

UNIVERSIDAD LA SALLE

LICENCIATURA EN INGENIERÍA DE SISTEMAS



“TÉCNICA DE REBANADO DE VIDEO”

MODALIDAD DE TITULACIÓN: PROYECTO DE GRADO

POSTULANTE: ROSALIA REYNAGA FUNES

TUTOR: ING. JUAN PABLO VON LANDWÜST GUMICIO

La Paz - Bolivia

2026

DEDICATORIA

Este Trabajo está dedicado a:

Todas las mujeres que
en algún momento les
dijeron que no podían
hacer ciencia.

AGRADECIMIENTOS

Al culminar mis estudios de pregrado con este trabajo, me siento orgullosa de cada uno de los pequeños pasos que me trajeron hasta aquí. Este camino no fue sencillo; mi desempeño académico se vio afectado por enfermedades y situaciones personales que hicieron de esta etapa un proceso mucho más difícil de lo esperado. Sin embargo, hoy puedo finalmente cerrar este capítulo con gratitud y orgullo.

Quiero reconocer también a quienes atraviesan dificultades similares. A veces el cuerpo no responde y la mente juega malas pasadas, pero aun así es posible seguir adelante, incluso avanzando poco a poco. Nada de esto habría sido posible sin el apoyo médico adecuado y sin las personas que me acompañaron en los momentos más difíciles.

Agradezco profundamente a mi familia por su cariño y apoyo incondicional.

A mi padre, por enseñarme desde pequeña la belleza de la ciencia y despertar en mí la curiosidad por comprender el mundo; a mi madre, por su apoyo incondicional y a mi hermana, María Renee, por acompañarme siempre, hacerme reír incluso a la distancia y recordarme que nunca estuve sola.

También guardo un agradecimiento especial para Bocky (+) y Mily, por cada pelito de felicidad que me han dado y gracias a ello hacer mi vida más llevadera.

A mis abuelos, que no pudieron verme graduarme pero cuya presencia vive siempre conmigo; y a mis tíos que partieron antes de que yo naciera, porque ellos saben cuánto los amo.

Gracias a C. Tangana por acompañarme con su música durante muchas noches de desvelo, y a Alexby11 y Radio Lely, que hicieron mi vida más bonita, y me ayudaron a sobrevivir a la pandemia a través de la pantalla.

Y finalmente, a Rosalía Reynaga Funes, porque sin su fortaleza y su convicción, nada de esto habría sido posible. Te amo.

ÍNDICE

ÍNDICE DE CONTENIDO

CAPITULO 1. GENERALIDADES.....	VIII
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES.....	2
1.2.1. Antecedentes Académicos	2
1.2.2. Antecedentes Institucionales	3
1.3. PLANTEAMIENTO DEL PROBLEMA.....	4
1.3.1. Identificación del Problema	5
1.3.2. Formulación del Problema	6
1.4. OBJETIVOS	6
1.4.1. Objetivo General.....	6
1.4.2. Objetivos Específicos.....	6
1.5. JUSTIFICACIÓN	7
1.5.1. Justificación Teórica	7
1.5.2. Justificación Social	7
1.5.3. Justificación Económica	8
1.6. ALCANCES Y LIMITACIONES	8
1.6.1. Alcances	8
1.6.2. Limites	9
CAPITULO 2. MARCO TEÓRICO	11
2.1. TÉCNICA	11
2.2. TÉCNICA DE REBANADO	11
2.3. GEOMETRÍA.....	11
2.4. MATEMÁTICA.....	12
2.4.1. Matemática Cuantitativa	13
2.4.2. Fundamentos algebraicos	17
2.5. ALGORITMO DE LÍNEA DE BRESENHAM.....	21
2.6. SEÑALES DE TRANSITO	22
2.6.1. Señales horizontales.....	23
2.6.2. Señales verticales.....	24
2.6.3. Flechas	26
2.7. ANÁLISIS DE DATOS	27
2.7.1. Sensores	29

2.7.2.	Datos	30
2.7.3.	Información y Conocimiento.....	30
2.7.4.	Variables y sus Dominios.....	31
2.7.5.	Datos Espacio Temporales	32
2.7.6.	Visualización	32
2.8.	VISIÓN POR COMPUTADOR.....	32
2.8.1.	Procesamiento de Imágenes	33
2.8.2.	Óptica	33
2.8.3.	Transformada de Radon	34
2.8.4.	Paradigma de Visión de David Marr	35
2.9.	HERRAMIENTAS DE SOFTWARE	37
2.9.1.	Python	38
2.9.2.	Open CV.....	38
2.9.3.	PIL (Python Image Library)	40
2.9.4.	Numpy.....	40
2.9.5.	Matplotlib	41
2.9.6.	Jupyter Notebook.....	41
2.9.7.	Tkinter	41
2.9.8.	CustomTkinter	42
2.9.9.	Streamlit	42
2.9.10.	Matlab	43
2.10.	MARCO DE TRABAJO.....	44
2.10.1.	Scrum.....	44
CAPITULO 3.	MARCO APLICATIVO.....	46
3.1.	PLANIFICACIÓN.....	47
3.2.	EXPLORATORIA.....	48
3.2.1.	Análisis de los procesos actuales de supervisión vial en ECOVIANA S.R.L	49
3.2.2.	Análisis del Algoritmo de Perspectiva.....	49
3.2.3.	Revisión de la funcionalidad del algoritmo	53
3.3.	MODELO MATEMÁTICO	54
3.3.1.	Geometría Euclidiana.....	55
3.3.2.	Matemática Cuantitativa	55
3.3.3.	Matemática Cualitativa	55
3.4.	MODELO COMPUTACIONAL	62

3.5.	PROTOTIPO	64
3.5.1.	Sprint Planning	65
3.5.2.	Producto backlog	66
3.5.3.	Sprint 1	66
3.5.4.	Sprint 2	69
3.5.5.	Sprint 3	74
3.5.6.	Sprint 4	77
3.5.7.	Sprint 5	84
3.6.	EVALUACIÓN TÉCNICA	90
3.7.	EVALUACIÓN ECONÓMICA	91
CAPITULO 4. CONCLUSIONES Y RECOMENDACIONES		93
4.1.	CONCLUSIONES	92
4.2.	RECOMENDACIONES.....	92
BIBLIOGRAFÍA		94
GLOSARIO	94	
ANEXOS	99	

ÍNDICE DE TABLAS

Tabla 1	Tabla de planificación	48
Tabla 2	Descripción del equipo scrum	65
Tabla 3	Historias de usuario.....	65
Tabla 4	Product Backlog del proyecto	66
Tabla 5	Sprint backlog del sprint 1	67
Tabla 6	Actualización de la tabla sprint backlog.....	68
Tabla 7	Product Backlog del proyecto al terminar el sprint 1	69
Tabla 8	Sprint backlog del sprint 1	70
Tabla 9	Actualización del Sprint backlog del sprint 2.....	73
Tabla 10	Product Backlog del proyecto al terminar el sprint 2	74
Tabla 11	Sprint backlog del sprint 3	75
Tabla 12	Actualización del Sprint backlog del sprint 3.....	76
Tabla 13	Product Backlog del proyecto al terminar el sprint 3	77
Tabla 14	Sprint backlog del sprint 4	78
Tabla 15	Actualización del Sprint backlog del sprint 4.....	83
Tabla 16	Product Backlog del proyecto al terminar el sprint 4	84
Tabla 17	Sprint backlog del sprint 4	84
Tabla 18:	Comparativa entre prototipos.....	88
Tabla 19	Actualización del Sprint backlog del sprint 4.....	89
Tabla 20	Product Backlog del proyecto al terminar el sprint 4	89
Tabla 21	Tabla descriptiva de gastos en internet.....	91
Tabla 22	Contabilización de trabajo dedicado en horas	91
Tabla 23	Costo por hora de trabajo en bolivianos.....	92
Tabla 24	Costo de las herramientas	92

ÍNDICE DE FIGURAS

Figura 1 Descripción grafica de los postulados de Euclides	13
Figura 2 Segmento y flecha según la geometría euclidiana.	20
Figura 3 Conjunto de flechas denominado F	20
Figura 4 Una imagen geométrica de la suma de vectores.	20
Figura 5 Línea de Bresenham.....	22
Figura 6 Ejemplo de señales de tránsito en Bolivia	23
Figura 7 Líneas segmentadas y continuas en video analizado.....	24
Figura 8 Línea segmentada en el manual oficial de la ABC	24
Figura 9 Línea continua en el manual de la ABC.	24
Figura 10: Señal vertical vial lateral derecha en el video analizado	25
Figura 11: Señal vertical vial lateral derecha en el manual de la ABC.....	25
Figura 12 Señal informativa lateral con flechas en el video analizado	26
Figura 13 Señal informativa lateral con flechas en el manual.....	26
Figura 14 Flechas rectas y de cruce a la derecha	26
Figura 15 Flecha con salida a la derecha en el manual del ABC.....	27
Figura 16 Flecha recta en el manual de la ABC.....	27
Figura 17 Descripción de un flujo común de la visión por computadora	33
Figura 18 Descripción grafica de los niveles propuestos por Marr partiendo de una imagen digital.....	36
Figura 19 Ejemplo de nivel bajo.....	36
Figura 20 Ejemplo de nivel medio	37
Figura 21 Ejemplo de nivel superior	37
Figura 22 Esquema presentado por Ken Schwaber	45
Figura 23 Resultado del código	50
Figura 24 Código para cambiar la perspectiva de un solo frame.....	51
Figura 25 Código para cambiar la perspectiva de todos los frames de un video y almacenar el resultado	52
Figura 26 Código para concatenar n frames de un video.....	53
Figura 27 Resultado final de la concatenación.....	54
Figura 28 Modelo matemático del espacio	56
Figura 29 Descripción del modelo dentro del contexto del escenario vial	58
Figura 30 Descripción de la modelación del movimiento	58
Figura 31 Pseudocódigo del modelo computacional	64
Figura 32 Resultado del modelo computacional	67
Figura 33 Mockup del prototipo.....	71
Figura 34 Diseño del prototipo	71
Figura 35 Pantalla del prototipo en la parte del procesamiento	72
Figura 36 Pantalla del prototipo mostrando la imagen resultante.....	73
Figura 37 Replanteamiento del algoritmo base	76
Figura 38 Diagrama de actividad UML	79
Figura 39 Pantalla de inicio	80
Figura 40 Pantalla principal con video ya cargado	80
Figura 41 Captura de pantalla después de pulsar la tecla “Enter”	81

Figura 42 Imagen sin edición.....	82
Figura 43 Imagen final con filtros y correcciones aplicadas.	82
Figura 44: Diagrama UML de secuencia	86
Figura 45 Wireframe del rediseño.....	87
Figura 46: Captura de pantalla del prototipo	88

ÍNDICE DE ECUACIONES

Ecuación 1: Función sucesor	13
Ecuación 2: Definición de operaciones binarias.....	14
Ecuación 3: Conjunto de los números naturales extendido con el cero	14
Ecuación 4: Proposición S_n	14
Ecuación 5: Principio de inducción matemática.....	15
Ecuación 6: Relación de equivalencia	15
Ecuación 7: Clase de equivalencia.....	16
Ecuación 8: Relación de equivalencia	16
Ecuación 9: Clases de equivalencia	16
Ecuación 10: Conjunto de los números $\mathbb{R}$	17
Ecuación 11. Transformada de Radon.....	34
Ecuación 12. Sistema de r y s	35
Ecuación 13 Modelo del sensor.....	59
Ecuación 14. Dominio del mapeo de transiciones.	60

RESUMEN

Este trabajo de titulación para la Licenciatura en Ingeniería de Sistemas presenta una herramienta de procesamiento y análisis de contenido multimedia denominado "rebanado de video". Consiste en el tratamiento de la información audiovisual como un espacio vectorial tridimensional, lo que permite la construcción de un volumen de datos sólido. Mediante la parametrización de este volumen, se generan cortes transversales (rebanadas) que sintetizan la información visual de una carretera en una imagen bidimensional. Esta técnica optimiza significativamente los tiempos de auditoría y revisión de metraje extenso. Debido a la versatilidad de la herramienta, que permite realizar cortes en diversos ángulos de los ejes espaciales y temporales, se propone su implementación como etapa de pre-procesamiento en sistemas avanzados de visión artificial, específicamente para el reconocimiento de señalética vial.

ABSTRACT

This final project for the Systems Engineering degree program presents a tool for processing and analyzing multimedia content called “video slicing.” It involves treating audiovisual information as a three-dimensional vector space, which enables the construction of a solid data volume. By parameterizing this volume, cross-sections (slices) are generated that synthesize the visual information of a road into a two-dimensional image. This technique significantly optimizes the time required for auditing and reviewing extensive footage. Due to the tool’s versatility—which allows for cross-sections at various angles along the spatial and temporal axes—its implementation is proposed as a pre-processing stage in advanced computer vision systems, specifically for the recognition of road signs.

CAPITULO 1.

GENERALIDADES

1.1. INTRODUCCIÓN

La infraestructura de carreteras juega un papel crucial en el avance socioeconómico de las naciones, al ser el pilar que permite el movimiento eficiente de individuos y bienes, impulsando sectores vitales como el turismo y el comercio. Por estos motivos se debe procurar que el estado de las carreteras sea óptimo para el tránsito, esto incluye las señales de tránsito.

Con el objetivo de elevar el estándar de la red vial, Bolivia se ha unido a entidades internacionales como el Banco de Desarrollo de América Latina y el Banco Interamericano de Desarrollo. Estas colaboraciones han sido fundamentales para emprender iniciativas como el levantamiento rápido de datos de 11.000 kilómetros de la Red Vial Fundamental del Estado Plurinacional de Bolivia, integrándose a una estrategia de mantenimiento vial proyectada a dos décadas.

En este contexto, se ha identificado la necesidad de localizar zonas de alto riesgo y segmentos propensos desgastes, lo cual ha motivado la adopción de tecnologías avanzadas para el análisis de información derivada de grabaciones en carreteras. Dicho avance tecnológico no solo promete realzar la señalización y conservación de las rutas, sino que también busca promover la gestión eficaz y sostenible de los recursos asignados a la infraestructura vial.

Este proyecto surge como respuesta a dicha demanda, proponiendo el desarrollo de una técnica de rebanado de video que en base a un video de la carretera, genere imágenes en 2 dimensiones, recuperando información importante para el análisis del estado de las carreteras, proyectos que la empresa ECOVIANA S.R.L suele gestionar.

Finalmente, se espera que la técnica de rebanado de video se consolide como una herramienta de apoyo en el análisis de grabaciones, permitiendo a la empresa mejorar la calidad de sus informes técnicos, optimizar la distribución de los recursos humanos y reducir el tiempo dedicado a cada auditoría vial. De este modo, será posible incrementar la capacidad operativa, desarrollar un mayor número de proyectos y obtener beneficios económicos sostenibles. En el largo plazo, las operaciones de la empresa contribuirán al progreso económico y social de Bolivia.

1.2.ANTECEDENTES

A continuación, se revisan algunos antecedentes académicos relacionados al presente trabajo y los antecedentes institucionales de la empresa ECOVIANA S.R.L.

1.2.1.Antecedentes Académicos

A continuación, se mencionarán trabajos similares al presente proyecto:

“DESARROLLO DE UN ALGORITMO DE STITCHING PARA SECUENCIAS DE IMÁGENES CON AMPLIOS MOVIMIENTOS DE CÁMARA” memoria presentada por Sebastián Andrés Gálvez Ortiz, en el año 2017, para obtener el título de ingeniero civil eléctrico en la Universidad De Chile de Santiago de Chile, Chile, cuyo objetivo general fue: “Desarrollar un algoritmo de stitching para sintetizar la información desplegada en secuencias de imágenes con importantes traslaciones de la cámara, tomando en consideración los intereses de la empresa Woodtech S.A. de aplicarlo sobre las imágenes de camiones con cargamento de madera y, a la vez, explorar su uso potencial para la reconstrucción tridimensional del camión.” (Gálvez Ortiz, 2017)

El algoritmo presentando por Gálvez se basa en la técnica stitching la cual consiste en juntar varias imágenes secuenciales según características para mejorar la visualización de la información, A diferencia de este proyecto en el cual se planea utilizar la técnica de slicing (técnica de rebanado) para obtener información de frames de un video.

“APRENDIZAJE FIN A FIN PARA LA CONDUCCIÓN AUTÓNOMA DE VEHÍCULOS DOMÉSTICOS USANDO VISIÓN ARTIFICIAL Y REDES NEURONALES CONVOLUCIONALES” proyecto de grado presentado por José Eduardo Laruta Espejo, en el año 2018, para la carrera de ingeniería electrónica, en la Universidad Mayor de San Andrés de La Paz, Bolivia, cuyo objetivo general fue: “Diseñar un sistema de aprendizaje “fin a fin” capaz de generar de comandos de control de vehículos domésticos basado en visión artificial y redes neuronales convolucionales para el diseño e implementación de sistemas de conducción autónoma.” (Laruta Espejo, 2018)

El proyecto previamente mencionado, utilizara redes neuronales para entrenar el sistema de conducción autónoma el cual detecta las líneas de la carretera. En contraste con el presente trabajo que no utilizara redes neuronales para el procesamiento de identificación de

elementos de la carretera, ni utilizara los datos recolectados en la detección para un control autónomo, más bien se enfoca en mostrar los datos de la detección de forma simple para que puedan ser comprendidos por los usuarios finales.

“DETECCIÓN Y CLASIFICACIÓN DE GRIETAS EN ASFALTO MEDIANTE REDES CONVOLUCIONALES Y PROCESAMIENTO DE IMÁGENES” memoria presentada por Claudio Isaías Urbina Lara, en el año 2020, para obtener el título de ingeniero civil eléctrico en la Universidad De Chile de Santiago de Chile, Chile, cuyo objetivo general fue: “Desarrollar un algoritmo que detecte, clasifique y analice la severidad de grietas presentes en imágenes de asfalto capturadas por el equipo LCMS en carreteras chilenas.” (Urbina Lara, 2020).

La diferencia con el actual proyecto es el uso de redes neuronales pix2pix y procesamiento de imágenes para detectar las grietas en el asfalto, en el caso del presente proyecto se utilizará un video como dato de entrada, no imágenes, y se hará énfasis en las señales de tráfico verticales y horizontales, no en las grietas del asfalto como se plantea en el documento mencionado anteriormente

1.2.2. Antecedentes Institucionales

ECOVIANA S.R.L. es una sociedad de responsabilidad limitada con sede principal en la ciudad de La Paz – Bolivia, la empresa fue fundada el 17 de mayo de 1975, ha incursionado en proyectos a lo largo de todo el país e incluso ha realizado trabajos con instituciones en el exterior. Entre sus clientes se encuentran diversas empresas e instituciones, destacando la Administradora Boliviana de Carreteras (ABC), el Ministerio de Medio Ambiente y Aguas (MMAyA) y el Gobierno Autónomo Municipal de La Paz. (Ecoviana S.R.L., 2024)

ECOVIANA S.R.L. cuenta con diversas certificaciones, entre las más importantes, certificación ISO 9001 de gestión de Calidad y certificación ISO 14001 de Gestión Medio Ambiental otorgada por TÜV Rheinland, organización reconocida internacionalmente, a nivel nacional cuentan con el certificado de Registro Nacional de Consultoría Ambiental (R.E.N.C.A) que les otorga permisos para realzar trabajos de consultoría ambiental conforme a la reglamentación de ley de Medio Ambiente de Bolivia. Todas estas certificaciones respaldan la calidad de los proyectos de la empresa y su prestigio a nivel nacional e internacional. (Ecoviana S.R.L., 2024)

1.2.2.1. Visión

La empresa ECOVIANA S.R.L. tiene como visión “ser una empresa de ingeniería de clase mundial que compite globalmente, reconocida por su prestigio y garantía de cumplimiento en la ejecución de servicios de consultoría, con liderazgo nacional y presencia a nivel internacional” (Ecoviana S.R.L., 2024).

1.2.2.2. Misión

Como misión la empresa busca “brindar servicios de consultoría de diseño, supervisión, gestión, y administración de obras de infraestructura, a través del desarrollo permanente de soluciones en el ámbito de la ingeniería civil, la socio-economía y el medio ambiente.” (Ecoviana S.R.L., 2024)

1.3. PLANTEAMIENTO DEL PROBLEMA

Las carreteras son vitales para el desarrollo de una región o país, mantenerlas en un estado óptimo para la circulación es importante, los organismos responsables deben asegurar que las vías sean transitables y seguras, manteniendo estas expeditas, debidamente iluminadas y con señalización adecuada.

La señalización inadecuada de carreteras aumenta el número de accidentes de tránsito, a su vez la gravedad de los accidentes se incrementa, pudiendo producir pérdidas humanas y lesiones permanentes, generando desconfianza de los conductores en la infraestructura vial.

Las causas de la señalización inadecuada pueden ser técnicas (distribución inadecuada de las señales), humanas (vandalismo, corrupción en la recepción de obras, personal no calificado) y administrativas (inventarios desactualizados, descoordinación, malos planes de mantenimiento).

La distribución inadecuada de señales de tránsito en puntos críticos -como ser señales de cruces de carretera y cruces peatonales- pueden causar maniobras abruptas y arriesgadas por parte de los conductores lo cual aumenta el riesgo de colisión y generan incumplimiento involuntario de la normativa de tránsito.

Entre las principales causas que provocan una inadecuada distribución de las señales de tránsito se pueden mencionar: la obstrucción de las señales por vegetación, la cantidad o ubicación incorrecta de las mismas, la presencia de mensajes contradictorios, la

sobresaturación de señales, el incumplimiento de las normativas técnicas y la desactualización de los planos, producto de la baja frecuencia de visitas del personal especializado a los puntos críticos.

Una alternativa técnica para mantener los planos actualizados y reducir los costos asociados al desplazamiento de personal especializado hacia los puntos críticos es el uso de videos capturados desde vehículos en movimiento. No obstante, esta metodología puede introducir errores en el análisis de la distribución de señales, debido a factores como los sesgos del observador —que tiende a ver lo que espera ver—, la fatiga visual o la distracción provocada por elementos irrelevantes presentes en los videos (como el cielo, cerros u objetos en la vía).

ECOVIANA S.R.L es una empresa que implementó el análisis de videos como herramienta para elaborar el inventario de la distribución de señales de tránsito en las carreteras. Sin embargo, las percepciones erróneas durante el proceso pueden afectar la precisión de los resultados, generando desconfianza en los informes técnicos y, en consecuencia, una posible disminución en la demanda de los servicios que ofrece la empresa. A continuación, se realiza la identificación y planteamiento del problema.

1.3.1. Identificación del Problema

La empresa ECOVIANA S.R.L dedicada al sector de infraestructura vial desarrolla actividades de supervisión, señalización, mantenimiento de carreteras, gestión en general de infraestructura vial.

Primero se realiza una auditoria de la zona asignada que consiste en analizar el estado actual de la carretera, en este proceso el personal recorre el tramo a pie y va anotando distintos datos, como la ubicación, distancia total del tramo, estado de las señales en carretera. También se realiza un recorrido con un vehículo, este vehículo tiene una cámara para realizar la captura de video del tramo, si es posible con sensores se recupera el estado topográfico (aunque por el alto costo de estos aparatos, no es una práctica común).

El análisis de los videos se realiza en gabinete, una vez que el equipo humano inicia sus labores en las oficinas. Actualmente, para la revisión y evaluación de las capturas se emplea software de reproducción de video, lo que exige que el personal observe grabaciones

extensas con el fin de identificar o confirmar posibles omisiones de señales en los tramos analizados. Este proceso, sin embargo, puede verse afectado por la fatiga visual y el sesgo del observador, lo que obliga a repetir las revisiones y, en consecuencia, incrementa la inversión de tiempo, esfuerzo y recursos humanos necesarios para garantizar la calidad de los informes técnicos y evitar imprecisiones.

1.3.2. Formulación del Problema

¿Cómo puede potenciarse el software de análisis de videos para reducir la información irrelevante presente en las grabaciones de carreteras y optimizar el proceso de inventariación de las señales de tráfico mediante una técnica de rebanado de video que genere imágenes 2D de la distribución de señales que reduzca el sesgo del observador y aumente la confianza en los estudios técnicos realizados por ECOVIANA S.R.L?

1.4. OBJETIVOS

Después de realizar el análisis de la problemática se plantea el objetivo general y los objetivos específicos

1.4.1. Objetivo General

Desarrollar una técnica de rebanación de video que permita generar imágenes 2D de la distribución de señales de tránsito a partir de videos de carreteras, con el fin de potenciar el software de análisis de videos de la empresa ECOVIANA S.R.L.

