

# DESARROLLO DE UNA SOLUCIÓN INFORMÁTICA PARA EL ANÁLISIS DE DATOS DEL SISTEMA DE PAGOS DE SEGURIDAD SOCIAL PARA EL SECTOR FINANCIERO COLOMBIANO<sup>1</sup>

DEVELOPMENT OF A COMPUTER  
SOLUTION FOR THE ANALYSIS OF DATA  
WITHIN SOCIAL SECURITY PAYMENT  
SYSTEM FOR THE COLOMBIAN FINANCIAL  
SECTOR

DESENVOLVIMENTO DE UMA  
SOLUÇÃO DE COMPUTADOR PARA  
ANÁLISE DE DADOS DO SISTEMA DE  
PAGAMENTO DE PREVIDÊNCIA PARA  
O SETOR FINANCEIRO COLOMBIANO

**Hermann Fuquen González**

Grupo de investigación  
COLINNOVACION.  
hfuquen@colinnovacion.com

**Juan Carlos Salavarieta**

Grupo de investigación  
COLINNOVACION.  
jcsalavarieta@colinnovacion.com

**Fernando Soto**

Investigador del proyecto ACH  
COLOMBIA Bogotá (Colombia);  
fasoto@achcolombia.com.co

6

**Fecha de Recepción:** 10 de abril de 2020

**Fecha aprobación:** 15 de junio de 2020

---

<sup>1</sup> Producto derivado del proyecto calificado como de innovación con el título “DESARROLLO DE UNA SOLUCIÓN INFORMÁTICA INTEGRADA CON EL SECTOR FINANCIERO QUE PERMITA LA CONSULTA Y ANÁLISIS DE INFORMACIÓN DE PILA – PLANILLA INTEGRADA DE LIQUIDACION DE APORTES”. Presentado y aprobado en la convocatoria de Beneficios Tributarios de COLCIENCIAS para ACH Colombia

## Resumen

Se presenta el desarrollo de una solución informática basado en una plataforma tecnológica que permite la consulta en línea de grandes volúmenes de información contenida en el sistema PILA – Planilla Integrada de Liquidación de Aportes, sistema donde se registran los pagos a la seguridad social de los trabajadores colombianos activos y pensionados. Esta información podrá ser utilizada por parte del sector bancario y entidades financieras del país para la toma de decisiones, con el fin de brindar un mayor acceso a los servicios financieros de personas que tienen limitada su posibilidad de acceso a servicios por falta de información fiable. La plataforma propuesta contempla distintos módulos de control definidos como servicios Web-API que permite la administración de varios servicios divididos en módulos. Los módulos funcionan como micro-componentes de aplicaciones, las cuales interactúan entre ellos a través del sistema de mensajería Kafka (Basado en una arquitectura definida de tubería de datos) a nivel de integración, operando con una arquitectura orientada a recursos para consumir servicios entre ellos.

**Palabras claves:** Análisis de Datos, Balanceador de Carga, Big Data, Pagos Seguridad Social, Plataformas Informáticas, Seguridad Informática, Servicios Web.

## Abstract

The article shows the development of a computer solution based on a technological

platform that allows online consultation of information contained in the system PILA – (Integrated Contribution Payment Form), where the payments to social security of active Colombian workers and pensioners are registered. This information could be use in the banking sector and financial institutions for decision-making. Therefore, the proposed system will provide greater access to financial services for people who have limited access to financial services due to a lack of reliable information. The proposed platform includes control modules defined as web API services that allows the administration of several services divided into modules. The modules work as micro-components of applications, which interact with each other through the Kafka messaging system (Based on a data pipeline architecture) at the integration level, operating with a resource-oriented architecture to consume services between them.

**Keywords:** Data Analysis, Load Balancer, Big Data, Social Security Payments, Computer Platforms, Computer Security, Web Services.

## Resumo

É apresentado o desenvolvimento de uma solução computacional baseada numa plataforma tecnológica que permita a pesquisa on-line de grandes volumes de informações contidas no sistema PILA - Sistema de Liquidação de Contribuições Integradas, sistema em que são registrados os pagamentos de seguridade social dos trabalhadores colombianos ativos e aposentados, Essas informações podem ser

usadas pelo setor bancário e pelas instituições financeiras do país para tomada de decisão, a fim de fornecer maior acesso a serviços financeiros para pessoas com acesso limitado a serviços devido à falta de informações confiáveis. A plataforma proposta inclui diferentes módulos de controle definidos como serviços de API da Web que permitem a administração de vários serviços divididos em módulos. Os módulos funcionam como micro-componentes de aplicativos, que interagem entre si por meio do sistema de mensagens Kafka (com base em uma arquitetura definida de canal de dados) no nível de integração, operando com uma arquitetura orientada a recursos para consumir serviços entre eles.

**Palavras chaves:** Análise de Datos, Balanceador de Carga, Big Data, Pagamentos de Seguro Social, Plataformas de TI, Segurança de Computadores, Serviços da Web.

## Introducción

El presente artículo presenta los resultados del proyecto desarrollado internamente por ACH Colombia en el cual se implementó un sistema informático que permite el análisis y consulta de la información del sistema de seguridad social PILA (Planilla Integrada de Liquidación de Aportes) para el sector financiero Colombiano, servicio de información el cual fue nuevo en el mercado Colombiano ya que no se contaba con un sistema de consulta estructurado y fiable sobre los datos laborales de cotizantes al sistema de seguridad social sustentado en sistemas informáticos avanzados.

