

PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA
Facultad De Ciencias e Ingeniería
Escuela de Ingeniería en Computación y Telecomunicaciones



**Bot de Asistencia Interactivo en Programas de Algoritmos y Diagramas de
Programación para la Asignatura de “Introducción a la Algoritmia”**

**Un proyecto presentado como requisito parcial para optar por el título de
Ingeniero en Ciencias de la Computación en la
Pontificia Universidad Católica Madre y Maestra.**

por

Natasha María López Concepción - 10141274

Vladimir Osvaldo Curiel Ovalles - 10141415

Asesor: Lisibonny Eustina Beato

Santiago de los Caballeros, República Dominicana

23 de julio del 2025

Tabla de Contenido

Resumen.....	10
Abstract.....	11
Introducción.....	11
Objetivos Del Proyecto.....	19
Justificación del Proyecto.....	20
Limitaciones del Proyecto.....	21
Capítulo I - Marco Teórico.....	22
 1.2. Definición de Términos y Glosario.....	35
 2. Solución Propuesta.....	39
 Impacto sobre los riesgos del proyecto.....	54
 2.5. Presupuesto.....	55
 2.6. Definición de la demostración.....	57
 2.7. Análisis y Diseño.....	58
 2.7.1. Análisis.....	58
 2.7.2. Diseño.....	65
 Desarrollo e Implementación.....	66
 Análisis de los Resultados.....	103
 Conclusiones y Recomendaciones.....	116
 Bibliografía.....	119

Índice de Tablas

Tablas	
Tabla 1.....	36
Tabla 2.....	38
Tabla 3.....	39
Tabla 4.....	45
Tabla 5.....	45
Tabla 6.....	46
Tabla 7.....	46
Tabla 8.....	48
Tabla 9.....	50
Tabla 10.....	63
Tabla 11.....	64
Tabla 12.....	65
Tabla 13.....	66
Tabla 14.....	91

Índice de Diagramas

Diagrama 1.....	59
Diagrama 2.....	61
Diagrama 3.....	62
Diagrama 4.....	63
Diagrama 5.....	65
Diagrama 6.....	65

Índice de Figuras

Figura 1	64
Figura 2	64
Figura 3	66
Figura 4	68
Figura 5	69
Figura 6	71
Figura 7	73
Figura 8	73
Figura 9	74
Figura 10	75
Figura 11	75
Figura 12	78
Figura 13	78
Figura 14	79
Figura 15	80
Figura 16	80
Figura 17	81
Figura 18	83
Figura 19	83
Figura 20	85
Figura 21	85
Figura 22	86
Figura 23	86
Figura 24	87
Figura 25	88
Figura 26	89

Figura 27.....	90
Figura 28.....	91
Figura 29.....	92
Figura 30.....	93
Figura 31.....	94
Figura 32.....	95
Figura 33.....	96
Figura 34.....	97
Figura 35.....	97
Figura 36.....	98
Figura 37.....	99
Figura 38.....	100
Figura 39.....	101
Figura 40.....	101
Figura 41.....	102
Figura 42.....	103
Figura 43.....	103
Figura 44.....	104
Figura 45.....	104
Figura 46.....	105
Figura 47.....	105
Figura 48.....	106
Figura 49.....	107
Figura 50.....	107
Figura 51.....	108
Figura 52.....	110
Figura 53.....	111
Figura 54.....	112

Figura 55.....114

Índice de Textos

Textos.....	
Texto 1.....	96
Texto 2.....	96
Texto 3.....	96
Texto 4.....	96

Índice de Anexos

Anexo A.....	127
--------------	-----

Resumen

La virtualidad ha dado un gran paso en la educación. En el área de la programación se ha generado la necesidad de herramientas automatizadas que puedan facilitar al estudiante para entender los algoritmos y diagramas vistos en sus clases en línea. En este contexto, la presente investigación propone un Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación que basado en inteligencia artificial pueda ayudar a la detección y reconocimiento de fragmentos de código y diagramas en imágenes extraídas de video grabaciones de la materia Introducción a la Algoritmia.

El problema central radica en la dificultad de los estudiantes para organizar y comprender información clave de las sesiones virtuales, lo que impacta su proceso de aprendizaje. Para abordar esta problemática, se implementa un enfoque metodológico en varias etapas. En primer lugar, se realiza una segmentación estructurada de los videos para identificar momentos relevantes. Luego, se lleva a cabo la detección de bloques de código en imágenes extraídas de los videos, asegurando la identificación precisa de áreas de interés. Posteriormente, se efectúa la extracción de texto de los fragmentos detectados, garantizando la recuperación efectiva del contenido programático. Finalmente, se aplica un proceso de análisis y estructuración de la información recuperada, permitiendo su organización de manera eficiente para su consulta y aprovechamiento en el aprendizaje.

Los resultados preliminares obtenidos validan la eficiencia de lo propuesto, demostrando que el bot es capaz de identificar y extraer fragmentos de código con alta precisión. Se concluye que esta herramienta puede ser de gran utilidad para estudiantes y docentes, facilitando la documentación y optimización del aprendizaje en entornos digitales.

Abstract

Virtual reality has taken a major step forward in education. In the field of programming, there has been a need for automated tools that can help students understand the algorithms and diagrams presented in online classes. In this context, this research proposes an Interactive Assistance Bot for Algorithms and Programming Diagrams that, based on artificial intelligence, can assist in the detection and recognition of code fragments and diagrams in images extracted from video recordings of the Introduction to Algorithms course.

The central problem lies in students' difficulty organizing and understanding key information from virtual sessions, which impacts their learning process. To address this issue, a multi-stage methodological approach is implemented. First, a structured segmentation of the videos is performed to identify relevant moments. Then, code blocks are detected in images extracted from the videos, ensuring the accurate identification of areas of interest. Text is then extracted from the detected fragments, ensuring the effective retrieval of the program content. Finally, a process of analyzing and structuring the retrieved information is applied, allowing it to be efficiently organized for reference and use in learning.

The preliminary results obtained validate the efficiency of the proposed approach, demonstrating that the bot is capable of identifying and extracting code fragments with high accuracy. It is concluded that this tool can be very useful for students and teachers, facilitating documentation and optimizing learning in digital environments.

Introducción

Antecedentes del Problema

Debido a la pandemia causada por el COVID-19, las clases impartidas por colegios, universidades y demás instituciones académicas se vieron en la obligación de adaptar la metodología para continuar con las labores académicas, adoptando así un entorno digital donde surgieron las clases virtuales. En la actualidad, a pesar de que las instituciones académicas han vuelto al formato de clases presenciales, las clases virtuales no han desaparecido, sino que se han mantenido dentro de la metodología de clases presenciales, así permitiendo a la vez, impartir clases a distancia de ser necesario e incluso grabar el contenido de la misma [1].

La problemática inicia cuando se involucran a los estudiantes, su capacidad de atención, los altos niveles de distracción al encontrarse en entornos no aptos para recibir clases a distancia, la ansiedad generada e incluso el miedo a realizar preguntas a los profesores o volver a pedir explicaciones del contenido que no se entendió. Una encuesta muestra que el 95% de los estudiantes encuestados utilizan su celular en clases virtuales para actividades no académicas en contra del 75% en clases presenciales, un incremento del 20% en sesiones virtuales [2]. Del mismo modo, los resultados presentados en otra encuesta a estudiantes dentro de la disciplina STEM (Science, Technology, Engineering and Mathematics), demostraron que el 51% de los estudiantes presentan más ansiedad al tener ciclos académicos completamente virtuales [3].

La situación identificada y el inminente crecimiento de la inteligencia artificial (IA), los ‘ChatBot’, IAs generativas que se basan en Modelos Extensos de Lenguaje (LLM) han provocado que los estudiantes vean como una opción más viable recurrir a estas herramientas de IA antes que recurrir a sus profesores. Resultados indican qué en la modalidad virtual, el 45% de los estudiantes discuten sobre tareas o calificaciones a diferencia del 54% en modalidad presencial, para una diferencia del 9% a favor de la presencialidad [4]. Asimismo, la misma muestra que el 58% de los estudiantes de modalidad virtual nunca interactúan con el maestro fuera de las clases mientras que en

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

la modalidad presencial podemos apreciar que el 43% de estos son los que no interactúan, probando un 15% más de interacción en la presencialidad [4].

Uno de los problemas que radica de este cambio es que al recurrir a estos modelos se pierde todo el contexto académico: el programa de la asignatura, las palabras del docente, el libro de texto de la materia e incluso como el profesor interpreta el libro y lo transmite a sus estudiantes. Se realizaron demostraciones de rendimiento con pruebas de 10 preguntas que indicaron que, modelos como Chat GTP (modelo de verano 2024) pueden obtener hasta 80% de la puntuación correcta en comparación con el promedio de profesionales del área llegando hasta un 96.7%, con una pérdida de conocimiento del 16.7%, además, afirman que existe una pérdida notable al agregar tablas grandes o complejas y lo mismo sucede con preguntas de contenido que fue mostrado en clase, ya que pierde todo el contexto en la que la información fue dada [5].

Esto causa que los estudiantes al final del día se encuentren consumiendo casi en su totalidad el contenido generado por IA que no está supervisado, y del mismo modo, realizando sus tareas de esta forma lo que estará bajando mucho el desarrollo de sus habilidades, el pensamiento lógico y crítico. Con esto, no se intenta justificar la ausencia del uso de las IA, ya que existen afirmaciones acerca de que el uso de la IA le ofrece a los estudiantes algunas innovaciones como lo puede ser: personalización de aprendizaje, incremento de la motivación para estudiar, predicción de calificaciones, entre otros beneficios más [6].

Antecedentes del Proyecto

Como respuesta a la problemática presente, no se ha encontrado una solución que resuelva explícitamente lo requerido en el escenario planteado pero, sí se pueden encontrar opciones para resolver los problemas por partes. Se puede fragmentar el problema en tareas más simples: Detección de programas de algoritmos, texto y formas en imágenes (que sirve para detectar las formas de los diagramas de programación.)

Diversas investigaciones muestran procesos rigurosos utilizando diversas técnicas en conjunto para lograr esas tareas más simples con fines distintos a los propuestos en este proyecto. Aún así la teoría detrás de cada investigación puede ofrecer un acercamiento más técnico y científico sobre las implementaciones que estos llevaron a cabo, sus pruebas, resultado y conclusiones. Se puede expandir de forma breve dentro de cada investigación propuesta.

En un proyecto llamado “DiagramVoice” [7], se presenta un sistema de comentarios automáticos para videos de clases en línea, el cual fue diseñado como una aplicación móvil para apoyar a estudiantes con discapacidad visual. Su sistema analiza materiales visuales como texto, imágenes naturales y diagramas presentes en los videos de conferencias. Para lograr esto, se utilizaron técnicas como la detección de transiciones de escena con la API PySceneDetect, el reconocimiento óptico de caracteres (OCR) mediante la API Naver Clova OCR, y la descripción automática de imágenes a través de la API Microsoft Azure Cognitive Computer Vision. Además, se desarrolló un algoritmo específico para generar comentarios de diagramas. Para la salida de voz, implementaron la API Google Cloud Text-to-Speech. Con este enfoque, buscaron mejorar la comprensión y el interés de los estudiantes con discapacidad visual en el contenido de las clases, promoviendo un entorno de aprendizaje independiente y accesible.

Los resultados del proyecto, demostraron que las técnicas que propusieron, mejoraron la satisfacción y la adaptación en la generación de comentarios automáticos para los materiales visuales en clases en línea. De la misma forma, superando así el uso de la

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

API de subtitulado de imágenes. Asimismo, esas técnicas ofrecieron una mejor experiencia educativa para aquellos estudiantes con discapacidad visual, los cuales pueden comprender de manera más efectiva los materiales virtuales de sus clases en línea.

En un proyecto llamado “BloSum” [8] , este presenta un sistema de conversión automática de diagramas de bloques en texto resumido. Su sistema analiza diagramas de bloques, que son usualmente utilizados para representar flujos de trabajo o procesos, y extrae su significado contextual en forma de tripletas (relaciones entre elementos). Estas tripletas ayudan a un modelo de lenguaje a generar resúmenes en lenguaje natural. Para lograr esto, su arquitectura se divide en cuatro partes principales: predicción de formas, predicción de texto, predicción de flechas y un generador de tripletas.

En la predicción de formas, se utilizó un modelo Faster R-CNN con una red de pirámide de características (FPN) y una arquitectura Inception-ResNet-v2 como backbone para detectar y clasificar las formas en el diagrama. Para la predicción de texto, se empleó EasyOCR, una herramienta de código abierto que detecta texto incluso en imágenes con ruido. En la predicción de flechas, se aplicaron técnicas como la transformada de Hough para detectar líneas y diferenciar entre el inicio y el final de las flechas. Finalmente, el generador de tripletas combina toda esta información para establecer relaciones entre las formas y generar tripletas (<H> <R> <T>), que representan las conexiones y relaciones en el diagrama.

Los resultados del proyecto demostraron que “BloSum” cumplió el objetivo de generar descripciones textuales a partir de diagramas de bloques. Asimismo, para la validación del modelo, el mismo fue probado en conjuntos de datos públicos de diagramas manuscritos (FC_A y FC_B). También, se creó un nuevo dataset CBD, el cual contiene diagramas computarizados anotados, siendo útiles para investigaciones futuras de texto a partir de datos visuales. Esto así, confirma la eficacia que tuvo el proyecto en la conversión de diagramas a texto.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

En un proyecto llamado “Diagram Parse Graphs (DPG)” [9] , al abordar el problema de la interpretación y el razonamiento de diagramas, (el cual es una tarea compleja que implica identificar la estructura de un diagrama y la semántica de sus componentes y sus relaciones), introdujeron el proyecto mencionado, el cual modela la estructura de los diagramas mediante nodos (que representan componentes como ilustraciones, cuadros de texto, flechas y cabezas de flecha) y aristas (que representan relaciones entre estos componentes). Se definió el análisis sintáctico de diagramas como la tarea de inferir DPGs para diagramas y se estudió la interpretación semántica y el razonamiento en el contexto de la respuesta a preguntas sobre diagramas.

Se desarrolló un método basado en LSTM (Long Short-Term Memory) para el análisis sintáctico de diagramas, denominado Deep Sequential Diagram Parser (Dsdp-Net), que selecciona secuencialmente relaciones entre los componentes para construir el DPG. Además, se propuso un modelo de atención basado en DPG para la tarea de responder preguntas sobre diagramas. Para validar su enfoque, se compiló un nuevo conjunto de datos con más de 5,000 diagramas y 15,000 preguntas y respuestas, con anotaciones exhaustivas de los componentes y sus relaciones.

Los resultados del proyecto demostraron que el modelo que se propuso cumplió con el objetivo de interpretar, razonar diagramas y responder preguntas sobre ellos. Asimismo, el modelo Dsdp-Net superó a otros métodos en la identificación de estructuras y relaciones dentro de los diagramas, como también los conjuntos de datos que se desarrollaron en el proyecto, permitiendo evaluar el rendimiento del sistema de manera efectiva, confirmando en sí la utilidad de ellos para el análisis automático de diagramas.

Descripción del Problema

El principal problema identificado es la falta de herramientas automatizadas para la organización y explicación de diagramas y códigos, lo cual dificulta el entendimiento de los estudiantes. Estos suelen enfrentar obstáculos para comprender los códigos presentados en las clases, así como los diagramas desarrollados durante ellas. Además, el miedo a pedir aclaraciones al profesor y la dificultad para buscar y analizar las grabaciones de las clases convierten este proceso en algo complicado y poco eficiente.

Este problema puede empeorar porque en muchas ocasiones, las grabaciones de las clases no cuentan con un sistema que permita una navegación estructurada por los temas tratados (existen plataformas que con IA lo hacen, pero no es el caso de todas), lo que genera frustración y pérdida de tiempo para los estudiantes. Asimismo, el formato digital de los materiales utilizados en las clases virtuales o híbridas no siempre está diseñado para ser accesible o fácil de seguir, lo que aumenta la dificultad en el aprendizaje virtual.

Del mismo modo, los profesores también enfrentan retos importantes al intentar facilitar el aprendizaje. No tienen herramientas que les permitan estructurar de manera clara y accesible el contenido impartido en clase, lo que dificulta la optimización de sus métodos de enseñanza. Esto afecta negativamente en la capacidad de los estudiantes para asimilar temas complejos, como programas de algoritmos y diagramas de programación.

Por lo que, existe la necesidad de diseñar soluciones tecnológicas que no sólo aborden estas dificultades y retos, sino que también se adapten a las nuevas formas de enseñanza y aprendizaje promovidas por la tecnología, las clases en línea y la IA. Estas herramientas deben ser capaces de automatizar la organización y explicación de contenidos complejos, brindar un acceso más sencillo y eficiente a los materiales de clase, y facilitar una mayor interacción entre estudiantes y profesores.

Planteamiento Inicial de la Solución

La solución al problema identificado consiste en desarrollar un bot inteligente integrado a una aplicación web, diseñado para automatizar tareas importantes en la organización, explicación y práctica de contenidos académicos explicados en las sesiones. Este sistema buscará atender tanto las necesidades de estudiantes como de profesores, optimizando el aprendizaje y facilitando la enseñanza en entornos virtuales y mixtos.

La aplicación permitirá a los profesores vincular su perfil de alguna plataforma de reuniones virtuales con el bot, otorgándole acceso a las sesiones de clase para que el bot pueda recopilar y procesar el contenido impartido. Al finalizar cada sesión, el bot procesa la información obtenida para:

1. Generar explicaciones personalizadas: Resumir los puntos principales de la clase, con un enfoque en clarificar diagramas, códigos.
2. Facilitar la navegación de grabaciones: Estructurar las grabaciones de clase en secciones temáticas, permitiendo a los estudiantes acceder rápidamente a los temas específicos de interés.

El bot garantizará la privacidad de los datos de profesores y estudiantes, limitándose al uso exclusivo de la información obtenida para los fines previamente mencionados. Datos irrelevantes serán descartados automáticamente, lo que asegura la protección de la información sensible.

Además, la aplicación web será diseñada para ser intuitiva y accesible, eliminando barreras tecnológicas que puedan obstaculizar su adopción. Los estudiantes podrán interactuar fácilmente con el contenido generado, lo que favorecerá su comprensión y motivación para aprender.

Objetivos Del Proyecto

Objetivo General

Desarrollar un bot automatizado integrado en una aplicación web que utilice herramientas de inteligencia artificial (IA) para facilitar la comprensión sobre programas de algoritmos y diagramas de programación explicados en las clases de la asignatura “Introducción a la Algoritmia” de carácter virtual, mejorando la accesibilidad a los materiales de estudio, la claridad en la explicación de conceptos complejos y la eficiencia en el aprendizaje.

Objetivos Específicos

1. Recolectar las sesiones virtuales grabadas de la asignatura “Introducción a la Algoritmia” que contengan programas de algoritmos y diagramas de programación para identificarlos con el fin de que el bot pueda esquematizar y explicar los procesos generados en dichas sesiones.
2. Implementar un sistema que utilice bibliotecas de IA y emplear explicaciones claras y detalladas de los diagramas de programación y programas de algoritmos presentados en las clases.
3. Evaluar el rendimiento del prototipo del modelo en la detección de programas de algoritmos y diagramas de programación, con el objetivo de obtener los resultados más precisos posibles.
4. Crear una aplicación web para estudiantes y docentes, que permita a los profesores vincular sus perfiles en alguna plataforma de reuniones virtuales para acceder a las clases y los alumnos para poder tener acceso a el contenido (explicaciones, acercamientos similares, ejercicios) generado por el bot de forma automatizada basado en el contenido de la sesión.

Justificación del Proyecto

El auge de las clases virtuales ha transformado el entorno educativo, permitiendo que tanto estudiantes como profesores accedan a contenidos académicos desde cualquier lugar, promoviendo la flexibilidad y adaptabilidad en la enseñanza. Sin embargo, esta dependencia creciente del formato virtual también ha evidenciado desafíos significativos, especialmente en la gestión y comprensión eficiente de los códigos y diagramas explicados en clase.

Los estudiantes suelen enfrentar dificultades para entender completamente las explicaciones proporcionadas por los profesores, lo que puede hacer más difícil el entendimiento de los temas tratados en clase y, a largo plazo, impactar de forma negativa su capacidad para aplicar los conocimientos adquiridos en situaciones futuras [10]. Este desafío resalta la necesidad de soluciones que ayuden a reforzar la claridad y accesibilidad del contenido académico en entornos virtuales.

En respuesta a lo presentado, la creación del Bot de Asistencia Interactivo en Programas de Algoritmos y Diagramas de Programación busca ofrecer una herramienta innovadora que expanda de manera significativa los programas de algoritmos y diagramas de programación presentados en las clases, proporcionando explicaciones más detalladas y contextualizadas por el contenido oficial de la asignatura. Este enfoque no solo facilita el aprendizaje para los estudiantes, sino que también actúa como un apoyo complementario para los profesores al estructurar mejor sus contenidos y métodos de enseñanza.

De la misma forma, este proyecto busca proporcionar una solución que beneficie tanto estudiantes como profesores para poder tener de manera efectiva una explicación más detallada de lo visto en esas líneas de códigos y esos diagramas con conceptos variados. Esto no solo contribuye con el entendimiento de los estudiantes en lo visualizado, sino que también contribuye en el mejoramiento de las explicaciones proporcionadas por el docente en la enseñanza virtual.

Limitaciones del Proyecto

El desarrollo e implementación del bot, enfrentará diversas limitaciones que pueden influir en la funcionalidad, alcance y eficiencia del mismo. A continuación, se describen las principales limitaciones identificadas y su impacto en el proyecto:

1. Privacidad y seguridad: Las grabaciones de las clases incluyen información sensible de estudiantes y docentes, lo que implica cumplir estrictamente con normativas de protección de datos (como GDPR) para evitar violaciones de privacidad.
2. Calidad del contenido capturado: La precisión de los algoritmos de LLM y análisis de video depende de la calidad del contenido capturado. Si las grabaciones de las clases tienen mala calidad de video, esto puede afectar el rendimiento del bot.
3. Actualizaciones y cambios en plataformas externas: La plataforma de videoconferencia con la que el bot debe integrarse podría cambiar su política, interfaz o API, lo que podría afectar la funcionalidad del bot y requerir actualizaciones o modificaciones constantes.
4. Enfoque exclusivo en códigos y diagramas: El bot solo capturará y explicará los programas de algoritmos y diagramas de programación presentados en la clase, excluyendo otros tipos de contenido que pudieran ser parte de la enseñanza del docente, excluyendo del mismo modo el audio.
5. Limitación a la clase de Introducción a la Algoritmia: El bot está diseñado exclusivamente para la clase de Introducción a la Algoritmia, lo que limita su uso a esta asignatura en específica.

Capítulo I - Marco Teórico

1.1 Marco Teórico

Educación en Algoritmos y Programación

El aprendizaje de algoritmos y programación es una clave importante para la formación de futuros desarrolladores y científicos de la computación. De la misma forma, la enseñanza de algoritmos requiere que los estudiantes desarrollen su razonamiento lógico para poder sobreponerse a todos los desafíos del proceso de la programación [11]. Por lo tanto, para mejorar la comprensión y reducir la dificultad inicial del proceso del aprendizaje algorítmico, es esencial utilizar metodologías y herramientas específicas, que permitan al estudiante visualizar de manera concreta los resultados de los programas escritos, así facilitando una base sólida para próximos desafíos y soluciones más sencillas.

El aprendizaje flexible ha demostrado ser una estrategia efectiva en la enseñanza de algoritmos y programación [12]. La implementación de entornos interactivos y recursos de aprendizaje adaptados al perfil de cada estudiante, permite que estos estudien a su propio ritmo y refuerzen conceptos claves asociados a sus primeros pasos en la programación. Estas estrategias complementan un enfoque para la enseñanza y representan un avance para la formación futura de dichos profesionales en el área.

Los diagramas de programación, como los diagramas de flujo, desempeñan un papel crucial en la visualización y comprensión de los algoritmos [13]. Se ha demostrado en investigaciones realizadas por instituciones educativas, que el uso de diagramas facilita la comprensión de los procesos en la enseñanza de programación. La repetición de ejercicios prácticos usando algoritmos, apoyados de diagramas, mejora el razonamiento lógico de los estudiantes permitiéndoles así, entender la ejecución de los programas.

La enseñanza de algoritmos presenta desafíos en el proceso de enseñanza y aprendizaje para los alumnos. Sin embargo, los estudiantes a menudo enfrentan dificultades para poder lidiar con la complejidad de los problemas propuestos o

encontrados, y aprender a usar las tecnologías que actualmente existen. Por lo mismo, una de las tecnologías que puede contribuir a mejorar, y que se adapte al perfil de la enseñanza del estudiante es el Sistema de Tutoría Inteligente (ITS). Según una investigación sobre el uso de ITS en la educación en ciencias de la computación, se han destacado estos sistemas al integrar la inteligencia artificial (IA), permitiendo así personalizar (dependiendo de las necesidades del alumno) el proceso educativo de cada uno [14]. Asimismo, la enseñanza de algoritmos haciendo uso de ITS, permite abordar las dificultades logrando así un enfoque adaptativo y una comprensión más profunda de los conceptos algoritmos a los estudiantes.

Métodos y Herramientas para la Detección de Transiciones en Videos y Captura de Diagramas y Códigos de Algoritmos

Al momento de hablar de un Sistema de Tutoría Inteligente (ITS) integrado con la IA, se debe destacar la necesidad de otorgarle al bot la información únicamente necesaria y válida para asistir con las necesidades de los estudiantes. Para esto, de las sesiones de clases es crucial extraer solo la información relevante que luego será procesada e interpretada por la IA, esto para asegurarse de que al momento de hacer consultas sobre la sesión de clase, no haya información no relacionada o que no aporte nada al estudiante. Para lograr esto es de vital importancia segmentar las sesiones de clases en aquellos trozos importantes que contienen tanto, diagramas y códigos de algoritmos, para esto se necesita poder detectar eficientemente los cambios en los videos de las sesiones de clase detectando cuando la imagen ha cambiado lo suficiente y extraer el contenido relevante.

La detección de transiciones en videos es un proceso tanto crucial para la segmentación del contenido visual y en el análisis de estos. Además, esta implica identificar fotogramas en los que el contenido de la escena va cambiando requiriendo calcular una métrica totalmente apropiada que pueda caracterizar el cambio de contenido de dos fotogramas y un umbral que asimismo determine si ese cambio fue suficientemente grande para definirlo como el corte de la escena [15]. Existen herramientas avanzadas que permiten identificar de forma automática el cambio de escena de cada video, entre ellas se pueden mencionar: la API PySceneDetect vista

antes en el proyecto [7] y Video Intelligence de Google Cloud [16], la cual no solo detecta escenas, sino que también detecta el texto presentado en los videos.

El proyecto , antes visto, utiliza la API PySceneDetect para detectar los cambios de escena de un video. Esta herramienta de código abierto, diseñada para identificar los cortes entre planos, analiza cada imagen y detecta transiciones cuando la diferencia visual entre fotogramas supera un umbral predefinido. Su capacidad de segmentación automática mejora la eficiencia en la edición de contenido visual, permitiendo extraer momentos clave de un vídeo.

Para lograr esta detección, PySceneDetect emplea diferentes algoritmos que analizan variaciones en la composición visual de los fotogramas. Estos métodos pueden basarse en diferencias de píxeles, histogramas de color o características espaciales y temporales. Dependiendo de la técnica utilizada, la detección de transiciones puede clasificarse en enfoques basados en umbrales, diferencias estadísticas o aprendizaje automático.

El detector de escenas basado en contenido (detect-content) [17] , detecta los cortes de un video al analizar la diferencia visual entre fotogramas consecutivos. Asimismo, este método detecta las transiciones o cortes que hay entre cada escena, es decir, evalúa si dos fotogramas adyacentes pertenecen a la misma escena o representan algún cambio significativo. Para que se logre todo lo mencionado, el detector convierte cada ilustración de RGB a HSV y calcula la diferencia de promedio entre sus valores, así si ella supera el umbral predefinido, pues se detecta un cambio de escena.

Además, el detector de escenas, incluye un detector de bordes lo que permite que se pueda identificar de mejor manera cada corte. Para ello, este permite ajustar la sensibilidad a los cambios en color, saturación, brillo y bordes mediante parámetros asignados a eso. Por otra parte, también se encuentra el detector de contenido adaptativo (detect-adaptive) [17] el cual tiene de similar la comparación de la diferencia de contenido entre ilustraciones, pero usa un promedio móvil para cada cambio de cada fotograma, lo que ayuda a que se reduzcan falsas detecciones en las escenas mediante hayan movimientos rápidos en la cámara.

También, existe el detector de escenas basado en umbral (detect-threshold) [17], este identifica si existe cambios en un video comparando el brillo de cada fotograma con un valor predefinido. Si el brillo promedio supera el umbral pues se detecta un corte de escena, PySceneDetect calcula el valor, promediando los colores rojo (R), verde (G) y azul (B) de todos los píxeles, obteniendo así un número entre 0.0 y 255.0 que estaría representando la luminosidad de la imagen.

El algoritmo de detección de cambios de escena [17], analiza la luminancia de cada fotograma en el espacio de color YCbCr para evitar errores causados por posibles cambios de iluminación. De esta forma, se extrae el brillo (canal Y) y se genera un histograma normalizado con un número de divisiones. Luego, se compara el histograma con el del fotograma anterior usando una correlación. Si esa similitud cae por debajo de un umbral pues se detecta un cambio de escena.

Y para finalizar los algoritmos implementados para lograr el PySceneDetector, se encuentra el detector de hash perceptual (detect-hash) [17], el cual calcula un hash para cada fotograma y lo compara con el del fotograma anterior. Si la diferencia entre esos hashes supera el umbral definido con anterioridad, pues se detecta un cambio de escena.

Por otro lado, la detección de fotografías o ilustraciones en imágenes pueden basarse en lo que sería la identificación de áreas rectangulares dentro de una imagen que ocupan un porcentaje del tamaño total de dicha imagen. Si el tamaño de ese rectángulo supera un umbral predefinido, se estaría considerando que esta región corresponde a una fotografía o ilustración. De la misma forma, se estarían almacenando las coordenadas de dichas imágenes para así facilitar en algún momento su análisis realizado posteriormente. Esas técnicas también se utilizan donde se aplican los métodos como el DPG [9] y redes neuronales para interpretar y extraer significado.

El reconocimiento óptico de caracteres (OCR) [18] es una tecnología que permite convertir imágenes de datos editables y procesables. Al escanear un documento o recibo, el archivo resultante es una imagen sin posibilidad de búsqueda o edición. Con OCR, el contenido se extrae y se digitaliza, permitiendo su uso y análisis.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Entre las herramientas más utilizadas para el propósito de OCR, se encuentra Tessaract OCR [19] la cual es de código abierto, mejora el proceso de extracción de texto de imágenes, facilita la comprensión del contenido visual de una ilustración. Su arquitectura sigue un enfoque en cadena de procesamiento que permite la conversión de imágenes de texto digital a través de pre procesamiento y análisis de componentes conectados. Este análisis realiza el paso de segmentar la imagen binarizada en componentes conectados, donde se identifican los contornos de los caracteres y se agrupan en elementos denominados “Blobs”. Asimismo, este permite detectar el texto tanto en negro sobre blanco, como en blanco sobre negro.

Por otra parte, la organización de los Blobs en líneas y palabras, ya identificados los componentes de texto, estos se agrupan en líneas y dentro de cada una de ellas, se diferencian palabras según el tipo de espaciado, donde para cada tipo de texto es diferente: texto con espaciado fijo, los caracteres se segmentan en celdas de igual ancho, texto con espaciado proporcional, Tessaract utiliza un método de detección de espacios basado en medidas estadísticas para identificar los límites entre palabras. De la misma forma, Tessaract incorpora algoritmos que permiten detectar y corregir la inclinación de las líneas de texto sin necesidad de rotar la imagen, manteniendo así la calidad del contenido. Ya para el reconocimiento de caracteres y palabras, se analizan las formas de los caracteres y se comparan con modelos pre-entrenados, asignando puntuaciones de confianza a cada posible interpretación. Una vez reconocidos los primeros caracteres, Tessaract utiliza los resultados para entrenar un clasificador adaptativo en tiempo real. Este mejora el reconocimiento de las siguientes líneas de texto al aprender patrones específicos del documento en cuestión, aumentando así la precisión en textos con fuentes no tan comunes y variaciones en la tipografía.

En los casos donde el espaciado entre caracteres es poco, Tessaract revisa las posibles interpretaciones y refina la segmentación de palabras, analiza la altura de cada carácter para así diferenciar entre sí es una letra mayúscula o minúsculas, como también mayúsculas pequeñas. Por último, Tessaract, genera el texto final en formato digital, permitiendo exportar los resultados en diferentes formatos.