1.4.2. Objetivos Específicos

- Analizar los procesos actuales de supervisión vial en ECOVIANA S.R.L para identificar los puntos críticos
- Diseñar el modelo de rebanado de video que permita la extracción automatizada de imágenes 2D de la distribución de señales de tránsito
- Crear un prototipo funcional de la técnica de rebanado de video utilizando herramientas de procesamiento de imágenes y análisis espaciotemporal.
- Evaluar cualitativamente el desempeño del sistema desarrollado, a través de grupos focales en ECOVIANA S.R.L.

- Documentar la técnica propuesta y el prototipo funcional.

1.5.JUSTIFICACIÓN

A continuación, se planteará la justificación técnica y la justificación social del presente proyecto.

1.5.1. Justificación Teórica

El proyecto se justifica técnicamente porque busca potenciar el software de análisis de videos actualmente utilizado por ECOVIANA S.R.L. mediante la incorporación de una técnica de rebanado de video capaz de generar imágenes 2D de la distribución de señales de tránsito.

Desde el punto de vista tecnológico, el proyecto implica el uso de métodos de procesamiento de imágenes, visión por computadora y análisis espaciotemporal, aplicados a un entorno real de supervisión vial. Esto favorece la innovación y modernización de los procesos internos de ECOVIANA S.R.L., brindándole una ventaja competitiva mediante la automatización y estandarización de la detección de señales de tránsito.

Además, la reducción del sesgo del observador mediante resultados visuales objetivos y reproducibles refuerza la validez técnica y científica de los estudios realizados, incrementando la confiabilidad de los inventarios viales.

1.5.2. Justificación Social

Socialmente, el proyecto tiene un impacto directo en la seguridad vial y la calidad de los servicios públicos de transporte. La implementación de una técnica más precisa y eficiente para la detección y registro de señales de tránsito permitirá mejorar el mantenimiento de la infraestructura vial, asegurando que las señales estén correctamente ubicadas, visibles y en buen estado.

Esto contribuye a reducir accidentes de tráfico relacionados con señalización deficiente o confusa, beneficiando a conductores, peatones y comunidades.

Asimismo, al optimizar los procesos de inventario y supervisión, el proyecto promueve la transparencia y eficiencia en la gestión pública o privada de redes viales, facilitando la toma de decisiones basadas en datos confiables.

A nivel organizacional, la propuesta también fomenta la capacitación técnica del personal de ECOVIANA S.R.L., incentivando el uso de nuevas tecnologías y fortaleciendo el desarrollo de competencias digitales en el sector vial.

1.5.3. Justificación Económica

Económicamente, la implementación del modelo de rebanado de video representa una inversión estratégica para ECOVIANA S.R.L., ya que permitirá reducir costos operativos asociados al análisis manual de grabaciones extensas. Al disminuir la cantidad de información irrelevante procesada, se optimiza el uso de recursos computacionales y humanos, logrando mayor productividad en menor tiempo.

A mediano y largo plazo, esta innovación fortalece la competitividad de ECOVIANA S.R.L. en el mercado de consultorías viales, posicionándola como una empresa pionera en la aplicación de técnicas inteligentes de análisis audiovisual.

Finalmente, los beneficios económicos también se reflejan en el ámbito público y social, ya que una gestión más eficiente de la infraestructura vial implica ahorros en mantenimiento y reducción de costos por accidentes o señalización inadecuada.

1.6. ALCANCES Y LIMITACIONES

En este apartado se definen los alcances y las limitaciones del presente proyecto

1.6.1. Alcances

El proyecto contemplará los siguientes puntos:

1.6.1.1. Alcance Temática.

La presente investigación se delimita temáticamente al desarrollo de una técnica de rebanado de video para la generación de imágenes 2D de la distribución de señales de tránsito, centrando su alcance técnico en la optimización del software de análisis de videos de ECOVIANA S.R.L.

El proyecto abarcará el análisis de videos de carreteras, con el objetivo de reducir información irrelevante y sesgos del observador, mediante la extracción automatizada de imágenes 2D que representen la ubicación y frecuencia de las señales de tránsito.

Se incluyen todas las fases relacionadas con la investigación, diseño del modelo de rebanado, desarrollo del prototipo funcional y evaluación cualitativa del desempeño, asegurando que la técnica cumpla con los requisitos de confiabilidad y utilidad práctica para los estudios técnicos de supervisión vial de la empresa.

El alcance temático no incluye la supervisión de otros elementos viales como marcas semáforos o estructuras adicionales, ni el desarrollo de un sistema de análisis de video completamente nuevo fuera del software existente de ECOVIANA S.R.L.

1.6.1.2. Alcance Geográfico.

El proyecto se llevará a cabo en las oficinas y áreas operativas de ECOVIANA S.R.L., así como en los tramos de carreteras cuyas grabaciones sean suministradas por la empresa. La validación y pruebas del prototipo se limitarán a los videos disponibles del entorno vial supervisado por ECOVIANA S.R.L., sin extenderse a otras regiones o carreteras externas que no cuenten con datos proporcionados por la empresa.

1.6.1.3. Alcance Temporal

El desarrollo del proyecto se realizará en un período de aproximadamente un año y 3 meses, incluyendo todas las fases: análisis de procesos actuales, diseño del modelo de rebanado de video, desarrollo del prototipo funcional, pruebas internas y evaluación cualitativa. Este marco temporal permitirá cumplir con los objetivos planteados, generar documentación técnica completa y obtener resultados que puedan ser aplicados directamente en las operaciones de ECOVIANA S.R.L.

1.6.2. Limites

En el presente trabajo se desarrolla un prototipo funcional que rebana un volumen espacio temporal estructurado como un video. La técnica de rebanado produce una imagen bidimensional (2D). No se contempla: la adquisición de nuevos videos se trabajará únicamente con el material existente y brindado por la empresa, pruebas cuantitativas masivas de desempeño ni métricas estadísticas fuera de la evaluación cualitativa mediante grupos focales internos.

El proyecto busca servir como herramienta de apoyo al software existente, sin reemplazar los métodos tradicionales de supervisión ni el análisis manual que la empresa realiza actualmente.

Dentro del proyecto no se realiza una detección de señales de tránsito ni otros objetos o irregularidades dentro la carretera, no identifica líneas de manera automática.

CAPITULO 2.

MARCO TEÓRICO

2.1. TÉCNICA

“La técnica es un conjunto de procedimientos, materiales o intelectuales, es aplicado en una tarea específica, con base en el conocimiento de una ciencia o arte, para obtener un resultado determinado” (Significados, 2023).

Dentro del contexto del presente proyecto una técnica es un conjunto de procedimientos matemáticos y computacionales para obtener un resultado determinado.

2.2. TÉCNICA DE REBANADO

La creación de la técnica de rebanado de video es el producto final que se espera obtener, incluye la investigación y aplicación de distintas ramas de la matemática como geometría proyectiva y su realización computacional, con las bases de la geometría proyectiva se puede convertir un sistema de 3 dimensiones a 2 dimensiones.

El Universo se concibe como un espacio de 4 dimensiones, como lo describe la geometría de Minkowski. El espacio – tiempo descrito por Minkowski es un espacio de 4 dimensiones (t, x, y, z) (Doran & Lasenby, Geometric Algebra for Physicists, 2003).

Cuando se captura un video, se proyecta el espacio cuatridimensional en un espacio tridimensional, (t, x, y) , este espacio tridimensional puede ser proyectado sobre la base de las ideas de la Transformada de Radon, a un espacio bidimensional (x, y) . La técnica de rebanado consiste en obtener cortes bidimensionales de información a partir de la proyección tridimensional que contiene el video.

2.3. GEOMETRÍA

La geometría es una rama fundamental de las matemáticas que se ocupa del estudio de las propiedades y relaciones de las figuras en el espacio. Históricamente, la geometría era una ciencia empírica centrada en la medición de longitudes, áreas y volúmenes, pero ha evolucionado considerablemente a lo largo de los siglos. Desde sus orígenes en la antigua Grecia con Euclides hasta su desarrollo moderno en la geometría no euclidiana y la geometría diferencial, la disciplina ha ampliado sus horizontes para incluir conceptos abstractos y multidimensionales.

En su obra Los Elementos, Euclides sentó las bases de la geometría clásica al definir los axiomas básicos y las propiedades de figuras planas y espaciales como puntos, rectas y

polígonos. En este enfoque tradicional, conocido como geometría euclidiana, las figuras se analizan en un espacio plano en el que las líneas paralelas nunca se cruzan y los ángulos de un triángulo siempre suman 180 grados. Sin embargo, en el siglo XIX, matemáticos como Carl Friedrich Gauss, Nikolai Lobachevsky y Bernhard Riemann ampliaron la geometría más allá de las ideas euclidianas y desarrollaron la geometría no euclidiana, en la que las leyes de los triángulos y las paralelas varían en función de la curvatura del espacio. (Hartshorne, 2013)

En el presente proyecto se utilizara la geometria euclidiana. Alrededor del año 300 a.C., Euclides de Alejandría escribió los Elementos, que constan de 13 libros con 465 proposiciones y 372 problemas. En los Elementos, se expone de manera lógica y estructurada los conocimientos de la época acerca del espacio en el que vivimos. Fundamentalmente, Euclides concibe el espacio como un conjunto continuo de puntos (un punto es algo que no tiene partes).

Además, estableció los cinco célebres postulados (Ver Figura 1):

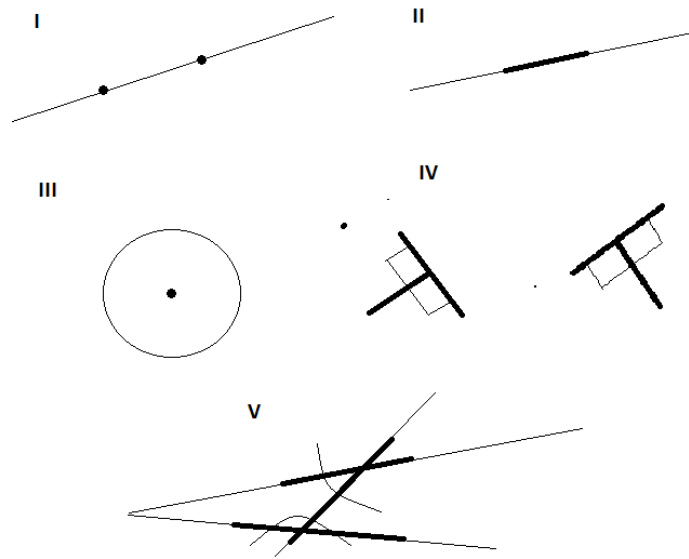
- I. Por dos puntos distintos pasa una recta.
- II. Un segmento rectilíneo puede ser siempre prolongado.
- III. Hay una única circunferencia con un centro y un diámetro determinados.
- IV. Los ángulos rectos son congruentes.
- V. Si una recta corta a dos rectas de forma que los ángulos interiores de un mismo lado son inferiores a dos rectos, las rectas cortadas se interceptan en un punto que está en el lado en el cual los ángulos son menores a dos rectos.

2.4.MATEMÁTICA

A continuación, se muestran los conceptos matemáticos a utilizar para la elaboración del presente proyecto. Se dividen entre Matemática cuantitativa y matemática cualitativa.

Figura 1

Descripción grafica de los postulados de Euclides



Nota: Descripción grafica de los 5 postulados de Euclides *Fuente:* (Wikipedia contributors, s.f.)

2.4.1. Matemática Cuantitativa

Según el matemático, d'Alambert, los números son abstracciones sacadas del mundo sensible. En el mundo sensible pueden estar los números Naturales o la idea del continuo (es decir los números Reales). Kronecker, defendía que la única realidad son los números Naturales, lo demás son construcciones de los matemáticos. En contraposición, René Thom defendía que la única realidad es el continuo, los números naturales son sólo un subconjunto de los números Reales.

2.4.1.1. Números naturales (N)

Sobre los números naturales se puede definir una función inyectiva llamada *sucesor*

Ecuación 1: Función sucesor

$$S: N \mapsto N$$

Donde:

S: función sucesor.

N: conjunto de los números naturales.

Sobre N , está dada una relación de equivalencia $=$. Se pueden definir dos operaciones binarias asociativas, llamadas *suma* y *multiplicación*, respectivamente.

Ecuación 2: Definición de operaciones binarias

$$+, \cdot : N \times N \mapsto N$$

Donde:

N : conjunto de los números naturales.

$+$: operación binaria llamada suma.

$\cdot$: operación binaria llamada multiplicación.

$N \times N$: producto cartesiano de N consigo mismo.

$\mapsto$: indica que la operación asigna a cada par ordenado de $N \times N$ un elemento de N .

Los números Naturales se pueden extender adicionando el cero

Ecuación 3: Conjunto de los números naturales extendido con el cero

$$\omega := \{0\} \cup N$$

Donde:

ω : conjunto de los números naturales extendido con el cero.

N : conjunto de los números naturales.

$\cup$: operación de unión de conjuntos.

0 : elemento agregado al conjunto de los números naturales.

Ahora, para cualquier número natural se cumple la Ecuación 4

Ecuación 4: Proposición $S(n)$

$$\forall n \in \omega : S(n) \neq 0$$

Donde:

n : elemento del conjunto ω .

$S(n)$: sucesor de n .

$\neq$: relación de desigualdad.

$\forall$: cuantificador universal ("para todo").

Si ϕ es un predicado unario tal que $\phi(0)$ es verdad y para cada número natural n , siendo verdadero $\phi(n)$ implica la verdad de $\phi(S(n))$; Entonces, $\phi(n)$ es verdadero para todos los naturales.

2.4.1.2. Números enteros ($\mathbb{Z}$)

Ecuación 5: Principio de inducción matemática

$$\phi(0) \wedge (\forall n \in \mathbb{N} : \phi(n) \rightarrow \phi(S(n))) \rightarrow \forall n \in \mathbb{N} : \phi(n)$$

Donde:

ϕ : predicado unario definido sobre los números naturales.

$\wedge$: conjunción lógica ("y").

$\rightarrow$: implicación lógica.

$\forall$: cuantificador universal.

n : número natural.

Los números enteros son el conjunto de números que incluye todos los números naturales (positivos), sus opuestos negativos y el cero. Se representan con la letra $\mathbb{Z}$ y no tienen fracciones ni decimales. (Editorial Etecé, 2025)

Sobre $\omega \times \omega$, se puede definir una relación de equivalencia (Ver Ecuación 6).

Los números enteros $\mathbb{Z}$, son el conjunto de las clases de equivalencia generadas por $\sim$

Ecuación 6: Relación de equivalencia

$$(m, n) \sim (q, r) \Leftrightarrow m + r = q + n$$

Donde:

$m, n, q, r \in \omega$: números naturales.

(m, n) y (q, r) : pares ordenados de números naturales.

$\sim$: relación de equivalencia definida sobre $\omega \times \omega$.

$\Leftrightarrow$: equivalencia lógica ("si y solo si").

Ecuación 7: Clase de equivalencia

$$\mathbb{Z} = (\omega \times \omega) / \sim$$

Donde:

$\mathbb{Z}$: conjunto de los números enteros.

$\omega \times \omega$: conjunto de todos los pares ordenados de números naturales.

$/ \sim$: conjunto cociente formado por las clases de equivalencia de la relación $\sim$.

2.4.1.3. Números racionales ($\mathbb{Q}$)

Sobre $\omega \times \omega$, se puede definir con una relación de equivalencia (Ver)

Ecuación 8: Relación de equivalencia

$$(m, n) \sim (q, r) \Leftrightarrow m \cdot r = q \cdot n$$

Donde:

$m, n, q, r \in \omega$: números naturales.

(m, n) y (q, r) : pares ordenados de números naturales.

$\sim$: relación de equivalencia definida sobre $\omega \times \omega$.

$\cdot$: operación de multiplicación en los números naturales.

$\Leftrightarrow$: equivalencia lógica ("si y solo si").

Los números racionales $\mathbb{Q}$, son el conjunto de las clases de equivalencia $[m, n]$ generadas por $\sim$.

Ecuación 9: Clases de equivalencia

$$\mathbb{Q} = \{ [m, n] : m, n \in \omega, n \neq 0 \}$$

Donde:

$\mathbb{Q}$: conjunto de los números racionales.

$[m, n]$: clase de equivalencia del par ordenado (m, n) .

$n \neq 0$: el denominador no puede ser cero.

Cada clase de equivalencia $[m, n]$ representa el número racional m / n

2.4.1.4. Números reales ($\mathbb{R}$)

El conjunto de los números reales se puede expresar como la unión de los números racionales y los números irracionales.

Ecuación 10: Conjunto de los números $\mathbb{R}$

$$\mathbb{R} = \mathbb{Q} \cup \mathbb{Q}^c$$

Donde:

$\mathbb{R}$: conjunto de los números reales.

$\mathbb{Q}$: conjunto de los números racionales.

$\mathbb{Q}^c$: complemento de $\mathbb{Q}$, correspondiente al conjunto de los números irracionales.

$\cup$: operación de unión de conjuntos.

2.4.2. Fundamentos algebraicos

A continuación, se mencionan los elementos algebraicos que serán utilizados en la construcción del modelo matemático.

2.4.2.1. Estructuras Algebraicas

Una estructura algebraica consta de:

- a) Un conjunto no vacío A (llamado conjunto subyacente, conjunto portador o dominio).
- b) Una colección de operaciones en A (típicamente, suma y multiplicación).
- c) Un conjunto finito de propiedades que estas operaciones deben satisfacer.

Las estructuras algebraicas básicas son grupos, anillos y campos.

El grupo es la estructura más básica y al mismo tiempo la más importante. Un grupo es un par $(G, \cdot)$ que consta de un conjunto G no vacío y una operación binaria " $\cdot$ ", es decir, un mapa: $\cdot : G \times G \mapsto G$, que cumple con la asociatividad, la existencia de un elemento neutro y la existencia de un elemento inverso.

- a) $(g \cdot h) \cdot k = g \cdot (h \cdot k) \quad \forall g, h, k \in G$ (Asociatividad).
- b) $\exists e \in G : \quad \forall g \in G : \quad e \cdot g = g$, ("Existencia de un elemento neutro")

c) $\forall g \in G \exists g' \in G: g' \cdot g = e$ (“Existencia de elemento inverso”).

Un grupo se llama Abelian o conmutativo si $(G, \cdot)$ satisface adicionalmente la propiedad conmutativa, descrita a continuación.

$$g \cdot h = h \cdot g \quad \forall g, h \in G \quad (\text{Conmutatividad})$$

Un anillo como unidad es una tripleta $(R, +, \cdot)$ que consiste en un conjunto R no vacío y dos operaciones, la suma:

$$+: R \times R \mapsto R,$$

y la multiplicación:

$$\cdot: R \times R \mapsto R,$$

tal que se cumple las siguientes propiedades:

- a) $(R, +)$ un grupo Abelian (con neutro 0_R).
- b) $(g \cdot h) \cdot k = g \cdot (h \cdot k) \quad \forall g, h, k \in G$, (“Asociatividad de la multiplicación”).
- c) Existe un elemento $1_R \in R$ tal que $1_R \cdot a = a$ para todo $a \in R$ (“la unidad”).
- d) $a \cdot (b + c) = a \cdot b + a \cdot c$ y $(b + c) \cdot a = b \cdot a + c \cdot a$ para todo $a, b, c \in R$ (“distributividad”).

2.4.2.2. Espacio vectorial

Sea, $(V, +)$ un Grupo Abelian. Si con el cuerpo $\mathcal{K}$, esta definida una ley de composición externa $\mathcal{K} \times V \mapsto V$ que para $\forall \lambda, \beta \in \mathcal{K}$ y $x, y \in V$ satisfacen los siguiente enunciados:

- a) $(\lambda + \beta)x = \lambda x + \beta x$
- b) $\lambda(x + y) = \lambda x + \lambda y$
- c) $\lambda(\beta x) = (\lambda\beta)x$
- d) $1x = x$, donde 1 es el neutro multiplicativo del cuerpo $\mathcal{K}$.

Un anillo con unidad $(R, +, \cdot)$ se llama conmutativo si $a \cdot b = b \cdot a$ para todo $a, b \in R$. Si $(R, +, \cdot)$ es un anillo con unidad, a es invertible en R cuando existe $a' \in R$ tal que $a \cdot a' = a' \cdot a = 1_R$, sea

$$R^* = \{a \in R \mid a \text{ es invertible}\}$$

en conjunto de invertibles en R .

Un anillo conmutativo con unidad $(R, +, \cdot)$ se llama campo (cuerpo) si $R^* = R \setminus \{0\}$.

En síntesis, desde la perspectiva de la matemática cualitativa:

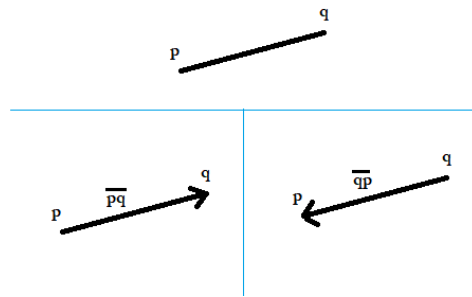
- a) $(\mathbb{Z}, +)$ es un grupo.
- b) $(\mathbb{Z}, +, \cdot)$ es un anillo con unidad.
- c) $(\mathbb{Z}, +, \cdot)$, junto con una relación de orden $<$, es un dominio de integridad ordenado.
- d) $(\mathbb{R}, +, \cdot)$ es un campo (cuerpo, field).

2.4.2.3.El grupo $(F, +)$

Los segmentos de la visión euclidiana se sustituyen por otros objetos llamadas flechas como se muestra en la Figura 2.

El conjunto de todas las flechas se denomina F . Este cambio permite introducir la operación de suma geométrica, como se muestra en la Figura 4, la resultante vector $a + b$ es lo mismo que $b + a$. Esto encuentra una expresión algebraica en la afirmación de que la suma es conmutativa. En el diagrama de la derecha el vector $a + b + c$ se construye de dos maneras diferentes, como $a + (b + c)$ y como $(a+b) + c$. El hecho de que los resultados sean iguales es una expresión geométrica de la asociatividad de la suma de vectores.

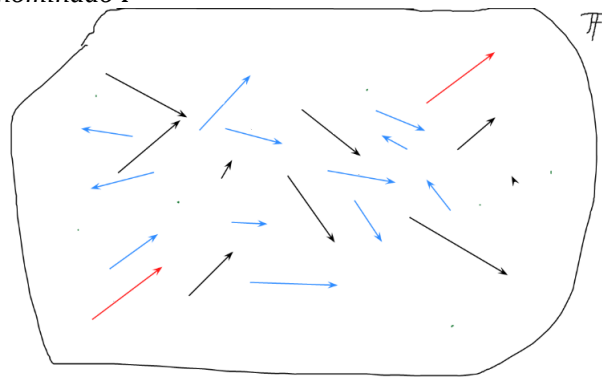
Figura 2
 Segmento y flecha según la geometría euclidiana.



Nota: Una flecha es un segmento dirigido, imagen creada en base a conceptos de vectores.

Fuente: Elaboración propia

Figura 3
 Conjunto de flechas denominado F

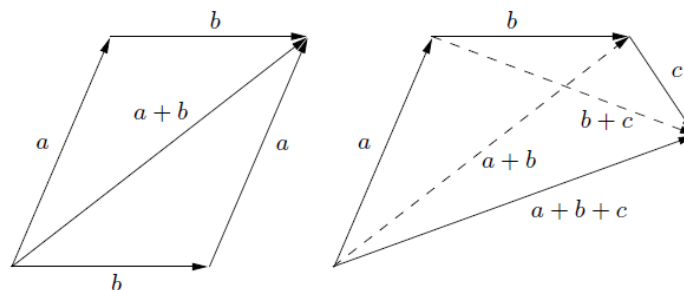


Nota: Las flechas son segmentos dirigidos que permiten modelar posiciones y segmentos. *Fuente:*

Elaboración propia

Figura 4

Una imagen geométrica de la suma de vectores.



Nota: El resultado de $a + b$ se forma sumando la cola de b a la cabeza de a . *Fuente:* (Doran & Lasenby, Geometric Algebra for Physicists, 2003)

2.4.2.4.Producto escalar

En el modelado del espacio, se usan como escalares a los números reales. Por esta razón, el resultado del producto de dos vectores será un número real $a \cdot b \in \mathbb{R}$. Se exige que la operación cumpla los siguientes axiomas:

- a) $a \cdot b = b \cdot a$,
- b) $a \cdot (\lambda b) = \lambda a \cdot b$,
- c) $a \cdot (b + c) = a \cdot b + a \cdot c$,
- d) $a \cdot a \geq 0$, cuando $a \neq 0$.

