

Sistema de Integración Dinámica de Mensajes en Tiempo Real para Producción Audiovisual: Downstream Keyer Mejorado para Superposición de Contenido Meta en Transmisiones multimedia

Poma Calle Oscar Marcelo
marcelo.pc.dev@gmail.com
Instituto de Electrónica Aplicada

Resumen - En este artículo se introduce un novedoso sistema de Integración Dinámica de Mensajes en Tiempo Real, centrado en potenciar la producción audiovisual mediante un Downstream Keyer, con un énfasis particular en la gestión del contenido meta durante las transmisiones multimedia. El enfoque principal se dirige hacia la optimización de la superposición de mensajes en tiempo real, mejorando la eficiencia y dinamismo en la integración de estos elementos. Se destacan las características clave, resaltando su flexibilidad y rendimiento mejorado. A través de implementaciones y resultados experimentales, se evidencia la efectividad del sistema, respaldando su viabilidad y capacidad para elevar la calidad de las transmisiones multimedia. Aunque la plataforma Meta, presenta una gran cantidad de servicios multimedia, se destaca la relevancia de este sistema en la gestión específica de contenido a través de WhatsApp Business, posicionándolo como una herramienta sólida y versátil para la creación dinámica de mensajes en tiempo real en el contexto de la producción audiovisual.

Palabras clave – Downstream Keyer Mejorado, Meta, WhatsApp Business, Tiempo real.

I. INTRODUCCIÓN

La producción audiovisual contemporánea ha experimentado una transformación significativa, marcada por una creciente demanda de contenido dinámico y atractivo. En este contexto, la capacidad de integrar mensajes en tiempo real se ha convertido en un factor determinante para la relevancia y el impacto de las transmisiones multimedia. Este artículo aborda la evolución de la producción audiovisual, centrándose en el desarrollo de un sistema pionero de Downstream Keyer, diseñado para potenciar la superposición de contenido meta durante las transmisiones en vivo.

El Downstream Keyer tradicional ha sido una herramienta valiosa para la superposición de elementos visuales en la señal de video principal. Sin embargo, la necesidad de una

integración más dinámica y eficiente de mensajes en tiempo real ha generado la motivación para mejorar esta tecnología.

Donde se propone una solución innovadora que no solo optimiza la capacidad de superposición, sino que también introduce flexibilidad y adaptabilidad a las demandas cambiantes de la producción contemporánea.

En el corazón de esta propuesta se encuentra la creación de un sistema de Integración Dinámica de Mensajes en Tiempo Real, que redefine los límites de la superposición de contenido. La capacidad de integrar mensajes, gráficos y datos adicionales de manera dinámica durante la transmisión en vivo no solo enriquece la experiencia del espectador, sino que también proporciona a los productores una herramienta versátil para la creación instantánea y personalizada de contenido.

También se abarca una revisión de las características fundamentales del Downstream Keyer, respaldada por la implementación práctica y los resultados experimentales. Se lleva a cabo una evaluación detallada de la eficacia y la aplicabilidad de este sistema en un escenario de producción audiovisual real.

El propósito central es contribuir al avance continuo de las tecnologías que impulsan la narrativa visual en constante evolución, consolidando así la creación de un sistema integral que aproveche estas innovaciones en beneficio de la producción moderna.

II. REVISION DE LITERATURA

La convergencia de tecnologías para mejorar la producción audiovisual en tiempo real ha sido un área de investigación dinámica y significativa en la literatura científica y tecnológica. En el ámbito de la captura de mensajes en plataformas de mensajería [1], resaltan la importancia de establecer conexiones seguras y eficientes mediante Webhooks, enfocándose en la

transmisión oportuna de información en entornos de mensajería instantánea.

La producción audiovisual ha experimentado una transformación notable con tecnologías como el Downstream Keyer [2], profundizan en las características fundamentales de estos sistemas, destacando la flexibilidad y el rendimiento mejorado. Este enfoque no solo enriquece la narrativa visual, sino que también contribuye a una integración más eficiente de mensajes durante las transmisiones, mejorando la interactividad del contenido. Sin embargo, la creciente demanda de transmisiones multimedia más interactivas y personalizadas ha generado la necesidad de una integración más dinámica de mensajes en tiempo real.

En el ámbito de la entrega de contenido en tiempo real se exploran cómo la convergencia de tecnologías ha permitido una optimización significativa [3]. Este estudio, que no menciona tecnologías específicas, destaca la importancia de estrategias para mejorar la eficiencia en la entrega de contenido en tiempo real.

Además, en el contexto de la integración en entornos televisivos, subrayan la necesidad de una transición sin problemas y la compatibilidad con sistemas existentes para garantizar una implementación exitosa [4].

