



Servicio para el transporte
y almacenamiento de
datos médicos

Entregable **2.2**
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
de servicios de análisis de datos de salud.*

Autorizaciones

<i>Fecha: 12/11/2021</i>	<i>Publicación: 15/11/2021</i>	<i>Versión: 2.0</i>
Seguimiento	Nombre completo	Fecha
Elaboró:	M. C. Diana Elizabeth Carrizales Espinoza	05/11/2021
Revisó:	Dr. Ricardo Marcelín Jiménez	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 producida 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 (p. ej., borde, niebla, nube o cualquier combinación de ellos) es trascendental para asegurar el intercambio seguro y confiable de datos sensibles (p. ej., expedientes clínicos, tomografías, resonancias magnéticas, etc.).

En este documento se describe **Painal**, una plataforma para el transporte y almacenamiento de datos médicos. **Painal** se compone de tres servicios principales:

- Un servicio de repositorio de catálogos para gestionar los servicios que se crean en la plataforma MUYAL y los e-sistemas que se publiquen.
- Un Mecanismo de usabilidad costo-beneficio para el almacenamiento y transporte de datos.
- Un Servicio de publicación/suscripción (pub/sub) para el manejo de catálogos, fuentes y repositorios de e-salud.

Índice

1. Introducción.....	6
2. Servicio para el transporte y almacenamiento de datos médicos.....	6
2.1. Servicio para el manejo, carga y descarga de servicios desde un catálogo/repositorio de servicios.....	8
2.1.1. Sistema federado para la distribución de contenidos y administración de requerimientos no funcionales.	9
Arquitectura del sistema.....	9
2.2. Mecanismo de usabilidad costo-beneficio para el almacenamiento y transporte de datos.	12
2.2.1. Servicios de almacenamiento confiables sin servidor: Principios de diseño	13
Construcción de estructuras de almacenamiento confiables	14
Creación de sistemas de almacenamiento P2P sin servidor en una malla para el almacenamiento de datos.....	16
Descripción de los componentes utilizados para el desarrollo del prototipo	18
2.3. Servicio de publicación/suscripción (pub/sub) para el manejo de catálogos, fuentes y repositorios.	19
2.3.1. Diseño y componentes del servicio	22
2.3.2. Componentes del servicio de publicación/suscripción: Agentes y clientes.....	24
Gestor de Painal.....	26
Gestor de metadatos	26
Gestor de flujos de trabajo.....	27
Flujo de trabajo de entrega.....	28
Flujo de trabajo de recuperación	29
3. Grupo de trabajo y Colaboraciones Interinstitucionales.....	31
3.1. Instituciones participantes	31
3.3. Grupo de trabajo intrainstitucional	32
3.4. Grupo de trabajo interinstitucionales.....	32

4. Estatus de Painal.....32

5. Anexos33

 Artículos33

 Posters.....35

6. Referencias36

Índice de figuras

Figura 1	Arquitectura del sistema	9
Figura 2	Arquitectura del sistema detallada	10
Figura 3	Representación conceptual de una estructura de almacenamiento	14
Figura 4	Representación conceptual de diferentes mallas de almacenamiento utilizadas para gestionar datos de manera confiable	17
Figura 5	Ejemplo de malla de almacenamiento.....	17
Figura 6	Las capas sobrepuestas del servicio Pub/Sub.	20
Figura 7	Los componentes del servicio Pub/Sub de Painal.....	23
Figura 8	Los componentes y subcomponentes del servicio de Pub/Sub de Painal.	24
Figura 9	Ejemplo de tubería de dispersión con una configuración $n = 5, k = 3$	29
Figura 10	Ejemplo de tubería de recuperación con una configuración $n = 5, k = 3$	30

1. Introducción

El volumen de datos producidos y gestionados por las organizaciones ha ido creciendo en los últimos años, esto debido a que los usuarios finales asociados a las organizaciones producen, almacenan y utilizan datos de forma constante y continua, lo que produce un efecto de acumulación de datos [1]. Los usuarios finales y las aplicaciones consumen los servicios de almacenamiento en la nube a través de un modelo de externalización denominado pago por uso (pay-as-you-go) [2]. A pesar de que estos servicios se construyen utilizando sistemas distribuidos, una acumulación constante de datos crea gradualmente una colección centralizada de datos en los servicios de almacenamiento. Esto no solo da lugar a un único punto de fallo en los escenarios de interrupción [3], sino que también produce una dependencia con el proveedor de estos servicios en la nube [4] conocida como vendor lock-in [5].

La gestión del almacenamiento de datos resulta así un proceso clave para que las organizaciones eviten la saturación y centralización de los nodos de almacenamiento en la nube, así como para mejorar el tiempo de respuesta de los intercambios de datos en línea con los socios y usuarios finales. Los aspectos de rendimiento y gestión del almacenamiento resultan críticos para los enfoques jerárquicos a través de infraestructuras heterogéneas (es decir, de entornos de *edge-fog-cloud* [6], [7]),

En este documento se presenta el servicio Painal, el cual se compone por un conjunto de servicios que permiten realizar el transporte, almacenamiento, de datos médicos y servicios a través de catálogos accedidos a través de un modelo de publicación suscripción. Estos servicios permiten realizar la compartición segura de servicios y datos entre diferentes instituciones con el objetivo de mejorar los procesos de compartición de datos sensibles.

2. Servicio para el transporte y almacenamiento de datos médicos¹²

¹ Para a descargar el software de Painal, dar clic [aquí](#).

² Para visitar la página web del proyecto, dar clic [aquí](#).

Painal es un conjunto de servicios y sistemas desarrollados para que las organizaciones de salud y la comunidad científica puedan i) almacenar, distribuir y localizar sistemas o servicios de procesamiento a través de catálogos de servicios ii) generar soluciones que permitan brindar usabilidad costo-beneficio del almacenamiento y transporte de datos y iii) almacenar, publicar y transmitir repositorios de datos de manera local (intrainstitucional) y federada (interinstitucional) utilizando un modelo de publicación/suscripción.

Painal permite crear catálogos de servicios en las cuales las organizaciones pueden colocar sus sistemas, servicios o aplicaciones para que otras instituciones de la federación puedan descargar y utilizar.

Por otro lado, para almacenar los resultados producidos por sus servicios y aplicaciones es necesario generar sistemas de almacenamiento que consideren las características de costo-beneficio del almacenamiento y transporte de datos. Painal ofrece una arquitectura de malla para generar el almacenamiento de datos considerando los recursos disponibles en la infraestructura de la organización que la está utilizando.

Por último, Painal permite crear catálogos de datos para el almacenamiento y compartición de datos entre múltiples organizaciones de forma segura. Dentro de los catálogos de datos se pueden almacenar los datos sin procesar o los resultados obtenidos por algún tipo de procesamiento. Cuenta con un mecanismo de publicación/suscripción que permite generar catálogos de datos, agregar nuevos datos a dichos catálogos y descargar el contenido de estos. Todo el proceso de publicación y suscripción se basa en la utilización de tokens de acceso que permiten verificar la entidad de las organizaciones de la federación y los permisos que tienen para acceder a los catálogos.

Painal se compone por los siguientes productos:

1. Servicio manejador de catálogos para la carga y descarga de servicios.
2. Mecanismo para el almacenamiento en la nube que brinda usabilidad costo-beneficio para el almacenamiento y transporte de datos.

3. Servicio de publicación suscripción para el manejo de catálogos, fuentes y repositorios de datos.

2.1. Servicio para el manejo, carga y descarga de servicios desde un catálogo/repositorio de servicios

En esta sección se presenta el diseño de un sistema de distribución de contenidos federado (FCDS, por sus siglas en inglés) desarrollado para Painal y empleado para crear servicios de sincronización de datos médicos que son tecnológicamente agnósticos de la infraestructura, lo cual permite a las organizaciones desplegar sus servicios en cualquier nube privada, pública o híbrida. La Figura 1 muestra una representación conceptual del FCDS de Painal, que permite compartir datos (p. ej., tomografías, mamografías y resonancias magnéticas) de forma sincrónica entre tres hospitales a través de una Red de Entrega de Contenidos (CDN, por sus siglas en inglés) [8] [9]. La motivación para el uso de la federación de servicios es que las organizaciones puedan tener un gobierno sobre sus servicios, datos, infraestructura y los requisitos no funcionales mediante la inclusión de métodos para cumplirlos. Además, los miembros de la federación pueden generar un conjunto de servicios que permita a los participantes compartir recursos y datos. Los servicios incluyen la preparación de datos y recuperación de datos [10] descritos en Zamná, para preprocesar los datos antes de distribuirlos a otros participantes de la federación o a la nube, aplicando diferentes filtros que cumplan con los requisitos no funcionales requeridos por una organización. Al combinar los catálogos de servicios con la compartición de recursos y datos se obtiene una alta fiabilidad, la distribución de la carga, la integridad de los datos, la confidencialidad de los datos y la independencia del proveedor de servicios de nube.