Con el producto escalar se puede encontrar tamaños de los vectores, distancias entre puntos y ángulos, de la siguiente forma:

- a) $|a|^2 = a \cdot a$,
- b) $|a \cdot b| \leq |a||b|$,
- c) $\cos \theta \equiv \frac{a \cdot b}{|a||b|}$
- d) Dos vectores a, b son ortogonales, cuando $a \cdot b = 0$.

2.5. Algoritmo de línea de Bresenham

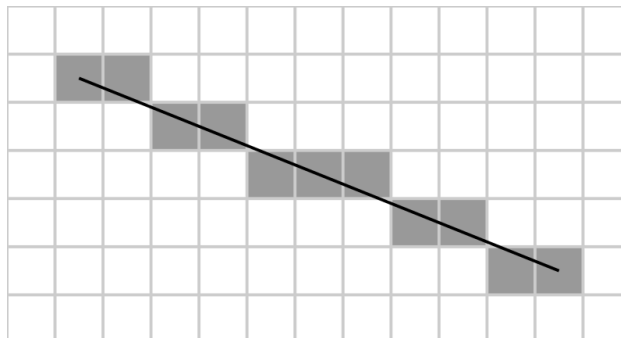
El algoritmo de línea de Bresenham es un método rápido y eficiente para trazar líneas rectas en dispositivos gráficos rasterizados, utilizando únicamente cálculos enteros, lo que lo hace muy rápido y sencillo de implementar. Fue desarrollado originalmente por Jack Elton Bresenham en 1962 para dibujar líneas en un plóter, y desde entonces se ha convertido en un algoritmo fundamental en gráficos por computadora. (Bresenham, 1965)

El algoritmo determina qué píxeles deben activarse para acercarse lo más posible a una línea recta entre dos puntos dados, eligiendo en cada paso entre dos posibles píxeles basándose en un error acumulado, sin necesidad de operaciones de coma flotante. Se adapta recorriendo el eje con mayor diferencia siempre avanzando de uno en uno, y para el otro eje decide si avanza o no en el siguiente pixel, logrando así una representación precisa y eficiente.

Además, existen variantes y extensiones del algoritmo que permiten rasterizar curvas y circunferencias. Su principal ventaja es la velocidad y la precisión sin usar operaciones costosas, haciendo que sea muy popular en sistemas donde se dibujan gráficos a nivel de píxeles.

Figura 5

Representación gráfica del algoritmo de línea de Bresenham



Nota: En la imagen notamos una línea negra, el algoritmo de línea de Bresenham ayuda a recuperar los píxeles sobre los cuales pasa la línea negra. *Fuente:* (Henrik Abelson, 2006)

2.6.SEÑALES DE TRANSITO

Una señal de tránsito (o señal de tráfico) es un dispositivo o signo colocado en la vía pública, cuyo propósito fundamental es transmitir información esencial a todos los usuarios, incluyendo conductores y peatones. Estas señales son cruciales para la seguridad vial, ya que su función se clasifica en tres objetivos principales: prevenir a los usuarios sobre posibles peligros o condiciones especiales en el camino, reglamentar las normas de circulación indicando prohibiciones, restricciones u obligaciones (como límites de velocidad o la señal de "Pare"), e informar y guiar sobre direcciones, destinos, servicios o puntos de interés. En esencia, las señales de tránsito dirigen el comportamiento en las calles y carreteras, asegurando una forma de transitar correcta, segura y fluida. Se presentan principalmente en dos formatos: el señalamiento vertical (placas instaladas en postes) y el señalamiento horizontal (marcas pintadas directamente sobre el pavimento, como las líneas de carril o los pasos peatonales)

En este apartado se describen algunas de las señales de tránsito utilizadas en dentro del territorio boliviano. Se utilizo un (recorte o parte) del tramo del que se tiene dentro de los

datos. Entre las señales de tránsito más comunes encontramos las señales horizontales, verticales, flechas.

Figura 6

Ejemplo de señales de tránsito en Bolivia



Nota: Imagen publicada por Vías Bolivia, muestra ejemplos de señales de tráfico en Bolivia

Fuente: (Vías Bolivia, 2023)

2.6.1. Señales horizontales

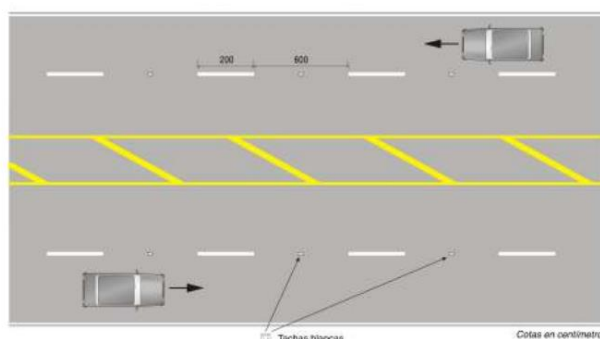
- a) Líneas de carril: La función principal de las líneas de pista es ordenar el tránsito y posibilitar un uso más seguro y eficiente de las vías, especialmente en zonas congestionadas. Estas líneas separan flujos de tránsito en la misma dirección, y pueden ser de dos tipos; continuas o segmentadas.
 - 1) Líneas segmentadas normales: se dispondrán en tramos de una vía, en donde se permite reglamentariamente la maniobra de cambio de pista. La demarcación elevada debe ser de color blanco e instalarse centrada en todas las brechas o brecha por medio (ABC; APIA XXI Ingenieros y Arquitectos Constructores, 2018)
 - 2) Línea blanca continua: La línea continua sobre la calzada significa que ningún conductor con su vehículo debe atravesarla ni circular sobre ella. La línea continua se utiliza para: demarcar la separación de carriles y demarcar el borde derecho de la calzada (ABC; APIA XXI Ingenieros y Arquitectos Constructores, 2018).

Figura 7
Líneas segmentadas y continuas en video analizado



Nota: Líneas segmentadas y continuas en video analizado *Fuente:* (Video Tramo La Paz - LLujturi)

Figura 8
Línea segmentada en el manual oficial de la ABC



Nota: Imagen referencial de la línea segmentada *Fuente:* (ABC; APIA XXI Ingenieros y Arquitectos Constructores, 2018)

Figura 9
Línea continua en el manual de la ABC.



Nota: Imagen referencial de la línea continua *Fuente:* (ABC; APIA XXI Ingenieros y Arquitectos Constructores, 2018)

2.6.2. Señales verticales

- a) Señal vertical vial lateral derecha: Se empleará en la ruta principal para advertir al conductor la proximidad a un empalme o desvío por el lado izquierdo o derecho, de la calzada, en la que hay tránsito en los dos sentidos, formando un ángulo aproximado de 90°

Figura 10

Señal vertical vial lateral derecha en el video analizado



Nota: Imagen de la señal vial lateral derecha en el video. *Fuente:* (Video Tramo La Paz - LLujturi)

Figura 11

Señal vertical vial lateral derecha en el manual de la ABC



Nota: Imagen de la señal vial lateral derecha en el manual. *Fuente:* (ABC; APIA XXI Ingenieros y Arquitectos Constructores, 2018)

- b) Señal informativa lateral con flechas: Esta señal vertical es de color azul general muestra nombres de los lugares a donde dirigen los desvíos. (ABC; APIA XXI Ingenieros y Arquitectos Constructores, 2018)

Figura 12

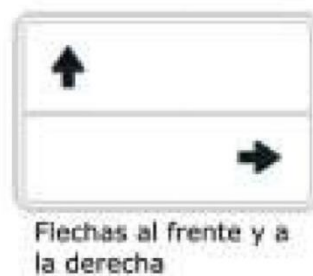
Señal informativa lateral con flechas en el video analizado



Nota: Imagen de la señal informativa lateral derecha en el video. *Fuente:* (Video Tramo La Paz - LLujturi)

Figura 13

Señal informativa lateral con flechas en el manual



Nota: Imagen de la señal vial lateral derecha en el manual. *Fuente:* (ABC; APIA XXI Ingenieros y Arquitectos Constructores, 2018)

2.6.3.Flechas

Son señales informativas que se usan para asociar pistas a determinados destinos y para indicar, antes y en una salida, la dirección sentida a seguir para llegar a ellos. la flecha debe ser oblicua ascendente u horizontal, representando adecuadamente el ángulo de la salida.

Figura 14

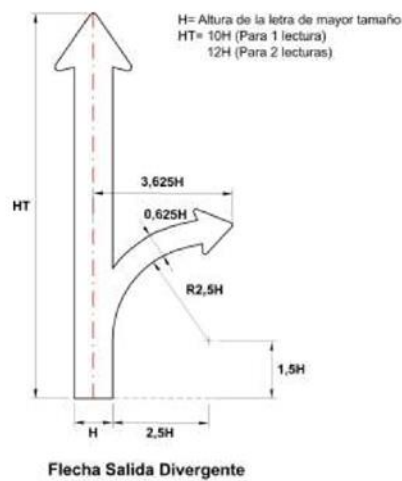
Flechas rectas y de cruce a la derecha



Nota: Imagen de las flechas rectas y de cruce a la derecha en el video. *Fuente:* (Video Tramo La Paz - LLujturi)

Figura 15

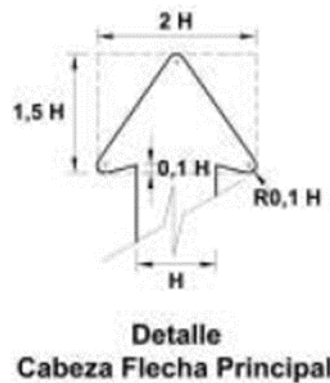
Flecha con salida a la derecha en el manual del ABC.



Nota: Imagen de las flechas de cruce a la derecha en el manual oficial de la ABC. Fuente: (ABC; APIA XXI Ingenieros y Arquitectos Constructores, 2018)

Figura 16

Flecha recta en el manual de la ABC



Nota: Imagen de las flechas rectas en el manual oficial de la ABC Fuente: (ABC; APIA XXI Ingenieros y Arquitectos Constructores, 2018)

2.7. ANÁLISIS DE DATOS

Según el libro "Data Analysis: Scientific Modeling and Practical Application", el análisis de datos se define como un campo amplio que abarca una serie de actividades esenciales en la era de la información. Este campo incluye áreas emergentes como la minería de datos y la minería web, así como enfoques tradicionales enfocados en la clasificación y la organización del conocimiento. El análisis de datos consiste en aplicar métodos y técnicas

para convertir datos en información valiosa utilizando diversos procesos como la clasificación, el clustering, la regresión, el análisis de series temporales y la modelización de relaciones entre variables (Gaul, Opitz, & Schader, 2000) .

El análisis de datos es un campo que ha experimentado una notable evolución en las últimas décadas. Desde sus humildes comienzos como nicho de la estadística, se ha convertido en un componente vital de la toma de decisiones en casi todos los sectores. El origen de la analítica de datos se remonta a la década de 1950, cuando las empresas empezaron a utilizar ordenadores para automatizar los procesos de datos manuales. Los primeros ordenadores centrales permitían realizar cálculos rudimentarios y análisis estadísticos. Esta época sentó las bases de lo que se convertiría en la analítica de datos moderna.

La década de 1980 fue testigo del auge de las hojas de cálculo, con la introducción de VisiCalc y, más tarde, Microsoft Excel. Estas herramientas revolucionaron el tratamiento de datos y lo hicieron accesible a un público más amplio. Este periodo marcó un cambio significativo del tratamiento de datos centrado en el mainframe a los ordenadores personales. También asistimos al desarrollo de sistemas de gestión de bases de datos como Oracle, IBM DB2 y Microsoft SQL Server. Estos sistemas ofrecían potentes herramientas para almacenar, organizar y recuperar grandes volúmenes de datos. Esto permitió una gestión de datos más estructurada y eficaz.

En los años 90 aparecieron las herramientas de Business Intelligence, que proporcionaban cuadros de mando y funciones de elaboración de informes. Herramientas como Cognos (posteriormente adquirida por IBM) y Business Objects permitieron a las empresas obtener información de sus datos, facilitando una mejor toma de decisiones. A mediados de la década de 2000 se produjo un cambio de paradigma con la llegada de Big Data. La proliferación de Internet, las redes sociales y los dispositivos IoT provocaron una explosión de datos. Tecnologías como Hadoop y las bases de datos NoSQL surgieron para gestionar este volumen masivo de información. La era del Big Data también fue testigo de la aparición del procesamiento y análisis de datos en tiempo real. Tecnologías de streaming de datos como Apache Kafka y plataformas como Spark Streaming permitieron a las organizaciones procesar y analizar los datos a medida que se generaban. Esta capacidad en tiempo real es vital en aplicaciones como la detección de fraudes, la supervisión del

rendimiento de la red y el suministro de contenidos personalizados. Los avances en el aprendizaje automático y la inteligencia artificial han permitido realizar análisis más sofisticados. El análisis predictivo, que utiliza datos históricos para hacer predicciones futuras, se convirtió en una poderosa herramienta para las empresas que buscan anticipar tendencias y tomar decisiones proactivas. Plataformas en la nube como AWS, Azure y Google Cloud ofrecían recursos escalables para crear e implantar modelos de análisis predictivo.

A pesar de su enorme potencial, el análisis predictivo se enfrenta a retos relacionados con la privacidad de los datos, los sesgos y la interpretabilidad de los modelos. Estas consideraciones éticas son primordiales a la hora de tomar decisiones basadas en conocimientos predictivos. También asistimos al auge de herramientas como Tableau, Power BI y D3.js, que facilitan la creación de representaciones visualmente atractivas de los datos. La visualización de datos se ha convertido en un elemento crucial para transmitir información a un público más amplio y facilitar una mejor comunicación dentro de las organizaciones.

En los últimos años se ha producido un aumento de la aplicación de la inteligencia artificial al análisis de datos. Los modelos de aprendizaje automático se utilizan ahora para automatizar tareas, descubrir patrones ocultos y generar perspectivas que podrían pasarse por alto con los enfoques tradicionales. Se trata de un nuevo y apasionante horizonte para todo el mundo y no sólo para la analítica, pero profundizaremos en ello un poco más en los próximos días. (Ayo-Vaughan, 2023)

2.7.1.Sensores

Los sensores son dispositivos que imitan la capacidad de percepción de los seres humanos, existen distintos tipos de sensores relacionados con los sentidos: vista, oído, tacto, es decir, que reaccionan a la luz, el sonido, el contacto, etc, y reaccionan a la información como nuestro cerebro reacciona a los estímulos recibidos por los sentidos. Estos dispositivos electrónicos nos permiten interactuar con el entorno, de forma que nos proporcionan información de ciertas variables que nos rodean para poder procesarlas y así generar órdenes o activar procesos. Actualmente se han convertido en algo cotidiano que podemos encontrar en múltiples aparatos domésticos como: mandos a distancia, sistemas de alarma y

seguridad, electrodomésticos, domótica, etc. De igual manera están presentes en automóviles, telefonía móvil, medicina y en los procesos de automatización industriales.

Con la incorporación de los sensores a los sistemas electrónicos se les ha dotado de cierta “inteligencia” artificial, ya que a través de la información que recolectan, y una vez procesada adecuadamente, permiten tomar con precisión y rapidez las mejores decisiones dentro del entorno para el que están diseñados dichos sistemas electrónicos. (Serna Ruiz, Ros García, & Rico Noguera, 2010)

2.7.2.Datos

Un dato se define como un conjunto discreto de hechos objetivos que representan aspectos concretos de la realidad. Aunque los datos son registros precisos de eventos o fenómenos, por sí mismos carecen de interpretación y no ofrecen explicaciones sobre las causas subyacentes. Su acumulación sin un propósito claro puede dificultar la identificación de información relevante. A pesar de estas limitaciones, los datos son fundamentales en la creación de información significativa para organizaciones y sectores diversos, ya que proporcionan la base sobre la cual se realizan análisis y tomas de decisiones. Sin embargo, es crucial entender que los datos por sí solos no ofrecen juicios de valor ni guían directamente hacia acciones específicas, ya que requieren interpretación y contextualización para convertirse en información útil y orientativa. Los datos son el resultado obtenido por los sensores físicos las cuales pueden ser imágenes en diferentes canales, por ejemplo, de RGB, Algunas veces, las imágenes pueden estar georreferenciadas. (Carrión, 2017)

2.7.3. Información y Conocimiento

Se trata de descubrir, a partir de las imágenes (datos), qué está presente en el mundo, dónde están ubicadas las cosas y cómo se interpreta la escena. Para extraer información y conocimiento las actividades recurrentes son: modelación matemática, modelación computacional, programación, y pruebas. Matemáticamente, en cada canal, una imagen se modela como una función de dos variables, cuyos valores son las intensidades de la luz. Los cambios bruscos en el gradiente permiten ubicar bordes en las imágenes. La modelación computacional tiene que ver con el uso de las mejores estructuras de datos para que las imágenes puedan ser almacenadas y procesadas en un computador. Una vez que se

ha desarrollado el modelo computacional, se pueden codificar los programas que realicen los algoritmos y luego probarlos. Al poder contar e identificar figuras en la imagen, es posible avanzar hacia la interpretación de la escena. Es decir, por ejemplo, si aparecen unos cuantos rectángulos blancos paralelos, podríamos interpretar ese conjunto de figuras como un paso de cebra. Otra fase en la obtención de información y conocimiento es la foto análisis, que consiste en etiquetar las figuras geométricas detectadas con denotaciones conceptuales del ámbito de la seguridad vial.

La minería de datos es la extracción de conocimiento valioso a partir de datos no estructurados utilizando algoritmos avanzados para descubrir patrones ocultos, tendencias y relaciones para tomar decisiones informadas en las organizaciones. (Xie & Zhang, 2022)

La fase del conocimiento se refiere a la foto interpretación, según nuestro criterio, esta es la fase que requiere de los aportes de un experto humano que permitan interpretar la escena y también evaluar la calidad del marcado de la señalización. El proceso automático solo puede ser una ayuda en esta fase, ya que una automatización completa requeriría de un tiempo más extenso del trabajo.

La visualización es una capacidad del ser humano que se puede dividir en dos formas, la percepción de un objeto que es visto o tocado y la imagen mental que se genera como resultado de esta percepción. (Merino, s.f.). En el contexto de big data, se refiere a la representación gráfica de información compleja en gráficos, mapas de calor y otros formatos visuales. Esta tecnología se centra en procesar datos y convertirlos en imágenes o visualizaciones atractivas y fáciles de interpretar para todos los usuarios, independientemente de su nivel de conocimientos técnicos. (Lia, 2023)

2.7.4. Variables y sus Dominios

Las fuentes de datos emiten valores que se pueden caracterizar como variables. A cada variable le corresponde un dominio que es el conjunto de los posibles valores que puede emitir la fuente. Los sensores establecen un mapeo entre un dominio de eventos y algún dominio de valores convenientemente elegido (a menudo un subconjunto de los números reales).

Una vez que las variables están definidas se realiza un análisis exploratorio de datos que es un enfoque utilizado en la ciencia de datos y estadística para analizar y resumir las características principales de los datos, generalmente con la ayuda de métodos visuales. Antes de aplicar modelos o realizar análisis complejos, EDA se utiliza para entender mejor los datos. (Xie & Zhang, 2022)

2.7.5.Datos Espacio Temporales

Los datos espaciotemporales se refieren a la información que combina componentes espaciales y temporales, captando cómo evolucionan los fenómenos tanto en el espacio como en el tiempo (Chen, 2023). Este tipo de dato ha sido utilizado en el desarrollo de técnicas como los modelos de regresión espaciotemporal para analizar grandes conjuntos de datos recogidos a lo largo del tiempo, lo que permite estudiar relaciones y patrones complejos en datos referenciados espacialmente (Schaumberg, 2022)

2.7.6.Visualización

La visualización es una capacidad del ser humano que se puede dividir en dos formas, la percepción de un objeto que es visto o tocado y la imagen mental que se genera como resultado de esta percepción. (Merino, s.f.)

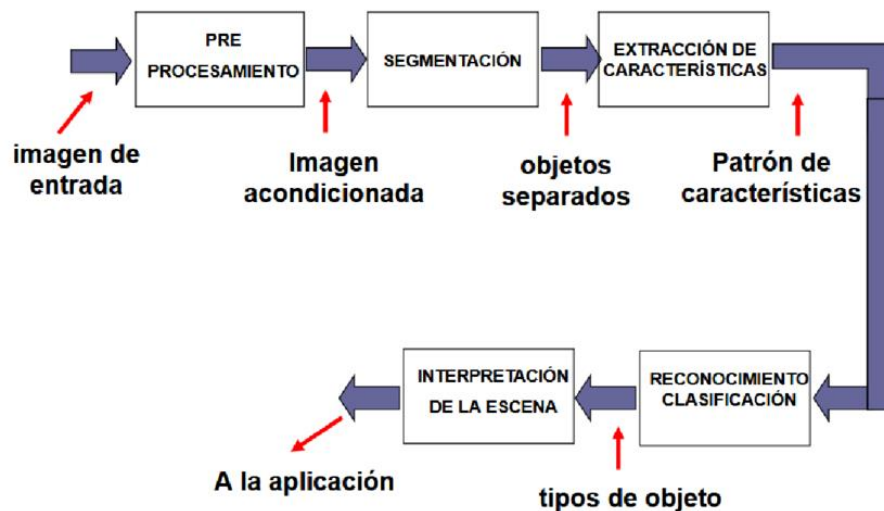
2.8.VISIÓN POR COMPUTADOR

El objetivo de la visión por computador (VC) es que las máquinas vean e interpreten el mundo como los seres biológicos. A partir de una imagen o secuencia de imágenes permite extraer información, analizar, interpretar el mundo. Es el resultado de décadas de trabajo en integrar la Biología, la Neurociencia y las Ciencias de la Computación.

En la Figura 17 se puede notar que existen varios pasos que se realizan desde la adquisición de imágenes hasta llegar a las aplicaciones que pueden ser: Reconocimiento facial, Sistemas Industriales de Visión, Vigilancia, Conducción Autónoma, etc., así como biométrico

Figura 17

Descripción de un flujo común de la visión por computadora



Nota: Descripción de un flujo común de la visión por computadora *Fuente:* Loaiza & Manzano (2012).

La visión por computadora está ligada a varias disciplinas que se estudian a lo largo de la formación para ingeniería de sistemas. El núcleo cercano a la VC se compone de reconocimiento de patrones aprendizaje automático, inteligencia artificial, arquitectura de computador e ingeniería de software. Sin embargo, se requieren conocimiento de otras disciplinas asociadas, a continuación, describiremos algunas de ellas. A continuación, se explorarán algunas de estas disciplinas.

2.8.1. Procesamiento de Imágenes

El procesamiento de imágenes es un campo de la ingeniería de sistemas, que se encarga del análisis, manipulación y uso de imágenes digitales en un entorno de computación científica. Puede ser aplicada en distintas áreas, como el reconocimiento facial, la edición de fotos y el diagnóstico médico. (Gonzalez & Woods, 2017)

2.8.2. Óptica

La óptica en física es una rama fundamental que se ocupa de la generación, propagación y detección de la luz, abarcando el estudio del comportamiento de la luz, las interacciones con la materia y el desarrollo de instrumentos para la detección de la luz. Este campo tiene

una rica historia, con importantes contribuciones desde la comprensión de la óptica ondulatoria dentro de la teoría del electromagnetismo de Maxwell, que condujo al nacimiento de la teoría y la óptica cuánticas. Los recientes avances en óptica han hecho posibles tecnologías innovadoras que abordan diversos retos sociales, como las comunicaciones, las ciencias médicas, la eficiencia energética y la sostenibilidad medioambiental. Además, la aplicación de los principios ópticos se ha extendido a ámbitos como la astronomía, donde se utilizan técnicas adaptadas de la óptica para visualizar materiales opacos como los tejidos vivos, lo que pone de manifiesto el carácter interdisciplinar y la relevancia práctica de la óptica en la ciencia moderna. (Merali, 2015)

2.8.3.Transformada de Radon

La transformada de Radon es una técnica matemática utilizada en el procesamiento de imágenes médicas que mapea una función tridimensional en un conjunto de proyecciones bidimensionales. Esencialmente, permite visualizar la estructura interna de un objeto a partir de las mediciones de rayos X o de otro tipo de radiación que lo atraviesa desde diferentes ángulos. (Beatty, 2012).