Técnicas y Algoritmos para el Procesamiento de Imágenes, Detección de Diagramas y Programas de Algoritmos

En la actualidad, al acercarnos al término “procesamiento de imágenes”, nos encontramos una serie de técnicas y procesos que permiten analizar, modificar e interpretar imágenes. El análisis de imágenes busca extraer información importante del contenido visual, no buscando mejorar la calidad, lo que lo hace esencial para la detección de objetos y la clasificación de imágenes [20]. La visión por computadora, como una extensión del procesamiento de imágenes, tiene el objetivo de crear sistemas artificiales capaces de interpretar el mundo visual de manera autónoma, aplicándose en áreas como el reconocimiento de patrones, la segmentación de imágenes y la navegación autónoma [20].

Para lograrlo, se pueden emplear métodos avanzados como los modelos ocultos de Markov (HMMs), árboles de decisión y análisis de componentes principales (PCA), los cuales han demostrado ser eficaces en la identificación de patrones tanto en imágenes como en señales de audio y texto [20]. Previamente se presentaron diversas implementaciones para lograr procesar, interpretar y entender algo tan complejo como lo puede llegar a ser un diagrama de flujo (de algoritmos) y como se puede obtener resultados categorizables en diversas clases que luego permiten crear relaciones como: conexiones, datos, decisiones, terminaciones, flechas, textos, procesos entre otras.

El Deep Learning se ha utilizado para abordar tareas complejas en el procesamiento de imágenes y la interpretación de diagramas. Los modelos más avanzados basados en redes neuronales convolucionales (CNN) han demostrado ser eficaces en la clasificación y detección de objetos, lo que es esencial para tareas como la interpretación de diagramas de flujo de algoritmos. En , se presentó el uso de modelos de detección de objetos como “Faster R-CNN” (que es una arquitectura de detección de objetos basada en redes neuronales convolucionales que utiliza una Red de Propuestas de Regiones (RPN) para generar regiones candidatas de objetos y

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

posteriormente clasificar y refinar sus posiciones), extendido con “Red Neuronal Pirámide de Características (FPN)” e interceptado con “Inception-ResNet-v2” (una red neuronal que combina los bloques Inception, que mejoran la eficiencia computacional capturando diferentes tamaños de características en paralelo, con conexiones residuales de ResNet, que facilitan la propagación del gradiente y permiten redes más profundas sin pérdida de información). Este enfoque ha sido utilizado para obtener clases que permiten categorizar componentes y crear relaciones en los diagramas.

En el ámbito de la interpretación secuencial de diagramas, se ha demostrado el uso de redes neuronales como “Deep Sequential Diagram Parser Network (Dsdp-Net)” en . Este modelo es una red neuronal profunda diseñada para analizar diagramas de manera secuencial. Su objetivo es identificar y comprender las relaciones entre los elementos que componen un diagrama, construyendo así un grafo que representa su estructura lógica. Para lograr esto, Dsdp-Net emplea una arquitectura que integra una red neuronal recurrente de tipo LSTM (Long Short-Term Memory), que facilita la memoria a largo plazo necesaria para capturar y modelar las dependencias secuenciales y espaciales presentes en los diagramas.

El procesamiento de imágenes, y más específicamente la visión por computadora, es una disciplina fundamental para la interpretación automática de imágenes y diagramas. En la actualidad, este tipo de tecnología está en el corazón de muchas aplicaciones de IA. El procesamiento de imágenes busca analizar, modificar e interpretar imágenes, extrayendo información clave sin alterar la calidad visual. Como se mencionó, la visión por computadora tiene como objetivo crear sistemas artificiales capaces de interpretar el mundo visual de manera autónoma, aplicándose en tareas como el reconocimiento de patrones, la segmentación de imágenes y la navegación autónoma .

Una de las herramientas más importantes en visión por computadora es el Reconocimiento Óptico de Caracteres (OCR), que se utiliza para extraer texto de imágenes. En , se utilizó la API de “Naver Clova OCR” para realizar dicho

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

reconocimiento de las imágenes y extraer el contenido encontrado en los diagramas, lo que permite llevar los diagramas a un formato que pueda ser comprendido por el sistema. Además, en [8] se realizó la comparación de modelos como “Faster R-CNN”, que es un modelo avanzado de detección de objetos que introduce una Red de Propuestas de Regiones (RPN) para mejorar la velocidad y precisión en la identificación de objetos en imágenes. Este modelo genera automáticamente regiones de interés mediante la RPN, lo que elimina la necesidad de métodos externos.

El procesamiento de imágenes también requiere de técnicas matemáticas para mejorar la precisión en la detección de formas y relaciones. En [8] se presenta la Transformada de Línea de Hough, una técnica popular para detectar cualquier forma que pueda representarse matemáticamente, especialmente útil en la detección de líneas o curvas en imágenes de diagramas, y que está disponible en la biblioteca OpenCV.

El Natural Language Processing (NLP) [21] se utiliza para procesar y comprender el lenguaje natural, lo cual es esencial cuando se generan explicaciones a partir de los diagramas o cuando se convierte texto visual en información estructurada. En , se presentó un enfoque utilizando técnicas de redes neuronales para transformar diagramas en descripciones textuales. Esto se logra mediante el uso de arquitecturas como las redes neuronales recurrentes (RNN) y LSTM, que permiten interpretar las relaciones espaciales y secuenciales entre los elementos en un diagrama.

Una de las tareas más complejas dentro del procesamiento de diagramas es mantener las relaciones precisas entre los distintos elementos visuales. Esto implica reconocer correctamente las conexiones, flechas y la estructura espacial de los diagramas. En se menciona que las capacidades actuales de los sistemas computacionales aún no alcanzan la eficiencia del sistema visual humano, lo que plantea un reto continuo para el desarrollo de nuevas técnicas en este campo. No obstante, al combinar las capacidades de las redes neuronales y las técnicas de visión por computadora, como la Transformada de Hough, una técnica fundamental en procesamiento de imágenes y visión por computadora, utilizada para detectar formas geométricas específicas, como

líneas rectas, círculos o elipses, en imágenes digitales. Su principio básico consiste en transformar puntos en el espacio de la imagen a un espacio de parámetros, facilitando la identificación de estas formas mediante un procedimiento de votación, se pueden mejorar estos procesos y permitir una interpretación más precisa de los diagramas .

Interpretación de Información Procesada de Diagramas y Programas de Algoritmos

Una tarea igual o más compleja que el procesamiento de imágenes, detección de diagramas y programas de algoritmos posiblemente es interpretar la información que se pudo clasificar, ya sea en clases o en el formato de salida obtenido, y luego transformarla en explicaciones claras y comprensibles para los usuarios. Esta interpretación requiere no solo extraer datos estructurados de los diagramas, sino también contextualizar su significado, identificar relaciones clave entre los elementos y generar respuestas coherentes que faciliten el aprendizaje. Además, implica la implementación de modelos avanzados que puedan traducir la representación visual en descripciones textuales precisas, asegurando que el contenido pueda ser comprendido por estudiantes y docentes en diversos niveles de conocimiento.

El enfoque presentado en [8] toman esas clases y las termina transformando en una tripleta “<Cabeza, Relación, Cola>” con el cual pueden interpretar el diagrama y realizar una explicación de este. Para lograr este cometido asocian el extremo con la forma más cercana. El acercamiento que llevan a cabo para la interpretación en [9] es realizado del mismo modo con redes neuronales, en específico “Dqa-Net”, dicho modelo es capaz de codificar el grafo generado del diagrama en hecho, y este aprende a asistir en los hecho relevantes, logrando ser capaz de responder preguntas acerca de los diagramas.

Una vez extraída la información estructurada de los diagramas y programas de algoritmos, es fundamental contar con técnicas de representación semántica que permitan una interpretación más cercana al razonamiento humano. Modelos como Graph Neural Networks (GNNs) [22], que son un tipo de red neuronal diseñada específicamente para procesar datos en forma de grafos , han demostrado ser eficaces

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

para capturar las relaciones entre los nodos en un grafo representativo del diagrama, facilitando la generación de explicaciones coherentes sobre su estructura y significado. Además, enfoques basados en transformers, como los utilizados en Vision-Language Models (VLMs) [23] que son modelos de inteligencia artificial que integran información visual (de imágenes o videos) con información lingüística (de texto), permitiendo que el modelo entienda y genere respuestas que combinan ambos tipos de datos. Estos modelos se entrena para asociar de manera conjunta el contenido visual y textual, lo que les permite realizar tareas que requieren una comprensión simultánea de ambos dominios, como responder preguntas sobre imágenes, generar descripciones de escenas o realizar tareas de búsqueda multimodal. Pueden integrar la información visual y textual para generar descripciones automáticas contextualizadas. La combinación de estas técnicas con modelos como Dqa-Net [9], que aprende a responder preguntas sobre diagramas mediante codificación de grafos, mejora la comprensión automatizada y la interacción con los usuarios. Estos avances permiten que los sistemas no sólo identifiquen y clasifiquen componentes de diagramas, sino que también generen explicaciones detalladas, haciendo que la interpretación de algoritmos visuales sea más accesible para estudiantes y profesionales.

Generación Automática de Información Interpretada con Lenguaje Natural

En el ámbito de la inteligencia artificial, el lenguaje natural y la generación automática de explicaciones, juegan un papel crucial en la interacción de humanos y computadoras. La capacidad que tiene que generar explicaciones comprensibles a partir de datos procesados, permite mejorar la transparencia y accesibilidad de los sistemas inteligentes, facilitando así su aplicación en diversas áreas principalmente la educación. Para lograr ese objetivo, es fundamental el uso del Procesamiento de Lenguaje Natural (PLN) [21], conocido también como NLP por sus siglas en inglés (Natural Language Processing), el cual es referido a la capacidad que tienen las máquinas para interpretar, analizar y generar el lenguaje humano. Asimismo, esta tecnología facilita la comprensión y estructuración del contenido textual, permitiendo así la transformación del lenguaje en representación inteligibles para usuarios [24]. Un caso particular para de aplicación de PLN es el OCR [18], donde la herramienta ya

vista Tesseract OCR [19], posibilitan la conversión de texto en imágenes a formatos editables.

Por otro lado, técnicas avanzadas de PNL, como T5 (Text-to-Text Transfer Transformer) y BART (Bidirectional and Auto-Regressive Transformers), muestran un gran potencial en lo que es la generación de explicaciones de algoritmos y diagramas de programación. De la misma forma, T5 [25] es un modelo de lenguaje basado en transformadores, donde la misma usa un modelo de codificador-decodificador. Este convierte cualquier tarea de PNL en una de conversión de texto a texto. En lugar de tratar diversas tareas de manera aislada, T5 transforma todas las entradas y salidas en secuencias de texto.

Ahora bien, T5 sigue el diseño de encoder-decoder, lo que significa que el encoder recibe el texto de entrada, lo analiza y lo convierte en una representación numérica comprensible para la máquina. El decoder, usa esta información para generar la respuesta de manera progresiva, prediciendo una palabra a la vez hasta completar la salida. Ya para entrenarse, el T5 pasa por dos fases, primero el pre-entrenamiento. El aprende sobre el lenguaje en sí, utilizando una técnica llamada “Span Corruption”, donde básicamente, partes del texto se eliminan y el modelo debe de adivinarlas, lo que así le permite entender patrones del lenguaje. Luego, en la fase de fine-tuning, se realizan tareas más concretas, como la traducción, clasificación de texto o generación de respuestas.

Por otro lado, BART [26], es otro modelo basado en transformadores que combina características de modelos de codificación-decodificación bidireccionales y modelos autoregresivos. Su arquitectura al ser basada en el codificador bidireccional, que analiza la información teniendo en cuenta el contexto completo, y decodificador autorregresivo, que genera texto de manera progresiva, este permite que aprenda mejor la estructura del lenguaje y genere las explicaciones más fluidas.

BART [26] se entrena eliminando las partes del texto de entrada y luego enseñándole a reconstruirlo. Durante esta fase, el modelo recibe versiones modificadas de los textos originales, donde se aplican diferentes tipos de alteraciones, como eliminando palabras, permutando oraciones y enmascarando fragmentos completos, generando así

explicaciones coherentes. Así, BART en la generación de texto es altamente efectivo, donde el puede crear resúmenes, traducir de forma automática, responder a preguntas y generar explicaciones, la cual la última mencionada puede servir de manera adecuada usando técnicas anteriormente mencionadas.

Por otra parte, modelos de captioning como Show, Attend, and Tell (SAT) han demostrado ser efectivos en las generaciones de descripciones de imágenes mediante mecanismos de atención visual [27]. Su principal innovación es que, en lugar de procesar una imagen como un todo en sí, el aprende a enfocar su atención en diferentes partes de la imagen, mientras genera cada palabra de la descripción. El SAT se divide en tres etapas, primero la imagen de entrada es procesada por redes neuronales convolucionales (CNNs) para extraer características visuales de la imagen. Luego, esta información es pasada a redes neuronales recurrentes (RNNs) para generar la descripción en lenguaje natural, palabra por palabra. Durante esa fase, se emplea el mecanismo de atención, lo que permite que el modelo seleccione dinámicamente qué partes de la imagen son más relevantes en cada momento de la generación de texto. Entre los tipos de mecanismo de atención que utiliza SAT está el suave (soft attention), el cual asigna pesos continuos en todas las regiones de la imagen, permitiendo que se enfoque en múltiples áreas al mismo tiempo, y el duro (hard attention) el cual selecciona solamente una única región de la imagen en cada paso de la generación del texto. A medida que el modelo genera cada palabra de la descripción, el mecanismo de atención ajusta su enfoque para centrarse en los objetos y detalles relevantes, logrando que se genere un texto descriptivo y preciso.

Aplicación de Técnicas y Algoritmos Previos en Plataformas como Servicio.

Para las tareas mencionadas anteriormente, se pueden utilizar las siguientes herramientas y/o librerías ya como funcionamiento de servicio tanto de manera de pago como versiones gratuitas de las mismas con usos limitados. De todos modos, sirven para ejemplificar las aplicaciones que se le puede llegar a dar al uso correcto de los algoritmos, librerías y tecnologías.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Textractify [28] es una forma rápida de extraer datos numéricos de videos o muchas fotos y exportarlos en formato de hoja de cálculo. Extrae y analiza automáticamente los datos numéricos que se muestran en los vídeos para ahorrar tiempo y mejorar la precisión. Limpia fácilmente los datos extraídos para centrarse en los datos exactos de interés antes de exportarlos [29]. El funcionamiento de esta herramienta parte del uso de la detección de cambios de frames (o un ajuste personalizado de frames) y utilizando reconocimiento óptico de caracteres (OCR) para extraer el fragmento de código de programación (aunque la herramienta dice estar especializada para datos numéricos y tablas).

BlackBox [30] AI es una herramienta de inteligencia artificial diseñada para ayudar a los desarrolladores a codificar de manera más eficiente, proporcionando fragmentos de código y asistencia en tiempo real. Funciona como un asistente de codificación que optimiza el proceso de programación. Sirve para facilitar el proceso de programación al permitir a los desarrolladores encontrar y utilizar fragmentos de código de manera eficiente. También ayuda a automatizar tareas, permitiendo que los programadores se concentren en aspectos más creativos y complejos de su trabajo [31]. Su funcionamiento es a partir del uso de técnicas avanzadas de procesamiento de lenguaje natural (NLP) y OCR .

OpenCV [32] (Biblioteca de visión por computadora de código abierto) es una biblioteca de software de aprendizaje automático y visión por computadora de código abierto. OpenCV se creó para proporcionar una infraestructura común para aplicaciones de visión por computadora y acelerar el uso de la percepción automática en los productos comerciales [33]. Dentro de sus muchas herramientas se pueden conseguir formas de detectar formas geométricas .

Pixelcode AI [34] convierte capturas de pantalla de código en código editable, impulsado por IA. Extrae fragmentos de código de tutoriales en vídeo en cuestión de segundos. Se puede seguir un tutorial e implementar el código por su cuenta podría consumir mucho tiempo rápidamente, pero no tiene por qué ser así. Simplemente es tomar una captura de pantalla del fragmento de código que se desea extraer y luego

pegarlo en un editor de código. Asimismo como las demás herramientas hace uso del OCR para poder detectar con eficiencia el código presente en pantalla.

1.2. Definición de Términos y Glosario

Tecnologías

- Inteligencia Artificial (IA): Rama de la ciencia dedicada al desarrollo de sistemas y dispositivos capaces de razonar, aprender y tomar decisiones de manera similar a la inteligencia humana o en situaciones donde la cantidad de datos supera la capacidad de análisis humano [35].
- Procesamiento de Imágenes / Visión por Computadora: La visión por computadora permite a las máquinas interpretar y tomar decisiones a partir de imágenes o videos. Por otro lado, el procesamiento de imágenes se enfoca en modificar o mejorar una imagen para hacerla más nítida o estéticamente atractiva, sin analizar su significado [36].
- Procesamiento de Lenguaje Natural (NLP): Es una tecnología basada en aprendizaje automático que permite a las computadoras analizar, interpretar y comprender el lenguaje humano [37].
- OCR (Reconocimiento Óptico de Caracteres): Tecnología que permite convertir imágenes de texto en texto digital y editable [38].
- Cola de mensajería: Mecanismo de comunicación asíncrona utilizado entre servicios, especialmente en arquitecturas de microservicios o sin servidor. Los mensajes enviados se almacenan temporalmente en una cola hasta que son recibidos y procesados por el servicio correspondiente [39].
- Celery Flower: Herramienta web para monitorear tareas distribuidas en Celery. Proporciona una interfaz visual que permite observar en tiempo real el estado de los trabajadores, el historial de tareas y su progreso dentro de un clúster [40].

Hardware

- CPU y GPU de alto rendimiento: La unidad central de procesamiento (CPU) puede incluir múltiples núcleos y es comúnmente conocida como el “cerebro” de la computadora. Por su parte, la unidad de procesamiento de gráficos (GPU) cuenta con una gran cantidad de núcleos más pequeños y especializados, que trabajan en conjunto para distribuir las tareas de procesamiento de manera paralela, logrando un alto rendimiento [41].
- Infraestructura en la nube: Conjunto de recursos de hardware y software que sustentan los servicios en la nube. Los proveedores de estos servicios gestionan centros de datos a nivel global, integrando miles de elementos de infraestructura de TI, como servidores, dispositivos de almacenamiento y equipos de red [42].

Lenguajes de Programación

- Python: lenguaje de programación muy utilizado en el desarrollo de software, aplicaciones web, ciencia de datos y aprendizaje automático (ML) [43].

Herramientas

- SuryaOCR: Aplicación de Reconocimiento Óptico de Caracteres (OCR) basada en Python, diseñada para extraer texto de imágenes [44].
- PySceneDetect: Herramienta para detectar automáticamente cambios de escena en videos .
- DINOv2: Sobresale en sistemas de recuperación de imágenes, donde el objetivo es encontrar imágenes similares dentro de una gran base de datos [45].
- YOLOv12: Presenta una arquitectura enfocada en la atención, apartándose de los enfoques tradicionales basados en CNN que se usaban en modelos YOLO previos, pero manteniendo la velocidad de inferencia en tiempo real, crucial para diversas aplicaciones [46].

Topologías de Redes

- Cliente-servidor: Modelo de comunicación que facilita la distribución de tareas entre diferentes equipos dentro de una red de computadoras. En este modelo, el cliente realiza solicitudes de servicios o recursos, mientras que el servidor responde a estas solicitudes proporcionando los servicios o datos requeridos [47].

Protocolos

- HTTP/HTTPS: HTTP es un protocolo de comunicación utilizado para la transferencia de datos entre clientes y servidores en la web, sin proporcionar seguridad en la transmisión. HTTPS es una versión segura de HTTP que utiliza tecnologías como SSL y TLS para cifrar los datos, protegiendo la comunicación contra interceptaciones y alteraciones [48].
- API REST: Una API REST es una interfaz de programación de aplicaciones que se basa en los principios de la arquitectura REST (Transferencia de Estado Representacional), diseñada para interactuar con servicios web. REST define un conjunto de reglas y mejores prácticas para la creación de APIs eficientes y escalables, facilitando la comunicación entre sistemas a través de solicitudes HTTP [49].

Técnicas de Seguridad

- Cumplimiento de GDPR: Reglamento General de Protección de Datos (GDPR) es una normativa adoptada por el Parlamento Europeo, el Consejo de la Unión Europea y la Comisión Europea para fortalecer y unificar la protección de datos de todos los individuos dentro de la Unión Europea (UE). También regula la exportación de datos personales fuera de la UE, buscando otorgar a los ciudadanos y residentes mayor control sobre sus datos personales y simplificar el marco regulatorio para los negocios internacionales al armonizar las regulaciones dentro de la UE [50].

Señales a Procesar

- Video (frames): Secuencias de imágenes extraídas de grabaciones de clases [51].

Servicios Vistos

- Google Cloud: Ofrece servicios como análisis de video, conversión de texto a voz y detección de escenas [52].
- Microsoft Azure Cognitive Services: Conjunto de APIs basadas en la nube desarrolladas por Microsoft para integrar capacidades de inteligencia artificial en aplicaciones. Estos servicios permiten añadir funcionalidades como reconocimiento de imágenes, análisis de lenguaje natural, traducción automática y reconocimiento de voz a tus soluciones, sin necesidad de tener habilidades avanzadas en aprendizaje automático [53].
- API OpenAI: Interfaz que permite integrar fácilmente los modelos de inteligencia artificial de OpenAI en aplicaciones propias. Funciona como un puente que facilita el uso de capacidades avanzadas como la generación y comprensión de texto, sin necesidad de conocer los detalles técnicos del modelo [54].

Técnicas Estadísticas

- Evaluación de modelos: Medición de la precisión, eficiencia y efectividad de los modelos de IA implementados.

2. Solución Propuesta

2.1. Definición del Proyecto

El presente proyecto tiene como objetivo principal desarrollar un bot automatizado capaz de procesar clases virtuales para extraer, analizar y organizar el contenido relacionado con código y diagramas. Para lograrlo, el bot analiza sesiones virtuales grabadas, seleccionando aquellas que contienen programas de algoritmos y diagramas de programación. Una vez identificados, los procesa y estructura la información de manera didáctica. Además, se ha integrado un sistema basado en inteligencia artificial que permite generar explicaciones claras. El desarrollo también incluye una interfaz web, donde los estudiantes pueden acceder a los informes generados de manera organizada e interactiva.

2.2. Productos del Proyecto

El proyecto se basa en el desarrollo de un bot automatizado capaz de reconocer código y diagramas a partir de videogramaciones de la materia Introducción a la Algorítmica. Este bot no solo identifica y extrae estos elementos, sino que también genera explicaciones detalladas sobre su funcionamiento, facilitando así la comprensión de los conceptos por parte de los estudiantes. De la misma forma, en este pre-proyecto se estará entregando la implementación inicial de un sistema de segmentación de videos utilizando PySceneDetect para identificar cambios de escena y extraer frames clave, reduciendo significativamente la cantidad de imágenes a procesar. Además, se incluirá el uso de YOLOv12 para la clasificación de imágenes con el fin de identificar bloques de código y diagramas en los frames obtenidos. Se documentará el proceso de preprocesamiento de videos, configuración de parámetros para la segmentación, construcción de datasets en Roboflow y entrenamiento del modelo de detección. También se presentarán métricas de evaluación preliminares y ejemplos visuales de los resultados obtenidos. Para lograr estos objetivos, se han definido las siguientes funcionalidades:

- Desarrollar un bot automatizado integrado en una aplicación web: El sistema utiliza herramientas de inteligencia artificial (IA) para facilitar la comprensión

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

de programas de algoritmos y diagramas de programación explicados en las sesiones virtuales de la asignatura “Introducción a la Algoritmia”. Este desarrollo permite que el proceso de aprendizaje se vuelva más interactivo y adaptativo, brindando apoyo personalizado durante el estudio.

- Recolectar las sesiones virtuales grabadas: Se enfoca en identificar y extraer aquellas sesiones que contengan programas de algoritmos y diagramas de programación, con el fin de que el bot pueda esquematizar y explicar los procesos presentados. Esta recolección se realiza de forma sistemática, garantizando que el contenido relevante sea procesado para su posterior análisis y explicación.
- Implementa un sistema con bibliotecas de IA: Utiliza dichas bibliotecas para generar explicaciones claras y detalladas de los diagramas y programas detectados durante las clases. La integración de estas herramientas permite que la explicación se adapte al contexto de cada ejemplo, facilitando la comprensión de conceptos complejos de manera didáctica.
- Evalúa el rendimiento del prototipo: Se verifica la precisión en la detección y explicación de los contenidos, garantizando que los resultados sean lo más precisos posible. Esta evaluación continua permite identificar oportunidades de mejora y ajustar los algoritmos para mantener un alto estándar en la calidad de la explicación ofrecida.
- Crea una aplicación web para estudiantes y docentes: Permite que los profesores vinculen sus perfiles en plataformas de reuniones virtuales y que los alumnos accedan al contenido (explicaciones, acercamientos similares, ejercicios) generado de forma automatizada por el bot. La aplicación facilita un entorno colaborativo donde tanto docentes como estudiantes pueden interactuar con el material educativo de manera ágil y organizada.

Las funcionalidades no incluidas:

- No capta otros tipos de contenido: El bot está diseñado exclusivamente para capturar y explicar programas de algoritmos y diagramas de programación, excluyendo otros elementos que pudieran formar parte de la enseñanza. Esto asegura que el enfoque se mantenga en lo esencial, evitando confusiones y dispersión en el procesamiento de información irrelevante.
- No abarca otras asignaturas: Su funcionamiento se limita a la clase de “Introducción a la Algoritmia”, por lo que no se aplicará a contenidos o asignaturas diferentes. Esta limitación permite concentrar los recursos y el desarrollo del sistema en una materia específica, optimizando el rendimiento y la relevancia de las explicaciones generadas.
- No garantiza rendimiento óptimo ante baja calidad de grabaciones: La precisión de los algoritmos y del análisis de video depende de la calidad del contenido capturado; grabaciones de mala calidad pueden afectar el desempeño del bot. En estos casos, el sistema podría experimentar dificultades para detectar correctamente los elementos a explicar, lo que subraya la importancia de contar con material de calidad.
- No asume integraciones automáticas con plataformas en constante cambio: Las actualizaciones, cambios en políticas, interfaces o APIs de las plataformas de videoconferencia requerirán ajustes y modificaciones constantes, ya que el bot no se adapta de forma automática a estos cambios. Esta limitación implica la necesidad de intervenciones técnicas periódicas para mantener la compatibilidad y funcionalidad del sistema.
- No omite consideraciones de privacidad y seguridad: Dado que las grabaciones contienen información sensible de estudiantes y docentes, el proyecto debe cumplir estrictamente con normativas de protección de datos (como GDPR), lo que implica no vulnerar la privacidad. Esta restricción

asegura que, a pesar de la automatización y el procesamiento de datos, se mantenga un compromiso firme con la seguridad y confidencialidad de la información.

2.3. Cronograma del Proyecto

Para el desarrollo del Pre Proyecto y Proyecto Final, se estableció un cronograma detallado que nos permitió organizar y distribuir las actividades de manera eficiente, asegurando el cumplimiento de cada etapa dentro de los plazos definidos. Cada cronograma incluye la fecha de inicio y finalización de cada actividad, su duración en días, la categoría a la que pertenece.

Tabla 1

Actividad	Inicio	Duración en días	Fin	Categoría
Revisión de la literatura, estudios previos y bibliografía	27/01/2025	5	31/01/2025	Marco Teórico
Redactar la información interpretada	01/02/2025	6	06/02/2025	Marco Teórico
Realizar la definición del proyecto	05/02/2025	4	08/02/2025	Marco Teórico
Delimitar el proyecto, qué hace o qué no hace	07/02/2025	3	09/02/2025	Metodología
Prototipo: Recopilación y análisis de características de videos de prueba	05/02/2025	5	09/02/2025	Prototipo
Prototipo: Extracción del contenido de los videos a utilizar	10/02/2025	4	13/02/2025	Prototipo
Definir de demostración	14/02/2025	3	16/02/2025	Metodología

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Realizar el análisis y diseño del proyecto	14/02/2025	3	16/02/2025	Metodología
Prototipo: Evaluación de frames relevantes detectando programas de algoritmos y diagramas de programación	14/02/2025	3	16/02/2025	Prototipo
Presentar avances 1 del prototipo	10/03/2025	1	10/03/2025	Prototipo
Prototipo: Construcción de programas de algoritmos y diagramas de programación recopilados en los frames, entrenamiento del modelo, pruebas y ajustes	17/02/2025	21	09/03/2025	Prototipo
Documentar la implementación	10/02/2025	49	30/03/2025	Documentación
Presentar avances prototipo 2	24/03/2025	1	24/03/2025	Prototipo
Entregar informe final	02/04/2025	1	02/04/2025	Documentación

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Backend HTTP	19/05/2025	43	30/06/2025	Backend
Comunicación entre backend y modelo	19/05/2025	13	31/05/2025	Backend
Interfaz Visual	26/05/2025	36	30/06/2025	Interfaz Visual
Presentar avances 1 del prototipo	09/06/2025	1	09/06/2025	Prototipo
Conectar servidor	09/06/2025	22	30/06/2025	Prototipo
Presentar avances 2 del prototipo	30/06/2025	1	30/06/2025	Prototipo
Comunicación entre plataforma web y plataforma de video conferencia	01/07/2025	8	08/07/2025	Plataforma de llamada
Despliegue y pruebas generales	08/07/2025	6	13/07/2025	Prototipo
Presentar avances 3 del prototipo	14/07/2025	1	14/07/2025	Prototipo
Mejoras del modelo	17/05/2025	73	28/07/2025	Prototipo
Entrega del Artículo sobre Proyecto	16/07/2025	1	16/07/2025	Artículo
Entrega del Informe Final del Proyecto	16/07/2025	1	16/07/2025	Documentación
Documentar avances del Proyecto	02/05/2025	75	15/07/2025	Documentación

Cronograma del proyecto del Pre-proyecto y Proyecto Final.

2.4. Plan de Administración de Riesgos

La planificación y gestión de riesgos es una parte fundamental en el desarrollo de cualquier proyecto, ya que permite anticipar y mitigar posibles amenazas que podrían comprometer el cumplimiento de sus objetivos. Un riesgo es un evento o condición incierta que, de ocurrir, podría afectar de manera negativa (o incluso positiva) el

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

desempeño del proyecto. Por ello, es crucial identificar los riesgos potenciales, evaluar su impacto y definir estrategias para gestionarlos de manera efectiva. Estos riesgos suelen dividirse en tres categorías principales:

Tabla 2

Tipo de Riesgos	Código
Riesgos del Proyecto	RP
Riesgos Técnicos	RT
Riesgos Externos	RE

Leyenda de tipos de Riesgo

Los riesgos del proyecto se refieren a eventos o condiciones inciertas que pueden afectar sus objetivos, ya sea de manera positiva o negativa. Los riesgos técnicos surgen cuando hay dificultades en la implementación de tecnologías, fallos en los sistemas o falta de experiencia técnica en el equipo. Por otro lado, los riesgos externos son aquellos factores fuera del control del equipo, como cambios en el mercado, nuevas regulaciones o situaciones imprevistas.

2.4.1. Enfoque metodológico para la evaluación de riesgos

El proceso de evaluación de riesgos en este proyecto se basa en un enfoque metodológico que permite identificar, analizar y gestionar los riesgos que pueden afectar el desarrollo y éxito del mismo. El objetivo principal es minimizar la probabilidad de ocurrencia de estos riesgos o mitigar sus impactos mediante un conjunto de acciones estratégicas. Asimismo, para una mejor organización de estos, se han identificado y categorizado los principales riesgos que pueden afectar el desarrollo del proyecto. Cada riesgo ha sido asignado a un código y se ha clasificado según su tipo.