Como antecedente a este proyecto, se contaba con un sistema ya estructurado y estandarizado de pagos de seguridad social ejecutado por distintas entidades, como un actor entre los denominados "Operadores de Información" en Colombia, que son regulados por el Ministerio de Salud y Protección Social [7]. En general, la información recopilada por los operadores de información es estandarizada y recolectada en bases de datos especializadas para tal fin, sin embargo, esta información no era aprovechada ni analizada para otros fines diferentes al control de pagos de la seguridad social de los trabajadores colombianos [8] hasta el desarrollo de la presente propuesta, para la plataforma SOI (Servicio Operativo de Información).

De otra parte, cabe resaltar que esta solución parte del problema de ausencia de información verídica que enfrentan las entidades financieras para la aprobación de productos financieros, que dependen muchas veces de certificados y documentos físicos, fácilmente alterables, que aumentan el riesgo de fraude. Por tanto, a través del proyecto se mitiga el riesgo de fraude de la información que pudiera proveer el mismo usuario. También se identificó que muchas personas tienen limitada su posibilidad de acceso a servicios financieros por falta de información idónea que respalde su historial financiero.

Este proyecto se enmarca en una innovación en proceso al complementar distintas técnicas informáticas para alcanzar una solución tecnológica desarrollada con capacidades propias de ACH Colombia que a través de una plataforma contempla distintos módulos de control definidos como servicios web (Web API) como son:

- Estadísticas y Tarificación: Realizar el registro de la actividad de los usuarios y sus consultas y estimar los esquemas de tarificación según los planes contratados con cada entidad financiera.
- Administración de Usuarios: Administra los usuarios del portal web a desarrollar para controlar su acceso.
- Monitoreo: Para uso interno de los administradores de la plataforma donde se verifica que las operaciones de cargue de información, consulta entre otras, se realicen de manera adecuada.
- Auditoría: Registro de todos los eventos que sucedan dentro de la plataforma como consultas, fallas de sistema, registros o modificaciones.
- Migración: Modulo correspondiente a la lectura de las fuentes transaccionales, realizando el proceso de ingestión de datos para cargar la información en la base de datos (cassandra) adaptando la información según las resoluciones definidas por las entidades competentes en reglamentar el sistema de seguridad social.
- Consultas: Componente encargado de hacer disponible la información persistente para las consultas del banco.

Los anteriores módulos funcionan como micro-componentes de aplicaciones, las cuales interactúan entre ellos a través del sistema de mensajería Kafka (Basado en una arquitectura definida de tubería de datos) a nivel de integración, operando en REST (Arquitectura orientada a recursos) para consumir servicios entre ellos.

El desafío tecnológico consiste en desarrollar esta plataforma basada en la información de PILA, lo cual implica una gestión inteligente de

la información que se sustenta en sistemas computacionales avanzados con sistemas de procesamiento de datos que permiten la gestión y reporte eficiente de información. El desarrollo propuesto definido a través de una arquitectura informática basada en servicios Web, REST (Arquitectura Orientada a Recursos), teoría de Grafos, balanceador HA Proxy, tubería de datos, bases de datos inmutables entre otros, enmarca la innovación en procesos informáticos como un conjunto de elementos de software, hardware que convergen a una final oferta en nuevos servicios al ofrecer información a las entidades financieras colombianas de alta fiabilidad y pertinencia para sus fábricas de crédito.

## Conceptos y marco teórico

En esta sección se describen algunos conceptos aplicados en la construcción de la plataforma tecnológica de ACH Colombia, entre los que se encuentran las tendencias en Big Data, Internet de las Cosas, SOA – Arquitectura Orientada a Servicios, Modelos de Web Services, balanceadores de carga entre otros.

### Tendencias en Big Data y Gestión de datos

En la literatura se establece el termino Big Data como una situación especial que se presenta en empresas o entidades que cuentan con altos volúmenes de información, donde la velocidad y la variedad de los datos exceden las capacidades de almacenamiento o computación para generar decisiones acertadas y oportunas [9]. Sin embargo, varios autores y especialistas del área de análisis de datos entienden el Big Data como un conjunto de herramientas y

técnicas necesarias para acceder, organizar y coleccionar conocimientos de grandes volúmenes de datos digitales que pueden ser estructurados o no [10].

Por tanto, un conjunto de nuevas herramientas informáticas ha sido desarrollada para el intercambio de datos, para lo cual fue necesario establecer parámetros específicos para necesidades comunes del sector industrial así como de las necesidades perceptibles de la sociedad y los mismos ciudadanos [1]. Por tanto, fue una prioridad la vinculación de cada uno de los actores en diferentes escenarios para establecer modelos estándares, medios de interacción comunes y protocolos, para enlazar plataformas, lenguajes tecnológicos específicos como XML, ASN.1, Web services etc. De esta forma investigaciones y desarrollos se continúan enfocando en aplicaciones específicas que soporten la interoperabilidad entre máquinas sobre una red, así surgió el concepto de virtualización o servicios web orientados a mejorar la interoperabilidad de los usuarios terminales usando internet [1].