En la reconstrucción tomográfica se presentan una serie de proyecciones medidas de 0 a 2π (aunque teóricamente sólo se necesitan proyecciones que cubran hasta π) en el senograma o transformada de Radon. Las proyecciones tomográficas pueden expresarse como la transformada de Radon del mapa de atenuaciones del objeto que se desea reconstruir, con lo cual el problema de la reconstrucción es equivalente a encontrar la inversa de la transformada de Radon. (García Ruíz & Reyes Mora, 2016)

La transformada de Radon $R\{f(x, y)\}$ de una función $f(x, y)$ se define como:

Ecuación 11. Transformada de Radon

$$p(r, \theta) = R\{f(x, y)\} = \int_{-\infty}^{\infty} f(r \cdot \cos \theta - s \cdot \sin \theta, r \cdot \sin \theta + s \cdot \cos \theta) ds$$

Fuente: (Ramírez Giraldo, Arboleda Clavijo, & H. McCollough, 2008)

donde r, s hacen parte de un nuevo sistema coordenado que es rotado un ángulo θ , como se ve en la ecuación 2.2.

Ecuación 12. Sistema de r y s

$$\begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Fuente: (Ramírez Giraldo, Arboleda Clavijo, & H. McCollough, 2008)

2.8.4.Paradigma de Visión de David Marr

El paradigma de visión de David Marr proporciona un marco teórico para la comprender cómo los seres humanos y las máquinas perciben el mundo visual que les rodea. Marr sugirió que la visión puede ser analizada en tres niveles distintos:

Nivel Computacional: Se centra en la función de calcular información tridimensional a partir de los datos bidimensionales que llegan a través de los ojos. En este nivel, se examina cómo el sistema visual procesa la información visual para crear una representación del mundo en tres dimensiones.

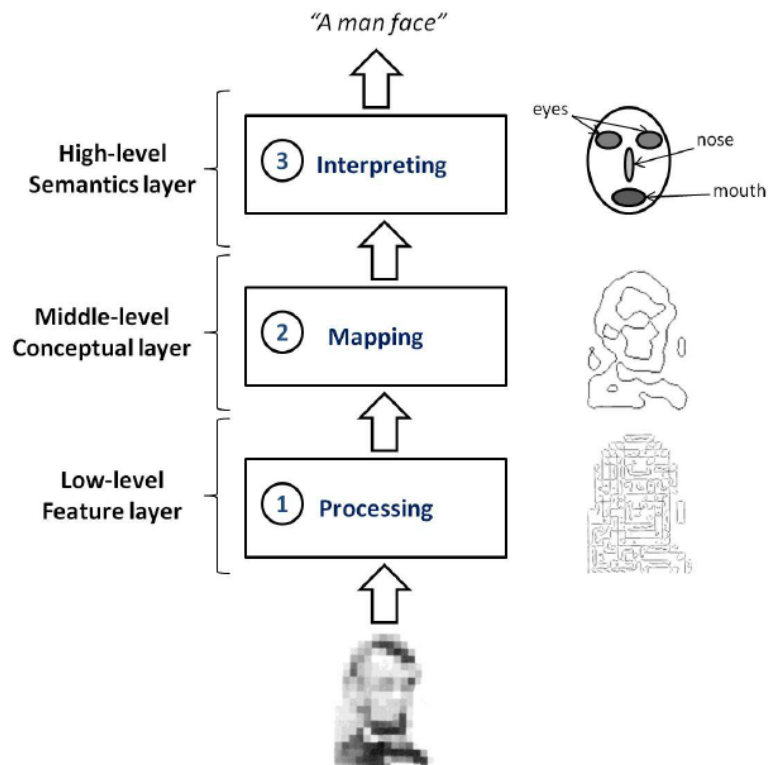
Nivel Representacional: Este nivel, se enfoca en la tarea de representar la información tridimensional de una manera que sea comprensible y útil para el sistema visual. Se trata de entender cómo se organizan y estructuran los datos para facilitar su interpretación y manipulación por parte del sistema visual.

Nivel de Tareas: El nivel de tareas aborda el uso práctico de la información representada para alcanzar objetivos específicos. Esto puede incluir la identificación de objetos, la navegación por un entorno o cualquier otra actividad que requiera el procesamiento visual para la toma de decisiones y la acción. (Marr, 2010)

Esto traducido a la visión por computador se expresa en la Figura 18. En el nivel bajo (Ver Figura 19) se detectas bordes y formas de los objetos presentes en los datos de entrada recopilados por los sensores.

Figura 18

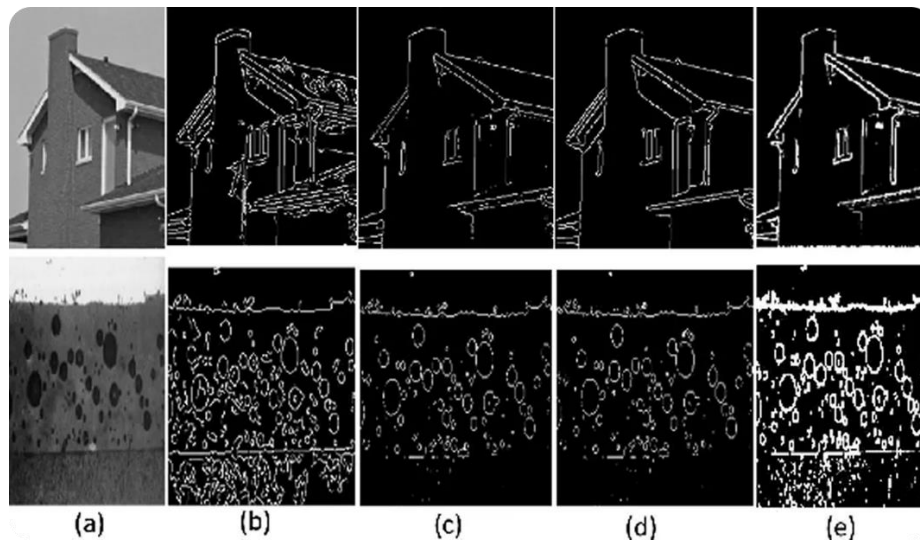
Descripción gráfica de los niveles propuestos por Marr partiendo de una imagen digital



Nota: Los tres niveles propuestos por Marr. *Fuente:* Basado en Marr de 1982 realizado por Pham (Pham, 2010).

Figura 19

Ejemplo de nivel bajo

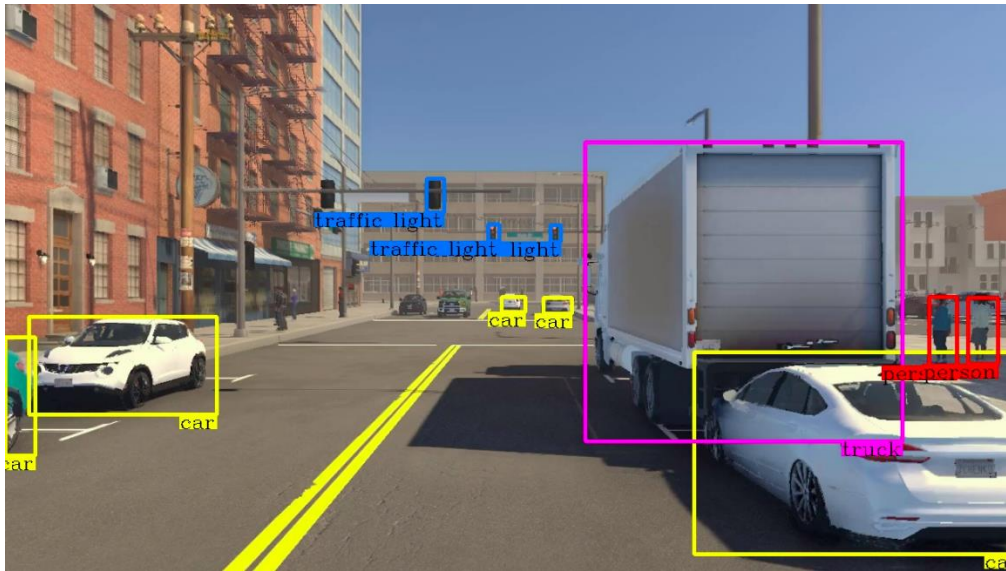


Nota: Ejemplo que muestra la imagen original junto con las salidas de diferentes métodos de detección de bordes. *Fuente:* (Dhar, 2016)

En el nivel medio la computadora ya es capaz de realizar el reconocimiento y clasificación de objetos, como se ve en la Figura 20.

Figura 20

Ejemplo de nivel medio



Nota: Detección de flechas y líneas. Fuente: (Sanjay, Maji, Harsh, & Washabaugh, 2020)

El nivel superior se relaciona con la interpretación del mundo, este es un nivel semántico que permite tomar decisiones y acciones, en la Figura 21 se hace referencia a la conducción autónoma, el auto analiza la señal de tránsito y reacciona según lo detectado.

Figura 21

Ejemplo de nivel superior



Nota: En este nivel, las computadoras ya logran entender las señales externas y tomar decisiones según esta información (Detección de defectos con visión 3D). Fuente: (Bcnvision, 2023)

2.9.HERRAMIENTAS DE SOFTWARE

En este apartado se explican las herramientas de software utilizadas para la realización del presente proyecto, desde lenguaje de código, bibliotecas y entornos. Al ser un prototipo experimental se utilizaron dos lenguajes principales, Python y Matlab. Primero se describe el lenguaje de programación Python y sus librerías.

2.9.1. Python

Python es un lenguaje de programación interpretado, de alto nivel y propósito general, ampliamente utilizado en el desarrollo de software. Destaca por su simplicidad y legibilidad, gracias a una sintaxis clara y sencilla que facilita su aprendizaje, especialmente para principiantes. Su diseño, cercano al lenguaje natural, permite que los programadores concentren su atención en la lógica del programa en lugar de lidiar con la complejidad de la sintaxis.

Aunque Python es un lenguaje orientado a objetos, también admite otros paradigmas como la programación procedural y funcional, lo que lo hace muy versátil y brinda a los desarrolladores la libertad de elegir el enfoque más adecuado para sus necesidades. Además, incluye una amplia biblioteca estándar que proporciona herramientas y módulos para realizar diversas tareas, reduciendo significativamente la necesidad de escribir código desde cero. Otro aspecto destacable es su sólida y activa comunidad, que ofrece numerosos recursos, tutoriales y documentación, facilitando el aprendizaje y el soporte técnico.

Python es un lenguaje generalista que se aplica a una amplia variedad de campos, como el desarrollo web, el análisis de datos, la inteligencia artificial, la automatización y muchos otros. Su capacidad para interactuar con otros lenguajes y plataformas lo convierte en una opción ideal para proyectos complejos. En el desarrollo web, frameworks como Django y Flask permiten construir aplicaciones robustas y eficientes. En la ciencia de datos, bibliotecas como Pandas, NumPy y Matplotlib son herramientas esenciales para el análisis y la visualización de información. Para el desarrollo de modelos de aprendizaje automático, frameworks como TensorFlow y PyTorch son ampliamente utilizados. Además, Python se usa frecuentemente para la automatización de tareas mediante scripts. (Toro Bonilla, 2022)

2.9.2. Open CV

OpenCV, una biblioteca de computación visual en Python provee herramientas para diversas operaciones de procesamiento de imágenes, incluyendo filtrado, detección de bordes, reconocimiento de características y seguimiento de objetos. Estas capacidades son fundamentales para el desarrollo de aplicaciones de visión artificial, como reconocimiento facial y seguimiento de objetos en tiempo real. Es una biblioteca de código abierto, es decir que podemos acceder al código fuente de OpenCV y modificarlo según sea necesario para adaptarlo a nuestras necesidades permitiendo una personalización. (Torrenti, 2024)

Su propósito fundamental es ofrecer una infraestructura común que simplifique el desarrollo de aplicaciones de visión por computadora y fomente su integración en productos comerciales. Esta biblioteca incluye más de 2,500 algoritmos optimizados que abarcan desde métodos tradicionales hasta técnicas avanzadas en el ámbito de la visión por computadora y el aprendizaje automático. Estos algoritmos permiten llevar a cabo diversas tareas, tales como la detección y reconocimiento de rostros, identificación de objetos, clasificación de acciones humanas en videos, seguimiento de movimientos de cámara, seguimiento de objetos en movimiento, extracción de modelos tridimensionales de objetos, generación de nubes de puntos 3D a partir de cámaras estéreo, creación de panoramas de alta resolución mediante la unión de imágenes, búsqueda de imágenes similares en bases de datos, eliminación de ojos rojos en fotografías, seguimiento de movimientos oculares, reconocimiento de escenas y establecimiento de marcadores para aplicaciones de realidad aumentada, entre otros.

OpenCV es compatible con varios lenguajes de programación, como C++, Python, Java y MATLAB, y es multiplataforma, operando en sistemas como Windows, Linux, macOS, iOS y Android. Además, está optimizada para aplicaciones en tiempo real, utilizando instrucciones MMX y SSE cuando están disponibles, y ofrece interfaces completas para CUDA y OpenCL, lo que permite su implementación en una amplia gama de dispositivos y arquitecturas de hardware.

Desde su lanzamiento en el año 2000, OpenCV ha sido adoptada de manera significativa tanto en el ámbito académico como en la industria, siendo utilizada en sectores como la seguridad, la medicina, la robótica móvil, la realidad aumentada y los vehículos autónomos. Su naturaleza de código abierto y su licencia Apache 2.0 facilitan su utilización y modificación en proyectos comerciales y de investigación. (Open CV, 2019)

2.9.3. PIL (Python Image Library)

La Python Image Library, conocida como PIL, es una librería de código abierto que facilita la edición y el procesamiento de imágenes mediante Python, brindando soporte a una amplia gama de formatos de imagen. Está optimizada para un acceso rápido a los datos y proporciona capacidades robustas para el procesamiento eficaz de imágenes. (Contributors, s.f.)

2.9.4. Numpy

Numpy es una librería de Python diseñada para manejar operaciones con grandes volúmenes de datos. Permite la representación de colecciones multidimensionales y se destaca por su eficiencia en la manipulación de datos, siendo el procesamiento de arreglos considerablemente más rápido en comparación con las listas tradicionales. (Sánchez, 2022)

El núcleo de NumPy es el objeto `ndarray`, una estructura que permite almacenar datos homogéneos en arrays multidimensionales. A diferencia de las listas estándar de Python, los arrays de NumPy son homogéneos, lo que significa que todos sus elementos tienen el mismo tipo y un tamaño fijo. Esta homogeneidad permite que las operaciones sean más eficientes y predecibles, optimizando el uso de recursos computacionales.

NumPy destaca por su rendimiento sobresaliente. Las operaciones realizadas en sus arrays son significativamente más rápidas que las ejecutadas con listas comunes, logrando eficiencias de procesamiento hasta 50 veces mayores. Esto resulta especialmente útil en aplicaciones que requieren cálculos intensivos, como el procesamiento de vectores y matrices. Además, la biblioteca ofrece una amplia gama de funciones matemáticas y estadísticas que permiten realizar sumas, multiplicaciones y transformaciones complejas sin necesidad de recurrir a bucles explícitos. Esto es posible gracias a la vectorización, una técnica que aplica operaciones a todos los elementos de un array de manera simultánea, mejorando notablemente el rendimiento.

Otra ventaja de esta librería es su capacidad de integración. Puede conectarse fácilmente con lenguajes como C, C++ y Fortran, lo que lo hace ideal para proyectos que demandan un rendimiento aún mayor. También trabaja en conjunto con otras bibliotecas populares como

Pandas, Matplotlib y SciPy, ampliando sus posibilidades y consolidándola como una herramienta clave para desarrolladores y científicos. (Sánchez Alberca, 2022)

2.9.5. Matplotlib

Matplotlib es una biblioteca completa para crear visualizaciones estáticas, animadas e interactivas en Python. Fue desarrollada por John Hunter (neurobiólogo) el 2002 con el objetivo de visualizar las señales eléctricas del cerebro de personas epilépticas, para ello deseaba replicar las funcionalidades de creación grafica de MATLAB en Python, es por este motivo que la sintaxis de esta biblioteca se asemeja a la sintaxis utilizada en el lenguaje de programación MATLAB. (Matplotlib development team, s.f.)

Uno de sus principales atributos es el control granular que permite a los desarrolladores ajustar meticulosamente cada elemento de la visualización, como tonalidades, estilos de líneas, ejes y legendas, garantizando así la elaboración de gráficos con un nivel de profesionalismo apto para publicaciones en cualquier contexto analítico o científico. Además de su notable capacidad gráfica, Matplotlib se distingue por su integración armoniosa dentro del entorno de Python. Se enlaza sin complicaciones con otras bibliotecas como NumPy y Pandas, facilitando la visualización directa de arreglos y DataFrames. (Aurora, 2024)

2.9.6. Jupyter Notebook

Jupyter Notebook es una herramienta de software libre y de código abierto que posibilita la creación de documentos que contienen código. Este software permite no solo escribir y ejecutar código sino también visualizar los resultados. Ofrece la capacidad de trabajar con bloques de código de manera individual y es compatible con Markdown. Jupyter Notebook se compone de una Aplicación Web basada en navegador y documentos de Notebook que presentan todo el contenido, incluyendo texto, imágenes y otros tipos de representaciones multimedia. (Jupyter, 2015)

2.9.7. Tkinter

Tkinter proporciona una interfaz orientada a objetos para el kit de herramientas Tcl/Tk, una capa GUI robusta e independiente de la plataforma. Esta arquitectura permite utilizar potentes componentes preconstruidos, conocidos como widgets, para construir los

elementos visuales de sus programas. Estos widgets incluyen controles GUI fundamentales como botones, etiquetas, campos de entrada, áreas de texto y barras de desplazamiento, así como elementos contenedores como marcos y lienzo para el diseño y el dibujo personalizado. Debido a su naturaleza multiplataforma, una aplicación Tkinter se ejecutará con un aspecto y un funcionamiento nativos en sistemas operativos populares como Windows, macOS y Linux.

El desarrollo de una aplicación Tkinter suele implicar dos pasos principales: crear y configurar los widgets y, a continuación, utilizar gestores de geometría para organizarlos dentro de la ventana de la aplicación. Tkinter ofrece tres gestores de geometría principales: `pack()`, que utiliza un diseño sencillo basado en bloques; `grid()`, que organiza los widgets en filas y columnas; y `place()`, que permite el posicionamiento absoluto utilizando coordenadas específicas. Es importante destacar que Tkinter es un marco de trabajo basado en eventos, es decir que el programa pasa la mayor parte del tiempo esperando las acciones del usuario (eventos). (Python Software Foundation, s.f.)

2.9.8.CustomTkinter

CustomTkinter es una librería de Python diseñada para el desarrollo de interfaces gráficas de escritorio con un estilo más contemporáneo y adaptable en comparación con el Tkinter convencional. Se basa en Tkinter, lo que significa que su uso es bastante similar, pero presenta widgets y ventanas que tienen un diseño más actual y uniforme en Windows, macOS y Linux. Su mayor beneficio es la facilidad para modificar colores, estilos y el aspecto visual general, incluyendo opciones de modo claro y oscuro. Además, es compatible con el escalado en pantallas de alta resolución, lo que mejora la apariencia de la interfaz en diferentes dispositivos. En la práctica, se utiliza para crear botones, etiquetas, campos de texto, interruptores y otros elementos de GUI sin necesidad de desarrollarlos desde el principio. Es una opción excelente si deseas que tus aplicaciones en Python tengan un aspecto más profesional sin perder la simplicidad que ofrece Tkinter. (Schimansky, s.f.)

2.9.9.Streamlit

Streamlit es una biblioteca basada en Python de código abierto, diseñada específicamente para ingenieros de machine learning, permite crear aplicaciones de aspecto impresionante

con sólo unas pocas líneas de código capaz de mostrar datos y recoger los parámetros necesarios para el modelado.

Uno de los aspectos importantes para que una aplicación tenga éxito es ofrecerla con una interfaz de usuario eficaz e intuitiva. Streamlit es la forma más fácil, especialmente para personas sin conocimientos de front-end, de poner su código en una aplicación web no exige experiencia ni conocimientos previos, no se requiere pasar días o meses para creando una aplicación web, es compatible con la mayoría de las bibliotecas de Python (por ejemplo, pandas, matplotlib, seaborn, plotly, Keras, PyTorch, SymPy(latex)), reduce la cantidad de líneas de código y el caché de datos simplifica los procesos y los aceleras. (DataCamp, 2024)

2.9.10. Matlab

Es una plataforma de cálculo que se emplea en aplicaciones de ingeniería y científicas, como el análisis de datos, la manipulación de imágenes y señales, los sistemas de control, las telecomunicaciones y la robótica. MATLAB comprende un lenguaje de programación, aplicaciones interactivas, librerías muy especializadas para usos en ingeniería y mecanismos para crear automáticamente código embebido. Simulink, un ambiente de diagramas de bloques para simular sistemas complejos multidominio, también tiene como base MATLAB. Se caracteriza por la convergencia de tres elementos clave:

Lenguaje de Programación de Alto Nivel: Un lenguaje optimizado para el cálculo numérico que reduce la complejidad técnica de la programación general.

Entorno Interactivo: Un espacio de desarrollo que permite experimentar con ideas, probar fórmulas y visualizar resultados de manera instantánea, facilitando el análisis de datos sin necesidad de construir infraestructuras de código desde cero.

Toolboxes (Cajas de Herramientas): Conjuntos de funciones especializadas que extienden las capacidades del programa hacia áreas específicas como la robótica, inteligencia artificial, finanzas computacionales y procesamiento de señales e imágenes.

Es ampliamente utilizado para el análisis de datos, el desarrollo de algoritmos y la creación de modelos matemáticos. Sus capacidades permiten no solo resolver ecuaciones y

manipular matrices, sino también realizar simulaciones detalladas de sistemas físicos en el tiempo, representando los resultados mediante gráficos avanzados en 2D y 3D. (The MathWorks Inc, s.f.)

2.10. Marco de trabajo

Las metodologías en el ámbito ingenieril son enfoques sistemáticos que guían la planificación, ejecución y control de proyectos. Estas metodologías son esenciales para asegurar que los proyectos se completen de manera eficiente, cumpliendo con los requisitos establecidos y dentro de los plazos y presupuestos asignados. A continuación, se presentan algunas de las metodologías más utilizadas por ingenieros. (Ruiz Huaraz, Valenzuela Muñoz, & Valenzuela Ramos, 2023)

2.10.1. Scrum

En el año 1986, Ikujiro Nonaka e Hirotaka Takeuchi publicaron el artículo "The New Product Development Game" el cual daría a conocer una nueva forma de gestionar proyectos en la que la agilidad, flexibilidad, y la incertidumbre son los elementos principales. (Atlassian, 2025)

Nonaka y Takeuchi analizaron cómo desarrollaban los nuevos productos las principales empresas de manufactura tecnológica: Fuji-Xerox, Canon, Honda, Nec, Epson, Brother, 3M y Hewlett-Packard que estando en el mismo entorno en el que se encontraban otras empresas, realizaban productos en menos tiempo, de buena calidad y menos costes. (Palacio, 2024)

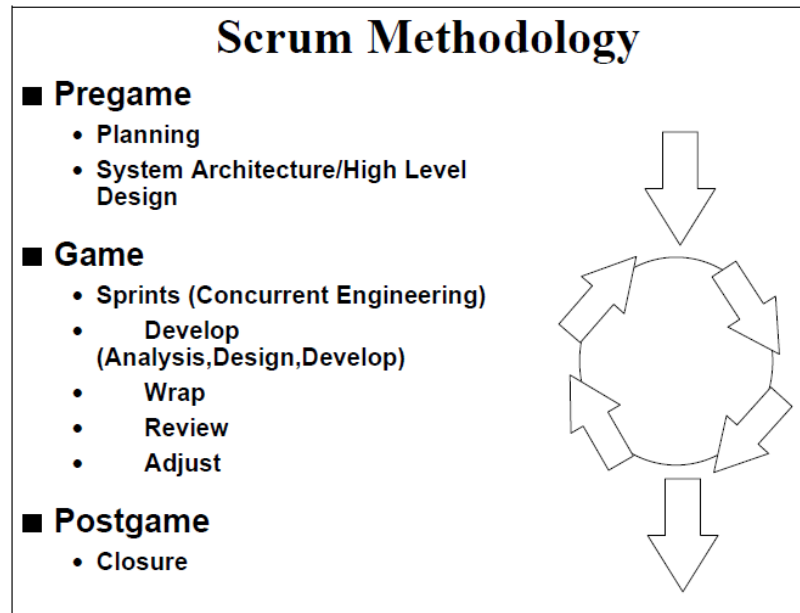
Observando a estas empresas notaron que el desarrollo del producto no seguía fases específicas con equipos especializados. En su lugar, comenzaban con requisitos amplios y un equipo multidisciplinar que se encargaba del producto desde el principio hasta el final. Esta forma de trabajo en equipo fue comparada con una formación ejecutada en el Rugby denominada Scrum, en la que ambos equipos, en cuclillas y sujetados entre sí, empujan para obtener el balón, y sacarlo de la formación sin tocarlo con la mano (Atlassian, 2025).

