



UNIVERSIDAD TECNOLÓGICA NACIONAL
Facultad Regional Córdoba

PROYECTO FINAL



Departamento de Ingeniería en Sistemas de Información

Materia: Proyecto Final

Curso: 5k4 2026 - Grupo 2

Docentes: Victoria Ahumada - Natalia Jaime

Autores:

- Escudero Garay Candela - 91516
- Lucero Franco Manuel - 96311
- Monchietti Francisco - 98630
- Olmos Francisco - 91854
- Romanutti Valentín - 94217

ClipIA

Analiza streams en vivo con IA, extrae los momentos clave y crea clips de forma totalmente autónoma. Luego, edita el material y lo publica en redes sociales automáticamente.

¿CÓMO FUNCIONA?



TRANSMISIÓN
EN VIVO



ANÁLISIS
EN TIEMPO REAL



SE DETECTA
EL MOMENTO



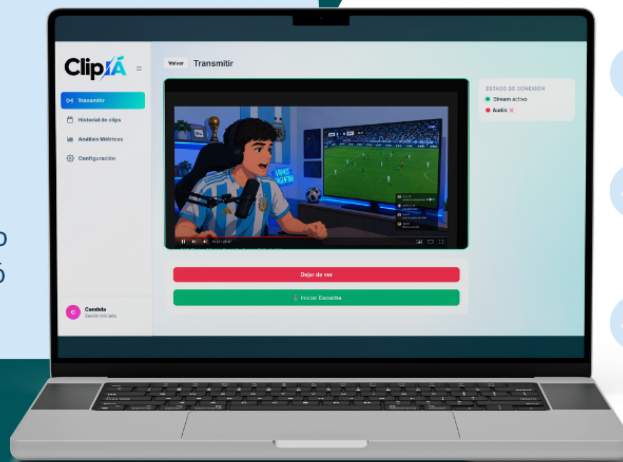
CLIP GENERADO



EDICIÓN
Y PUBLICACIÓN
AUTOMÁTICA

PROBLEMAS

- ✗ Crear clips manualmente consume tiempo
- ✗ Terceros capitalizan tu contenido
- ✗ El público llega cuando la transmisión terminó



SOLUCIÓN

- ✓ Generación de clips autónoma y eficiente
- ✓ Clips listos al mismo tiempo del vivo
- ✓ Publicación inmediata que atrae espectadores al vivo

HERRAMIENTAS Y TECNOLOGÍAS



UTN

Facultad Regional Córdoba

Ingeniería en Sistemas de Información
Cátedra: Proyecto Final -
Categoría: Producto - 2026

EQUIPO

- Candela Escudero - escuderoogarayc@gmail.com
- Valentin Romanutti - valentinromanutti@gmail.com
- Manuel Lucero - manu.lucero07@gmail.com
- Francisco Monchietti - franmonchietti@gmail.com
- Francisco Olmos - olmosfranciscotomas@gmail.com

DOCENTE

- Natalia Jaime
- Victoria Ahumada



ClipIÁ: Detección y Generación Automática de Clips Virales en Transmisiones en Vivo

Escudero, Candela – Lucero, Manuel – Monchietti, Francisco - Olmos, Francisco – Romanutti, Valentín

Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

ClipIÁ nació como respuesta a un problema que enfrenta todo emisor de contenido en línea: las transmisiones en vivo duran horas, pero los momentos que realmente valen la pena compartir son pocos y se pierden apenas termina el vivo. Recortarlos a mano exige tiempo que la mayoría de los creadores de contenido no tiene. ClipIÁ resolvió ese problema escuchando la transmisión mientras ocurre, detectando con inteligencia artificial los momentos con mayor potencial viral y generando el clip de forma automática. El desarrollo se planificó en una hoja de ruta de seis releases incrementales. El flujo esencial de detección y generación automática de clips se implementó y se desplegó en un entorno productivo real, donde fue puesto a prueba por un cliente externo al equipo. Más que una herramienta de edición, ClipIÁ se propuso como la manera de viralizar contenido mientras todavía está sucediendo.

Palabras Clave

Detección de momentos virales, transmisión en vivo, inteligencia artificial, procesamiento de audio y video, streaming, generación automática de clips. Categoría: Producto.

Introducción

La transmisión de contenido en vivo se consolida como uno de los formatos de mayor crecimiento dentro del consumo digital. Sin embargo, la mayor parte de ese contenido tiene una vida útil limitada al momento de su emisión: los fragmentos con potencial de viralización (una jugada destacada, un momento cómico, una reacción espontánea) suelen perderse dentro de transmisiones que pueden extenderse por varias horas. Actualmente, identificarlos y recortarlos se realiza en forma manual, lo que exige ver la transmisión completa, demanda tiempo, resulta poco escalable para quienes transmiten con frecuencia y

hace que muchos momentos valiosos nunca se difundan fuera del vivo original.

