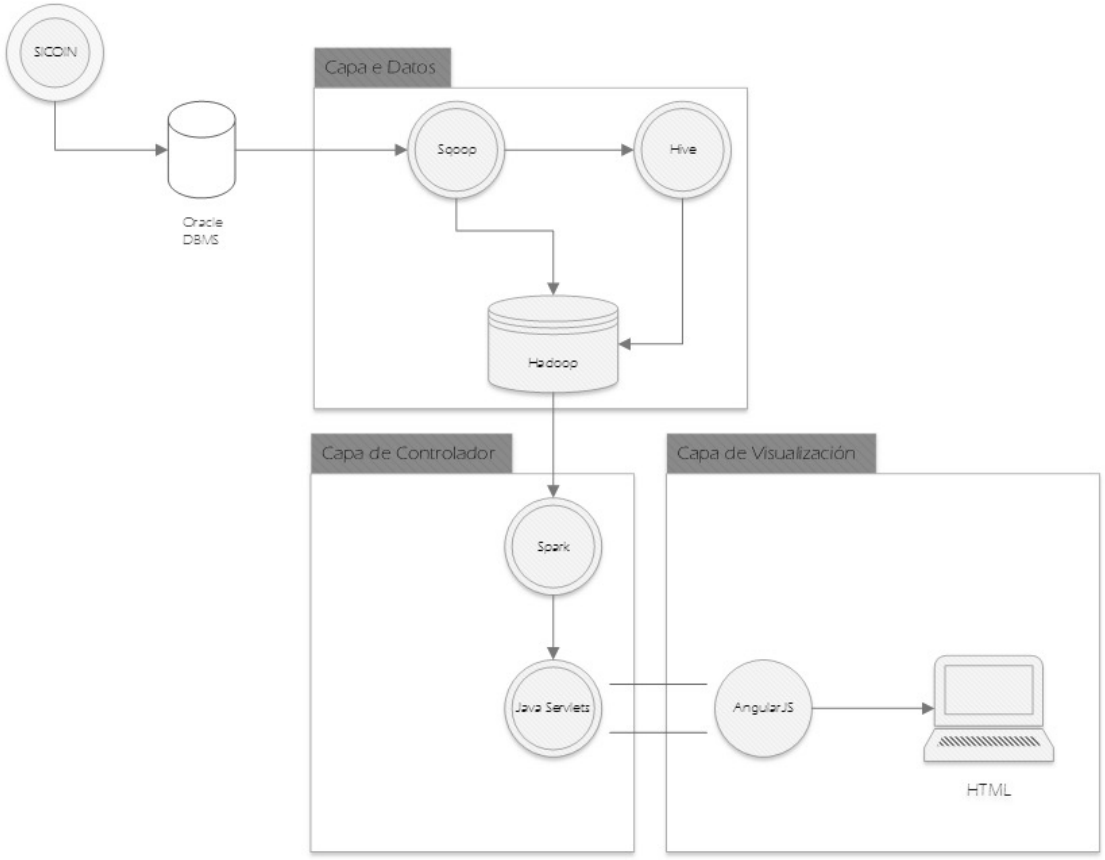
Finalmente los Java Servlets son la puerta de acceso común a la información contenida dentro del *HDFS* esto lo hace a través de la interfaz de programación de aplicaciones que ofrece Spark. Permite la conexión entre la capa de presentación y capa de datos.

El siguiente paso dentro de nuestro flujo es AngularJS que se encarga de trasladar las peticiones de datos de los usuarios a los Servlets, además de esto se encarga de proveer la funciones adicionales para la presentación de datos al *HTML5* quien finalmente se encarga de la generación de la vista.

Todo este flujo tiene el orden antes descrito que es el que logra cumplir con el objetivo de lograr dar acceso a los usuarios del sistema a información masiva vertida en un sistema de inteligencia de negocios utilizando la tecnología de Hadoop.

A continuación se muestra gráficamente el flujo de la comunicación:

Figura 14. **Comunicación componentes de la solución propuesta**



Fuente: elaboración propia.

Un flujo de información adecuado y en el orden correspondiente es como una sinfonía bien interpretada donde la suma de todos los instrumentos dentro de la orquesta juegan un rol para que el total de la música escuchada sea la que realmente disfrute el oyente. En muchas ocasiones un oído no entrenado puede perder la percepción de alguno de los instrumentos involucrado pero en definitiva se puede apreciar la música como un todo. Es igual en la solución propuesta de Big Data, cada subsistema en su lugar permite que el usuario tenga acceso al análisis de la información desde la perspectiva de inteligencia de negocios.

Además del flujo de información cabe mencionar las capacidades adicionales que provee esta propuesta al Minfin, que hoy en día no podría plantearse hacer dado la tecnología actual que utilizan.