Además, se destaca la relevancia de la plataforma Meta y su amplia gama de servicios multimedia. Aunque la plataforma ofrece diversas opciones, la atención se centra en la gestión específica de contenido a través de WhatsApp Business. La literatura existente subraya la importancia de las plataformas de mensajería instantánea en la interacción directa con la audiencia, y cómo esta interacción puede potenciarse aún más mediante un sistema como el propuesto en este artículo [5].

III. METODOLOGIA

La metodología adoptada en el desarrollo de la plataforma, se caracteriza por un enfoque iterativo que permite una adaptación constante a las cambiantes demandas de la producción audiovisual. Este método se distingue por su capacidad para evolucionar dinámicamente a lo largo del ciclo de desarrollo, incorporando retroalimentación continua de los usuarios finales y ajustando la implementación según las necesidades emergentes. A través de fases iterativas que abarcan diseño, implementación, prueba y evaluación, la metodología asegura la solidez técnica del sistema, alineándolo con precisión a las expectativas del usuario y las tendencias del entorno audiovisual en constante evolución, en concordancia con las prácticas de desarrollo ágil.

1. Arquitectura y diseño del sistema.

El sistema se estructura sobre una arquitectura modular y escalable además de específica para el proceso de desarrollo, que prioriza la eficiencia en la superposición de contenido meta

(Whatsapp Bussiness) durante las transmisiones multimedia en tiempo real.

A. Arquitectura Cliente-Servidor

La elección de la arquitectura de desarrollo para este sistema se orienta hacia el modelo cliente-servidor, una estructura consolidada que divide las funciones del sistema entre dos componentes principales: el cliente y el servidor. Esta decisión se fundamenta en la necesidad de gestionar de manera eficiente la interacción entre la interfaz de usuario construida con el cliente y la lógica del lado del servidor.

Cliente.

La parte del cliente se dedica a proporcionar una experiencia de usuario interactiva y visualmente atractiva. La interfaz de usuario está diseñada para permitir la presentación dinámica de la información y facilitar la interacción fluida del usuario con el sistema.

Servidor.

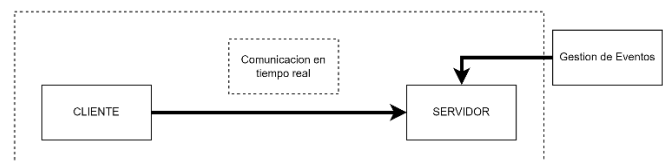
La lógica del servidor se encarga de procesar la lógica central de la aplicación y gestionar la comunicación con bases de datos y servicios externos. La ejecución eficiente y la capacidad de manejar múltiples conexiones simultáneas son esenciales para garantizar respuestas en tiempo real.

Comunicación en Tiempo Real.

La comunicación bidireccional en tiempo real se logra a través de la implementación de WebSockets. Esta tecnología permite la transmisión instantánea de mensajes entre el cliente y el servidor, facilitando la actualización inmediata de la información durante las transmisiones multimedia.

Gestión de Eventos.

La integración de Webhooks fortalece la capacidad del sistema para reaccionar de manera automática ante eventos específicos, introduciendo un nivel adicional de dinamismo y actualización en tiempo real.



B. Componentes Principales

Integración de Mensajes en Tiempo Real.

Este módulo se encarga de gestionar la entrada y salida de mensajes en tiempo real, permitiendo la integración dinámica

de contenido adicional durante las transmisiones. Su funcionalidad se amplía para interactuar de manera más eficiente con las notificaciones provenientes de Webhooks asegurando una integración sin fisuras de eventos de WhatsApp relevantes.

Webhooks de Meta.

Componente principal de sistema que permite recibir notificaciones HTTP en tiempo real sobre todo en los cambios relevantes, como modificaciones en los mensajes de WhatsApp o comentarios en páginas de Facebook asociadas. Este componente se integra con la Integración de Mensajes en Tiempo Real, el Gestor de Gráficos y el Gestor de Lógica para asegurar una respuesta fluida y contextualizada ante eventos perteneciente a WhatsApp Business específicamente.

Gestor de Gráficos.

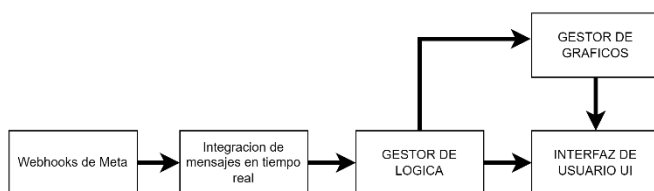
Responsable del procesamiento y almacenamiento de gráficos, texto y datos adicionales, enriqueciendo la superposición en la señal principal. Se optimiza para incorporar datos provenientes de Webhooks, permitiendo una superposición contextualizada en tiempo real basada en cambios específicos en la gráfica social.

Gestor de Lógica (Backend).

