Servicio de acceso seguro a servicios de e-salud y/o sistemas de analítica.

Entregable **2.1**
Proyecto **41756**

Responsable técnico:
Dr. José Luis González Compeán
Profesor-Investigador, Cinvestav Tamaulipas

Muyal-Ilal:
Plataforma tecnológica para la gestión, aseguramiento, intercambio y preservación de grandes volúmenes de datos en salud y construcción de un repositorio nacional

Autorizaciones

Fecha: 12/11/2021	*Publicación: 15/11/2021*	*Versión: 2.0*
Seguimiento	**Nombre completo**	**Fecha**
Elaboró:	M. C. Diana Elizabeth Carrizales Espinoza	05/11/2021
Revisó:	Dr. Miguel Morales Sandoval	11/11/2021
Autorizó:	Dr. José Luis González Compeán	15/11/2021

Control de cambios

Versión	Fecha	Bitácora
1.0	05/11/2021	Documento Inicial
2.0	12/11/2021	Documento Final

Resumen

En los últimos años, la producción de datos ha crecido de forma exponencial debido a la producción continua de datos por fuentes como dispositivos de IoT (p. ej., electrocardiogramas, máquinas de rayos x y espirómetros) y dispositivos de usuarios finales (p. ej., celulares, tabletas, laptops y estaciones de trabajo).

En este contexto, los métodos para acceder y gestionar los datos han cambiado para hacer frente a este crecimiento. Actualmente, los datos no se almacenan en un solo lugar, sino que son almacenados en diferentes ubicaciones durante su ciclo de vida. Lo anterior da como resultado una gestión de datos jerárquica para producir una respuesta rápida al analizar un gran volumen de datos en escenarios actuales de manejo de grandes volúmenes de datos (big data). En este sentido, la creación de sistemas para el acceso, preparación, manejo y entrega de datos en entornos de múltiples infraestructuras o modelos de cómputo actuales y futuros (p. ej., borde, niebla, nube o cualquier combinación de ellos) es trascendental para asegurar el intercambio seguro y confiable de datos sensibles (p. ej., los del ámbito médico, tales como expedientes clínicos, tomografías, resonancias magnéticas, etc.).

En este documento se describe ***Chimalli****, una herramienta computacional conformada por un conjunto de servicios que permiten a las instituciones de salud, profesionales de salud, pacientes y/o comunidad científica acceder a los servicios de e-salud y/o sistemas de analítica para asegurar el manejo, preparación y acceso seguro y confiable de datos médicos.* ***Chimalli*** *verifica que cada sistema de e-salud observe, en forma automática y transparente, las normas nacionales e internacionales garantizando privacidad, confidencialidad, integridad y disponibilidad de los contenidos, así como estableciendo tolerancia a fallas de servicios/servidores y creando registros inmutables en una red privada (blcokchain) para dar trazabilidad al manejo de los datos. Además, crea automáticamente redes de cripto-contendores y blockchain para sistemas de e-Salud construidos con Nez .*

Índice

Índice de figuras

Índice de tablas

1. Introducción

La producción de dispositivos de IoT ha observado un aumento exponencial en los últimos años [1]. Por lo tanto, el volumen de datos producidos y administrados por las organizaciones ha aumentado [2] debido a que los usuarios finales asociados con las organizaciones producen, almacenan y usan datos de manera constante y continua, lo que produce un efecto de acumulación de datos [3].

En escenarios reales, esta tendencia da como resultado un entorno de procesamiento de big IoT data donde grandes repositorios de datos son producidos continuamente por dispositivos IoT (volumen), y posteriormente son procesados por múltiples procedimientos (variedad) para obtener información útil utilizada como entrada (valor y veracidad) en procesos críticos de toma de decisiones (velocidad) [4], [5]. Tradicionalmente, el cómputo en la nube [6] ha sido un soporte para escenarios de big data [7], [8] y la solución más popular para almacenar y procesar datos de dispositivos IoT [9]. Sin embargo, a medida que los sistemas se escalan y la cantidad de datos aumenta en escenarios de big data, una recopilación y procesamiento de datos centralizados se vuelve inviable.

En entornos organizacionales, como en hospitales, los datos deben procesarse, conservarse y compartirse con otras organizaciones de manera rentable. Además, las organizaciones deben cumplir con diferentes requisitos obligatorios no funcionales impuestos por las leyes, protocolos y normas de cada país. Estos requisitos incluyen la seguridad de los datos durante su transporte y conservación, estableciendo controles sobre el acceso de los datos (privacidad) y recursos (control de acceso), la integridad y confidencialidad de los datos, así como su confiabilidad.

En este documento se presenta el servicio Chimalli, el cual está compuesto por un conjunto de servicios para el acceso, manejo, preparación y entrega de datos médicos. Dichos servicios permiten a las instituciones de salud, profesionales de la salud, pacientes y/o comunidad científica acceder a servicios de e-salud y/o sistemas de analítica para obtener información útil que ayude a mejorar la toma de decisiones en escenarios de salud.

Las principales características que provee Chimalli a los datos son:

- Confiabilidad
- Eficiencia
- Integridad
- Confidencialidad
- Seguridad

2. Servicio de acceso seguro a servicios de e-salud y/o sistemas de analítica[1,2,3]

Chimalli es un servicio que permite a las instituciones de salud, profesionales de la salud, pacientes y/o comunidad científica acceder de forma segura a los servicios de e-salud y/o sistemas de analítica. Para ello, cuenta con esquemas de preparación/ recuperación de datos, y mecanismos de control de acceso.

Chimalli garantiza que los datos y los tomadores de decisiones sean aptos para realizar procesos de análisis. Además, permite validar y registrar cualquier operación de compartición de datos que se realice dentro del sistema de e-salud.
Chimalli permite alcanzar un 70% de las regulaciones estandarizadas en forma internacional para el manejo seguro de datos sensibles, y cubre todas las fases de interconexión establecidas por las normas oficiales.

El resumen de seguridad creado por Chimalli también ha permitido revelar las tareas de ciberseguridad que dependen de actividades realizadas por personal de salud para que las instituciones creen un plan para implementarlas.

Chimalli asegura el anonimato de los datos, así como la confidencialidad mediante el cifrado de los datos entrantes y salientes de los sistemas de e-Salud. Además, permite detectar alteraciones en los datos. También permite la gestión automática de contratos inteligentes, la gestión automática de transacciones y la verificabilidad de transacciones de forma confidencial.

Chimalli incluye los siguientes productos:

1. Servicios de preparación y recuperación de datos médicos configurables que incluya los requerimientos de seguridad, trazabilidad, integridad y eficiencia.
2. Mecanismos de trazabilidad de datos basados en blockchain.
3. Mecanismos de control de acceso de usuarios.
4. Servicios de validación de normas oficiales mexicanas y protocolos DICOM/HL7.
5. Un servicio que permite la utilización de técnicas de criptografía de siguiente generación para la transformación de datos en objetos seguros.

[1] Para descargar el software de Chimalli, dar clic *aquí*.
[2] Para visitar la página web del proyecto, dar clic aquí.
[3] Para ver la infografía técnica de Chimalli, dar clic aquí. Para ver la infografía para público en general, dar clic aquí.

2.1. Servicios de preparación y recuperación de datos médicos configurables que proveen seguridad, trazabilidad, integridad y eficiencia a los datos

En los procesos de preparación de datos para escenarios reales de gestión de datos, diferentes requerimientos no funcionales (RNFs) (por ejemplo, seguridad, eficiencia, y confiabilidad) deben ser considerados debido a las normas de gestión de la salud (por ejemplo, las normas oficiales mexicanas NOM-024-SSA3-2010 y NOM-004-SSA3-2012) y las leyes impuestas por los gobiernos y organizaciones [10], [11].

En esta sección, se detalla el esquema de preparación de datos que añade propiedades no funcionales a los datos. La preparación de los datos se realiza antes del transporte de estos a través de los flujos de datos (cargados para su almacenamiento o transmitidos utilizando entornos no controlados como Internet y la nube [12]). Primero se describe la estructura de procesamiento de tuberías definida en Chimalli para crear los esquemas de preparación, y más tarde se describen los RNFs elegidos para ser añadidos a la tubería.

La estructura de los esquemas de preparación de datos se encuentra construido en forma de una tubería, la cual se modelo con base en un gráfico acíclico dirigido (DAG, por siglas en inglés de Directed Acyclic Graph). En el DAG, los nodos representan a los algoritmos de los RNFs, y las aristas representan la entrada requerida por los algoritmos y los resultados producidos por ellos. Por lo tanto, una tubería puede incluir tantos algoritmos de NFRs como sea necesario para cumplir con las normas y leyes internacionales para el almacenamiento e intercambio de datos sensibles. De este modo, la ejecución secuencial de los RNFs crea una tubería de procesamiento.

Los algoritmos para cumplir con los RNFs en los flujos de datos son computacionalmente costosos. Esto se debe a que los algoritmos añaden retrasos a cada etapa de la tubería definida por un DAG. Para mitigar este impacto en el rendimiento de la preparación, los esquemas consideran bifurcaciones para producir paralelismo de datos y procesamiento de tareas concurrentes en cada etapa de la tubería.

Las etapas invocan al gestor de paralelismo y crean clones de los algoritmos de RNFs para asignarles carga de trabajo, lo que convierte a estos nodos en trabajadores. Además, el gestor de paralelismo despliega un balanceador de carga para mejorar el rendimiento del procesamiento de datos/tareas. Este modelo de procesamiento produce un paralelismo implícito que permite la ejecución del algoritmo de los RNFs de forma paralela y/o concurrente. Esto reduce el tiempo necesario para preparar los datos antes de transportarlos a través de los flujos de datos.

2.1.1. Esquemas de preparación para la creación de flujos de datos costo-eficiencia, seguros y confiables.

Los siguientes RNFs se añaden a la gestión de datos en los flujos de datos:

- **La confiabilidad** se aplica para mitigar los problemas causados por las deficiencias de la infraestructura donde se procesan y almacenan los datos [13], [14]. Este requisito se consigue utilizando el conocido algoritmo IDA [15].
- **La seguridad** (CIA: *confidencialidad, integridad* y *control de acceso*), es añadida a los datos para resolver los problemas que surgen cuando se comparten datos a través de entornos no controlados y no confiables (por ejemplo, la nube [16], [17]).
- **Costo-Eficiencia**, es conseguida mediante técnicas de compresión y deduplicación para reducir el número de contenidos o la cantidad de datos que hay que preprocesar. Además, estas técnicas reducen los datos enviados a la nube y los costos de subcontratar servicios para realizar las tareas de gestión de datos.

La Figura 1 muestra un ejemplo del DAG de un esquema de preparación. Este DAG tiene cuatro etapas (compresión, deduplicación, codificación y cifrado) para añadir NFRs tales como confiabilidad, seguridad, integridad y costo-eficiencia.

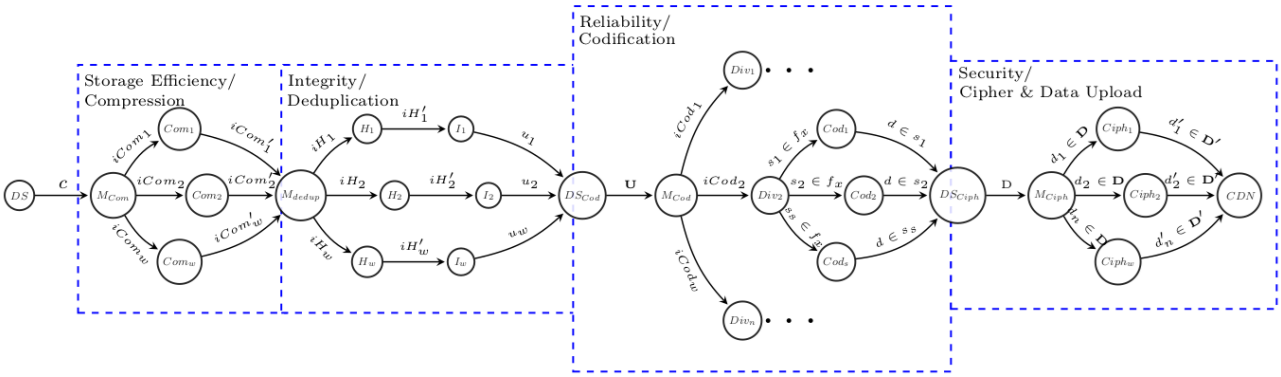


Figura 1. Esquemas de preparación de datos modelados como un DAG.