Listado de Riesgos

Tabla 3

Subsistema afectado	Riesgos	Causa	Código
General	Mal entendimiento de los requerimientos	Falta de conocimientos sobre cómo obtener los requerimientos	RP01
General	Elección de patrones de diseño no convenientes	Apresuramiento en la elección del patrón	RT01
General	Cronograma no realista	Subestimación de las dimensiones del proyecto	RP02
General	Inexperiencia en manejo de proyectos	Nunca haber trabajado en proyectos de esa envergadura que requieran un nivel de documentación y planificación mayo	RP03
General	Requerimientos y funciones desarrolladas no coinciden	Falta de entendimiento en los requisitos	RT02

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

General	Falta de conocimientos o habilidades de los involucrados	Tecnologías o temas nuevos, desconocidos por los involucrados.	RP04
Desarrollo del subsistema web	Falta de entendimiento en los principios de diseño e implementación de aplicaciones web	Poca experiencia en el diseño e implementación de aplicaciones web	RT03
OCR	Calidad de la transmisión del docente: La calidad de video puede ser insuficiente para capturar detalles.	La conexión a internet del docente es inestable	RT04
Integración con plataformas	Compatibilidad a largo plazo: Riesgo de que el bot no sea compatible con futuras versiones de las plataformas llamadas	Cambios en las APIs de las plataformas de llamadas o en las tecnologías subyacentes que podrían afectar la integración.	RT05

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Infraestructura	Exceso de demanda de recursos: OCR y LLMs requieren mucha capacidad computacional y ancho de banda.	Recursos limitados y alta demanda de procesamiento en tiempo real.	RT06
Calidad de los Informes Generados	Generación de informes incompletos o inexactos, lo cual podría generar confusión en los estudiantes.	Errores en el reconocimiento de voz o en la interpretación de temas discutidos.	RT07
Latencia en la Generación de Informes	Demora en la disponibilidad de los informes al final de la sesión, lo que puede reducir la utilidad para los estudiantes.	Procesamiento lento de OCR y LLM debido a limitaciones de hardware o eficiencia de código.	RT09
Rendimiento de LLMs en Tiempos de Pico	Los modelos de lenguaje pueden experimentar caídas de rendimiento o tiempos de	Limitación de la infraestructura para soportar procesamiento intensivo en tiempo real.	RT10

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

	respuesta lento en horarios de alta demanda.		
Errores en la Segmentación de la Sesión	Fallos al dividir la sesión en segmentos temáticos claros, dificultando la organización del informe.	Falta de precisión en algoritmos de segmentación automática o errores en el reconocimiento de texto.	RT11
Consumo Excesivo de Memoria	Alto consumo de memoria por los modelos de OCR y LLM, que puede llevar al sistema a ralentizarse o bloquearse.	Modelos pesados y optimización inadecuada de la memoria.	RT12
Pérdida de Datos Durante la Transmisión	Pérdida de fragmentos de información durante la transmisión en vivo, lo que podría afectar la	Fallos en la transmisión de datos por conexiones de red inestables.	RT13

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

		integridad del informe final.		
UX/Comportamiento del usuario	Reducción del compromiso de estudiantes: Podrían disminuir su atención en clases al saber que habrá un informe.	Expectativa de confiar solo en los informes generados y no en la participación activa.	RP05	
Percepción del usuario	Riesgo ético de reemplazo de docentes: Percepción de que el bot podría reemplazar la enseñanza directa.	Interpretación errónea de los objetivos del proyecto como un reemplazo de la interacción docente.	RP06	
Adaptabilidad a Diferentes Formatos de Clases	Adaptabilidad a Diferentes Formatos de Clases	Variabilidad en los formatos de enseñanza y dificultad para el bot de distinguir entre escenas de algoritmos o diagramas simultáneamente.	RP07	
Seguridad y	Privacidad y	Captura de datos	RE01	

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Privacidad	Seguridad de Datos: Riesgo de capturar y almacenar información sensible durante las grabaciones.	personales sin medidas de protección adecuadas o sin consentimiento explícito.	
Obsolescencia tecnológica	Evolución de la tecnología y dependencia: El software podría volverse obsoleto.	Avance rápido en IA y cambios en plataformas educativas.	RE02

Listado de Riesgos del proyecto

Para evaluar los riesgos del proyecto de manera estructurada, se han elaborado diversas tablas que permiten clasificar y cuantificar su impacto. En primer lugar, se presenta una escala de probabilidad que asigna un valor porcentual a cada nivel de ocurrencia, desde "Improbable" hasta "Frecuente". Luego, se establece una escala de severidad que mide el impacto de cada riesgo en el proyecto, clasificándose en niveles que van desde "Despreciable" hasta "Catastrófico".

A partir de estas escalas, se define una matriz de impacto de riesgos que permite determinar la gravedad de cada evento mediante el cálculo del producto entre la probabilidad y la severidad. Dependiendo del resultado, el riesgo se clasifica como "Crítico", "Moderado" o "Bajo", lo que a su vez indica la urgencia de una acción correctiva. Finalmente, se presenta una tabla resumen con los riesgos identificados, su probabilidad e impacto estimado, así como su clasificación y el código correspondiente para su seguimiento y gestión.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

A continuación, se detallan los porcentajes correspondientes a cada nivel de probabilidad, severidad e impacto de riesgos en el proyecto:

Escala de probabilidad

Tabla 4

Nivel	Probabilidad
0.90	Frecuente
0.70	Probable
0.50	Ocasional
0.30	Aislado
0.10	Improbable

Leyenda de escala de probabilidad

Escala de severidad

Tabla 5

Nivel	Severidad
0.80	Catastrófica
0.40	Significativa
0.20	Moderada
0.10	Baja
0.05	Despreciable

Leyenda de escala de severidad

Impacto de Riesgos en el Proyecto

Tabla 6

Símbolo	Descripción	Valor	Acción
C	Crítico	Mayor que 0.17	Se realiza acción correctiva inmediatamente
M	Moderado	Entre 0.06 y 0.17	Se realiza acción correctiva a corto plazo
B	Bajo	Menor que 0.05	No requiere acción correctiva o se realiza en cualquier momento

Leyenda de impacto sobre los riesgos del proyecto

Tabla 7

Probabilidad estimada	Impacto estimado	Impacto de riesgo	Tipo de riesgo	Código
0.50	0.40	0.20	M	RP01
0.70	0.20	0.14	M	RP02
0.50	0.40	0.20	M	RP03
0.50	0.40	0.20	M	RP04
0.50	0.20	0.10	M	RP05
0.50	0.20	0.10	M	RP06

0.50	0.20	0.10	M	RP07
0.30	0.40	0.10	M	RT01
0.50	0.40	0.20	M	RT02
0.50	0.40	0.20	M	RT03
0.30	0.20	0.06	M	RT04
0.50	0.40	0.20	M	RT05
0.70	0.80	0.56	C	RT06
0.50	0.40	0.20	M	RT07
0.50	0.20	0.10	M	RT08
0.70	0.40	0.28	C	RT09
0.50	0.20	0.10	M	RT10
0.50	0.40	0.20	M	RT11
0.50	0.20	0.10	M	RT12
0.70	0.80	0.56	C	RE01
0.50	0.40	0.20	M	RE02

Impacto sobre los riesgos del proyecto

2.4.2. ¿Cómo manejará sus riesgos?

A medida que avancemos en la identificación de riesgos en el proyecto, se desarrollará un plan de respuesta y contingencia con el objetivo de gestionarlos de manera efectiva. Este plan considerará estrategias para evitar, mitigar o aceptar los riesgos según su impacto y probabilidad. Además, se establecerán criterios para activar medidas correctivas y asignar recursos según sea necesario. Aún en etapa de

planificación, se busca garantizar que cualquier eventualidad pueda abordarse de forma rápida y eficiente para minimizar interrupciones y asegurar el cumplimiento de los objetivos del proyecto.

2.5. Presupuesto

Tabla 8

Plataforma	Créditos Usados	Costo por crédito (\$US)	Tiempo Usado (min)	Gasto Total (\$US)	Gasto Total (RD\$)	Gasto Esperado (\$US)	Gasto Esperado (RD\$)
Roboflow	48.71	3	1425	0	0	147	8,558.25
Lightning AI	15	1	2100	0	0	15	873.75
Kaggle	-	-	3600+	0	0	0	0
Deepnote	-	-	3600	25	1,456.25	25	1,456.25
Total	-	-	-	25	1,456.25	187	10,888.25

Presupuestos calculados del proyecto

El presupuesto detallado del proyecto refleja los gastos realizados y previstos en varias plataformas y recursos necesarios para el desarrollo de los modelos y entrenamientos. El gasto en Roboflow se refiere a los créditos utilizados para el entrenamiento de modelos y la creación de datasets, donde se especifica un gasto total de \$0, debido a que se aprovechan los créditos gratuitos mensuales ofrecidos a estudiantes. Lightning AI se refiere al uso de créditos para realizar pruebas y entrenamientos de modelos con una Tesla T4, y aunque se ha utilizado una cantidad de créditos, el gasto total sigue siendo cero, ya que se están utilizando los créditos gratuitos para usuarios normales.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

En Kaggle, el gasto se ha limitado a 30 horas gratuitas por usuario para realizar pruebas y entrenamientos, por lo que no ha generado costo alguno. Por último, el gasto en DeepNote corresponde a la utilización de recursos en la plataforma para entrenamientos de modelos con una Tesla T4, y el gasto real es de \$25, ya que no se conocían otras plataformas gratuitas al inicio del proyecto. El presupuesto también considera un porcentaje para contingencias, que se añadirá a medida que surjan necesidades adicionales o imprevistas en el desarrollo del proyecto.. Además de estos costos, se ha considerado el tiempo invertido en el desarrollo del proyecto, estimando un total de 24 horas semanales dedicadas a tareas como entrenamiento de modelos, análisis de resultados y optimización de procesos.

Equipos utilizados

Los equipos utilizados en este proyecto incluyen dos laptops con especificaciones variadas que se adaptan a diferentes necesidades. La primera, una HP Laptop 15-dw2xxx, cuenta con un procesador Intel Core i5-1035G1, 16GB de RAM y una tarjeta gráfica Intel UHD Graphics. Esta laptop fue adquirida en noviembre de 2020 con un costo de \$871.57 US, equivalente a RD\$55,000 en noviembre de 2021. La segunda laptop, una Apple Macbook Air M2, está equipada con el procesador Apple Silicon M2, 8GB de RAM y un SSD de 256GB. Esta fue adquirida en octubre de 2023 por un precio de \$998.35 US, lo que equivale a RD\$63,000 en ese mismo mes. Ambos equipos ofrecen un rendimiento adecuado para tareas de desarrollo y procesamiento, aunque con características que se ajustan a necesidades y presupuestos diferentes.

Tabla 9

Descripción del Equipo	Especificaciones	Costo (\$US)	Costo (RD\$)
Laptop 1	DESKTOP-IOD4 T7Q HP Laptop 15-dw2xxx Procesador: Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz RAM: 16GB Gráfica: Intel(R) UHD Graphics	Noviembre del 2020 \$871.57	Noviembre del 2021 RD\$55,000
Laptop 2	Apple Macbook Air M2 Procesador Apple Silicon M2, 8GB de RAM, 256GB de SSD	Octubre del 2023 \$998.35	Octubre del 2023 RD\$63,000

Detalles de los equipos utilizados para la realización del proyecto

2.6. Definición de la demostración

La presentación del proyecto se realizará en un entorno académico, preferiblemente en una sala destinada para exposiciones o en un aula habilitada para presentaciones. Este espacio deberá estar adecuadamente preparado, con la disposición de sillas, proyectores o pantallas que permitan mostrar los resultados de manera clara. Además, será necesario disponer de una computadora para ejecutar el prototipo desarrollado en la interfaz web creada para su propósito y exhibir los resultados alcanzados.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Es fundamental contar con una conexión a internet confiable, que posibilite el acceso a los recursos en línea y la consulta de datos relevantes durante la demostración. El ambiente debe ser tranquilo y bien iluminado, favoreciendo la concentración de los jurados para facilitar la comprensión y análisis de la explicación del proyecto.

2.7. Análisis y Diseño

Este apartado se divide en dos partes esenciales para el desarrollo de nuestro proyecto: el análisis y el diseño. Ambas fases son fundamentales para entender el funcionamiento general del sistema y planificar cómo se llevará a cabo su implementación detallada.

2.7.1. Análisis

El presente análisis tiene como objetivo proporcionar una visión general de lo que llevamos hasta ahora en el proyecto, abarcando los procesos involucrados y los objetivos de cada uno. Para ello, se abordarán dos aspectos clave que visualizan cómo funciona el sistema hasta el momento: Diagrama de procesos el cual representa cómo fluye la información a través de los diferentes procesos del sistema y cómo estos interactúan entre sí. Casos de uso donde éste describe las funciones del sistema desde la perspectiva del usuario, identificando las principales interacciones y requerimientos.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Diagrama 1

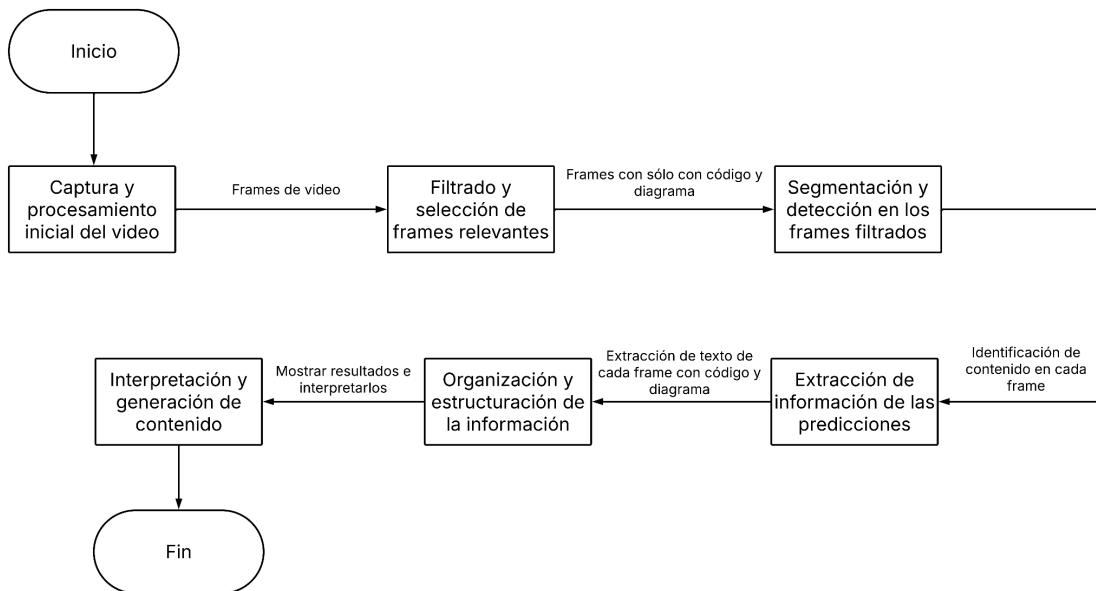


Diagrama de procesos del proyecto actualmente

El diagrama de procesos que hemos desarrollado, representa de manera clara el flujo de trabajo de lo que lleva nuestro sistema en estos momentos. En este caso, el primer paso en el diagrama de procesos, consiste en la captura y procesamiento inicial del video, donde en esta etapa, se obtiene el vídeo original y lo convierte en una serie de imágenes individuales (frames) utilizando para esto la herramienta PySceneDetect. Este proceso transforma el video en un conjunto de datos estáticos, donde nos permite analizar cada imagen por separado. Asimismo, para poder filtrar y seleccionar los frames que son relevantes, pasamos a la fase de filtrado y selección de frames, donde se aplica la técnica de visión por computadora con YOLOv12 para así poder identificar y seleccionar únicamente los frames que contienen código o diagramas, descartando aquellos que solo muestran contenido/elementos irrelevantes. Por otro lado, se utiliza DINOv2, tecnología de aprendizaje profundo, para poder comparar y eliminar esas escenas similares, así evitando la redundancia de imágenes casi idénticas y reduciendo también la cantidad de datos que el sistema necesita procesar para poder segmentar y detectar el contenido en las imágenes seleccionadas. Esta fase usa nuevamente YOLOv12, donde clasifica cada imagen en una de dos categorías: código o diagrama. Luego, se realiza un análisis más detallado dentro de cada frame para obtener las predicciones precisas sobre la ubicación de los bloques de código.

Después de identificar y segmentar correctamente el contenido, se pasa a la extracción de información de las predicciones, donde en esta etapa se utilizan técnicas de procesamiento de imágenes con cv2, para recortar las áreas detectadas dentro de cada frame, así aislando visualmente los fragmentos de código o diagramas. De la misma forma, se emplean herramientas como SuryaOCR, para poder extraer el texto dentro de las áreas identificadas, permitiendo convertir el contenido visual en datos textuales, es decir, extraer el texto que se asocia a la imagen que le corresponda.

Para la organización y estructuración de la información extraída, el sistema clasifica los datos en dos grupos: código y diagramas, teniendo en cuenta que cada tipo de contenido sea tratado según las características que tienen. Luego de esto, se prepara la información para ser interpretada mediante Modelos de Lenguaje Avanzado (LLM). Se estructuran los datos de forma comprensible para que el modelo pueda analizar el contenido visualizado sin ninguna ambigüedad.

Por último, para la interpretación y generación de contenido, se utiliza el modelo de inteligencia artificial OpenAI o Phi 4 para poder interpretar la información procesada. Este permite generar las explicación detalladas y comprensibles sobre el código o los diagramas identificados en el video. Asimismo, usa IA generativa para poder recrear y mejorar el contenido detectado, siendo este asegurado para que el resultado final sea claro y comprensible para los estudiantes.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Diagrama 2

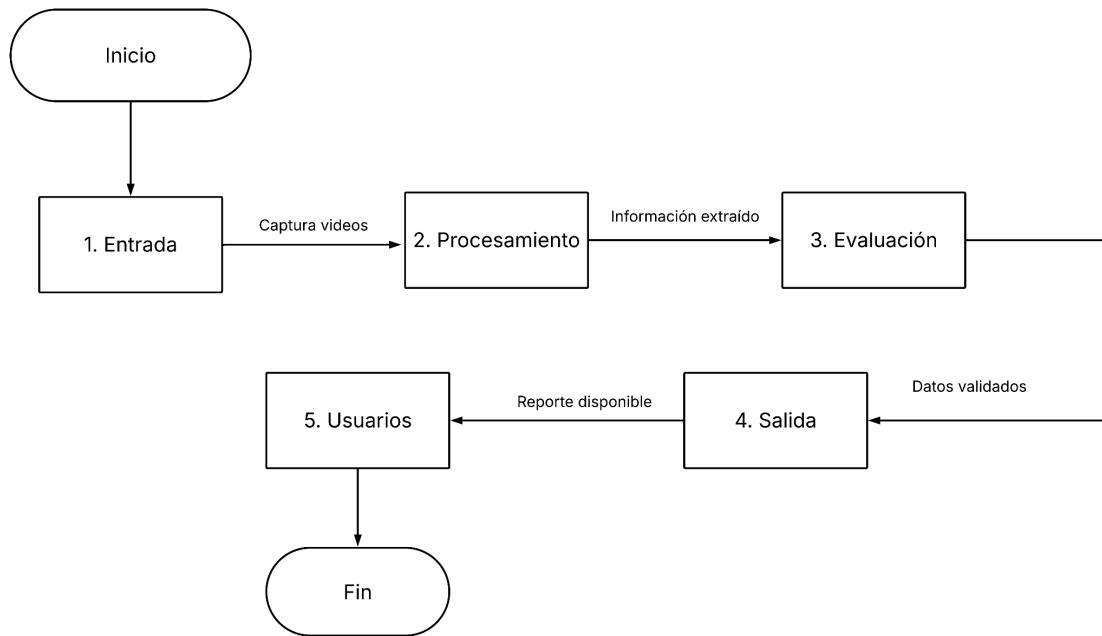


Diagrama general del Bot

El prototipo utilizará videos de clases pregrabadas así permitiendo el procesamiento automatizado del contenido educativo. Asimismo, usando herramientas de OCR, análisis de código y reconocimiento de diagramas, se extraerán los elementos clave de las clases, garantizando una interpretación precisa de la información presentada en pantalla. Posteriormente, se realizará una validación de los datos extraídos, asegurando su precisión. También, la información procesada se estructurará en reportes organizados y accesibles, facilitando la consulta del material de manera clara y comprensible donde los estudiantes podrán acceder al contenido a través de una plataforma web, brindando una experiencia intuitiva y centralizada para la consulta de los reportes generados.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Diagrama 3

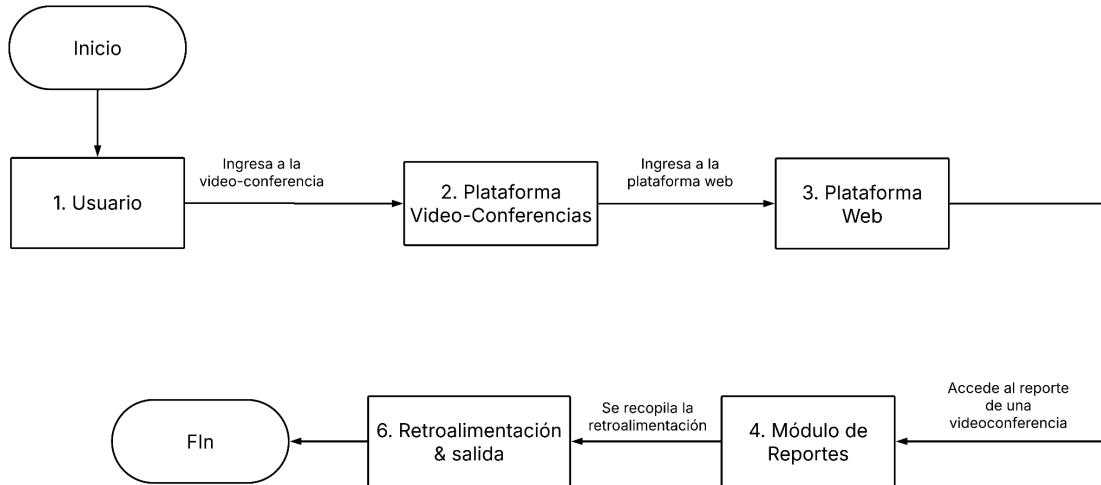


Diagrama general del Cliente

El objetivo del prototipo para el cliente es facilitar la consulta de información clave extraída de estas sesiones, permitiendo a los estudiantes acceder a reportes organizados con códigos y diagramas relevantes para su aprendizaje. El usuario final de este sistema es el estudiante que participa en las videoconferencias y que posteriormente consulta los reportes generados. Estos reportes son extraídos de las grabaciones de clases mediante herramientas avanzadas de reconocimiento óptico de caracteres (OCR), LLM y análisis de código. Para la captura del contenido, el sistema se integra con una plataforma de videoconferencias, que es el medio utilizado para la transmisión y grabación de las clases. A partir de estas grabaciones, se extrae la información visualizada para posteriormente procesarla y organizarla. Asimismo, los reportes generados son almacenados y accesibles a través de una plataforma web, la cual permite a los estudiantes ingresar y consultar el material de manera estructurada. Dentro de esta plataforma, se encuentra el módulo de reportes, donde los estudiantes pueden visualizar las explicaciones de los diagramas y códigos extraídos de las videoconferencias. Este apartado facilita la revisión del contenido presentado en clase y ayuda a reforzar el aprendizaje.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Por último, se tendrá un proceso de retroalimentación, el cual permite validar el contenido generado y asegurar su calidad. Brindando una herramienta efectiva para el estudio y la consulta de los temas vistos en clase

Diagrama 4

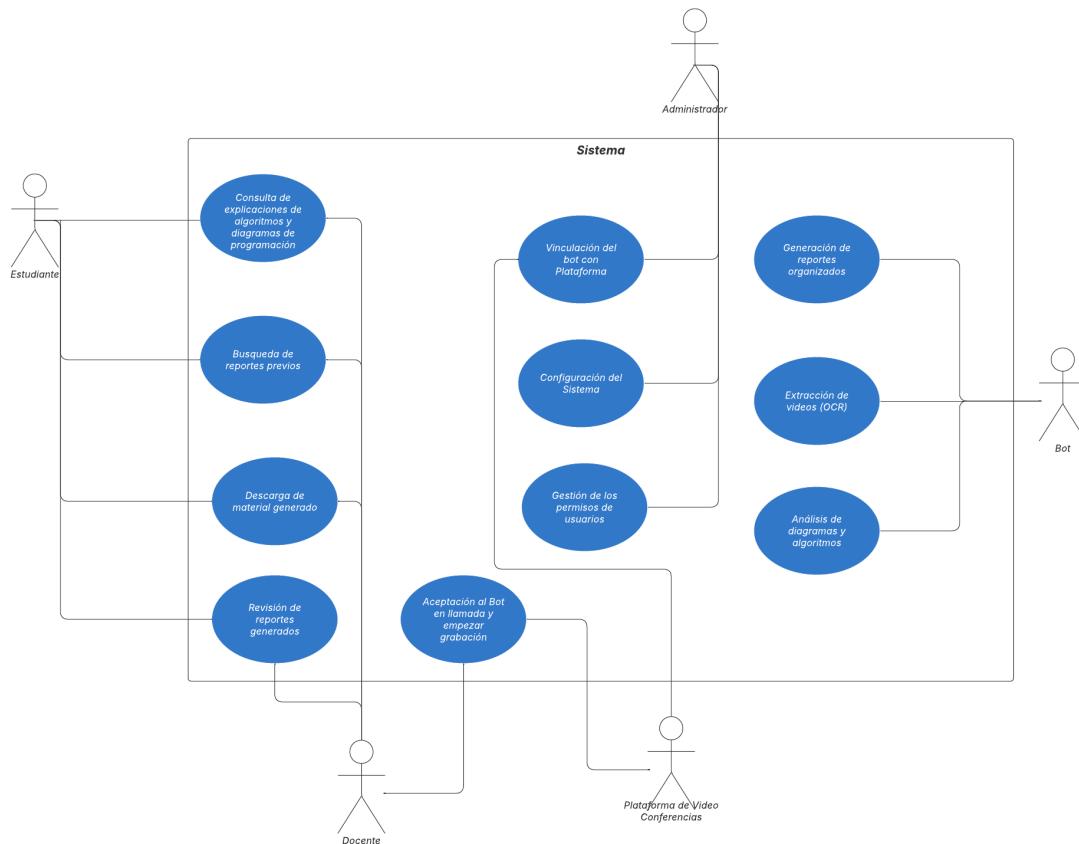


Diagrama de casos de uso

Bot de Asistencia Interactivo en Programas de Algoritmos y Diagramas de Programación para la Asignatura de “Introducción a la Algoritmia” es el encargado de analizar los algoritmos y generar reportes en videoconferencias. Asimismo, está diseñado para facilitar la comprensión y documentación de explicaciones de programación durante clases virtuales. Su objetivo principal es analizar en tiempo real los videos de conferencias en los que se presentan diagramas y algoritmos, extrayendo información relevante mediante tecnología de OCR y modelos LLM. Posteriormente, genera reportes organizados que pueden ser consultados por estudiantes y docentes.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

La plataforma también permite la administración de permisos y configuración del sistema para garantizar un uso eficiente y seguro.

Los actores principales del sistema incluyen estudiantes, docentes, administradores y un bot automatizado. Los estudiantes pueden acceder a la plataforma para consultar explicaciones sobre algoritmos y diagramas de programación, buscar reportes previos y descargar material generado a partir del análisis de videoconferencias. Por otro lado, los docentes desempeñan un papel fundamental en la validación del contenido generado por el bot, ya que pueden revisar los reportes, proporcionar retroalimentación y aceptar la participación del bot en las videollamadas.

El administrador del sistema es el encargado de gestionar la configuración del software, asegurándose de que el bot esté correctamente vinculado con la plataforma de videoconferencias y que los usuarios tengan los permisos adecuados. Entre sus responsabilidades se incluyen la asignación de roles, la supervisión de la generación de reportes y la optimización del análisis realizado por el bot.

El bot es un componente clave del sistema, ya que su función es extraer información de los videos mediante OCR, analizar diagramas y algoritmos, y generar reportes detallados. Una vez que el docente acepta su participación en una videoconferencia, el bot inicia la grabación y el análisis del contenido. Posteriormente, organiza la información obtenida en reportes estructurados que pueden ser revisados por los usuarios.

La interacción entre los actores y el sistema comienza cuando un docente inicia una videoconferencia y permite que el bot participe en la sesión. Durante la conferencia, el bot graba la sesión, extrae información y analiza los algoritmos y diagramas presentados. Una vez concluida la reunión, el sistema genera reportes organizados que quedan disponibles para que los docentes los revisen y los estudiantes los consulten. Además, el administrador puede realizar ajustes en la configuración del sistema y en los permisos de acceso de los usuarios según sea necesario.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Diagrama 5

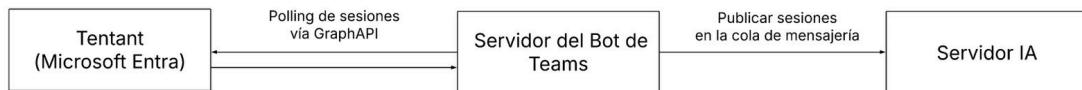


Diagrama de integración con Teams

En este diagrama se representa el flujo de integración entre el entorno institucional de Microsoft Entra (antes Azure Active Directory), el servidor del bot de asistencia y el sistema de inteligencia artificial.

El servidor del bot realiza un *polling* de las sesiones de clase mediante la API de Microsoft Graph, extrae la información relevante, y luego publica estos datos en una cola de mensajería que es consumida por el servidor encargado del procesamiento con IA.

2.7.2. Diseño

Diagrama 6

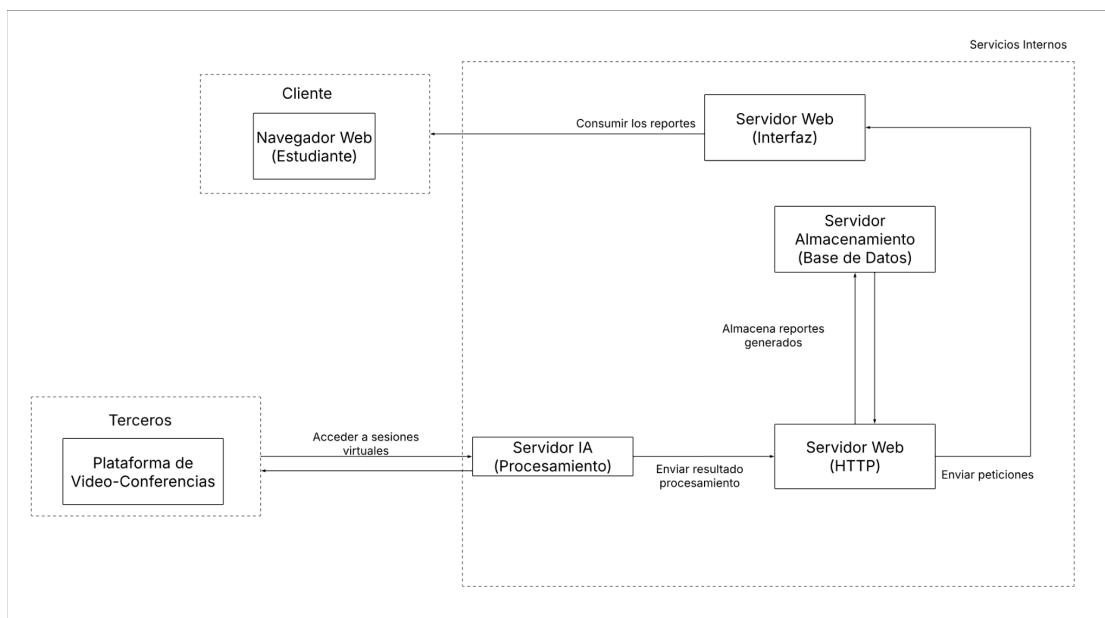


Diagrama de despliegue

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

El sistema está diseñado para que los estudiantes puedan acceder fácilmente a los reportes generados a través de un navegador web. Desde ahí, pueden revisar y explorar la información extraída de sus clases de manera organizada e interactiva.

Para garantizar que todo funcione sin problemas, el sistema se apoya en varios servidores. En primer lugar, los servidores externos incluyen un servidor de almacenamiento, donde se guardan los archivos multimedia de las sesiones, y la plataforma de reuniones virtuales, que es la fuente de donde el bot extrae la información para procesarla.

Además, dentro de la infraestructura interna, el servidor web se encarga de mostrar la plataforma donde los estudiantes pueden visualizar sus reportes. Por su parte, el servidor del bot actúa como un auditor automatizado, conectándose a las sesiones virtuales para capturar y organizar la información relevante.

El procesamiento de los videos extraídos que realiza el bot, donde técnicas de inteligencia artificial permiten interpretar el contenido y estructurarlo de manera útil. Una vez procesada la información, los reportes quedan almacenados en una base de datos, asegurando que los estudiantes puedan acceder a ellos en cualquier momento.

Desarrollo e Implementación

Segmentación con PySceneDetect

Evaluar videos completos resulta en una tarea compleja, considerar cada frame de un video podría tardar bastante al considerar la cantidad de frames resultantes de una grabación que en promedio, dependiendo de las diversas plataformas, puede oscilar entre 16 fotogramas por segundo (FPS) hasta los 24 FPS siendo este un estándar de para formatos de videos, que en promedio resultaría en 12,600 FPS por cada 10 minutos de video. Por lo que, para hacer esta tarea mucho más llevable, la solución propuesta implementa PySceneDetect, una biblioteca de Python que contiene funciones para detectar cambios de escenas en videos, con la cual para un ejemplo de 1 escena se obtienen 3 frames, cabe destacar que las escenas contienen pueden

contener varios minutos de video, independientemente la cantidad, la reducción es representativa.