En el año 1995 Ken Schwaber presentó en OOPSLA (conferencia anual Object-Oriented Programming, Systems, Languages & Applications) una metodología de desarrollo de software basada en un ambiente Scrum el cual presentaba una serie de fases y artefactos

como: pregame, game, postgame, planning, sprints, wrap, etc (Ver Figura 2.9) Algunos se han mantenido hasta la actualidad y otros se fueron descartando (Palacio, 2024).

Figura 22

Esquema presentado por Ken Schwaber



Nota: Esquema presentado por Ken Schwaber Fuente: (Schwaber, 1996)

2.10.1.1.Ciclo scrum

El proceso comienza con una visión general del resultado deseado, detallando las funcionalidades prioritarias. Cada ciclo de desarrollo o iteración, conocido como sprint, finaliza con la entrega de una parte funcional del producto, llamada incremento. La duración de los sprints puede oscilar entre 1 y 3 semanas. Es común que mantengan una duración constante para establecer una cadencia, aunque pueden ajustarse según las necesidades (Palacio, 2024).

Scrum incluye diversos artefactos, roles y eventos. Los roles principales son el desarrollador, el propietario del producto y el Scrum Master. Los artefactos comprenden la pila del producto, la pila del sprint y el incremento. Los eventos incluyen el sprint, la reunión de planificación del sprint, el scrum diario, la revisión del sprint y la retrospectiva del sprint. (Palacio, 2024)

CAPITULO 3.

MARCO APLICATIVO

3.1.PLANIFICACIÓN

Se utilizará el marco de trabajo Scrum que permite entregar avances de valor de forma continua e incremental a lo largo del proyecto. Esto es especialmente útil en un proyecto de investigación, dado que se puede obtener resultados tangibles y retroalimentación temprana a medida que el proyecto se desarrolla. El proyecto tiene las siguientes fases: exploratoria, modelo matemático, modelo computacional y prototipo.

En la fase exploratoria se realiza un análisis de la información presente en el tramo a estudiar para posteriormente comparar la información con el resultado obtenido por la técnica de rebanado, también se examina el rendimiento de los algoritmos con el resultado final.

En la fase del modelo matemático se construye el modelo matemático del volumen espacio temporal contemplando dos enfoques matemáticos. Uno es cuantitativo, ya que se usan cantidades numéricas y el otro enfoque es cualitativo, que permite introducir estructuras algebraicas para describir las posiciones y relaciones en el espacio geométrico.

En la fase del modelo computacional se aplican las ideas descritas en el modelo matemático utilizando estructuras de datos, procesamiento de imágenes entre otros.

En la siguiente fase se desarrolla el prototipo aplicando el modelo computacional realizado en la fase previa, es el producto final. Para el desarrollo del prototipo se utilizará el marco de trabajo, Scrum, se revisan las historias de usuario, se definen los requerimientos para el producto final, se crea el product backlog y se realizan iteraciones.

A continuación, en la Tabla 1 se desglosa la planificación, contemplando las fases previamente mencionadas, los entregables de cada fase y la fecha conclusión.

El marco de trabajo SCRUM se aplica solo en la fase de desarrollo del prototipo por lo tanto en las fases previas son de carácter exploratorio, e investigativo, incluye análisis del estado actual del proceso de la empresa, aplicación de las teorías expuestas en el marco teórico, entre otras,

Tabla 1*Tabla de planificación*

FASE	TAREAS-ACTIVIDADES	ENTREGABLES
3.2 Exploratoria	3.2.1. Algoritmo de procesamiento de imágenes que permitan manipular la información en un volumen	Algoritmo realizado
	3.2.2. Revisión de la funcionalidad de los algoritmos previamente investigados	3.2.2.1. Resultados de los algoritmos investigados
3.3. Modelo Matemático	3.3.1. Geometría Euclidiana	Postulados de Euclides
	3.3.2. Matemática Cuantitativa	Elección de herramientas matemáticas a usar para la construcción de los elementos de la carretera
	3.3.3. Matemática Cualitativa	3.3.3.1. Estructuras algebraicas 3.3.3.2. Espacio vectorial 3.3.3.3 Espacio Euclidiano 3.3.3.4 Modelación de los Objetos 3.3.3.5 Modelación del Movimiento 3.3.3.6 Modelación del Sensor 3.3.3.7 Modelación del Rebanado
3.4. Modelo Computacional	Creación del modelo computacional que implemente la técnica	Pseudocódigo del algoritmo para llevarlo a un prototipo
3.5. Prototipo	3.5.1 Sprint Planning	Historias de usuario
	3.5.2. Producto backlog	Product backlog
	3.5.3. Sprint 1	Aplicación del algoritmo base y visualización de resultados
	3.5.4. Sprint 2	Prototipo con interfaz de usuario y desplegado como aplicación web
	3.5.5. Sprint 3	Análisis del rendimiento y optimización del algoritmo base
	3.5.6. Sprint 4	Prototipo final

Nota: Descripción del cronograma. *Fuente:* Elaboración propia

3.2. EXPLORATORIA

Durante esta etapa se realiza distintas acciones previas a la construcción del modelo matemático de manera formal, se explorarán algoritmos que permitan generar un volumen tridimensional, que planteen soluciones a la problemática y se evaluara sus resultados, también se realizara un análisis a los procesos actuales de la empresa. También se analizará la información del tramo que será estudiado específicamente de las señales de tránsito, tanto verticales como horizontales.

3.2.1. Análisis de los procesos actuales de supervisión vial en ECOVIANA S.R.L

El flujo de trabajo para el inicio de un proyecto de refacción en la empresa ECOVIANA comienza con un exhaustivo estudio de campo sobre el tramo vial a intervenir. Un equipo técnico se desplaza al terreno para registrar de forma manual variables críticas como el deterioro superficial de la carretera, la calidad del asfalto y el estado de la señalización vial. De manera complementaria, se realiza una grabación audiovisual continua de todo el trayecto desde un vehículo en movimiento. Este método tradicional genera un gran volumen de información dispersa que depende enteramente de la precisión del personal en el sitio, sirviendo el video como el único respaldo visual de las condiciones reales de la infraestructura.

Posteriormente, en la etapa de gabinete, el personal técnico debe realizar una validación cruzada para contrastar las anotaciones manuales con el registro audiovisual, lo que implica revisar minuciosamente grabaciones que suelen superar las cuatro horas de duración. Este procedimiento manual demanda una cantidad considerable de tiempo y recursos, además de ser propenso a errores por fatiga visual al inspeccionar el video cuadro por cuadro. Con el propósito de mitigar esta problemática y optimizar los tiempos de auditoría, la técnica de rebanado de video se introduce como una solución capaz de sintetizar el contenido completo de la grabación en una única imagen panorámica, facilitando una interpretación cronológica de todo el tramo vial.

3.2.2. Análisis del Algoritmo de Perspectiva

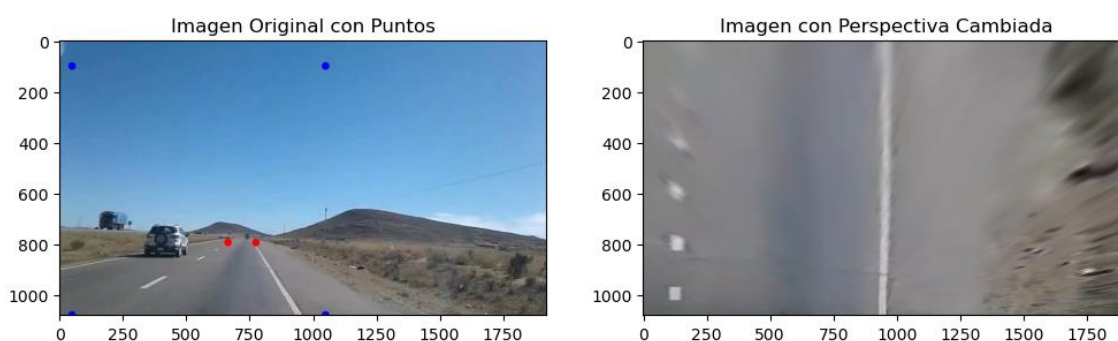
El algoritmo desarrollado ejecuta una transformación geométrica en cada cuadro (frame) del video original con el objetivo de normalizar la perspectiva visual, almacenar los cuadros modificados y, posteriormente, concatenarlos de forma secuencial para generar una imagen panorámica continua que facilite la detección de la señalización vial. Para corregir la distorsión de perspectiva, se requiere determinar una matriz de transformación de 3×3 , conocida como matriz de homografía. El cálculo de esta estructura matemática exige definir cuatro puntos de control en la imagen de origen y sus cuatro puntos correspondientes en el espacio de destino, bajo la condición de que ningún conjunto de tres puntos sea colineal. A nivel computacional, el flujo se implementó mediante la librería OpenCV, donde primero se utiliza la función `cv2.getPerspectiveTransform` empleando los

puntos de origen y destino, y una vez calculada la matriz, se aplica la función `cv2.warpPerspective` sobre el cuadro para proyectar la imagen corregida. Esta corrección es fundamental para eliminar el efecto de fuga visual propio de las cámaras en movimiento y estandarizar las dimensiones de la calzada.

Con el fin de calibrar el sistema, se realizaron pruebas experimentales para determinar las coordenadas espaciales óptimas que permitieran obtener el cambio de perspectiva deseado. En la Figura 23, se ilustra este proceso de calibración, donde los puntos rojos representan las coordenadas de origen en el espacio distorsionado y los puntos azules denotan la proyección de destino establecida. Tras validar la precisión de la transformación, se structured un algoritmo automatizado encargado de aplicar este cambio de perspectiva a cada cuadro del video de manera secuencial, exportando y guardando el archivo de video modificado directamente en el mismo directorio local. Este almacenamiento intermedio asegura que se mantenga un respaldo del video corregido antes de proceder con las etapas de síntesis gráfica. Finalmente, para realizar el rebanado y consolidar los cuadros procesados en una única representación visual continua, se recurrió a la manipulación matricial aprovechando que, a nivel informático, las imágenes digitales se representan como arreglos multidimensionales.

Figura 23

Resultado del código



Nota: Resultado de cambiar la perspectiva de un solo frame *Fuente:* Elaboración propia

Específicamente, se empleó la función `vstack` de la librería NumPy para concatenar de forma vertical los arreglos correspondientes a cada imagen. El algoritmo opera seleccionando la cantidad de cuadros que se desean integrar, definiéndose un límite de 10 frames como el umbral óptimo tras las pruebas de rendimiento. A medida que se realiza la

lectura del video, los cuadros se acumulan en una variable denominada frames, y una vez que el sistema verifica que se ha alcanzado la cantidad requerida, se ejecuta la función de concatenación y se almacena la imagen resultante en el directorio de trabajo. Mediante este control iterativo de búfer, el sistema procesa la información en bloques definidos, optimizando significativamente la gestión de memoria RAM durante la ejecución del programa.

Figura 24

Pseudocódigo para cambiar la perspectiva de un solo frame

- | | |
|--|---|
| <ol style="list-style-type: none"> 1. INICIO 2. Importar librerías necesarias: <ol style="list-style-type: none"> 2.1. OpenCV 2.2. numpy 2.3. matplotlib.pyplot 3. Cargar y preparar la imagen: <ol style="list-style-type: none"> 3.1. SI la imagen no se carga correctamente (es nula): <ol style="list-style-type: none"> 3.1.1. Mostrar mensaje de error: "Error al cargar la imagen." 3.1.2. Terminar la ejecución. 3.2. FIN SI 3.3. Convertir la imagen de espacio de color BGR a RGB para su correcta visualización. 3.4. Obtener las dimensiones de la imagen (altura h y ancho w). 4. Configurar los puntos para la transformación: <ol style="list-style-type: none"> 4.1. Definir el arreglo de puntos de origen (src). 4.2. Definir el arreglo de puntos de destino (dst). 5. Dibujar puntos de referencia en la imagen original: <ol style="list-style-type: none"> 5.1. PARA CADA punto en el arreglo de origen (src): <ol style="list-style-type: none"> 5.1.1. Convertir las coordenadas del punto a valores enteros. 5.1.2. Dibujar un círculo de color rojo (relleno, radio 15) en la imagen original RGB. 5.2. FIN PARA | <ol style="list-style-type: none"> 6. Procesar la transformación de perspectiva: <ol style="list-style-type: none"> 6.1. Calcular la matriz de transformación (M) utilizando los puntos de origen (src) y destino (dst). 6.2. Aplicar la transformación de perspectiva a la imagen original usando la matriz M y las dimensiones (w, h). 6.3. Convertir la imagen resultante (transformada) de BGR a RGB. 7. Dibujar puntos de referencia en la imagen transformada: <ol style="list-style-type: none"> 7.1. PARA CADA punto en el arreglo de destino (dst): <ol style="list-style-type: none"> 7.1.1. Convertir las coordenadas del punto a valores enteros. 7.1.2. Dibujar un círculo de color azul (relleno, radio 15) en la imagen transformada RGB. 7.2. FIN PARA 8. Desplegar la interfaz gráfica con los resultados con el título: "Imagen con Perspectiva Cambiada". <ol style="list-style-type: none"> 8.1. Mostrar la ventana gráfica en pantalla. 9. FIN |
|--|---|

Nota: Código realizado en Python *Fuente:* Elaboración propia

Figura 25

Código para cambiar la perspectiva de todos los frames de un video y almacenar el resultado

```

1 import cv2
2 import numpy as np
3
4 # Ruta del video de entrada y salida
5 input_video_path = 'videoLaPaz-Llujturi_LIMPIO.mp4'
6 output_video_path = 'videoLaPaz-Llujturi_Limpio_Perspectiva.mp4'
7
8 # Abrir el video de entrada
9 cap = cv2.VideoCapture(input_video_path)
10
11 # Obtener las propiedades del video
12 frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
13 frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
14 fps = int(cap.get(cv2.CAP_PROP_FPS))
15
16 # Definir el codec y crear el objeto VideoWriter para el video de salida
17 fourcc = cv2.VideoWriter_fourcc(*'mp4v')
18 out = cv2.VideoWriter(output_video_path, fourcc, fps, (frame_width, frame_height))
19
20 # Puntos de referencia (ajusta estos puntos según tus necesidades)
21 h, w = frame_height, frame_width
22 # Puntos de referencia
23
24 src = np.float32([[50, h],
25                  [1000+50, h],
26                  [460+200+5, 700+150-55],
27                  [535+200+40, 700+150-55]])
28
29 # Puntos de destino
30 dst = np.float32([[50, h],
31                  [1000+50, h],
32                  [50, 100],
33                  [1000+50, 100]])
34
35 # Obtener la matriz de transformación
36 M = cv2.getPerspectiveTransform(src, dst)
37
38 while cap.isOpened():
39     ret, frame = cap.read()
40     if not ret:
41         break
42
43     # Aplicar la transformación de perspectiva
44     imgPerspectiva = cv2.warpPerspective(frame, M, (w, h), flags=cv2.INTER_LINEAR)
45
46     # Escribir el frame transformado en el video de salida
47     out.write(imgPerspectiva)
48
49 # Liberar el objeto VideoCapture y VideoWriter
50 cap.release()
51 out.release()
52
53 # Cerrar todas las ventanas abiertas de OpenCV (si las hay)
54 cv2.destroyAllWindows()
55

```

Nota: Código para cambiar la perspectiva de todos los frames de un video y almacenar el resultado en un nuevo video *Fuente:* Elaboración propia

Figura 26

Código para concatenar n frames de un video

```

1 import cv2
2 import numpy as np
3
4 # Ruta del video
5 video_path = 'videoLaPaz-Llujturi_Limpio_Perspectiva.mp4'
6
7 # Abrir el video
8 cap = cv2.VideoCapture(video_path)
9
10 frames = []
11 frame_count = 0
12
13 while frame_count < 15:
14     ret, frame = cap.read()
15     if not ret:
16         break
17     frames.append(frame)
18     frame_count += 1
19
20 # Liberar el video
21 cap.release()
22
23 # Asegurarse de que tenemos 15 frames
24 if len(frames) < 10:
25     print("El video no tiene suficientes frames.")
26 else:
27     # Concatenar los frames horizontalmente
28     concatenated_image = np.vstack(frames)
29
30     # Guardar la imagen resultante
31     cv2.imwrite('frames_concatenados.jpg', concatenated_image)
32
33     print("Frames concatenados y guardados en 'frames_concatenados.jpg'")
34

```

Nota: Código para concatenar n frames de un video *Fuente:* Elaboración propia

3.2.3.Revisión de la funcionalidad del algoritmo

A continuación, se analizarán los resultados obtenidos por el algoritmo utilizando el video del tramo vial La Paz – Llujturi

3.2.3.1.Resultados del algoritmo

Los resultados obtenidos evidencian limitaciones en la capacidad del algoritmo para reconstruir la totalidad de la infraestructura vial. Si bien el sistema logra discernir la línea segmentada blanca, la línea continua presenta distorsiones significativas en su geometría. Asimismo, la señalización vertical no es recuperada por el proceso y la línea continua del extremo izquierdo resulta imperceptible en la imagen final, determinando que el rendimiento del algoritmo actual sea subóptimo para este escenario.

Figura 27

Resultado final de la concatenación.



Nota: Prueba del código con 15 frames *Fuente:* Elaboración propia

3.3.MODELO MATEMÁTICO

La infraestructura vial está compuesta por cuerpos físicos como ser el asfalto, señales verticales, señales horizontales, señales de tipo informativo, etc. Estos cuerpos se encuentran desplegados en un espacio (mundo). Modelar matemáticamente el mundo y los elementos o cuerpos de la estructura vial es importante porque se constituye en un lenguaje formal para la comunicación interdisciplinaria con los ingenieros de transporte.

El modelado matemático del espacio donde ocurre el escenario vial, en el presente proyecto, utiliza dos enfoques matemáticos. Uno es cuantitativo, ya que se usan cantidades

numéricas. El otro enfoque es cualitativo, que permite introducir estructuras algebraicas para describir el espacio geométrico de la escena vial. Por ejemplo, es preferible usar el concepto de vector posición en lugar de ternas de coordenadas cartesianas. Las ternas cartesianas, serán útiles cuando se represente computacionalmente el vector posición.

También se debe realizar los siguientes modelos: uno que describa el movimiento relativo de los objetos en la escena vial dentro del espacio geométrico; otro que represente al sensor y su desplazamiento a lo largo de dicha escena; un tercero que modele la salida del sensor, es decir, las imágenes planas donde se proyecta la escena tridimensional; y, por último, modelo matemático que caracterice el volumen espacio-temporal captado por la cámara.

Teniendo el volumen espacio temporal se puede definir varias estrategias de rebanado. A continuación, se procede a describir los componentes señalados.

3.3.1.Geometría Euclidiana

La geometría euclidiana es la base para el desarrollo del modelo matemático, el análisis de la carretera se hace con espacio euclidianos, como se menciona en el apartado 2.3. existen varios tipos de geometría, pero se decidió tomar la geometría euclidiana por que se estima que puede tener mejores resultados, además el video base representa un plano, y la descripción de los objetos se dificultaría si se tomara una geometría de curvas, por ejemplo.

3.3.2.Matemática Cuantitativa

A nivel cuantitativo se tomarán en cuenta los números naturales, enteros, racionales y reales (composición descrita en el punto 2.4.1) para la construcción de los distintos elementos de la carretera dentro de los espacios euclidianos.

3.3.3.Matemática Cualitativa

Las primeras abstracciones de la realidad sensible han sido los números. La matemática moderna se caracteriza por la abstracción de abstracciones que deriva en la construcción de estructuras algebraicas.

3.3.3.1.Estructuras Algebraicas

Es importante definir las estructuras algebraicas utilizadas en el modelo matemático, ya que existen diversas geometrías con condiciones específicas. Por lo tanto, para modelar un

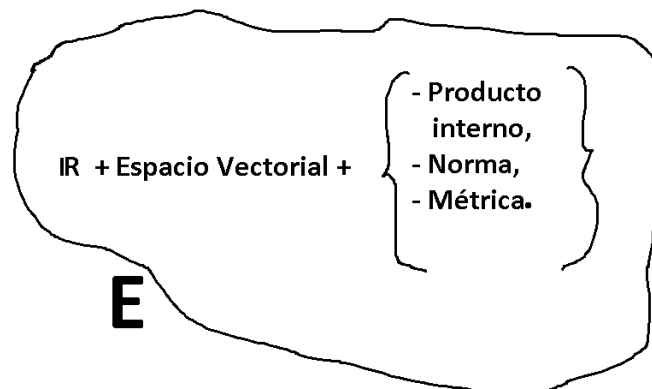
evento dentro de nuestro espacio tangible, es fundamental identificar estas estructuras y determinar las operaciones que se pueden realizar con ellas.

3.3.3.2.Espacio Euclidiano Para el Modelo.

El espacio vectorial es un objeto matemático que describe bien los puntos, segmentos y rectas de la geometría euclidiana. Sin embargo, en esta geometría hay otras categorías como ser: distancia entre puntos, tamaño de los segmentos, circunferencias y ángulos entre rectas. Estas categorías requieren de la introducción de una nueva operación el producto interno o producto escalar (procesos detallados en la página 20).El espacio vectorial de flechas más el producto escalar modelan la geometría euclidiana requerida para el presente proyecto.

Figura 28

Modelo matemático del espacio



Nota: El modelo matemático del espacio se realiza con un espacio vectorial real y un producto interno (escalar) Fuente: Elaboración propia

3.3.3.3.Modelación de los Objetos del Escenario Vial.

Un escenario vial está compuesto de varios objetos o cuerpos como ser: Calzada, señales verticales, señales horizontales, etc. Estos cuerpos se modelan como una lista abstracta de las partículas que lo componen, como se muestra en la Figura 29, donde $y = \chi(p)$ denota el punto y que ocupa la partícula p en el espacio euclidiano. La función $\chi(p)$ es una función biunívoca, entre la región $\mathcal{R}$ y la lista de partículas $\mathcal{B}$, de tal forma que dos partículas no pueden ocupar el mismo punto y a cada punto y de la región le corresponde una partícula $p = \chi^{-1}(y)$. Además, si está dado un origen O , la posición $\chi(p)$ se modela como un vector-posición que parte del origen O y llega al punto $\chi(p)$.

En el espacio euclidiano, una partícula p del cuerpo ocupa el punto $y = \chi(p)$. Esta función es inyectiva, es decir que dos partículas no pueden ocupar el mismo punto. Si está dado un origen, la posición $\chi(p)$ se modela con un vector posición que parte del origen y llega al punto $\chi(p)$.

3.3.3.4. Modelación del Movimiento dentro del Escenario Vial

Cuando los cuerpos se mueven en el mundo, ocupan distintas regiones del espacio euclidiano. El automóvil que tiene la cámara, por ejemplo, se mueve por la carretera, ocupando en cada momento una región diferente del espacio. En la Figura 30 se puede apreciar dos regiones (o configuraciones) $\mathcal{R}_0$ que es la región inicial y $\mathcal{R}_t$ que representa la región que ocupan las partículas en un tiempo t que va de un tiempo inicial t_0 a un tiempo final t_1 . Como antes, $\mathcal{B}$ es la lista de partículas, en la figura, la partícula p , ocupa una posición $y = \chi_t(p)$ en el tiempo t . Sin embargo, también se tiene que $y = \hat{y}^t(x)$, donde x es la posición que ocupaba la partícula p en la región de referencia $\mathcal{R}_0$.

Cuando termina el movimiento, es decir $t = t_1$, u denota el vector desplazamiento neto de la posición x . Las posiciones se construyen respecto a un sistema de referencia con origen O y los versores de dirección e_i .