7.1.2. Capacidades adicionales ganadas con Big Data

Hoy en día se tienen varios flujos de información hacia el Minfin desde otras instituciones estatales y no estatales, como por ejemplo, información de indicadores calculados desde otras entidades, información sobre los ingresos recolectados por la Superintendencia de Administración Tributaria SAT, la información de las transferencias bancarias por parte de Banco de Guatemala Banguat y otras instituciones. Pero toda esta información fluye con una estructura previamente definida, la información tiene esquema desde el momento en que es almacenada en los sistemas transaccionales. Esto pone límites al momento de requerir nueva información de entidades que no almacenan su información respetando las reglas definidas por el Minfin, como por ejemplo, entidades que no comparten los catálogos que se encuentran definidos en el Sicoin.

Con la implementación de esta nueva tecnología de Big Data, se puede obtener información que es fácilmente almacenada en los sistemas *HDFS* que no requieren definir un esquema específico. Podemos pensar que existen posibilidades infinitas de la transformación de la información recibida desde la demás entidades de gobierno y el cruce de esta con la información financiera detalla que se tiene ya en el Minfin.

El poder de realizar tareas de procesamiento de información que no impacten los sistema actuales transaccionales permite que no exista un cuello de botella en las que los usuarios del Sicoin compitan entre ingreso de información y acceso a la misma. Un problema muy cotidiano en todas las instituciones de gobierno que reciben grandes volúmenes de información, como por ejemplo la SAT.

Los costos de implementar tecnologías de código abierto aseguran que la sostenibilidad de este tipo de proyectos pueda ser de un menor costo, lo que pueda garantizar su uso en el tiempo, masificar los accesos y poder compartir los recursos de conocimientos desarrollados en una institución puedan ser aprovechados por otras que están iniciando.

CONCLUSIONES

Con la presente propuesta se ha dejado un valioso conocimiento de cómo aplicar diferentes tecnologías para la implementación de la arquitectura de Big Data en el análisis del presupuesto público nacional. Esto se ha logrado evidenciando como es hoy en día la tecnología de Big Data.

1. Se puede concluir que se han documentado los aspectos necesarios que se deben de tomar en cuenta al momento de dar los primeros pasos en el mundo de la tecnología de Big Data, haciendo ver de forma clara las características y cualidades que tienen cada una de las herramientas propuestas.

2. Fue definida la arquitectura de software necesaria para almacenar y analizar la información del presupuesto nacional, con la evidencia documental de cada uno de los componentes que deberá de agregarse al sistema de inteligencia de negocios.

3. Se definió la arquitectura de hardware óptima para las necesidades actuales del Minfin, además de proveer del diagrama de red que describe el número de servidores y cual deberá ser su función dentro de la arquitectura para Big Data.

4. Fue propuesto el conjunto de herramientas de software de código abierto y de licenciamiento libre que pueden implementar la tecnología de Big Data tomando como base la propuesta Hadoop y el conjunto de aplicaciones que dan soporte a esté.

5. Se especificaron las herramientas de software de código abierto y de licenciamiento libre para el análisis de la información, basándose en el sistema *Spark* que provee las herramientas más versátiles y poderosas, haciendo posible el análisis en tiempo real.

6. Se utilizó un diagrama de flujo para describir la comunicación de la información entre todas las herramientas propuestas para el almacenamiento, así como el análisis de la información financiera y de ejecución. Se ve cuál es el camino que toma la información desde la toma de la misma hasta su transformación en conocimiento.

RECOMENDACIONES

1. A través de todas las páginas anteriores se ha intentado hacer ver al lector una nueva perspectiva de cómo se puede suscitar un cambio en la forma en la que se analiza la información de carácter presupuestaria por parte del Minfin. Se ha tratado de mostrar una perspectiva nueva para el país, pero ya usada por algunos otros gobiernos alrededor del mundo como el caso de Estados Unidos de América, Reunió Unido, Suiza y otros más.[32] En muchos casos ha provisto de información en tiempos que antes eran impensados y esto mejora los tiempos en los que el gobierno puede reaccionar ante malas prácticas de una institución, mejorar un sector específico dada la retroalimentación de su ejecución, alertas tempranas de flujos de contribuyentes en el pago específico de algún impuesto.

2. Lo peor que nos pueda pasar es tener grandes cantidades de información y no poder ser capaces de que nuestras decisiones como guatemaltecos no sean basadas en información. El no tener nos limita el no saber qué cantidad de información podría ser analizada, lo que de cierta manera justificaría el seguir tomando decisiones de país utilizando el criterio personal, pero teniendo grandes volúmenes de información y no utilizarla es como tener un chaleco salvavidas y terminar ahogándose por no usarlo.

[32] Fuente: https://datafloq.com/read/4-benefits-public-sector-governments-start-big-dat/171

3. Si bien es cierto que mientras mayor sea el volumen de la información mayores son los retos y los costos para analizarla, se ha demostrado que utilizando software de código abierto se tiene un coste que tiende a cero en el aspecto de licenciamiento de software que es el rubro más alto en todos la mayoría de estos tipos de proyectos. Se tienen cifras en las que un servidor con un costo alrededor de 100 000 dólares americanos que tenga en su hardware una especificación de 1024 núcleos de procesamiento, requeriría un licenciamiento de millones de dólares para instalar un RDBMS como *Oracle*. Al contrario del caso expuesto en el que las características del hardware donde se ejecutará la solución no implican un problema, al contrario, mientras mejor sean, más aprovechable será por la arquitectura de Hadoop.