La primera etapa (compresión de datos) de esta tubería de preparación, añade costo-eficiencia al reducir el tamaño de cada contenido ($c_x \in C$) en la fuente de datos (DS). Además, aplica una técnica de deduplicación de datos la cual identifica archivos duplicados en la fuente de datos. Esta etapa añade integridad a los datos generando la huella digital (conocida en inglés como hash) de cada contenido (h(c)). El cálculo del hash se centra en lograr dos objetivos:

i) Identificar los contenidos replicados antes de enviarlos al esquema de preparación de datos.

ii) Detectar alteraciones cuando los usuarios descargan los archivos.

La etapa de codificación aplica el Algoritmo de Dispersión de Información (IDA, por siglas en inglés de Information Dispersal Algorithm) [28][25] para añadir tolerancia a fallos en escenarios de interrupción o indisponibilidad de datos o servidores/nodos. En esta etapa, el gestor divide cada contenido/archivo entrante en n segmentos, que contienen redundancia suficiente para recuperar el archivo original utilizando cualquier m de los n segmentos (donde m < n).

En la última etapa (cifrado), los datos se cifran utilizando técnicas basadas en AES [18] para la confidencialidad y CP-ABE [19] para aplicar control de acceso. Esta etapa crea un objeto cifrado que incluye controles de acceso. Los sincronizadores locales/federados se añaden como una etapa más de la tubería.

2.1.2. Eficiencia basada en patrones de paralelismo implícito

Como se puede observar en la Figura 1, diferentes patrones de paralelismo se pueden construir para aumentar la eficiencia de las tareas en los esquemas de preparación. Dichos patrones se pueden observar en la Figura 1 en forma de bifurcaciones, en los cuales se observan dos tipos diferentes de patrones: i) manejador/trabajador, y ii) divide&vencerás.

El patrón manejador/trabajador (M/T) procesa los datos en diferentes fases, como la clonación, la distribución de tareas y la supervisión de la ejecución de tareas. En la fase de clonación, el manejador crea instancias de contenedores virtuales (CV), que representan clones de una etapa determinada de la tubería de preparación.

Estos clones se denominan trabajadores. En la fase de distribución de tareas, el manejador lee el contenido almacenado en una dirección de origen de datos (OD) y crea una lista de tareas utilizando un conjunto de rutas ($P = \{Ruta_1, Ruta_2, \ldots, Ruta_n\}$). Cada ruta se distribuye de forma balanceada a los trabajadores utilizando el algoritmo two choices [20], que permite elegir al trabajador con menor carga de trabajo [21]. Véase un ejemplo de M/T (Mhash) en la etapa de deduplicación en la Figura 1.

En la fase de supervisión, el componente manejador verifica que los trabajadores entreguen los resultados de su tarea asignada a la siguiente etapa de los esquemas. Cada trabajador realiza las etapas de hashing e indexamiento sobre un conjunto de contenidos que le asigna el manejador. Los trabajadores calculan la huella digital (hash) de los contenidos, y posteriormente las huellas son indexadas en un servicio de metadatos. El servicio de indexación almacena los hashes únicos en una tabla para mantener la consistencia de los archivos previamente indexados, lo que permite añadir la propiedad de integridad a los datos.

El patrón divide&vencerás incluye entidades tales como divide, trabajadores y vences. Divide es una instancia de software que divide los datos en s segmentos (sin redundancia), que son procesados por s trabajadores (similares a los trabajadores del patrón M/T). Divide consolida los resultados de cada trabajador en un único resultado para entregarlo a la siguiente etapa de procesamiento (véase D&C (Divx) en la Figura 1).

Nota: Para más información acerca de los servicios de preparación y recuperación de datos médicos, ver el artículo **"FedFlow: A Federated Platform to Build Secure Sharing and Synchronization Services for Health Dataflows"** en el Anexo B.2.

2.2. Mecanismos de trazabilidad de datos basados en blockchain

El proceso de trazabilidad juega un papel importante dentro de los flujos trabajo debido a que brindan la posibilidad de identificar el origen y las distintas etapas por las que ha pasado un producto a lo largo de todo su ciclo de vida (proceso productivo, distribución y logística, hasta llegar a un consumidor final).

Este proceso cumple una parte fundamental dentro del proyecto debido a que permitirá a cualquiera de las entidades involucradas (personal médico y usuarios finales) acceder a la información de cada una de las etapas por las cuales ha pasado el contenido digital, verificando si este cumple con las acciones pactadas y que ha sido procesado por las entidades correctas.
Lo anterior posibilita aceptar o rechazar el expediente digital basado en la información del flujo del producto (traza) apoyando de esta manera la toma de decisiones y mejorando la confianza en el resultado obtenido.

El mecanismo de trazabilidad de Chimalli provee las características de trazabilidad y verificabilidad a cada uno de los productos que se procesan en cualquiera de los servicios construidos a través de la plataforma de e-Salud. Este mecanismo permite realizar trazabilidad tanto interna como externa (dentro de una misma institución, así como colaboraciones entre varias de ellas) de los productos digitales que son procesados y manejados a través de los sistemas de e-Salud.

Además, la tecnología de blockchain permite almacenar y transmitir información de forma transparente, distribuida y segura sin un órgano central de control. Esta tecnología utiliza una base de datos segura compartida por diferentes usuarios autorizados para que todos puedan comprobar la validez de los procesos realizados en el flujo de trabajo en cada uno de sus bloques. Debido a las características

previamente mencionadas, la blockchain tiene muchas ventajas para el sector de cadenas de suministros (enfoque utilizado en el actual proyecto de flujos de tareas en salud).

2.2.1. Diseño Conceptual y de interacción

El servicio de trazabilidad y verificabilidad de contenidos médicos tiene el objetivo de asegurar el registro inmutable de cada una de las acciones que se realicen sobre cada uno de los activos digitales que son procesados en las diferentes cadenas de valor generadas a través del servicio de construcción de servicios de e-Salud. La Figura 2 muestra la interacción entre ambos servicios.

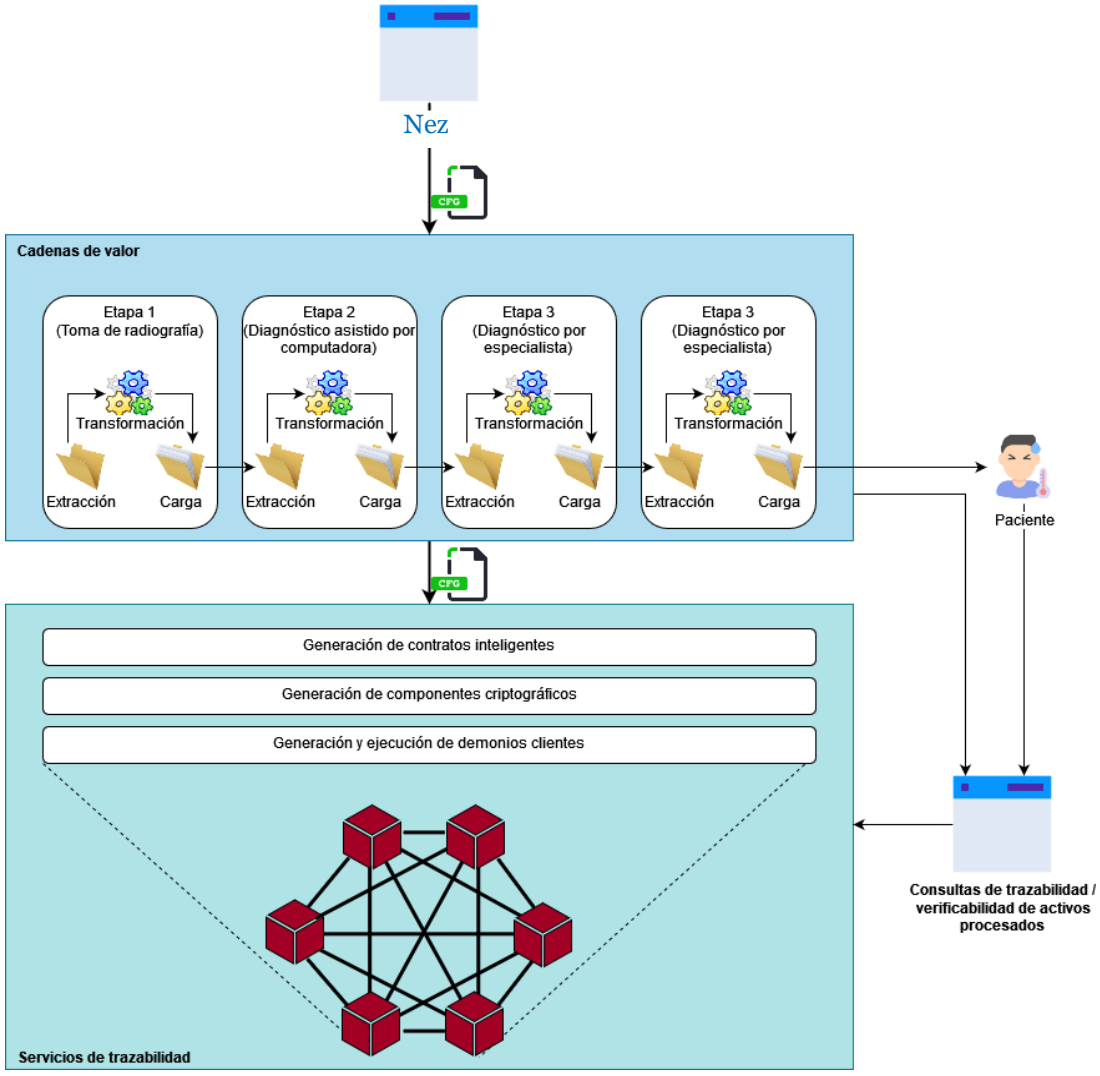


Figura 2. Diseño conceptual e interacción

El proceso comienza con el diseño de los sistemas de e-Salud en Nez, en donde se definen las etapas del sistema, así como su orden de ejecución (cabe destacar que la red de bloques encadenados es desplegada en el momento inicial del sistema de

construcción). A partir de ello el sistema genera un archivo de configuración que contiene la información necesaria para el despliegue y ejecución de los elementos requeridos para el uso de la red de verificabilidad.

A partir del archivo de configuración provisto por Nez, el sistema de trazabilidad crea automáticamente los elementos necesarios para el despliegue de la solución. Una vez creados los archivos necesarios para el despliegue, el sistema de trazabilidad ejecuta cada uno de los archivos generados, y comienza con la generación del contrato inteligente para cada una de las cadenas de valor.

Es importante destacar que, durante este proceso, también se crean los componentes criptográficos para que cada una de las etapas involucradas pueda realizar las transacciones correspondientes en cada uno de sus procesos.

Al final del proceso, se generan distintos *clientes* para cada una de las fases del sistema de e-Salud, es decir, tanto en la extracción como en la carga de información. De esta manera es posible mantener un registro de cada uno de los puntos por los cuales ha pasado cada uno de los activos procesados.

2.2.2. Elementos utilizados en el mecanismo de trazabilidad

El servicio de trazabilidad hace uso de las siguientes herramientas para su funcionamiento: Hyperledger Sawtooth, Docker, y Docker Compose. Además, Javascript es utilizado como lenguaje principal y MySQL como gestor de base de datos.

Hyperledger Sawtooth: es una solución empresarial para construir, implementar y ejecutar redes y aplicaciones de contabilidad distribuida (soluciones de bloques encadenados). Este proporciona una plataforma extremadamente modular y flexible para implementar actualizaciones basadas en transacciones al estado compartido entre entidades mediante algoritmos de consensos. Las características principales de *Hyperledger Sawtooth* son: la distribución, inmutabilidad, seguridad, separación entre los niveles de aplicación y el sistema principal, la ejecución de transacciones paralelas, y el consenso dinámico.

Docker y Docker compose: Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software. Proporciona una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Por otro lado, Docker Compose es una herramienta para definir y ejecutar aplicaciones Docker de varios contenedores, creando una red virtual que permite la comunicación entre ellos de forma eficiente y simple.