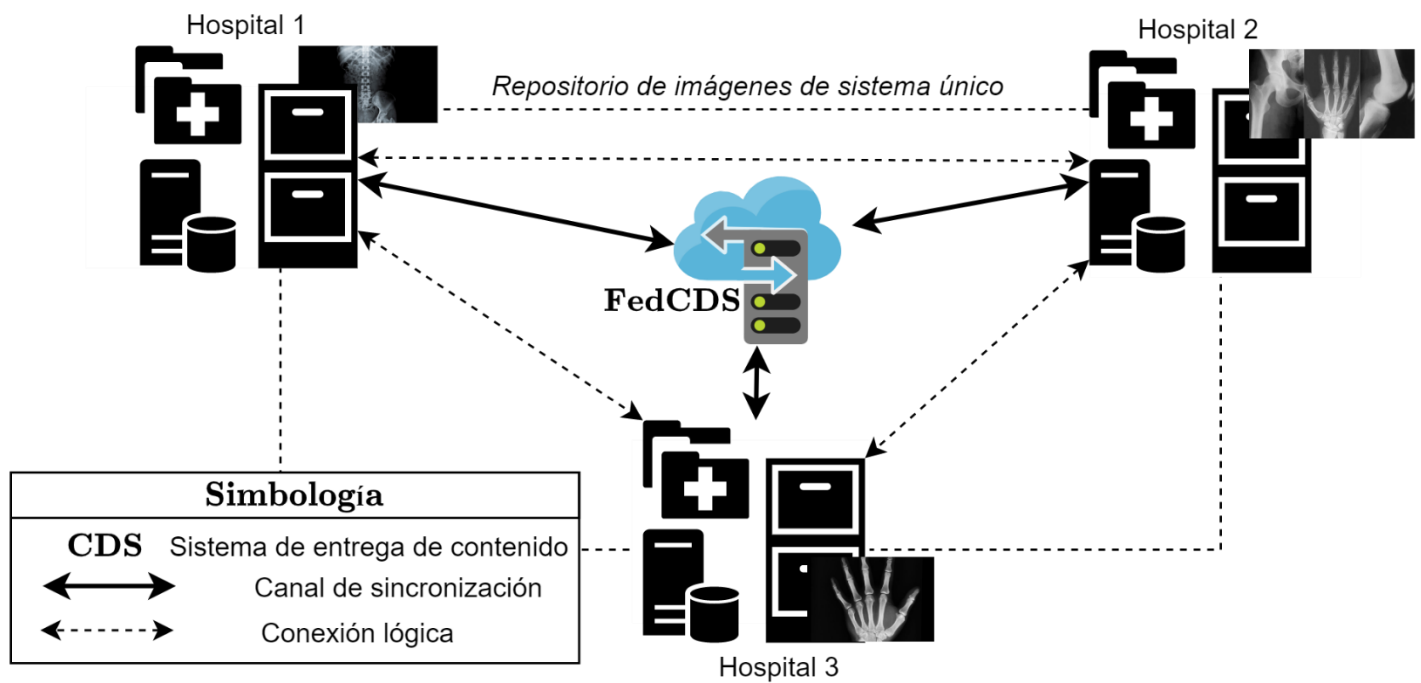


Figura 1 Arquitectura del sistema

2.1.1. Sistema federado para la distribución de contenidos y administración de requerimientos no funcionales.

En esta sección se describen los principios de diseño del sistema federado de distribución de contenidos (FCDS, por sus siglas en inglés) de Painal. Se presentan la arquitectura y los principales componentes del sistema, así como los esquemas basados en patrones paralelos en contenedores para preparar/recuperar los datos cuando se cargan/descargan a través del FCDS de Painal. Estos esquemas gestionan los requisitos no funcionales requeridos por las organizaciones para cumplir con las regulaciones impuestas por las organizaciones y los gobiernos.

Arquitectura del sistema

La Figura 2 muestra la arquitectura detallada y los principales componentes del FCDS de Painal para un escenario compuesto por tres hospitales diferentes, de esta manera como los servicios apilados de tres organizaciones distintas. Cada organización (hospital) despliega dos tipos de servicios: servicios federados y servicios locales. El objetivo de distinguir estos dos tipos de servicios es que cada organización tiene el control de sus datos y recursos, pero, al mismo

tiempo algunos servicios permiten a las organizaciones compartir recursos y datos de forma transparente y segura. Los servicios federados son todos aquellos que son visibles para otras organizaciones. A través de estos servicios, organizaciones tienen acceso a los recursos y datos, a los que tienen acceso autorizado, disponibles en otras organizaciones (imágenes médicas o historias clínicas). Para ello, los servicios federados para cada organización son los siguientes:

- **Front-end:** Es la capa a través de la cual los clientes y participantes de la federación acceden a los servicios de la organización. Dispone de una API que permite la interacción con otros participantes, así como la validación de tokens y un proxy para redirigir las peticiones a los servicios de los niveles inferiores o disponibles en otras organizaciones de la federación.
- **Autenticación:** Esta capa se encarga de validar el acceso de los usuarios externos, que son miembros de la federación, a una organización específica.
- **Pub/Sub:** Se encarga de gestionar las solicitudes de publicación y suscripción de datos (imágenes o historias clínicas o historias clínicas) de los usuarios pertenecientes a la federación.

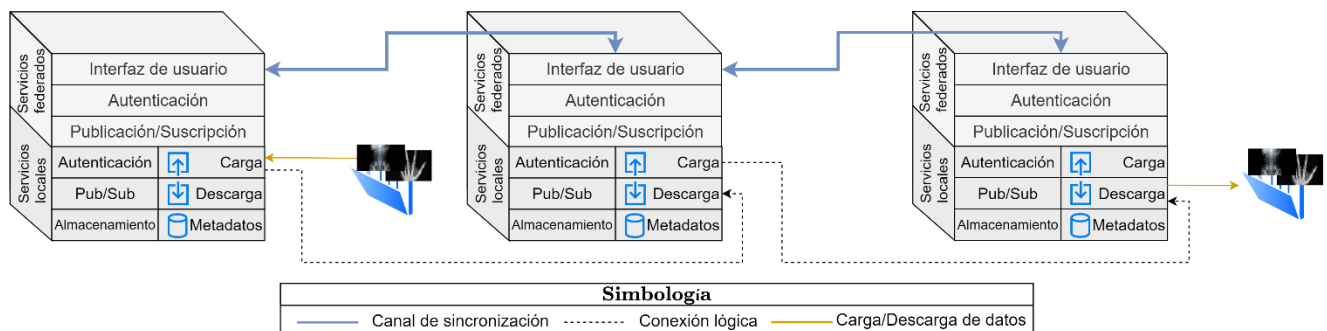


Figura 2 Arquitectura del sistema detallada

Los servicios locales se encargan de gestionar el sistema dentro de cada organización, incluida la gestión de los usuarios internos, de esta manera como los datos que se manejan en la organización. La idea de estos servicios es que cada organización pueda gestionar sus catálogos de datos de forma independiente a los publicados en la federación. En el caso de que los usuarios quieran publicar sus datos en la federación, podrán hacerlo con solo cambiar

el estado del catálogo de "local/privado" a "federado", o igualmente dando acceso solo a un determinado grupo de usuarios de la federación. Dentro de los servicios locales, se incluyen tres capas para gestionar la autenticación de usuarios y tokens de la organización, gestionar las publicaciones y suscripciones, y gestionar el almacenamiento, así como los metadatos asociados a estos servicios. Además, estos servicios incluyen dos clientes: uno de carga y otro de descarga de datos. El cliente de carga se encarga de preparar los datos para su publicación y posteriormente enviarlos a través de la red de distribución de contenidos a todos aquellos usuarios que se hayan suscrito a los datos, tanto dentro de la organización como en la federación. Por su parte, el cliente de descarga se encarga de realizar el proceso inverso: obtener los datos de la CDN y recuperarlos para que los usuarios puedan visualizarlos y manipularlos. La idea de estos clientes es que permitan una sincronización eficaz y segura de los datos entre organizaciones de forma automática.

El intercambio de mensajes y metadatos entre los miembros de la federación se efectúa a través de API REST, mientras que los datos se transportan utilizando una red de entrega de contenidos del servicio de publicación/suscripción de Painal descritos en la sección 2.3 de este documento. Los esquemas de preparación de datos [10] se despliegan para añadir requisitos no funcionales a los datos antes de ser transportados. Estos esquemas procesan los datos para añadir propiedades como la rentabilidad, la fiabilidad y la seguridad para ser compartidos con otras organizaciones. Del mismo modo, los esquemas de recuperación de datos se implementan para recuperar los datos originales aplicando las operaciones inversas de los esquemas de preparación (por ejemplo, descomprimiendo los datos comprimidos en el esquema de preparación) Estos esquemas son descritos con mayor profundidad en la documentación del componente Zamná.

Nota: Para más información acerca del servicio para el manejo, carga y descarga de servicios desde un catálogo/repositorio de servicios, ver el artículo **“A Federated Content Distribution System to Build Health Data Synchronization Services”** en el Anexo B.1.

2.2. Mecanismo de usabilidad costo-beneficio para el almacenamiento y transporte de datos.

En esta sección se presenta una arquitectura de malla para el almacenamiento de datos que permite crear y operar servicios de almacenamiento configurables, fiables y flexibles para infraestructuras heterogéneas, la cual permite crear servicios de almacenamiento confiables sin servidor (SeRSS) en Painal.

La arquitectura de malla para el almacenamiento de datos se basa en una estructura de almacenamiento, que se asocia para gestionar nodos de almacenamiento (SN, por sus siglas en inglés). Los SN son sistemas de almacenamiento tradicionales que incluyen sistemas como la entrega/recuperación de archivos, balanceadores de carga, distribución de datos y los sistemas de colocación, así como los nano servicios para atender los requisitos no funcionales (NFR, por sus siglas en inglés) basados en el algoritmo de dispersión de la información (IDA) [11], [12] para esquemas tolerantes a fallos, esquemas de preparación para resolver problemas de seguridad, así como patrones paralelos para mejorar la eficiencia de los componentes de los SN.

Los componentes de los SNs se encapsulan en contenedores virtuales, lo que permite a las organizaciones desplegar los SNs en infraestructuras múltiples y heterogéneas. En el servicio SeRSS de Painal, estos componentes se gestionan utilizando una estructura de almacenamiento, que convierte las SNs en piezas de software independientes y portátiles (microservicios) que pueden acoplarse a otros SNs en forma de redes de almacenamiento descentralizadas (P2P).

En los nano servicios de fiabilidad, SeRSS de Painal implementa una técnica de codificación/decodificación de datos basada en el Algoritmo de Dispersión de Información (IDA) [11], [12] descrita a mayor profundidad en Zamná. Esta técnica divide cada archivo en n porciones redundantes (que se distribuirán en n nodos de almacenamiento), donde m porciones son suficientes para recuperar el archivo original (m representa el número mínimo de nodos disponibles necesarios para recuperar los

archivos). Cuando se utiliza esta técnica, los sistemas de almacenamiento pueden soportar $n - m$ cortes/fallos con el coste de $|F|/m$, donde $|F|$ es el tamaño (en bits) del archivo original que se va a almacenar. Por lo tanto, hay una cantidad de información redundante igual a $|F|(n - m)/m$ bits que son la clave para soportar las m porciones redundantes que faltan o fallan. Los nano servicios de seguridad implementan servicios de criptografía para garantizar cualquier integridad, confidencialidad o autenticación, mientras que los nano servicios de patrones paralelos mejoran el rendimiento de los nano servicios descritos anteriormente.