Las arquitecturas y sistemas disponibles para soportar la interoperabilidad de máquinas y por ende de usuarios se resumen las más relevantes para este proyecto a continuación:

### **SOA – Arquitectura Orientada a Servicios (Service Oriented Architecture)**

Esta es una propuesta de arquitectura informática para la construcción de aplicaciones basadas en servicios disponibles

en la red. Las aplicaciones desarrolladas bajo SOA pueden ser reutilizables ya que deben ser diseñadas para ser flexibles, adicionalmente son basadas en servicios definidos bajo funcionalidades corporativas, pudiendo ser usadas por clientes desde otras aplicaciones o procesos empresariales. SOA garantiza el uso de los componentes empresariales actuales, así como la infraestructura física para mejorar la flexibilidad en la comunicación de servicios o procesos ya creados. Cada servicio tiene independencia para la resolución de tareas específicas, y adicionalmente permite la integración con otros servicios según su definición [11].

Existe un paradigma usado en este tipo de arquitectura conocido como de Encontrar-Conectar-Ejecutar (find-bind-execute), donde el proveedor del servicio lo registra abiertamente; con este registro es posible encontrar el servicio dependiendo de los criterios de búsqueda por parte del cliente. De esta forma, si el registro contiene el servicio solicitado se indica el contrato y la ubicación final del servicio [11].

Este tipo de servicios se conoce a nivel de aplicaciones sobre la web como Web Services, lo cual se podría describir como la tecnología que permite la conexión en una arquitectura SOA en internet o una red [11].

### **Web Services**

Este tipo de servicios está cambiando de forma radical el campo de acción web aumentando las capacidades de las transacciones en línea, caracterizadas por la interrelación entre mercados o negocios denominada interacción negocio a

consumidor (business-to-consumer - B2C). Este tipo de transacciones se caracteriza por modelos de comunicación entre programas o aplicación Web Services basados en estándares emergentes como de Transferencia de Protocolo de Hipertexto (HyperText Transfer Protocol - HTTP), Lenguaje de Marcado Extensible (Extensible Markup Language - XML), Protocolo de Acceso a Objetos Simple (Simple Object Access Protocol - SOAP), Lenguaje de Descripción de Servicios Web (Web Services Description Language - WSDL), y el catálogo de negocios de Internet (Universal Description, Discovery, and Integration - UDDI) [12]

En términos generales, las tecnologías basadas en Web Services proveen un lenguaje natural o modelo de programación en un ambiente normalizado que acelera la integración de aplicaciones dentro y fuera de las corporaciones. La integración de aplicaciones mediante los Web Services genera sistemas empresariales con bajo acoplamiento y alta cohesión, ya que este tipo de servicios basados en la web, encapsulan aplicaciones existentes junto con activos empresariales relacionados con sistemas de información [12].

Los Web Services son una interface que describe una serie de operaciones con acceso a través de red mediante mensajes servicios web basados en REST full, permitiendo la mejora de tareas específicas o un conjunto de tareas. La definición de este tipo de servicios es posible gracias a la utilización de REST (Representational State Transfer), el cual permite mayor flexibilidad, facilidad de implementación y está teniendo un crecimiento en las arquitecturas de ampliación de servicios, incluyendo el

formato de los mensajes para especificar las operaciones, requiriendo menos memoria y menos capacidad de operación que otros servicios web [12].

Un sistema básico de Web Service tiene dos actores. Quien solicita y usa el servicio (Consumidor), así como el Proveedor del servicio que presenta la interface y la implementación del Web Service [13]. En el próximo apartado se resumen los modelos estandarizados basados en la arquitectura funcional de los Web Services para diferentes requerimientos de los usuarios finales.

Los Web Services usan el protocolo de transporte HTTP como medio para el intercambio de información, en donde interactúan fundamentalmente el cliente y el servidor. El cliente genera la petición y el servidor recibe, verifica y genera una respuesta hacia el cliente, dicha comunicación se realiza de forma remota usando la WEB, quien provee una serie de estándares y definiciones para permitir una transmisión ordenada basado en el modelo cliente/servidor. De esta forma, existen varias arquitecturas que permiten el acceso a los Web Service desde diferentes perspectivas, las cuales han evolucionado a través de la industria TI [2].

### **XML-RPC – Remote Procedure Call**

RPC, es un protocolo desarrollado bajo UNIX, como método para construir aplicaciones entorno a sistemas con características diferentes usando Internet. Este modelo permite la división de aplicaciones en componentes y otorga soporte para la comunicación en red [3].

Según la definición de comunicación remota RPC, fue desarrollado un mecanismo basado en lenguaje XML y el protocolo HTTP para enviar y recibir información de un computador denominado cliente, a un computador tipo servidor, con una codificación básica XML [3].

### **Protocolo de Acceso a Objetos Simple (SOAP – Simple Object Access Protocol)**

Es un protocolo propuesto por un consorcio internacional web denominado W3C encargada de desarrollar protocolos y estándares con el fin asegurar el correcto uso de la Web. Este protocolo permite la interconexión entre Web Services, basado en el procedimiento remoto XML-RPC. Por tanto se estimaría a SOAP como una evolución del protocolo mencionado anteriormente, más completo y maduro en términos de mejorar la comunicación de procedimientos funcionales a servicios basados en el lenguaje XML [3].