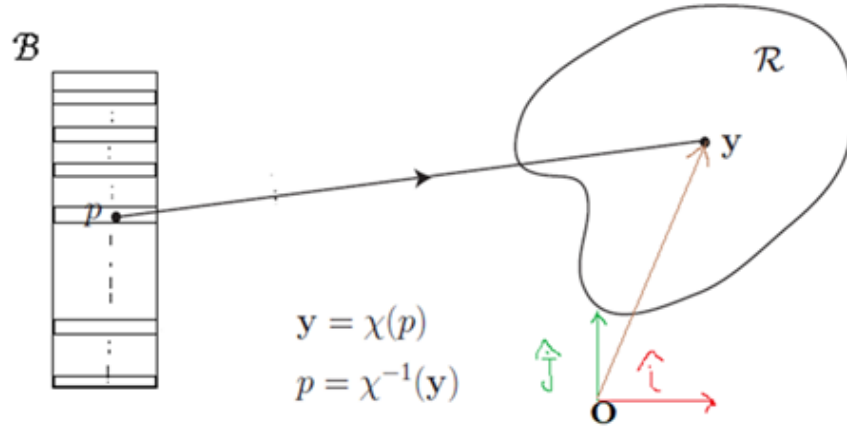
3.3.3.5. Modelación del Sensor

Una vez definido el espacio geométrico que contiene los elementos del escenario vial, ahora es posible modelar un sensor que se mueve por la escena. Dentro del contexto del presente trabajo, el sensor es una cámara que capta imágenes bidimensionales a través del tiempo. El plano de la cámara se considera un espacio euclídeo bidimensional, sobre el cual se definen las perspectivas y proyecciones provenientes del espacio tridimensional. La rama de la matemática que estudia el reflejo del mundo real en un plano bidimensional se denomina geometría proyectiva, la cual provee el marco teórico necesario para describir las distorsiones ópticas generadas por el lente. Nótese que las líneas paralelas pintadas sobre la carretera convergen visualmente a un punto; este punto se denomina punto de fuga o del infinito. El concepto de líneas paralelas se sustituye por el concepto de líneas que se encuentran en un punto situado en el infinito, pero que en realidad este punto se ubica físicamente en el plano focal de la cámara. Dicha proyección en el plano focal resulta matemática y computacionalmente indispensable, dado que permite establecer las

ecuaciones de homografía requeridas para revertir la distorsión de la perspectiva y reconstruir fielmente la geometría de la vía.

Figura 29

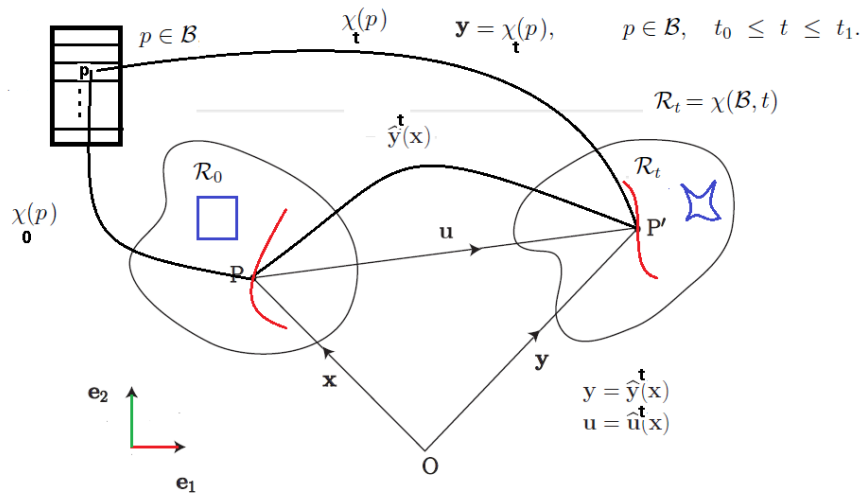
Descripción del modelo dentro del contexto del escenario vial



Nota: Los elementos que componen un escenario vial se modelan como una lista abstracta de partículas que se vuelven accesibles a través de la región que ocupan en el espacio euclidiano. Esta región $\mathcal{R}$ se llama configuración. *Fuente:* Elaboración propia

Figura 30

Descripción de la modelación del movimiento



Nota: Durante un intervalo de tiempo $[t_0, t_1]$, el cuerpo, al moverse, ocupa diferentes regiones $\mathcal{R}_t = \chi(\mathcal{B}, t)$ del espacio. Se puede modelar el movimiento desde la perspectiva de la partícula p o desde la perspectiva del punto geométrico x que ocupa. *Fuente:* Elaboración propia

Si establecemos un punto de origen en el espacio 3D de la carretera, el escenario vial se muestra como una escena estática, es decir, las posiciones de las señales horizontales y verticales no cambian respecto a este punto de origen. Sin embargo, si consideramos el origen ubicado en el punto focal de la cámara, la escena respecto al sensor se mueve, las posiciones de las señales van cambiando con relación a este punto. Esto hace que, en cada momento del tiempo, los sensores de luz de la cámara proyecten en un plano diferentes intensidades que se modelan con una función $f(x, y; t)$ de dos variables (x, y) parametrizada por el tiempo t . Los valores concretos que toma la función $f(x, y; t)$ en una escena constituye un volumen espacio temporal.

El sensor se modela como una máquina (Ecuación 13)

Ecuación 13 Modelo del sensor

$$\Sigma = (T, Q, U, \phi, Y, h, \lambda, \hat{\lambda}).$$

que consiste de:

Un conjunto de tiempos T ; Los fenómenos que se están estudiando, pueden ocurrir en tiempo continuo o en tiempo discreto. Se exige que el conjunto de tiempos sea cuantitativo, con orden total (para construir las secuencias). También podrían tener una estructura algebraica de grupo.

Un conjunto no vacío Q , llamado espacio de estados de Σ ; Con este conjunto se pretende modelar los estados internos de la cámara, el principal si la cámara está encendida.

Un conjunto no vacío U , llamado espacio de valores entrada de Σ ; Este conjunto representa los objetos presentes en el escenario vial, estos objetos generan patrones de luz. Por lo tanto se puede considerar que el conjunto U contiene como elementos estos patrones de luz.

En el presente trabajo, no se entra en más detalle sobre la naturaleza de estos patrones porque el interés se centra no en la entrada sino en la salida del sensor. Sólo, es necesario recalcar que la entrada al sensor tiene que ver con los objetos presentes en el escenario vial.

Un mapeo de transiciones de estados $\phi: D_\phi \mapsto Q$, el cual se explica detalladamente (Ver Ecuación 14) Esta función describe los posibles cambios de estado de la cámara (mayor apertura, tiempo de obturación, etc, en función al estado en que se encuentra y al streaming

recibido). Algunas cámaras pueden tener estas funcionalidades de forma automática. La cámara puede cambiar de estados dependiendo de las características de lo que está capturando que está recibiendo.

Ecuación 14. Dominio del mapeo de transiciones.

$$\{(\tau, \sigma, x, \omega) | \sigma, \tau \in T, \sigma \leq \tau, x \in Q, \omega \in U^{[\sigma, \tau)}\};$$

Fuente: Elaboración Propia

El mapeo de transición de estados ϕ modela estos cambios de estado. La cámara, al ser un sistema físico, tiene restricciones físicas. Estas restricciones se modelan con el conjunto D_ϕ . Por ejemplo, los tiempos σ y τ no pueden ser tales que superen el límite físico de la máquina. Por lo tanto, las características físicas de cada cámara en particular definen el dominio.

Hablamos de un dominio, porque puede haber cosas que la cámara no distingue. Sólo pertenece al dominio aquello que la cámara distingue.

Un conjunto Y llamado espacio de valores de salida de Σ ; representa el conjunto de imágenes bidimensionales está caracterizado, entre otras cosas, por su altura y su ancho.

La cámara, en un momento dado del tiempo, genera una imagen bidimensional con ciertas características, el conjunto de estas se represente con Y . Computacionalmente, una imagen se puede caracterizar por su resolución, por su altura, por su ancho, el tamaño del archivo, el formato del archivo (JPEG, PNG, BMP, etc), la profundidad del color, metadatos, etc.

Un mapeo $h: T \times Q \mapsto Y$, llamado mapeo de medidas; Este mapeo modela el hecho de que la imagen digital obtenida depende del momento en que se capturó y de las características de la cámara. Computacionalmente, es una función que permite recuperar la imagen capturada en un momento del tiempo.

Un mapeo $\lambda: D_\lambda \mapsto Y$, definido sobre el dominio:

$$\{(\tau, \sigma, \omega) | \sigma, \tau \in T, \sigma \leq \tau, \omega \in U^{[\sigma, \tau)}\};$$

Este mapeo modela la posibilidad de obtener una imagen capturada al final de un tiempo. Si el tiempo sigue cambiando, la imagen corresponde al último tiempo. Computacionalmente, esta función puede modelar la captura en tiempo real.

Un mapeo $\hat{\lambda}: D_{\hat{\lambda}} \mapsto \bigcup_{\sigma \leq \tau} Y^{[\sigma, \tau]}$, definido sobre el dominio:

$$\{(\tau, \sigma, \omega) | \sigma, \tau \in T, \sigma \leq \tau, \omega \in U^{[\sigma, \tau]}\},$$

donde:

$$\hat{\lambda}(\tau, \sigma, \omega)(t) := \lambda(t, \sigma, \omega|_{[\sigma, \tau]})$$

para cada $t \in [\sigma, \tau]$. Este mapeo, a diferencia del anterior, modela la captura de la escena y su almacenamiento en formato de video. El conjunto $\bigcup_{\sigma \leq \tau} Y^{[\sigma, \tau]}$ contiene videos de diferente duraci3n, incluido el de duraci3n cero. A los elementos de este conjunto se los designa como el volumen espacio temporal. Este volumen espacio temporal es el que puede ser rebanado.

En el presente trabajo, se ha dedicado esfuerzo en la modelaci3n del sensor para establecer como las caracterfsticas de la captura de datos pueden determinar la calidad del video obtenido. Representa un marco formal para construir un lenguaje com3n con los ingenieros del 3rea de transporte.

3.3.3.6. Modelaci3n del Rebanado

En sntesis, la t3cnica de rebanado consiste de los siguientes pasos:

Se empieza con un elemento $v \in \bigcup_{\sigma \leq \tau} Y^{[\sigma, \tau]}$, que fue capturado por el sensor y representa lo que se conoce como *volumen espacio temporal*. Computacionalmente, v es un video. Es decir, es un flujo de im3genes bidimensionales (stream) que tiene varias formas de almacenar y desplegar videos en una computadora.

El segundo paso es seleccionar la direcci3n $\vec{d}$ del rebanado del volumen espacio temporal para producir una imagen bidimensional. En este sentido, rebanar es obtener una restricci3n $v|_{\vec{d}}$ de v , en las direcciones deseadas. Las direcciones m3s 3tiles, en la presente aplicaci3n, son las horizontales y verticales.

Posteriormente, una vez obtenido el rebanado $r = v|_{\vec{d}}$, se puede aumenta la informaci3n, realizando anotaciones sobre la imagen. En el presenta trabajo, se considera que las anotaciones a realizar sobre la imagen provienen de un procesamiento inteligente que genera un data frame que contiene coordenadas e informaci3n textual, el cual se usa como insumo para a3adir texto a la imagen del rebanado.

En matemáticas, una función restringida es una función que se limita a un subconjunto particular del dominio original. En otras palabras, es la misma función, pero se considera solamente en un intervalo o subconjunto específico de los valores que originalmente podía tomar. Formalmente, si se tiene una función $f: A \subseteq \mathbb{R}^3 \rightarrow \mathbb{R}$, que representa el volumen espacio temporal $f(x, y; t)$, una rebana es la función restringida que se define solo en un subconjunto $B \subseteq A$, con la misma fórmula, pero limitada a los puntos de B . Esto significa que la nueva función $f|_B$ solo toma valores del subconjunto B . La forma que se tome B define las dimensiones de la restricción. Si B es un solo punto la dimensión de salida será cero, si B es un segmento de recta, el resultado será unidimensional. Para obtener los diferentes cortes o rebanados en 2 dimensiones, se pueden construir restricciones del del volumen espacio temporal, por ejemplo, manteniendo un valor fijo para una de las variables (x, y) y variando la otra, en este caso, se realiza un corte en profundidad, ya sea horizontal o vertical.

3.4.MODELO COMPUTACIONAL

El modelo computacional del sensor permite pensar en la posibilidad de construcción de la sensorización híbrida que consiste en sensores físicos potenciados con algoritmos de software que potencian la captura (sensores virtuales). Por ejemplo, las cámaras pueden estar sujetas a vibraciones que pueden ser corregidas por software. El modelo matemático del sensor se puede realizar mediante una sensorización híbrida. Muchas de las características de la cámara que exige el modelo matemático se pueden resolver a través de código que analice los datos capturados para tomar decisiones de cambiar los valores de los componentes para generar imágenes más nítidas.

Computacionalmente, los sensores híbridos se pueden modelar utilizando instancias de clases en lenguajes orientados a objetos. Las clases Camera, Sensor, Video, Imagen son comunes en algunos lenguajes. En el presente trabajo se usó la clase VideoCapture, que es una parte del módulo cv2 (OpenCV) en Python, y que se utiliza para capturar video desde una cámara o un archivo de video. La clase VideoCapture permite configurar la cámara, como la resolución, el framerate, el formato de video, etc. También permite leer frames individuales del video capturado o del archivo de video.

Un elemento $v \in \bigcup_{\sigma \leq \tau} Y^{[\sigma, \tau]}$ es un video y entra como el parámetro filename de la clase VideoCapture. El método read() modela computacionalmente al mapeo $h: T \times Q \mapsto Y$ de

medidas Recuperando todas las imágenes, se construye un arreglo numpy que contiene las imágenes con su profundidad de color; esto representa al volumen espacio-temporal que en el código se denota con `frame_array`. El núcleo de NumPy es el objeto `ndarray`, una estructura que permite almacenar datos homogéneos en arrays multidimensionales. A diferencia de las listas estándar de Python, los arrays de NumPy son homogéneos, lo que significa que todos sus elementos tienen el mismo tipo y un tamaño fijo. Esta homogeneidad permite que las operaciones sean más eficientes y predecibles, optimizando el uso de recursos computacionales.

La dirección del rebanado $\vec{d}$, en nuestro caso, se limita a direcciones horizontales y verticales. Para establecerla, se requiere indicar los valores de cada uno de los índices del `frame_array`, por ejemplo: `(1100:299:-1, 880, 0:100, :)`; que codifica una dirección de rebanada horizontal a la altura 880.

Una vez obtenida la dirección del rebanado, se genera una imagen 2D que es la restricción $r = v|_{\vec{d}}$ del volumen espacio-temporal en esa dirección. Esto se realiza con el comando: `slice_colors = frames_array[1100:299:-1, 880, 0:1000, :]`. Los métodos que ofrece la clase `VideoCapture` pueden ser complementados para darle mayor versatilidad al sensor híbrido.

Primero se lee el video en el entorno, a cada frame se le cambia el modelo de color a de BGR a RGB (ya que la clase `VideoCapture` lee los videos en formato BGR). Los frames son almacenados en un arreglo que recupera el alto, el ancho, el número de frame y los canales de color. Ej.: Forma de `frames_array`: `(1080, 1920, 1124)` o `(1080, 1920, 1124,3)`. Todo el procedimiento pertenece al paso 2 del pseudocódigo.

Como se estableció en el modelo matemático de la rebanado, debemos mantener alguno de estos valores estáticos para obtener las rebanadas, en el caso de las señales horizontales, se mantiene el valor correspondiente al alto del frame en un número estático, el valor del ancho y el número de frame varía en el tiempo, lo cual es representado con series (una lista de valores desde una posición inicial a una posición final). Después de evaluar los mejores valores para cada variable, se muestra el resultado obtenido en la rebanada. Este paso es el número 3 dentro del pseudocódigo.

Luego el paso 4 del pseudocódigo hace referencia a la posibilidad de añadir texto para complementar la imagen de salida, haciéndola más legible y entendible para todo público.

Figura 31

Pseudocódigo del modelo computacional

- 1. INICIO**
- 2. Cargar el video y extraer frames**
 - 2.1. Definir la ruta del video.
 - 2.2. Abrir el archivo de video.
 - 2.3. Inicializar una lista vacía para almacenar los cuadros.
 - 2.4. Mientras haya cuadros disponibles en el video:
 - Leer un cuadro del video.
 - Si no hay más cuadros, salir del bucle.
 - Convertir el formato de color del cuadro según sea necesario.
 - Agregar el cuadro procesado a la lista.
 - 2.5. Liberar los recursos del archivo de video.
 - 2.6. Convertir la lista de cuadros a una estructura de datos multidimensional.
- 3. Realizar una selección de cuadros**
 - 3.1. Seleccionar un subconjunto de cuadros de la estructura de datos según:
 - Rango de índices de cuadros (en orden inverso).
 - Porción específica de altura.
 - Rango limitado del ancho.
 - Todas las capas de color disponibles.
 - 3.2. Mostrar la selección obtenida.
- 4. Agregar texto a una imagen seleccionada**
 - 4.1. Usar la selección realizada como base para agregar texto.
 - 4.2. Configurar parámetros del texto, incluyendo:
 - Estilo de fuente.
 - Contenido del texto.
 - Posición en la imagen.
 - Tamaño, color y grosor del texto.
 - 4.3. Agregar múltiples textos en posiciones específicas:
 - En direcciones horizontales, incluir contenido relacionado.
 - En direcciones verticales, agregar etiquetas o identificadores.
 - 4.4. Mostrar la imagen modificada.
- 5. FIN**

Nota: Descripción del algoritmo utilizado en el modelo computacional. *Fuente:* Elaboración Propia

3.5.PROTOTIPO

Para la creación del prototipo de software aplicando la técnica de rebanado se utiliza las herramientas del marco de trabajo Scrum En primer lugar, se define el equipo Scrum. A pesar de que este proyecto se realiza de forma individual, el tutor y la empresa supervisan el avance periódicamente. En la Tabla 2 se detallan los miembros del equipo Scrum y sus respectivas funciones.

Tabla 2

Descripción del equipo Scrum.

Rol	Responsable	Funciones
Desarrollador	Rosalía Reynaga	Realizar las tareas del proyecto, gestionar el backlog del sprint.
Product Owner	ECOVIANA S.R.L	Proporcionar datos capturados comunicar con claridad los requisitos y feedback sobre el trabajo realizado.
Scrum Master	Tutor	Facilitar reuniones, eliminar obstáculos, mejorar procesos.

Nota: Descripción del equipo Scrum. Fuente: Elaboración propia

3.5.1.Sprint Planning

En el sprint planning se realizaron dos entregables, la elaboración de las historias de usuario y el producto backlog

3.5.1.1.Historias de usuario

Para la elaboración de historias de usuario se optó por un modelo simple destacando las funcionalidades más importantes del sistema para después desglosarlo en tareas más pequeñas. También se recopilaron requerimientos de la empresa.

Se realizaron 4 historias de usuario, se asignó un número de historia de usuario y se utilizaron las preguntas como, quiero y para, para conseguir requerimientos concretos y tener la guía inicial para empezar a realizar el prototipo funcional, este proceso se detalla en Tabla 3

Tabla 3

Historias de usuario

N° de historia	Como	Quiero	Para
HU-1	Usuario	Que el prototipo procese los datos (videos) y muestre el resultado del corte o slice	Comprobar el funcionamiento basico
HU-2	Usuario	Interactuar con el prototipo mediante una interfaz grafica	Cargar datos y ver resultados sin ejecutar código
HU-3	Desarrollador	Optimizar los cálculos en MATLAB	Reducir el uso de memoria
HU-4	Usuario	Una versión aplicable localmente	Utilizarla sin conexión y con mejor rendimiento.

Nota: Historias de usuario. Fuente: Elaboración propia

3.5.2.Producto backlog

Luego de crear las historias de usuario y asignarles un id se determinó el product backlog del proyecto.

3.5.2.1.Creación del product backlog

En el producto backlog se asigna cada historia de usuario a un sprint, también se añade una columna “Estado” que servirá como Taskboard y una columna “Prioridad” que define la prioridad del sprint.

Tabla 4

Product Backlog del proyecto

Nº DE HISTORIA	DESCRIPCIÓN	PRIORIDAD	SPRINT	ESTADO
HU-1	Desplegar la técnica de rebanado en un prototipo	Alta	1	No se ha iniciado
HU-2	Desarrollar una interfaz gráfica para el prototipo	Alta	2	No se ha iniciado
HU-3	Migración y optimización en MATLAB	Media	3	No se ha iniciado
HU-4	Desarrollo de nueva interfaz gráfica con Spyder y Tkinter	Alta	4	No se ha iniciado

Nota: Descripción del product backlog *Fuente:* Elaboración propia

3.5.3. Sprint 1

Durante esta primera iteración se priorizó el diseño y la implementación de la arquitectura base del prototipo funcional, estableciéndose los cimientos de software sobre los cuales se integraron los módulos posteriores. Para el desarrollo de esta fase se seleccionó el lenguaje de programación Python debido a su versatilidad en el manejo de bibliotecas de visión por computadora, empleando el entorno de desarrollo interactivo Jupyter Notebook para facilitar la documentación y el análisis rápido de los resultados preliminares. A continuación, se presenta la planificación de actividades correspondiente a este ciclo inicial.

3.5.3.1.Sprint Backlog

En la Tabla 5 se muestran las tareas que se llevarán a cabo en este sprint, indicando a que historia de usuario pertenecen y el estado.

Tabla 5*Sprint backlog del sprint 1*

HU	DESCRIPCIÓN	ESTADO
HU-1	Implementar el modelo computacional en el IDE Jupyter Notebook	No se ha iniciado
HU-1	Cargar videos de prueba y verificar si el procesamiento es correcto.	No se ha iniciado
HU-1	Visualizar resultados (gráficos y métricas iniciales).	No se ha iniciado

Nota: Descripción de las tareas a realizar en el sprint 1 *Fuente:* Elaboración Propia

3.5.3.2.Desarrollo del sprint 1

A nivel de codificación la programación de la técnica de rebanado se realizó en un entorno de Jupyter Notebook, utilizando las librerías OpenCV2, para realizar las modificaciones a las imágenes como el cambio del modelo de color o añadir formas a algún frame, Numpy para manipular la estructura de datos en la cual se encuentra el video y matplotlib para visualizar los resultados del procesamiento dentro del mismo Jupyter Notebook

Primero se lee el video en el entorno de Jupyter Notebook, para las pruebas preliminares se usarán los frames con escala de grisis, a cada frame se le cambia el modelo de color a escala de grises para minimizar la carga del computacional. Los frames son almacenados en un arreglo que recupera el alto el ancho y el número de frame. Ej.: Forma de frames_array: (1080, 1920, 1124)

Figura 32

Resultado del modelo computacional



Nota: Resultado del modelo computacional aplicado en Jupyter Notebook con texto puesto de forma manual. *Fuente:* Elaboración propia

Como se estableció en el modelo matemático de la rebana, debemos mantener alguno de estos valores estáticos para obtener las rebanas, en el caso de las señales horizontales, se mantiene el valor correspondiente al alto en un numero estático, el valor del ancho y el número de frame varia en el tiempo, lo cual es representado con series(una lista de valores desde una posición inicial a una posición final). Después de evaluar los mejores valores para cada variable, se muestra el resultado obtenido en la rebanada.

Se aplico el pseudocodigo presentado previamente, vale aclarar que dentro de los alcances no está previsto que la técnica de rebanado detecto automáticamente las señales, pero para que se comprenda mejor el resultado y los datos obtenidos del resultado se optó por poner una pequeña descripción de las señales de manera manual

3.5.3.3.Sprint Review

Se presentó al equipo Scrum el prototipo que aplica el modelo computacional, desarrollado en Jupyter Notebook. Para las pruebas se utilizó un video de un sector con varias señales horizontales, como flechas y líneas.

El resultado obtenido cumple con lo propuesto: se logró recuperar una “rebanada” del video donde se aprecian las señalizaciones sin la presencia de autos u otros elementos que producen ruido visual. Con esto, el entregable se considera completado.

Se procede a la actualización del sprint backlog en Tabla 6 que forma parte de los artefactos de Scrum.