ClipIÁ propone abordar este problema mediante un sistema de inteligencia artificial que analiza en tiempo real el audio, el contexto y la imagen de una transmisión en vivo, con el objetivo de detectar automáticamente los momentos con mayor potencial viral y generar los clips correspondientes sin intervención humana. El valor diferencial radica en la inmediatez: viralizar un fragmento mientras la transmisión original todavía está ocurriendo.

En las secciones siguientes se describen las herramientas y la metodología empleadas, los resultados alcanzados (incluyendo el despliegue en un entorno productivo y la validación con un cliente piloto), una discusión sobre su alcance y limitaciones, y las conclusiones y líneas de trabajo futuro.

Elementos del Trabajo y Metodología

El desarrollo del proyecto se organizó bajo el marco ágil Scrum, con sprints de dos semanas de duración y un Sprint Backlog de historias de usuario estimadas en Story Points, lo que permitió planificar el trabajo, medir el avance real respecto de lo comprometido y ajustar la capacidad del equipo en los sprints subsiguientes. El equipo se conformó por cinco integrantes con roles diferenciados: desarrollo backend y arquitectura, desarrollo frontend y documentación, desarrollo frontend y diseño UX/UI, ingeniería de inteligencia artificial, y aseguramiento de calidad (QA).

Sobre la base de un mapeo de historias de usuario (user story mapping), el alcance completo del producto se organizó en una

hoja de ruta de seis releases incrementales. Las dos primeras releases concentran el flujo esencial del producto —detección y generación automática de clips, gestión de cuentas y configuración de programas— y las cuatro restantes incorporan de forma progresiva la publicación automática en redes sociales, la edición asistida de clips, métricas de desempeño y la gestión comercial de suscripciones.

La arquitectura se organizó en capas independientes y desacopladas, orquestadas con Docker Compose [6]. El frontend se desarrolló en Angular [1] (componentes standalone, lazy loading de rutas) con PrimeNG [2] para la interfaz y HLS.js para reproducir el video en vivo, por tratarse de un framework tipado y robusto para una SPA con estado en tiempo real. El backend se implementó con FastAPI [3] sobre Python [8], en capas de routers, servicios, repositorios, esquemas y modelos. La ingesta de la transmisión se resolvió con un servidor Nginx con módulo RTMP [4], que recibe la señal emitida desde cualquier aplicación compatible con dicho protocolo, como por ejemplo OBS [9], y la persistencia se implementó sobre PostgreSQL [5].

El módulo de inteligencia artificial constituye el componente central: procesa audio, contexto e imagen para determinar en tiempo real si un fragmento tiene potencial viral. Una limitación externa del proveedor de IA utilizado inicialmente obligó a migrar a un proveedor distinto, decisión que se abordó como una decisión de arquitectura informada. Todo el sistema se desplegó además sobre Amazon Web Services (AWS) [7], dejando atrás el entorno local y quedando disponible como ambiente productivo accesible fuera de las máquinas del equipo.

Como apoyo al proceso de trabajo, el equipo incorporó herramientas de inteligencia artificial: Claude [10] se utilizó para la documentación del proyecto,

mientras que Codex [11] y Claude [10] se emplearon como asistentes de programación durante el desarrollo del sistema.

Resultados

El equipo implementó el flujo esencial del producto: conectar una transmisión emitida mediante el protocolo RTMP (compatible con OBS y otras aplicaciones de transmisión), iniciar la escucha de audio y video, ejecutar el análisis automático de detección de momentos virales y generar el clip correspondiente sin intervención manual, además de la gestión básica de la biblioteca de clips y la publicación manual.

A esto se sumó la gestión de cuentas de usuario, la configuración de perfiles de clips por categoría —humor y deportes— y la organización del contenido en torno al concepto de programa, incluyendo su creación, selección, filtrado y búsqueda dentro de la biblioteca. Con este flujo implementado, el sistema cubre de punta a punta el objetivo que da origen al proyecto: detectar y generar automáticamente clips a partir de una transmisión en vivo.

El sistema se desplegó de manera integral en AWS [7], quedando disponible como servicio remoto y ya no dependiente de los entornos locales del equipo. A partir de ese despliegue se estableció una relación con un cliente externo al equipo, que aceptó utilizar la plataforma de forma piloto sobre sus propias transmisiones, constituyendo la primera validación del sistema con un usuario real.

Discusión

Los resultados permiten sostener que el objetivo planteado (detectar y generar automáticamente clips a partir de una transmisión en vivo) se cumple con el flujo implementado, lo cual valida la viabilidad técnica de la propuesta.

La incorporación de un cliente piloto externo aporta una validación que

trasciende las pruebas internas y constituye un primer indicio de aceptación del producto. Se trata, sin embargo, de una validación inicial acotada a un único cliente, por lo que sus resultados son una señal preliminar y no una prueba representativa ante una base de usuarios más amplia.