### **REST – Modelo de Arquitectura Orientada a Recursos**

Se podría traducir como Transferencia de Estado Representacional, no es un protocolo, se acerca más a una representación de arquitectura para la construcción de Web Services a través de Internet. Este modelo fue el resultado de una tesis de doctorado del año 2000 por Roy Fielding, quien también es uno de los autores del protocolo HTTP [4].

Este tipo de modelo se usa como un estilo de arquitectura para el diseño de aplicaciones Web Services en red. Facilitando la conexión entre máquinas con un uso sencillo de HTTP a diferencia de otro tipo de modelos complejos como RPC o SOAP este estilo de arquitectura

tiene el mismo esquema de cliente-servidor, es decir, se genera la solicitud por parte del cliente, el servidor procesa la solicitud y genera la respuesta al cliente. La diferencia radica en que la respuesta y la solicitud están enfocadas en recursos. Tiene la ventaja de aprovechar la existencia de aplicaciones, sistemas, objetos actuales para producir aplicaciones de forma más simple que otro tipo de arquitecturas, ya que fue basado en la simplicidad de la web, y además de heredar el uso de XML y HTTP. Así mismo está en capacidad de adoptar otros estándares dispuestos para la creación de Web Services adicionales [2].

Los Web Services en este tipo de arquitectura expresan su información y funcionalidad bajo la denominación de recurso, el cual se identifica a través de un URI (Identificación de Recurso Uniforme) [2]. La interacción de los clientes con el recurso debe estructurarse bajo palabras, que para el protocolo HTTP se denominan verbos, validando el paradigma de las arquitecturas orientadas al servicio (encontrar, enlazar y ejecutar), pero en este caso como es orientado al concepto de recurso. Se tiene Leer (Get), Crear (Post), Actualizar (Put) y Eliminar (Delete) [14]. El concepto de recurso basado en HTTP puede verificarse gráficamente en la Fig. 1:

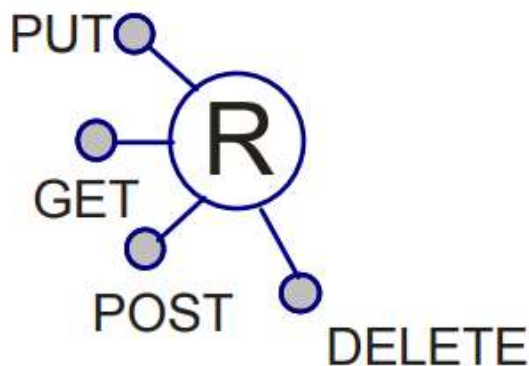


Fig. 1. Definición de Recurso en REST.

Fuente: Pautasso (2010)

### Balancedores de Carga

Debido al alto volumen de información y de transacciones esperado en la plataforma propuesta por ACH Colombia en este proyecto, a continuación, se revisan distintas metodologías para el balanceo de carga de datos.

#### Balancedor Software HA-Proxy

Este tipo de balanceo de carga permite a un sistema la participación de varios servidores en un mismo servicio y realizar el mismo trabajo. Por tanto, el uso de balancedores de carga para evitar que el servidor llegue a su límite de carga de procesos, monitoreando cada uno de ellos es indispensable para procesos que generan una alta cantidad de tráfico de información [5].

HA-Proxy es una solución gratuita, rápida y fiable, que ofrece a los usuarios un proxy para TCP y HTTP de alta disponibilidad, con control de balanceo de carga. Ha sido usada por años y se definiría como un estándar de código abierto compatible con varias distribuciones de Linux. La ventaja de este tipo de dispositivos virtuales es la facilidad para acoplarse en arquitecturas de red

existentes, sin exponer la información de servidores hacia la web [5].

### Apache Kafka

Es una solución para la intermediación de mensajes para la solución de problemas en tiempo real de cualquier aplicación tipo software, es usado en ambientes de trabajo con gran cantidad de volumen de información, garantizando la distribución eficiente a varios consumidores de forma ágil. Debido a la necesidad de crear un controlador de mensajes para la conexión entre aplicaciones, fue incluido el termino Message Broker en los términos relacionados con Big Data, o concentrador de mensajes, como si se tratara de un servidor de mensajes permitiendo un tratamiento específico según el tipo de información que se requería obtener de las aplicaciones. Kafka provee una integración compacta entre la información generada por los clientes y proveedores, evitando bloqueos en el flujo de la información [15].

Kafka es una solución tipo Open Source, definido como un sistema de control de mensajería distribuida de aplicaciones web, creado con las siguientes características [15]:

**Mensajería Persistente:** Diseñado para evitar en cualquier forma la pérdida de datos.  
**Alto caudal para procesar información:** diseñado bajo la idea de Big Data, está en capacidad de soportar millones de mensajes por segundo.

**Distribuido:** permite un orden teniendo en cuenta que existen mensajes dispersos en y hacia varios destinos.



Soporte de Múltiples Cliente: está en capacidad de soportar varias plataformas como Java, .NET, PHP, Ruby, y Python.

Tiempo Real: mensajes producidos por los proveedores deben ser inmediatamente vistos por los clientes, basado en sistemas basados en eventos como Complex Event Processing (CEP).