2.2.3. Componentes que integran el mecanismo de trazabilidad

Existen tres elementos que integran la solución de trazabilidad y verificabilidad, los cuales se listan a continuación: un generador de archivos YAML de clientes para la red, cada uno de los clientes y la red de bloques encadenados (ver Figura 3). El servicio recibe un archivo de configuración a través del cual despliega y ejecuta todos los elementos necesarios para el uso de este servicio. Cada uno de los componentes se lista a continuación.

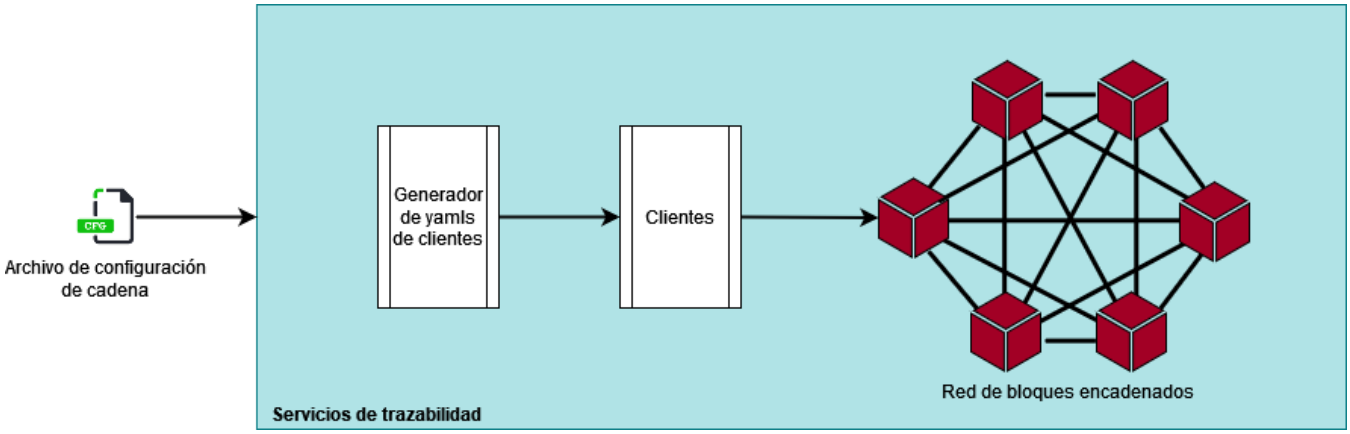


Figura 3. Componentes del servicio de trazabilidad.

Red de bloques encadenados: para el desarrollo y despliegue de la red Blockchain se hizo uso del framework Hyperledger Sawtooth. En la Figura 4 se observa cada uno de los componentes que integran Sawtooth, los cuales se listan a continuación:

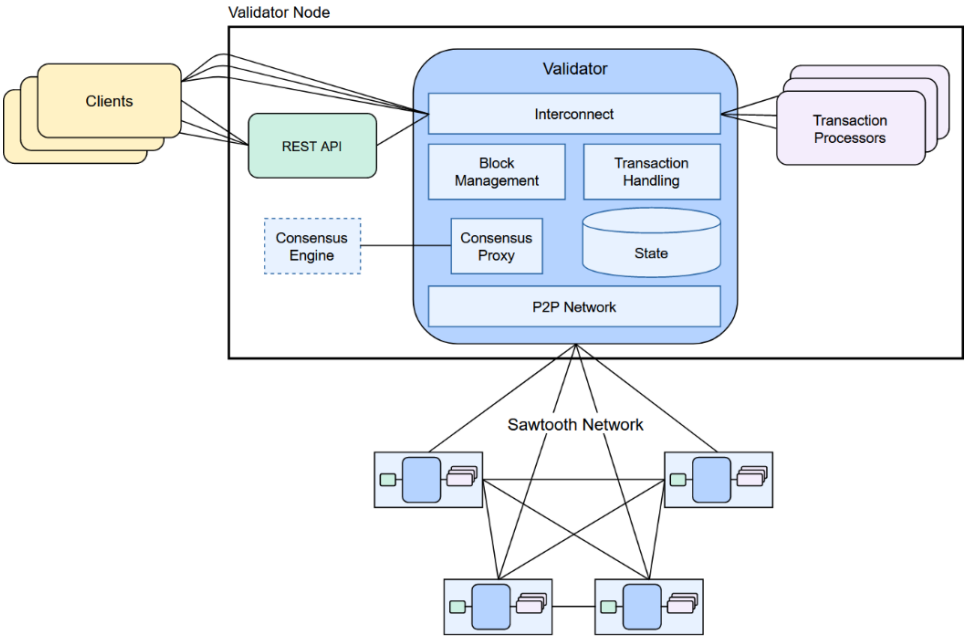


Figura 4. Arquitectura Sawtooth.

- **Cliente:** componente externo de la red cuyo objetivo es realizar consultas y registros del procesamiento del activo.
- **REST API:** Sawtooth provee una API para realizar las consultas y registros a la Blockchain de forma simple. Permite a los clientes interactuar con un validador utilizando peticiones HTTP/JSON comunes.
- **Validador:** componente responsable de validar lotes de transacciones, combinarlos en bloques, mantener el consenso con la red Sawtooth y coordinar la comunicación entre clientes, procesadores de transacciones y otros validadores en la red.
- **Procesador de transacciones:** valida las transacciones y actualiza el estado según las reglas definidas por la familia de transacciones asociada (contrato establecido).

Generador automático de elementos clientes: En la Figura 3 se muestra que el proceso en el servicio de trazabilidad comienza cuando un usuario generador de cadenas crea un nuevo flujo de procesamiento. En ese punto se genera un archivo de configuración que contiene la información de cada una de las etapas implicadas en el proceso, este archivo es enviado al servicio de trazabilidad para que se comience con la generación del material necesario para el uso de la red de cadenas de bloques.

En la Figura 5 se observa el diagrama de flujo del proceso que realiza el generador automático de elementos "clientes", el cual recibe como entrada el archivo de configuración e itera por cada una de las etapas que tiene descritas, y por cada una de ellas crea las carpetas que contienen los archivos de despliegue de los clientes, además crea el material criptográfico necesario para cada una de estas entidades.

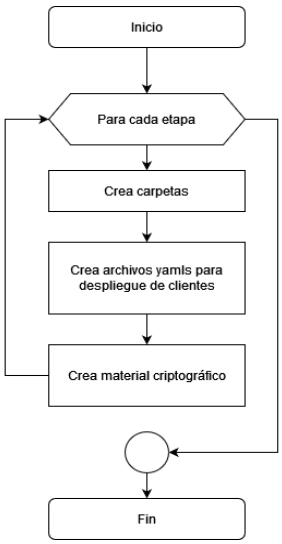


Figura 5. Diagrama de flujo del proceso realizado por el generador automático de elementos clientes.

Cliente

El cliente es la entidad capaz de consultar y registrar transacciones a la red de bloques encadenados. Dentro del modelo utilizado en la plataforma, las cadenas de valor contienen etapas, las cuales contienen una fuente de datos, un proceso de transformación y un espacio para colocar los productos resultantes. En este sentido, el servicio de trazabilidad crea un elemento cliente en cada una de las fuentes y espacios resultantes para el registro de cada proceso de transformación, arribo y envío de activos digitales.

Nota: Para más información acerca del mecanismo de trazabilidad y verificabilidad de contenidos médicos, ver el reporte técnico "**Trazabilidad y verificabilidad de contenidos médicos**" en el **Anexo A.1.**

2.3. Servicio para la transformación de datos en objetos seguros mediante el uso de técnicas de criptografía de siguiente generación

El cómputo en la nube se está convirtiendo en el nuevo núcleo de aplicaciones y servicios. Se espera que, en los próximos años, el 49 % de los datos se almacenen en la nube [22]. Al igual que el almacenamiento en la nube, los servicios de entrega de contenido se han convertido en piedras angulares para que las organizaciones, los usuarios finales y los trabajadores participen en cualquiera de los flujos de trabajo organizativos en línea, trabajo remoto parcial/total u oficina en casa [23].

Para evitar incidentes o mitigar riesgos que aún surgen en la nube, como alteraciones de datos [24], privacidad, violaciones de confidencialidad y accesos no autorizados, las organizaciones deben entregar o recuperar información a/de socios o usuarios finales de forma segura y transparente [25]. Para ello, *Chimalli* cuenta con un servicio de patrones paralelos que permite construir sistemas de seguridad eficientes para que las organizaciones compartan, intercambien y rastreen información confidencial en la nube.

En este servicio, los criptosistemas de seguridad y el software de blockchain se convierten en servicios en la nube independientes y autónomos llamados *cripto-contenedores*. Para mejorar la eficiencia de los servicios de seguridad y la experiencia del servicio del usuario final, se agrega un patrón paralelo implícito junto con el balanceo de carga a los cripto-contenedores. Este servicio permite a las organizaciones respaldar patrones de intercambio de información en línea entre múltiples participantes al acoplar conjuntos de cripto-contenedores para cumplir con múltiples combinaciones de requisitos de seguridad (por ejemplo, confidencialidad,

integridad, no repudio, autenticación y trazabilidad). Este servicio de *Chimalli* cuenta con dos características principales:

- ***Flexibilidad*** para integrar, sobre la marcha y bajo demanda, tantas aplicaciones de seguridad (criptosistemas) como inquietudes expresadas por las organizaciones (para las etapas), por los participantes de cada flujo de trabajo organizacional, en un único sistema de seguridad integral.

  En este método, los criptosistemas de seguridad y el software de blockchain se convierten en servicios en la nube independientes y autónomos llamados cripto-contenedores. Los conjuntos de cripto-contenedores se combinan para crear sistemas de seguridad en la nube para admitir flujos de trabajo organizacional en línea que incluyen a varios participantes. Un framework basado en este método, crea sistemas de seguridad en la nube que cumplen con múltiples combinaciones de requisitos de seguridad, como confidencialidad, integridad, no repudio, autenticación y trazabilidad.

- ***Eficiencia*** utilizando un modelo de programación paralela de gestión de datos basado en la combinación de patrones y esquemas de balanceo de carga, que están integrados en los cripto-contenedores. El modelo es utilizado para crear dos esquemas de patrones paralelos llamados "*Pipeline*" y "*Overlapped*".

  - El patrón *Pipeline* incluye dos tipos de patrones: un *pipe&filter* (tuberías y filtros) en combinación con el patrón manejador/*trabajador*. El primer patrón organiza los criptosistemas en forma de tuberías, mientras que el segundo despliega estas tuberías como trabajadores para ejecutarlos en paralelo. Este esquema fue diseñado para codificar/decodificar conjuntos de archivos/tareas pequeñas (por ejemplo, asegurar archivos pequeños con un tamaño de clave pequeño de 128 bits) en paralelo.

  - El patrón *Overlapped* acopla criptosistemas independientes para que se ejecuten de manera superpuesta, mientras que los criptosistemas que incluyen un tipo de dependencia se acoplan en forma de tubería. Todos los criptosistemas se ejecutan como una tubería, que también se gestionan como trabajadores (en un patrón de *manejador/trabajador*); como resultado, las tuberías Overlapped también se ejecutan en paralelo. Este esquema fue diseñado para codificar/decodificar conjuntos de grandes contenidos/tareas (por ejemplo, proteger archivos con un tamaño de clave grande de 192 y 256 bits) en paralelo.

Este servicio tiene dos componentes principales:

1. Un método de seguridad múltiple en la nube para crear servicios de gestión de seguridad de la información confidencial flexibles, integrales y eficientes.

2. Dos nuevos patrones paralelos eficientes e implícitos como ***PFP*** y ***Overlapped*** para mejorar significativamente el rendimiento de los sistemas de seguridad en la nube, así como la experiencia de servicio de los usuarios finales.

2.3.1. Método para la creación de servicios de intercambio seguro y eficiente de información en la nube

Para crear servicios de intercambio de información en la nube eficientes y seguros se cuenta con una metodología de cuatro fases (ver Figura 6). Estos servicios permiten que múltiples usuarios asociados a diferentes organizaciones intercambien datos confidenciales entre sí en forma de flujos de trabajo organizacional.

La *primera fase* comprende la definición de los participantes autorizados para acceder a los datos, así como la secuencia válida de interacciones esperadas al compartir e intercambiar datos. Como resultado de esta fase, se crea un esquema de gestión de la información que describe un flujo de trabajo organizacional. Un ejemplo de tal esquema se muestra en la Figura 6.