La idea básica es que las organizaciones puedan elegir los parámetros de almacenamiento a alto nivel (es decir, el número de nodos de almacenamiento necesarios para una solución determinada) y los parámetros NFR (por ejemplo, n, m o los tipos de seguridad y eficiencia requeridos por los usuarios finales). SeRSS utiliza esta información para construir automáticamente un sistema de almacenamiento fiable en forma de sistema P2P sin servidor que puede ser consumido por los usuarios finales como un servicio, que puede ser desplegado en infraestructuras múltiples y heterogéneas. SeRSS también crea automáticamente los correspondientes nano servicios NFR y los añade al SN en los servicios de almacenamiento. Desarrollamos e implementamos un prototipo basado en la arquitectura de malla de almacenamiento de SeRSS y construimos servicios de almacenamiento sobre ella. Se realizó un estudio de caso basado en la gestión de imágenes médicas mediante el uso de estos servicios de almacenamiento para evaluar el concepto SeRSS. La evaluación experimental reveló la viabilidad del uso de SeRSS para construir soluciones de almacenamiento flexibles y fiables que mitiguen las interrupciones en infraestructuras múltiples y heterogéneas y los problemas de bloqueo de proveedores [4]

2.2.1. Servicios de almacenamiento confiables sin servidor: Principios de diseño

En esta sección, se describen los principios de diseño de una arquitectura de malla de almacenamiento para crear y operar servicios de almacenamiento fiables. En primer lugar, se describen los componentes de una estructura de almacenamiento, que se asocian a los nodos de almacenamiento (SN, por sus siglas en inglés). Además, en esta sección, se describe

un conjunto de nano servicios que proporcionan características no funcionales como la fiabilidad seguridad y eficiencia. Por último, los componentes para construir sistemas de almacenamiento en forma de un sistema P2P son descritos.

Construcción de estructuras de almacenamiento confiables

Las estructuras de almacenamiento gestionadas en los servicios SeRSS de Painal permiten a las organizaciones almacenar y gestionar los datos de manera eficiente y confiable. Cada estructura de almacenamiento se despliega como un contenedor virtual (VC, por sus siglas en inglés) sobre recursos físicos. Esto añade la característica de portabilidad a las estructuras construidas, permitiendo a las organizaciones mover la estructura de almacenamiento a través de diferentes infraestructuras. La Figura 3 muestra la representación en pila de una estructura de almacenamiento. Estas estructuras implementan dos componentes principales para gestionar los datos dentro de una organización: i) un sistema de almacenamiento, y ii) un esquema de entrega.

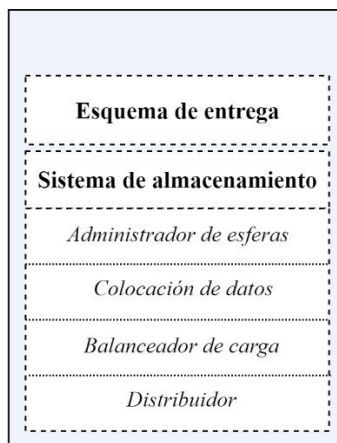


Figura 3 Representación conceptual de una estructura de almacenamiento

El sistema de almacenamiento se encarga de gestionar la colocación de los datos a través de los recursos de almacenamiento de una organización. En este contexto, el método de colocación de datos que se ha implementado se basa en la metáfora de las esferas en los contenedores (*balls-into-bins*) [11], donde los datos se gestionan como esferas que se asignan en un conjunto de contenedores (recursos de almacenamiento). Para lograr este objetivo, el sistema de almacenamiento implementa un gestor de esferas encargado de gestionar los

metadatos de los datos que llegan a la estructura de almacenamiento. Los datos se entregan a un sistema de colocación de datos, que llama a un servicio de balanceo de carga basado en el algoritmo de dos opciones (*two choices*) [6] [11], para crear una distribución justa de los segmentos entre las ubicaciones de almacenamiento. Un distribuidor se encarga de asignar los segmentos producidos en las ubicaciones de almacenamiento.

El esquema de distribución permite el intercambio de datos entre diferentes estructuras de almacenamiento desplegadas en diferentes entornos (es decir, borde, niebla o nube, Edge, Fog y Cloud en inglés). Estos esquemas imponen el cumplimiento de los requerimientos no funcionales (por ejemplo, la disponibilidad y la eficiencia) en la gestión de datos impuestos por los gobiernos y las organizaciones para la gestión de datos sensibles (i.e., NIST [13], COBIT5 [14], ISO27001-7 [15], y normas mexicanas de gestión de datos [16] [17]). Los esquemas de entrega se construyen y gestionan como nano servicios NFR basados en el algoritmo de dispersión de la información (IDA) para la tolerancia a fallos, esquemas de preparación para resolver problemas de seguridad (por ejemplo, la verificación de la integridad usando SHA3), y patrones paralelos para mejorar la eficiencia de los componentes de las estructuras de almacenamiento.

El nano servicio de confiabilidad codifica los datos utilizando el conocido algoritmo IDA [11], [12]. IDA divide un archivo de tamaño $|F|$ en n segmentos redundantes. Los segmentos se distribuyen en diferentes nodos de almacenamiento, y el archivo original puede recuperarse con m segmentos, donde $m < n$. En este sentido, un sistema de almacenamiento puede soportar $n - m$ fallos. Además, se aplican nano servicios de seguridad a los datos para verificar su integridad (por ejemplo, usando un algoritmo de hash), mecanismos de control de acceso para permitir que sólo los usuarios autorizados accedan a los datos (por ejemplo, utilizando técnicas de sobres digitales), y confidencialidad mediante el cifrado de los datos.

Los componentes de la estructura de almacenamiento, incluidos los nano servicios NFR, pueden desplegarse como patrones paralelos para mejorar la eficiencia en el procesamiento de los datos. Estos patrones paralelos se implementan como patrones de

contenedores virtuales y pueden configurarse en cada esquema de entrega. Estos patrones son descritos a profundidad en el servicio de Zamná en la Sección 2.1.

Creación de sistemas de almacenamiento P2P sin servidor en una malla para el almacenamiento de datos

Una arquitectura de malla para el almacenamiento de datos permite construir un sistema de almacenamiento seguro y confiable en forma de sistema P2P sin servidores. La idea básica es ofrecer a las organizaciones la posibilidad de construir soluciones de almacenamiento basadas en sus recursos disponibles en cualquiera de los recursos locales, la niebla o la nube (*Edge, Fog o Cloud*). Además, las organizaciones pueden elegir los parámetros a alto nivel (es decir, el número de nodos de almacenamiento necesarios para una solución edge-fog-cloud determinada) y el servicio de SeRSS de Painal construye automáticamente el sistema de almacenamiento P2P sin servidor.

En este contexto, la infraestructura de la organización se gestiona como una malla de recursos. La **Error! Reference source not found.** muestra una representación conceptual de las soluciones de almacenamiento desplegadas con esta arquitectura de malla. Como se puede observar, sobre la malla se despliegan diferentes soluciones de almacenamiento. Las soluciones de almacenamiento se construyeron desplegando un conjunto de estructuras de almacenamiento (SS, por sus siglas en inglés) como P2P. Estas soluciones de almacenamiento utilizan tablas hash distribuidas (DHT, por sus siglas en inglés) de Chord. Como se puede observar en la **Error! Reference source not found.**, se pueden desplegar diferentes soluciones sobre la misma malla de recursos. Estas soluciones pueden diferir entre sí en cuanto al número de nodos de la estructura, los requisitos no funcionales gestionados por las estructuras de almacenamiento y los recursos utilizados.

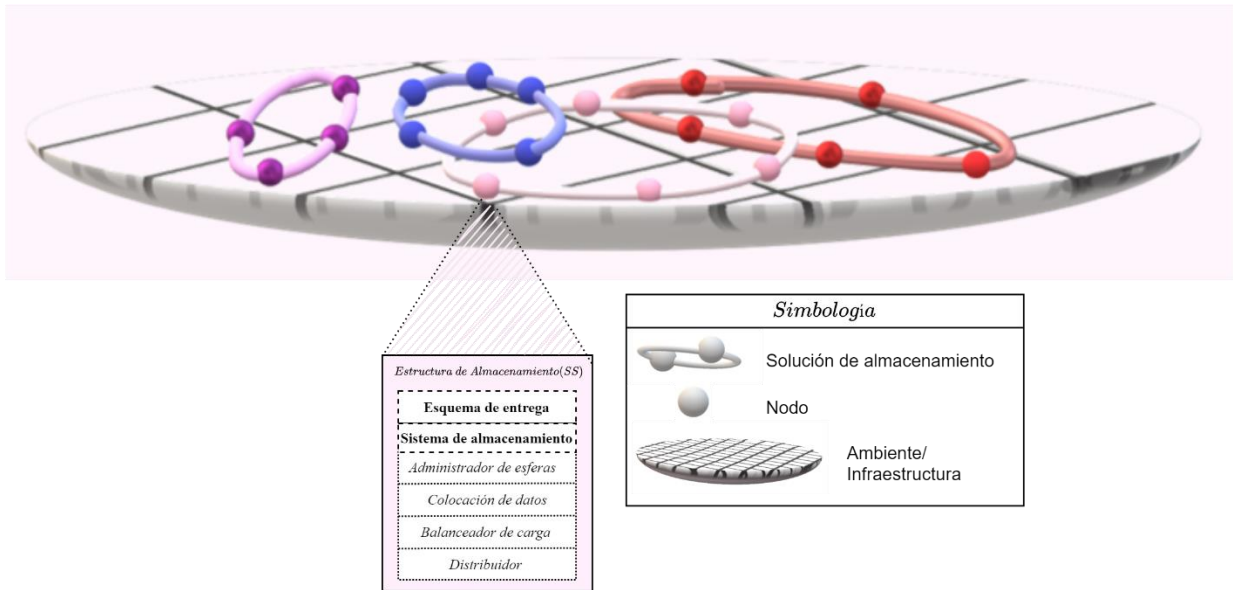


Figura 4 Representación conceptual de diferentes mallas de almacenamiento utilizadas para gestionar datos de manera confiable

La malla de recursos se construye partiendo los nodos de almacenamiento físico (ps , por sus siglas en inglés) en un conjunto de particiones virtuales (vp , por sus siglas en inglés). Estas particiones virtuales son básicamente divisiones lógicas de los nodos de almacenamiento con una porción limitada de los recursos (memoria, CPU y capacidad de almacenamiento) de la infraestructura física (ps). La Figura 5 muestra un ejemplo de malla construida dividiendo cada elemento de un conjunto de nodos de almacenamiento (véase ps_1, ps_2, \dots, ps_p en el eje horizontal x de la Figura 5) en q particiones virtuales (véase vp_1, vp_2, \dots, vp_q en el eje vertical y de la Figura 5). En esta arquitectura, la solución se despliega en una malla de nodos de almacenamiento (M). Esta M está compuesta por un conjunto de nodos de almacenamiento físico ps (eje horizontal x de la malla), donde cada ps está dividido en q particiones denominadas almacenamiento virtual vp (eje vertical y de la malla) y se define como la siguiente malla:

$$M = \begin{bmatrix} ps_1vp_1 & \dots & ps_pvp_1 \\ \vdots & \ddots & \vdots \\ ps_1vp_q & \dots & ps_pvp_q \end{bmatrix}$$

Figura 5 Ejemplo de malla de almacenamiento

Donde q es el número de particiones virtuales en los nodos de almacenamiento, y p es el número de nodos de almacenamiento físicos. Así, el número de recursos disponibles en la

mallla (M) es $p \times q$. Cada celda de la mallla es un recurso de almacenamiento disponible, donde un nodo de almacenamiento virtual (SN, por sus siglas en inglés) puede desplegarse como un contenedor virtual (VC, por sus siglas en inglés), que contiene una estructura de almacenamiento. Un nodo SN en la mallla puede adoptar cualquiera de los siguientes roles:

- Rol de servicio (ser): para atender las solicitudes realizadas por usuarios finales.
- Rol de gestor (mgr): se encarga de ejecutar las tareas de almacenamiento (asignación, localización y equilibrio de carga) y de añadir los requisitos no funcionales a los datos.
- Rol de nodo (node): Se encarga de preservar datos.