### Sistema de Bases de Datos Cassandra

La Base de Datos Inmutable – Cassandra se originó en 2007 dentro de Facebook como método para solucionar un problema de la compañía, donde existía un gran volumen de información y no era posible escalar con los métodos tradicionales en su momento [16].

Apache Cassandra es una base de datos NoSQL distribuida y basada en un modelo de almacenamiento de «clave-valor», de código abierto que está escrita en Java. Permite grandes volúmenes de datos en forma distribuida; por ejemplo, lo usa Twitter para su plataforma. Su objetivo principal es la escalabilidad lineal y la disponibilidad. La arquitectura distribuida de Cassandra está basada en una serie de nodos iguales que se comunican con un protocolo P2P con lo que la redundancia es máxima. Está desarrollada por Apache Software Foundation. Cassandra ofrece soporte robusto para múltiples centros de datos, con la replicación asincrónica sin necesidad de un servidor maestro, que permiten operaciones de baja latencia para todos los clientes [16].

### Metodología SCRUM

Este Proyecto fue desarrollado bajo

metodología SCRUM el cual es un framework para administrar el desarrollo de productos complejos. No se puede catalogar como un proceso o técnica, pero dentro de SCRUM es posible aplicar varios de estos [17].

SCRUM por tanto se define generalmente como un marco de trabajo para la gestión ágil de proyectos, usado para generar entregables con alto valor agregado al cliente en forma iterativa. SCRUM es una metodología de desarrollo de productos la cual es incremental y evolutiva. Los requisitos se identifican y se listan en un lugar definido llamado el backlog del producto. Las iteraciones, llamadas Sprints, normalmente duran 30 días. En cada sprint el grupo de desarrollo selecciona del backlog un conjunto de ítems de mayor prioridad, y los desarrolla de tal forma que el backlog se convierte en el artefacto base de la medida de progreso del proyecto. Durante el sprint el equipo trabaja en sus tareas sin modificarlas con nuevos requisitos. Todos los días los miembros del equipo se reúnen (SCRUM diario) con el líder del equipo (Master SCRUM) para contestar las tres preguntas referidas al progreso del proyecto [17].

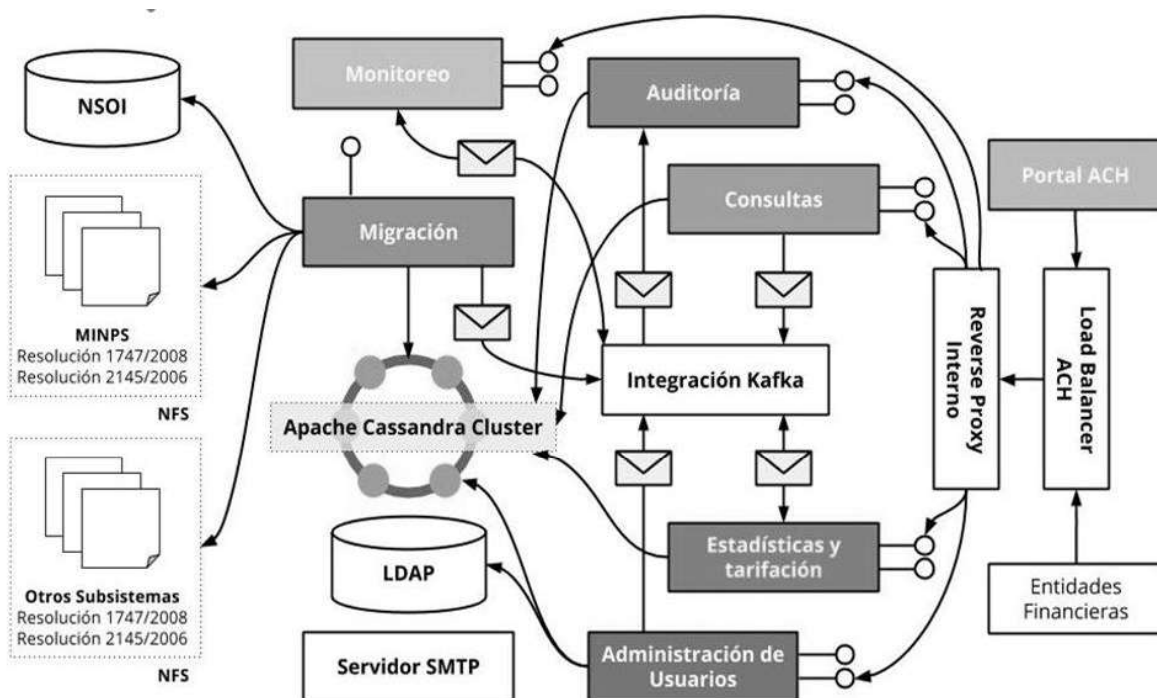
Este framework consiste en la interacción de diferentes equipos dentro de un grupo de trabajo con la definición para cada uno de ellos de roles, eventos y reglas. Cada uno de estos componentes tiene un propósito específico y es esencial para el éxito y el uso de SCRUM [17].