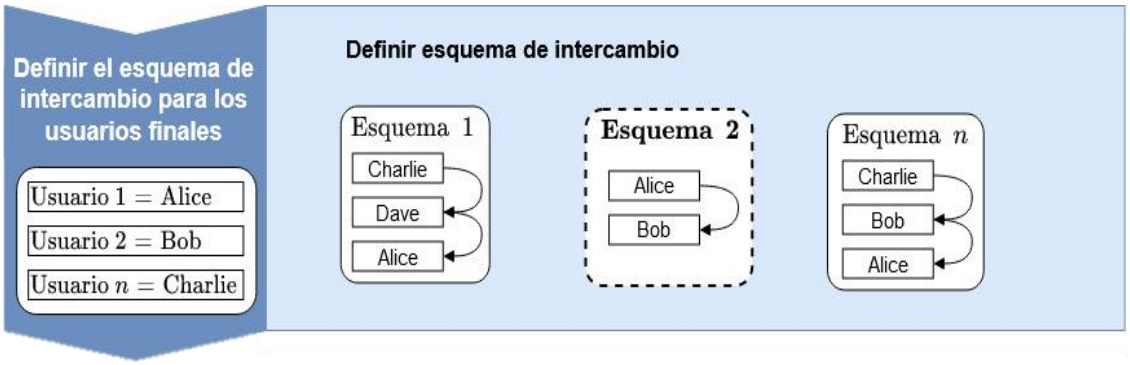


Figura 6. Fase 1 para la creación de servicios de intercambio de información eficientes y seguros.

En la segunda fase, los diseñadores pueden elegir bloques de seguridad para un cripto-contenedor entre los bloques disponibles en el repositorio de servicios de seguridad. Los bloques elegidos están organizados en forma de patrón.
En este sentido, existen dos tipos de patrones disponibles: i) **Pipelines paralelos** (vea **PFP** en la Figura 7) o **Pipelines Overlapped** (vea **Overlapped** en la Figura 7). Cuando los usuarios finales lanzan un cripto-contenedor, los criptosistemas elegidos por ellos siguen el patrón asignado al cripto-contenedor.

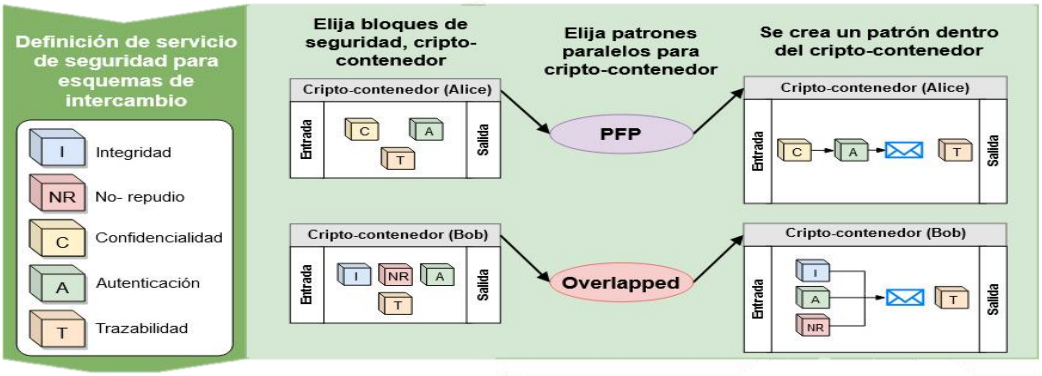


Figura 7. Fase 2 para la creación de servicios de intercambio de información eficientes y seguros.

En la *tercera fase* (ver Figura 8), los parámetros de nivel de seguridad y eficiencia son agregados al cripto-contenedor creado en la fase anterior. El parámetro de seguridad indica la resistencia de la seguridad (por ejemplo, la longitud de la clave criptográfica). El parámetro de eficiencia viene dado por el número de trabajadores utilizados por el patrón de ese cripto-contenedor. Este parámetro determina la cantidad de tuberías que se ejecutarán de manera concurrente.

Para crear un flujo de trabajo, se crea un cripto-contenedor para cada etapa de ese flujo (uno por cada participante considerado en los esquemas definidos en la *fase 1*). Esto significa que las tres fases descritas anteriormente se repiten hasta crear tantos cripto-contenedores como etapas consideradas en un flujo de trabajo.

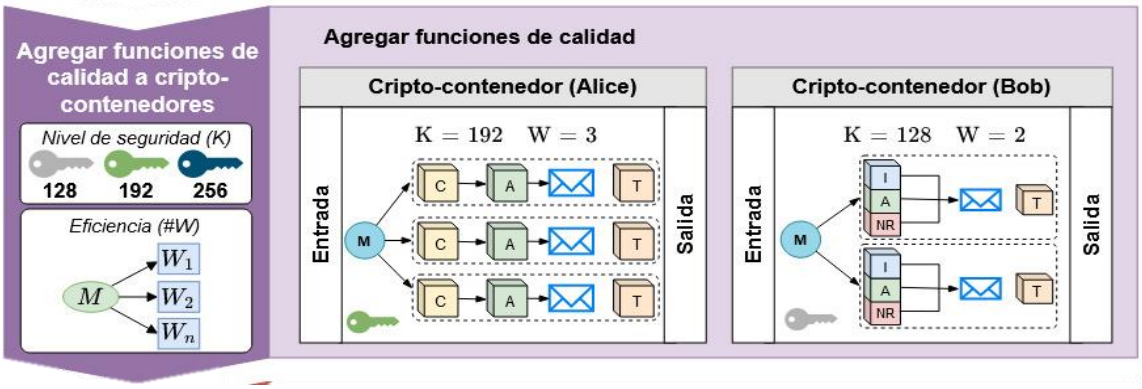


Figura 8. Fase 3 para la creación de servicios de intercambio de información eficientes y seguros.

En la *última fase* (Figura 9), los cripto-contenedores se encadenan para crear un servicio de información seguro, que establecerá controles sobre el intercambio de datos a través del flujo de trabajo resultante definido por cada esquema.

Esto significa que los participantes en los flujos de trabajo de intercambio de información confidencial no están obligados a hacer cumplir las políticas de seguridad (encriptar los datos antes de enviarlos a otro participante), ya que esta misma tarea la realizan los cripto-contenedores de manera automática y transparente. Lo anterior es clave para que las organizaciones enfrenten la paradoja de la información de

privacidad [26], [27] donde los usuarios finales no aplican la seguridad a sus operaciones de intercambio, incluso cuando saben que esta es una tarea crítica antes de compartir datos confidenciales.

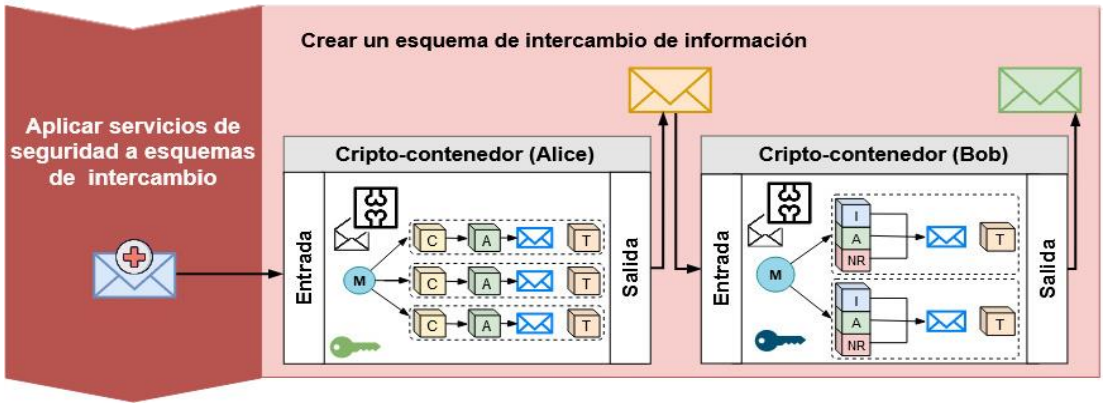


Figura 9. Fase 4 para la creación de servicios de intercambio de información eficientes y seguros.

La Figura 6 muestra un ejemplo en donde se construyen tres esquemas de administración diferentes (casos de uso) sin alterar la codificación de las aplicaciones de seguridad ni las aplicaciones de los usuarios finales. Además, los parámetros de los cripto-contenedores y los esquemas se pueden modificar en función de las necesidades de seguridad de la información de los usuarios finales.

2.3.2. Repositorio de bloques de seguridad para cripto-contenedores

Se creo un repositorio de bloques de seguridad, el cual es un conjunto de bloques de seguridad disponibles para que los usuarios finales incluyan estos bloques en sus cripto-contenedores (ver Tabla 1).

Este repositorio incluye bloques de seguridad como criptosistemas simétricos (Estándar de cifrado avanzado, AES [28]), cifrado basado en emparejamiento (firmas cortas) y cifrado basado en atributos (CP-ABE) [29], así como un bloque de trazabilidad, que se basa en un nuevo algoritmo propuesto en este servicio llamado TraceChain, el cual se utiliza para los cripto-contenedores registrando cada transacción de intercambio de información en una cadena de bloques privada [30].

AES se utiliza para el cifrado de datos masivos. Mientras que CP-ABE [31] se utiliza para hacer cumplir criptográficamente el control de acceso. Tanto *CP-ABE* como las firmas cortas (aquí referidas como *SSign*) utilizan un emparejamiento bilineal (asimétrico) computable eficiente e: $\boldsymbol{G_1} \times \boldsymbol{G_2} \rightarrow \boldsymbol{G_T}$, con $\boldsymbol{G_1} - \langle g_1 \rangle$, $\boldsymbol{G_2} - \langle g_2 \rangle$ y $\boldsymbol{G_T}$, los cuales son grupos cíclicos de orden r. En la práctica, $\boldsymbol{G_1}, \boldsymbol{G_2}$ son subgrupos de una curva elíptica definida sobre un campo finito F_q , y $\boldsymbol{G_T}$ es el grupo multiplicativo del campo de

extensión $\boldsymbol{F}_{qk}$, con k referido como el grado de incrustación de la curva elíptica, que es el número entero positivo más pequeño tal que r divide a $q^k - 1$. Los procedimientos de firma y verificación en *SSign* se implementan utilizando la instancia de función hash SHA.

La combinación de estos algoritmos y funciones da como resultado la adición de diferentes propiedades de seguridad a la información mediante un cripto-contenedor como se describe en la Tabla 1.

TraceChain es un algoritmo creado para agregar tanto trazabilidad como verificabilidad a los cripto-contenedores. Este algoritmo incluye un conjunto de funciones (listadas en la Tabla 1) para crear un modelo de negocio para una cadena, creando contratos inteligentes por cada par de cripto-contenedores, así como para registrar las transacciones realizadas por cada cripto-contenedor.

Este algoritmo permite a los usuarios finales rastrear contenidos administrados por cripto-contenedores y realizar desafíos para verificar un contenido dado que ha sido compartido o intercambiado a través de una cadena de cripto-contenedores.

Tabla 1. Repositorio de bloques basados en seguridad y rastreo de criptosistemas y algoritmos.