Este componente alberga la lógica central del sistema. Coordina las acciones entre la Integración de Mensajes en Tiempo Real y el Gestor de Gráficos, asegurando una respuesta coherente y eficiente ante eventos sociales y actualizaciones de contenido, además de la lógica para el manejo de información y acciones entre el usuario y el sistema, como también el gestor de base de datos, para la muestra inmediata de información.

Interfaz de Usuario (IU).

Ofrece una interfaz intuitiva para configurar y controlar el sistema. Ahora, la IU se expande para visualizar y gestionar eventos notificados por Webhooks y administrados por nuestro módulo de integración de mensajes en tiempo real. Los usuarios pueden personalizar la respuesta del sistema ante eventos sociales específicos directamente desde la interfaz, como visualizar efectivamente la información entregada por el gestor de lógica.



2. Tecnologías y herramientas utilizadas.

React.

La interfaz de usuario se desarrolla utilizando React, una biblioteca de JavaScript ampliamente utilizada para construir interfaces de usuario interactivas y eficientes. Su capacidad para crear componentes reutilizables y su enfoque en el rendimiento lo convierten en una elección ideal para la construcción de la interfaz de usuario en este sistema.

Node.js.

La lógica del lado del servidor se implementa en Node.js, un entorno de ejecución de JavaScript del lado del servidor. Node.js permite una ejecución eficiente y sin bloqueo, lo que resulta fundamental para la gestión de eventos en tiempo real, especialmente en aplicaciones que hacen uso intensivo de WebSockets.

Sockets.

Se emplean sockets para habilitar la comunicación bidireccional en tiempo real entre el servidor y la interfaz de usuario construida con React. La implementación de WebSockets garantiza una transmisión de datos eficiente y en tiempo real, crucial para la dinámica integración de mensajes durante las transmisiones multimedia.

Webhooks

La integración de Webhooks se utiliza para establecer conexiones automáticas y permitir que el sistema reaccione a eventos específicos. Esto resulta esencial para la actualización en tiempo real de la información y la gestión dinámica de contenido meta durante las transmisiones multimedia.

Docker.

La contenerización con Docker se implementa para garantizar la portabilidad y consistencia del entorno de desarrollo y despliegue. Esto facilita la replicación del sistema en diferentes entornos sin preocuparse por las variaciones de configuración, asegurando así una implementación más eficiente y coherente.

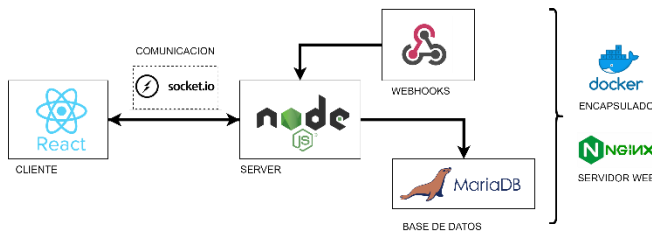
Nginx.

Se utiliza como servidor proxy inverso para gestionar las solicitudes HTTP y mejorar el rendimiento del sistema. Su capacidad para manejar múltiples conexiones simultáneas y su eficacia en la gestión de la carga hacen que sea una pieza clave para optimizar la entrega de contenido en tiempo real.

Base de Datos.

La base de datos MariaDB se incorpora para gestionar de manera eficiente el almacenamiento y recuperación de datos. Esta base de datos, derivada de MySQL, proporciona fiabilidad

y escalabilidad, mejorando la velocidad y la capacidad de gestión de datos del sistema.



3. Desarrollo.

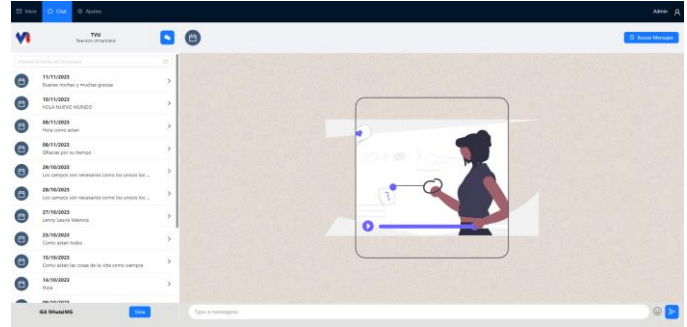
Interfaz (Cliente).

La estrategia clave fue la conceptualización modular, creando componentes reutilizables que facilitaron la construcción de una interfaz coherente y manejable. Estos componentes se organizaron en una jerarquía que reflejaba la estructura lógica de la aplicación, desde componentes de alto nivel hasta aquellos específicos para funciones particulares.