Las características presentes en los videos, tomando en cuenta que son sesiones virtuales académicas, se estima una duración promedio de entre 1 hora con 40 minutos a 2 horas con 30 minutos, de todos modos es posible procesar clases más cortas como tutorías o clases cortas de 30 minutos a 1 hora. Las plataformas de reuniones virtuales cuentan con grabaciones en alta calidad, con lo que se estima que se contará con videos entre 720p a 1080p, asegurando así que los cálculos de los píxeles globales sea buena y pueda ser diferenciada correctamente por la librería a utilizar. En última instancia se estiman sesiones de clases en el idioma español, pero no limitados a este.

PySceneDetect cuenta con diversas funciones, entre las que vale destacar ‘detect_scenes()’, responsable de procesar el video y conseguir los cambios de escenas, esto mediante el uso de sus detectores, de los cuales se pueden destacar el ‘AdaptiveDetector’ y el ‘ThresholdDetector’. Respectivamente, el primero a nivel teórico, es el objeto más personalizable entre todos los manejadores de detección que la librería ofrece, su manera de funcionar se basa en el cálculo de píxeles a nivel global y realizando un cálculo de diferencia a nivel de tono, saturación, brillo (HSV), mientras que el segundo realiza el cálculo a nivel global de píxeles con una diferencia a ser establecida, se puede observar un resultado en la figura 1. Otra función importante es ‘get_scene_list()’, la cual retorna el intervalo de tiempo en el que inicia y termina una escena, retornando con los minutos y segundos de cada una, presente en la figura 2.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Figura 1

Detected: 2 Progress: 0%	110/27033 [00:00<01:12, 371.87frames/s]New scene found at frame 37.
New scene found at frame 81.	
Detected: 3 Progress: 9% █	2418/27033 [00:06<01:00, 408.38frames/s]New scene found at frame 2347.
Detected: 4 Progress: 16% ██	4346/27033 [00:10<00:57, 396.13frames/s]New scene found at frame 4286.
Detected: 6 Progress: 45% █████	12171/27033 [00:30<00:36, 411.58frames/s]New scene found at frame 12111.
New scene found at frame 12160.	
Detected: 7 Progress: 50% ████	13397/27033 [00:33<00:34, 395.21frames/s]New scene found at frame 13347.
Detected: 8 Progress: 53% █████	14456/27033 [00:36<00:31, 399.28frames/s]New scene found at frame 14376.
Detected: 10 Progress: 56% █████	15268/27033 [00:38<00:29, 401.80frames/s]New scene found at frame 15185.
New scene found at frame 15218.	
Detected: 11 Progress: 57% █████	15430/27033 [00:38<00:29, 393.38frames/s]New scene found at frame 15374.
Detected: 12 Progress: 71% ███████	19198/27033 [00:48<00:19, 395.94frames/s]New scene found at frame 19145.
Detected: 13 Progress: 77% ███████	20937/27033 [00:52<00:15, 387.30frames/s]New scene found at frame 20879.
New scene found at frame 20944.	
Detected: 16 Progress: 91% ████████	24712/27033 [01:02<00:06, 383.80frames/s]New scene found at frame 24635.
New scene found at frame 24687.	
Detected: 17 Progress: 94% █████████	25394/27033 [01:04<00:04, 393.25frames/s]New scene found at frame 25324.
Detected: 18 Progress: 98% ██████████	26450/27033 [01:06<00:01, 394.73frames/s]New scene found at frame 26368.
Detected: 20 Progress: 99% ███████████	26893/27033 [01:07<00:00, 385.14frames/s]New scene found at frame 26839.
New scene found at frame 26862.	
Detected: 21 Progress: 100% ███████████	27010/27033 [01:08<00:00, 383.00frames/s]New scene found at frame 26959.
New scene found at frame 27008.	
Detected: 22 Progress: 100% ███████████	27033/27033 [01:08<00:00, 394.89frames/s]

Resultado de “detect_scenes()” de un video de 30 minutos.

Figura 2

Escena 1: 00:00:00.000 – 00:00:02.467
Escena 2: 00:00:02.467 – 00:00:05.400
Escena 3: 00:00:05.400 – 00:02:36.476
Escena 4: 00:02:36.476 – 00:04:45.750
Escena 5: 00:04:45.750 – 00:13:27.447
Escena 6: 00:13:27.447 – 00:13:30.714
Escena 7: 00:13:30.714 – 00:14:49.851
Escena 8: 00:14:49.851 – 00:15:58.455
Escena 9: 00:15:58.455 – 00:16:52.392
Escena 10: 00:16:52.392 – 00:16:54.592
Escena 11: 00:16:54.592 – 00:17:04.993
Escena 12: 00:17:04.993 – 00:21:16.407
Escena 13: 00:21:16.407 – 00:23:12.014
Escena 14: 00:23:12.014 – 00:23:16.347
Escena 15: 00:23:16.347 – 00:27:22.428
Escena 16: 00:27:22.428 – 00:27:25.895
Escena 17: 00:27:25.895 – 00:28:08.364
Escena 18: 00:28:08.364 – 00:29:17.968
Escena 19: 00:29:17.968 – 00:29:49.370
Escena 20: 00:29:49.370 – 00:29:50.904
Escena 21: 00:29:50.904 – 00:29:57.371
Escena 22: 00:29:57.371 – 00:30:00.638
Escena 23: 00:30:00.638 – 00:30:02.304

Resultado de “get_scene_list()” de un video de 30 minutos.

A partir de las escenas identificadas, al momento de guardar las imágenes con la función ‘save_images()’, hay una serie de parámetros importantes a tomar en consideración.

Para obtener la mayor calidad y cantidad de frames necesarios de cada escena, es necesario establecer diversos parámetros de la función ‘save_images()’:

- ‘**num_images** (int)’: establece cuantas imágenes serán guardadas para cada escena identificada. Se eligieron 3, ya que esto permite tener el inicio, mitad y final de cada escena.
- ‘**image_extension** (str)’: establece el formato de salida entre los permitidos (jpg, png, webp). Se eligió ‘png’ ya que es un formato de imagen más estándar con mejor calidad
- ‘**encoder_param** (int)’: establece la calidad o compresión que tendrá la imagen resultante. Para ‘png’ los valores oscilan entre 1-9, siendo 9 un resultado 100% fiable.
- ‘scale (float)’: establece el factor por el cual la imagen será escalada, utilizando la interpolación establecida.
- ‘interpolation (Interpolation)’: establece el tipo de interpolación cuando se realiza una modificación de tamaño en la imagen resultante, ya sea por el escalado o eligiendo medidas específicas.

Es importante entender la interpolación de imágenes, uno de los parámetros utilizados al momento de guardar las imágenes, ya que de este depende mucho la calidad de las imágenes reescaladas. Este proceso consiste en calcular y generar nuevos valores de píxeles cuando transformamos una imagen, por ejemplo, al hacer un zoom in o zoom out. Asimismo, existen diferentes métodos de interpolación, cada uno con características específicas que los hacen más adecuados para distintas necesidades en términos de calidad y velocidad de procesamiento. Existen diversos tipos, pero la elegida en esta ocasión es ‘LANCZOS4’, este tipo utiliza un filtro basado en la función matemática ‘sinc()’, que es una técnica avanzada para reconstruir señales a partir de muestras. Los resultados son muy suaves y fieles a la imagen original, lo que lo convierte en una opción de alta calidad. Sin embargo, es más lento debido a la complejidad de los cálculos involucrados, aunque, en teoría, el método LANCZOS4 brinda los mejores resultados en calidad visual, la realidad es que requiere mayores recursos de procesamiento. De todos modos, dado que buscamos obtener imágenes excepcionalmente suaves y precisas en el prototipo, se ha optado por utilizar LANCZOS4 a pesar de su costo computacional, ya que esto garantizará una mejor fidelidad en las imágenes procesadas.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Previamente se obtuvo una lista detallada con los intervalos exactos (timestamps y frames) donde ocurrían los cambios de escena. Con esta información, se puede generar un arreglo de diccionarios iniciales que contiene un identificador de escenas, timestamps, frames y sus respectivas imágenes, como se muestra en la figura 3. Esto es esencial para iniciar el proceso de extracción de la información de cada escena, en futuros pasos, se irá categorizando y filtrando las escenas hasta obtener un arreglo con exclusivamente las escenas que contengan códigos o diagramas.

Toda la implementación de PySceneDectect es abstraída a una clase personalizada que cuenta con los métodos necesarios para hacer la tarea mucho más simple y escalable, como se puede apreciar en la figura X.

Figura 3

```
{
  "duration": "00:30:02.304",
  "categories": [],
  "scenes": [
    {
      "scene_id": 1,
      "start_timestamp": "00:00:00.000",
      "end_timestamp": "00:00:02.467",
      "images": [
        {
          "name": "scene_1_frame_1",
          "path": "video/scenes/scene_1_frame_1",
          "category": "default"
        },
        {
          "name": "scene_1_frame_2",
          "path": "video/scenes/scene_1_frame_2",
          "category": "default"
        },
        {
          "name": "scene_1_frame_3",
          "path": "video/scenes/scene_1_frame_3",
          "category": "default"
        }
      ]
    },
    {
      "scene_id": 2,
      "start_timestamp": "00:00:02.467",
      "end_timestamp": "00:00:04.934",
      "images": [
        {
          "name": "scene_2_frame_1",
          "path": "video/scenes/scene_2_frame_1",
          "category": "default"
        },
        {
          "name": "scene_2_frame_2",
          "path": "video/scenes/scene_2_frame_2",
          "category": "default"
        }
      ]
    }
  ]
}
```

Diccionario resultante de “`extract_video_from_detection()`” de un video de 30 minutos.

Clasificación de Imágenes con YOLOv12

Una vez obtenidas todas las escenas con sus respectivos frames, es necesario clasificarlos, ya sea que la imagen presente códigos de algoritmos o diagramas de flujo de programación, para conseguir esto, se debe ser capaz de entender qué es lo que contiene cada frame. Una tarea como ésta resulta compleja de realizar de forma iterativa o recurrente un uso intenso de un OCR, por lo que la solución propuesta parte del uso de un algoritmo de visión artificial llamado “You Only Look Once”

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

(YOLO) que permite detectar objetos en tiempo real, la solución desarrollada se aprovecha del último modelo lanzado, YOLOv12.

Para entrenar un modelo capaz de clasificar elementos dentro de una imagen, o clasificar si una imagen contiene códigos de algoritmos o diagramas de flujo de programación, es necesario crear un dataset con ciertas características específicas, como un etiquetado y clasificado. Para crear el dataset de YOLOv12, se utiliza la plataforma ‘Roboflow’, una plataforma diseñada para manejar y etiquetar imágenes.

Para aprovechar el buen uso de PySceneDetect y su segmentación de videos, se hace uso de dichas escenas para la creación de dichos datasets, cada dataset se enfoca en una tarea en específico, esto para poder tener modelos más ligeros enfocados en tareas particulares antes que un gran modelo que intente resolver todos los problemas. Todas las características de los dataset se muestran en las tablas 10, 11, 12 y 13, una característica global es la cantidad de “epochs” de entrenamiento, que son 300.

En primer lugar, se suben las imágenes de las escenas extraídas con PySceneDetect y se marcan los bloques de código de cada una de ellas. Para hacerlo, dibujamos rectángulos (bounding boxes) alrededor de las áreas de interés, asegurándonos de que cada bloque estuviera bien delimitado.

Después de etiquetar las imágenes, Roboflow se encargó de dividirlas automáticamente en tres grupos: 70% para entrenamiento (train), 20% para validación (valid) y 10% para pruebas (test). Además, este proceso es importante porque ayuda a que el modelo aprenda correctamente, verificando su desempeño con imágenes que no ha visto antes.

Tabla 10

Datasets del modelo para detectar código: *code-snippet-video-class-detection*

Leyenda:

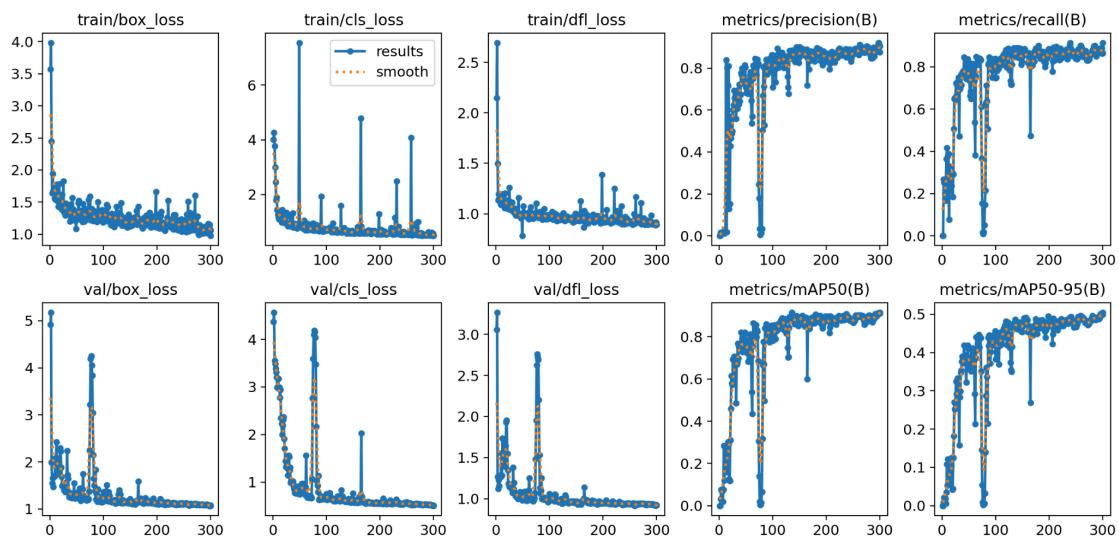
Preprocesamiento Básico: Auto-Orientación y Resize a 640x640

Versión	Imágenes	Anotaciones	Preprocesamiento	Postprocesamiento	Clases	mAP
v1	151	N/A	Básico	N/A	code_snippet	92.1%

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

v22	194	N/A	Básico, Escala Gris	N/A	code_snippet	95.8%
v26	375	N/A	Básico, Escala Gris	Rotación Horizontal	code_snippet	87.8%
v27	241	3868	Básico, Escala Gris	N/A	code_snippet code_bracket	91.4%

Figura 4



Resultados del entrenamiento de la v27 del modelo “code-snippet-video-class-detection”

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Tabla 11

Datasets del modelo para detectar diagramas de drawio: drawio-videoclass-detection

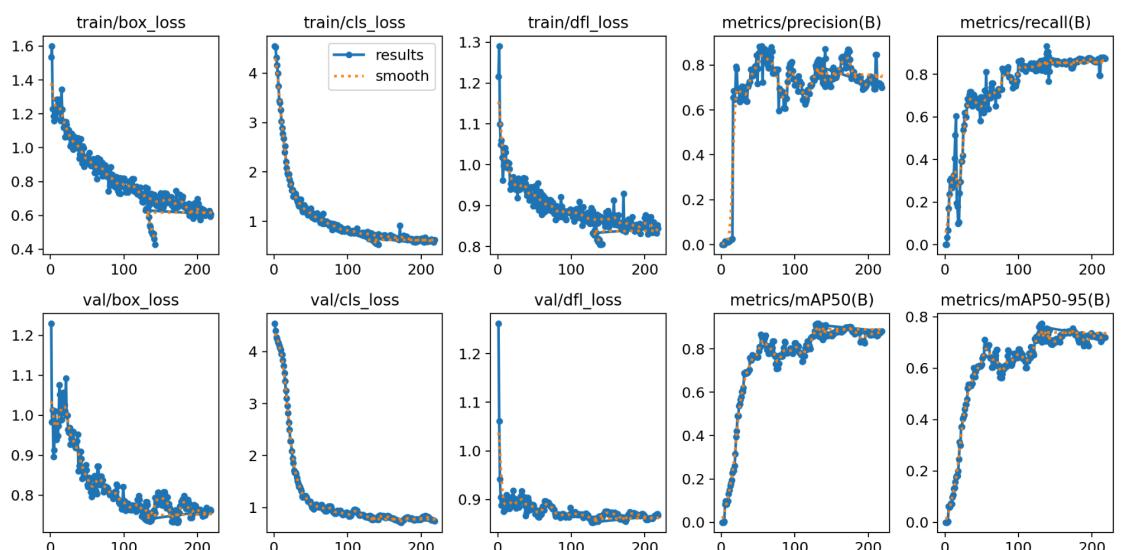
Leyenda:

Preprocesamiento ‘Básico’: Auto-Orientación y Resize a 640x640

Clase ‘diagram’: decision-arrow, decision-node, initial-node, final-node, function-node, normal-arrow, operation-node, print-node, variable-node

Versión	Imágenes	Anotaciones	Preprocesamiento	Postprocesamiento	Clases	mAP
v6	63	472	Básico	Rotación Horizontal	diagram	91.7%
v11	100	472	Básico, Escala Gris	Rotación Horizontal, Zoom	diagram	70.7%
v15	61	472	Básico, Escala Gris	Rotación Horizontal, Escalado Gris	diagram	87.6%

Figura 5



Resultados del entrenamiento de la v6 del modelo “drawio-videoclass-detection”

Tabla 12

Datasets del modelo para detectar de diagramas de flowgorithm: flowgorithm-videoclass-detection

Leyenda:

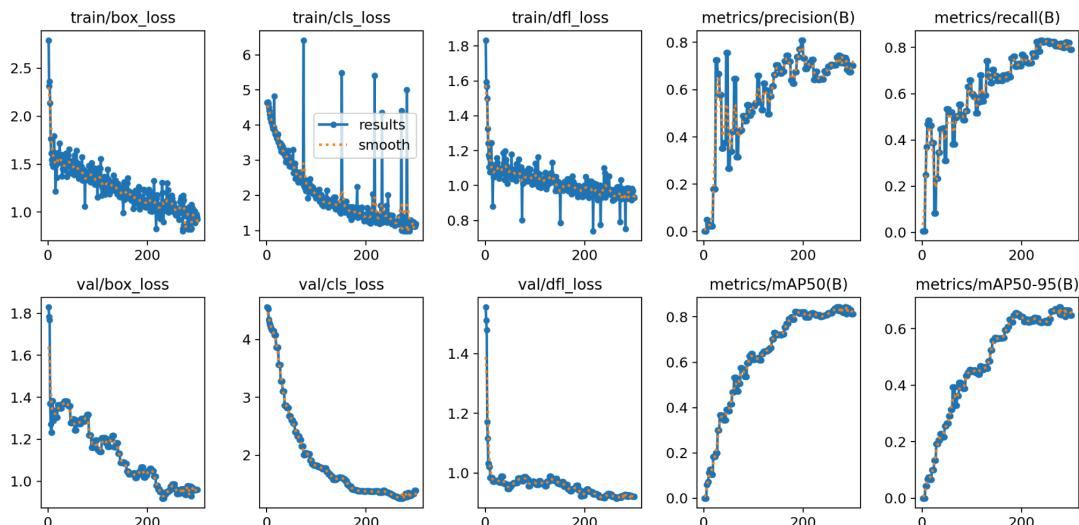
Preprocesamiento ‘Básico’: Auto-Orientación y Resize a 640x640

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Clase 'diagram': decision-arrow, decision-node, initial-node, final-node, normal-arrow, operation-node, output-node, input-node

Versión	Imágenes	Anotaciones	Preprocesamiento	Postprocesamiento	Clases	mAP
v1	53	430	Básico, Escala Gris	Rotación Horizontal y Vertical	diagram	85.4%
v2	19	430	Básico, Escala Gris	N/A	diagram	84.1%
v3	29	430	Básico, Escala Gris	Rotación Horizontal	diagram	81.5%

Figura 6



Resultados del entrenamiento de la v2 del modelo “flowgorithm-videoclass-detection”

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Tabla 13

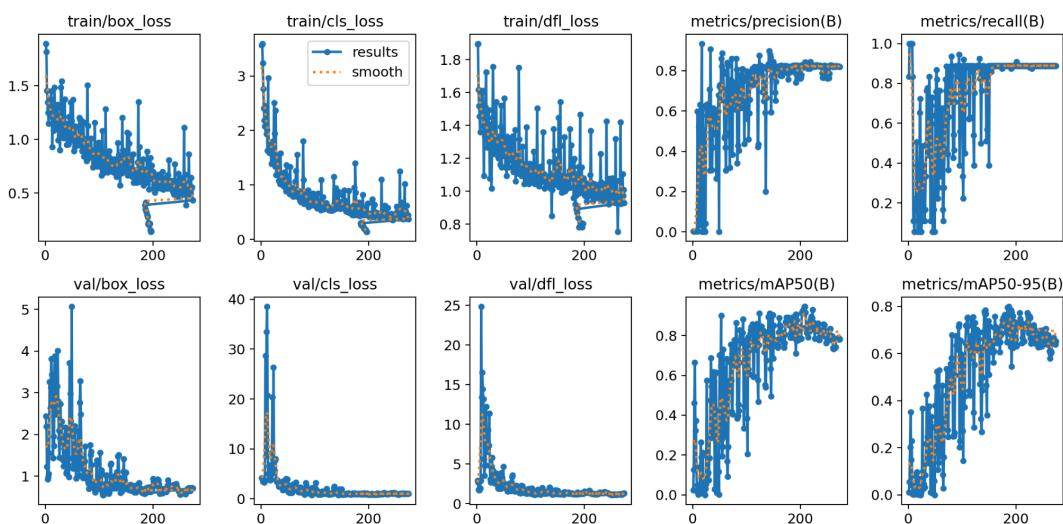
Datasets del modelo para detectar diferentes tipos de diagramas: drawio-vs-flowgorithm

Leyenda:

Preprocesamiento ‘Básico’: Auto-Orientación y Resize a 640x640

Versión	Imágenes	Anotaciones	Preprocesamiento	Postprocesamiento	Clases	mAP
v4	92	59	Básico, Escala Gris	Rotación Horizontal	drawio, flowgorithm	85.6%
v5	146	59	Básico, Escala Gris	Rotación Horizontal y Vertical	drawio, flowgorithm	89.4%

Figura 6



Resultados del entrenamiento de la v5 del modelo “drawio-vs-flowgorithm”

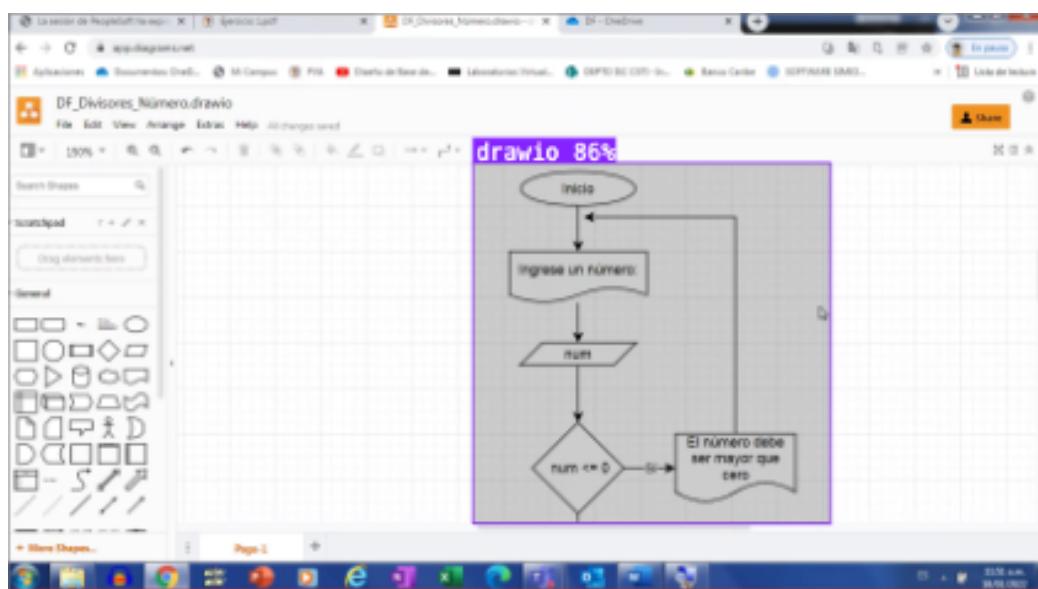
Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Para la tabla 10, es necesario realizar aclaraciones y observaciones de lugar. El mejor modelo que se encuentra en la tabla (marcado de negrita), no es el modelo entrenado con más imágenes, ni con mayor media de precisión global (mAP), ya que las pruebas realizadas llevaron a la conclusión de que era necesario detectar las posibles líneas de código aisladas de las llaves para el control de la estructura del código, esto debido a la dificultad de detección de las llaves por parte de los diversos modelos de OCR.

Para la tabla 12, es necesario explicar que, a pesar de tener una mAP ligeramente peor (1.3%), el incremento de la precisión es significativo, del 68.3% al 75%, aunque hay una pérdida de recall del 85.1% al 80.9%. La precisión es un atributo que se busca con más preferencia y tratar de tener unos valores más equilibrados. También es importante destacar el poco contenido que se pudo obtener para el dataset, otorgando resultados muy mejorables.

Completada la creación de datasets, se pueden utilizar los modelos para evaluar cada uno de los frames resultantes de las detecciones de PySceneDetect, para detectar por ejemplo qué clase de diagrama es al que corresponde como se puede observar en la figura 7. Así mismo, se pueden obtener las predicciones de dicha imagen, tal como se muestra en la figura 8.

Figura 7



Visualización de predicción de un frame con diagrama de flujo.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Figura 8

```
{  
    "predictions": [  
        {  
            "x": 1170,  
            "y": 557,  
            "width": 592,  
            "height": 600,  
            "confidence": 0.857,  
            "class": "drawio",  
            "class_id": 0,  
            "detection_id": "956672ac-7aaa-4f33-9159-941dac76d20b"  
        }  
    ]  
}
```

Resultado de predicción de un frame con diagrama de flujo.

Se procesan todas las imágenes obtenidas y se va agregando la clase de cada categoría y finalmente se evalúa si la cantidad de clases cumple con el mínimo establecido (mínimo 50% de apariciones para contar como contenido principal de la clase). Con esto se modifica el diccionario anterior y hasta este punto se tendría algo como lo mostrado en la figura 9, donde ya se encuentra la clase principal del video y cada frame tiene su clase.

Figura 9

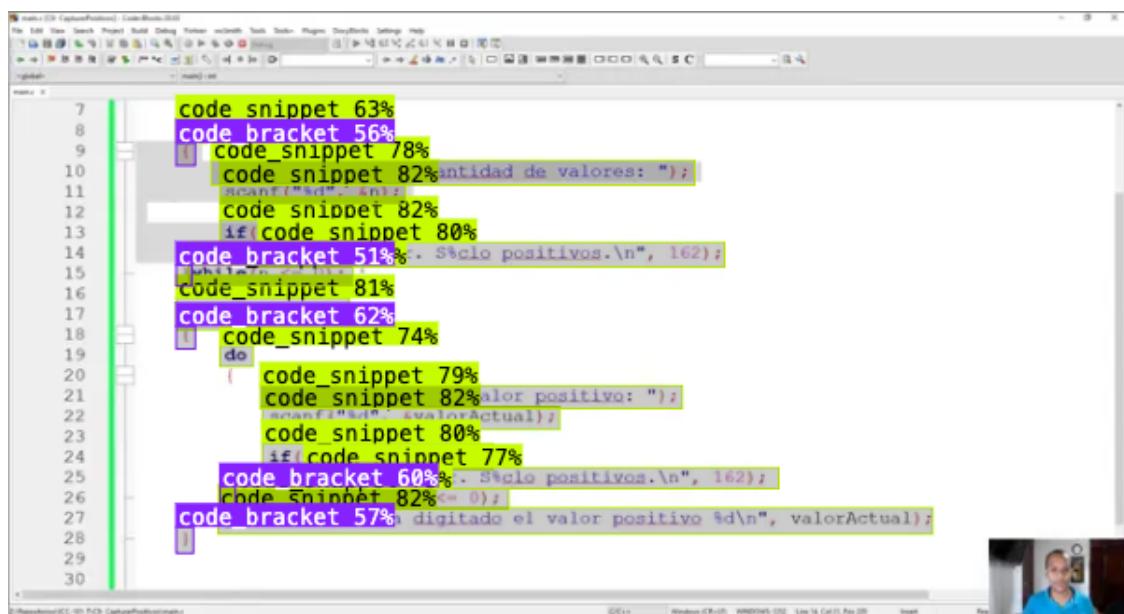
```
{  
    "duration": "00:30:02.304",  
    "categories": [  
        "drawio"  
    ],  
    "scenes": [  
        {  
            "scene_id": 1,  
            "start_timestamp": "00:00:00.000",  
            "end_timestamp": "00:00:02.467",  
            "images": [  
                {  
                    "name": "scene_1_frame_1",  
                    "path": "video/scenes/scene_1_frame_1",  
                    "category": "drawio"  
                },  
                {  
                    "name": "scene_1_frame_2",  
                    "path": "video/scenes/scene_1_frame_2",  
                    "category": "drawio"  
                },  
                {  
                    "name": "scene_1_frame_3",  
                    "path": "video/scenes/scene_1_frame_3",  
                    "category": "drawio"  
                }  
            ]  
        },  
    ]  
},
```

Diccionario representativo del video actualizado con su clase.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

De manera ilustrativa, se puede observar como funcionaría al momento de evaluar un código en la figura 10 y 11, donde se escanean los códigos.

Figura 10



Visualización de predicciones de un frame con código de algoritmos.

Figura 11

```
{  
    "predictions": [  
        {  
            "x": 324,  
            "y": 297.5,  
            "width": 408,  
            "height": 15,  
            "confidence": 0.824,  
            "class": "code_snippet",  
            "class_id": 1,  
            "detection_id": "3cdaa347-ae71-4cda-bc22-bc65ce99c7d3"  
        },  
        {  
            "x": 154.5,  
            "y": 132.5,  
            "width": 67,  
            "height": 15,  
            "confidence": 0.823,  
            "class": "code_snippet",  
            "class_id": 1,  
            "detection_id": "98d2d43d-c429-4209-b7d4-57b43c28e36c"  
        },  
        {  
            "x": 230,  
            "y": 238.5,  
            "width": 170,  
            "height": 13,  
            "confidence": 0.822,  
            "class": "code_snippet",  
            "class_id": 1,  
            "detection_id": "f2a2a2a2-a2a2-42a2-b2a2-b2a2a2a2a2a2"  
        }  
    ]  
}
```

Resultado de predicciones de un frame con código de algoritmos.

Segmentación con DINO v2

Tras obtener las escenas de los videos con PySceneDetect y clasificarlas con YOLO entre las posibles clases, no es preciso confiar en que la detección de PySceneDetect fue perfecta, por lo que es necesario validar porcentualmente que tan similares son las imágenes y filtrarlas. Para dichos fines se utilizó DINOv2, un modelo de visión por computadora basado en Transformers.

DINOv2 es un modelo que emplea varios algoritmos y componentes especializados para procesar imágenes de manera eficiente y extraer características relevantes. Entre estos componentes se encuentran:

- **AutoImageProcessor:** Este componente, parte de la biblioteca Transformers, facilita el proceso de preparación de imágenes antes de que sean ingresadas al modelo. Se encarga automáticamente de ajustar detalles como el tamaño, la normalización y la preparación de las imágenes, asegurando que sean compatibles con el modelo pre entrenado sin necesidad de preocuparse por estos aspectos técnicos.
- **AutoModel:** También incluido en Transformers, AutoModel simplifica el proceso de cargar modelos pre entrenados. Una vez cargado el modelo, se puede aplicar directamente a las imágenes procesadas para extraer características útiles o hacer predicciones, lo que hace que el uso de modelos avanzados sea mucho más sencillo, sin tener que entrenarlos desde cero.

Dado que el proyecto emplea DINOv2 a través de Transformers, AutoImageProcessor y AutoModel resultan ser opciones convenientes dentro de las diversas opciones posibles debido a su fácil integración y compatibilidad con este modelo, particularmente el '*facebook/dinov2-base*'.

Las características de las imágenes resultantes son bastantes similares, entiéndase que en los momentos importantes, donde hay códigos de algoritmos o diagramas de flujo de programación, comparten un fondo muy similar (mayoritariamente blanco) con líneas de texto representando dichos códigos o figuras geométricas representando los

diagramas. Dichas características hacen que las imágenes resulten muy similares a nivel de conteo global de píxeles, incluso para el espacio de colores HSV previamente mencionado.

Para analizar las similitudes entre las imágenes, las escenas extraídas son procesadas para identificar similitudes y agrupar las imágenes redundantes. Se eliminan las escenas repetitivas, conservando sólo aquellas con diferencias significativas, que dada la naturaleza de los videos explicada previamente y una robusta verificación humana, se determinó que un gran porcentaje de las imágenes con una diferencia mayor del 0.5% se trata de una imagen con un código o diagrama distinto, estos valores se pueden apreciar en la figura 12.