Criptosistema	Algoritmo	Descripción	Servicio de seguridad
AES	$k \leftarrow KeyGen(\lambda)$ $CT \leftarrow Encrypt(k, D)$ $D \leftarrow Decrypt(k, CT)$	Genera (internamente) una clave de cifrado k con una resistencia que cumple con λ. Cifra D usando la clave k. Descifra el texto cifrado CT de AES.	Confidencialidad
CP-ABE	$Setup(\lambda)$ $CT \leftarrow Encrypt(PK, M, A)$ $SK_u \leftarrow KeyGen(MK, S_u)$ $M \leftarrow Decrypt(PK, CT, SK_u)$	Crea la clave pública PK y la clave privada maestra MK, con una resistencia que cumple con λ. Cifra M dada la estructura de acceso A (política) Crea una clave privada para el usuario u dado el conjunto de atributos S_u. El usuario u descifra el texto cifrado CP-ABE CT usando la clave privada asociada SK_u.	Control de acceso

SSign	$\{SPK_u, SSK_u\} \leftarrow KeyGen(\lambda)$ $\sigma_u \leftarrow Sign(C, SSK_u)$ $Verify(SPK_u, C, \sigma_u)$ $\rightarrow [true/false]$	Crea el par de claves $\{SPK_u, SSK_u\}$ para el usuario u, con una resistencia que cumple con λ. El usuario u firma con SSK_u el hash del contenido C, creando así la firma $\sigma_u(H[C])$. El usuario u verifica la firma σ_u sobre el contenido C.	Integridad Autenticación No-repudio
TraceChain	$BCModel(VChain)$ $BCModelSC(SB_A, SB_B)$ $TransPair(SC, C_{in})$	Crea un modelo de negocio para una cadena de valor $(VChain_i)$. Crea un contrato inteligente SC por cada par de cripto-contenedores $SC - (SB_A \cup SB_B) \in VChain_i$ Registra el hash H_{in} del contenido entrante C_{in}y el H_{out} del sobre digital SDE obtenido de C_{in}.	Trazabilidad Verificabilidad

2.3.3. Patrones paralelos implícitos para cripto-contenedores

La creación de un patrón no es una tarea trivial, ya que las aplicaciones de seguridad se definen comúnmente mediante una codificación estricta que reduce la flexibilidad del cambio dinámico y sobre la marcha de los parámetros de seguridad. En la práctica, las aplicaciones de seguridad están disponibles principalmente en forma de conjuntos de funciones, o binarios invocados por un lanzador de aplicaciones [29].

En este método las aplicaciones de seguridad son administradas como bloques independientes, que pueden acoplarse entre sí (tuberías) para crear un único servicio (cripto-contenedor). Por lo tanto, los diseñadores del sistema pueden crear *patrones* que definen la disposición de los bloques de seguridad dentro de un cripto-contenedor para crear una ejecución paralela y/o concurrente de los bloques de seguridad (*SB, por sus siglas en ingles "Security Block"*).

2.3.3.1. PFP: patrón paralelo implícito Pipe&Filter

PFP es una disposición de una tubería paralela basada en el patrón Pipe&Filter (*P&F*) que ejecuta bloques de seguridad (aplicaciones) de manera secuencial para crear una ejecución concurrente de bloques. En este patrón, surge una saturación

cuando cada algoritmo de seguridad debe esperar los resultados del anterior (especialmente los últimos), lo que produce cuellos de botella y tiempos de espera para algunas etapas, así como periodos de inactividad para otras.

Para evitar este tipo de problema, la tubería *P&F* se clona para crear *trabajadores* ($worker\{w_1, w_2, \dots, w_n\}$) y un *manejador* (*M*), que se agregan a este patrón para crear una gestión paralela implícita de trabajadores, que crea un patrón de tubería paralela basado en la estrategia ***manejador/trabajador*** (*Patrón MW*).

La Figura 10 muestra una representación conceptual de este patrón. En esta representación, se muestran los filtros que son bloques de seguridad (SB) como AES, CP-ABE y Ssign. Esta tubería de seguridad ejecuta los tres bloques en una secuencia definida por la dirección de las tuberías (utilizando la memoria disponible en el cripto-contenedor que está ejecutando estos bloques). En esta tubería, el primer y el segundo *SB* están procesando los contenidos *C1 y C2* cuando el tercer SB está procesando el contenido *C3*; como resultado, los tres contenidos se procesan de manera simultánea y ningún bloque de seguridad está inactivo.

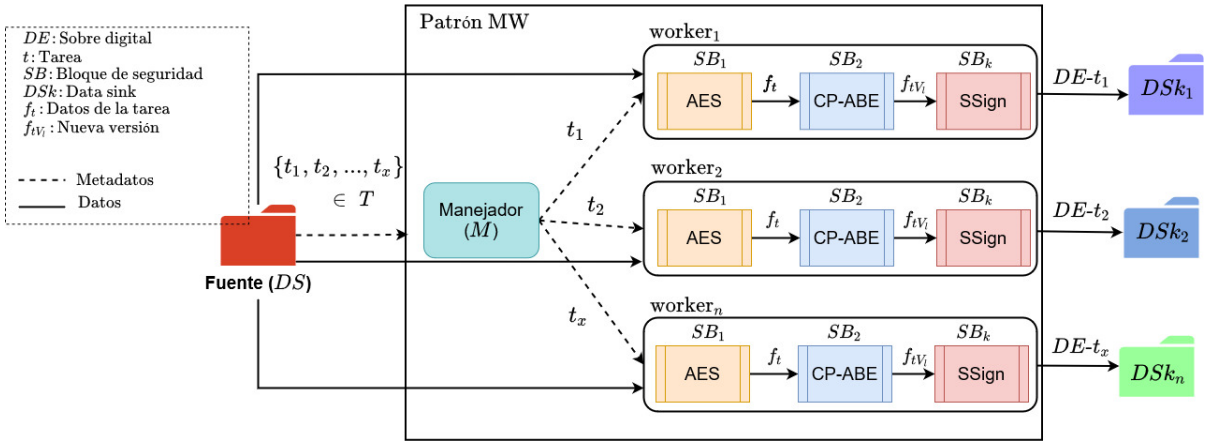


Figura 10. Ejemplo de un cripto-contenedor que incluye PFP.

PFP está conformado por una fuente de datos (DS) que recibe un conjunto de tareas ($\{t_1, t_2, \dots, t_x\} \in T$). Las tareas son administradas y procesadas por el patrón **manejador/trabajador** (Patrón MW). Dentro de él está el patrón P&F que recibe una tarea (t_x), y los datos salientes producidos ($DE - t_x$) por las aplicaciones (SB) son enviado a un almacén de datos(DSk).
La implementación principal recae en un patrón *P&F* utilizado para encadenar aplicaciones (Filtros: *SB*) de manera adyacente a través de rutas de E/S (Tuberías) donde el número de bloques *SB* debe ser mayor que 1 ($k \geq 1$). Por consiguiente, esta implementación es clonada por el *Patrón MW*, que recibe el bloque y la cantidad de clones requeridos por el cripto-contenedor. De igual forma, el ***manejador*** (*M*) es el

encargado de crear, coordinar y establecer el control sobre un número específico de trabajadores (*tuberías*).

En este patrón se utiliza una estrategia de balanceo de carga entre los trabajadores actuales, para después entregar esta carga de trabajo al trabajador indicado. Además, se incluyó un mecanismo de balanceo de carga implícito (*LB*) en el bloque *manejador* para distribuir la carga de trabajo a los trabajadores de manera justa [32]. Cada trabajador extrae, de una fuente de datos (*DS*), los archivos (t_x) asignados por el *manejador,* y ejecuta la tubería de seguridad para procesar los contenidos adquiridos; como resultado, cada trabajador procesa el contenido de forma independiente ($DE - t_x$).

Además de la distribución de una carga de trabajo balanceada, otra ventaja de este mecanismo es que se pueden agregar/eliminar nuevos trabajadores fácilmente sin realizar cambios importantes en las aplicaciones de seguridad ejecutadas en los cripto-contenedores.

Deben tenerse en cuenta algunas consideraciones ya que el rendimiento del cripto-contenedor podría disminuir al aumentar el número de trabajadores. Por lo tanto, un aspecto crucial es encontrar un número adecuado de trabajadores para una carga de trabajo determinada y de esta forma algunas condiciones de la tarea subyacente se aceleran por la configuración de este patrón.

2.3.3.2. Overlapped: un patrón paralelo para administrar tareas de seguridad "pesadas"

En el patrón ***Overlapped***, se modificó la disposición de los bloques de seguridad dentro de un cripto-contenedor para crear una ejecución superpuesta de las aplicaciones de seguridad. El primer paso es ejecutar simultáneamente bloques de seguridad que no tienen dependencias y crear un gráfico acíclico dirigido (*DAG*) y otro para bloques de seguridad con dependencias.

La Figura 11 muestra un ejemplo de bloques de seguridad organizados como un patrón superpuesto (**Overlapped**) en un cripto-contenedor, y cómo el patrón se clona n veces para crear trabajadores ($worker\{w_1, w_2, \dots, w_n\}$). Como resultado, todos los clones aseguran el contenido de manera paralela en un solo cripto-contenedor.

Más específicamente, la Figura 11 muestra un ejemplo de un patrón donde los bloques de seguridad (SB), como la generación de claves *AES* (AES-KG), y las firmas digitales (*SSing* sobre t_x), se ejecutan de manera concurrente, mientras que los servicios *AES* y *CP-ABE* (que tienen una dependencia funcional con *AES*-KG) se ejecutan cuando se

resuelve esa dependencia (una vez que *AES-KG* produce la clave [key] requerido por ambos algoritmos). En este punto, se pueden ejecutar otros servicios después de producir la firma del contenido entrante t_x (por ejemplo, *BCTracing* para registrar la transformación de t_x en una cadena de bloques). La Figura 11 también muestra cómo el *manejador* (M) del patrón *overlapped* se encarga de balancear la carga distribuida a los clones de cripto-contenedores.

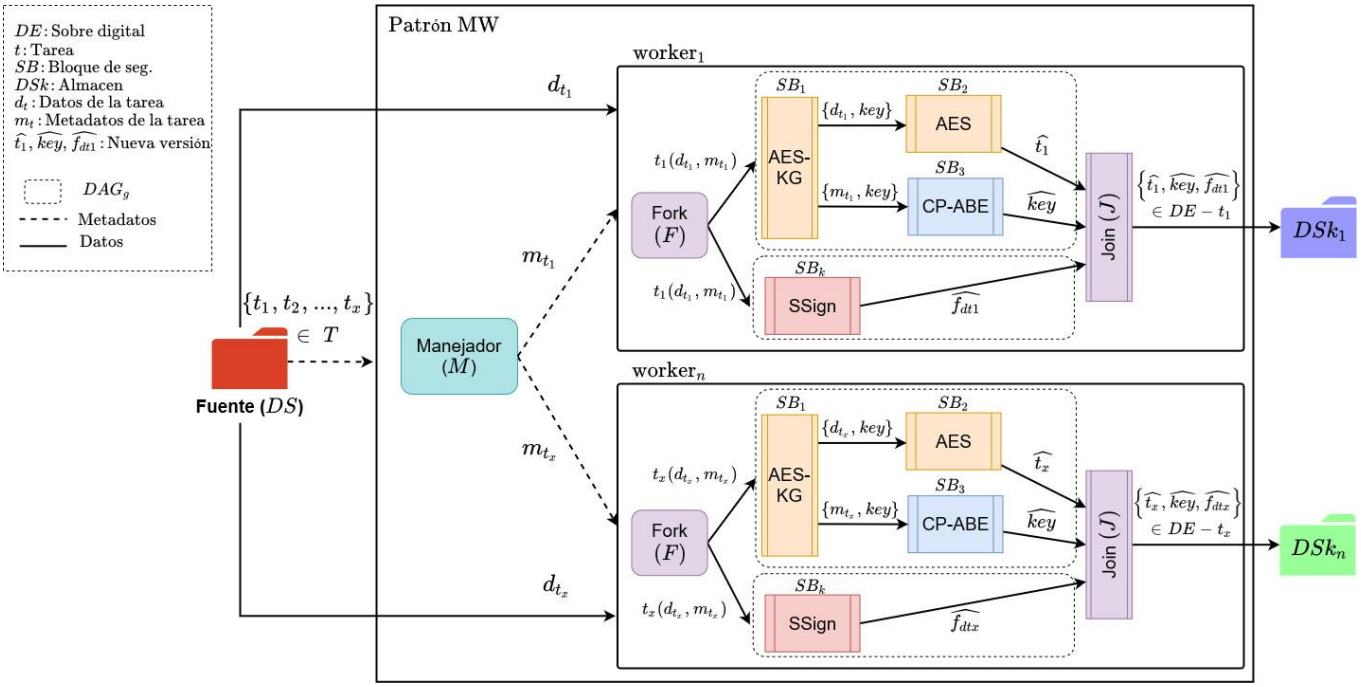


Figura 11. Ejemplo de un cripto-contenedor que implementa un patrón Overlapped.

El patrón *overlapped* produce dos tipos de paralelismo dentro de un cripto-contenedor. El primero se genera cuando los bloques de seguridad se ejecutan en paralelo dentro de un clon de patrón (dentro del $worker_n$). El segundo es producido por la porción de carga asignada a los trabajadores (clones del patrón superpuesto original), lo que produce un procesamiento y manejo concurrente de contenidos por los clones del patrón (ejecutados por el *Patrón MW*).