Tenga en cuenta que al menos un servicio ser y un gestor mgr deben estar asignados en la mallla. La selección de los nodos ser y mgr son realizados durante el diseño y construcción.

Descripción de los componentes utilizados para el desarrollo del prototipo

La arquitectura de mallla y los componentes de la estructura de almacenamiento se implementaron como un prototipo desarrollado principalmente en lenguaje de programación C. El componente IDA utilizado en la estructura de almacenamiento también se ha implementado utilizando el lenguaje de programación C. El intercambio de datos entre los nodos de almacenamiento virtual se realiza a través de una red de entrega de contenidos del servicio de publicación/suscripción de Painal. En la mallla, los nodos de almacenamiento virtual de una solución de almacenamiento se gestionan como una red P2P implementada mediante el protocolo Chord implementado en lenguaje Python. Los nodos de almacenamiento virtual se despliegan como contenedores virtuales utilizando la plataforma de contenedores Docker. Estos contenedores virtuales añaden portabilidad a los nodos, ya que los componentes del almacenamiento, la estructura se encapsula en el contenedor con sus dependencias (es decir, bibliotecas, paquetes, SO, variables de entorno), lo que permite el despliegue de los nodos en diferentes infraestructuras.

Nota: Para más información acerca del servicio para la construcción de soluciones de almacenamiento costo-beneficio, ver el artículo "**SeRSS: a storage mesh architecture to build serverless reliable storage services**" en el Anexo B.2.

2.3. Servicio de publicación/suscripción (pub/sub) para el manejo de catálogos, fuentes y repositorios.

En esta sección, se describe el servicio publicación/suscripción (pub/sub) sobre un almacenamiento en la nube diversificado. El servicio de pub/sub de Painal divide la entrega de contenidos en dos capas. La primera capa se basa en patrones de publicación-suscripción que permiten devolver el control de los metadatos al propietario del contenido. Esta capa establece los siguientes roles (i) Editores, usuarios que producen nuevos contenidos, (ii) Usuarios finales, clientes externos que se suscriben a los contenidos, y (iii) Editores/administradores, usuarios de la organización encargados de aceptar/rechazar tanto las publicaciones como las suscripciones. Un esquema de gestión de recursos permite a las organizaciones establecer el control de acceso necesario para mantener los flujos de metadatos, así como el flujo de los patrones pub/sub entre editores y usuarios finales. La segunda capa se basa en la dispersión de la información [12] sobre una plataforma de almacenamiento multi-nube con la que el servicio de pub/sub consigue un uso eficiente del espacio de almacenamiento y una alta fiabilidad. En esta capa, la dispersión se realiza en el lado del editor mediante el uso de mecanismos resistentes llamados flujos de trabajo de entrega, que dividen los contenidos en un conjunto de bloques redundantes y anónimos (utilizando procesos descritos en Zamná) que se distribuyen en múltiples proveedores de almacenamiento. De esta manera se garantiza que un determinado proveedor no recibe suficientes bloques para reconstruir el contenido original. Por tanto, la forma de reconstruir el contenido original se mantiene en el lado del editor. Los usuarios finales se encargan de obtener los contenidos publicados mediante flujos de trabajo de recuperación, que recuperan un subconjunto de bloques y reconstruyen los contenidos en el lado del usuario final. Esto significa que los flujos de trabajo de recuperación pueden obtener contenidos incluso cuando algunas ubicaciones de almacenamiento en la nube no están disponibles. Como resultado, se reducen los riesgos de la dependencia del proveedor y permite a la organización externalizar el almacenamiento de contenidos de forma controlada. La Figura 6 muestra las capas de metadatos y de flujo de contenidos de la superposición del servicio de pub/sub y cómo se lleva a cabo la entrega de contenidos como un esquema de colaboración. La Figura 6 también muestra un ejemplo en el que un editor envía una

publicación de contenido $|C|$ a la capa de flujo de metadatos (etiqueta *Pub*). Cuando un editor de la organización autoriza la publicación de este contenido, el editor se encarga de dispersarlo a múltiples ubicaciones de almacenamiento en la nube mediante un flujo de trabajo de entrega (etiqueta *Dy*), que se ejecuta en la computadora personal del editor. El contenido se añade al catálogo y los usuarios finales autorizados ya pueden suscribirse al contenido publicado y dispersado (etiqueta *Sub*). Los usuarios finales con suscripciones autorizadas se encargan de recuperar los contenidos mediante flujos de trabajo de recuperación (etiqueta *Retr*). Esta superposición permite al servicio de pub/sub minimizar los riesgos de las situaciones de bloqueo del proveedor y los escenarios en los que hay una falta de control de los procedimientos críticos.

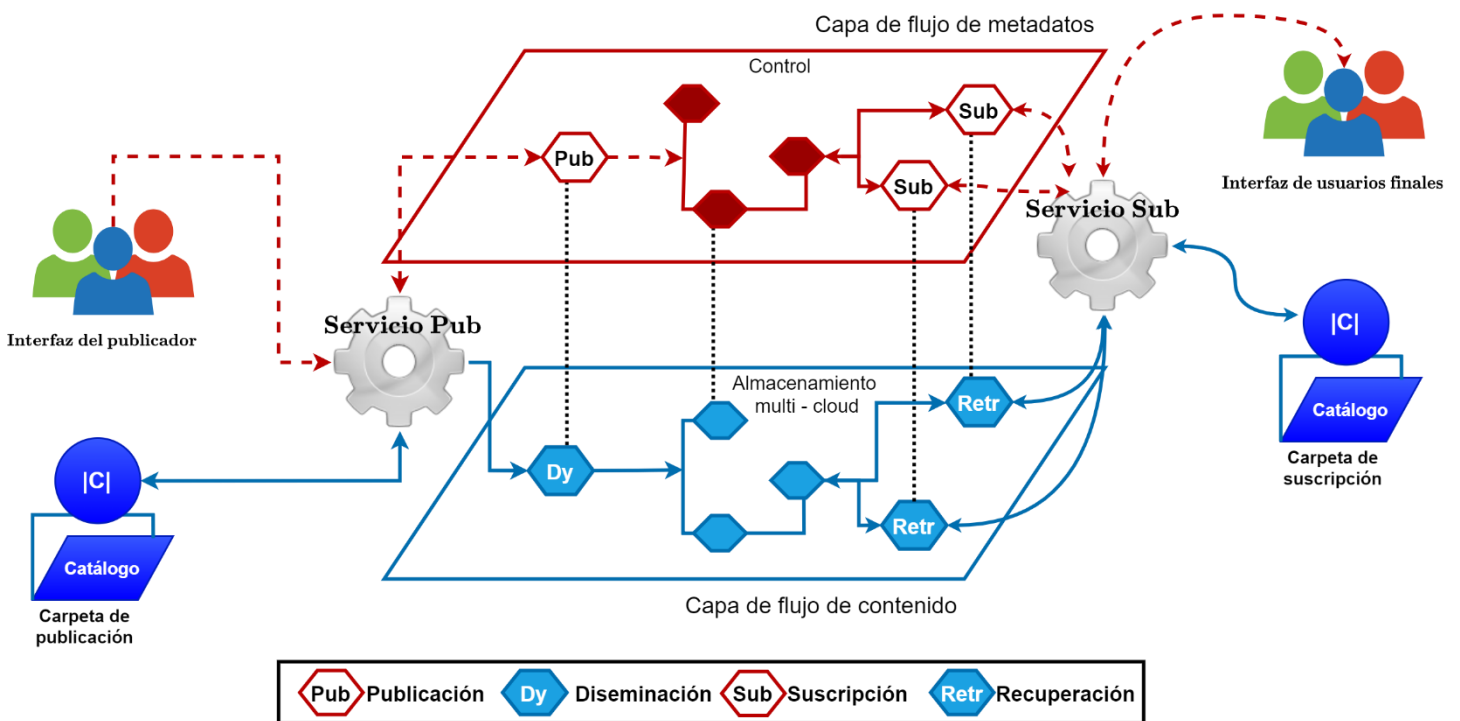


Figura 6 Las capas superpuestas del servicio Pub/Sub.

Las aportaciones del servicio de publicación/suscripción son las siguientes:

1. Entrega colaborativa de contenidos basada en una superposición de publicación/suscripción: Hemos desarrollado un modelo RESTful de publicación/suscripción que permite a los usuarios finales y a los editores colaborar en

el proceso de entrega de contenidos. Simplifica la gestión de los recursos y el control de las etapas de la entrega de contenidos. Los flujos de trabajo generados se basan en el procesamiento multinúcleo y el transporte de flujo continuo para mitigar la sobrecarga producida en el lado del usuario final y del editor.