Ya así, una vez segmentadas automáticamente las escenas de los videos, se procede a extraer las características relevantes de cada imagen utilizando DINOv2. Este modelo permite generar representaciones numéricas (features) que capturan detalles visuales importantes de cada imagen. Para lograr esto, se siguen los siguientes pasos:

- **Preprocesamiento Automático:** Las imágenes extraídas de cada escena fueron procesadas para asegurar que tuvieran el formato adecuado antes de ser ingresadas a DINOv2. Este preprocesamiento incluyó la conversión a los colores adecuados, redimensionamiento y normalización.
- **Extracción de Características Visuales:** Las imágenes procesadas fueron introducidas en DINOv2, el cual generó representaciones numéricas que captaron características visuales clave del contenido, valores que se aprecian en la figura 12. Estas representaciones permiten realizar análisis posteriores, como identificar patrones, similitudes o diferencias entre las imágenes.

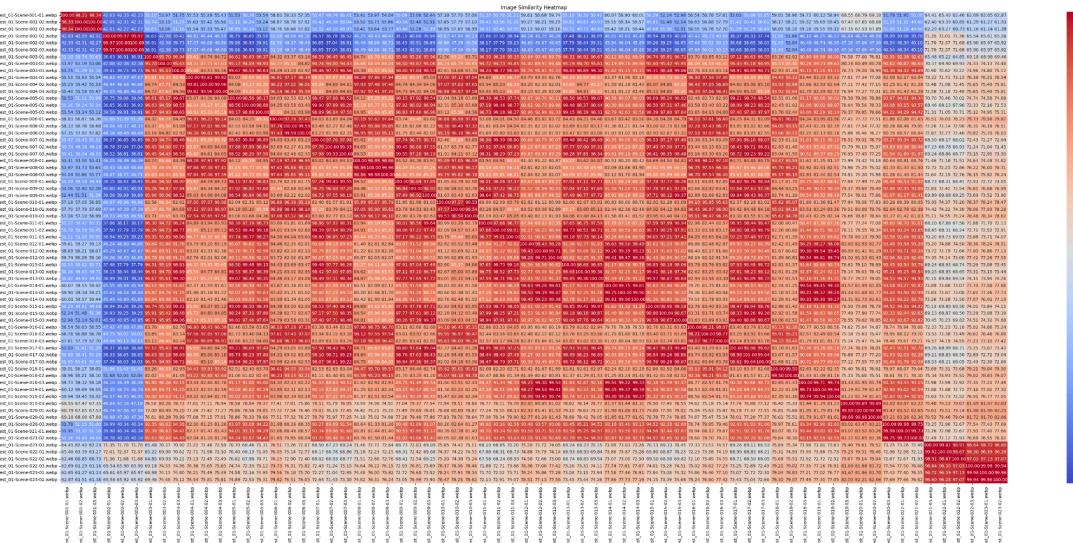
Es importante destacar de qué forma se puede conseguir construir la figura 12, DINOv2 no provee una forma automática para construir un mapa de calor para mostrar los resultados, sino que nos retorna esos feature previamente mencionados. Se presenta la tarea de modificar el formato de salida de DINOv2 para llevarlo al formato de matriz, esto se logra utilizando la librería ‘sklearn’, particularmente la función ‘cosine_similarity()’, que calcula la similitud del coseno, que es una medida de similitud entre dos vectores, su valor varía entre -1 y 1, donde:

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

- 1 significa que los vectores son idénticos (el ángulo es 0 grados).
- 0 indica que los vectores son ortogonales (sin similitud, el ángulo es 90 grados).

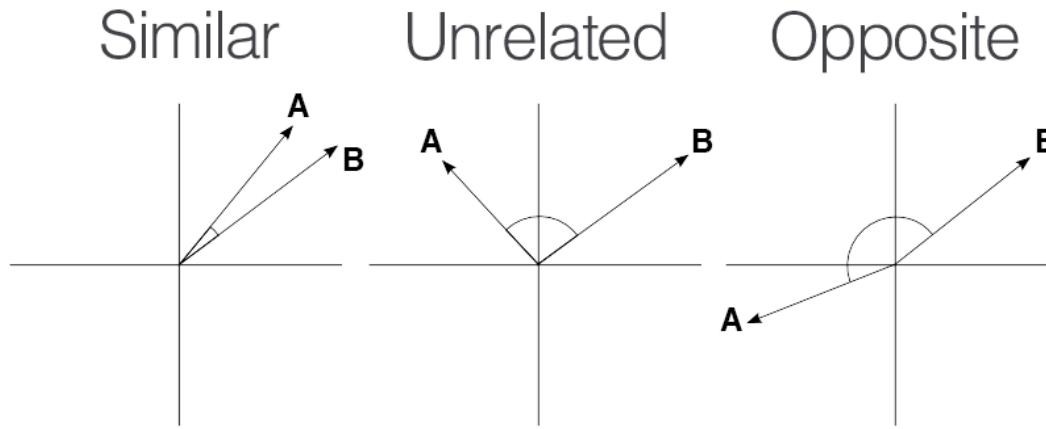
Una representación gráfica se puede apreciar en la figura 13.

Figura 12



Features analizadas de todos los grupos de 3 frames del video de 30 minutos.

Figura 13



Gráficos de similitud del coseno.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Tomando en consideración los resultados, se puede filtrar el diccionario inicial que contiene el video con todos sus frames ya clasificados, y tener menos información repetida que evaluar, se presenta en la figura 14 un posible resultado con menos frames por escena (en este caso, se puede apreciar la escena con id 1, contiene una escena menos).

Figura 14

```
{  
    "duration": "00:30:02.304",  
    "categories": [  
        "drawio"  
    ],  
    "scenes": [  
        {  
            "scene_id": 1,  
            "start_timestamp": "00:00:00.000",  
            "end_timestamp": "00:00:02.467",  
            "images": [  
                {  
                    "name": "scene_1_frame_1",  
                    "path": "video/scenes/scene_1_frame_1",  
                    "category": "drawio"  
                },  
                {  
                    "name": "scene_1_frame_2",  
                    "path": "video/scenes/scene_1_frame_2",  
                    "category": "drawio"  
                }  
            ]  
        },  
        {  
            "scene_id": 2,  
            "start_timestamp": "00:00:02.467",  
            "end_timestamp": "00:00:05.400",  
            "images": [  
            ]  
        }  
    ]  
}
```

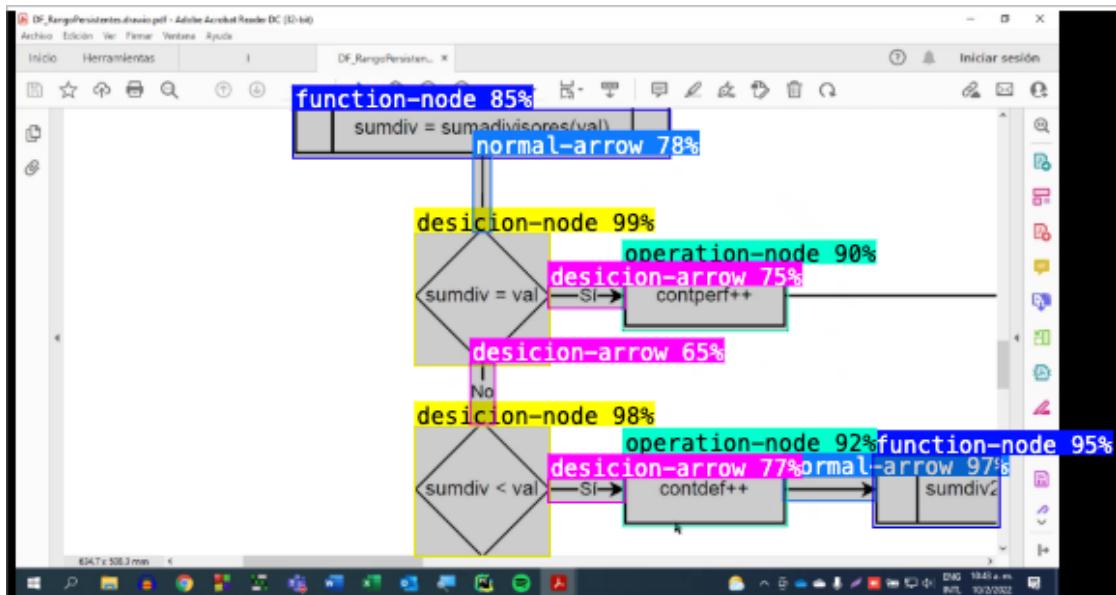
Diccionario filtrado con frames altamente similares(>99.5%).

Predicciones de Imágenes con YOLOv12

Una vez que se clasificaron correctamente las clases válidas y se filtraron los frames potencialmente similares (>99.5%), se puede aplicar el mejor modelo designado para cada clase, ya que como fue mencionado previamente, se prefiere la creación de diversos modelos para abordar cada clase y maximizar los resultados precisos. En la figura 10 y 11 ya se pudo apreciar como resulta el cálculo de las predicciones para imagen que contiene códigos de algoritmos, un ejemplo para diagramas de flujo de programación se muestra en la figura 15 y 16.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Figura 15



Visualización de predicciones de un frame con diagramas de flujo.

Figura 16

```
{  
  "predictions": [  
    {  
      "x": 869,  
      "y": 474.5,  
      "width": 224,  
      "height": 221,  
      "confidence": 0.992,  
      "class": "desicion-node",  
      "class_id": 1,  
      "detection_id": "abae3673-7e7c-462c-947e-84c2fc725136"  
    },  
    {  
      "x": 869,  
      "y": 793,  
      "width": 224,  
      "height": 216,  
      "confidence": 0.985,  
      "class": "desicion-node",  
      "class_id": 1,  
      "detection_id": "53578d67-2676-4335-90ea-cf140cd18568"  
    },  
    {  
      "x": 1444,  
      "y": 788.5,  
      "width": 154,  
      "height": 41,  
      "confidence": 0.972,  
      "class": "normal-arrow",  
      "class_id": 1,  
      "detection_id": "53578d67-2676-4335-90ea-cf140cd18568"  
    }  
  ]  
}
```

Resultado de predicciones de un frame con diagramas de flujo.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Luego de obtener las predicciones necesarias, es necesario establecer un umbral mínimo para considerar que es una frame valido, esto debido a que considerar un frame con un (1) solo diagrama puede no aportar mucho o tratarse de un error del modelo utilizado en YOLOv12, asimismo sucede con los códigos de algoritmos, contabilizar luego de seis (6) líneas ha sido funcional, debido a que esto se piensa para un mínimo código que incluye las librerías mínimas necesarias y un main vacío, con el umbral establecido nos aseguramos de captar frames realmente relevantes, maximizando la información obtenida con la menor cantidad de frames posibles. En la figura 17 se muestra el resultado del diccionario principal luego de pasar los umbrales.

Figura 17

```
"scenes": [
  {
    "scene_id": 1,
    "start_timestamp": "00:00:00.000",
    "end_timestamp": "00:00:02.467",
    "images": [
      {
        "name": "scene_1_frame_1",
        "path": "video/scenes/scene_1_frame_1",
        "category": "drawio",
        "predictions": [
          {
            "x": 1238.5,
            "y": 470.5,
            "width": 275,
            "height": 111,
            "confidence": 0.996,
            "class": "operation-node",
            "class_id": 6,
            "detection_id": "201782e2-df37-4c86-b7ba-017cb0c2961a"
          },
          {
            "x": 1235,
            "y": 793.5,
            "width": 274,
            "height": 119,
            "confidence": 0.993,
            "class": "operation-node",
            "class_id": 6,
            "detection_id": "201782e2-df37-4c86-b7ba-017cb0c2961b"
          }
        ]
      }
    ]
  }
]
```

Resultado del diccionario principal con las predicciones de un frame con diagramas de flujo.

Extracción de texto con Surya OCR

Es crucial poder interpretar con detalles qué está sucediendo dentro de los diagramas o una línea de código, por lo que se requiere saber qué está escrito en los apartados previamente mencionados, para ello, se ha optado por un OCR. Llevar a cabo ésta tarea, representa un reto complejo, no todos los OCR están capacitados para entender texto complejo como puede ser el presente.

La solución propuesta consiste en utilizar nuevamente cv2 para preparar cada caja identificada para el proceso de reconocimiento de caracteres.

Inicialmente, con cv2 se realizan las siguientes tareas:

- Se abre la imagen con el método ‘imread()’.
- Se calcula el centro de la imagen con el resultado de las predicciones en las coordenadas (x , y).
- Con la anchura y altura se delimita la el tamaño de la caja y se delimitan 4 puntos: x1, x2, y1, y2.
- Con los puntos obtenidos se corta la imagen conectando los 4 puntos delimitando “image[y1:y2, x1:x2]”
- Se reescalía con el método “resize()” con un factor de escalado de x2.
- Se cambia el formato de colores a formato en escala de grises.

Con la imagen preprocesada para el OCR de Surya, se utiliza el método ‘recognition_predictor()’ se obtiene un listado muy detallado del contenido extraído, en la figura 18 se puede observar el texto luego de validar correctamente a que frame pertenece el texto. En la figura 19 se puede observar un ejemplo de OCR aplicado a fragmentos de códigos de algoritmos externos al video trabajado en los ejemplos, dado que no contiene código.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Figura 18

```
"predictions": [
    {
        "x": 1238.5,
        "y": 470.5,
        "class": "operation-node",
        "text": "contperf++"
    },
    {
        "x": 1235,
        "y": 793.5,
        "class": "operation-node",
        "text": "contdef++"
    },
    {
        "x": 1043,
        "y": 789,
        "class": "desicion-arrow",
        "text": "Si"
    },
    {
        "x": 1444.5,
        "y": 791.5,
        "class": "normal-arrow"
    },
    {
        "x": 870,
        "y": 194.5,
        "class": "function-node",
        "text": "sumdiv = sumadivisores(val)"
    },
]
```

Resultado del diccionario principal con las predicciones de un frame con diagramas de flujo.

Figura 19

```
"predictions": [
    {
        "text": "char modachr = str[0];",
        "y": 218.48
    },
    {
        "text": "int freqmoda = contarchr (str, modachr), freq;",
        "y": 258.46
    },
    {
        "text": "for ( int ind = 1; str[ind] != 0; ind ++ )",
        "y": 333.68
    },
    {
        "text": "{",
        "y": 373.12
    },
    {
        "text": "freq = contarchr (str, str [ind] ) ;",
        "y": 451.34
    },
    {
        "text": "if ( freq > freqmoda",
        "y": 492.99
    },
    {
        "text": "{",
        "y": 531.09
    },
    {

```

Resultado del diccionario principal con las predicciones de un frame con códigos de algoritmos.

Interpretación y explicación de los datos usando LLMs

El diccionario principal ya cuenta con una variedad de información útil, como lo son:

- Duración
- Categorías
- Escenas:
 - TimeStamps
 - Frames:
 - Categoría
 - Texto

Toda esta información es crucial para utilizar un LLM que sea capaz de ahora entender toda esta información y darle un sentido a la información suelta. El contenido que se puede generar a partir de este punto se ve limitado a las capacidades del LLM elegido, por lo que utilizar uno con buenas capacidades (cantidad de billones con los que fue entrenada) puede marcar la diferencia. A pesar de lo mencionado, es crucial encontrar un modelo que se adapte a la limitaciones técnicas o al presupuesto estimado para estos fines.

El presente ejemplo, propone el uso de ‘Phi-4’ para entornos limitados en vRAM, pero mejores modelos más exigentes pueden producir mejores resultados. Del mismo modo, presupuesto elevados puede permitir el uso del API de OpenAI con sus modelos con capacidad de razonar.

El uso de ‘Phi-4’ está basado nuevamente en ‘transformers’, del cual se debe importar el método ‘pipeline()’, con los siguientes parámetros:

- ‘**type** (str)’: Se establece el tipo de generación que se va a estar creando.
- ‘**model** (str)’: Se establece el modelo que se estará usando.
- ‘**model_kwargs** (dict)’: Se establecen los parámetros adicionales para el modelo.
- ‘**device_map** (str)’: Se establece el dispositivo que se usará (cpu | gpu (cuda:0))

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Se procede a crear el objeto de mensaje que es una arreglos de solicitudes y finalmente se envía el mensaje como parámetro al pipeline y se especifica el parámetro ‘max_new_tokens (int)’ como cantidad máxima de caracteres, al utilizar el método y aplicando el procesamiento necesario podemos obtener un resultado presente en la figura 20, 21, 22 y 23.

Figura 20

```
"predictions": [
    {
        "x": 1238.5,
        "y": 470.5,
        "class": "operation-node",
        "text": "contperf++",
        "explanation": "Nodo de operación: Incrementa el contador de números perfectos ('contperf') cuando el número evaluado"
    },
    {
        "x": 1235,
        "y": 793.5,
        "class": "operation-node",
        "text": "contdef++",
        "explanation": "Nodo de operación: Incrementa el contador de números deficientes ('contdef') cuando la suma de los div"
    },
    {
        "x": 1043,
        "y": 789,
        "class": "desicion-arrow",
        "text": "Sí",
        "explanation": "Flecha de decisión: Indica el flujo tomado cuando la condición previa ('sumadiv < val') se cumple, lle"
    },
    {
        "x": 1444.5,
        "y": 791.5,
        "class": "normal-arrow",
        "explanation": "Flecha normal: Indica el flujo natural del diagrama desde la evaluación hacia la función 'sumadivisore"
    },
]
```

Resultado del LLM al analizar las predicciones de los diagramas de flujo.

Figura 21

```
{
    "x": 870,
    "y": 194.5,
    "class": "function-node",
    "text": "sumdiv = sumadivisores(val)",
    "explanation": "Nodo de función: Llama a la función 'sumadivisores' pasando el valor 'val' y guarda el resultado obten"
},
{
    "x": 868.5,
    "y": 792,
    "class": "desicion-node",
    "text": "sumadiv = val",
    "explanation": "Nodo de decisión: Evalúa si la suma de los divisores ('sumadiv') es exactamente igual al valor origina"
},
{
    "x": 868.5,
    "y": 475.5,
    "class": "desicion-node",
    "text": "sumadiv < val",
    "explanation": "Nodo de decisión: (Probable error tipográfico; debería decir 'sumadiv < val'). Evalúa si la suma de di"
},
{
    "x": 1043,
    "y": 469,
    "class": "desicion-arrow",
    "text": "Sí",
    "explanation": "Flecha de decisión: Indica el camino tomado cuando la condición 'sumadiv < val' es verdadera, llevando"
},
```

Resultado del LLM al analizar las predicciones de los diagramas de flujo.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Figura 22

```
"predictions": [
    {
        "text": "int freqmoda = contarchr (str, modachr), freq;",
        "explanation": "Declara la variable freqmoda inicializada con el resultado de la función contarchr que cuenta cuád
        "y": 258.46
    },
    {
        "text": "for ( int ind = 1; str[ind] != 0; ind ++ )",
        "explanation": "Inicia un ciclo for que recorre cada carácter del arreglo str, comenzando en la posición 1 hasta
        "y": 333.68
    },
    {
        "text": "{",
        "explanation": "Apertura del bloque del ciclo for, indica el inicio de las instrucciones que se ejecutarán en cad
        "y": 373.12
    },
    {
        "text": "freq = contarchr (str, str [ind] ) ;",
        "explanation": "Llama a la función contarchr para contar cuántas veces aparece el carácter actual (str[ind]) dent
        "y": 451.34
    },
    {
        "text": "if ( freq > freqmoda",
        "explanation": "Evalúa si la frecuencia recién calculada (freq) es mayor que la frecuencia más alta registrada ha
        "y": 492.99
    },
    {
        "text": "{",
        "explanation": "Inicio del bloque condicional if, se ejecuta solo si la condición freq > freqmoda es verdadera."
        "y": 531.09
    }
]
```

Resultado del LLM al analizar las predicciones del código de algoritmos.

Figura 23

```
"predictions": [
    {
        "text": "int freqmoda = contarchr (str, modachr), freq;",
        "explanation": "Declara la variable freqmoda inicializada con el resultado de la función contarchr que cuenta cuád
        "y": 258.46
    },
    {
        "text": "for ( int ind = 1; str[ind] != 0; ind ++ )",
        "explanation": "Inicia un ciclo for que recorre cada carácter del arreglo str, comenzando en la posición 1 hasta
        "y": 333.68
    },
    {
        "text": "{",
        "explanation": "Apertura del bloque del ciclo for, indica el inicio de las instrucciones que se ejecutarán en cad
        "y": 373.12
    },
    {
        "text": "freq = contarchr (str, str [ind] ) ;",
        "explanation": "Llama a la función contarchr para contar cuántas veces aparece el carácter actual (str[ind]) dent
        "y": 451.34
    },
    {
        "text": "if ( freq > freqmoda",
        "explanation": "Evalúa si la frecuencia recién calculada (freq) es mayor que la frecuencia más alta registrada ha
        "y": 492.99
    },
    {
        "text": "{",
        "explanation": "Inicio del bloque condicional if, se ejecuta solo si la condición freq > freqmoda es verdadera."
        "y": 531.09
    }
]
```

Resultado del LLM al analizar las predicciones del código de algoritmos.

Implementación adicional - Recreación de contenido con cliente de OpenAI

Como fue mencionado previamente, el uso de modelos con la capacidad de razonar, puede sacarle un provecho muy importante a las predicciones y explicaciones que fueron creadas a partir del uso de un LLM local con menos potencia. Se puede instruir con una serie de instrucciones muy específicas de qué forma se debe tratar el código de algoritmos o los diagramas de flujo de programación para que intente conseguir el resultado más fiel posible.

Utilizando el método ‘client.chat.completions.create()’ del cliente de la API de OpenAI, con el set de instrucciones, es posible obtener un resultado como el mostrado en la figura 25, aunque no es 100% igual, es una representación del código explicado en dicha sesión muy fiel. El código original se muestra en la figura 24.

Figura 24

```
int wordcount(char str[])
{
    int ind, countword = 0;

    for ( ind = 0; str[ind] != 0; ind++ )
        if ( str[ind] != ' '
            && (str[ind+1] == ' ' || str[ind+1] == 0) )
            countword++;
}
```

```
for ( int ind = 1; str[ind] != 0; ind++ )
{
    freq = contarchr(str,str[ind]);
    if ( freq > freqmoda )
    {
        freqmoda = freq;
        modachr = str[ind];
    }
}
return modachr;
```

```

int contarchr(char str[], char chr)
{
    int ind, contchr = 0;

    for ( ind = 0; str[ind] != 0; ind++ )
        if ( tolower(chr) == tolower(str[ind]) )
            contchr++;

    return contchr;
}

```

Código original extraído de una sesión de clases.

Figura 25

```

'int contarchr(char str[], char chr) {\n'
'    int ind, contchr = 0;\n'
"    for (ind = 0; str[ind] != '\x00'; ind++) {\n"
'        if (tolower(chr) == tolower(str[ind])) {\n'
'            contchr++; \n'
'        }\n'
'    }\n'
'    return contchr;\n'
'}\n'
'\n'
'char modachar(char str[]) {\n'
'    char moda = str[0];\n'
"    int freqmoda = contarchr(str, moda);\n"
"    for (int ind = 1; str[ind] != '\x00'; ind++) {\n"
'        int freq = contarchr(str, str[ind]);\n'
'        if (freq > freqmoda) {\n'
'            moda = str[ind];\n'
'            freqmoda = freq;\n'
'        }\n'
'    }\n'
'    return moda;\n'
'}\n'
'\n'
'int wordcount(char str[]) {\n'
'    int ind, countword = 0;\n'
"    for (ind = 0; str[ind] != '\x00'; ind++) {\n"
"        if (str[ind] != ' ' && (str[ind + 1] == ' ' || str[ind + 1] == "
"'\x00')) {\n"
'            countword++; \n'
'        }\n'
'    }\n'
'    return countword;\n'
'}\n'

```

```

'int wordcount(char str[]) {\n'
'    int ind, countword = 0;\n'
"    for (ind = 0; str[ind] != '\x00'; ind++) {\n"
"        if (str[ind] != ' ' && (str[ind + 1] == ' ' || str[ind + 1] == "
"'\x00')) {\n"
'            countword++; \n'
'        }\n'
'    }\n'
'    return countword;\n'
'}\n'

```

Reconstrucción del código de algoritmos con el uso exclusivo de las predicciones.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Servidor HTTP, Base de Datos & Cola de Mensajería

El servidor HTTP es la puerta de entrada de todas las funcionalidades aisladas del bot (integraciones con plataformas externas son expuestas de forma individual). Es quien se encarga de recibir las peticiones del cliente, entiéndase peticiones para: autenticación, subida de videos, consulta de videos procesados y la validación de los mismos datos asegurando el uso de datos correctos en todo momento. Su función principal va desde recibir el video, mandarlo a la cola para que el bot lo procese y guardar el resultado en la base de datos.

Figura 26

```
@asynccontextmanager
async def lifespan(app: FastAPI) -> Generator[None, Any, None]:
    db: AsyncIOMotorDatabase[Any] = get_main_db()
    try:
        await db.command(command="ping")
    except PyMongoError as e:
        raise RuntimeError(f"Failed to connect to MongoDB: {e}")
    yield

VIDEO_DIR = Path("videos")
VIDEO_DIR.mkdir(parents=True, exist_ok=True)

load_dotenv()  "dotenv": Unknown word.
app = FastAPI(
    title="LogicMate Backend",
    description="LogicMate Backend API, a RESTful API for LogicMate",
    version="1.0.0",
    lifespan=lifespan,
)

app.add_middleware(
    middleware_class=CORSMiddleware,
    allow_origins=[os.getenv(key="FRONT_END_URL", default="*")],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

@app.get(path="/health", tags=["root"])
async def health_check() -> dict:
    return {"status": "ok", "message": "LogicMate Backend running"}

app.include_router(
    router=user_router,
)
app.include_router(
    router=auth_router,
)
app.include_router(
    router=video_router,
)
```

Inicialización del servidor HTTP con FastAPI.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Dicho servidor continua con el lenguaje de programación Python con el Framework FastAPI, este tiene una integración directa con OpenAPI, que se encuentra en “/docs” que nos proporciona la siguiente documentación:

Figura 27

The screenshot shows the API documentation for the LogicMate Backend API, version 1.0.0, using OAS 3.1. The interface is organized into sections: root, users, auth, videos, and Schemas.

- root**: Contains a single endpoint: GET /health - Health Check.
- users**: Contains five endpoints:
 - GET /api/v1/users/ - List Users
 - POST /api/v1/users/ - Create User
 - GET /api/v1/users/{user_id} - Get User
 - PUT /api/v1/users/{user_id} - Update User
 - DELETE /api/v1/users/{user_id} - Delete User
- auth**: Contains one endpoint: POST /api/v1/auth/login - Create Token.
- videos**: Contains four endpoints:
 - GET /api/v1/videos/ - List Videos
 - POST /api/v1/videos/ - Create Video
 - GET /api/v1/videos/{video_id} - Get Video
 - POST /api/v1/videos/process - Process Videos
- Schemas**: Shows definitions for various schema objects:
 - Body_process_videos_api_v1_videos_process_post
 - HTTPValidationError
 - Image
 - Prediction
 - Scene
 - Scene
 - TokenRequestDTO

Todos los métodos HTTP disponibles a través de la API.

Donde se puede observar los endpoints que el servidor provee para consultar la información, esto cuenta con la información de cada uno, el tipo de dato que recibe y el que retorna.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Toda esta información es almacenada en una base de datos no relacional (NoSQL), para manejar la complejidad de los datos que contiene el resultado final del procesamiento de un video, se optó por utilizar una base de datos documental, específicamente MongoDB, desplegada desde un contenedor de Docker.

Figura 28

```
►Run All Services | You, 3 days ago | 2 authors (You and one other)
services:
  db:
    ▷ Run Service
    db:
      image: mongo:latest
      container_name: db
      ports:
        - "27017:27017"
      volumes:
        - mongo_main_data:/data/db
      environment:
        MONGO_INITDB_ROOT_USERNAME: ${MONGO_INITDB_ROOT_USERNAME}
        MONGO_INITDB_ROOT_PASSWORD: ${MONGO_INITDB_ROOT_PASSWORD}
        MONGO_INITDB_DATABASE: ${MONGO_INITDB_DATABASE}

  db_studio:
    ▷ Run Service
    db_studio:
      image: mongo-express:latest
      container_name: db_studio
      restart: always
      ports:
        - "8082:8081"
      environment:
        ME_CONFIG_MONGODB_ADMINUSERNAME: ${MONGO_INITDB_ROOT_USERNAME}
        ME_CONFIG_MONGODB_ADMINPASSWORD: ${MONGO_INITDB_ROOT_PASSWORD}
        ME_CONFIG_MONGODB_URL: ${MONGO_URL}
        ME_CONFIG_BASIC_AUTH: ${ME_CONFIG_BASIC_AUTH}
      depends_on:
        - db
```

Docker Compose de la base de datos y la interfaz visual de la misma.

Se utiliza Mongo Express para desplegar una interfaz visual, donde se puede hacer la gestión y visualización de todos los datos guardados. La colección ‘processed_calls’ es la que guarda los id de las llamadas de teams para evitar el re-procesamiento de las mismas. La colección ‘users’ contiene los datos básicos de los usuarios. Finalmente la colección ‘vídeos’, es el resultado del procesamiento de aquellas sesiones que fueron sometidas.

Figura 29

Mongo Express Database: logicmate_db

Collections

				Collection Name	
				processed_calls	
				users	
				videos	

Database Stats

Collections (incl. system.namespaces)	3
Data Size	3.85 KB
Storage Size	77.8 KB
Avg Obj Size #	549 Bytes
Objects #	7
Indexes #	4
Index Size	98.3 KB

Tablas necesarias en la base de datos.

Es importante entender que el servidor HTTP no es una integración por encima del Bot explicado previamente, sino que este vive de manera aislada y se levanta de forma independiente para mantener las responsabilidades segregadas. Por lo mismo tanto, es necesario comunicar el Bot con el servidor, este no cuenta con ningún framework HTTP para mantener comunicación mediante el protocolo web y para mantenerlo de esta forma (ya que no es algo que necesite), se implemente la cola de mensajería.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Para el desarrollo de dicha protocolo de comunicación se utilizó Celery que es “una cola de tareas asíncrona de código abierto”, el cual tiene un SDK para Python, lo que permite una integración directa. Esta corre encima de una base de datos en tiempo real en memoria, Redis, que se encarga de mantener los mensajes en cola. Celery provee un mecanismo para envolver un programa en un servicio de workers, lo que sin necesidad de modificar el bot, este ahora pueda recibir diversos videos, procesarlos en paralelo e incluso procesarlos aunque la petición se haya realizado mientras el bot estaba apagado.

Figura 30

```
celery = Celery(  
    main="logicmate_bot",  
    broker=config.get(key="/CELERY_BROKER_URL"),  
    backend=config.get(key="/CELERY_RESULT_BACKEND"),  
)  
  
celery.conf.imports = ["logicmate.main"]  
  
logging.info(  
    msg="Celery broker and backend configured successfully.",  
)  
  
logging.info(  
    msg="Bot is waiting for tasks to process videos.",  
)  
  
  
@celery.task(name="logicmate_bot.process_video")  
def process_video_task(  
    video_bytes: bytes, users_emails: list[str], current_user_email: str  
) -> None:  
    with open(file="/tmp/video.mp4", mode="wb") as f:  
        f.write(buffer=video_bytes)  
  
    logging.info(  
        msg=f"Processing video for user {current_user_email} and users {users_emails}. to notify them.",  
)  
  
    start_bot(  
        video_path="/tmp/video.mp4",  
        config=config,  
)
```

Inicialización del bot con la cola de mensajería Celery-Redis con “workers” para multi-procesamiento.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Celery provee una interfaz visual web para que sea posible observar las máquinas que sirven como workers y las tasks (mensajes) pendientes en la cola.

Figura 31

The figure consists of two screenshots of the Celery Flower web interface. The top screenshot shows the 'Workers' page. It has a header with links for Flower, Workers, Tasks, Broker, and Documentation. Below the header, there is a search bar and a table with columns: Worker, Status, Active, Processed, Failed, Succeeded, Retried, and Load Average. One worker is listed: 'celery@Vladimirs-MacBook-Air-116.local' is 'Online' with 0 active, 0 processed, 0 failed, 0 succeeded, 0 retried, and a load average of 2.77, 3.47, 4.28. A summary row 'Total' shows 0 for all metrics. The bottom screenshot shows the 'Tasks' page. It has a similar header and search bar. The table has columns: Name, UUID, State, args, kwargs, Result, and Received. Four tasks are listed under the 'logicmate_bot.process_video' name, each with a different UUID. All tasks are in the 'SUCCESS' state. The 'args' column shows binary data, and the 'kwargs' column shows an empty dictionary. The 'Result' column shows 'None' and the 'Received' column shows the date '2025-07-15 23:26'.

Interfaz web de Celery para visualizar la cola de videos pendientes a procesar..

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Y se puede observar como el bot ahora maneja los diversos videos con workers diferentes de ser necesario.