4. Bajo mi perspectiva, tengo que mencionar que al realizar esta propuesta de solución me ha quedado claro que la necesidad existe y cada día se hace más grande. La evidencia histórica de nuestro presupuesto público hace ver que los recursos año con año son menos que los del año anterior y nuestras necesidades como país cada vez son mayores dado el índice de crecimiento poblacional. Somos más con menos recursos para que seamos atendidos por los servicios públicos.

5. La tecnología ya existe y el implementar en los sistemas actuales del Minfin se deja plasmado en esta propuesta. Se recopila el conjunto de herramientas necesarias para llevar a cabo una implantación exitosa de una tecnología de primer nivel utilizada ya en países de primer nivel.

Por todo lo antes expuesto espero haber llenado las expectativas del lector en el acompañamiento por todas estas páginas de conocimiento desde una serie de fuentes de información que son empresas de vanguardia en algunos casos y en otros son autores de libros que se encuentran dentro de las implementaciones exitosas de esta tecnología en un sin fin de proyectos tanto para la iniciativa privada como para gobierno.

Esperemos algún día poder tener acceso a la información de gobierno de una manera fácil y sencilla, que no necesitemos ser expertos en la materia para poder entenderla o tener la restricción del papel para poder analizarla.

BIBLIOGRAFÍA

1. CEPAL. CEPAL STAT - *Base de Datos y Publicaciones Estadistícas*. Naciones Unidas, [en línea] <http://www.cepal.org/en/> [Consuta: 3 de junio de 2016].

2. Congreso de la República de Guatemala. *Constitución Política de Guatemala. Guatemala*, 1985. 109 p.

3. Congreso de la República de Guatemala. *Ley Orgánica del Presupuesto*, 1997. 21 p.

4. FALKNER, Jayson & JONES, Kevin. *Servlets and JavaServer Pages*. s.l. : Addison-Wesley, 2004. 784 p.

5. FASALE, Amol & KUMAR, Nirmal. *YARN Essentials.* Birmingham - Mumbai : PACKT Publishing, 2015. 176 p.

6. GARTNER. IT Glossary. *Big Data*. [en línea] < http://www.gartner.com/it-glossary/big-data/>. [Consulta: 29 de mayo de 2016.].MURTHY, Arun & VAVILAPALLI, Vinod & EADINE, Douglas & NIEMIEC, Joseph & MARKHAM, Jeff. *Apache Hadoop YARN: Moving beyond MapReduce and Batch Processing with Apache Hadoop 2* (Addison-Wesley Data & Analytics). Uppers Saddle River : Addison-Wesley, 2014. 400 p.

7. HARRIS, DAVID B. *Creating a Knowledge Centric Information Technology Environment*. Seattle, WA : Harris Training & Cosulting Services, 1996 [en línea] < http://eprints.rclis.org/24722/1/Dave-Harris.pdf> [Consulta: el 10 de junio de 2016].

8. KARAU, Holden & KONWINSKI, Andy & WENDELL, Patric & ZAHARIA, Matei. *Learning Spark - Lightning-Fast Data Analysis. Sebastopol*, O'Reilly Media, Inc., 2015. 276 p.

9. KURZ, Josh. *Mastering AngularJS Directives*. Birmingham : Packt Publishing, 2014. 210 p.

10. LARS, George. *HBase The Definitive Guide*. USA : O'Reilly, 2011. 556 p.

11. LASKOWSKI, Jacek. *Mastering Apache Spark. Web* : GitBooks, 2015, [en línea] <https://www.gitbook.com/download/pdf/book/jaceklaskowski/mastering-apache-spark> [Consulta: 19 de abril de 2016].

12. LAWSON, Bruce & SHARP, Remy. Html5, *Introducing. Berkeley : New Riders Voices That Matter,* 2011. 295 p.

13. Ministerio de Finanzas Públicas. *Aprendiendo Aspectos Básicos del Presupuesto. Ministerio de Finanzas Públicas*. [en línea] <http://consultaciudadana.Minfin.gob.gt/Documents/Aprendiendo_Aspectos_B%C3%A1sicos_del_Presupuesto%20v1.1.pdf> [Consulta: 10 de febrero de 2016.].

14. Ministerio de Finanzas Públicas. *Clasificador Presupuestario*. [en línea] < http://portalgl.Minfin.gob.gt/Descargas/Documents/Clasificador%20Presupuestario%20E4.pdf>. [Consulta: 1 de Julio de 2016].

15. MURTHY, Arun & VAVILAPALLI, Vinod & EADINE, Douglas & NIEMIEC, Joseph & MARKHAM, Jeff. *Apache Hadoop YARN: Moving beyond MapReduce and Batch Processing with Apache Hadoop 2* (Addison-Wesley Data & Analytics). Uppers Saddle River : Addison-Wesley, 2014. 400 p.