2. Una estrategia de diversificación basada en la gestión de riesgos para el almacenamiento multi-nube: Esta estrategia virtualiza las múltiples ubicaciones de almacenamiento disponibles en una organización como una plataforma de almacenamiento multi-nube unificada, que es más sencilla de gestionar que las ubicaciones separadas. Un método de colocación de contenidos permite al servicio pub/sub realizar la asignación y localización de contenidos en esta plataforma para evitar poner todos los huevos/contenidos en la misma cesta/ubicación de almacenamiento en la nube. Este método distribuye los contenidos utilizando una política de evaluación de riesgos, que define el nivel de riesgo para cada cesta/ubicación de la plataforma, así como un conjunto de acciones de respuesta para mitigar un conjunto reducido de riesgos expresados por los editores y las organizaciones sobre la gestión de contenidos en el sistema de entrega de contenido (CDS, por sus siglas en inglés).
3. Niveles de enmascaramiento de fallos: se incluye tres niveles de enmascaramiento de fallos. El primer nivel enmascara las interrupciones del servicio de los usuarios finales y los editores. Este nivel se consigue gracias a las técnicas de dispersión de información aplicadas a los flujos de trabajo de entrega y recuperación (descritos en Zamná). El segundo nivel enmascara el desastre en el sitio de la organización desde los usuarios finales. Este nivel se basa en un esquema de federación en el que un conjunto de socios absorbe la carga de la organización durante su interrupción. El último nivel enmascara los efectos secundarios de los retrasos de la geodiversidad de todos los usuarios de del servicio pub/sub mediante el almacenamiento de los contenidos en caché y su traslado a una ubicación cerca de los usuarios finales.

2.3.1. Diseño y componentes del servicio

A continuación, se describen los componentes y subcomponentes del servicio de publicación/suscripción de Painal, así como los roles de los usuarios de la superposición Pub/Sub.

La abstracción del catálogo es el componente de metadatos básico para que las organizaciones establezcan los controles de los flujos de contenidos entre los usuarios que producen nuevos contenidos (editores), los clientes externos que se suscriben a los contenidos (usuarios finales) y los usuarios de la organización encargados de aceptar/rechazar tanto las publicaciones como las suscripciones (editores/administradores). En Painal, los contenidos se asignan y localizan utilizando catálogos; como resultado, solo los contenidos añadidos a un catálogo pueden ser publicados/entregados o suscritos/recuperados. Cuando una organización crea un catálogo, también define los atributos de un conjunto de editores que pueden añadir contenidos al catálogo, así como los grupos de usuarios finales que pueden suscribirse a los contenidos. La Aplicación Cliente es un componente que permite a los editores añadir contenidos fabricados a los catálogos y dispersarlos a la plataforma de almacenamiento multi-nube. También permite a los usuarios finales suscribirse y recuperar los contenidos publicados. Los agentes Pub/Sub y el gestor de flujo de trabajo son los componentes que se encargan de recibir y servir las peticiones de las Aplicaciones Cliente.

El gestor de Painal es el componente encargado de controlar los flujos Pub/Sub en la capa de metadatos y de coordinar el almacenamiento de contenidos en la capa de flujo de contenidos.

La Figura 7 muestra los componentes del servicio pub/sub de Painal, como la aplicación cliente, los agentes, el gestor y los manejadores. La Figura 7 también describe los roles de los editores y los usuarios finales a través de un ejemplo intuitivo que representa los flujos de metadatos y contenidos en la superposición. En este ejemplo, un editor añade el contenido [C] a un catálogo mediante la aplicación *cliente*, que invoca un patrón de publicación (etiqueta *Pub*). Un agente *Pub/Sub* recibe la solicitud y realiza tareas de control de acceso.

El agente envía la solicitud de publicación autorizada al gestor de Painal, que invoca una tarea de asignación. En respuesta, el gestor de Painal envía un mensaje de autorización a los agentes del gestor elegidos para servir las tareas de almacenamiento y envía sus ubicaciones a la *Aplicación cliente* con la que se dispersará de contenido $|C|$. La *Aplicación Cliente* invoca un manejador de flujo de trabajo de entrega que divide el contenido $|C|$ en n bloques ($C = \{c1, \dots, c5\}$ para este ejemplo) y los transporta a la plataforma multi-nube utilizando la información enviada por el servicio pub/sub de Painal (etiqueta Dy). El *Agente Manejador* envía un mensaje de estado completo al gestor de Painal, que actualiza la lista de las publicaciones del catálogo. Esta lista también es actualizada por las *Aplicaciones Cliente* de los usuarios finales autorizados, que ya pueden invocar patrones de suscripción (etiqueta Sub). El *Gestor* aplica el mismo proceso de autorización a los flujos de trabajo de recuperación que el aplicado durante el flujo entrega. El flujo de trabajo de recuperación autorizado (etiqueta $Retr$) solo recupera los k bloques necesarios para construir el contenido $|C|$ en el lado del usuario final ($k = 3$ en este ejemplo, puede ver más detalles sobre el algoritmo de dispersión de información (IDA) en Zamná).

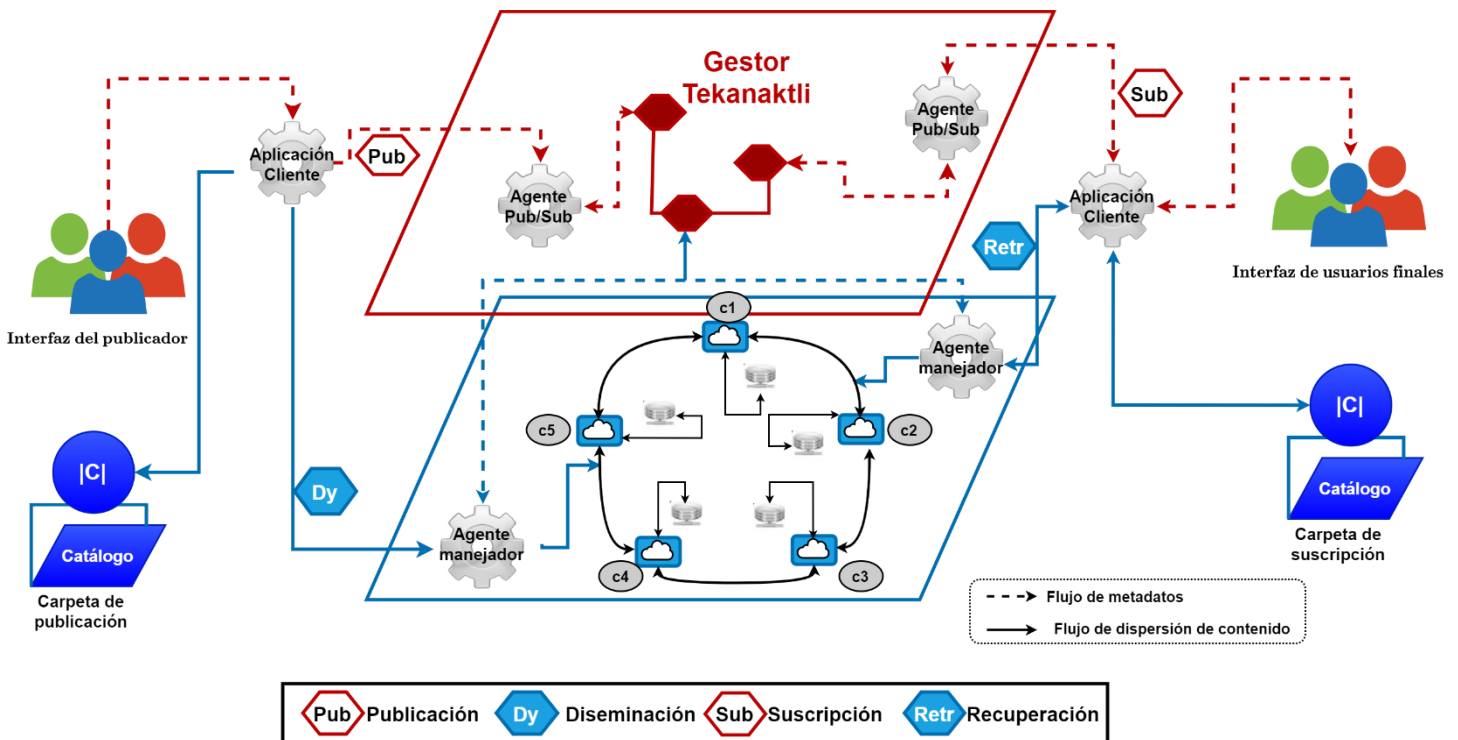


Figura 7 Los componentes del servicio Pub/Sub de Painal

Es importante señalar que el agente distribuidor del servicio pub/sub de Painal no envía todo el contenido [C] a una única ubicación de almacenamiento en la nube. Painal solo envía bloques codificados con nombres anónimos a diferentes nubes, que los gestionan como archivos normales. La plataforma de Painal permite a las organizaciones definir la estrategia de distribución de los n bloques en función de sus políticas organizativas y de los recursos disponibles.

2.3.2. Componentes del servicio de publicación/suscripción: Agentes y clientes.

En el servicio de pub/sub de Painal, una aplicación del tipo cliente puede asumir diferentes roles de usuario (p. ej. Editores, publicadores o usuarios finales) dependiendo de cómo el usuario esté conectado en esa aplicación. Este componente incluye un conjunto de aplicaciones de paneles web para cada tipo de usuario, así como herramientas de Pub/Sub y manejadores de flujo de trabajo. La Figura 8 muestra todos los subcomponentes del servicio de pub/sub de Painal y se utilizará en esta sección para representar los flujos de metadatos entre los componentes y subcomponentes.

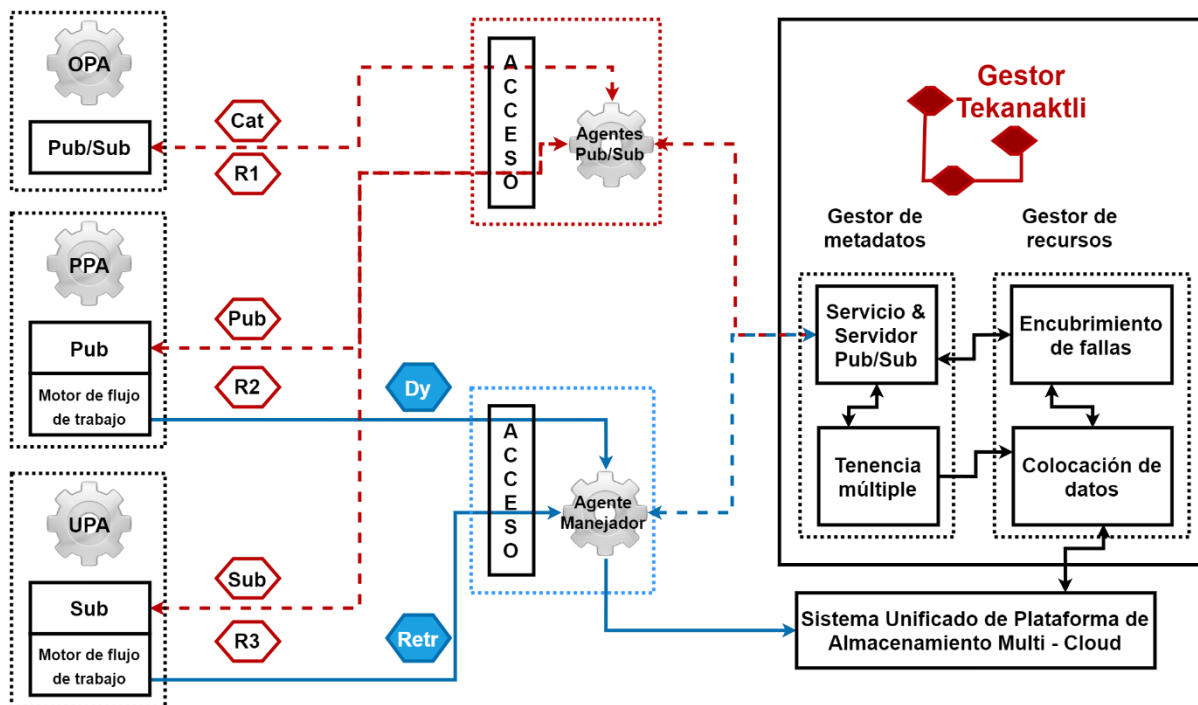


Figura 8 Los componentes y subcomponentes del servicio de Pub/Sub de Painal.