Figura 32

```
----- celery@Vladimirs-MacBook-Air-116.local v5.5.3 (immunity)
--- **** ---
-- ***** macOS-26.0-arm64-arm-64bit 2025-07-15 19:17:50
- *** -
- ** ----- [config]
- ** ----- .> app:      logicmate_bot:0x13fa62740
- ** ----- .> transport: redis://localhost:6379/0
- ** ----- .> results:   redis://localhost:6379/1
- *** - * -- .> concurrency: 8 (prefork)
-- ***** -- .> task events: OFF (enable -E to monitor tasks in this worker)
--- **** --
----- [queues]
    .> celery          exchange=celery(direct) key=celery

[tasks]
    . logicmate_bot.process_video

[2025-07-15 19:17:51,529: INFO/MainProcess] Connected to redis://localhost:6379/0
[2025-07-15 19:17:51,543: INFO/MainProcess] mingle: searching for neighbors
[2025-07-15 19:17:52,620: INFO/MainProcess] mingle: all alone
[2025-07-15 19:17:52,620: INFO/MainProcess] celery@Vladimirs-MacBook-Air-116.local ready.
[2025-07-15 19:19:14,325: INFO/MainProcess] Events of group {task} enabled by remote.
[2025-07-15 19:26:03,824: INFO/MainProcess] Task logicmate_bot.process_video[89f49575-1f87-4f04-8ac0-caf42de24124] received
[2025-07-15 19:26:04,074: INFO/ForkPoolWorker-8] Processing video for user None and users ['natashalopez@logicmatebot.onmicrosoft.com',
', 'vladimircuriel@logicmatebot.onmicrosoft.com']. to notify them.
[2025-07-15 19:26:04,388: INFO/ForkPoolWorker-8] Task logicmate_bot.process_video[89f49575-1f87-4f04-8ac0-caf42de24124] succeeded in
0.36865525000030175s: None
[2025-07-15 19:26:04,494: INFO/MainProcess] Task logicmate_bot.process_video[14b7e4d4-e754-4319-af21-eaa74f128667] received
[2025-07-15 19:26:04,501: INFO/ForkPoolWorker-8] Processing video for user None and users ['vladimircuriel@logicmatebot.onmicrosoft.com']. to notify them.
[2025-07-15 19:26:04,502: INFO/ForkPoolWorker-8] Task logicmate_bot.process_video[14b7e4d4-e754-4319-af21-eaa74f128667] succeeded in
0.00135270900195566416s: None
[2025-07-15 19:26:04,526: INFO/MainProcess] Task logicmate_bot.process_video[edc6cc4a-993b-4b47-9bbc-3ed7d823a0de] received
[2025-07-15 19:26:04,547: INFO/ForkPoolWorker-8] Processing video for user None and users ['natashalopez@logicmatebot.onmicrosoft.com',
', 'vladimircuriel@logicmatebot.onmicrosoft.com']. to notify them.
[2025-07-15 19:26:04,555: INFO/ForkPoolWorker-8] Task logicmate_bot.process_video[edc6cc4a-993b-4b47-9bbc-3ed7d823a0de] succeeded in
0.010445541003718972s: None
[2025-07-15 19:26:46,654: INFO/MainProcess] Task logicmate_bot.process_video[c00f50e4-238c-48dd-8d80-9fcf9ab94afb] received
[2025-07-15 19:26:46,659: INFO/ForkPoolWorker-8] Processing video for user None and users ['vladimircuriel@logicmatebot.onmicrosoft.c
```

Procesamiento de videos en paralelo con Celery.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Integración con Microsoft Teams

Previamente fue mencionado que la integración de Microsoft Teams (Teams) es un proceso que se hace fuera del servidor y funciona de manera independiente. Esto separa las limitaciones que la integración conlleva, como el requerimiento de una organización con diversidad de licencias de Microsoft 365 activas para una cantidad determinada de usuarios y los permisos necesarios. Las licencias necesarias son:

- Microsoft 365: Es la base de todo, lo que permite la creación de usuarios, almacenamiento de videos en SharePoint, calendarios y sobre todo la Graph API para realizar la comunicación.
- Microsoft 365 E5: Se necesita esencialmente para otorgar permisos a la Graph API de telefonía, lo que permite usar el endpoint '/callRecords', que contiene la información esencial de la llamada para llegar al video.
- Microsoft Teams: Es la plataforma en sí para realizar las videoconferencias, su uso no va más allá del uso normal para agendar llamadas y crear las sesiones.

Figura 33

The screenshot shows the 'Sus productos' (Your products) section in the Microsoft 365 Admin Center. It displays a list of products assigned to users, including Microsoft 365 Business Standard, Microsoft 365 E5 (no Teams), Microsoft Entra ID Free, and Microsoft Teams Enterprise. The table includes columns for Nombre del producto, Licencias asignadas, Cantidad comprada, Licencias disponibl..., Estado de la suscri., Fecha de renovaci..., Perfil de facturaci..., Canal de compra, and Tipo de licencia. A search bar and filter options are visible at the top of the table.

Nombre del producto	Licencias asignadas	Cantidad comprada	Licencias disponibl...	Estado de la suscri...	Fecha de renovaci...	Perfil de facturaci...	Canal de compra	Tipo de licencia	
Microsoft 365 Business Standard	NUEVO	1	25	24	Activo	Se renueva el 7/8/2025 con 25 licencias de pago	LogicMate	Comercial directo	Basada en el uso
Microsoft 365 E5 (no Teams)	NUEVO	1	25	24	Activo	10/8/2025	LogicMate	Comercial directo	Basada en el uso
Microsoft Entra ID Free	NUEVO	No disponible	1	No disponible	Activo	No disponible	LogicMate	Comercial directo	Basada en el uso
Microsoft Teams Enterprise	NUEVO	1	25	24	Activo	10/8/2025	LogicMate	Comercial directo	Basada en el uso

Portal web de Microsoft 365 con las licencias necesarias.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Es necesario tener un Tenant en Microsoft Entra ID, esto para registrar un aplicación, obtener la Graph API y poder acceder a los endpoints que esta provee.

Figura 34

The screenshot shows the Microsoft Entra App registrations page for the 'LogicMate Bot' application. The left sidebar includes links for Overview, Quickstart, Integration assistant, Diagnose and solve problems, Manage (Branding & properties, Authentication, Certificates & secrets, Token configuration, API permissions, Expose an API, App roles, Owners, Roles and administrators, Manifest), Support + Troubleshooting (New support request), and Get Started Documentation. The main content area displays the 'Essentials' section with details like Display name (LogicMate Bot), Application (client) ID (35726a2c-4801-4186-a8fb-de8696aeab75), Object ID (1bc33f65-a45f-4ac4-b92c-e792702b5192), Directory (tenant) ID (aed2e25d-43e4-4b04-9491-b1020e24d885), Supported account types (My organization only), Client credentials (0_certificate_1_secret), Redirect URLs (Add a Redirect URI), Application ID URI (Add an Application ID URI), and State (Activated). A note at the bottom states: 'Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure Active Directory Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph.' Below this is a 'Build your application with the Microsoft identity platform' section with icons for various services.

Tenant creado para poder registrar aplicaciones.

Con la aplicación creada, se necesitan otorgar una serie de permisos:

Figura 35

The screenshot shows the Microsoft Entra App registrations page for the 'LogicMate Bot' application, specifically the 'API permissions' section. The left sidebar includes links for Overview, Quickstart, Integration assistant, Diagnose and solve problems, Manage (Branding & properties, Authentication, Certificates & secrets, Token configuration, API permissions, Expose an API, App roles, Owners, Roles and administrators, Manifest), Support + Troubleshooting (New support request), and Get Started Documentation. The main content area displays the 'Configured permissions' section, which lists permissions granted by users/admins. It shows 13 permissions under 'Microsoft Graph (13)': Calendars.Read, Calendars.ReadBasic, CallRecords.Read.All, Directory.Read.All, Files.Read, Files.Read.All, Files.Read.All, Group.Read.All, OnlineMeetings.Read, OnlineMeetings.Read, Sites.Read.All, User.Read, and User.Read.All. All permissions are marked as 'Granted for LogicMate'. There is also a link to 'Add a permission' and 'Grant admin consent for LogicMate'.

Permisos de la API para acceder a los recursos necesarios.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Todos estos permisos son necesarios para poder acceder a la información de los usuarios de la organización, leer el calendario de los usuarios y del Teams, leer directorios para poder encontrar la grabación, permiso a leer archivos para poder descargar la grabaciones, permiso a obtener los registros de las llamadas, permiso para leer los grupos de SharePoint que es donde están las grabaciones.

Al tener todos los permisos necesarios, se procede a realizar un integración por código, que se encarga tanto de hacer polling de datos para obtener las nuevas sesiones (o una acercamiento más avanzado con Web Hooks, para que sea Microsoft la que avisa que hay un evento nuevo). Dicha integración se encarga tanto del mencionado polling como también mandar mensajes a la cola de mensajería y proveer un endpoint para obtener el calendario de un usuario.

Figura 36

```
@asynccontextmanager
async def lifespan(app: FastAPI) → Generator[None, Any, None]:
    """Start background polling on startup, clean up on shutdown."""
    thread = threading.Thread(target=polling_loop, args=(bot, celery), daemon=True)
    thread.start()
    logging.info(msg="Started background polling thread")
    yield
    logging.info(msg="Shutting down")

app = FastAPI(
    title="LogicMate Teams Bot API",
    description="...",
    version="1.0.0",
    lifespan=lifespan,
)

@app.get(path="/users/{user_email}/calendar", tags=["calendar"])
async def list_user_calendar(user_email: str) → list[dict[str, Any]]:
    try:
        events: list[dict[Any, Any]] = bot.get_user_calendar_events(user_email=user_email)
    except Exception as e:
        logging.error(msg=f"Error fetching calendar for {user_email}: {e}")
        raise HTTPException(
            status_code=500, detail="Failed to retrieve calendar events"
        )
    return events
```

Inicialización del servidor con la integración de Teams.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Como dicha integración tiene la responsabilidad de siempre estar haciendo polling de datos hacia el Graph API, pero también tiene que exponer el endpoint del calendario, es necesario que el polling se ejecute en un hilo aparte para no bloquear el endpoint.

Esto se conecta con la misma base de datos previamente mostrada, es quien se encarga de la colección “processed_calls” para agregar los id de las sesiones de Teams. El polling de datos sucede cada determinado tiempo:

Figura 37

```
def polling_loop(bot: LogicMateTeamsBot, celery: celery.Celery) -> None:
    """ Enqueue celery task to process the video.
    * Mark call_id as processed locally.
    """
    while True:
        logging.info(msg="Polling for new Teams call recordings ...")
        try:
            for call in bot.get_call_details():
                call_id: Any = call["call_id"]
                participants: Any = call["participants"]

                recording_url: Any = call["recording_url"]
                if not recording_url:
                    logging.warning(msg=f"No recording URL for call_id={call_id}")
                else:
                    logging.info(msg=f"Downloading video for call_id={call_id}")
                    response: requests.Response = requests.get(url=recording_url)
                    response.raise_for_status()
                    video_bytes: bytes | Any = response.content

                    task: Any = celery.send_task(
                        name="logicmate_bot.process_video",
                        args=[video_bytes, participants, None],
                    )
                    logging.info(
                        msg=f"Enqueued video call_id={call_id} with {len(obj/participants)} participants as task_id={task.id}"
                    )

                bot.db.add(call_id=call_id)

        except Exception as e:
            logging.error(msg=f"Error during polling loop: {e}")

INFO:     Uvicorn running on http://0.0.0.0:8081 (Press CTRL+C to quit)
INFO:     Started reloader process [39652] using StatReload
INFO:     Started server process [39658]
INFO:     Waiting for application startup.
INFO: Polling for new Teams call recordings...
INFO: Started background polling thread
INFO:     Application startup complete.
INFO: Downloading video for call_id=54d4ed84-dc49-4071-8248-888bf5a24308
INFO: Enqueued video call_id=54d4ed84-dc49-4071-8248-888bf5a24308 with 2 participants as task_id=54fe3368-82c1-471f-9a41-9a84ff767c72
INFO: Downloading video for call_id=e7aff475-29a9-456d-9bca-d477a91f2723
INFO: Enqueued video call_id=e7aff475-29a9-456d-9bca-d477a91f2723 with 1 participants as task_id=b2230e4e-094b-404d-92dc-f45942e76145
INFO: Polling for new Teams call recordings...
```

Función de solicitud de datos a Teams de forma periódica.

Análisis de los Resultados

El presente análisis se fundamenta en los resultados obtenidos a lo largo del proceso de segmentación, clasificación y validación de escenas de videos académicos. La combinación de técnicas de visión por computadora y procesamiento de lenguaje natural permitió optimizar el análisis, reduciendo drásticamente la cantidad de frames a procesar y garantizando la extracción de información significativa. A continuación, se describen en detalle los hallazgos cuantitativos y cualitativos.

Segmentación y Extracción de Frames con PySceneDetect

Para abordar la alta densidad de frames en videos (donde, por ejemplo, un video de 10 minutos grabado a entre 16 y 24 FPS puede alcanzar aproximadamente 12,600 frames), se implementó PySceneDetect. Con esta herramienta se logró:

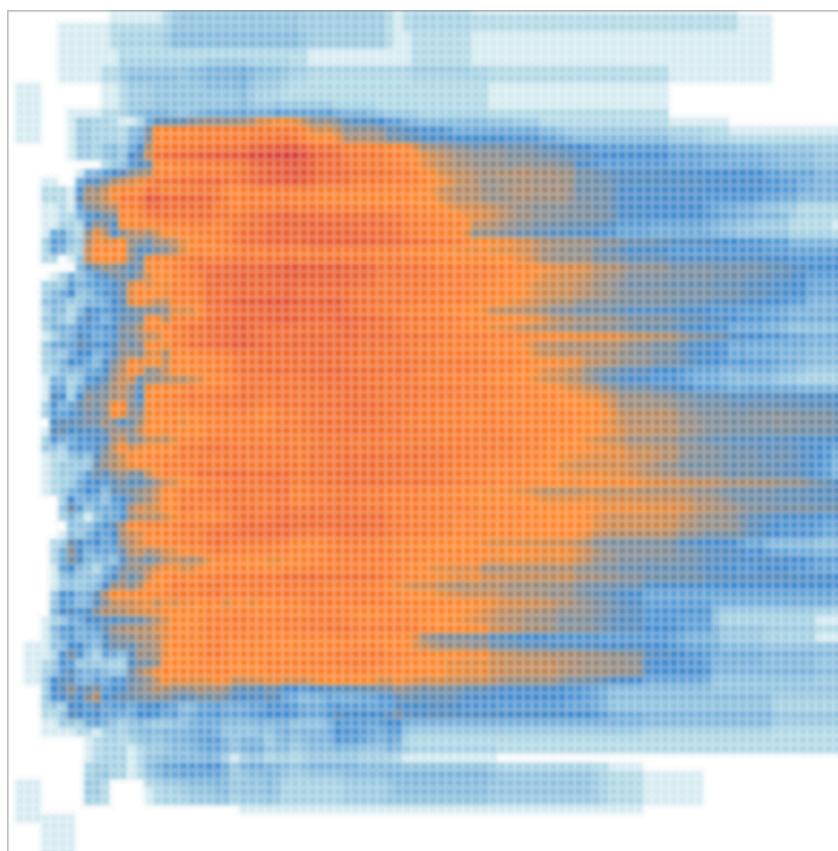
- Reducción de Frames: De cada escena se extraen 3 frames (inicio, mitad y final), lo que reduce significativamente el volumen de datos sin comprometer la representación visual.
- Parámetros de Configuración:
 - num_images: 3
 - image_extension: “png” (por su alta calidad estándar)
 - encoder_param: 9 (garantizando la máxima calidad en la compresión)
 - Interpolación: Se empleó el método LANCZOS4, basado en la función sinc(), que ofrece resultados visuales muy precisos a pesar de su mayor costo computacional.
 - Resolución de Video: Las grabaciones en alta calidad (entre 720p y 1080p) aseguran que el análisis de píxeles (por ejemplo, en espacio HSV) se realice con alta fidelidad.

Estos ajustes permiten transformar un conjunto masivo de frames en un subconjunto representativo, facilitando el procesamiento posterior.

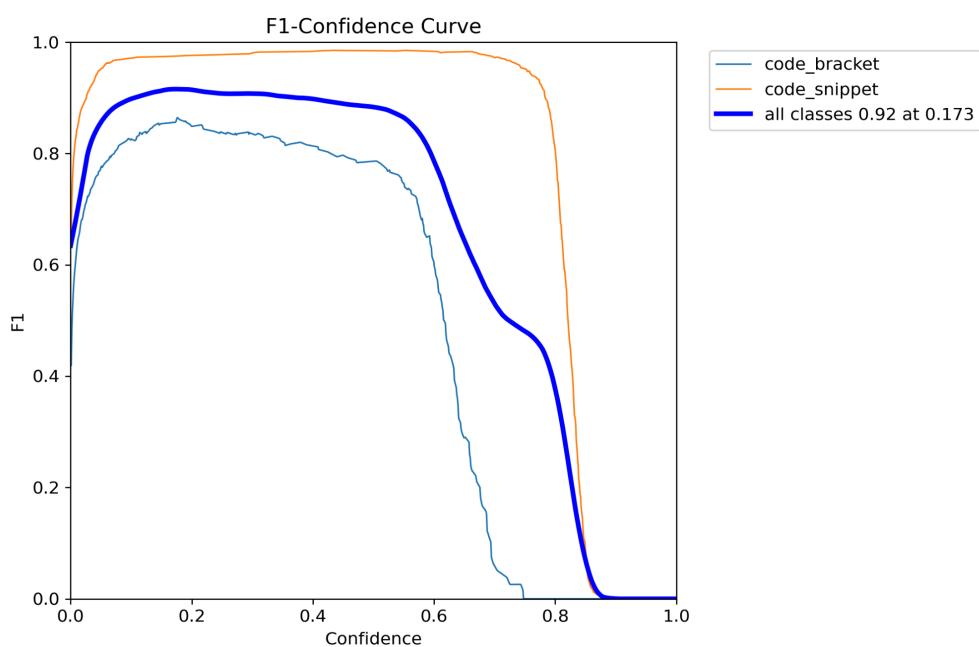
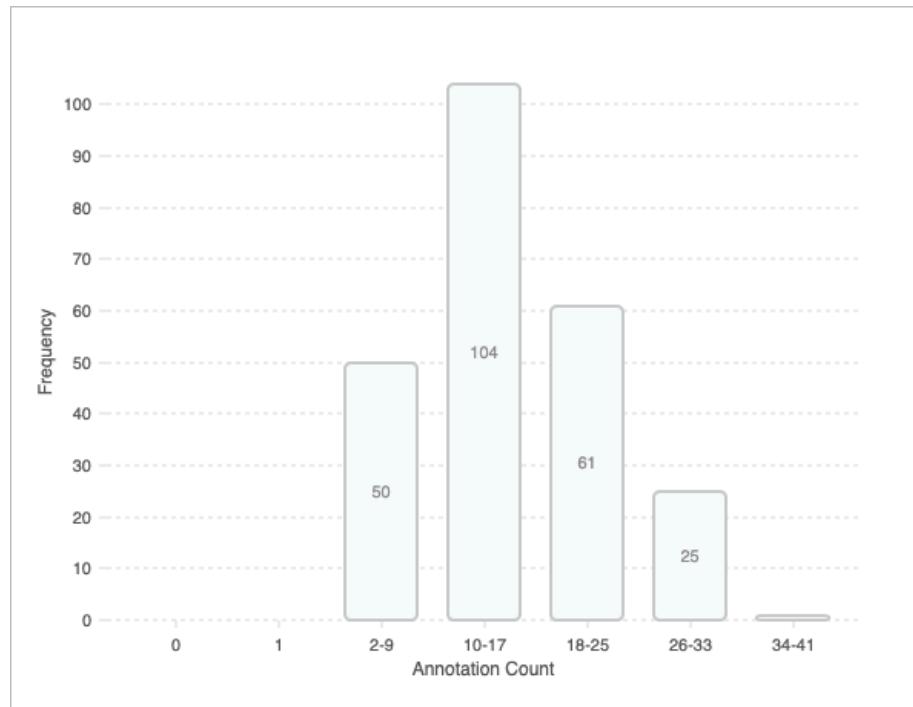
Clasificación de Imágenes con YOLOv12

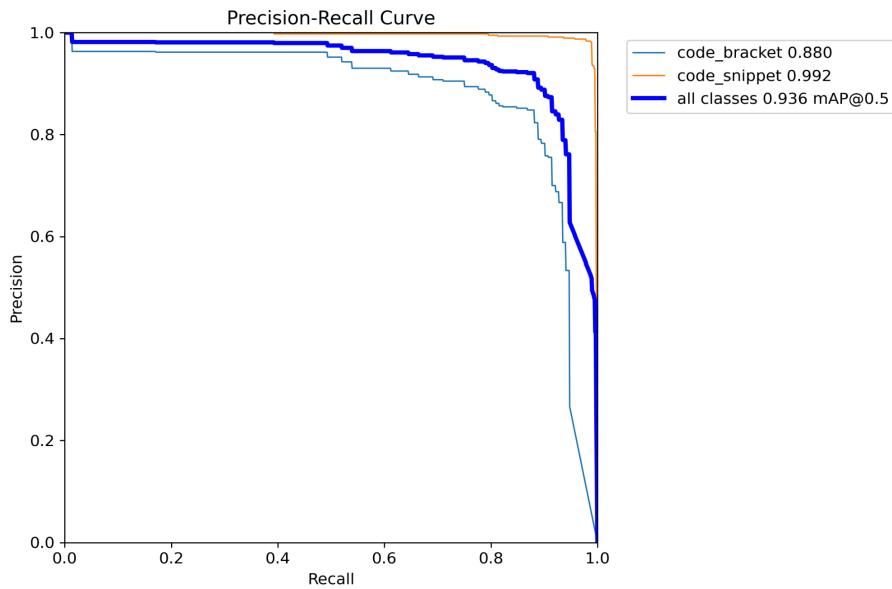
Para la detección y clasificación de elementos (códigos y diagramas), se entrenaron modelos basados en YOLOv12 sobre datasets previamente creados a través de la segmentación de PySceneDetect. Las tablas 10, 11, 12 y 13, previamente mostradas ilustran el resultado de los modelos entrenados y sus distintos parámetros. Se puede explorar más resultados de interés a través de los diversos modelos presentes, como lo pueden ser los mapas de calor de las anotaciones y el promedio de las clases de los datasets que fueron creados para entrenar a YOLOv12 en las figuras 52, 53, 54 y 55.

Figura 52



Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación



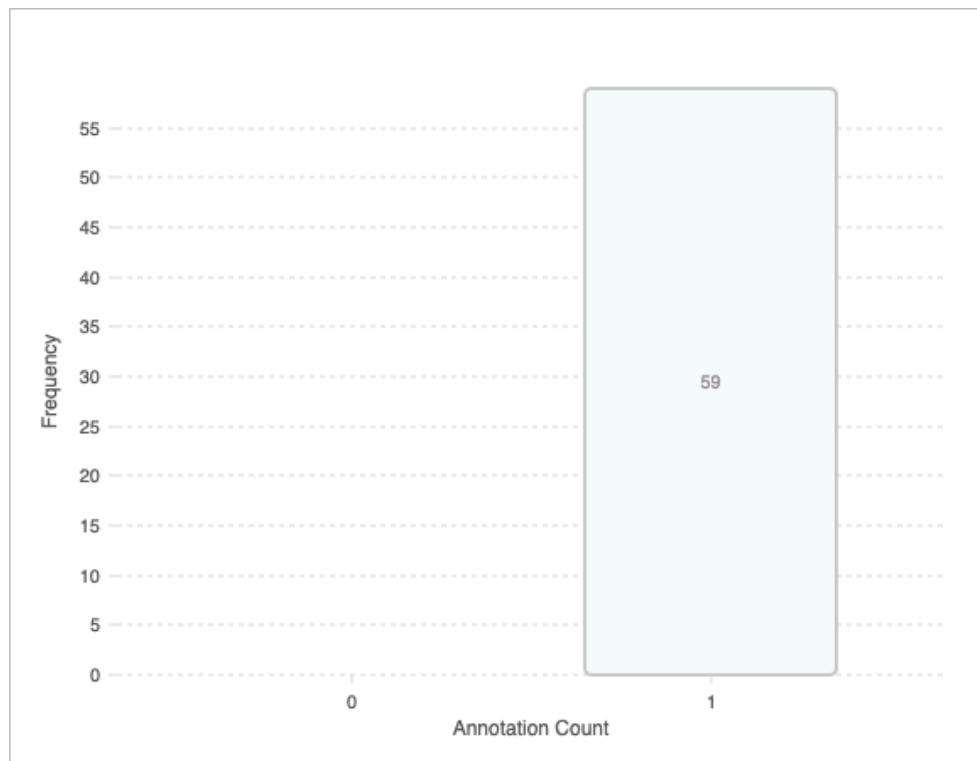
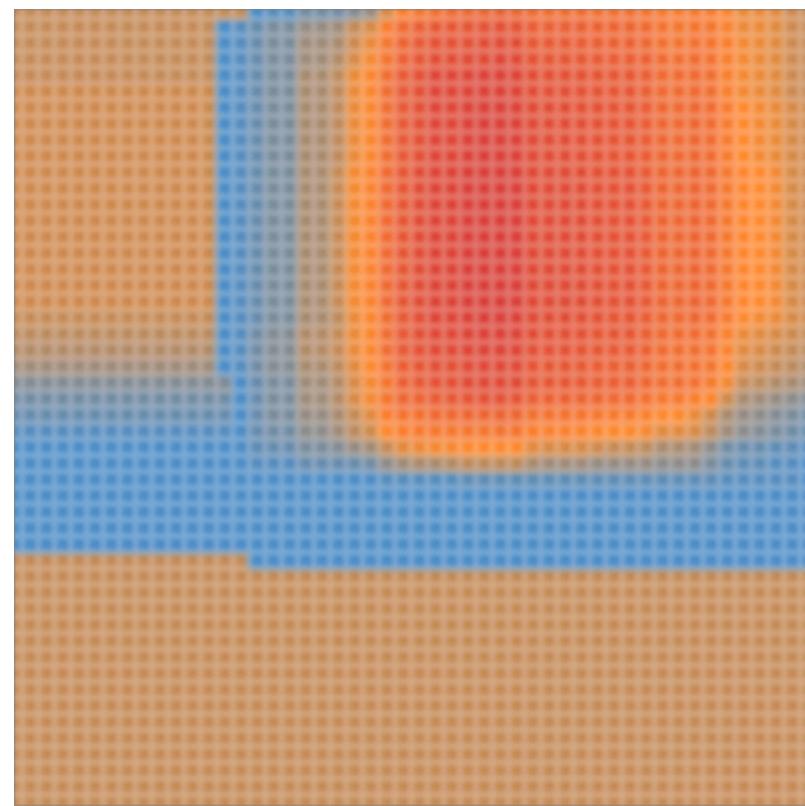


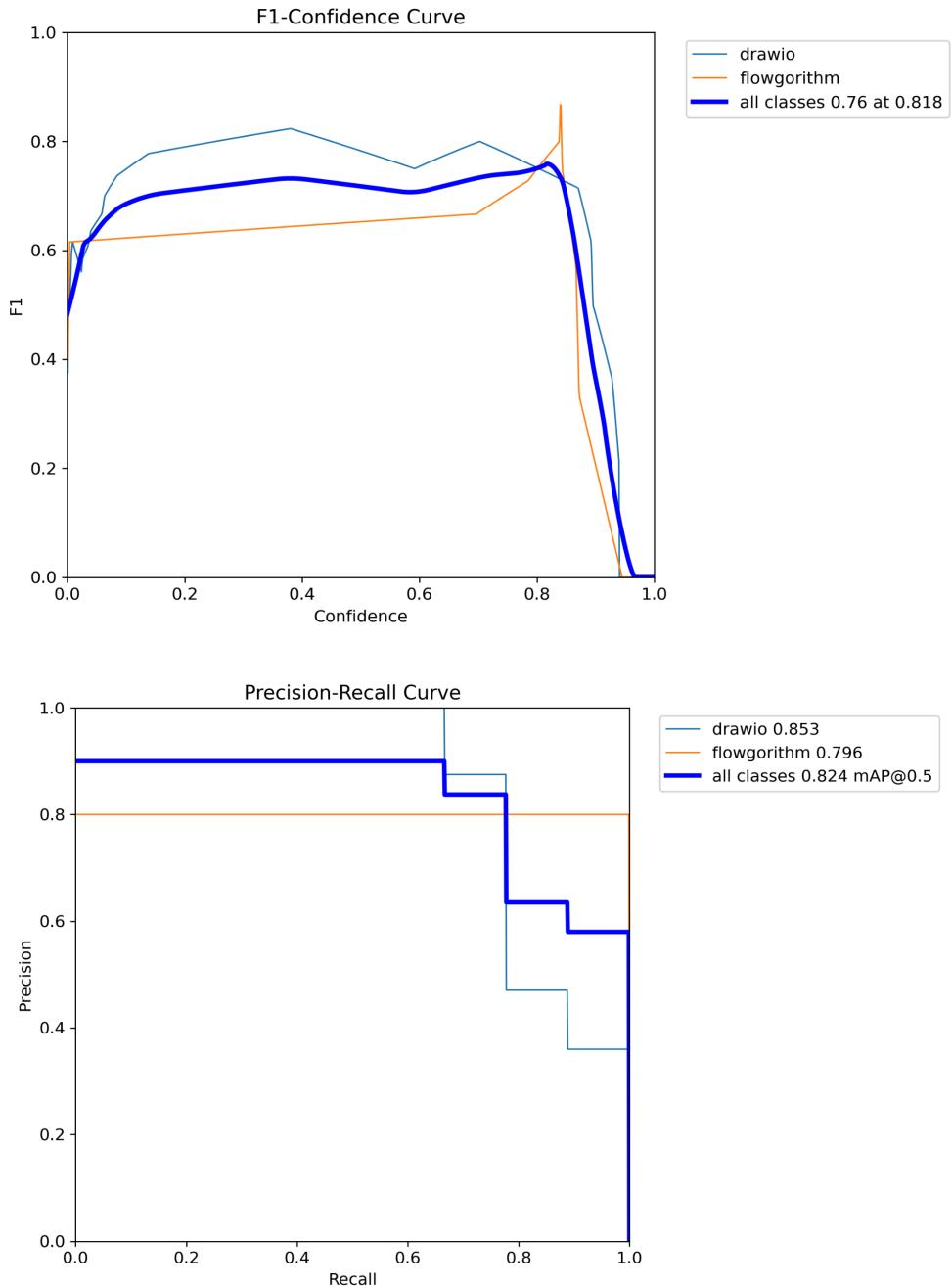
Mapa de calor de anotaciones y promedio de clases del dataset de `code-snippet-video-class-detection`

Para la figura 52, resultados muestran que el dataset presenta una distribución de anotaciones bastante concentrada en un rango intermedio (10–17), lo que indica que la mayoría de los ejemplos contienen una cantidad moderada de anotaciones, mientras que existen pocos casos con anotaciones muy bajas o muy altas (como se ve en las barras de 0–1 y 34–41). El mapa de calor, por su parte, muestra la idea de que hay una región predominante de valores intensos (en naranja) frente a áreas menos densas (en azul), esto puede significar que el conjunto de datos no está uniformemente distribuido. Ambos gráficos muestran que la clase “code_snippet” logra un desempeño muy alto, manteniendo una F1 y una relación precisión-recall más consistente a lo largo de distintos umbrales de confianza, mientras que la clase “code_bracket” presenta un F1 algo menor y una caída más pronunciada en su curva de precisión-recall, sin embargo, la combinación general (all classes) alcanza métricas sólidas (F1 de 0.92 y mAP de 0.936), mostrando un buen rendimiento global del modelo.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Figura 53