16. ORACLE ENTERPRISE ARCHITECTURE WHITE PAPER. *An Enterprise Architect's Guide to Big Data*. [en líneas] <http://www.oracle.com/technetwork/topics/entarch/articles/oea-big-data-guide-1522052.pdf > [Consulta: 30 de junio de 2016.].

17. REICHLER, Sabrina. *Los deflactores del gasto público: El caso de la educación básica y terciaria en la Argentina*. La Plata : Universidad Nacional de La Plata, 2005, [en línea] <http://www.depeco.econo.unlp.edu.ar/maestria/tesis/039-tesis-reichler.pdf> [Consulta: 17 de junio de 2016].

18. RUTHERGLEN, Jason & WAMPLER, Dean & CAPRIOLO, Edward. *Programming Hive: Data Warehouse and Query Language for Hadoop*. Cambridge : O'Reilly, 2012. 350 p.

19. Secretaría de Hacienda. *Programa de reforma de la administración financiera gubernamental*. [en línea] <http://administracionfinanciera.mecon.gov.ar/documentos/AF_Programa_Reforma.pdf >. [Consulta:1 de julio de 2016].

20. SPIEGEL, Murray R. Teoría y problemas de probabilidad y estadística. México : MacGraw-Hill, 2010. 357 p.

21. WADKAR, Sameer & SIDDALINGAIAH, Madhu & VENNEER, Jason. *Pro Apache Hadoop*. Washington : Apress, 2014. 437 p.

22. XIN, Reynold. *Apache Spark officially sets a new record in large-scale sorting*. Databricks. [en línea] <https://databricks.com/blog/2014/11/05/spark-officially-sets-a-new-record-in-large-scale-sorting.html>. [Consulta: 1 de julio de 2016].

ANEXOS

Anexo 1. **Tabla de los clasificadores presupuestarios**

Grupo	Sub Grupo	Renglón	Descripción
0			SERVICIOS PERSONALES
	01		*Personal en Cargos Fijos*
		011	Personal permanente
		012	Complemento personal al salario del personal permanente
		013	Complemento por antigüedad al personal permanente
		014	Complemento por calidad profesional al personal permanente
		015	Complementos específicos al personal permanente
		016	Complemento por transporte al personal permanente
		017	Derechos escalafonarios
		018	Complemento por diferencial cambiario al personal en el exterior
	02		*Personal Temporal*
		021	Personal supernumerario
		022	Personal por contrato
		023	Interinatos por licencias y becas

Continuación anexo 1.

		024	Complemento personal al salario del personal temporal
		025	Complemento por antigüedad al personal temporal
		026	Complemento por calidad profesional al personal temporal
		027	Complementos específicos al personal temporal
		028	Complemento por transporte al personal temporal
		029	Otras remuneraciones de personal temporal
	03		*Personal por Jornal y a Destajo*
		031	Jornales
		032	Complemento por antigüedad al personal por jornal
		033	Complementos específicos al personal por jornal
		034	Complemento por transporte al personal por jornal
		035	Retribuciones a destajo
	04		*Servicios Extraordinarios*
		041	Servicios extraordinarios de personal permanente
		042	Servicios extraordinarios de personal temporal
		043	Servicios extraordinarios de personal por jornal

Continuación anexo 1.

		044	Servicios extraordinarios por turnos a médicos de guardia
	05		*Aportes Patronales*
		051	Aporte patronal al IGSS
		052	Aporte patronal al INTECAP
		053	Cuota por seguros sociales por personal en el exterior
		054	Cuota recreacional
		055	Aporte para clases pasivas
	06		*Dietas y Gastos de Representación*
		061	Dietas
		062	Dietas para cargos representativos
		063	Gastos de representación en el interior
		064	Gastos de representación en el exterior
	07		*Otras Prestaciones Relacionadas con Salarios*
		071	Aguinaldo
		072	Bonificación anual (Bono 14)
		073	Bono vacacional
		074	Compensación costo de vida por servicios en el exterior
		079	Otras prestaciones
	08		*Personal Contratado por Organismos Internacionales*
		081	Personal administrativo y operativo
1			**SERVICIOS NO PERSONALES**
	11		*Servicios Básicos*

Continuación anexo 1.

		111	Energía Eléctrica
		112	Agua
		113	Telefonía
		114	Correos y telégrafos
	12		*Divulgación, Impresión y Encuadernación*
		121	Divulgación e información
		122	Impresión, encuadernación y reproducción
	13		*Viáticos y Gastos Conexos*
		131	Viáticos en el exterior
		132	Viáticos de representación en el exterior
		133	Viáticos en el interior
		134	Compensación por kilómetro recorrido
		135	Otros viáticos y gastos conexos
	14		*Transporte y Almacenaje*
		141	Transporte de personas
		142	Fletes
		143	Almacenaje
	15		*Arrendamientos y Derechos*
		151	Arrendamiento de edificios y locales
		152	Arrendamiento de tierras y terrenos
		153	Arrendamiento de máquinas y equipos de oficina
		154	Arrendamiento de maquinaria y equipo de construcción
		155	Arrendamiento de medios de transporte
		156	Arrendamiento de otras máquinas y equipo
		157	Arrendamiento de equipo de cómputo

Continuación anexo 1.