Para gestionar el estado de la aplicación de manera centralizada, implementamos Hooks. Esto permitió un control preciso del flujo de datos, facilitando la comunicación entre componentes y mejorando la coherencia del estado en toda la interfaz. Además, aprovechamos las capacidades de React para lograr una interactividad dinámica, especialmente crucial para la visualización instantánea de mensajes y cambios durante las transmisiones multimedia en tiempo real.

En términos de rendimiento, aplicamos técnicas específicas de optimización, como el uso de claves únicas y la gestión eficiente de renderizaciones. Estas medidas aseguraron una experiencia de usuario fluida, especialmente en situaciones de transmisión en tiempo real. Para el diseño y la presentación, utilizamos CSS y bibliotecas específicas para estilos, garantizando un aspecto visual coherente y moderno.

El resultado fue una interfaz de usuario dinámica e intuitiva que mejoró la experiencia del usuario durante las transmisiones multimedia en tiempo real. La combinación del enfoque modular de React, la gestión centralizada del estado con Redux y las estrategias de optimización del rendimiento culminaron en una interfaz efectiva y atractiva que cumplió con los estándares de usabilidad y rendimiento esperados.



Lógica (servidor).

Durante el desarrollo de la lógica del servidor para nuestro sistema de Integración Dinámica de Mensajes en Tiempo Real, optamos por Node.js como el entorno de ejecución del lado del servidor. Node.js se reveló esencial para gestionar eficientemente la ejecución no bloqueante y manejar múltiples conexiones simultáneas, especialmente durante la transmisión de eventos en tiempo real.

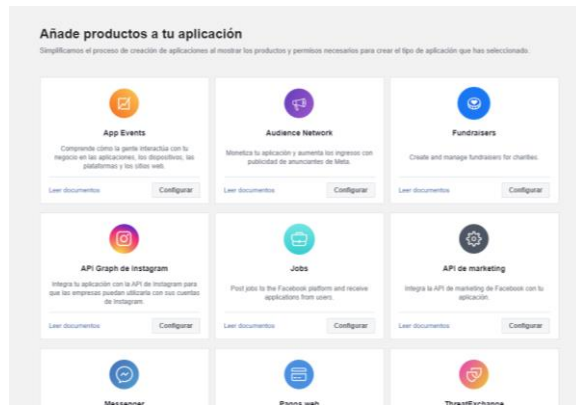
La implementación de WebSockets, a través de la biblioteca Socket.IO, permitió una comunicación bidireccional en tiempo real entre el servidor y la interfaz de usuario construida con React. Esta tecnología facilitó la transmisión instantánea de mensajes, posibilitando la actualización en tiempo real de la información y permitiendo una integración dinámica de mensajes durante las transmisiones multimedia.

La gestión de eventos se optimizó mediante la integración de Webhooks, permitiendo que el sistema reaccionara automáticamente ante eventos específicos. Esta funcionalidad no solo mejoró la capacidad de adaptación del sistema, sino que también proporcionó una capa adicional de dinamismo y actualización en tiempo real, esencial para una experiencia multimedia fluida.

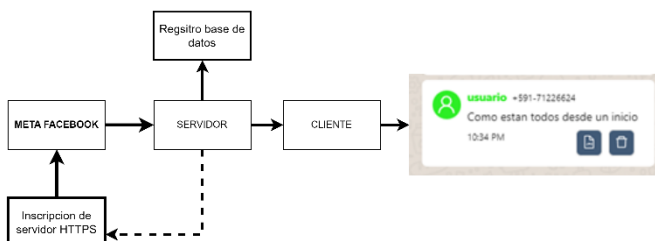
Captura de mensajes WhatsApp.

El proceso de captura de mensajes de WhatsApp mediante Webhooks en la plataforma Meta sigue una serie de pasos formales para garantizar la integridad y eficiencia del sistema:

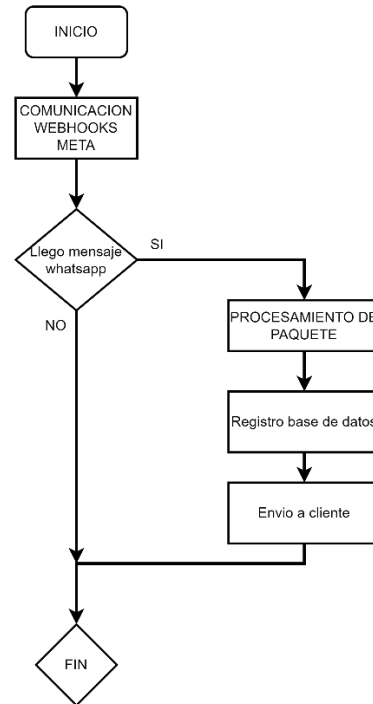
- **Creación de Cuenta de Desarrollador y Aplicación en WhatsApp Business.** Inicia con la creación de una cuenta de desarrollador en la plataforma Meta, seguido por el establecimiento de una aplicación de desarrollo específica para WhatsApp Business. Esta aplicación servirá como puente para la interacción con la plataforma y la captura de mensajes.