Este tipo de patrón es bastante adecuado para garantizar contenidos sensibles muy grandes (por ejemplo, imágenes médicas/satelitales, repositorios de Big Data, etc.) considerados con altos niveles de seguridad en las operaciones de compartición, intercambio (entrega y recuperación) y rastreo requeridas en operaciones comerciales y procesos de toma de decisiones.

Nota: Para más información acerca del Método para la creación de servicios de intercambio seguro y eficiente de información en la nube, ver el artículo **"Hermes, a parallel pattern method for secure and efficient cloud information sharing"** en el **Anexo B.1.**

2.4. Mecanismos de control de acceso de usuarios

Para poder cumplimentar con el requisito de seguridad se creó un repositorio de bloques de seguridad. Dicho repositorio cuenta con un conjunto de bloques de seguridad disponibles para que los usuarios finales puedan incluirlos en sus cripto-contenedores.

Este repositorio incluye bloques de seguridad como criptosistemas simétricos (estándar de cifrado avanzado, AES [28]), cifrado basado en emparejamiento (firmas cortas) y cifrado basado en atributos (CP-ABE) [29], así como un bloque de trazabilidad, que se utiliza para registrar cada transacción de intercambio de información en una cadena de bloques privada [30].

En este contexto, CP-ABE [31] se utiliza para hacer cumplir criptográficamente el *control de acceso*. Los criptosistemas basados en emparejamiento que producen firmas cortas se utilizan para servicios de autenticación, no repudio e integridad. Estos criptosistemas pueden proporcionar cualquiera de los niveles de seguridad equivalentes a 128, 192 o 256 bits, que cumplen con la mayoría de los estándares (por ejemplo, NIST [33], [34]).

Tanto CP-ABE como las firmas cortas utilizan un emparejamiento bilineal (asimétrico) computable eficiente: e: $\boldsymbol{G}_1 \times \boldsymbol{G}_2 \rightarrow \boldsymbol{G}_T$, con $\boldsymbol{G}_1 - \langle g_1 \rangle$, $\boldsymbol{G}_2 - \langle g_2 \rangle$ y $\boldsymbol{G}_T$, los cuales son grupos cíclicos de orden r. En la práctica, $\boldsymbol{G}_1$, $\boldsymbol{G}_2$ son subgrupos de una curva elíptica definida sobre un campo finito F_q , y $\boldsymbol{G}_T$ es el grupo multiplicativo del campo de extensión $\boldsymbol{F}_{qk}$, con k referido como el grado de incrustación de la curva elíptica, que es el número entero positivo más pequeño tal que r divide a $q^k - 1$. Los procedimientos de firma y verificación se implementan utilizando la instancia de la función hash SHA. La combinación de estos algoritmos y funciones da como resultado la adición de diferentes propiedades de seguridad a la información mediante un cripto-contenedor.

2.4.1. Funcionalidad de CP-ABE

CP-ABE es un algoritmo de cifrado basado en atributos de política de texto cifrado con política de control de acceso oculto. Permite a los propietarios de datos compartir sus datos cifrados mediante el almacenamiento en la nube con usuarios autorizados mientras se mantienen ciegas las políticas de control de acceso. En CP-ABE, los atributos juegan un papel clave en la aplicación de control de acceso.
En un esquema CP-ABE, un conjunto de atributos se encuentra relacionado con la clave privada del usuario, mientras que una estructura de acceso está relacionada con el texto cifrado. Un usuario puede descifrar el texto cifrado solo si su conjunto de atributos satisface la política de acceso.

CP-ABE crea la clave pública PK y la clave privada maestra MK, con una resistencia que cumple con λ. Posteriormente cifra M dada la estructura de acceso A (política), y crea una clave privada para el usuario u dado el conjunto de atributos S_u. El usuario u descifrará el texto cifrado CP-ABE CT utilizando la clave privada asociada SK_u.

El algoritmo utilizado en CP-ABE es el siguiente:
$Setup(\lambda)$
$CT \leftarrow Encrypt(PK, M, A)$
$SK_u \leftarrow KeyGen(MK, S_u)$
$M \leftarrow Decrypt(PK, CT, SK_u)$

Nota: Para más información acerca del Método para la creación de servicios de intercambio seguro y eficiente de información en la nube, ver el artículo **"Hermes, a parallel pattern method for secure and efficient cloud information sharing"** en el **Anexo B.1.**

2.5. Servicios de validación de normas oficiales mexicanas y protocolos DICOM/HL7

En los procesos de preparación, transporte, compartición y almacenamiento de datos para escenarios reales de gestión de datos sensibles, diferentes requerimientos no funcionales (NFRs) (por ejemplo, seguridad, eficiencia, y confiabilidad) deben ser considerados debido a las normas de gestión de la salud (por ejemplo, las normas oficiales mexicanas NOM-024-SSA3-2010 y NOM-004-SSA3-2012) y las leyes impuestas por los gobiernos y organizaciones [10], [11].

Para asegurar el cumplimiento de estas normas y leyes, es necesario contar con un servicio que permita su validación. En este sentido, en este proyecto se desarrolló un servicio que permite determinar el grado de cumplimiento de los flujos de trabajo generados en las soluciones de e-Salud con base en los marcos de referencia de normas internacionales (NIST, ISO 27001:2013 y COBIT 5) y nacionales (NOM-024-SSA3-2010) para el transporte y almacenamiento de datos sensibles.

El programa desarrollado cumple con las siguientes características:

- Funcionales:
    - Preprocesamiento de los datos para leer y manipular los archivos desde el código.

- Identificación de los flujos de trabajo que conforman los servicios de e-Salud.
- Consulta de fuentes de información utilizando APIs para obtener datos y características contextuales de los contenedores, las cuales representan las tareas que ejecuta el contenedor.
- Búsqueda de palabras clave para determinar el cumplimiento de las normas internacionales y nacionales.
- Obtención del porcentaje de cumplimiento, y generación de un reporte donde se visualizan los resultados.
- Descubrimiento de los flujos de trabajo asociados a los archivos de configuración del servicio de e-salud, y generación de una representación de este en un grafo acíclico dirigido.

- No funcionales:

  - Norma nacional que se verifica: NOM-024-SSA3-2010.
  - Normas internacionales que son verificadas: NIST, ISO 27001:2013 y COBIT 5.
  - Eficiencia y eficacia en la realización de tareas.

2.5.1. Funcionalidad del servicio de e-Salud (Alcance)

En este servicio, los archivos de configuración representan el servicio que será desplegado. El programa recibe como entrada los archivos de configuración y determina que normas nacionales e internacionales cumple dicho servicio.

El cumplimiento es mostrado mediante un reporte en donde se especifica el porcentaje de cumplimiento del servicio según las normas correspondientes. Además, el servicio realiza el descubrimiento del flujo de trabajo asociado a los archivos de configuración del servicio de e-Salud.

Las normas internacionales y nacionales son representadas mediante litas de verificación. Cada lista de verificación contiene los requerimientos de la norma correspondiente. Los requerimientos pueden ser garantizados por el servicio creado (de forma sistemática) o mediante intervención del usuario (de forma asistida).

2.5.2. Datos de entrada y salida del servicio

Entradas: En este servicio los datos de entrada son los archivos de configuración generados por la herramienta de creación de servicios y las listas de verificación de las normas nacionales e internacionales:

- .cfg: Representa el archivo de configuración generado por el modelo. En este archivo, se definen los bloques de construcción (BB), patrones (PATTERN), requerimientos no funcionales (NRF), etapas de procesamiento (STAGE) y flujos de trabajo (WORKFLOW).
- stages.json: Este archivo contiene la declaración de las etapas en el flujo, los directorios de entrada y salida de la etapa, y los datos de identificación de los contenedores utilizados en el servicio de e-Salud.
- docker-compose.yml: Contiene la declaración de los contenedores que se despliegan con la plataforma de compose o swarm.
- Normas nacionales e internacionales: Son capturadas en un diccionario de datos con una estructura definida.

Salidas: La salida del servicio de validación es un reporte para cada una de las normas nacionales e internacionales. En dicho reporte se visualiza el porcentaje de cumplimiento de cada una de las normas, así como una representación gráfica del flujo de trabajo como un grafo acíclico dirigido.

2.5.3. Diseño y desarrollo del servicio de validación

La Figura 12 muestra el diseño del servicio de validación. Como se puede observar, el servicio cuenta con los siguientes módulos:

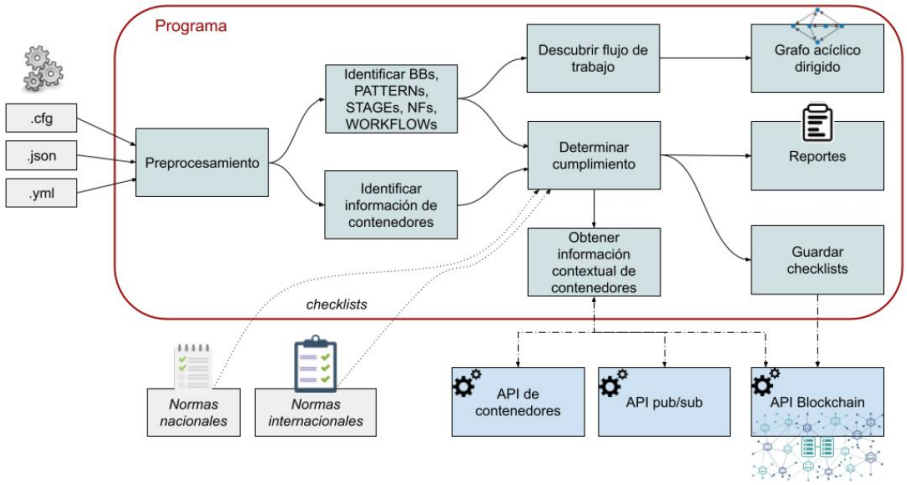


Figura 12. Representación conceptual del servicio de validación.

i) **Preprocesamiento**: Este módulo realiza la lectura de los archivos de configuración. Para identificar los flujos de trabajo, se implementó un módulo que reconoce los bloques de construcción, patrones, etapas y requerimientos no funcionales definidos en el archivo de configuración .cfg utilizando expresiones

regulares. Los componentes identificados son agregados en un diccionario de datos para su posterior consulta.

ii) **Identificación de los flujos de trabajo e información de contenedores**: Para identificar los flujos, bloques de construcción, patrones, requerimientos no funcionales y etapas de procesamiento, se utiliza el diccionario de datos. El diccionario contiene la lista de claves que cuentan con información sobre los procesos que se realizan. Algunas de las claves disponibles en el diccionario se listan a continuación: BB, name, command, image, PATTERN, task, STAGE, source, sink, transformation, WORKFLOW, catalogs, NRF, application, type, etc.

La declaración de los contenedores, así como su información de identificación se realiza en los archivos de configuración. Para obtener esta información, se utiliza el objeto decodificado por la biblioteca json en el preprocesamiento de datos. El objeto JSON decodificado contiene el nombre del flujo de trabajo, la información de cada etapa, la información de identificación de los contenedores utilizados, el nombre y el hash del contenedor.

iii) **Determinación del cumplimiento de las normas:** Las normas nacionales e internacionales fueron capturadas de forma manual en la lista de verificación directamente en el código. Para ello, se utilizaron estructuras de datos tipo diccionario. Cada diccionario representa la lista de verificación de una norma.

Para determinar el cumplimiento de las normas nacionales e internacionales, se utilizan algunas APIs para la consulta de información contextual de los contenedores que despliegan el servicio de e-Salud. Estas APIs representan fuentes de información en las cuales se basará el servicio para determinar el nivel de cumplimiento de cada una de las normas dentro del servicio de e-Salud.

iv) **Descubrimiento del flujo de trabajo:** Para el descubrimiento del flujo de trabajo, el servicio procesa la información obtenida por el módulo de preprocesamiento de datos. Posteriormente, el servicio utiliza la información contenida en la clave "STAGE" del diccionario de datos. En un flujo de trabajo, una etapa *i* representa un nodo en el grafo acíclico dirigido. Las entradas representan los nodos incidentes, mientras que las salidas representan los nodos salientes.