		158	Derechos de bienes intangibles
	16		*Mantenimiento y Reparación de Maquinaria y Equipo*
		161	Mantenimiento y reparación de maquinaria y equipo de producción
		162	Mantenimiento y reparación de equipo de oficina
		163	Mantenimiento y reparación de equipo médico, sanitario y de laboratorio
		164	Mantenimiento y reparación de equipos educacionales y recreativos
		165	Mantenimiento y reparación de medios de transporte
		166	Mantenimiento y reparación de equipo para comunicaciones
		167	Mantenimiento y reparación de maquinaria y equipo de construcción
		168	Mantenimiento y reparación de equipo de cómputo
		169	Mantenimiento y reparación de otras maquinarias y equipos
	17		*Mantenimiento y Reparación de Obras e Instalaciones*
		171	Mantenimiento y reparación de edificios
		172	Mantenimiento y reparación de viviendas
		173	Mantenimiento y reparación de bienes nacionales de uso común

Continuación anexo 1.

		174	Mantenimiento y reparación de instalaciones
		175	Mantenimiento y reparación de construcciones militares
		176	Mantenimiento y reparación de otras obras e instalaciones
		177	Mantenimiento y reparación de bienes nacionales de uso no común
	18		*Servicios Técnicos y Profesionales*
		181	Estudios, investigaciones y proyectos de factibilidad
		182	Servicios médico-sanitarios
		183	Servicios jurídicos
		184	Servicios económicos, contables y de auditoria
		185	Servicios de capacitación
		186	Servicios de informática y sistemas computarizados
		187	Servicios por actuaciones artísticas y deportivas
		188	Servicios de ingeniería, arquitectura y supervisión de obras
		189	Otros estudios y/o servicios
	19		*Otros Servicios no Personales*
		191	Primas y gastos de seguros y fianzas
		192	Comisiones a receptores fiscales y recaudadores
		193	Comisiones a colocadores de pólizas

Continuación anexo 1.

		194	Otras comisiones y gastos bancarios
		195	Impuestos, derechos y tasas
		196	Servicios de atención y protocolo
		197	Servicios de vigilancia
		198	Recompensas para seguridad del Estado
		199	Otros servicios no personales
2			**MATERIALES Y SUMINISTROS**
	21		*Alimentos y Productos Agropecuarios*
		211	Alimentos para personas
		212	Alimentos para animales
		213	Productos animales
		214	Productos agroforestales, madera, corcho y sus manufacturas
		215	Productos agropecuarios para comercialización
		219	Otros alimentos y productos agropecuarios
	22		*Minerales*
		221	Carbón mineral
		222	Minerales metálicos
		223	Piedra, arcilla y arena
		224	Pómez, cal y yeso
		225	Minerales no metálicos
		229	Otros minerales
	23		*Textiles y Vestuario*
		231	Hilados y telas
		232	Acabados textiles
		233	Prendas de vestir

Continuación anexo 1.

		239	Otros textiles y vestuario
	24		*Productos de Papel, Cartón e Impresos*
		241	Papel de escritorio
		242	Papeles comerciales, cartones y otros
		243	Productos de papel o cartón
		244	Productos de artes gráficas
		245	Libros, revistas y periódicos
		246	Textos de enseñanza
		247	Especies timbradas y valores
		249	Otros productos de papel, cartón e impresos
	25		*Productos de Cuero y Caucho*
		251	Cueros y pieles
		252	Artículos de cuero
		253	Llantas y neumáticos
		254	Artículos de caucho
		259	Otros productos de cuero y caucho
	26		*Productos y Químicos y Conexos*
		261	Elementos y compuestos químicos
		262	Combustibles y lubricantes
		263	Abonos y fertilizantes
		264	Insecticidas, fumigantes y similares
		265	Asfalto y similares
		266	Productos medicinales y farmacéuticos
		267	Tintes, pinturas y colorantes
		268	Productos plásticos, nylon, vinil y P.V.C.
		269	Otros productos químicos y conexos
	27		*Productos de Minerales no Metálicos*

Continuación anexo 1.

		271	Productos de arcilla
		272	Productos de vidrio
		273	Productos de loza y porcelana
		274	Cemento
		275	Productos de cemento, pómez, asbesto y yeso
		279	Otros productos de minerales no metálicos
	28		*Productos Metálicos*
		281	Productos siderúrgicos
		282	Productos metalúrgicos no férricos
		283	Productos de metal
		284	Estructuras metálicas acabadas
		285	Materiales y equipos diversos
		286	Herramientas menores
		289	Otros productos metálicos
	29		*Otros Materiales y Suministros*
		291	Útiles de oficina
		292	Útiles de limpieza y productos sanitarios
		293	Útiles educacionales y culturales
		294	Útiles deportivos y recreativos
		295	Útiles menores médico-quirúrgicos y de laboratorio
		296	Útiles de cocina y comedor
		297	Útiles, accesorios y materiales eléctricos
		298	Accesorios y repuestos en general
		299	Otros materiales y suministros

Continuación anexo 1.