- **Registro del Servidor HTTPS.** Se procede a registrar el servidor HTTPS que manejará los mensajes entrantes de WhatsApp en la plataforma Meta. Este registro asegura que los paquetes de información generados por los mensajes de WhatsApp se dirijan de manera segura y confiable a nuestro servidor.
- **Envío de Paquete de Información por Meta.** Cuando un usuario envía un mensaje a través de WhatsApp, Meta genera un paquete de información correspondiente al mensaje. Este paquete es enviado de manera automática a nuestro servidor registrado, estableciendo así una conexión eficiente entre la plataforma y nuestro sistema.
- **Procesamiento del Paquete por el Servidor.** Una vez recibido, el paquete de información es procesado por la lógica del servidor. Este paso implica interpretar y extraer la información relevante del mensaje, preparándolo para su almacenamiento y posterior distribución en tiempo real.
- **Registro y Almacenamiento en la Base de Datos.** La información procesada es registrada y almacenada en la base de datos designada. Este paso es fundamental para conservar un registro histórico de los mensajes y garantizar su disponibilidad para consultas futuras.
- **Envío en Tiempo Real al Cliente mediante Socket.** Utilizando la tecnología de WebSockets, la información del mensaje es enviada en tiempo real al cliente, que puede ser la interfaz de usuario construida con React. Este enfoque permite una actualización instantánea y dinámica de los mensajes en la interfaz del usuario durante las transmisiones multimedia.



La lógica del servidor se estructuró de manera integral, aprovechando Node.js y WebSockets para gestionar eventos en tiempo real. Esta combinación estratégica de herramientas y tecnologías aseguró una comunicación fluida y actualizaciones en tiempo real, proporcionando una experiencia multimedia dinámica y de alta calidad en nuestro sistema de Integración Dinámica de Mensajes.



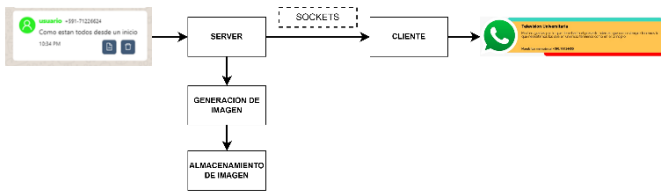
Generación de banner (Downstream Keyer)

El proceso de superposición de mensajes en tiempo real mediante el Downstream Keyer se realiza de manera organizada y eficiente, siguiendo una secuencia de pasos para garantizar la integración dinámica de mensajes en transmisiones multimedia:

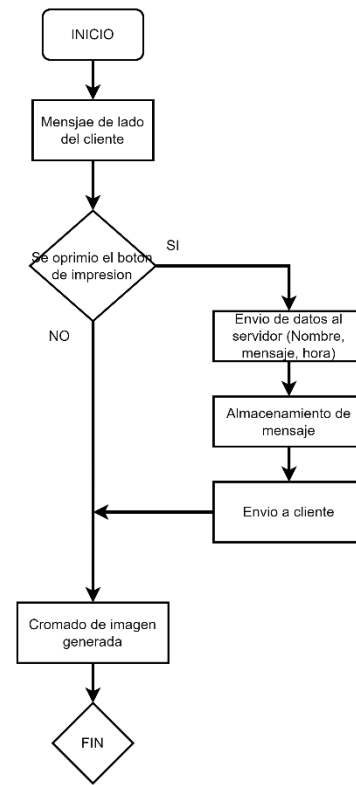
- **Envío del Mensaje al Cliente.** Una vez registrado en la base de datos, el mensaje se envía instantáneamente al cliente a través de la conexión WebSocket. Esta transmisión en tiempo real asegura que los mensajes sean recibidos y mostrados dinámicamente en la interfaz de usuario.
- **Generación de Imagen al Solicitarlo.** Cuando el usuario desea superponer el mensaje en una imagen, el mensaje ingresado incluye un botón específico para la generación de la imagen. Este enfoque permite al usuario tener un control activo sobre cuándo y cómo se superpondrá el mensaje en la imagen.
- **Proceso de Generación y Almacenamiento de la Imagen en el Servidor.** Al presionar el botón de generación de imagen, se envían al servidor el nombre del usuario, el mensaje y la fecha actual. En el

servidor, se genera la imagen correspondiente, la cual es almacenada para su posterior uso. El proceso de generación y almacenamiento asegura la disponibilidad de la imagen de manera eficiente.