- **Aplicación de tablero de organización (ODA, por sus siglas en inglés):** La aplicación del cliente permite a los editores crear catálogos, asignar editores a un determinado catálogo y definir políticas de suscripción basadas en acuerdos y atributos. Los usuarios finales de diferentes comunidades o segmentos de población pueden acceder a un catálogo determinado utilizando credenciales con sus atributos. El administrador/editor de una organización también puede definir los horarios de publicación por catálogo. La Figura 8 muestra un ejemplo de ODA creando un nuevo catálogo (etiqueta *Cat*) y el conjunto de respuestas enviadas por el agente de Pub/Sub indicando el estado de las publicaciones y suscripciones (etiqueta *R1*).
- **Aplicación de perfil de editor (PPA, por sus siglas en inglés):** Esta aplicación permite a los editores obtener la lista de catálogos en los que están autorizados a añadir contenidos. PPA crea tantas carpetas como catálogos haya en esa lista, por lo que los editores pueden añadir contenidos a los catálogos disponibles utilizando su perfil en el panel de control o simplemente añadiendo los contenidos a las carpetas creadas por PPA. Este componente también incluye un demonio³ que busca nuevos contenidos en estas carpetas, y cuando se detecta un nuevo contenido, el demonio invoca una solicitud de *Pub* (Ver etiqueta *Pub* en la Figura 8). Cuando PPA recibe la autorización del agente *Pub/Sub* (etiqueta *R2*), invoca al gestor de flujo de trabajo para construir un flujo de trabajo de entrega (etiqueta *Dy*). Aunque la organización define el calendario de publicación, el agente manejador dispersa los contenidos a los repositorios de la plataforma de almacenamiento multi-nube o (ULS, por sus siglas en inglés) como política de respaldo. Como se muestra en la Figura 8 el agente manejador envía una notificación al gestor de Painal cuando ha finalizado el proceso de dispersión.
- **Aplicación de perfil de usuario final (UPA, por sus siglas en inglés):** Esta aplicación permite a los usuarios finales suscribirse a contenidos de una lista de catálogos publicados. Cuando un usuario final se suscribe a un contenido determinado (subetiqueta), esta aplicación invoca al manejador de flujo de trabajo para crear un flujo de trabajo de recuperación y obtener el contenido suscrito (subetiqueta). El agente manejador notifica el estado de esta operación al gestor de Painal cuando

³ Un demonio, daemon o *dæmon* (de sus siglas en inglés Disk And Execution MONitor), es un tipo especial de proceso informático que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario (es un proceso no interactivo).

dicha operación finaliza. Este registro permite al gestor de Painal realizar tareas como la notificación a los editores para crear estadísticas de uso y la creación de un informe de rastreo de rutas de ese contenido. Un usuario final puede suscribirse a todo un catálogo y el UPA recuperará automáticamente cada nuevo contenido añadido al catálogo siempre que se autorice la operación (Contenido publicado, contenido suscrito). Los usuarios finales pueden elegir los contenidos de los catálogos mediante un panel de control web. UPA también incluye una lista de contenidos ya suscritos y disponibles en el equipo local.

Cada cliente/agente es provisto por el gestor de Painal con un Token de Autorización válido para enviar solicitudes válidas al servicio Pub/Sub de control de Painal. El gestor de Painal utiliza el Token⁴ de Autorización como acceso de control y como método para establecer las políticas de seguridad.

Gestor de Painal

El gestor de Painal para el servicio de pub/sub incluye gestores de metadatos y de contenidos. El *gestor de metadatos* se encarga de los flujos Pub/Sub entre los principales componentes del servicio de pub/sub de Painal y los clientes/agentes, mientras que el *gestor de recursos* que se encarga de la asignación/ubicación de contenidos y la gestión de recursos.

Gestor de metadatos

El gestor de metadatos incluye los siguientes módulos:

- *Subsistema de servicios*. Este componente se encarga de recibir las solicitudes de tokens de autorización, así como de recibir las publicaciones y los pedidos de suscripción procedentes de los clientes/agentes del servicio de pub/sub de Painal. El

⁴ Token (informática), también llamado componente léxico es una cadena de caracteres que tiene un significado coherente en cierto lenguaje de programación. Token de seguridad, utilizado para facilitar el proceso de autenticación de usuarios.

subsistema registra las transacciones realizadas por los editores, las organizaciones y los usuarios finales (ver la Figura 8).

- *Subsistema de publicación y suscripción.* Este módulo se encarga de las órdenes de publicación y suscripción autorizadas por el subsistema de servicios (ver la Figura 8). Una política basada en atributos de rol en este módulo permite al servicio de pub/sub de Painal aceptar/rechazar órdenes de publicación y suscripción. Este módulo también incluye un servicio de alerta que envía notificaciones a los editores de contenidos y a los administradores de las organizaciones sobre las suscripciones de contenidos. También envía notificaciones al usuario final sobre nuevas publicaciones.
- *Subsistema de tenencia múltiple (multi-tenant).* Este módulo gestiona las propiedades del catálogo y de los contenidos, así como las cuentas de los usuarios utilizando una base de datos. Este módulo asegura que los contenidos de cualquier editor, usuario final o administrador están aislados y permanecen invisibles para otros usuarios. Este módulo envía solicitudes de asignación o localización al gestor de recursos para un determinado contenido asociado a un determinado catálogo (Ver Figura 8).

Este gestor es un componente crítico que se instala en una instancia de la nube situada en una infraestructura de nube privada. La Fig. 3 muestra un ejemplo de los flujos de metadatos entre los agentes, el servicio, los subsistemas Pub/Sub y Multi-tenant. La Figura 8 también muestra cómo el gestor de recursos atiende las solicitudes de asignación/ubicación enviadas por los agentes y autoriza a los gestores de flujo de trabajo a transportar datos a un sistema unificado (ULS). Detallaremos el gestor de recursos y el ULS una vez que el gestor de flujo de trabajo sea descrito en la siguiente sección.

Gestor de flujos de trabajo

Este componente es el responsable de administrar los flujos de trabajo de entrega para transportar los contenidos desde los ordenadores de los editores a la plataforma de almacenamiento multi-cloud. Adicionalmente, permite administrar los flujos de trabajo de recuperación para transportar los contenidos desde la plataforma de almacenamiento a los ordenadores de los usuarios finales. Los flujos de trabajo se basan en tuberías de

procesamiento que incluyen dos fases básicas. La primera fase es la codificación/decodificación de los contenidos, que se basa en algoritmos de información dispersa (IDA) [12] y la segunda etapa es la distribución de los contenidos codificados/codificados, que se basa en el streaming⁵ continuo y paralelo [8]. Este gestor de flujos de trabajo es descrito de manera más profunda en la documentación de Zamná.

Flujo de trabajo de entrega

El flujo de trabajo de entrega ha sido diseñado para que los editores dispersen los contenidos a n ubicaciones de almacenamiento en la nube. Este flujo de trabajo incluye dos flujos de metadatos (publicación y gestión) y un flujo de contenidos (canal de dispersión). El flujo de trabajo comienza en la fase de publicación, cuando la aplicación *PPA* envía una solicitud para entregar un contenido $|C|$ y para obtener un *token* de autorización válido. El flujo de trabajo continúa en la fase de gestión, cuando el gestor de Painal recibe un token de autorización de un editor que muestra el interés de publicar un contenido $|C|$. El gestor acepta/rechaza la publicación y envía una solicitud de ubicaciones de almacenamiento al subsistema de gestión de recursos y espera una respuesta. El gestor produce un token de servicio válido para ULS, determina n URLs relativas que mapean a n ubicaciones de almacenamiento en la nube diferentes. El gestor de Painal responderá al *PPA* con el token de servicio y las n ubicaciones. El último paso ocurre en la fase de entrega, donde el *PPA* activa el gestor de flujo de trabajo en modo de dispersión. Este gestor construye una tubería de dispersión que considera tres etapas. En la etapa inicial, el gestor introduce las localizaciones de almacenamiento (URLs) y el token de servicio como parámetros de entrada, lee el contenido que se va a dispersar, crea tantos procesos como núcleos en el ordenador donde se ejecuta el gestor, y envía esta configuración a la siguiente etapa, la etapa de transformación. En esta etapa, el gestor divide el contenido $|C|$ en n bloques, codifica cada segmento (añadiendo redundancia) y los envía a la última etapa, la etapa de transporte. En esta etapa, se crean n flujos a URLs relativas para que el gestor transfiera los diferentes bloques. En la etapa de transporte, el gestor también se encarga de cerrar los flujos activos cuando termina la codificación y transmisión de los bloques, así como de informar de

⁵ El streaming es lo que le permite consumir contenido en línea sin tener que esperar a que se descargue.

los errores que puedan surgir. El PPA notifica el estado de dispersión al gestor del servicio de pub/sub de Painal que determina el calendario de publicación y envía alertas a los usuarios finales sobre este contenido. La Figura 9 muestra un ejemplo del pipeline de dispersión cuando el gestor utiliza una configuración $n = 5$ (se generan 5 dispersos codificados) y $k = 3$ (solo se requieren 3 dispersos para reconstruir el contenido $|C|$).