3			**PROPIEDAD, PLANTA, EQUPO E INTANGIBLES**
	31		*Bienes Preexistentes*
		311	Tierras y terrenos
		312	Edificios e instalaciones
		313	Otros bienes muebles preexistentes
		314	Edificios e instalaciones militares
		315	Adquisiciones de bienes de uso común
	32		*Maquinaria y Equipo*
		321	Maquinaria y equipo de producción
		322	Equipo de oficina
		323	Equipo médico-sanitario y de laboratorio
		324	Equipo educacional, cultural y recreativo
		325	Equipo de transporte
		326	Equipo para comunicaciones
		327	Maquinaria y equipo para la construcción
		328	Equipo de cómputo
		329	Otras maquinarias y equipos
	33		*Construcciones por Contrato*
		331	Construcciones de bienes nacionales de uso común
		332	Construcciones de bienes nacionales de uso no común
		333	Construcciones militares
	34		*Equipo Militar y de Seguridad*
		341	Equipo militar y de seguridad

Continuación anexo 1.

	35		*Libros, Revistas y Otros Elementos Coleccionables*
		351	Libros, revistas y otros elementos coleccionables
	36		*Obras de Arte*
		361	Obras de Arte
	37		*Animales*
		371	Animales
	38		*Activos Intangibles*
		381	Activos intangibles
4			**TRANSFERENCIAS CORRIENTES**
	41		*Transferencias Directas a Personas*
		411	Ayuda para funerales
		412	Prestaciones póstumas
		413	Indemnizaciones al personal
		414	Indemnizaciones por perdida de valores
		415	Vacaciones pagadas por retiro
		416	Becas de estudio en el interior
		417	Becas de estudio en el exterior
		419	Otras transferencias a personas
	42		*Prestaciones de Seguridad Social*
		421	Pensiones
		422	Jubilaciones y/o retiros
		423	Prestaciones por incapacidad temporal
		424	Prestaciones globales por incapacidad permanente
		425	Prestaciones especiales de rehabilitación

Continuación anexo 1.

		426	Gastos de entierro
		427	Pensiones a sobrevivientes
		428	Prestaciones por invalidez, vejez y sobrevivencia
		429	Otras prestaciones y pensiones
	43		*Transferencias a Entidades del Sector Privado*
		431	Transferencias a instituciones de enseñanza
		432	Transferencias a instituciones de salud y asistencia social
		433	Transferencias a instituciones científicas y tecnológicas
		434	Transferencias a entidades religiosas
		435	Transferencias a otras instituciones sin fines de lucro
		436	Transferencias a cooperativas
		437	Transferencias a empresas privadas
	44		*Transferencias de Carácter Específico*
		441	Transferencias al Organismo Judicial
		442	Transferencias a la Corte de Constitucionalidad
		443	Transferencias a la Universidad de San Carlos de Guatemala
		444	Transferencias a municipalidades
		445	Transferencias al deporte
		446	Transferencias para alfabetización

Continuación anexo 1.

	45		*Transferencias al Sector Público no Empresarial*
		451	Transferencias a la Administración Central
		452	Transferencias al Instituto Guatemalteco de Seguridad Social –IGSS-
		453	Transferencias a entidades descentralizadas y autónomas no financieras
		454	Transferencias a instituciones públicas financieras
		455	Cuota sostenimiento Superintendencia de Bancos
		456	Servicios Gubernamentales de Fiscalización
		457	Transferencias de impuestos del INFOM a municipalidades
		459	Transferencias a otras entidades del sector publico
	46		*Transferencias al Sector Público Empresarial*
		461	Transferencias a empresas públicas no financieras
	47		*Transferencias Al Sector Externo*
		471	Transferencias a gobiernos extranjeros
		472	Transferencias a organismos e instituciones internacionales
		473	Transferencias a organismos regionales
5			**TRANSFERENCIAS DE CAPITAL**
	51		*Transferencias al Sector Privado*

Continuación anexo 1.

		511	Transferencias a personas y unidades familiares
		512	Transferencias a instituciones sin fines de lucro
		513	Transferencias a cooperativas
		514	Transferencias a empresas privadas
	52		*Transferencias de Carácter Específico*
		521	Transferencias al Organismo Judicial
		522	Transferencias a la Universidad de San Carlos de Guatemala
		523	Transferencias a las municipalidades
		524	Transferencias al deporte
	53		*Transferencias al Sector Público no Empresarial*
		531	Transferencias a la Administración Central
		532	Transferencias a los Consejos de Desarrollo Urbano y Rural
		533	Transferencias a entidades descentralizadas y autónomas no financieras
		534	Transferencias a instituciones públicas financieras
		535	Transferencias de impuestos del INFOM municipalidades
		539	Transferencias a otras entidades del sector publico
	54		*Transferencias al Sector Público Empresarial*

Continuación anexo 1.