Tabla 6

Actualización de la tabla sprint backlog

HU	DESCRIPCIÓN	ESTADO
HU-1	Implementar el modelo computacional en el IDE Jupyter Notebook	Completado
HU-1	Cargar videos de prueba y verificar si el procesamiento es correcto.	Completado
HU-1	Visualizar resultados (gráficos y métricas iniciales).	Completado

Nota: Actualización de la columna “Estado” *Fuente:* Elaboración Propia

3.5.3.4.Sprint Retrospective

Programar con Jupyter Notebook permitió una fácil visualización de los resultados para probar las coordenadas correctas que nos ayudaron a obtener la rebanada, aunque al no poder ver el video cargando, solo quedaba esperar que la técnica se aplique viendo atentamente la pantalla, además se recomienda aplicar una interfaz de usuario.

Para finalizar este sprint se actualiza el producto backlog (Tabla 7) del proyecto.

Tabla 7

Product Backlog del proyecto al terminar el sprint 1

Nº DE HISTORIA	DESCRIPCIÓN	PRIORIDAD	SPRINT	ESTADO
HU-1	Desplegar la técnica de rebanado en un prototipo	Alta	1	Completado
HU-2	Desarrollar una interfaz gráfica para el prototipo	Alta	2	No se ha iniciado
HU-3	Migración y optimización en MATLAB	Media	3	No se ha iniciado
HU-4	Desarrollo de nueva interfaz gráfica con Spyder y Tkinter	Alta	4	No se ha iniciado

Nota: Actualización de la columna “Estado” del Product Backlog del proyecto al terminar el sprint

1 Fuente: Elaboración propia

3.5.4.Sprint 2

Durante esta iteración se priorizó el diseño y la integración de la primera versión de la interfaz gráfica de usuario, transformando el código base en un prototipo funcional e interactivo. Para el desarrollo de este componente visual se utilizó el lenguaje de programación Python en conjunto con la librería Streamlit, cuya implementación facilitó la creación rápida de un entorno web accesible para el despliegue del sistema. A continuación, se presenta la planificación de actividades correspondiente a este periodo.

3.5.4.1.Sprint Backlog

En la Tabla 8 se exponen las tareas programadas para la iteración actual, especificando la historia de usuario a la que están vinculadas y su respectivo estado de ejecución. Esta organización permite realizar un seguimiento claro del progreso del desarrollo y asegurar

que cada tarea técnica responda directamente a una necesidad real del sistema, garantizando el cumplimiento de los objetivos establecidos para esta fase del proyecto.

Tabla 8

Sprint backlog del sprint 2

HU	DESCRIPCIÓN	ESTADO
HU-2	Crear un mock up del prototipo según requerimientos	No se ha iniciado
HU-2	Aplicar el modelo del mockup en una página web con StreamLit	No se ha iniciado
HU-2	Desplegar la aplicación en la web	No se ha iniciado
HU-2	Implementar un botón de upload para subir videos	No se ha iniciado
HU-2	Implementar un componente que muestre la imagen resultante del proceso	No se ha iniciado
HU-2	Implementar un texto cuando el proceso ha finalizado	No se ha iniciado

Nota: Descripción de las tareas a realizar en el sprint 1 *Fuente:* Elaboración Propia

3.5.4.2.Desarrollo del Sprint 2

En el ámbito de la programación, el comienzo del desarrollo y la validación de la técnica de rebanado se realizaron en un entorno de Jupyter Notebook. Sin embargo, se observó que era necesario desvincular el algoritmo de esta plataforma, ya que ejecutar directamente desde un cuaderno técnico puede resultar complicado, confuso y poco intuitivo para aquellos usuarios finales que no tienen un perfil técnico en programación. Para abordar esta limitación y asegurar que la herramienta sea accesible, se diseñó una interfaz gráfica dedicada que permita simplificar la lógica del código subyacente.

Para cumplir con este requerimiento se crea un mockup del diseño del prototipo (Ver Figura 33) que anticipa la disposición tanto espacial como funcional de la aplicación. En esta propuesta visual se incluye un encabezado con el nombre de la plataforma, un elemento de carga de archivos pensado para importar videos, y un área de salida conocida como "Resultado" donde se mostrará la imagen panorámica final. Además, se decidió omitir módulos de autenticación o inicio de sesión, dado que el sistema funciona localmente y no necesita guardar datos de usuario de manera persistente. Siguiendo esta misma idea de simplicidad y enfoque en una sola función, se decidió no incluir una barra de

navegación, centralizando todas las interacciones en una única pantalla principal para mejorar la experiencia del usuario.

Figura 33

Mockup del prototipo

Nombre del prototipo

Subir el video

Drag and drop
Browse Files

Resultado

Nota: Diseño previo del prototipo funcional. *Fuente:* Elaboración propia

Una vez acabado el diseño, se replican los componentes del mock up en una página web con Streamlit (Ver Figura 34).

Figura 34

Diseño del prototipo

Prototipo funcional de la técnica de rebanado de video ↗

Sube un video para procesarlo y generar una imagen de salida.

Sube tu video aquí

☁

Drag and drop file here

Limit 200MB per file • MP4, AVI, MOV, MKV, MPEG4

Browse files

Nota: Diseño de la página web del prototipo con Streamlit

Para facilitar el acceso al prototipo, se decidió desplegar la aplicación web utilizando el servicio Streamlit Community Cloud. Este servicio gratuito se encarga de instalar las

dependencias y configurar el servidor automáticamente. El proceso es simple, se debe subir el archivo a un repositorio GitHub junto a dos archivos de configuración de tipo txt (para las bibliotecas y otros) y conectar la cuenta GitHub con Streamlit Community Cloud, una vez dados los permisos e ingresar a la plataforma de Streamlit permite elegir una aplicación desde un repositorio de GitHub para desplegarla.

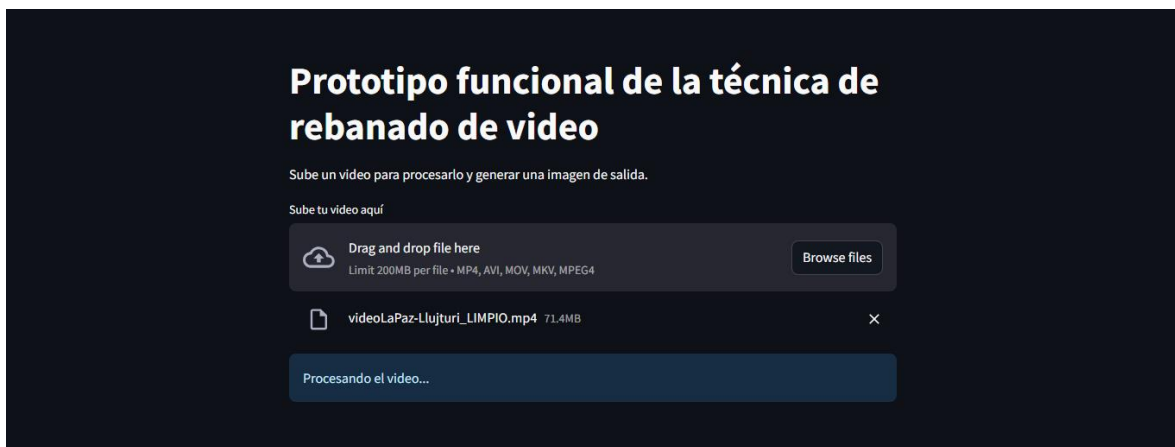
3.5.4.3.Sprint Review

Durante el Sprint Review se realizan demostraciones del funcionamiento del prototipo, control de tiempo, manejo de memoria y verificación de interfaz intuitiva para el usuario.

En la aplicación se sube el video del tramo elegido, en este caso La Paz – Llujturi, internamente se almacena como un archivo temporal, el sistema muestra un mensaje de procesando mientras se aplica la técnica y cuando el proceso termina el mensaje indica que el proceso ha finalizado y se muestra la imagen resultante. A pesar de funcionar bien de manera local, al desplegarlo en la web luego de procesar un par de videos StreamLit mostraba un mensaje de error referido a los paquetes.

Figura 35

Pantalla del prototipo en la parte del procesamiento

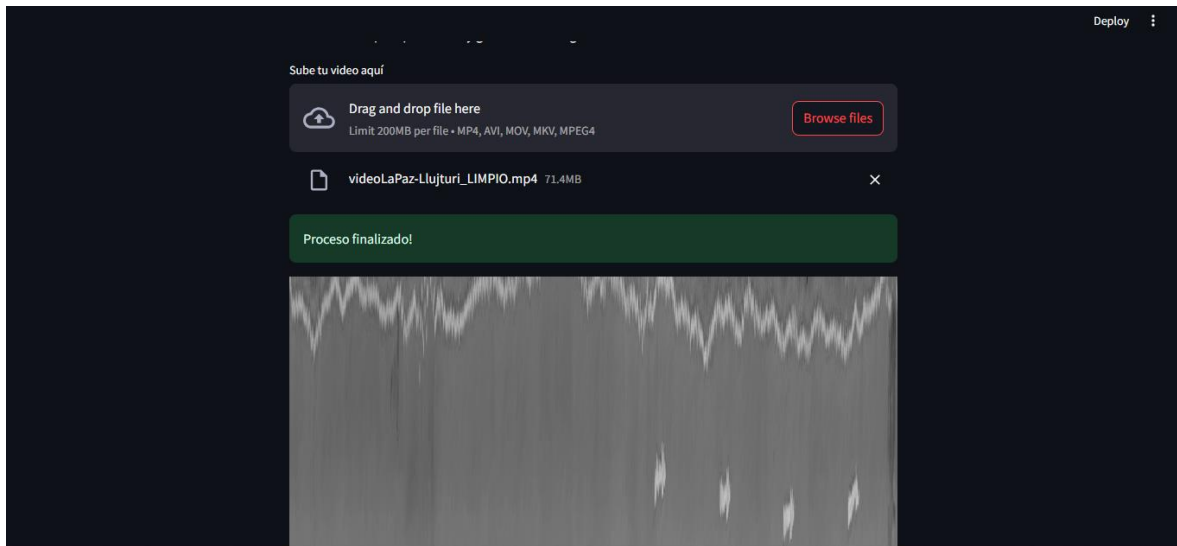


Nota: En esta pantalla el video ya ha sido subido y se muestra un mensaje que indica que el video se está procesando. Fuente: Prototipo funcional de la técnica de rebanado de video.

Después de analizar el error que mostraba Streamlit se detectó que el prototipo consumía recursos que resultaban excesivos para la plataforma de despliegue Streamlit Community Cloud. Una vez revisados los resultados del presente sprint se procede a la actualización del Sprint backlog

Figura 36

Pantalla del prototipo mostrando la imagen resultante



Nota: En esta pantalla el video ya ha sido procesado, se muestra un mensaje para que el usuario sepa que el proceso ha concluido y se despliega la imagen resultante. *Fuente:* Prototipo funcional de la técnica de rebanado de video.

Tabla 9

Actualización del Sprint backlog del sprint 2

HU	DESCRIPCIÓN	ESTADO
HU-2	Crear un mock up del prototipo según requerimientos	Completado
HU-2	Aplicar el modelo del mockup en una página web con StreamLit	Completado
HU-2	Desplegar la aplicación en la web	Completado
HU-2	Implementar un botón de upload para subir videos	Completado
HU-2	Implementar un componente que muestre la imagen resultante del proceso	Completado
HU-2	Implementar un texto cuando el proceso ha finalizado	Completado

Nota: Actualización del estado de las tareas a realizadas en el sprint 2 *Fuente:* Elaboración Propia

3.5.4.4. Sprint Retrospective

El prototipo cumple con todas las historias de usuario. Sin embargo, el Scrum Master (tutor) recomienda revisar el uso de recursos del servicio Streamlit Community Cloud y el usuario desea poder elegir la dirección del rebanado, ya que el rebanado podría no dar los resultados esperados en diversos casos.

Para finalizar el sprint se procede con la actualización del product backlog del proyecto (Tabla 10)

Tabla 10

Product Backlog del proyecto al terminar el sprint 2

Nº DE HISTORIA	DESCRIPCIÓN	PRIORIDAD	SPRINT	ESTADO
HU-1	Desplegar la técnica de rebanado en un prototipo	Alta	1	Completado
HU-2	Desarrollar una interfaz gráfica para el prototipo	Alta	2	Completado
HU-3	Migración y optimización en MATLAB	Media	3	No se ha iniciado
HU-4	Desarrollo nueva interfaz gráfica con Spyder y Tkinter	Alta	4	No se ha iniciado

Nota: Actualización de la columna “Estado” del Product Backlog del proyecto al terminar el sprint

2 Fuente: Elaboración propia

3.5.5.Sprint 3

Durante esta iteración, se contempló la migración temporal del algoritmo base al entorno de MATLAB con el objetivo de optimizar la gestión de recursos computacionales. Asimismo, se incorporaron los requerimientos solicitados por el usuario, orientados a brindar flexibilidad en la selección de la dirección del rebanado. A continuación, se presenta la planificación correspondiente a este periodo, listando las actividades que serán ejecutadas.

3.5.5.1.Sprint Backlog

En este apartado se detallan las tareas programadas para la iteración actual, especificando la historia de usuario a la que están vinculadas y su respectivo estado de ejecución. Esta organización permite realizar un seguimiento claro del progreso del desarrollo y asegurar que cada tarea técnica responda directamente a una necesidad real del sistema, facilitando la gestión del tiempo y garantizando el cumplimiento de las metas establecidas para este sprint. Asimismo, este registro actúa como un instrumento de auditoría interna que permite evaluar el impacto de la migración de entorno sobre los tiempos de entrega previstos. A través de este desglose estructurado, el tribunal evaluador puede constatar la correspondencia directa entre los requerimientos funcionales y las pruebas de rendimiento

ejecutadas. De este modo, se asegura la trazabilidad metodológica de cada componente de software integrado en esta fase del proyecto.

Tabla 11

Sprint backlog del sprint 3

HU	DESCRIPCIÓN	ESTADO
HU-3	Re implementar el modelo computacional en MATLAB	No se ha iniciado
HU-3	Analizar formas de optimización.	No se ha iniciado
HU-3	Comparar ambos algoritmos	No se ha iniciado

Nota: Descripción de las tareas a realizar en el sprint 3 *Fuente:* Elaboración Propia

3.5.5.2.Desarrollo del sprint 3

Se optó por trasladar el algoritmo básico de la técnica de rebanado a Matlab para tener mayor control de las variables y del flujo de los datos, además permite hacer debug de una forma más cómoda que usando Jupyter Notebook. Se detectó que, en la primera versión, se leían todos los frames y se almacenaban en una variable (evidentemente esta variable crecía en dependencia del tipo de video analizado), se notó que hacer esta primera lectura de todo el video no era necesario ya que se podría simplemente tomar la línea que al usuario le interesa recortar. De esta forma en lugar de leer todo los frames que generaban una gran cantidad de datos temporales y además inservibles para el fin del algoritmo, se optó por solo tomar los pixeles de la dirección del corte, la rebanada no pierde información.

Para cumplir con el análisis de la técnica, se decidió utilizar el algoritmo de línea de Bresenham, el cual se implementó como una función propia dentro del desarrollo. Aunque existen librerías que ya incluyen este algoritmo, el equipo prefirió recrear su lógica desde cero, evitando depender de módulos externos y entendiendo mejor su funcionamiento interno.

Esta decisión fue bastante útil, ya que permitió tener un control total sobre el proceso. Gracias a eso, se pudo analizar con más detalle la cantidad de datos recuperados, la forma y el orden en que se obtenían, además de las ubicaciones y formatos en los que se almacenaban dentro del proyecto. Tener esta visión más cercana del funcionamiento del algoritmo ayudó a comprobar el rendimiento del sistema y a realizar mejoras. Finalmente el algoritmo replanteado y cumpliendo la misma funcionalidad se describe en la Figura 37,

comparando ambos algoritmos, en el primero se almacenaba todo el video temporalmente, involucrando mucha memoria, ahora el video se lee una vez y en cada lectura de frame se guarda solo la información deseada, ya no todo el frame.

Figura 37

Replanteamiento del algoritmo base

- 1. INICIO**
- 2.** Establecer 2 puntos
- 3.** Utilizar el algoritmo de línea de Bresenham con los puntos previamente elegidos para recuperar las coordenadas de los pixeles por los cuales pasaría una línea recta entre los puntos
- 4.** Almacenar las coordenadas en una variable global
- 5.** Lectura de frame por frame del video elegido
 - 5.1.** Cambiar frame de BGR a RGB
 - 5.2.** Si existen coordenadas de una línea
 - Extraer el valor del pixel correspondiente al frame del video
 - Añadir los valores obtenidos a una variable de tipo lista (llamaremos fila_pixeles)
- 6.** Mostrar fila_pixeles
- 7. FIN**

Nota: Pseudocódigo del algoritmo base con mejor rendimiento y menos uso de memoria. *Fuente:*

Elaboración propia

3.5.5.3.Sprint Review

Se analizaron los resultados de procesamiento, el nuevo flujo de trabajo es mas optimo en varios aspectos, desde los niveles lógicos hasta el consumo de recursos, es evidente que el nuevo algoritmo utiliza menos memoria temporal, el flujo es más robusto, por lo tanto el objetivo del sprint se cumplió. Se procede a actualizar el estado de las tareas del sprint backlog

Tabla 12 Actualización del Sprint backlog del sprint 3

HU	DESCRIPCIÓN	ESTADO
HU-3	Re implementar el modelo computacional en MATLAB	Completado
HU-3	Analizar formas de optimización.	Completado
HU-3	Comparar ambos algoritmos	Completado

Nota: Actualización de la columna de “Estado” después del sprint 3 *Fuente:* Elaboración Propia

3.5.5.4.Sprint Retrospective

La migración temporal de Python al entorno de MATLAB fue altamente beneficioso para el prototipo, ahora el uso de memoria es menor y el usuario podrá seleccionar la dirección del rebanado, sin embargo para evitar futuros problemas de uso de memoria el tutor juntamente con el desarrollador plantea generar un prototipo local, dado que la empresa ECOVIANA S.R.L tiene computadores robustos.

Para finalizar el sprint se procede con la actualización del product backlog del proyecto (Tabla 13)

Tabla 13

Product Backlog del proyecto al terminar el sprint 3

N° DE HISTORIA	DESCRIPCIÓN	PRIORIDAD	SPRINT	ESTADO
HU-1	Desplegar la técnica de rebanado en un prototipo	Alta	1	Completado
HU-2	Desarrollar una interfaz gráfica para el prototipo		2	Completado
HU-3	Migración y optimización en MATLAB	Media	3	Completado
HU-4	Desarrollo nueva interfaz gráfica con Spyder y Tkinter	Alta	4	No se ha iniciado

Nota: Actualización de la columna “Estado” del Product Backlog del proyecto al terminar el sprint

3 Fuente: Elaboración propia

3.5.6.Sprint 4

Durante este sprint se creará un prototipo que funcione de manera local, usando el algoritmo mejorado del sprint 3, además se aplicara una interfaz de usuario que permita visualizar con claridad el resultado final es decir el rebanado, la imagen resultante. Ahora el usuario podrá elegir la dirección del rebanado, para esto también se creará una interfaz gráfica usando eventos como clics y de pulsación de tecla.

También se mejora la salida mediante el uso de filtros y transformaciones propias de OpenCV. En la siguiente sección se presenta el sprint backlog y las actividades correspondientes a su desarrollo.

3.5.6.1.Sprint Backlog

En la Tabla 14 se muestran las tareas que se llevarán a cabo en este sprint, indicando a que historia de usuario pertenecen y el estado.

Tabla 14

Sprint backlog del sprint 4

HU	DESCRIPCIÓN	ESTADO
HU-4	Traducir el algoritmo optimizado de MATLAB a Python	No se ha iniciado
HU-4	Implementar una interfaz grafica	No se ha iniciado
HU-4	Implementar lógica con detección de eventos para que el usuario elija la dirección del rebanado.	No se ha iniciado
HU-4	Editar con herramientas de OpenCV la imagen de salida	No se ha iniciado
HU-4	Añadir la funcionalidad de guardar el resultado en un directorio seleccionado por el usuario	No se ha iniciado

Nota: Descripción de las tareas a realizar en el sprint 4 *Fuente:* Elaboración Propia

3.5.6.2.Desarrollo del sprint 4

Para crear el prototipo, se cambió de Jupyter Notebook a Visual Studio ya que este puede manejar eventos y abrir ventanas en el sistema (elemento excesivamente importante para implementar las nuevas funcionalidades).

La traducción del algoritmo creado en el sprint anterior se migro a Python, este paso fue relativamente fácil, a nivel de programación, se cambió la sintaxis y se añadieron algunas validaciones extras. El punto clave de este sprint es la creación de la lógica para mejorar la experiencia del usuario. En esta parte se usaron las librerías Tkinter, Matplotlib, OpenCV, Numpy y OS.

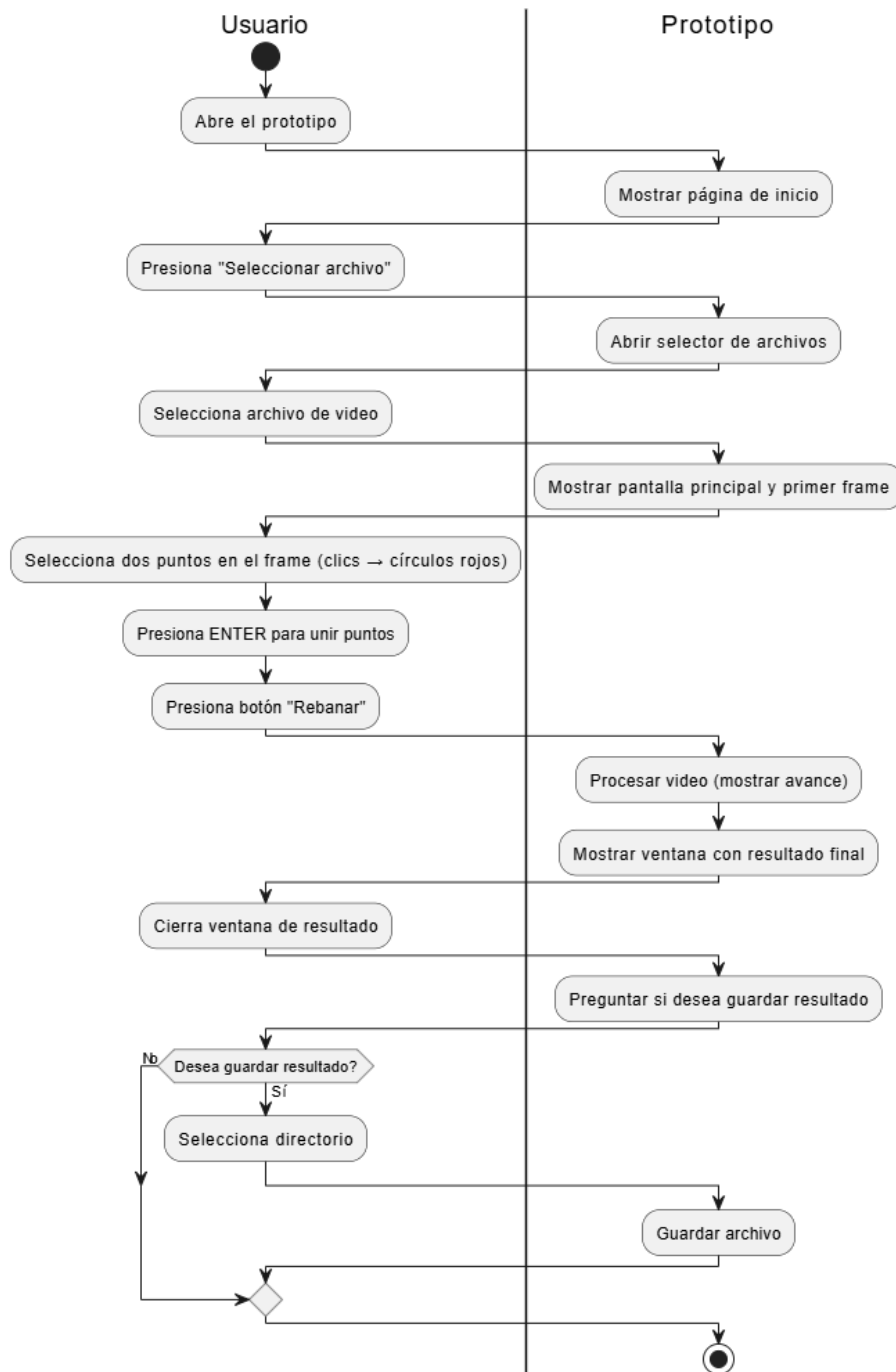
El siguiente paso fue descifrar como lograr que el usuario indique la dirección del rebanado, primero se consideró la opción de que el usuario ingrese las coordenadas, es decir un valor numérico pero esto sería realmente poco intuitivo además de consumir tiempo del usuario final, lo cual justamente esperamos reducir, finalmente se decidió que el usuario elija dos puntos.