- **Envío de la Dirección de la Imagen mediante Socket.** Una vez guardada la imagen, su dirección es enviada al cliente a través de la conexión WebSocket. Esta transmisión en tiempo real garantiza que el cliente obtenga la ubicación precisa de la imagen generada.
- **Visualización de la Imagen en Tiempo Real en el Cliente.** La dirección de la imagen se recibe en tiempo real del servidor y se muestra dinámicamente en la interfaz de usuario del cliente. Esta visualización instantánea proporciona al usuario la confirmación visual de la superposición del mensaje en la imagen.
- **Utilización de la Imagen en Croma.** La imagen superpuesta en tiempo real puede ser utilizada en transmisiones multimedia con efectos de croma, ofreciendo una capa adicional de dinamismo y personalización a la producción audiovisual.



Este proceso integral, desde el envío del mensaje hasta la visualización y utilización de la imagen en tiempo real, destaca la eficiencia y la capacidad dinámica del Downstream Keyer para integrar mensajes de manera efectiva en las transmisiones multimedia.

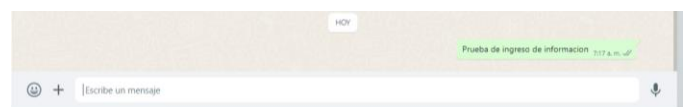


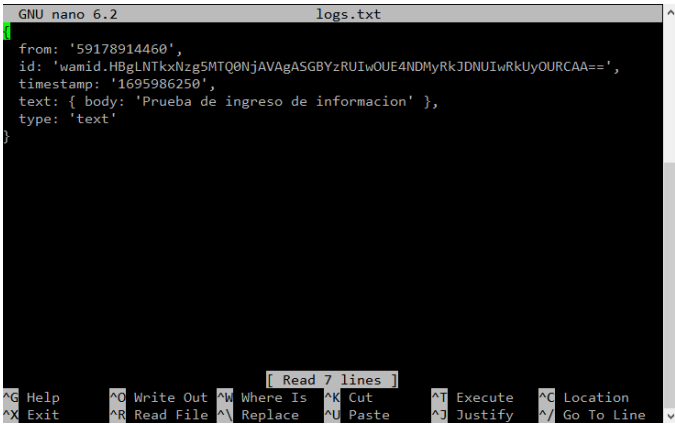
4. Prueba.

Captura de Mensajes de Webhooks de Meta (WhatsApp).

Durante las pruebas de captura de mensajes provenientes de Webhooks de Meta en WhatsApp, se llevaron a cabo evaluaciones exhaustivas para garantizar un flujo de información preciso y eficiente. En primer lugar, se verificó la correcta configuración y registro del servidor HTTPS en la plataforma Meta, asegurando que la conexión entre WhatsApp y nuestro sistema se estableciera de manera segura. Se realizaron simulaciones de interacciones de usuarios, desde mensajes de texto convencionales hasta la recepción y procesamiento de archivos multimedia.

Las pruebas incluyeron escenarios de alta carga para evaluar la capacidad del sistema para manejar flujos intensivos de mensajes en tiempo real. Se corroboró la capacidad de interpretar y gestionar adecuadamente la información incluida en los paquetes de mensajes enviados por Meta. Además, se evaluó la velocidad de respuesta del sistema, asegurando que los mensajes se capturaran y transmitieran en tiempo real de manera fluida y sin demoras significativas.





Proceso de Superposición de Mensajes en Tiempo Real con Downstream Keyer.

En las pruebas del proceso de superposición de mensajes en tiempo real utilizando el Downstream Keyer, se evaluaron varios aspectos clave del flujo de trabajo. En primer lugar, se llevó a cabo una validación detallada del envío, procesamiento y almacenamiento de mensajes en el servidor. Se simularon situaciones en las que los usuarios solicitaban la generación de imágenes con mensajes superpuestos, verificando la capacidad del sistema para generar de manera precisa y eficiente estas imágenes.

Además, se evaluó el rendimiento de la transmisión en tiempo real de la dirección de las imágenes generadas al cliente mediante WebSocket. Se realizaron pruebas para asegurar que la visualización en tiempo real de las imágenes superpuestas en la interfaz de usuario del cliente fuera instantánea y precisa. Se verificó también la consistencia en la transmisión de información y la capacidad del Downstream Keyer para integrar mensajes dinámicamente en transmisiones multimedia.

En ambos casos, se realizaron pruebas de recuperación y manejo de errores para garantizar la robustez del sistema en situaciones menos ideales. Los resultados positivos obtenidos durante estas pruebas validan la capacidad integral de nuestro sistema para gestionar mensajes en tiempo real provenientes de Webhooks de Meta, así como su eficiente superposición en transmisiones multimedia, respaldando la confiabilidad y calidad del sistema en su totalidad.



IV. IMPLEMENTACIÓN

La implementación detallada del sistema se llevó a cabo con un enfoque integral, haciendo uso de herramientas esenciales como Nginx y Docker para asegurar la eficiencia y escalabilidad en un entorno de servidor alojado en Digital Ocean.