		541	Transferencias a empresas públicas no financieras
	55		*Transferencias al Sector Externo*
		551	Transferencias a gobiernos extranjeros
		552	Transferencias a organismos e instituciones internacionales
		553	Transferencias a organismos regionales
6			**ACTIVOS FINANCIEROS**
	61		*Adquisición de Títulos y Valores*
		611	Títulos y valores a corto plazo
		612	Títulos y valores a largo plazo
	62		*Compra de Acciones y/o Participaciones de Capital*
		621	Compra de acciones y/o participaciones a empresas privadas nacionales
		622	Compra de acciones y/o participaciones a empresas públicas no financieras
		623	Compra de acciones y/o participaciones a instituciones públicas financieras
		624	Compra de acciones y/o participaciones a organismos internacionales
		625	Compra de acciones y/o participaciones a otras organizaciones del sector externo
	63		*Concesiones de Préstamos a Corto Plazo*
		631	Préstamos al sector privado
		632	Préstamos a la Administración Central

Continuación anexo 1.

		633	Préstamos a las entidades descentralizadas y autónomas no financieras
		634	Préstamos a instituciones públicas financieras
		635	Préstamos a empresas públicas nacionales
		636	Préstamos a empresas municipales
		637	Préstamos a municipalidades
		638	Préstamos a gobiernos extranjeros y organizaciones del sector externo
	64		*Concesiones de Préstamos a Largo Plazo*
		641	Préstamos al sector privado
		642	Préstamos a la Administración Central
		643	Préstamos a las entidades descentralizadas y autónomas no financieras
		644	Préstamos a instituciones públicas financieras
		645	Préstamos a empresas públicas nacionales
		646	Préstamos a empresas municipales
		647	Préstamos a municipalidades
		648	Préstamos a gobiernos extranjeros y organizaciones del sector externo
	65		*Incremento de Disponibilidades*
		651	Incremento de caja y bancos
		652	Incremento de inversiones financieras temporales
	66		*Incremento de Cuentas a Cobrar*

Continuación anexo 1.

		661	Incremento de cuentas comerciales a cobrar a corto plazo
		662	Incremento de otras cuentas a cobrar a corto plazo
		663	Incremento de cuentas comerciales a cobrar a largo plazo
		664	Incremento de otras cuentas a cobrar a largo plazo
	67		*Incremento de Documentos a Cobrar*
		671	Incremento de documentos comerciales a cobrar a corto plazo
		672	Incremento de otros documentos a cobrar a corto plazo
		673	Incremento de documentos comerciales a cobrar a largo plazo
		674	Incremento de otros documentos a cobrar a largo plazo
	68		*Incremento de Activos Diferidos y Anticipos a Contratistas*
		681	Incremento de activos diferidos a corto plazo
		682	Anticipos a contratistas y proveedores a corto plazo
		683	Incremento de activos diferidos a largo plazo
		684	Anticipos a contratistas y proveedores a largo plazo
		685	Incremento de fideicomisos

Continuación anexo 1.

7			**SERVICIOS DE LA DEUDA PUBLICA Y AMORTIZACION DE OTROS PASIVOS**
	71		*Servicio de la Deuda Pública Interna*
		711	Intereses de la deuda interna a corto plazo
		712	Comisiones y otros gastos de la deuda interna a corto plazo
		713	Amortización de la deuda interna a corto plazo
		714	Intereses de la deuda interna a largo plazo
		715	Comisiones y otros gastos de la deuda interna a largo plazo
		716	Amortización de la deuda interna a largo plazo
		719	Amortización deficiencias netas del Banco de Guatemala
	72		*Servicio de la Deuda Externa*
		721	Intereses de la deuda externa a corto plazo
		722	Comisiones y otros gastos de la deuda externa a corto plazo
		723	Amortización de la deuda externa a corto plazo
		724	Intereses de la deuda externa a largo plazo
		725	Comisiones y otros gastos de la deuda externa a largo plazo
		726	Amortización de la deuda externa a largo plazo

Continuación anexo 1.

	73		*Intereses, Comisiones y Gastos por Préstamos Obtenidos*
		731	Intereses por préstamos del sector privado
		732	Intereses por préstamos del sector público no financiero
		733	Intereses por préstamos del sector público financiero
		734	Intereses por préstamos del sector externo
		735	Comisiones y gastos por préstamos del sector privado
		736	Comisiones y gastos por préstamos del sector público no financiero
		737	Comisiones y gastos por préstamos del sector publico financiero
		738	Comisiones y gastos por préstamos del sector externo
	74		*Amortización de Préstamos a Corto Plazo*
		741	Amortización de préstamos del sector privado
		742	Amortización de préstamos de la Administración Central
		743	Amortización de préstamos de entidades descentralizadas y autónomas no financieras
		744	Amortización de préstamos de instituciones públicas financieras
		745	Amortización de préstamos de empresas públicas no financieras

Continuación anexo 1.