El corazón de este tipo de metodologías es el Sprint, un periodo de tiempo que dura entre 2 semanas o hasta 1 mes, donde se genera un avance del producto a desarrollar. Cada Sprint inicia al momento de terminar el anterior. Está compuesto por los siguientes componentes [6]:

Planeación del Sprint: El trabajo a ser desarrollado por el Equipo SCRUM es planificado. Reunión Diaria de SCRUM: Es un

periodo de 15 minutos para coordinar las actividades que serán desarrolladas en el día. Revisión del Sprint: Es desarrollada al terminar el Sprint para verificar la evolución del desarrollo del producto y evaluar modificaciones según los atrasos generados, si existen. Retrospectiva del Sprint: Es realizado por el equipo SCRUM para sugerir mejoras internas para que sean aprobadas en el siguiente Sprint.

Fig. 2. Modelo de arquitectura funcional informática.



Fuente: Autores

### Transacciones Electrónicas por Internet en Colombia

Para que la presente iniciativa pudiera tener el impacto esperado, es primordial identificar las diferentes tendencias del mercado y un indicador fundamental es el nivel de penetración de Internet en Colombia. Gracias a este indicador se puede identificar la oportunidad de invertir en nuevos sistemas,

herramientas y procesos que den seguridad, promoviendo el uso de sistemas de pago electrónico. Debido a que esta iniciativa depende de la difusión en el uso de los procesos de pago electrónico de los aportes a la seguridad social a través de PILA en Colombia, es importante resaltar el desarrollo que ha tenido el país en cuanto a los sistemas de pagos electrónicos.

Al aumentar los niveles de confianza para hacer pagos, mayor será el número de usuarios que aprovecharán los sistemas de en línea mejorando el impacto de este proyecto. En Colombia se evidencia un alto componente de disponibilidad de tecnología asociada con el uso de internet y herramientas TIC. El primer trimestre (1T) de 2016 presentó un aumento en las conexiones de internet banda ancha representados en 25.552.594 suscriptores en el país, lo que representa un 52.4% de penetración. Para el mismo periodo, pero del año 2018 (1T-2018) el número de suscriptores aumentó a 28.388.361 lo que representa un aumento de 5.2% en un año [19].

También se resalta que los Medios de pago electrónicos en Colombia (MPE) los cuales representan un servicio adicional que prestan entidades financieras y comerciales las cuales requieren de una plataforma tecnológica certificada para realizar este tipo de transacciones. En los últimos años se ha incrementado el número de canales electrónicos, como el Internet (por medio de terminales fijas y móviles), el uso de tarjetas de crédito y el uso de Internet en teléfonos móviles. Esta tendencia promueve el acceso a mayor escala de servicios financieros, pagos más eficientes, más seguridad en las transacciones y disminución en trámites y costos [18].

## Materiales y métodos

Para el desarrollo de este proyecto se definieron varias etapas contenidas en actividades específicas. A continuación, se

destacan la metodología ejecutada para este proyecto.

### Diseño de arquitectura informática basada en Web-Services

Este proyecto fue liderado por la coordinación de Desarrollo y Arquitectura TI (CDA-TI) de ACH Colombia, que se encarga del desarrollo y la prestación de servicios tecnológicos específicos de ACH Colombia hacia clientes externos, recolectando las necesidades expuestas por los bancos a través de reuniones de trabajo para permitir consolidar una arquitectura estándar que minimice el impacto a las operaciones internas de los actores involucrados del servicio a desarrollar por parte de ACH. Por tanto, se desarrolló una primera etapa de Definición de necesidades y validación para el servicio, seguido de una segunda etapa de Modelación Funcional de la Solución TI para Consulta PILA.

En la segunda etapa se determinó que la solución tecnológica denominada SOI Data, estaría basada en el modelo SOA, específicamente a una representación de arquitectura basada en los recursos actuales que se encuentran desplegados en las infraestructuras disponibles entre ACH-Colombia y las entidades financieras, este modelo de sistema distribuido o para el caso de aplicaciones distribuidas denominado REST, es fundamental para generar desde la

CDA-TI un modelo de flujo de información para consultadas estructuradas. Por tanto, se implementó un modelo de arquitectura funcional de solución tecnológica, el cual se puede apreciar de forma detallada en la Fig.2.

## Diseño desde el punto de vista del Front Office

Desde el punto de vista del Front Office en este modelo (Ver Fig. 2, parte derecha), el usuario de este sistema, en este caso la entidad Financiera, podrá realizar desde el portal web de ACH-Colombia (Portal ACH -conectado desde cualquier terminal externa con conexión a internet), el envío de requerimientos a través del Balanceador de Carga ACH (Load Balancer ACH en Fig. 2), direccionando las solicitudes de los usuarios inmediatamente al Balanceador HA-Proxy (Reverse Proxy Interno) seleccionado para la nueva solución tecnológica. La comunicación entre la entidad financiera y el banco se realiza mediante los canales de comunicación equipados con encriptación para garantizar la seguridad de la información que se envía.

Las micro-aplicaciones generadas por los Web API dispuestos a través de los módulos diseñados (Ver la Fig. 2, cuadros resaltados) requirieron infraestructura física adicional y dedicada, como parte de la selección de tecnologías. Estos recursos físicos adicionales están concentrados en el recurso web dispuesto para este tipo de sistemas distribuidos, minimizando los requerimientos de software a ser especificados, otra de las ventajas de servicios web basados en micro aplicaciones.