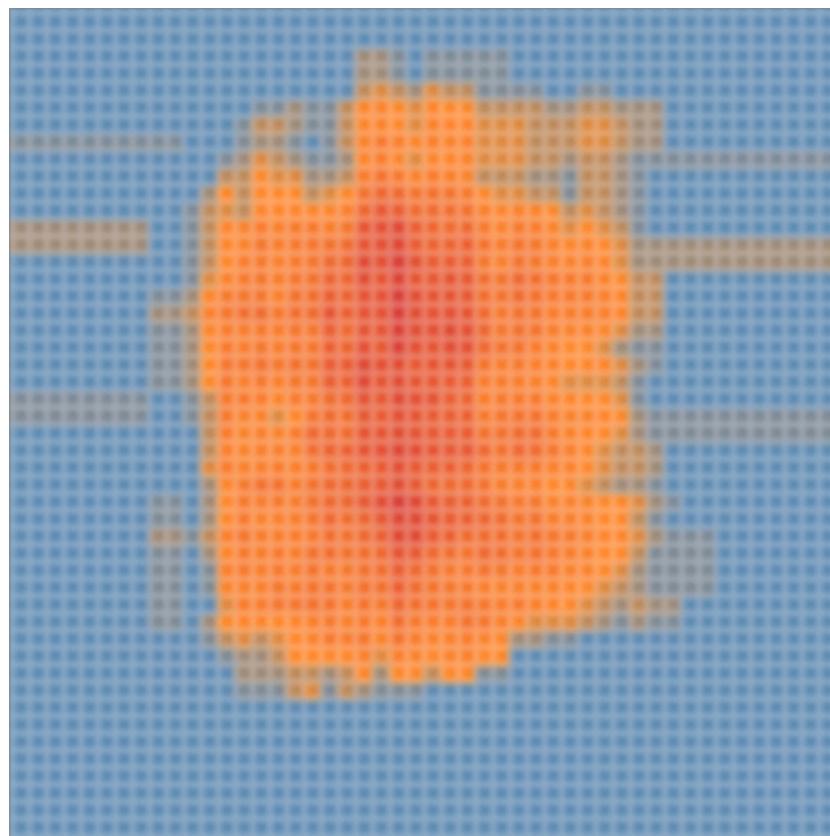


Mapa de calor de anotaciones y promedio de clases del dataset de **drawio-vs-flowgorithm**

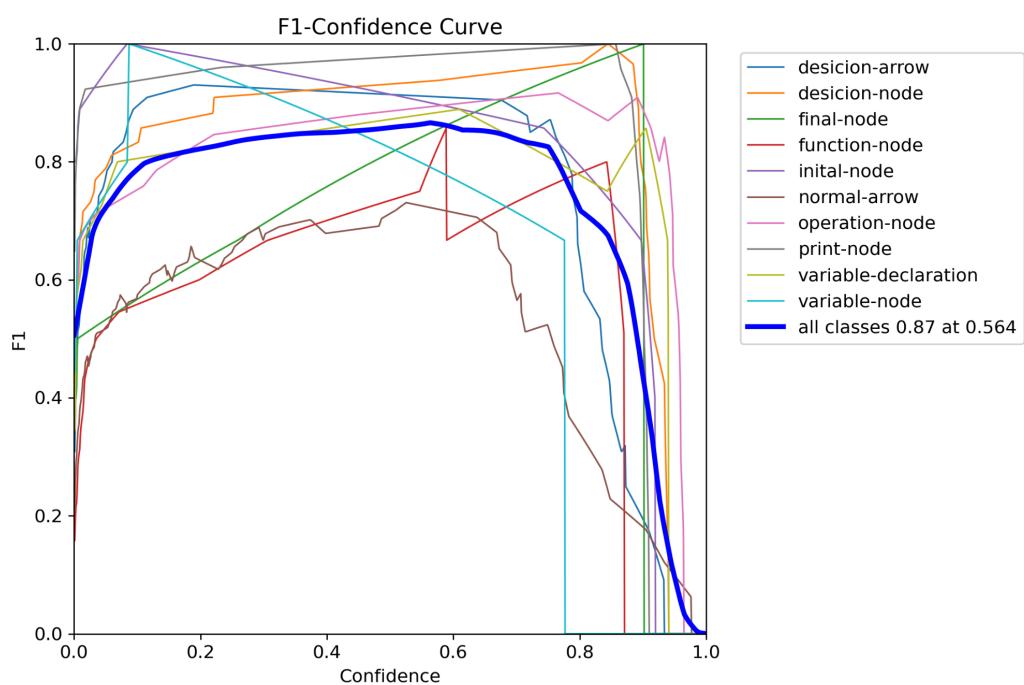
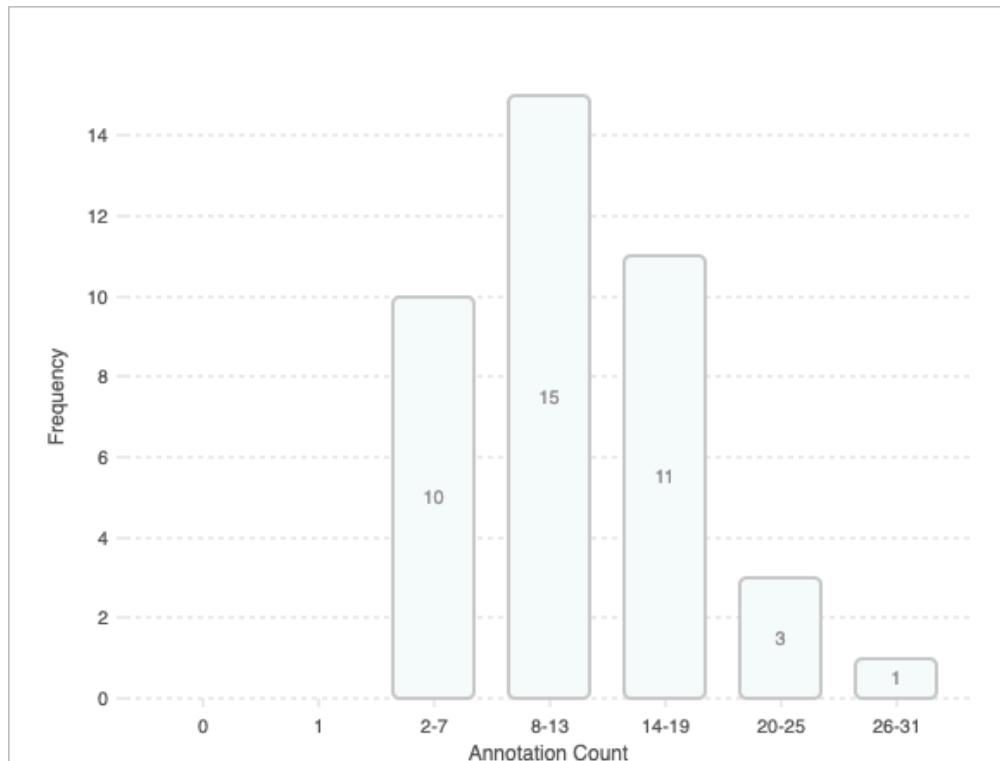
Para la figura 53, los resultados muestran que el dataset presenta una concentración marcada en el recuento de anotaciones, evidenciado por la única barra dominante en el gráfico de barras que indica que la mayoría de los elementos cuentan con exactamente una anotación; de igual forma, el mapa de valor muestra una zona de alta intensidad en valores (en tonos rojos y naranjas) contrastada por áreas con menor

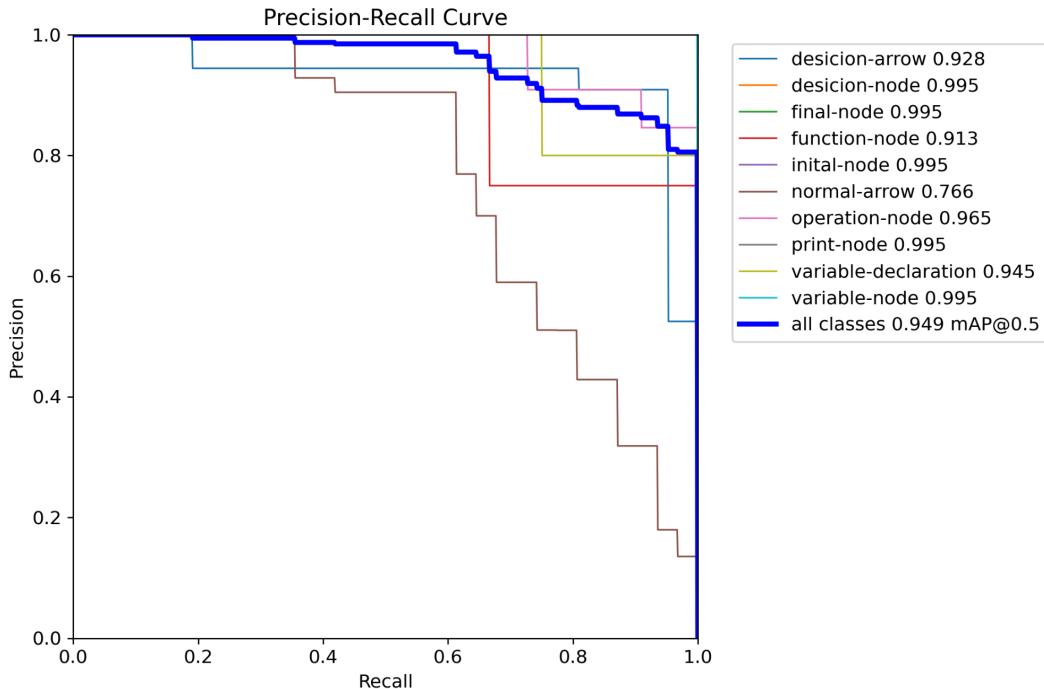
densidad (en tonos azules), esto puede significar que el conjunto de datos no está uniformemente distribuido. Los gráficos muestran que la clase “drawio” logra un F1 superior (alrededor de 0.85) y mantiene buena precisión-recall en la mayoría de los umbrales, mientras que “flowgorithm” alcanza un F1 ligeramente menor (cerca de 0.80) pero con un rendimiento más estable a través de la curva, en conjunto, todas las clases obtienen un mAP@0.5 de 0.824 y un F1 de 0.76 a un umbral de confianza de 0.818, lo que sugiere un desempeño global sólido, aunque se observa que la curva de “flowgorithm” presenta una caída más pronunciada en precisión o recall en ciertos puntos, mostrando que podría beneficiarse de un ajuste adicional en la clasificación.

Figura 54



Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación



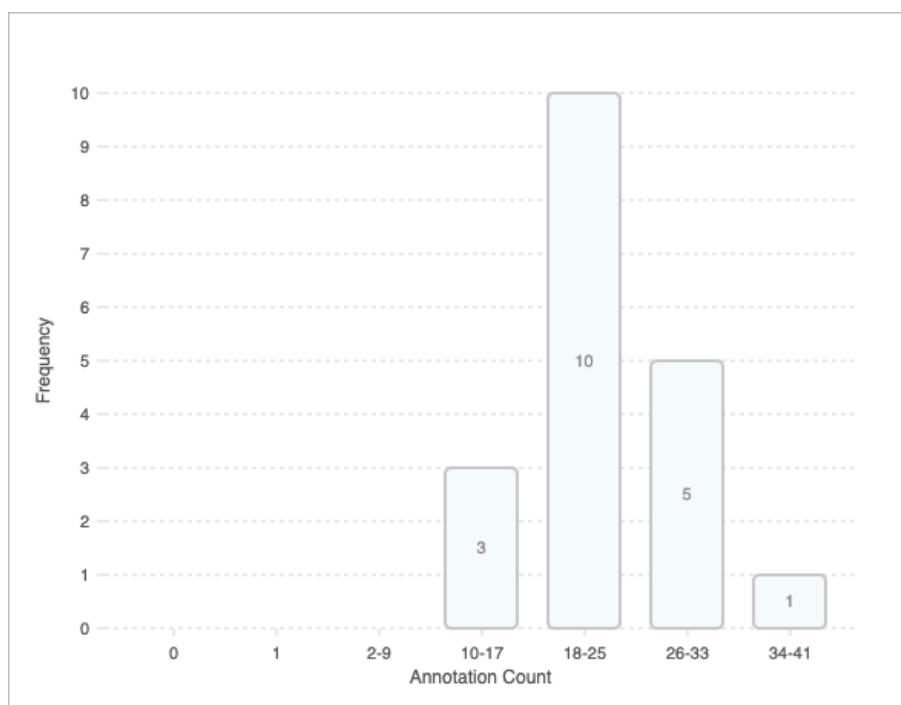


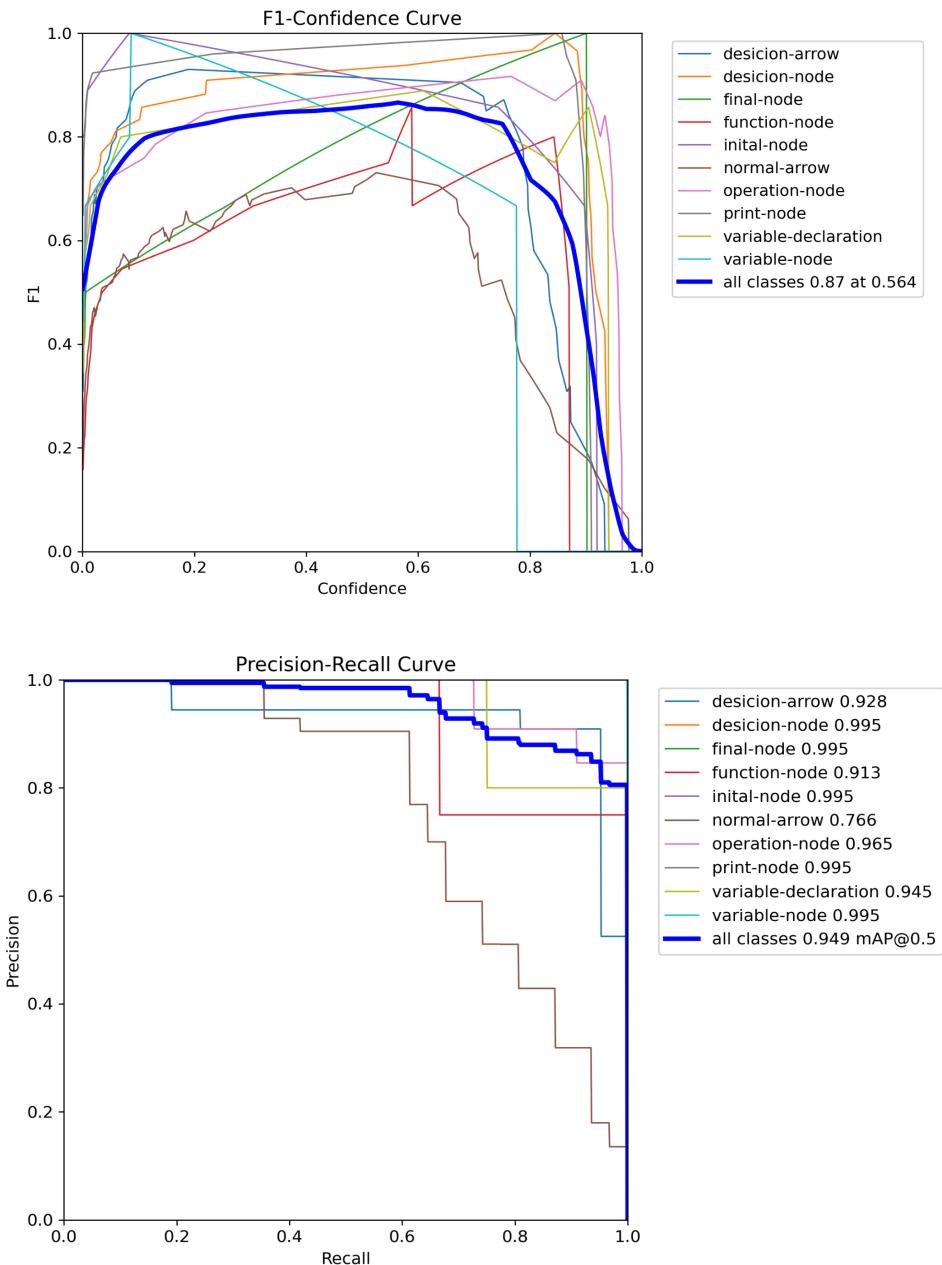
Mapa de calor de anotaciones y promedio de clases del dataset de drawio-videoclass-detection

Para la figura 54, los resultados muestran que el dataset presenta una distribución de anotaciones con una concentración mayoritaria en el rango de 8–13 (15 elementos) y 14–19 (11 elementos), mientras que se observan frecuencias menores en 2–7 (10 elementos), 20–25 (3) y 26–31 (1); este patrón indica que la mayoría de los ejemplos contienen una cantidad moderada de anotaciones, aunque se registran valores ligeramente más bajos o más altos en menor medida. El mapa de calor, por su parte, evidencia una región central con tonalidades rojas y naranjas de alta intensidad rodeada de zonas con valores inferiores (en azul), lo que sugiere que el conjunto de datos no está uniformemente distribuido y se concentra de manera notable en una zona específica. Los resultados indican que la mayoría de las clases (decision-node, final-node, initial-node, print-node, variable-node) alcanzan valores de precisión y recall casi perfectos (0.995), mientras que otras, como normal-arrow, muestran un desempeño algo menor (0.766); en conjunto, el modelo logra un F1 de 0.87 a un umbral de confianza de 0.564 y un mAP@0.5 de 0.949, lo cual sugiere un rendimiento global sólido, aunque se podrían afinar los casos donde la curva evidencia caídas más notorias para lograr un equilibrio aún mejor entre precisión y recall.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Figura 55





Mapa de calor de anotaciones y promedio de clases del dataset de *flowgorithm-videoclass-detection*

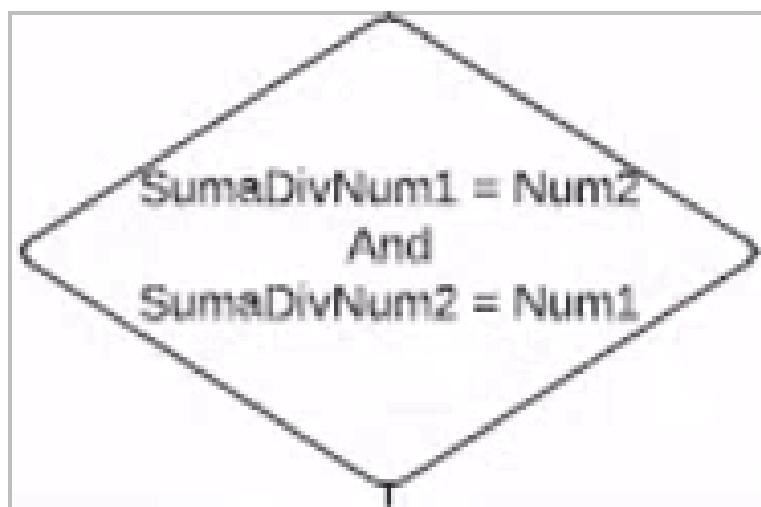
Para la figura 55, los resultados muestran que el dataset presenta una mayor concentración de anotaciones en el rango de 18–25 (10 elementos) y, en menor medida, en 26–33 (5) y 34–41 (1), mientras que los rangos inferiores (2–9 y 10–17) exhiben frecuencias todavía más bajas (3 cada uno); esto indica que la mayoría de los ejemplos se agrupan en un intervalo de anotaciones intermedio-alto. El mapa de calor, por su parte, evidencia una zona destacada en tonalidades naranja y roja rodeada de áreas en azul, lo cual refuerza la idea de que el conjunto de datos está desbalanceado,

con una concentración notable de valores en un área específica y una menor representación en los extremos. Los gráficos muestran que la mayoría de las clases (decision-node, final-node, initial-node, print-node, variable-node, etc.) alcanzan valores de F1, precisión y recall muy altos (alrededor de 0.995) a lo largo de diferentes umbrales de confianza, lo cual indica un excelente desempeño en la identificación de esas categorías. Sin embargo, clases como “normal-arrow” y “function-node” exhiben curvas con caídas más pronunciadas o F1 algo más bajo (0.766 y 0.913 respectivamente), lo que sugiere mayor dificultad para el modelo en dichas categorías. Aun así, el rendimiento global, con un F1 promedio de 0.87 al 0.564 de confianza y un mAP@0.5 de 0.949, evidencia un modelo bastante sólido, pero que podría mejorar en aquellas clases con menor desempeño al ajustar umbrales o refinar el entrenamiento.

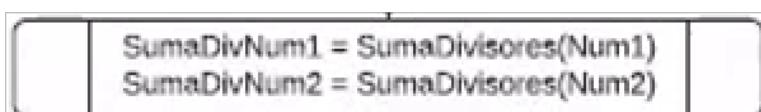
Niveles de precisión de los OCR

Previamente fue mencionado la elección de Surya OCR por los resultados obtenidos en las pruebas, aquí se muestran ejemplos de complejidad media para realizar pruebas de los escenarios reales que pueden ser encontrados dentro de las clases que serán evaluadas en el día a día. En la tabla 5 se encuentran las diferentes métricas.

Texto 1



Texto 2



Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Texto 3

```
valorpi+=4*(pow(-1,n)*(1.00/((2*n)+1)));
```

Texto 4

```
printf("%.0f[%.0f/%.2f]\n",val,sumDiv,val,div);
```

Tabla 14

Diferencias de precisión entre los OCR probados.

OCR	Texto 1	Texto 2	Texto 3	Texto 4	Total
Tesseract	100%	0%	68.2%	21.4%	47%
Easy	7.4%	38.5%	24.4%	19.6%	22%
Paddle	0%	0%	0%	0%	0%
Surya	100%	100%	100%	50%	88%

Interpretación y explicación de los datos usando LLMs

Evaluar y analizar el rendimiento de la generación por parte de los LLMs puede resultar en una tarea complicada, ya que las métricas de evaluación entran en que tan bien queda explicado el contenido, esto debe ser evaluado con el ojo humano y ser consciente del contexto que se le otorga por parte de las predicciones. Las pruebas resultantes otorgan una explicación a nivel de línea de código y al trabajar cada línea individual se puede generar una explicación de la predicción completa para finalmente extraer a todas las predicciones, pudiendo entender completamente el código. Esto se demuestra en la figura 24 donde se muestra la recreación del código, mostrando que un LLM puede llegar a ser capaz de reestructurar un código aunque se le ingresen muchas repeticiones del código. Cabe destacar que ese resultado solo ha sido posible de obtener con el uso de la API de OpenAI con el modelo ‘o3-mini’.

Conclusiones y Recomendaciones

Para finalizar, el desarrollo de este proyecto ha logrado integrar herramientas de visión por computadora y procesamiento de lenguaje natural con el objetivo de analizar contenidos de videos académicos. También, se ha logrado realizar una interfaz web, intuitiva para el usuario donde puede procesar el video grabado en clases y visualizar el reporte creado por el Bot. Asimismo, se logró una integración exitosa con la plataforma institucional Microsoft Teams, a través de la cual el sistema puede conectarse utilizando Microsoft GraphAPI para recuperar las sesiones virtuales impartidas por los docentes. Esta integración automatiza el acceso a las clases, facilitando su análisis sin necesidad de intervención manual.

Por consiguiente, se ha logrado también segmentar los videos en escenas significativas mediante PySceneDetect, extrayendo tres frames representativos por escena, lo cual reduce drásticamente la cantidad de datos sin perder información esencial. Además, la clasificación de imágenes mediante YOLOv12, combinada con la validación de similitud a través de DINOv2, ha permitido identificar de forma precisa elementos como códigos y diagramas de flujo. La extracción de texto con Surya OCR y la posterior interpretación con modelos de lenguaje (LLMs) han transformado datos visuales en información estructurada y comprensible.

Sin embargo, el análisis del proyecto también ha revelado limitaciones y resultados inesperados. Por ejemplo, se observó cierta dificultad en la detección de elementos aislados (como las llaves en el código), lo cual destaca la necesidad de afinar los parámetros y técnicas de preprocesamiento. Además, la variabilidad en la distribución de anotaciones de los datasets utilizados para entrenar YOLOv12 el cual indica que se requiere un mayor esfuerzo en el balanceo y enriquecimiento de datos. Estos hallazgos sugieren que, a pesar de los logros alcanzados, es recomendable continuar perfeccionando las herramientas y metodologías para abordar de forma más robusta los desafíos que se puedan presentar en un futuro.

La herramienta PySceneDetect ha demostrado ser eficaz en la segmentación de videos, permitiendo extraer únicamente los frames representativos (inicio, mitad y final) de cada escena. Este enfoque reduce significativamente la cantidad de

información a procesar sin comprometer la fidelidad del contenido visual, lo que es crucial para el análisis de videos de larga duración.

Asimismo, los modelos entrenados con YOLOv12 han permitido la detección y clasificación de códigos y diagramas, logrando altos niveles de precisión. Sin embargo, la variabilidad observada en las distribuciones de los datasets resalta la necesidad de aplicar técnicas de balanceo y enriquecimiento de datos para mejorar la uniformidad y robustez del entrenamiento.

Por otro lado, la aplicación de DINOv2 ha sido fundamental para validar y filtrar los frames extraídos, eliminando imágenes redundantes mediante la medición de similitud de características visuales. Esto garantiza que únicamente se conserven los frames con diferencias significativas, lo que optimiza el análisis posterior.

Además, el uso de Surya OCR ha permitido extraer de forma precisa el texto de los elementos detectados en los frames, facilitando la interpretación de códigos y diagramas. Se sugiere que su rendimiento se pueda potenciar aún más entrenando con un dataset especializado que contemple la diversidad y complejidad de textos en contextos académicos.

De la misma forma, la integración de LLMs ha sido clave para transformar la información visual y textual en explicaciones coherentes y estructuradas, posibilitando una reconstrucción precisa del contenido. Este enfoque representa un paso importante hacia la automatización en el análisis y comprensión de materiales educativos.

Para futuros desarrollos, es importante mejorar y balancear los datasets utilizados en el entrenamiento de YOLOv12. El análisis del mismo, muestra una concentración en rangos específicos de anotaciones, lo que muestra la necesidad de implementar técnicas de muestreo estratificado y balanceo de clases. Estas acciones no solo mejorarían la uniformidad de los datos, sino que también permitirían detectar de forma más precisa tanto los casos con pocas anotaciones como aquellos con una alta

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

concentración, reduciendo así la incidencia de falsos negativos y fortaleciendo el rendimiento del modelo.

También, se recomienda focalizar esfuerzos en optimizar el proceso de extracción de texto mediante Surya OCR. Entrenar este sistema con un dataset especializado, que incluya una amplia variedad de ejemplos reales de códigos y diagramas académicos, podría incrementar significativamente su precisión y adaptabilidad. Junto con la optimización de algoritmos de preprocesamiento y la exploración de modelos de lenguaje más avanzados, estas mejoras permitirán superar las limitaciones actuales del sistema, abriendo el camino para una aplicación más robusta y eficiente en el análisis automatizado de contenidos educativos.

Bibliografía

- [1] W. A. Pirela-Espina, “Brecha digital y calidad de la educación universitaria Latinoamérica durante el Covid-19,” *Rev. Electrónica En Educ. Pedagog.*, vol. 6, no. 11, pp. 43–57, Jul. 2022, doi: 10.15658/rev.electron.educ.pedagog22.11061104.
- [2] K. A. Aivaz and D. Teodorescu, “College Students’ Distractions from Learning Caused by Multitasking in Online vs. Face-to-Face Classes: A Case Study at a Public University in Romania,” *Int. J. Environ. Res. Public. Health*, vol. 19, no. 18, p. 11188, Sep. 2022, doi: 10.3390/ijerph191811188.
- [3] E. Pennino, C. Ishikawa, S. Ghosh Hajra, N. Singh, and K. McDonald, “Student Anxiety and Engagement with Online Instruction across Two Semesters of COVID-19 Disruptions,” *J. Microbiol. Biol. Educ.*, vol. 23, no. 1, pp. e00261-21, Apr. 2022, doi: 10.1128/jmbe.00261-21.
- [4] CCCSE, “The Online Student: Impact of Course Modality on Engagement,” 2023, CCCSE. doi: 10.26153/TSW/48699.
- [5] S. Avramovic and I. Avramovic, “Exploring the Potential Benefits and Limitations of Using an AI Text- Generation Tool in Education: An Examination of ChatGPT’s Performance on Assessments”, [Online]. Available: <https://www.ingentaconnect.com/contentone/aupha/jhae/2024/00000040/00000002/art00004#>
- [6] S. Lytvynova, N. Rashevska, and S. Proskura, “The use of artificial intelligence in teaching students programming languages”.
- [7] N. Eun and J. Lee, “DiagramVoice: Automatic Lecture Video Commentator for Visually Impaired Students Supporting Diagram Commentary,” in *Proceedings of Ninth International Congress on Information and Communication Technology*, vol. 1013, X.-S. Yang, S. Sherratt, N. Dey, and A. Joshi, Eds., in *Lecture Notes in Networks and Systems*, vol. 1013. , Singapore: Springer Nature Singapore, 2024, pp. 381–391. doi: 10.1007/978-981-97-3559-4_31.
- [8] S. Bhushan and M. Lee, “Block Diagram-to-Text: Understanding Block Diagram Images by Generating Natural Language Descriptors,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2022*, Online only: Association for Computational Linguistics, 2022, pp. 153–168. doi: 10.18653/v1/2022.findings-acl.15.
- [9] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi, “A Diagram Is Worth A Dozen Images,” 2016, *arXiv*. doi: 10.48550/ARXIV.1603.07396.
- [10] E. F. Llanga-Vargas, M. E. Guacho-Tixi, C. E. Andrade-Cuadrado, and M. R. Guacho-Tixi, “Dificultades de aprendizaje en modalidad virtual,” *Polo Conoc.*, vol. 6, no. 8, Art. no. 8, Aug. 2021, doi: 10.23857/pc.v6i8.2979.
- [11] MACHUCA VIVAR, Silvio Amable; SAMPEDRO GUAMAN, Carlos Roberto; PALMA RIVERA, Diego Paúl y CANIZARES GALARZA, Fredy Pablo., “Desarrollo de la lógica de programación en estudiantes de sistemas de Uniandes Santo Domingo.,” vol. vol.17, pp. 214–224, Apr. 2021.
- [12] J. J. Arellano Pimentel and S. P. Canedo Ibarra, “EpAA: Entorno para el Aprendizaje de Algoritmos. Una experiencia educativa desde la perspectiva del aprendizaje flexible.,” *Edutec Rev. Electrónica Tecnol. Educ.*, no. 79, pp. 63–79,

Mar. 2022, doi: 10.21556/edutec.2022.79.2451.