Nota: Para más información acerca del servicio de validación de las normas oficiales para el transporte y almacenamiento de datos sensibles, ver el reporte técnico "**Implementación de un servicio para la caracterización de procesos asociados al flujo de trabajo de sistemas de diagnóstico de cáncer de hueso largo asistido por inteligencia artificial**" en el **Anexo A.2.**

3. Grupo de trabajo y Colaboraciones Interinstitucionales

A continuación, se listan las instituciones participantes en el proyecto, así como el equipo de trabajo intrainstitucional e interinstitucional.

3.1. Instituciones participantes

Tabla 2. Instituciones participantes en el desarrollo del servicio Chimalli.

Institución	Tipo de entidad	Área
CINVESTAV Tamaulipas	Entidades *entregantes*	Grupo de Gestión de Datos y Redes de Computadoras.
Universidad Carlos III de Madrid (UC3M)		Grupo de Arquitectura de Computadoras y Tecnologías (ARCOS).
Universidad Autónoma Metropolitana (UAM)		Departamento de Ing. Eléctrica, áreas de Redes y Telecomunicaciones/Procesamiento Digital de Señales e Imágenes Biomédicas .
Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE)		Departamento de Electrónica y Telecomunicaciones de la división de física aplicada de Ensenada y Unidad Monterrey.

3.2. Grupo de trabajo intrainstitucional

Tabla 3. Equipo de trabajo intrainstitucional.

Institución	Participante	Tipo de participación en el Proyecto	Nivel SNI
Cinvestav Tamaulipas	Dr. José Luis González Compeán	Responsable Técnico del proyecto. Participa en todas las etapas del proyecto.	**1**
	Dr. Miguel Morales Sandoval	Sistemas de seguridad informática y criptografía de siguiente generación.	**2**
	MC. Diana Elizabeth Carrizales Espinoza	Diseño de sistemas de seguridad informática. Sistemas de preparación de datos médicos para cumplimentar las normas nacionales e internacionales referentes al intercambio seguro de datos.	**N**
	Ing. Catherine Alessandra Torres Charles	Realiza su tesis sobre el proyecto. Sistemas de preparación de datos médicos para cumplimentar las normas nacionales e internacionales referentes al intercambio seguro de datos en infraestructuras heterogéneas (edge-fog-cloud)	**N**
	Ing. Genaro Juan Sánchez Gallegos	Esquemas de patrones de paralelismo para mejorar la eficiencia de los sistemas de e-salud.	**N**

	Ing. Jesús Ignacio Castillo Barrio	Esquemas de alta disponibilidad para mejorar la eficiencia de los sistemas de almacenamiento que soportan a los sistemas de e-salud	**N**

3.3. Grupo de trabajo interinstitucionales

Tabla 4- Equipo de trabajo interinstitucional

Institución	Participante	Tipo de participación en el Proyecto	Nivel SNI
UC3M	Dr. Jesús Carretero Pérez	Revisión de arquitectura y diseño.	**N**
UAM	Dr. Ricardo Marcelín Jiménez	Sistemas de distribuidos y preparación y almacenamiento de datos.	**1**
CICESE	Dr. Andrei Tchernykh	Diseño de criptosistemas. Desarrollo de búsquedas cifradas para sistemas PACS.	**2**

4. Estatus de Chimalli

La Figura 13 muestra el resumen de los productos de Chimalli. En él se encuentra el estatus de cada uno de los productos: comprometido o no comprometido, y un resumen de su estado (diseño, en desarrollo, en evaluación, prototipado o en producción).

Entregable	ID	Producto	Estado					Etapa		Comprometido	NMT
			Diseño	En desarrollo	En Evaluación	Prototipado	En Producción	1	2		
Zamná	Muyal-Chimalli-P1	Servicios de preparación y recuperación de datos médicos configurables que incluya los requerimientos de seguridad, trazabilidad, integridad y eficiencia.	*	*	*	*		✓		Si	5
	Muyal-Chimalli-P2	Mecanismos de trazabilidad de datos basados en blockchain.	*	*	*	*		✓		Si	4
	Muyal-Chimalli-P3	Mecanismos de control de acceso de usuarios.	*	*	*	*		✓		Si	6
	Muyal-Chimalli-P4	Servicios de validación de normas oficiales mexicanas y protocolos DICOM/HL7.	*	*		*		✓		Si	4
	Muyal-Chimalli-P5	Servicio que permita la utilización de técnicas de criptografía de siguiente generación para la transformación de datos en objetos seguros.	*	*	*	*		✓		Si	5
	Muyal-Chimalli-P6	Un sistema que permita cumplimentar con las normas y protocolos oficiales.	*	*		*			✓	No	4
	Muyal-Chimalli-P7	Esquemas de preparación y recuperación de datos que permitan cumplir con los requerimientos no funcionales establecidos en las normas.	*	*	*	*			✓	No	5
	Muyal-Chimalli-P8	Esquemas de preparación de datos para el cifrado e intercambio seguro de datos.	*	*	*	*			✓	No	5
	Muyal-Chimalli-P9	Esquema de búsqueda de datos cifrados en la nube	*	*		*		✓		No	5

Simbología	
*	Terminado
X	En proceso
NMT	Nivel de Madurez Tecnológica

Figura 13. Resumen de los productos de Chimalli.

ANEXOS

A. Reportes Técnicos

A.1. Reporte técnico "Trazabilidad y verificabilidad de contenidos médicos"

Servicios de trazabilidad.

El proceso de trazabilidad juega un papel importante dentro de los flujos de cadenas de valor debido a que hace referencia a la posibilidad de identificar el origen y las distintas etapas por las que ha pasado un producto a lo largo de todo su ciclo de vida (proceso productivo, distribución y logística, hasta llegar a un consumidor final).

Este proceso cumple una parte fundamental dentro del presente proyecto debido a que permitirá a cualquiera de las entidades involucradas (personal médico y usuarios finales) acceder a la información de cada una de las etapas por las cuales ha pasado el contenido digital, verificando si este cumple con…

Leer reporte completo…

A.2. Reporte técnico "Implementación de un servicio para la caracterización de procesos asociados al flujo de trabajo de sistemas de diagnóstico de cáncer de hueso largo asistido por inteligencia artificial"

Objetivo

Desarrollar un programa que a partir de unos archivos de configuración .cfg, .json y .yml, generados por un creador de servicios, determine el grado de cumplimiento de los flujos de trabajo asociados al archivo de configuración, con base en marcos de referencia de normas internacionales (NIST, ISO 27001:2013 y COBIT 5) y nacionales (NOM-024-SSA3-2010).

El programa, además, descubre el fujo …

Leer reporte completo…

B. Artículos

B.1. Hermes, a parallel pattern method for secure and efficient cloud information sharing

Security information systems are key tools for organizations to prevent/mitigate incidents/risks such as unauthorized accesses, data alterations as well as violations of privacy and confidentiality. However, in sensitive information sharing real cloud scenarios, the efficiency issues and lack of flexibility of these systems represent an open challenge. This paper presents Hermes, a parallel pattern method for building efficient security systems for organizations to share, exchange, and tracing sensitive information in the cloud. In this method, on-premises security cryptosystems and blockchain software are converted into independent and autonomous cloud services called SecBoxes. A novel implicit parallel pattern in conjunction with load balancing are added to SecBoxes for improving the efficiency of security services and the end-user service experience. Hermes allows organizations to support online information sharing patterns among multiple participants by coupling sets of SecBoxes for meeting multiple combinations of security requirements such as confidentiality, integrity, non-repudiation, authentication, and traceability…
Leer artículo completo…

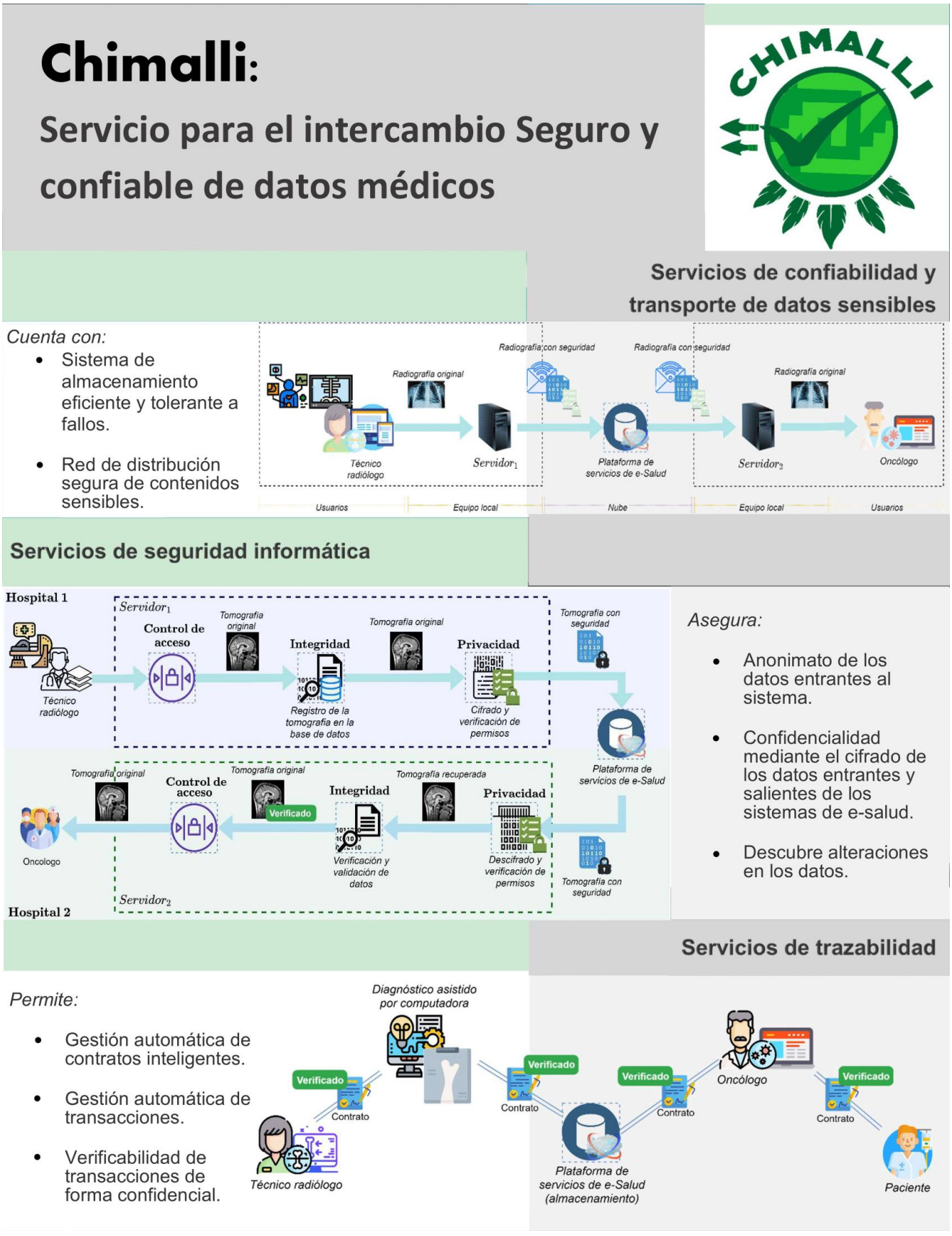
B.2. FedFlow: A Federated Platform to Build Secure Sharing and Synchronization Services for Health Dataflows

Data synchronization and content delivery services are key to support healthcare dataflows built by organizations. These types of services must prepare and process the data to accomplish mandatory non-functional requirements such as security and reliability imposed by the governmental protocols and norms. Moreover, organizations expect to perform these preparation processes in secure and cost-efficient manners. This is a challenge as multiple applications, infrastructures, and platforms participate in healthcare dataflows. This paper presents FedFlow, a federated content distribution platform to build infrastructure-agnostic health data sharing and synchronization services to support healthcare dataflows. FedFlow manages user accounts and content catalogs (e.g., medical imagery) using Pub/Sub and delivery models. In FedFlow, organizations can use these models to create and manage intra-dataflows (e.g., data exchange inside hospitals) by using private cloud resources as well as inter-dataflows or federated services for exchange data with other organizations (e.g., hospitals exchanging data through multiple cloud resources). FedFlow creates secure and efficient data sharing and synchronization patterns for intra-dataflows and inter-dataflows by using implicit parallel data preparation schemes…
Leer artículo completo…

C. Posters

C.1. Poster cualitativo de Chimalli

Descargar archivo PDF aquí.



C.2. Poster cuantitativo de Chimalli

Descargar archivo PDF aquí.