El sistema presenta limitaciones a señalar: la precisión de la detección depende de la calidad del modelo de IA y del tipo de contenido transmitido, y la dependencia de un proveedor externo expuso al equipo a restricciones ajenas a su control, lo que evidenció la importancia de un módulo de IA desacoplado del resto del sistema. Quedan fuera del alcance evaluado en este trabajo, y restan por evaluar, la publicación automática en redes sociales, la edición asistida de clips, las métricas de desempeño y la escalabilidad del sistema en AWS ante transmisiones más largas o más clientes concurrentes.

Conclusión

El desarrollo de ClipIÁ demuestra que es posible construir un sistema capaz de detectar y generar automáticamente clips de momentos destacados a partir de una transmisión en vivo, integrando ingesta de video, inteligencia artificial y procesamiento en tiempo real dentro de una arquitectura desacoplada por capas. El flujo esencial definido como objetivo del proyecto se implementó, se encuentra desplegado en un entorno productivo sobre AWS e incorpora personalización y organización del contenido para el usuario final. La validación temprana con un cliente piloto acerca a ClipIÁ de un producto utilizado por usuarios reales por sobre un desarrollo exclusivamente académico.

El trabajo futuro está definido en la hoja de ruta del proyecto: la Release 3 incorporará la publicación automática en YouTube e Instagram con supervisión humana y la edición asistida de clips; la Release 4 sumará la publicación en TikTok y X; la

Release 5 incorporará métricas de desempeño de los clips publicados; y la Release 6 cerrará el alcance con la gestión comercial de suscripciones y planes, junto con la adaptación automática de clips en base a métricas.

Agradecimientos

Los autores agradecen al Jefe de Trabajos Prácticos y al Profesor Adjunto de la cátedra Proyecto Final por el acompañamiento brindado durante el desarrollo del proyecto, y al cliente piloto que aceptó validar la plataforma sobre sus propias transmisiones.

Referencias

- [1] Angular - <https://angular.dev> (jul. 2026).
- [2] PrimeNG - <https://primeng.org> (jul. 2026).
- [3] FastAPI - <https://fastapi.tiangolo.com> (jul. 2026).
- [4] Ngix-RTMP - [GitHub - arut/nginx-rtmp-module](https://github.com/arut/nginx-rtmp-module): NGINX-based Media Streaming Server (jul. 2026).
- [5] PostgreSQL - <https://www.postgresql.org/docs> (jul. 2026).
- [6] Docker - <https://docs.docker.com/compose> (jul. 2026).
- [7] Amazon Web Services - <https://docs.aws.amazon.com> (jul. 2026).
- [8] Python - <https://www.python.org> (jul. 2026).
- [9] OBS Studio - <https://obsproject.com> (jul. 2026).
- [10] Claude - <https://www.anthropic.com/claude> (jul. 2026).
- [11] Codex - <https://chatgpt.com/codex> (jul. 2026).

Datos de Contacto

Monchietti, Francisco - Universidad Tecnológica Nacional, Facultad Regional Córdoba. E-mail: franmonchietti@gmail.com

Olmos, Francisco - Universidad Tecnológica Nacional, Facultad Regional Córdoba. E-mail: olmosfranciscotomas@gmail.com

Escudero, Candela - Universidad Tecnológica Nacional, Facultad Regional Córdoba. E-mail: escudero garayc@gmail.com

Lucero, Franco Manuel - Universidad Tecnológica Nacional, Facultad Regional Córdoba. E-mail: manu.lucero07@gmail.com

Romanutti, Valentín - Universidad Tecnológica Nacional, Facultad Regional Córdoba. E-mail: valentinromanutti@gmail.com

PLANILLA PARA CATALOGAR EL PROYECTO FINAL

AÑO	2026	CURSO Y NRO. DE GRUPO	5k4 - G02
NOMBRE DEL SISTEMA / PROYECTO			
CiplÁ			
CATEGORÍA (Solución de Negocio / Producto / Proyecto de Impacto Social)			
Producto			
HERRAMIENTAS Y TECNOLOGÍA UTILIZADAS			
ÁMBITO DE APLICACIÓN	NOMBRE Y VERSIÓN		
ENTORNO DE DESARROLLO	Visual Studio Code 1.128		
REPOSITORIOS Y VERSIONADO	GitHub		
PROGRAMACIÓN	Angular 21.2.13, Python 3.14, FastAPI 0.136.1, Nginx RTMP module, Caddy		
BASE DE DATOS	PostgreSQL 18.6		
COMUNICACIÓN INTERNA	Discord 10.0.26200 WhatsApp		
CAPACITACIÓN	Youtube , foros		
PRUEBAS DEL SISTEMA	Pruebas manuales: - Navegador - Postman v12.23.1 - Swagger UI 5.32.13 Test unitarios - Vitest 4.1.10 - Pytest 9.0.1		
GESTIÓN DEL PROYECTO	Jira 2026		
DOCUMENTACIÓN	Google Drive		
MODELOS	Draw.io Figma Stitch Canva		