- [13] Ávila Clavijo, Jhovanny Gustavo , Pazmiño Piedra, Juan Pablo, Erazo Álvarez, Juan Carlos, “Aprendizaje basado en problemas y técnicas de programación en la especialidad de electrónica de consumo,” vol. vol.7, pp. 800–819, Oct. 2022.
- [14] R. Francisco and F. Silva, “Intelligent Tutoring System for Computer Science Education and the Use of Artificial Intelligence: A Literature Review;,” in *Proceedings of the 14th International Conference on Computer Supported Education*, Online Streaming, --- Select a Country ---: SCITEPRESS - Science and Technology Publications, 2022, pp. 338–345. doi: 10.5220/0011084400003182.
- [15] J. Yu and M. D. Srinath, “An efficient method for scene cut detection,” *Pattern Recognit. Lett.*, vol. 22, no. 13, pp. 1379–1391, Nov. 2001, doi: 10.1016/S0167-8655(01)00085-X.
- [16] Google API, *Video Intelligence de Google Cloud*. [Online]. Available: https://cloud.google.com/video-intelligence/docs/text-detection#video_detect_text-drest
- [17] “PySceneDetect”, [Online]. Available: <https://www.scenedetect.com/api/>
- [18] “¿Qué es el reconocimiento óptico de caracteres (OCR)?” [Online]. Available: <https://aws.amazon.com/es/what-is/ocr/>
- [19] Tesseract, *Tesseract OCR*. [Online]. Available: <https://github.com/tesseract-ocr/tesseract?tab=readme-ov-file#about>
- [20] W. Burger and M. J. Burge, *Digital Image Processing: An Algorithmic Introduction*. in Texts in Computer Science. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-031-05744-1.
- [21] Alexander Gelbukh, “Procesamiento de Lenguaje Natural y sus Aplicaciones”, [Online]. Available: https://d1wqxts1xzle7.cloudfront.net/30768432/Procesamiento_de_lenguaje_natural_y_sus_aplicaciones-libre.pdf?1392139645=&response-content-disposition=inline%3B+filename%3DProcesamiento_de_lenguaje_natural_y_sus.pdf&Expires=1739306508&Signature=IIuw34s5jGEfmt-VWzafB5lPqYy8W7qsDikZbf38yxPh4t1qvE50TGXIQDwAeNy2JSyf93Eu4CxqsRgSWiGgRAO1CavIu9b5KXPnu2CMN8pEd2uboSmjB27uRX~wK17bnIPGp39u1iKWrRncRn8yeWYMjqGKsdMhtgdm3puYN9KyHU0vqDvL6jyb2DPhRI-f1ghu6qgN2arvOjsejAhISEn03Hfsf~GxhfYHmbW7o7~UBNmG3EzRWPch~iX4s-tcdSULROhTGuOVHLRkqB4cdqtBx4YHb09LQwqXauFcf2NhJPIyNhDXDxI0NeRDZK0V7cSzpUEBT9dpjoVdNZhQ_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA
- [22] X. Zheng *et al.*, “Graph Neural Networks for Graphs with Heterophily: A Survey,” 2022, *arXiv*. doi: 10.48550/ARXIV.2202.07082.
- [23] Y. Hou, B. Gileadereli, Y. Tu, and M. Sachan, “Do Vision-Language Models Really Understand Visual Language?,” 2024, *arXiv*. doi: 10.48550/ARXIV.2410.00193.
- [24] P. Bagga and K. Stathis, “Towards Explainable Strategy Templates using NLP Transformers,” 2023, doi: 10.48550/ARXIV.2311.14061.
- [25] C. Raffel *et al.*, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” 2019, *arXiv*. doi: 10.48550/ARXIV.1910.10683.
- [26] M. Lewis *et al.*, “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” in *Proceedings*

- of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, 2020, pp. 7871–7880. doi: 10.18653/v1/2020.acl-main.703.
- [27] H. Liu and T. Brailsford, “Reproducing ‘Show, Attend and Tell: Neural Image Caption Generation with Visual Attention,’” *J. Phys. Conf. Ser.*, vol. 2589, no. 1, p. 012012, Sep. 2023, doi: 10.1088/1742-6596/2589/1/012012.
- [28] Textractify, “Textractify: Revolutionizing Text Extraction from Videos,” Textractify: Revolutionizing Text Extraction from Videos. [Online]. Available: <https://textractify.com/blog/transcription-vs-extraction>
- [29] Textractify, “Textractify,” Textractify. [Online]. Available: <https://textractify.com/>
- [30] S. Ghafforov, “Blackbox AI: Cracking the Code on Conversational Innovation,” Blackbox AI: Cracking the Code on Conversational Innovation. [Online]. Available: <https://medium.com/@sharifghafforov00/blackbox-ai-cracking-the-code-on-conversational-innovation-6c4949b0c415>
- [31] Z. Latif, “Blackbox.ai : Code with Confidence,” x.ai : Code with Confidence. [Online]. Available: <https://www.techwrix.com/blackbox-ai-code-with-confidence/>
- [32] A. Rosebrock, “OpenCV shape detection,” OpenCV shape detection. [Online]. Available: <https://pyimagesearch.com/2016/02/08/opencv-shape-detection/>
- [33] OpenCV, “OpenCV,” OpenCV. [Online]. Available: <https://opencv.org/about/>
- [34] Pixelcode AI, “Pixelcode AI - Extract code from video tutorials,” Pixelcode AI - Extract code from video tutorials. [Online]. Available: <https://chromewebstore.google.com/detail/pixelcode-ai-extract-code/cojidefaaghkfamaebmfhccfcajbiopd>
- [35] “¿Qué es la inteligencia artificial o IA?,” Google Cloud. Accessed: Mar. 28, 2025. [Online]. Available: <https://cloud.google.com/learn/what-is-artificial-intelligence>
- [36] “Visión por ordenador vs procesamiento de imágenes: Las diferencias clave.” Accessed: Mar. 28, 2025. [Online]. Available: <https://www.ultralytics.com/es/blog/computer-vision-vs-image-processing-the-key-differences>
- [37] “¿Qué es el procesamiento de lenguaje natural? - Explicación del procesamiento de lenguaje natural - AWS,” Amazon Web Services, Inc. Accessed: Mar. 28, 2025. [Online]. Available: <https://aws.amazon.com/es/what-is/nlp/>
- [38] “OCR con Google AI,” Google Cloud. Accessed: Mar. 28, 2025. [Online]. Available: <https://cloud.google.com/use-cases/ocr>
- [39] aws, “¿Qué es una cola de mensajería?,” ¿Qué es una cola de mensajes? [Online]. Available: <https://aws.amazon.com/es/message-queue>
- [40] “Flower Celery Monitoring.” [Online]. Available: https://appliku-com.translate.goog/post/celery-flower/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc
- [41] “CPU o GPU: ¿cuál es la diferencia?,” Intel. Accessed: Mar. 28, 2025. [Online]. Available: <https://www.intel.com/content/www/xl/es/products/docs/processors/cpu-vs-gpu.html>

- [42] “¿En qué consiste la infraestructura en la nube?: Explicación sobre la infraestructura de computación en la nube: AWS,” Amazon Web Services, Inc. Accessed: Mar. 28, 2025. [Online]. Available: <https://aws.amazon.com/es/what-is/cloud-infrastructure/>
- [43] “¿Qué es Python? - Explicación del lenguaje Python - AWS,” Amazon Web Services, Inc. Accessed: Mar. 28, 2025. [Online]. Available: <https://aws.amazon.com/es/what-is/python/>
- [44] “Surya OCR Implementation,” Lightning AI. Accessed: Mar. 28, 2025. [Online]. Available: <https://lightning.ai>
- [45] “Dinov2 Image Similarity Search | Restackio.” Accessed: Mar. 28, 2025. [Online]. Available: <https://www.restack.io/p/similarity-search-answer-dinov2-image-similarity-cat-ai>
- [46] Ultralytics, “YOLO12.” Accessed: Mar. 28, 2025. [Online]. Available: <https://docs.ultralytics.com/es/models/yolo12>
- [47] “¿Cómo funciona el modelo cliente-servidor?,” IONOS Digital Guide. Accessed: Mar. 28, 2025. [Online]. Available: <https://www.ionos.com/es-us/digitalguide/servidores/know-how/modelo-cliente-servidor/>
- [48] “HTTP y HTTPS: diferencia entre los protocolos de transferencia. AWS,” Amazon Web Services, Inc. Accessed: Mar. 28, 2025. [Online]. Available: <https://aws.amazon.com/es/compare/the-difference-between-https-and-http/>
- [49] “¿Qué es una API de REST?” Accessed: Mar. 28, 2025. [Online]. Available: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- [50] “GDPR: Lo que debes saber sobre el reglamento general de protección de datos.” Accessed: Mar. 28, 2025. [Online]. Available: <https://www.powerdata.es/gdpr-proteccion-datos>
- [51] M. Roncero-Bazarría, “Guías BibUpo: Conceptos clave sobre edición multimedia: el vídeo digital: ¿Qué es un frame?” Accessed: Mar. 28, 2025. [Online]. Available: https://guiasbib.upo.es/edicion_multimedia_video/frame
- [52] “Servicios de cloud computing,” Google Cloud. Accessed: Mar. 28, 2025. [Online]. Available: <https://cloud.google.com/>
- [53] “Azure Cognitive Services: What they are, pricing, and examples.” Accessed: Mar. 28, 2025. [Online]. Available: <https://www.dev4side.com/en/blog/azure-cognitive-services>
- [54] “¿Qué es la API de OpenAI?” [Online]. Available: <https://www.datacamp.com/es/tutorial/guide-to-openai-api-on-tutorial-best-practices>

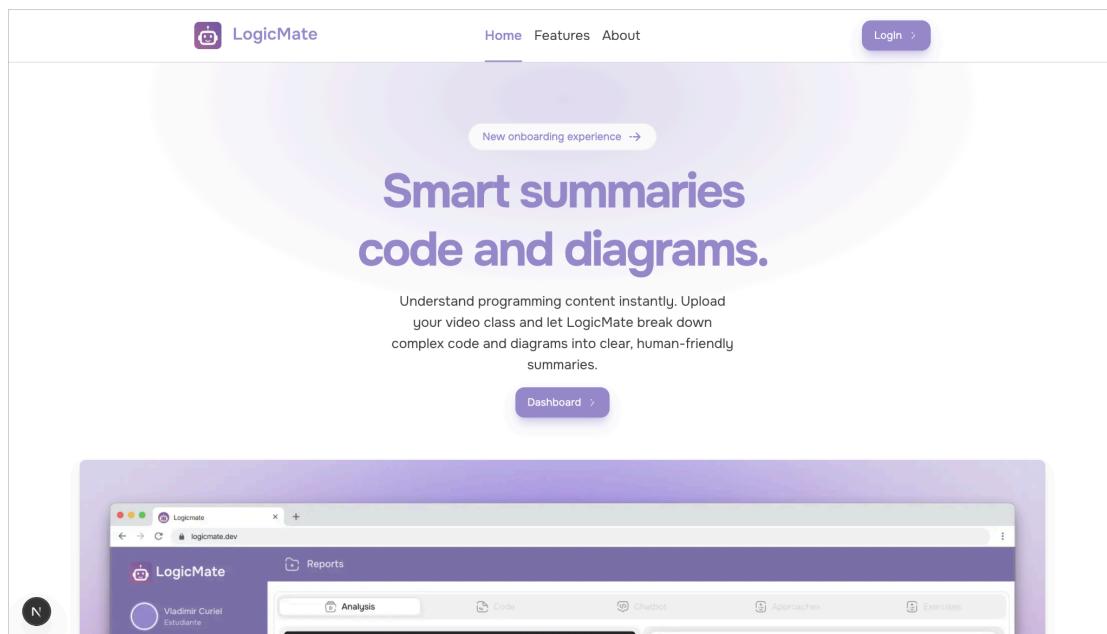
Anexos

Anexo A

Interfaz Web

En esta sección se presenta la interfaz web desarrollada para presentar el Bot de forma intuitiva al usuario. Esta representa el punto de interacción directa entre el usuario y la aplicación. El diseño está enfocado en ofrecer una experiencia intuitiva, accesible y visualmente coherente con los objetivos del proyecto al estudiante y docente que lo utilicen.

Figura 38

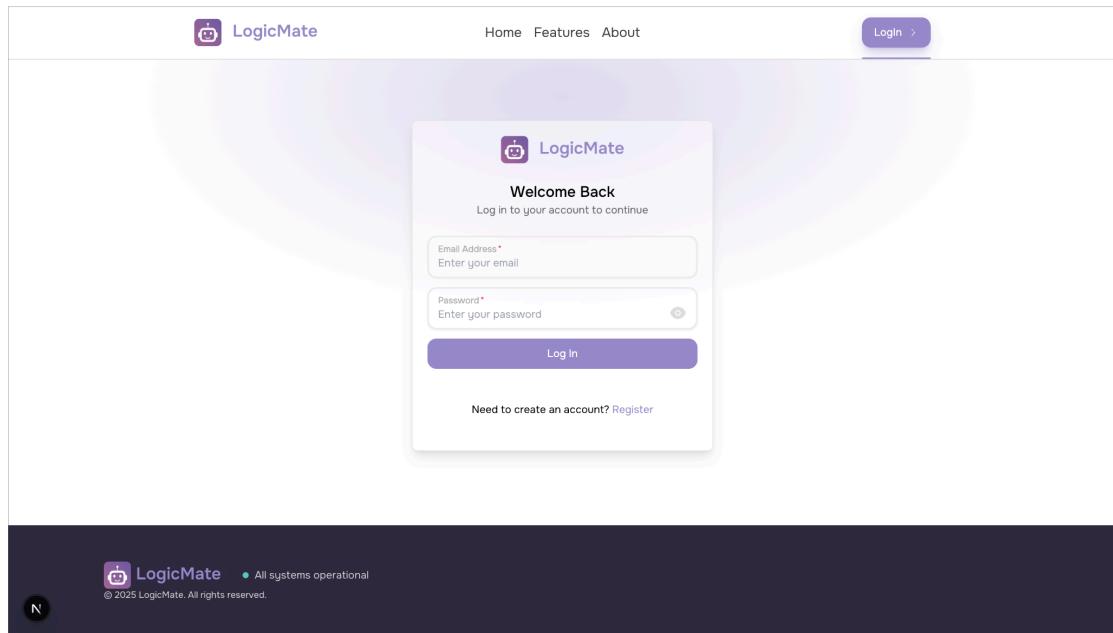


Página de inicio de la interfaz web

Esta es la pantalla principal al ingresar a la plataforma, donde el usuario puede ver una pequeña introducción de qué tratará el Bot llamado *Logic Mate* el Bot desarrollado, incluyendo este una vista previa de su funcionamiento. También, en esta visual el usuario puede iniciar sesión por medio del botón de Login. Además, al hacer clic en el botón Dashboard, este será redirigido a la pantalla de inicio de sesión, donde deberá ingresar sus credenciales para acceder y utilizar las funcionalidades del bot.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Figura 39

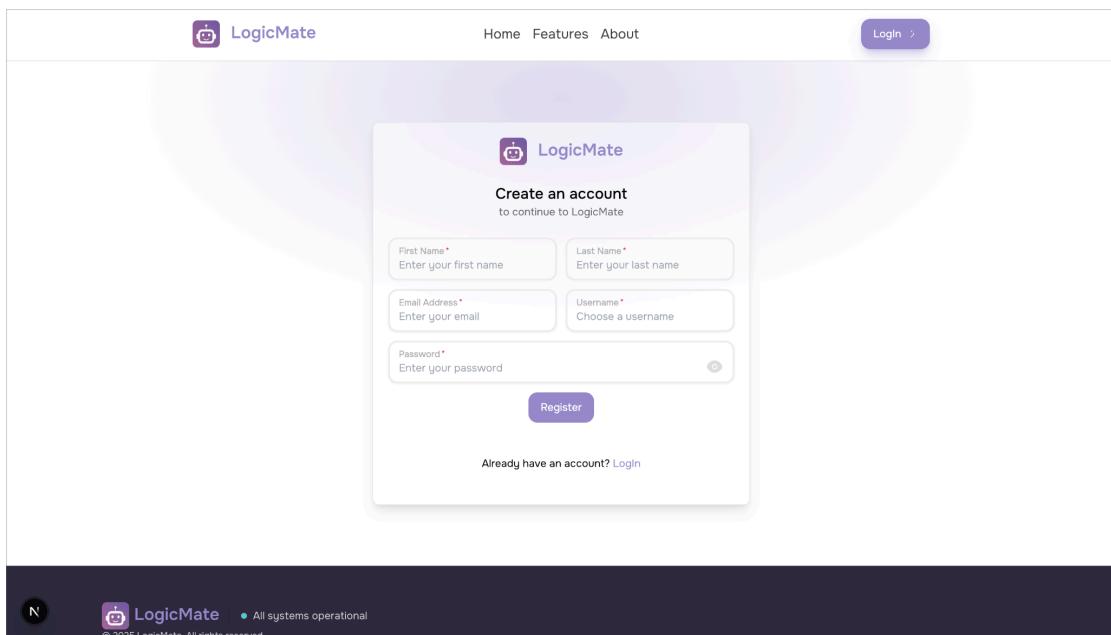


Página de Inicio de Sesión (Login)

Aquí el usuario debe de ingresar sus credenciales (correo y contraseña) para acceder a las funcionalidades del Bot. En caso de que no esté registrado, la interfaz le da la oportunidad de poder registrarse y luego poder iniciar sesión.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Figura 40

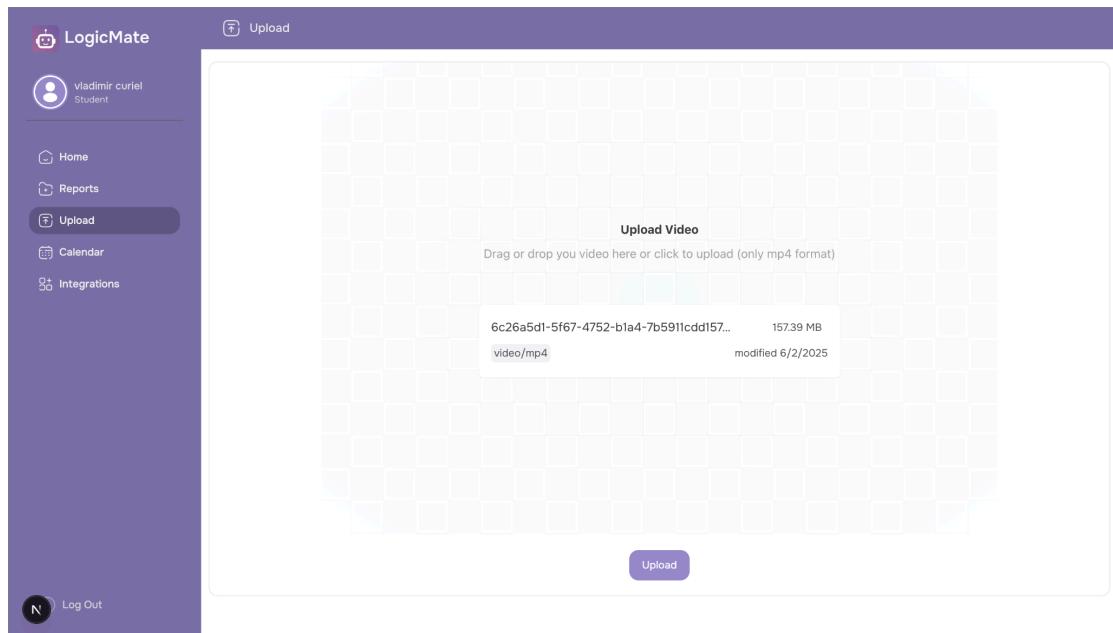


Página de Registro

Aquí, el usuario puede completar sus datos para crear una cuenta. Si el usuario no tiene sus credenciales, no podrá hacer uso del Bot y sus funcionalidades que ofrece. Se debe utilizar el mismo correo institucional para que las clases realizadas en Teams luego de ser procesadas salgan de forma automática.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

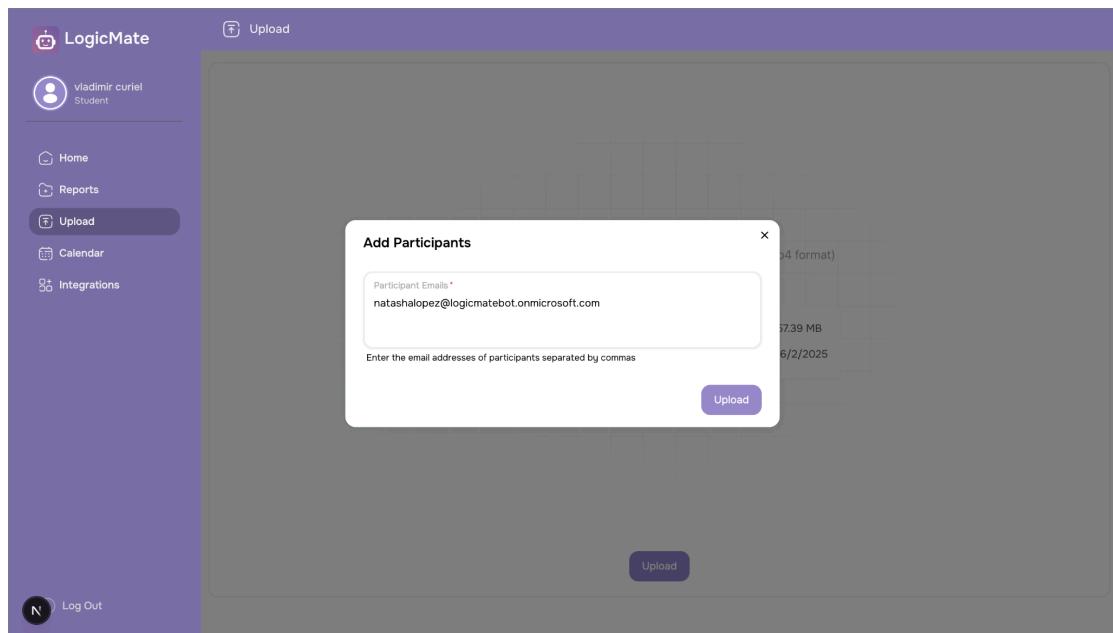
Figura 41



Página de subir video

En esta sección, el usuario puede seleccionar y cargar un video.

Figura 42

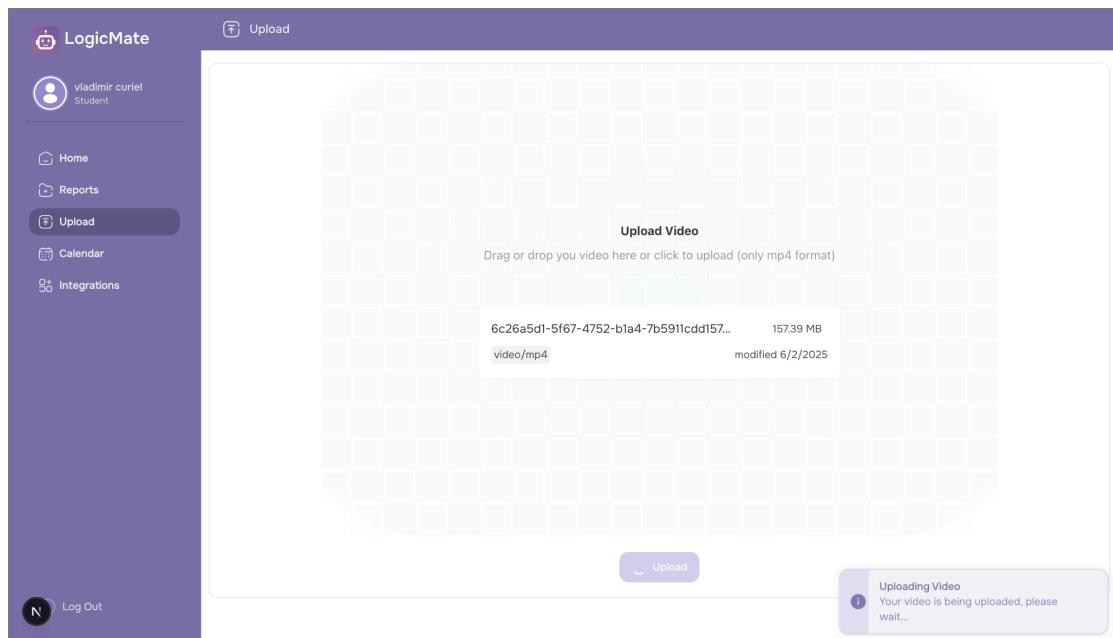


Página para compartir el procesamiento a mas participantes

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Esta visual permite que más participantes puedan acceder al reporte del video y puedan hacer uso del mismo para sus estudios.

Figura 43



Página donde se visualiza como se va procesando el video

El usuario puede visualizar como el Bot va procesando el video para así crear el reporte.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Figura 44

The screenshot shows the LogicMate application interface. On the left is a sidebar with a purple header containing the LogicMate logo and the user's name, vladimir curiel Student. Below the header are five menu items: Home, Reports (which is highlighted in blue), Upload, Calendar, and Integrations. At the bottom of the sidebar is a Log Out button. The main content area has a light gray header with the text "Reports" and a "View All Reports" button. Below this is a card titled "Aproximación de π con la Serie de Leibniz". The card features a thumbnail image of a computer screen displaying a terminal window with code and output. To the right of the thumbnail is a summary text: "El video presenta una serie de fragmentos de código en C, ilustrando el proceso de desarrollo, depuración y corrección de programas diseñados para calcular el valor de la constante matemática π (pi) utilizando diferentes métodos, como la serie de Leibniz...". At the bottom of the card is a "View Report" button.

Página de Reportes listados

Esta vista muestra todos los reportes que tiene el usuario. Cada reporte puede ser seleccionado para visualizarlo detenidamente.

Figura 45

The screenshot shows the LogicMate application interface. The sidebar is identical to Figure 44. The main content area has a light gray header with tabs for Analysis, Code, Flow Diagram, Chatbots AI, Approaches AI, and Exercises AI. The Analysis tab is selected. Below the header is a Microsoft Teams video player window showing a video titled "CSTI-1860-4159 - Introducción a la Algoritmia". The video player includes a timestamp (2024-05-24 14:05 UTC), names of participants (Alejandro José Liz Reynoso, Alejandro José Liz Reynoso, General), and a duration (40:38). Below the video player are two code snippets. The first snippet, under the "Code" tab, is titled "Inclusión de cabeceras y definición de constantes" and contains the following C code:

```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <conio.h>
4
5 #define MAXCOL 4
6 #define MAXFIL 20
7 #define ENTER 13
```

The second snippet, under the "Code" tab, is titled "Declaración anticipada de la función pi" and contains the following C code:

```
00:12:00 - 00:13:00
Se declara la función 'pi' antes del 'main' para que el compilador conozca su firma. Esto permite llamarla desde el 'main' aunque su implementación esté más abajo.
```

Página de visualización del reporte creado

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Aquí se puede visualizar el reporte ya creado. El usuario puede ir viendo la sesión junto al reporte que el Bot realizó. También, puede visionar cada corte de escena donde se está explicando el código u/o diagrama de flujo visto en clases. Puede seguir la explicación del docente junto al reporte creado.

Figura 46

The screenshot shows the LogicMate platform interface. On the left, there's a sidebar with a user profile for 'Vladimir curiel Student' and navigation links for Home, Reports, Upload, Calendar, and Integrations. The main area is titled 'Dashboard' and contains tabs for Analysis, Code, Flow Diagram, Chatbot AI, Approaches AI, and Exercises AI. The 'Code' tab is active, displaying a code editor with the file 'main.c'. The code is as follows:

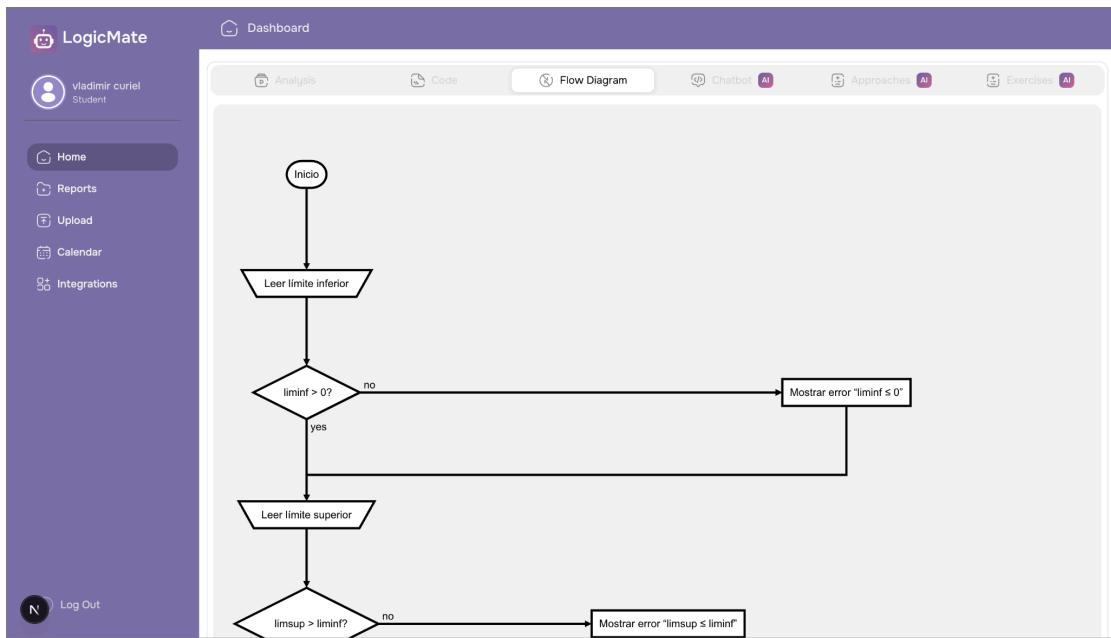
```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <conio.h>
4
5 #define MAXCOL 4
6 #define MAXFIL 20
7 #define ENTER 13
8
9 float pi(int);
10
11 int main() {
12     int liminf, limsup, cantterm;
13     int cantcolimp, cantfilimp;
14     char tecla;
15
16     do {
17         printf("Límite Inferior: ");
18         scanf("%d", &liminf);
19         if (liminf <= 0)
20             printf("Límite inferior debe ser mayor que cero.\n");
21     } while (liminf <= 0);
22
23     do {
24         printf("Límite Superior: ");
25         scanf("%d", &limsup);
26         if (limsup <= 0)
27             printf("Límite superior debe ser mayor que cero.\n");
```

Página donde se visualiza el código visto en clase

La interfaz también permite que el usuario pueda ver el código generado por el docente en clases y tenerlo a mano para cualquier ocasión donde lo quiera usar para repaso junto al reporte creado por el Bot, o poder probarlo y confirmar con el docente.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

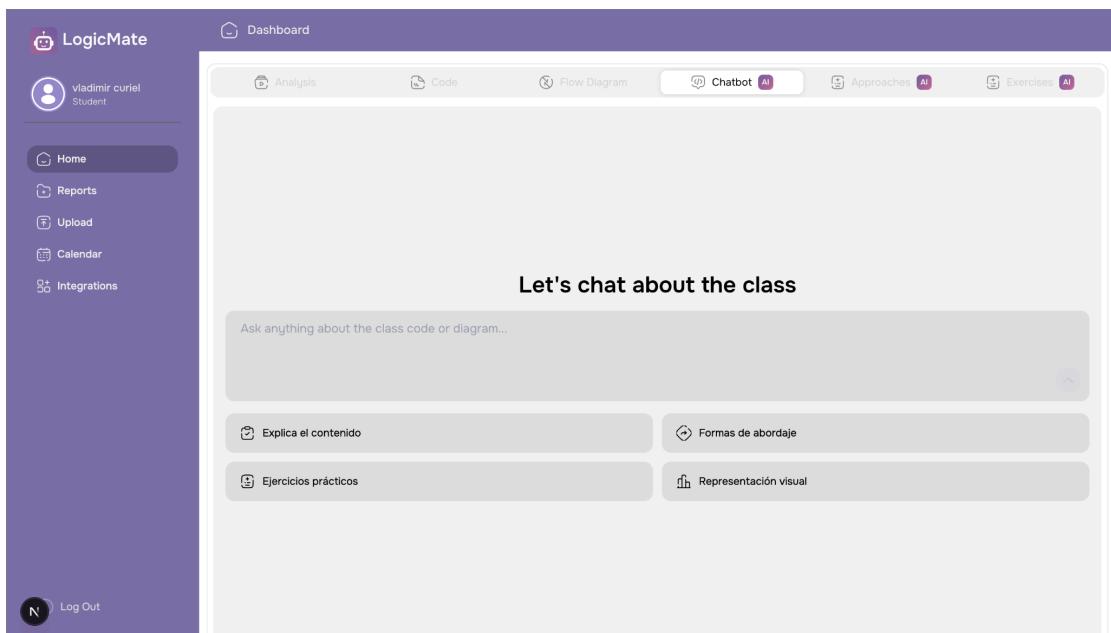
Figura 47



Página donde se presenta el diagrama visto en clases

De forma similar al código, el usuario puede ver el diagrama de flujo empleado en clases.

Figura 48



Página para charlar con el Bot directamente

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Se creó un espacio, donde el usuario puede hacerle preguntas sobre la clase procesada al Bot directamente. Este puede explicar el contenido visto en clases, crear ejercicios sobre la misma, hacer una representación visual para mayor entendimiento de lo explicado, y demás.

Figura 49

The screenshot shows the LogicMate platform interface. On the left, there's a sidebar with a user profile for 'Vladimir curiel Student', navigation links for Home, Reports, Upload, Calendar, Integrations, and a Log Out button. The main area is titled 'Dashboard' and contains several cards: 'Serie de Nilakantha', 'Método de Monte Carlo', 'Integral definida de 0 a 1 de 4 / (1 + x²)', 'Producto de Wallis', and 'Serie de Gregory alterna optimizada'. Below these cards, a text box states: 'Una versión optimizada de la serie de Leibniz que elimina variables auxiliares innecesarias y usa el índice para el cálculo.' A code editor window displays the following C code:

```
approach-5-original.c
1 float pi(int nterm) {
2     float sum = 0;
3     int den = 1;
4     for (int i = 1; i <= nterm; i++, den += 2) {
5         if (i % 2 == 1)
6             sum += 1.0 / den;
7         else
8             sum -= 1.0 / den;
9     }
10    return 4 * sum;
11 }
```

Página de enfoques visto en la clase

El usuario puede revisar detenidamente los enfoques vistos en la clase, donde puede ver una opción optimizada o explicada detalladamente, para mayor compresión.

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Figura 50

The screenshot shows the LogicMate platform's dashboard. On the left sidebar, there are links for Home, Reports, Upload, Calendar, and Integrations. The main area is titled 'Dashboard' and contains several cards:

- Cálculo del área de una corona circular
- Volumen de una esfera
- Longitud de una circunferencia
- Área de un sector circular
 - Escribe una función en C que calcule el área de un sector circular dados el radio y el ángulo en grados. Usa una función para calcular π .
 - Show Solution
 - Press to expand
- Velocidad angular de un objeto en rotación

Página de ejercicios propuestos para practicar lo visto en clases

Se visualiza un apartado para obtener ejercicios generados automáticamente por el bot junto a sus soluciones. Estos basados en el contenido de la clase procesada.

Figura 51

The screenshot shows the LogicMate platform's calendar. On the left sidebar, there are links for Home, Reports, Upload, Calendar, and Integrations. The main area is titled 'Calendar' and shows a monthly grid for August 2025. An event is highlighted on Tuesday, August 15, with a tooltip providing details:

Bases de Datos
Fundamentos de P...
Introducción a la A...

Introducción a la Algoritmia
This event starts at 03:00 PM and ends at 03:30 PM
Organized by: Vladimir Curiel
Location: LogicMate

Página del calendario

Bot de Asistencia Interactivo en Algoritmos y Diagramas de Programación

Por último, en esta vista se puede ver detalladamente las reuniones de la materia en particular, Introducción a la Algoritmia, donde el usuario puede tener para un día en específico clases o reuniones de estudio con el docente. Este calendario está conectado con la plataforma Teams.