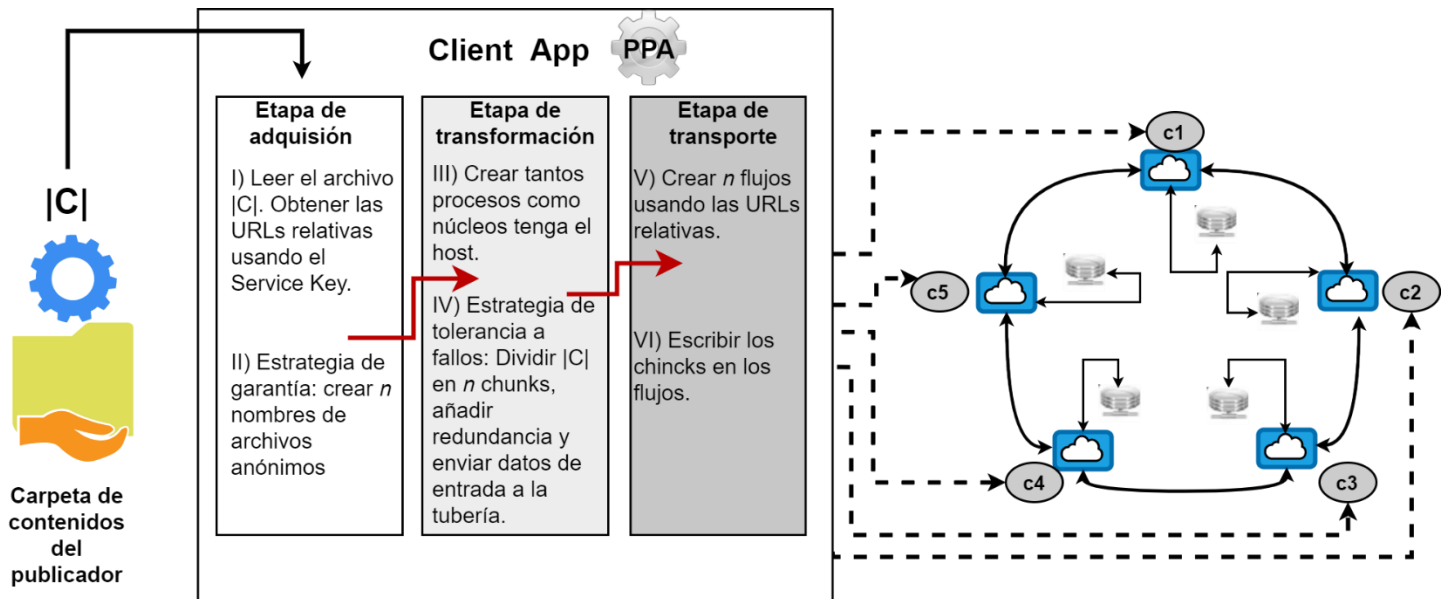


Figura 9 Ejemplo de tubería de dispersión con una configuración $n = 5, k = 3$

Flujo de trabajo de recuperación

El flujo de trabajo de recuperación se ha diseñado para que los usuarios finales recuperen contenidos de las ubicaciones de almacenamiento en la nube mediante un procedimiento de decodificación multihilo (multithreading). Este flujo de trabajo incluye dos flujos de metadatos (procesos de suscripción y gestión) y un flujo de contenidos (canal de recuperación).

Este flujo de trabajo comienza en la fase de suscripción, en la que el usuario final, a través de la aplicación UPA, envía una solicitud de suscripción de contenido $|C|$ al gestor Painal.

En la (fase de gestión), el gestor de Painal verifica el token de autorización, autoriza/rechaza la suscripción para el contenido $|C|$ y envía una solicitud de k ubicaciones al gestor de recursos. El gestor recibe la solicitud y determina las k ubicaciones de almacenamiento en la nube disponibles de las n utilizadas en la asignación de ese contenido, y produce un token de servicio para esta operación válido para el ULS. El gestor de Painal responde a la UPA con

el token de servicio, las k URLs relativas que se asignan a las k ubicaciones de almacenamiento en la nube diferentes, y los métodos necesarios para acceder a ellas.

En la fase de recuperación, se activa un gestor de recuperación en *UPA*, que se encarga de crear una tubería de procesamiento de recuperación que considera tres etapas. En la etapa inicial, el gestor introduce las ubicaciones de almacenamiento (URLs) y el token de servicio como parámetros de entrada, escribe un archivo con el nombre del contenido C (que será recuperado), crea tantos procesos como núcleos en el ordenador donde se ejecuta el gestor, envía todos los ajustes de configuración a la etapa de transformación y espera una respuesta. En la etapa de transformación, el gestor solicita a la etapa de transporte k flujos de transmisión asociados a las URLs relativas (lugares de almacenamiento) y espera la respuesta. La etapa de transporte crea los flujos necesarios, lee los bloques correspondientes y devuelve los resultados a la etapa de transformación. En esta etapa, el gestor reconstruye el contenido $|C|$ utilizando los k bloques, y envía los resultados obtenidos de vuelta a la etapa inicial, donde cada bit recibido se escribe en el archivo asociado al contenido C , utilizando el sistema de archivos local. Finalmente, cuando el proceso de decodificación de C ha terminado, el gestor cierra los flujos activos e informa a *UPA* de los errores encontrados, si los hay. La aplicación *UPA* notifica el estado del proceso de recuperación al gestor del servicio pub/sub de Painal. La Figura 10 muestra un ejemplo de una tubería de procesamiento de recuperación con una configuración $n = 5$ (se generan 5 dispersos codificados) y $k = 3$ (solo se requieren 3 dispersos para reconstruir el contenido $|C|$).

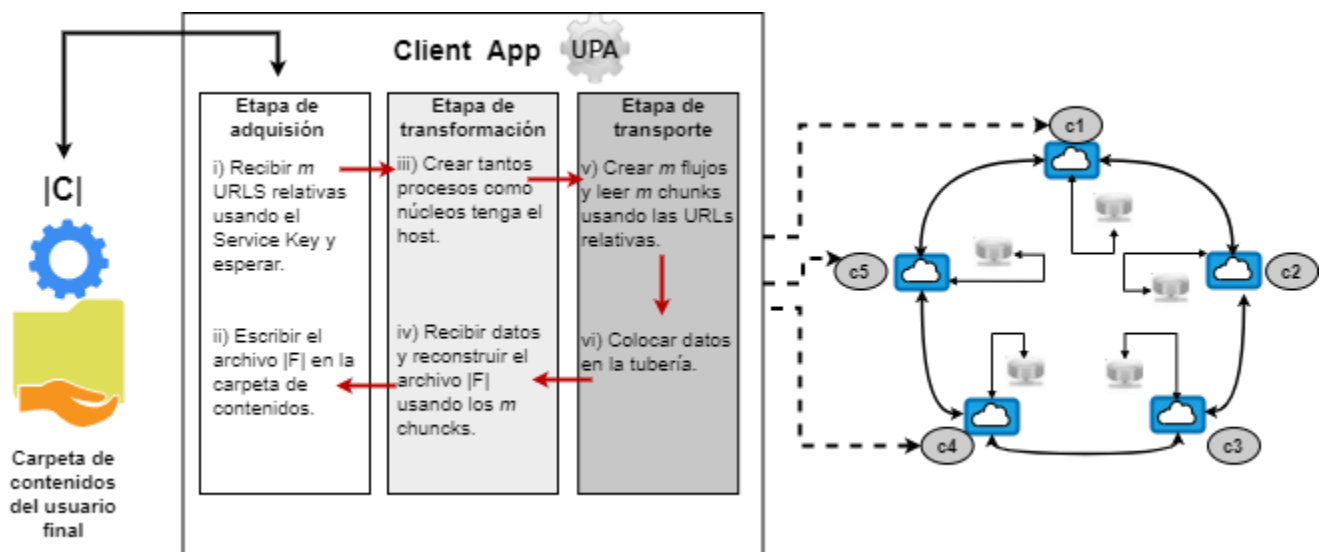


Figura 10 Ejemplo de tubería de recuperación con una configuración $n = 5$, $k = 3$

Nuestra implementación de la etapa de transformación se basa en la tecnología de bloques de construcción de hilos de Intel (TBB) [18] y la etapa de transporte se implementó utilizando las bibliotecas Curl [19] .

La implementación multihilo de nuestros gestores de dispersión/recuperación mejora el rendimiento de las tareas de codificación y decodificación al aprovechar los múltiples núcleos que se encuentran habitualmente en los dispositivos actuales. Esta técnica permite que el gestor reduzca la sobrecarga de codificación, lo que hace factible la introducción de un esquema tolerante a fallos en el lado del editor de contenidos/usuario final. La implementación del proceso de entrega de contenidos como flujo continuo permite al gestor evitar la escritura de bloques en los discos locales.

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 1 . Instituciones participantes en el desarrollo del servicio Zamná.

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.

3.3. Grupo de trabajo intrainstitucional

Tabla 2 . 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. Ricardo Marcelín Jiménez	Responsable del desarrollo de las operaciones del servicio de almacenamiento.	1
	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
	M.C. Hugo German Reyes Anastacio	Diseño de almacenamiento tolerante a fallas para la plataforma	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
	Ing. Catherine Alessandra Torres Charles	Sistemas de preparación de datos médicos para cumplimentar las normas nacionales e internacionales referentes al intercambio seguro de datos en infraestructuras heterogéneas	N

3.4. Grupo de trabajo interinstitucionales

Tabla 3 Equipo de trabajo interinstitucional

Institución	Participante	Líneas de investigación	Nivel SNI
UAM	Dr. Ricardo Marcelín Jiménez	Sistemas de almacenamiento.	1
CICESE	Dr. Andrei Tchernykh	Diseño de criptosistemas. Desarrollo de búsquedas cifradas para sistemas PACS.	2

4. Estatus de Painal

Entregable	Producto	Estado					Etapa		Comprometido	NMT
		Diseño	En desarrollo	En Evaluación	Prototipado	En Producción	1	2		
	Un servicio que permita el manejo, carga y descarga de servicios desde un catálogo/repositorio de servicios.	*	*	*	*		✓		Si	6
	Mecanismo que permita brindar usabilidad costo beneficio del almacenamiento y transporte de datos.	*	*	*	*		✓		Si	6
	Un servicio basado en el modelo publicación/suscripción (pub/sub) para el manejo de catálogos, fuentes y repositorios de diferentes tipos de datos clínicos.	*	*	*	*		✓		Si	6
	Red de distribución de contenidos para el intercambio de datos en el perímetro niebla nube.	*	*	*	*			✓	No	4
	Una red de distribución de contenido que permita la carga, intercambio y descarga automática de datos.	*	*	*	*			✓	No	6

Simbología	
*	Terminado
X	En proceso
NMT	Nivel de Maduración Tecnológica

5. Anexos

Artículos

I. A Federated Content Distribution System to Build Health Data Synchronization Services

In organizational environments, such as in hospitals, data have to be processed, preserved, and shared with other organizations in a cost-efficient manner. Moreover, organizations have to accomplish different mandatory non-functional requirements imposed by the laws, protocols, and norms of each country. In this context, this paper presents a Federated Content Distribution System to build infrastructure-agnostic health data synchronization services. In this federation, each hospital manages local and federated services based on a pub/sub model. The local services manage users and contents (i.e., medical imagery) inside the hospital, whereas federated services allow the cooperation of different hospitals sharing resources and data. Data preparation schemes were implemented to add non-functional requirements to data. Moreover, data published in the content distribution system are automatically synchronized to all users subscribed to the catalog where the content was published.