		746	Amortización de préstamos de municipalidades
		747	Amortización de préstamos de gobiernos extranjeros
		748	Amortización de préstamos de organismos e instituciones regionales e internacionales
	75		*Amortización de Préstamos a Largo Plazo*
		751	Amortización de préstamos del sector privado
		752	Amortización de préstamos de la Administración Central
		753	Amortización de préstamos de entidades descentralizadas y autónomas no financieras
		754	Amortización de préstamos de instituciones públicas financieras
		755	Amortización de préstamos de empresas públicas no financieras
		756	Amortización de préstamos de municipalidades
		757	Amortización de préstamos de gobiernos extranjeros
		758	Amortización de préstamos de organismos e instituciones regionales e internacionales
	76		*Disminución de Cuentas a Pagar a Corto y Largo Plazo*
		761	Disminución de cuentas comerciales a pagar a corto plazo

Continuación anexo 1.

		762	Disminución de cuentas a pagar a corto plazo con contratistas
		763	Disminución de gastos de personal a pagar a corto plazo
		764	Disminución de cuentas a pagar por impuestos, tasas y derechos a corto plazo
		765	Disminución de cuentas a pagar por intereses a corto plazo
		766	Disminución de otras cuentas a pagar a corto plazo
		767	Disminución de cuentas comerciales a pagar a largo plazo
		769	Disminución de otras cuentas a pagar a largo plazo
	77		*Disminución de Depósitos de Instituciones Públicas Financieras*
		771	Disminución de depósitos a la vista
		772	Disminución de depósitos en cuentas de ahorros y plazo fijo
	78		*Disminución de Documentos a Pagar a Corto y Largo Plazo y Conversión de la Deuda*
		781	Disminución de documentos comerciales a pagar a corto plazo
		782	Disminución de otros documentos a pagar a corto plazo
		783	Disminución de documentos comerciales a pagar de largo plazo

Continuación anexo 1.

		784	Disminución de otros documentos a pagar de largo plazo
		785	Conversión de la deuda interna a largo plazo, en a corto plazo
		786	Conversión de la deuda interna a corto plazo, en a largo plazo
		787	Conversión de la deuda externa a largo plazo, en a corto plazo
		788	Conversión de la deuda externa a corto plazo, en a largo plazo
	79		*Disminución de Otros Pasivos*
		791	Disminución de pasivos diferidos a corto plazo
		792	Disminución de pasivos diferidos a largo plazo
		793	Disminución de provisiones para cuentas incobrables
		794	Disminución de provisiones para pérdidas de inventarios
		795	Disminución de provisiones para beneficios sociales
		796	Disminución de reservas técnicas
		799	Disminución de otros pasivos a corto plazo
8			**OTROS GASTOS**
	81		*Impuestos Directos*
		811	Impuestos Directos
	82		*Descuentos y Bonificaciones*

Continuación anexo 1.

		821	Descuentos por ventas
		822	Bonificaciones por ventas
		823	Devoluciones
	83		*Depreciación y Amortización*
		831	Depreciación del activo fijo
		832	Amortización del activo intangible
	84		*Beneficios Sociales*
		841	Beneficios Sociales
	85		*Reservas Técnicas*
		851	Reservas legales
		852	Reservas de capitalización
		859	Otras reservas técnicas
	86		*Otras Pérdidas*
		861	Cuentas incobrables
		862	Pérdidas de inventarios
		863	Pérdidas por venta de activos
		864	Pérdidas por operaciones cambiarias en intereses
		865	Pérdidas por operaciones cambiarias en amortización
		866	Otras pérdidas de operaciones
		867	Otras pérdidas ajenas a la operación
		868	Pérdidas en la colocación de valores públicos
	87		*Disminución del Patrimonio*
		871	Disminución del capital social e institucional
		872	Disminución de aportes por capitalizar
		873	Disminución de reservas

Continuación anexo 1.

		874	Disminución de resultados por dividendos a distribuir
		875	Disminución de resultados por utilidades a transferir a la Administración Central
		876	Disminución de resultados por utilidades a transferir a empleados
		877	Disminución de resultados acumulados
		878	Disminución de resultados por utilidades a transferir a municipalidades
	88		*Intereses de Instituciones Públicas Financieras*
		881	Intereses de instituciones públicas financieras bancarias
		882	Intereses de instituciones públicas financieras no bancarias
	89		Reclamos por Seguros de Hipotecas
		891	Reclamos por seguros de hipotecas
9			**ASIGNCIONES GLOBALES**
	91		*Gastos Imprevistos*
		911	Emergencias y calamidades públicas
		912	Siniestros y gastos conexos
		913	Sentencias judiciales
		914	Gastos no previstos
	99		*Créditos de Reserva*
		991	Créditos de reserva

Fuente: Manual de clasificaciones presupuestarias para el sector público de Guatemala.

.