Chimalli:

Servicio para el intercambio Seguro y confiable de datos médicos

Servicios de confiabilidad y transporte de datos sensibles

71 **petabytes** de imágenes médicas almacenados en el sistema de almacenamiento ***PACS*** por el Instituto Nacional de Rehabilitación.

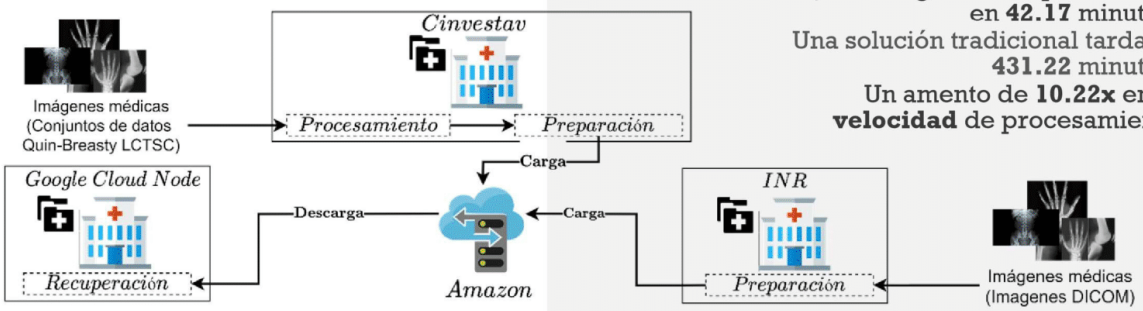
En comparación con el sistema de almacenamiento PACS, *Chimalli* reduce el almacenamiento total de los archivos hasta en un **70%**

*Esto significa una reducción en costos del **70%** de almacenamiento en la nube.*

Ejemplo de intercambio de datos entre organizaciones usando *Chimalli*

Los esquemas costo-eficiencia de *Chimalli* reducen el tiempo para procesar los

100,971 imágenes son procesadas en **42.17** minutos.
Una solución tradicional tardaría **431.22** minutos.
Un amento de **10.22x** en la **velocidad** de procesamiento



Chimalli registra eficientemente las operaciones realizadas en la blockchain

En Amazon Managed Blockchain, ***Chimalli*** te permite **ahorrar** hasta un **32%** de costos en la trazabilidad de **100,971** imágenes médicas en comparación a una solución tradicional.

Servicios de seguridad informática

Los servicios de seguridad informática de ***Chimalli*** permiten **cumplir en un 70% las normas** del ISO 27001-13, NIST y COBIT. Sin estos servicios, este porcentaje baja al **20%** considerando solo tolerancia a fallos.

Características de *Chimalli*:

- ✓ Confiabilidad
- ✓ Eficiencia
- ✓ Integridad
- ✓ Confidencialidad
- ✓ Seguridad

D. Tesis

D.1. Método de construcción de servicios de seguridad informática para sistemas de continuidad en infraestructuras heterogéneas de cómputo

La unión de infraestructuras Edge-Fog-Cloud en una sola infraestructura coherente (llamada EFC, por sus siglas en inglés) resulta clave para que las organizaciones gestionen el ciclo de vida de los datos producidos por ambientes del Internet de las cosas (IoT, por sus siglas en inglés). Un EFC permite a las organizaciones crear un esquema de continuidad (de manejo, análisis y/o procesamiento) desde que los datos son producidos por los dispositivos (generalmente sensores), adquiridos y preparados (típicamente en el Edge), hasta que son procesados (en el Fog / Cloud) y finalmente almacenados y consumidos mediante servicios o procesos para la toma de decisiones (Cloud). El tránsito y acceso a los datos y/o información producida a través de las distintas fronteras de un EFC resulta crucial para procesos de toma de decisiones o análisis de datos en tiempo real. Sin embargo, introduce amenazas y riesgos de seguridad de los datos cuando estos son sensibles, como en el caso de aplicaciones médicas o entornos empresariales, militares, industriales, entre otros.

Los tipos de servicio de seguridad requerido en EFC cambian en función de las amenazas y riesgos de seguridad que se observan en cada frontera. En este sentido, el principal problema es proveer servicios de seguridad heterogéneos, en forma continua (ininterrumpida) durante el ciclo de vida de los datos, lo cual no es trivial debido a la heterogeneidad de los recursos de estas infraestructuras. La heterogeneidad de los recursos se observa en las capacidades computacionales de los dispositivos en cada infraestructura, los requerimientos del procesamiento en cada capa, las capacidades de almacenamiento o su localización geográfica. En este trabajo de tesis se propone diseñar, desarrollar y evaluar un método de construcción de servicios de seguridad informática continua para EFC, el cual se propone este basado en un modelo de colocación de servicios de seguridad encapsulados en módulos de procesamiento genéricos independientes, que se acoplen automáticamente mediante estructuras de entrada…

Leer protocolo de tesis completo…

Referencias

[1] M. A. Malik, «Internet of Things (IoT) healthcare market by component (implantable sensor devices, wearable sensor devices, system and software), application (patient monitoring, clinical operation and workflow optimization, clinical imaging, fitness and wellness measu,» *Global opportunity analysis and industry forecast, 2014–2021,* pp. Allied Market Research, 124, 2016.

[2] J. &. R. D. Gantz, «The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east.,» *IDC iView: IDC Analyze the future, 2007(2012),* pp. 1-16, 2012.

[3] . D. Reinsel, J. Gantz y . J. Rydning, «The digitization of the world from edge to core,» *IDC White Paper,* 2018.

[4] M. Marjani, . F. Nasaruddin, A. Gani, A. Karim, . I. A. T. Hashem, A. Siddiqa y I. Yaqoob, «Big IoT data analytics: architecture, opportunities, and open research challenges,» *IEEE Access,* pp. 5247--5261, 2017.

[5] M. R. Anawar, S. Wang, M. Azam Zia, A. K. Jadoon, U. Akram y S. Raza, «Fog computing: An overview of big IoT data analytics,» *Wireless Communications and Mobile Computing,* 2018.

[6] B. P. Rimal, E. Choi y I. Lumb, «A taxonomy and survey of cloud computing systems,» *2009 Fifth International Joint Conference on INC, IMS and IDC,* pp. 44--51, 2009.

[7] V. Mosco, «To the cloud: Big data in a turbulent world,» *Routledge,* 2015.

[8] V. Malik y S. Singh, «Cloud, Big Data \& IoT: Risk Management,» *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon),* pp. 258--262, 2019.

[9] A. Botta, W. De Donato, V. Persico y A. Pescapé, «Integration of cloud computing and internet of things: a survey,» *Future generation computer systems,* pp. 684--700, 2016.

[10] H. Mier y T. Delgadillo, «Regulación del acceso al expediente clínico con fines de investigación en México,» *Revista CONAMED,* vol. 22, pp. 27--31, 2018.

[11] Phillips, «International data-sharing norms: from the OECD to the General Data Protection Regulation (GDPR),» *Human genetics,* vol. 137, pp. 575--582, 2018.

[12] C. B. Tan, M. H. A. Hijazi, Y. Lim y A. Gani, «A survey on proof of retrievability for cloud data integrity and availability: Cloud storage state-of-the-art, issues, solutions and future trends,» *Journal of Network and Computer Applications,* vol. 110, pp. 75--86, 2018.

[13] Gunawi, Hao, S. O, Laksono, Satria, Adityatama y Eliazar, «Why does the cloud stop computing?: Lessons from hundreds of service outages,» *SoCC, ACM,* pp. 1--16, 2016.

[14] Bala y Chana, «Fault tolerance-challenges, techniques and implementation in cloud computing,» *International Journal of Computer Science Issues (IJCSI),* vol. 9, p. 288, 2012.

[15] R. Marcelín-Jiménez, J. L. Ramírez-Ortíz, E. R. De La Colina, M. Pascoe-Chalke y J. L. González-Compeán, «On the Complexity and Performance of the Information Dispersal Algorithm,» *IEEE Access,* pp. 159284--159290, 2020.

[16] Bhushan y Gupta, «Security challenges in cloud computing: state-of-art,» *International Journal of Big Data Intelligence,* vol. 4, pp. 81--107, 2017.

[17] French-Baidoo, Asamoah y Oppong, «Achieving confidentiality in electronic health records using cloud systems,» *IJCNIS,* vol. 10, p. 18, 2018.

[18] Morales, Gonzalez, Diaz y Sosa, «A pairing-based cryptographic approach for data security in the cloud,» *IJISP,* vol. 17, pp. 441--461, 2018.

[19] Odelu, Rao, Kumari, Khan y Choo, «Pairing-based CP-ABE with constant-size ciphertexts and secret keys for cloud environment,» *Computer Standards & Interfaces,* vol. 54, pp. 3--9, 2017.

[20] M. Mitzenmacher, «The power of two choices in randomized load balancing,» *IEEE Transactions on Parallel and Distributed Systems,* vol. 12, pp. 1094--1104, 2001.

[21] P. Morales-Ferreira, M. Santiago-Duran, C. Gaytan-Diaz, J. Gonzalez-Compean, V. J. Sosa-Sosa y I. Lopez-Arevalo, «A data distribution service for cloud and containerized storage based on information dispersal,» *SOSE,* pp. 86--95, 2018.

[22] F. Della Rosa, «Worldwide Software as a Service and Cloud Software Forecast, 2020–2024,» International Data Corporation (IDC), 2020.

[23] E. a. H. J. J. a. O. A. a. R. D. a. S. G. a. T. H.-Y. Brynjolfsson, «COVID-19 and remote work: An early look at US data,» National Bureau of Economic Research, 2020.

[24] J. a. M. B. a. V. W. Kelly Finnerty and Sarah Fullick and Helen Motha and Navin Shah, Cyber Security Breaches Survey 2019, Department for Digital, Culture, Media and Sport, 2019.

[25] C. H. a. W. C. a. L. Y. a. Y. D. a. S. J. a. Y. T. a. H. C. a. D. Chen, «Toward security as a service: A trusted cloud service architecture with policy customization,» *Journal of Parallel and Distributed Computing,* vol. 149, pp. 76-88, 2021.

[26] S. Kokolakis, «Privacy attitudes and privacy behaviour: A review of current research on the privacy paradox phenomenon,» *Computers and Security,* July 2015.

[27] C. Jang y W. Sung, «Beyond the Privacy Paradox: The Moderating Effect of Online Privacy Concerns on Online Service Use Behavior,» *Telematics and Informatics,* p. 101715, 2021.

[28] J. a. R. V. Daemen, The Design of Rijndael: The Advanced Encryption Standard (AES), Springer Nature, 2020.

[29] M. Morales-Sandoval, J. L. Gonzalez-Compean, A. Diaz-Perez y V. J. Sosa-Sosa, «A Pairing-based Cryptographic Approach for Data Security in the Cloud,» *Int. J. Inf. Secur.,* vol. 17, p. 441–461, August 2018.

[30] F. B. a. S. H. a. T. Halevi, «Supporting private data on Hyperledger Fabric with secure multiparty computation,» *IBM Journal of Research and Development,* vol. 63, nº 2/3, pp. 3-8, 2019.

[31] N. a. L. J. a. Z. Y. a. G. Y. Chen, «Efficient CP-ABE Scheme with Shared Decryption in Cloud Storage,» *IEEE Transactions on Computers,* 2020.

[32] V. J. Sosa-Sosa, A. Barron, J. L. Gonzalez, J. Carretero y I. Lopez-Arevalo, «Improving Performance and Capacity Utilization in Cloud Storage for Content Delivery and Sharing Services,» *IEEE Transactions on Cloud Computing,* pp. 1-1, 2020.

[33] D. Giry, «NIST Report on Cryptographic Key Length and Cryptoperiod (2020),» Key length, 2020.

[34] E. a. B. E. a. B. W. a. P. W. a. S. M. a. o. Barker, «Recommendation for Key Management: Part 1-General,» National Institute of Standards and Technology, Technology Administration, 2020.

[35] A. CloudFront, «Amazon cloudfront,» 2014. [En línea]. Available: URL: http://aws.amazon.com/cloudfront. [Último acceso: 2019 07 15].

[36] Gonzalez, Perez, Sosa-Sosa, Sanchez y Bergua, «SkyCDS: A resilient content delivery service based on diversified cloud storage,» *SIMPAT, Elsevier,* pp. 64--85.