[Leer artículo completo...](#)

II. SeRSS: a storage mesh architecture to build serverless reliable storage services

Cloud storage has been the solution for organizations to manage the exponential growth of data observed over the past few years. However, end-users still suffer from side-effects of cloud service outages, which particularly affect edge-fog-cloud environments. This paper presents SeRSS, a storage mesh architecture to create and operate reliable, configurable, and flexible serverless storage services for heterogeneous infrastructures. SeRSS manages storage nodes as independent and portable storage structures deployed on virtual containers. In SeRSS mesh, the storage nodes include systems such as delivery/retrieval of files, load balancing, data distribution, and placement systems, as well as non-functional nanoservices to add reliability, security, and efficiency properties to the data. The end-users, of SeRSS, can on-the-fly create and on-demand operate reliable storage services by varying the parameters of the storage structures at a high level, depending on both their capacity and performance needs. SeRSS uses the end-users parameters to automatically build a P2P network of serverless decentralized storage managed as a service and deployable on multiple infrastructures. A case study was conducted based on-the-fly building of storage services to manage medical imagery. The experimental evaluation revealed the efficiency of SeRSS to manage and store data in a reliable manner in heterogeneous infrastructures.

[Leer artículo completo...](#)

Posters

Poster cualitativo de Painal

Tekanaktli:

Servicio para el transporte y almacenamiento de datos médicos

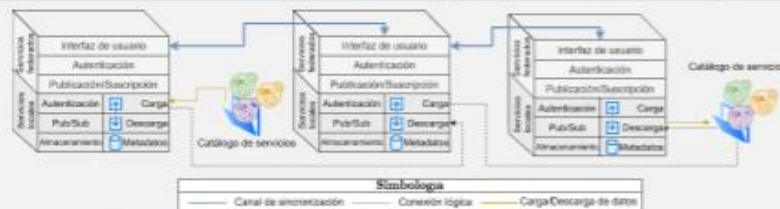


Tekanaktli

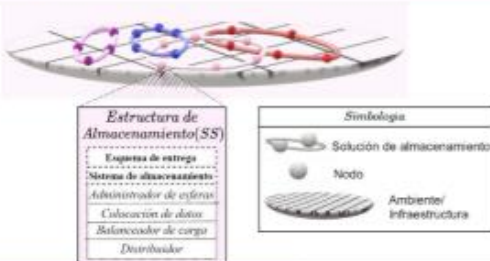
Servicio para el manejo, carga y descarga de servicios por medio de catálogos

Cuenta con:

- Mecanismos para la creación de catálogos de servicios
- Esquemas de publicación y suscripción a los catálogos



Mecanismo de usabilidad costo-beneficio para el almacenamiento y transporte de datos.



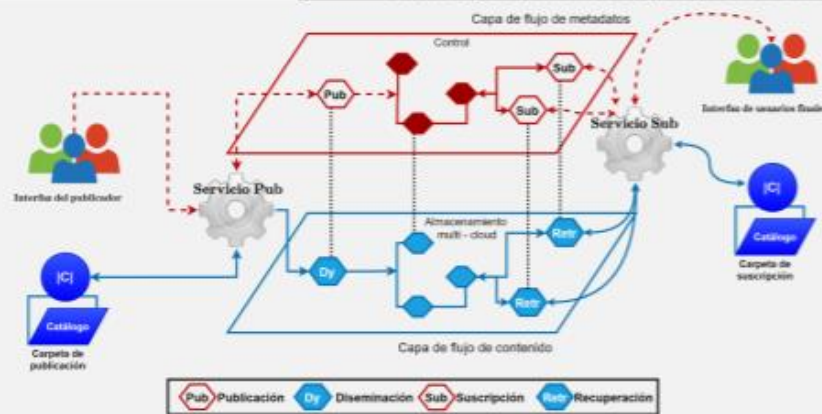
Permite:

- Generar una malla para el almacenamiento de datos
- Proveer eficiencia en la utilización de recursos durante le ejecución de las aplicaciones de almacenamiento
- Reducir el consumo de espacio utilizado y los costos de almacenamiento

Servicio de publicación/suscripción para el manejo de catálogos, fuentes y repositorios.

Permite:

- Crear catálogos de datos
- Restringir el acceso a los datos de los catálogos
- Utilizar aplicaciones encargadas de la carga y descarga
- Realizar el traslado seguro de datos entre entidades
- Utilizar los recursos disponibles en la infraestructura utilizada



6. Referencias

- [1] M. 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 Measur,» *Allied Market Research*, p. 124, 2016.
- [2] C.-P. Deng, T. Wang, T. SH Teo y Q. Song, «Organizational agility through outsourcing: Roles of it alignment, cloud computing and knowledge transfer,» *International Journal of Information Management*, 2021.
- [3] H. S. Gunawi, M. Hao, R. O. Suminto, A. Laksono, A. D. Satria, J. Adityatama y K. J. Eliazar, «Why does the cloud stop computing?: Lessons from hundreds of service outages.,» *In Proceedings of the Seventh ACM Symposium on Cloud Computing,*, 2016.
- [4] J. Opara-Martins, R. Sahandi y F. Tian, «Critical review of vendor lock-in and its impact on adoption of cloud computing,» *In International Conference on Information Society (i-Society 2014)*, 2014.
- [5] D. Reinsel-John y G.-J. Rydning, «The digitization of the world from edge to core,» *Framingham: International Data Corporation*, 2018.
- [6] R. Beraldi, H. Alnuweiri y A. Mtibaa, «A power-of-two choices based algorithm for fog computing,» *IEEE Transactions on Cloud Computing*, vol. 8, p. 698–709, 2018.
- [7] D. D. Sánchez-Gallegos, D. Carrizales-Espinoza, H. G. Reyes-Anastacio, J. L. Gonzalez-Compean, J. Carretero, M. Morales-Sandoval y A. Galaviz-Mosqueda, «From the edge to the cloud: A continuous delivery and preparation model for processing big IoT data,» *Simulation Modelling Practice and Theory*, vol. 105, p. 102136, 2020.
- [8] D. Higuero, J. M. Tirado, J. Carretero, F. Félix y A. de La Fuente, «HIDDRA: a highly independent data distribution and retrieval architecture for space observation missions,» *Astrophysics and Space Science*, vol. 321, p. 169–175, 2009.
- [9] J. L. Gonzalez, J. C. Perez, V. J. Sosa-Sosa, L. M. Sanchez y B. Bergua, «SkyCDS: A resilient content delivery service based on diversified cloud storage,» *Simulation Modelling Practice and Theory*, vol. 54, p. 64–85, 2015.
- [10] J. L. Gonzalez-Compean, V. J. Sosa-Sosa, A. Diaz-Perez, J. Carretero y R. Marcellin-Jimenez, «FedIDS: a federated cloud storage architecture and satellite image delivery service for building dependable geospatial platforms,» *International journal of digital earth*, vol. 11, p. 730–751, 2018.
- [11] P. Morales-Ferreira, M. Santiago-Duran, C. Gaytan-Diaz, J. L. Gonzalez-Compean, V. J. Sosa-Sosa y I. Lopez-Arevalo, «A data distribution service for cloud and containerized storage based on information dispersal,» de *2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, 2018.
- [12] R. Marcellín-Jimenez, J. L. Ramírez-Ortiz, E. R. De La Colina, M. Pascoe-Chalke y J. L. Gonzalez-Compean, «On the complexity and performance of the information dispersal algorithm,» *IEEE Access*, vol. 8, p. 159284–159290, 2020.
- [13] A. Ibrahim, C. Valli, I. McAteer y J. Chaudhry, «A security review of local government using NIST CSF: a case study,» *The Journal of Supercomputing*, vol. 74, p. 5171–5186, 2018.
- [14] Y. Durachman, Y. Chairunnisa, D. Soetarno, A. Setiawan y F. Mintarsih, «IT security governance evaluation with use of COBIT 5 framework: A case study on UIN Syarif Hidayatullah library information system,» de *2017 5th International Conference on Cyber and IT Service Management (CITSM)*, 2017.

- [15] S. Choubey y A. Bhargava, «Significance of ISO/IEC 27001 in the Implementation of Governance, Risk and Compliance,» *International Journal of Scientific Research in Network Security and Communication*, vol. 6, p. 30–33, 2018.
- [16] S. E. C. T. O. R. SALUD y D. E. SUBSECRETARIA DE PREVENCIÓN Y PROMOCIÓN, «NORMA Oficial Mexicana NOM-024-SSA3-2010, Que establece los objetivos funcionales y funcionalidades que deberán observar los productos de Sistemas de Expediente Clínico Electrónico para garantizar la interoperabilidad, procesamiento, interpretación, confidencialidad, seguridad y uso de estándares y catálogos de la información de los registros electrónicos en salud.».
- [17] D. E. SERVICIOS DE SALUD Y SERVICIOS, S. A. L. U. D. DE CHIHUAHUA y P. E. T. R. O. L. E. O. S. MEXICANOS, «NORMA OFICIAL MEXICANA NOM-004-SSA3-2012, DEL EXPEDIENTE CLINICO PREFACIO En la elaboración de esta norma participaron: CONSEJO DE SALUBRIDAD GENERAL SECRETARIA DE SALUD».
- [18] A. Ghuloum, «Future Proof Data Parallel Algorithms and Software on Intel Multicore Architecture,» *Intel Technology Journal*, vol. 11, n^o 4, 2007.
- [19] «libcurl,» The multiprotocol file transfer library, [En línea]. Available: <http://curl.haxx.se/libcurl>. [Último acceso: 10 Nov 2014].
- [20] F. d. A. L. Fuentes, J. M. Almanza, R. Marcelín-Jiménez y B. Velázquez-Méndez, «Efficient content distribution and storage p2p system based on information dispersal,» de *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2019.