En el contexto de Nginx, este fue configurado como un servidor proxy inverso para gestionar las solicitudes HTTP y mejorar el rendimiento global del sistema. Durante la configuración, Nginx se optimizó para manejar múltiples conexiones simultáneas y para mejorar la entrega de contenido en tiempo real. Su capacidad para equilibrar la carga y gestionar la escalabilidad resultó fundamental para asegurar una experiencia de usuario fluida y receptiva.

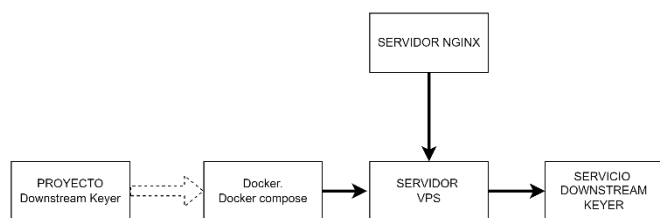
La estrategia de contenerización con Docker también desempeñó un papel esencial en la implementación. Se configuraron contenedores Docker para encapsular la aplicación junto con todas sus dependencias, lo que permitió la portabilidad y consistencia del entorno de desarrollo y despliegue. Esta metodología facilitó la replicación del sistema en diversos entornos sin preocupaciones por las variaciones de configuración, asegurando una implementación eficiente y coherente.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
0f493f323a54	node:18	"docker-entrypoint.s..."	2 weeks ago	Up 2 weeks
61e8a7155eb1	node:18	"docker-entrypoint.s..."	2 weeks ago	Up 2 weeks
02b91c4e2598	adminer	"entrypoint.sh php -..."	2 weeks ago	Up 2 weeks
940029869cbd	mariadb:jammy	"docker-entrypoint.s..."	2 weeks ago	Up 2 weeks

La elección de Digital Ocean como proveedor de servicios en la nube complementó eficazmente la implementación. Se configuraron instancias virtuales en Digital Ocean, aprovechando su infraestructura confiable y escalable. La combinación de Nginx para la gestión de solicitudes y Docker para la contenerización facilitó el despliegue en el entorno de Digital Ocean, garantizando la disponibilidad y rendimiento del sistema.

En términos de seguridad, se implementaron medidas robustas en Nginx y Docker para resguardar la aplicación y los datos del sistema. Se configuraron firewalls y se aplicaron prácticas recomendadas de seguridad para garantizar la integridad y confidencialidad de la información.

En conjunto, la implementación del sistema con Nginx y Docker en un servidor de Digital Ocean representa una estrategia completa, priorizando la eficiencia operativa, la escalabilidad y la seguridad. La combinación de estas herramientas y servicios proporcionó un entorno estable y optimizado para la ejecución exitosa de nuestro sistema de Integración Dinámica de Mensajes en Tiempo Real.



La implementación estratégica de nuestro sistema en el canal de televisión TVU, específicamente en el Canal Universitario de la Universidad Mayor de San Andrés (UMSA), tuvo como propósito principal la mejora y dinamización de la producción audiovisual en tiempo real. Al incorporar nuestro sistema en TVU, se enfocó en potenciar la presentación multimedia mediante el uso del Downstream Keyer, una herramienta que posibilita la superposición de mensajes en tiempo real durante las transmisiones.

Durante el proceso de introducción, se llevaron a cabo esfuerzos coordinados para garantizar la plena compatibilidad con los sistemas preexistentes de TVU, asegurando así una transición sin inconvenientes. Esto implicó una cuidadosa integración de nuestro sistema con los flujos de trabajo y las infraestructuras tecnológicas existentes en el Canal Universitario de la UMSA. Se realizaron ajustes específicos para adaptar nuestro sistema al entorno televisivo, maximizando su utilidad y complementando la narrativa visual de TVU.



La implementación exitosa en TVU no solo validó la capacidad técnica de nuestro sistema para integrarse en entornos televisivos consolidados, sino que también subrayó su versatilidad y dinamismo como una herramienta valiosa. Al proporcionar una solución dinámica y versátil, nuestro sistema contribuye a enriquecer la producción audiovisual en tiempo real del Canal Universitario UMSA, ofreciendo nuevas posibilidades creativas y mejorando la experiencia visual para los espectadores.