A través de Kafka se controlan los mensajes generados por los Web-API basados en las micro-aplicaciones generadas por la consulta realizada a través de las entidades financieras hacia ACH-Colombia, a través de cada una de las API o módulos expuestos en la Fig. 2, como por ejemplo Consultas, Auditoría, Migración etc. Existe un controlador a nivel informático, para gestionar los servicios

generados por las aplicaciones, y permitir tener una arquitectura distribuida de servicios como la planteada por ACH-Colombia a través de REST. Zookeeper se encargará de administrar los servicios generados a través del control de mensajería Kafka, permitiendo que la información esté habilitada en un flujo continuo entre las entidades financieras y el proveedor que para este proyecto será ACH-Colombia.

Este tipo de servicios como fue explicado anteriormente, podría ser proporcional a una serie de recursos físicos y virtuales según la concepción de la solución, pero en este caso y a partir de la tarea realizada entre la CDA-TI y distintos proveedores, fue posible aprovechar la infraestructura física actual de ACH-Colombia.

## Diseño desde el punto de vista del Back Office

En cuanto a las actividades de apoyo del sistema (Back Office) de esta solución se determinó la parte transaccional y marco legal de la Solución Tecnológica, donde la Nueva Base de datos de SOI (NSOI), contiene los archivos relacionados con la legislación que rige la seguridad social en Colombia, actualmente o aquellos que eran modificados o expedidos por el Ministerio de Salud y Protección Social, la cual rige a la Planilla Integrada de Liquidación de Aportes – PILA, y adicionalmente contiene la información de los usuarios aportantes a la seguridad social en Colombia y que son usuarios únicamente de la plataforma SOI. Esta situación genera flexibilidad en esta solución diseñada para beneficiar a las personas naturales y jurídicas al tener información centralizada de su histórico de contribuciones. El módulo de migración que se describirá más adelante alojara esta

información en la base de datos Cassandra, obteniendo los beneficios que este tipo de bases de datos no relacional dispone para el alojamiento de datos basados en sistemas distribuidos, garantizando un correcto tratamiento de la información en la solución propuesta por ACH-Colombia.

Todo el desarrollo de la plataforma descrita se realizó bajo metodología SCRUM. Para el desarrollo de los módulos se desarrollarán puntualmente las siguientes actividades:

- Se desarrollarán los modelos de extracción de información, los cuales permitirán extraer de las diferentes fuentes (Minps y Otros Operadores) los datos que integrarán la base de datos del servicio SOI DATA.
- Desarrollo de modelos de integración WebAPI. Este es el mecanismo que permitirá realizar el consumo de la información desde el banco hacia el sistema de información SOI DATA, específicamente con los campos que requiera la entidad que ejecuta la consulta
- Desarrollo de módulos de Auditoría y Control de Eventos. Estos módulos permitirán realizar seguimiento a las consultas realizadas por cada una de las entidades vinculadas al servicio, identificando a usuarios que hayan realizado operación sobre el sistema de información SOI DATA.

## Resultados

La implementación de toda la arquitectura dio como resultado la nueva plataforma informática que permitió la consulta de la información por parte de las entidades financieras. Inicialmente se realizó una prueba piloto con una entidad bancaria para

verificar la funcionalidad de la plataforma y probar las capacidades de tráfico esperadas. Posteriormente se vincularon otras tres entidades financieras adicionales al piloto y se vincularán 4 entidades adicionales en el 2018, con estas entidades se procedió con el proceso de implementación tecnológica según las actividades a desarrollar las cuales se describen a continuación:

- Desarrollo Pruebas y Certificación Banco, estos puntos corresponden a las actividades que se llevan a cabo con cada entidad financiera para obtener sus desarrollos, pruebas y certificación de las diferentes funcionalidades definidas para la vinculación con ACH Colombia.
- Integraciones Tecnológicas ACH-Banco, hace parte de la conexión propia que debe hacerse entre ACH Colombia y la entidad financiera que se vinculará al servicio, esto teniendo en cuenta los requerimientos técnicos definidos a nivel de comunicaciones, seguridad informática y mecanismos de consulta (Ej: Web API).
- Piloto ACH Banco: Es la salida a producción de manera controlada con la entidad financiera definida para ello, tiene un período aproximado de dos semanas para cada funcionalidad (Integración banco y Portal).
- Para la implementación del servicio en los bancos que entran a utilizar la información, se entregaron los requerimientos de orden técnico para la vinculación de estas entidades al servicio.
- Dentro del proceso de implementación se ejecutaron las pruebas de certificación entre ACH y la entidad a vincular.

Dentro de la fase de integración con Bancos del proyecto se han definido 7 Sprints según la metodología SCRUM; cada sprint corresponde a 10 días de trabajo en desarrollo, también se realizaron las ceremonias de Reunión de Revisión Diaria SCRUM (Review Daily Scrum Meeting), para el control y/o aprobación de las actividades definidas. Todos los eventos quedan registrados en la herramienta AIGA (Aplicación de gestión de proyectos de software bajo metodología AGIL).