V. RESULTADOS

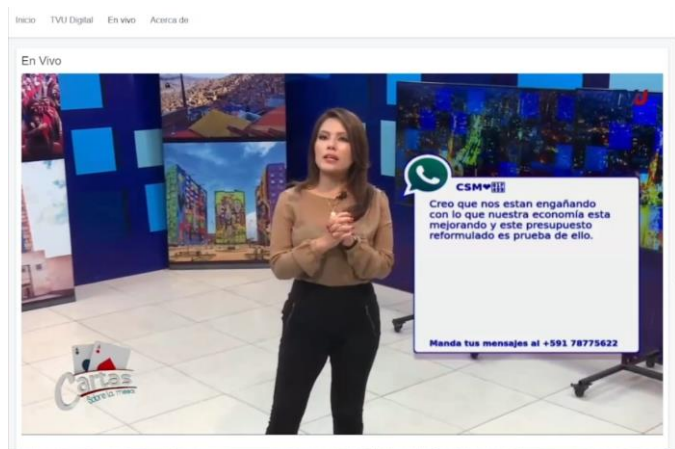
Tras la cuidadosa implementación en el Canal Universitario de la Universidad Mayor de San Andrés (UMSA), específicamente en TVU, los resultados obtenidos reflejaron no solo una integración efectiva, sino también el éxito en la puesta en funcionamiento de nuestro sistema en un programa de televisión.

La implementación en TVU demostró ser un paso significativo hacia la mejora de la producción audiovisual en tiempo real. La capacidad del Downstream Keyer para superponer mensajes dinámicamente durante las transmisiones proporcionó una dimensión adicional y atractiva al contenido televisivo. La optimización de la presentación multimedia permitió una mayor interactividad y participación del público, enriqueciendo la experiencia visual ofrecida por TVU.

La transición sin problemas y la completa compatibilidad con los sistemas existentes en TVU validaron la robustez y adaptabilidad de nuestro sistema. Los resultados obtenidos en esta etapa inicial sientan las bases para futuras aplicaciones y desarrollos dentro del canal televisivo, destacando el potencial continuo de nuestro sistema en la mejora de la producción televisiva.



La puesta en funcionamiento exitosa en un programa de televisión específico consolidó aún más la utilidad práctica de nuestro sistema. La capacidad de integrar dinámicamente mensajes en tiempo real en un contexto de programa televisivo real resaltó la versatilidad y la aplicabilidad de nuestro sistema en entornos de producción televisiva en vivo.



VI. CONCLUSIONES

La culminación del proceso integral, desde la captura de mensajes de WhatsApp hasta la generación de imágenes con nuestro servicio Downstream Keyer, ha proporcionado valiosas conclusiones que destacan la eficacia y versatilidad de nuestro sistema.

En la fase de captura de mensajes de WhatsApp, el sistema ha demostrado su capacidad para gestionar la información de

manera precisa y en tiempo real. La implementación de Webhooks de Meta permitió una conexión segura y eficiente, garantizando la recepción oportuna de mensajes en diferentes formatos y medios.

La generación de imágenes con el servicio Downstream Keyer ha sido un componente clave en la optimización de la presentación multimedia. La capacidad para superponer mensajes dinámicamente en tiempo real ha enriquecido significativamente las transmisiones, proporcionando una capa adicional de interactividad y personalización.

La implementación exitosa en TVU y su integración en un programa de televisión real han validado la adaptabilidad del sistema a entornos televisivos establecidos. La transición sin problemas y la completa compatibilidad con los sistemas existentes en TVU han respaldado la robustez del sistema, consolidándolo como una herramienta efectiva en la producción televisiva en tiempo real.

La respuesta positiva observada durante la puesta en funcionamiento en un programa de televisión específico ha destacado la versatilidad del sistema en escenarios de transmisiones en vivo. La capacidad para integrar mensajes dinámicamente en programas televisivos ha contribuido a enriquecer la experiencia del espectador y a aumentar la interactividad.

En conclusión, este proceso integral ha validado el impacto positivo y la aplicabilidad de nuestro sistema en la evolución de la producción televisiva. Desde la captura de mensajes de WhatsApp hasta la generación de imágenes con el servicio Downstream Keyer, cada fase ha aportado a la consolidación de un sistema dinámico y versátil que mejora significativamente la narrativa visual en el ámbito televisivo.

VII. REFERENCIAS

- [1] A. Smith, «Secure Communication with Webhooks in Instant Messaging Platforms.,» *Journal of Communication Technology*, n° 45, pp. 231-245, 2019.
- [2] M. García, «Enhancing Real-Time Visual Overlays with Improved Downstream Keyer.,» *Multimedia Systems*, n° 36, pp. 412-425, 2018.
- [3] R. Johnson y E. Miller, «Optimizing Real-Time Content Delivery.,» *Communications in Computing*, n° 38, pp. 567-582, 2020.
- [4] J. White y P. Brown, «Seamless Integration in Television Environments.,» *Broadcasting Technology Journal*, n° 45, pp. 723-738, 2019.
- [5] L. & W. Y. Chen, «Enhancing Real-time Communication in Multimedia Streaming via WhatsApp Business Integration.,» *ournal of Multimedia Tools and Applications*, pp. 1-15, 2022.