Como resultado de este proyecto se ha completado el desarrollo de la arquitectura informática y sus distintos módulos: Capa de Aplicación, Capa de Bases de datos (Cassandra), esquema de conectividad y seguridad informática, lo que permite la operación del servicio de consultas al SOI DATA, con los modelos de software de alta eficiencia planteados. Se completó la implementación de la arquitectura de la aplicación para los siguientes módulos: Componente de Migración de Información, módulo de Consulta individual y Consulta Masiva para persona Natural y Jurídica, Administración de Usuarios, Auditoría. La arquitectura de despliegue asociada se ha completado y está en ejecución.

## Conclusiones

El presente proyecto ofrece un aporte importante para el sector financiero colombiano, ya que permite contar con información fiable de usuarios de la plataforma SOI que administra ACH Colombia, para la optimización de los procesos de referenciación de clientes a través de una plataforma informática que provee datos a través de un sistema de información seguro.

Uno de los logros alcanzados en la ejecución de este proyecto fue el completar la implementación de este servicio con los niveles de calidad y confiabilidad esperados en corto tiempo, a través de la aplicación de las metodologías de desarrollo SCRUM. Se comprobó que esta metodología es adecuada para este tipo de proyectos de desarrollo de software por lo que será conveniente su utilización en futuros proyectos.

Como siguientes etapas de desarrollo de este proyecto se espera contar con acceso a información integral de pagos de seguridad social de los trabajadores formales en Colombia para robustecer esta plataforma de información.

También se espera continuar con la ejecución de proyectos que complementen el uso y análisis de estos datos para distintos fines, como análisis macro y microeconómicos que impacten la toma de decisiones de distintas entidades, basado en un mayor conocimiento del perfil laboral de la población colombiana.

## Agradecimientos

Los autores agradecen a Ministerio de Ciencia y Tecnología (Minciencias) por la calificación de este proyecto como de innovación en la convocatoria de beneficios tributarios, el cual permitió la ejecución del mismo y los impactos generados en el sistema financiero colombiano.

## Referencias

- [1]. Sundmaeker, H. Guillemain, P. Friess y Woelfflé, S. Vision and Challenges for

- Realising the Internet of Things, Bruselas: European Commission, 2010.
- [2]. Mironela, P. «The Importance of Web Services Using the RPC and REST Architecture,» International Conference on Computer Technology and Development, pp. 377-379, 2009.
- [3]. Cerami, E. Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL, USA: O'Reilly Media, Inc., 2002.
- [4]. Castillo, P. Bernier, J. Arenas, M. Merelo, J. y García-Sánchez, G. «SOAP vs REST: Comparing a master-slave GA implementation,» First International Workshop of Distributed Evolutionary computation in Informal Environments, pp. 1-6, 2011.
- [5]. Tarreau, W. «MAKING APPLICATIONS SCALABLE WITH LOAD BALANCING,» September 2006. [En línea]. Available: [https://www.haproxy.com/static/media/uploads/eng/resources/art-2006-making\\_applications\\_scalable\\_with\\_lb.pdf](https://www.haproxy.com/static/media/uploads/eng/resources/art-2006-making_applications_scalable_with_lb.pdf). [Último acceso: 29 June 2016].
- [6]. Tutorials Point, Scrum - Agile framework for completing complex projects, Hyderabad: Tutorials Point , 2015.
- [7]. Ministerio de Protección Social, «Afiliación, pago y recaudo de aportes al Sistema general de seguridad social en salud,» Bogotá, 2008.
- [8]. Banco de la República, «Reporte de Sistemas de Pago,» Banco de la Republica, Bogotá, 2017.
- [9]. Troester, M. «Big Data meets Big Data Analytics: Three key technologies for extracting real-time business value from the Big Data that threatens to overwhelm traditional computing architectures,» SAS Institute Inc, 2012.
- [10]. Ramirez, J. «Sistemas de información de mercados en el sector de banca y finanzas de Bogotá, como herramienta dinámica de investigación,» Universidad Nacional de Colombia, Bogotá, 2015
- [11]. Mahmood, Z. «Service Oriented Architecture: A New Paradigm for Enterprise Application Integration. ,,» de 11th WSEAS International Conference on COMPUTERS, 2007.
- [12]. Gottschalk, K., Graham, S., Kreger, H., & Snell, J. (2002). Introduction to web services architecture. IBM systems Journal, 41(2), 170-177.
- [13]. Baltopoulos, L. «Introduction to Web Services,» CERN School of Computing (iCSC), Ginebra, Suiza., 2005.
- [14]. Pautasso, C. «WS-\* vs. RESTful Services,» de European Conference on Web Service, 2010.

- [15]. Garg, N. Apache Kafka, Birmingham, UK.: Packt Publishing Ltd, 2013.
- [16]. Hewitt, E. Cassandra: The Definitive Guide, Estados Unidos: O'Reilly Media, Inc., 2010.
- [17]. Schwaber, K. Agile project management with Scrum, Microsoft Press, 2004.
- [18]. CCIT - Fedesarrollo, «Coyuntura TIC - El uso de los medios de pago electrónicos en Colombia,» Cámara Colombiana de Informática y Telecomunicaciones, Bogotá, 2015.
- [19]. MINTIC, «Boletín trimestral de las TIC - Cifras Primer Trimestre de 2017,» Ministerio de Tecnologías de la Información y las Comunicaciones, Bogotá, 2017. Representations for Identifying Normal