El prototipo mostraría el primer frame del video y el usuario deberá hacer click para elegir los puntos, en el lugar que el usuario haga click se verán unos puntos de color rojo,

posteriormente el usuario debe apretar la tecla Enter, el prototipo dibuja una línea verde discontinua entre ambos puntos y esta será la línea que defina la dirección del rebanado.

Figura 38

Diagrama de actividad UML

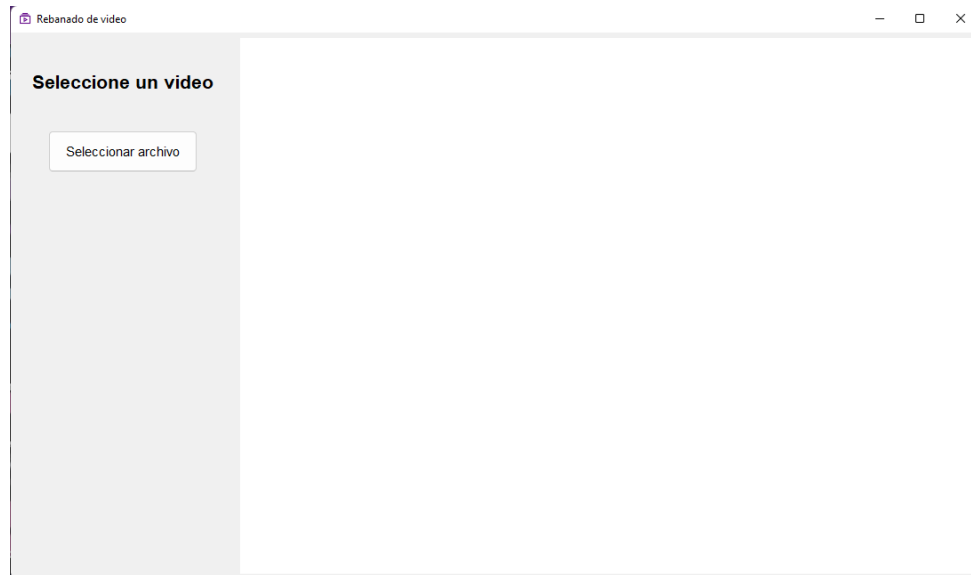


Nota: Ilustración de la interacción esperada del usuario con el prototipo. Fuente: Elaboración

Propia

Figura 39

Pantalla de inicio del prototipo

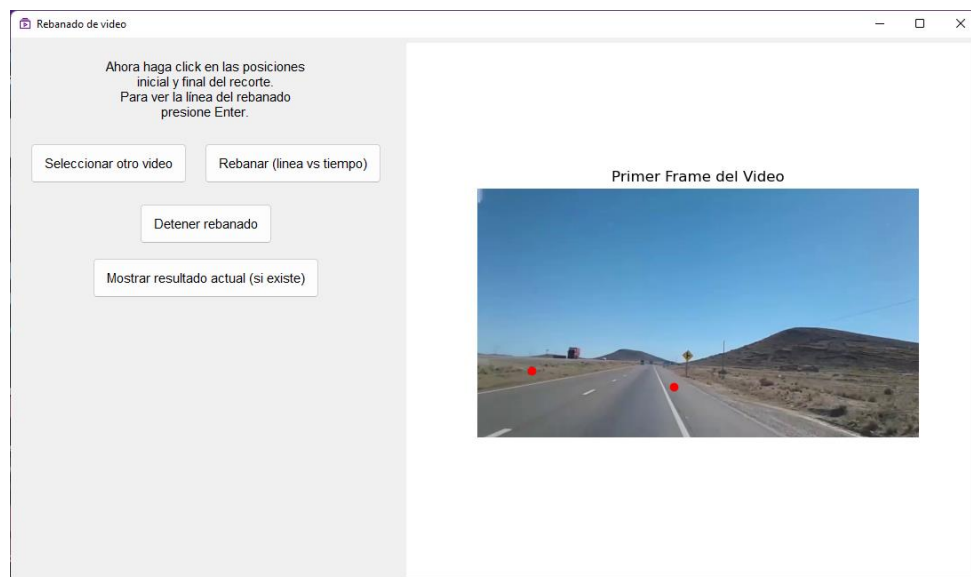


Nota: Captura de pantalla de la ventana del prototipo recién iniciado. *Fuente:* Prototipo técnica de rebanado

La primera pantalla (Ver Figura 39) se mantiene simple y con una interfaz limpia, instrucciones claras para el usuario y un botón subir el video

Figura 40

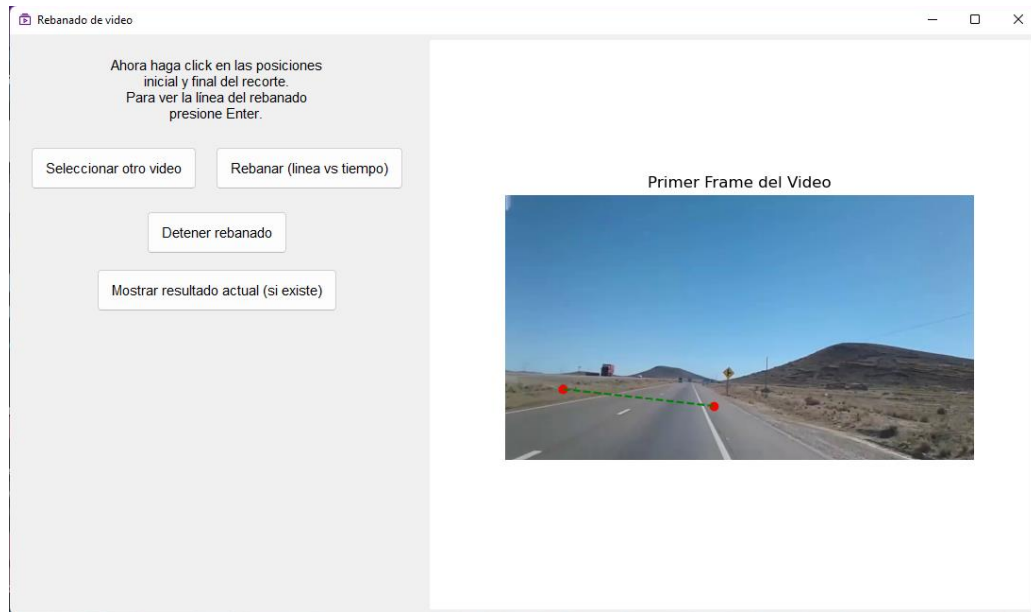
Pantalla principal con video ya cargado



Nota: En la parte izquierda, se puede apreciar los puntos rojos, esta captura es tomada luego de que el usuario seleccione los puntos. *Fuente:* Prototipo técnica de rebanado

Figura 41

Captura de pantalla después de pulsar la tecla “Enter”



Nota: En la parte izquierda en el frame se puede apreciar la línea verde que ser la dirección del rebanado. Fuente: Prototipo técnica de rebanado

Para evitar problemas como tomar puntos que no estén en el frame, internamente se validaba que los puntos seleccionados estén dentro el frame, también se validó la cantidad de puntos, es decir el usuario solo puede seleccionar 2 puntos, al tercer click se borra las coordenadas elegidas y los puntos se borran del frame.

Una vez elegida la dirección, hay un botón con el texto “Rebanar” que inicia el procesamiento, pero antes de iniciar internamente valida si existen las coordenadas de la línea (dirección) elegida por el usuario, también existe un botón que regresa todo al punto de inicio, en caso de que se quiera detener el procesamiento y volver a iniciar.

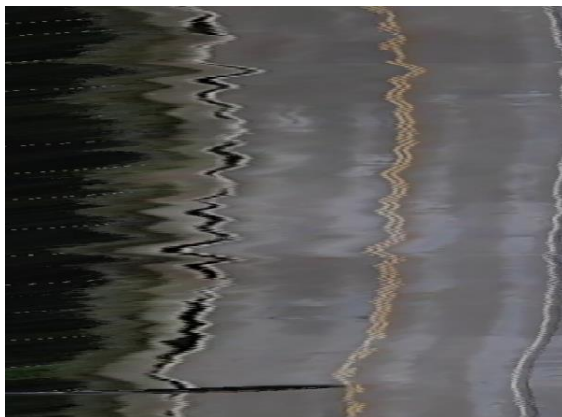
Es posible ver el proceso del rebanado de dos formas, existe la opción de ver un preview del resultado final, pero también el usuario puede ver en su pantalla el video avanzando. Todos los elementos de la interfaz se hicieron con Tkinter, desde la creación de la pantalla, inclusión del icono, el uso de botones, el manejo de lo que el usuario ve en su pantalla, también se utilizó esta librería para abrir la ventana de dialogo para guardar la imagen en cualquier parte que el usuario decida.

Una vez terminado el procesamiento, el prototipo muestra una imagen final, esta imagen final pasa por una edición previa antes de mostrarla al usuario.

Como se puede ver en las siguientes imágenes,(Figura 42 y Figura 43) la primera es el resultado bruto sin edición que mejore la visualización, vale mencionar que el tamaño de la imagen depende directamente de la cantidad de frames que tenga el video de entrada, en este caso el video de entrada duraba alrededor de 20 segundos y tenía una captura de aproximadamente 30 fotogramas por segundo. En la segunda imagen, notamos la carretera con mayor facilidad, en el video de prueba utilizado no existen muchas señales, lo más notorio es el cambio en la línea amarilla del centro, ya que al inicio es segmentada luego se vuelve continua

Figura 42

Imagen sin edición



Nota: Imagen sin filtros ni edición Fuente:
Resultado de prototipo

Figura 43

Imagen final con filtros y correcciones aplicadas.



Nota: Imagen re escalada de forma horizontal para asemejarse más al tamaño real y aplicación de contraste CLAHE. Fuente: Resultado de prototipo

3.5.6.3.Sprint Review

Se realizaron todas las tareas previstas para este sprint, involucro un poco más de análisis de lo previsto pero finalmente se cumplieron todos los requerimientos. En cuanto a las pruebas, el programa responde adecuadamente a videos de corta duración, pero se identificó un problema con videos que superan los 10 minutos.

Tabla 15

Actualización del Sprint backlog del sprint 4

HU	DESCRIPCIÓN	ESTADO
HU-4	Traducir el algoritmo optimizado de MATLAB a Python (Spyder)	Completado
HU-4	Implementar una interfaz grafica	Completado
HU-4	Implementar lógica con detección de eventos para que el usuario elija la dirección del rebanado.	Completado
HU-4	Editar con herramientas de OpenCV la imagen de salida	Completado
HU-4	Añadir la funcionalidad de guardar el resultado en un directorio seleccionado por el usuario	Completado

Nota: Actualización de la columna estado al finalizar el sprint 4 *Fuente:* Elaboración Propia

3.5.6.4.Sprint Retrospective

Se presentó el prototipo funcional que surgió de este sprint ante el tutor académico y el representante de la empresa; ambos ofrecieron recomendaciones para la mejora del sistema. Desde la empresa, se destacó la necesidad técnica de manejar archivos de video extensos, superando la capacidad de prueba de dos horas. Por su parte, el asesor sugirió realizar ajustes en la interfaz de usuario (UI). Dado que se alcanzaron todos los objetivos establecidos para este sprint, se procedió a actualizar el product backlog. Para abordar las nuevas observaciones, se decidió llevar a cabo un quinto sprint centrado solamente en tareas de optimización y pruebas, las cuales permitirán garantizar la estabilidad del software en situaciones de gran demanda de procesamiento. Esta fase adicional incluye la realización de pruebas de carga de estrés para evaluar la respuesta del algoritmo ante la disminución de recursos del sistema. Así, se pretende reducir posibles fallos por desbordamiento de búfer y fugas de memoria durante la lectura de flujos multimedia prolongados. Con la incorporación de estas actividades, se asegura la entrega de un producto final sólido que cumple de manera precisa con los estándares de calidad requeridos tanto en el ámbito académico como en el entorno operativo real de la empresa.

Tabla 16*Product Backlog del proyecto al terminar el sprint 4*

Nº DE HISTORIA	DESCRIPCIÓN	PRIORIDAD	SPRINT	ESTADO
HU-1	Desplegar la técnica de rebanado en un prototipo	Alta	1	Completado
HU-2	Desarrollar una interfaz gráfica para el prototipo	Alta	2	Completado
HU-3	Migración y optimización en MATLAB	Media	3	Completado
HU-4	Desarrollo nueva interfaz gráfica con Spyder y Tkinter	Alta	4	Completado
HU-5	Optimización del procesamiento	Alta	5	No se ha iniciado
HU-6	Mejorar interfaz gráfica	Alta	5	No se ha iniciado

Nota: Actualización de la columna “Estado” del Product Backlog del proyecto al terminar el sprint

4 *Fuente:* Elaboración propia

3.5.7.Sprint 5

Durante esta iteración sprint, se abordaron las observaciones identificadas en la etapa anterior. El proceso de optimización incluyó diversos ajustes técnicos que permitieron obtener resultados superiores a los proyectados inicialmente. Asimismo, se llevó a cabo un rediseño de la interfaz gráfica y se optimizaron componentes clave de la experiencia de usuario, asegurando un sistema más intuitivo y eficiente.

3.5.7.1.Sprint Backlog

En la Tabla 14 se muestran las tareas que se llevaran a cabo en este sprint, indicando a que historia de usuario pertenecen y el estado.

Tabla 17*Sprint backlog del sprint 5*

HU	DESCRIPCIÓN	ESTADO
HU-5	Optimización del procesamiento	No se ha iniciado
HU-6	Mejorar la interfaz grafica	No se ha iniciado

Nota: Descripción de las tareas a realizar en el sprint 5 *Fuente:* Elaboración Propia

3.5.7.2.Desarrollo del sprint 5

Durante el desarrollo del Sprint 5 se realizaron modificaciones significativas sobre la arquitectura inicial del sistema de rebanado de video, motivadas principalmente por los problemas identificados en iteraciones anteriores relacionados con el consumo excesivo de memoria, el bajo rendimiento en videos extensos y las limitaciones del procesamiento tradicional basado en almacenamiento completo de cuadros.

En iteraciones previas, el sistema cargaba y almacenaba la totalidad de los *frames* en la memoria RAM antes de procesarlos. Durante este sprint, se replicaron los momentos críticos de falla y se determinó que, para videos extensos, el objeto **NumPy** resultante crecía de manera exponencial, agotando los recursos computacionales del sistema.

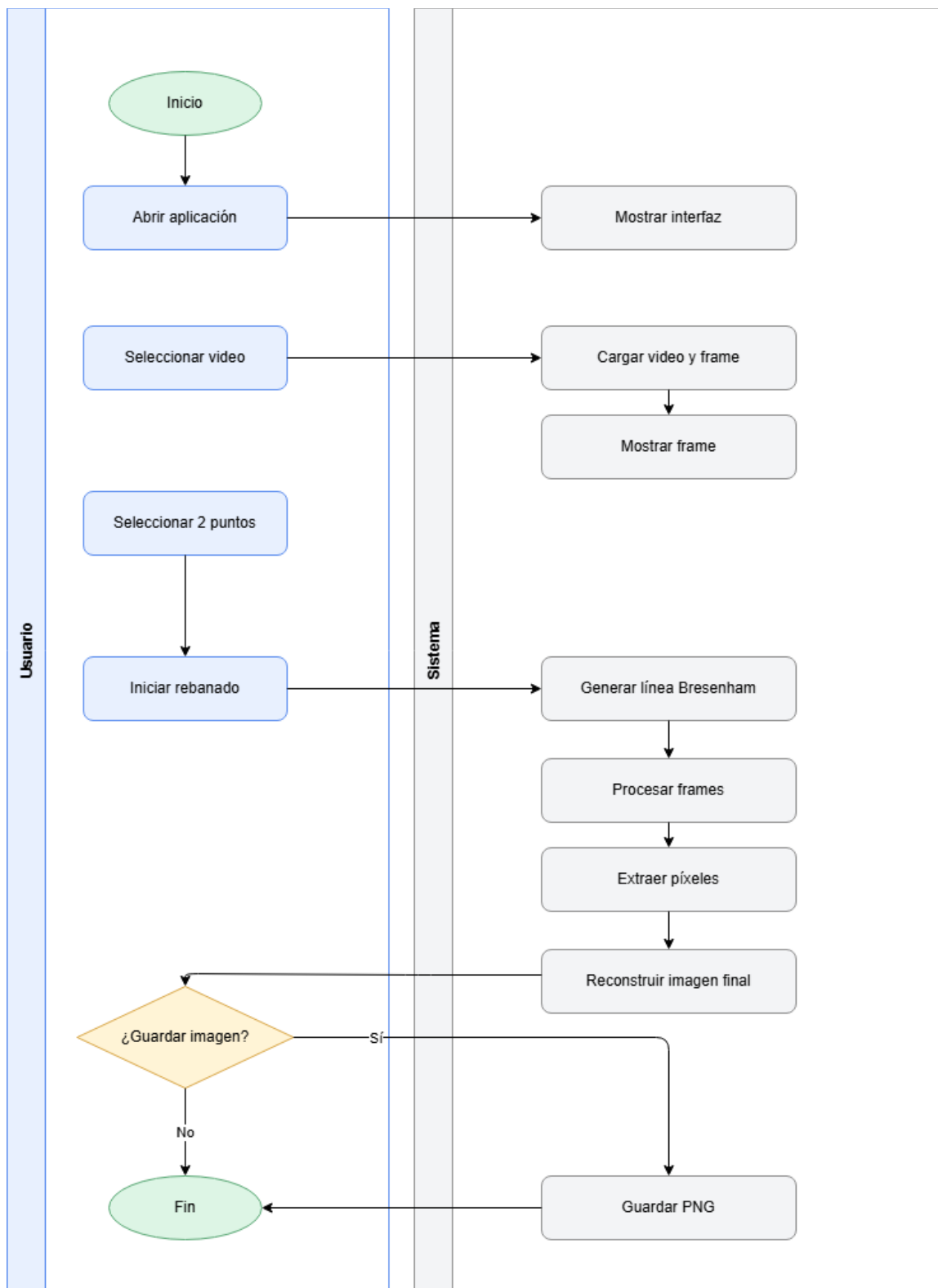
Para solventar esto, se implementaron tres cambios fundamentales:

- **Gestión de Almacenamiento:** Se integró el uso de estructuras **NumPy memmap**, permitiendo que los datos se almacenen temporalmente en el disco (archivo memmap.dat) en lugar de saturar la memoria principal.
- **Suspensión de Previsualización:** Se identificó que reproducir el video en la interfaz mientras se realizaba el rebanado consumía recursos innecesarios. Por ello, se optimizó el flujo para que el procesamiento ocurra en segundo plano, maximizando la velocidad.
- **Estandarización de FPS:** Se verificó que, para el análisis de videos largos, una tasa de **30 fps** era suficiente para mantener la precisión sin sobrecargar el sistema, ajustando la lógica de guardado de la imagen final

Se mantuvo la lógica de obtención de la "rebanada" mediante el algoritmo de Bresenham, el cual calcula con precisión la trayectoria lineal entre los puntos seleccionados por el usuario. No obstante, se detectó una limitación crítica en Matplotlib: al intentar visualizar imágenes resultantes de videos muy largos, estas rebasaban los límites de renderizado de la librería. Ante esto, se optó por migrar todas las visualizaciones a una librería más eficiente para el manejo de grandes volúmenes de píxeles, garantizando que el usuario pueda ver el resultado final sin errores de visualización..

Figura 44

Diagrama UML de secuencia



Nota: Diagrama que refleja el flujo de la interacción del usuario con el prototipo.

Fuente: Elaboración propia

Figura 45*Wireframe del rediseño***Rebanado de video**

Instrucciones: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id.

Tiempo aproximado:--:--



Nota: Wireframe del diseño aplicado al prototipo en el sprint 5. Fuente: Elaboración propia

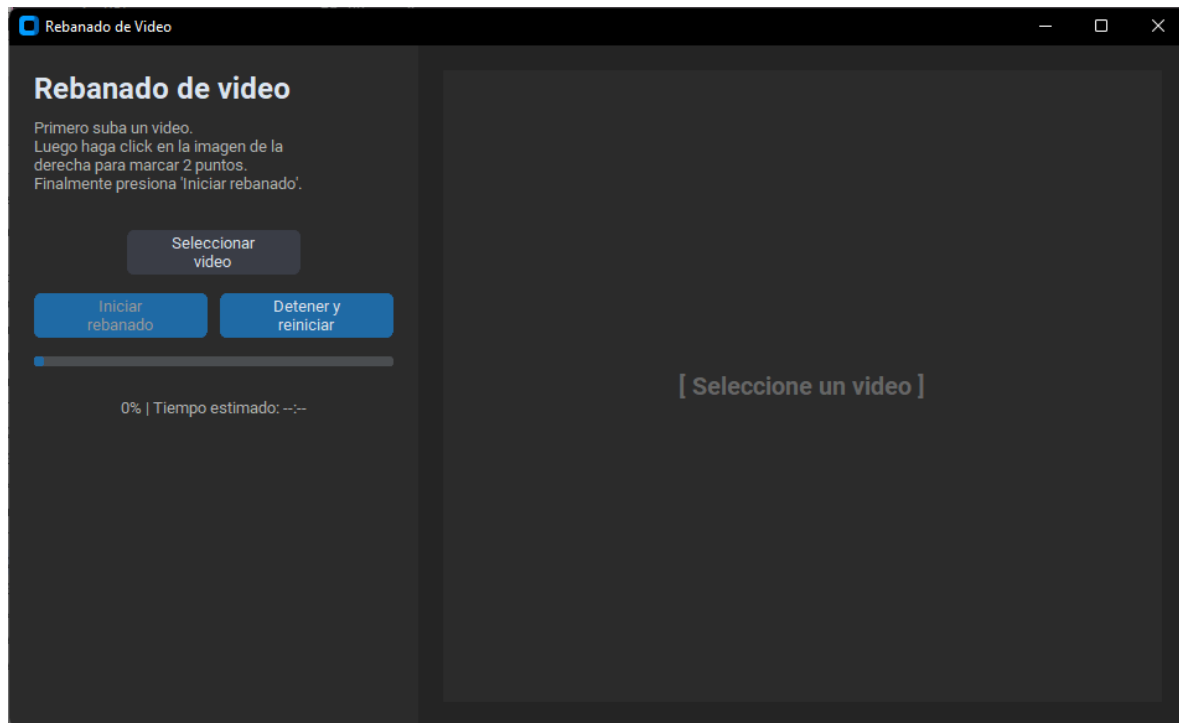
Dentro del aspecto visual, se priorizó una estética minimalista orientada a optimizar la interacción directa con el usuario. La interfaz gráfica se simplificó mediante la supresión de la página de inicio, reduciendo la fricción operativa y concentrando todas las funcionalidades en una única ventana principal compuesta por tres controles esenciales. El primero es el botón para la selección o sustitución del archivo de video, el cual despliega de forma automatizada el explorador de archivos del sistema operativo para la carga del recurso multimedia. El segundo corresponde al activador del procesamiento, estructurado con un bloque de validación lógica y restricciones de control que impiden la ejecución en vacío mediante la verificación previa de la inicialización de variables. Finalmente, se integró un control de detención y restauración, cuya función es interrumpir de manera inmediata el flujo del algoritmo y reestablecer los parámetros y la interfaz a su estado inicial.

De manera complementaria, se incorporaron mecanismos dinámicos de retroalimentación en tiempo real para guiar al usuario durante el ciclo de ejecución. Entre estos componentes destaca una barra de progreso diseñada para la visualización del estado porcentual del procesamiento de los cuadros, acompañada por un estimador de tiempo restante (Estimated Time of Arrival, ETA) que mejora la percepción de control sobre la tarea en curso. Asimismo, se implementó un sistema de etiquetado cronológico automático que añade

marcas temporales indexadas sobre la imagen panorámica final, facilitando la interpretación ágil y secuencial de la infraestructura vial a lo largo de todo el trayecto analizado.

Figura 46

Captura de pantalla del prototipo



Nota: Interfaz grafica final del prototipo. Fuente: Elaboración propia

3.5.7.3.Sprint Review

Los objetivos establecidos para este sprint se cumplieron exitosamente. Por un lado, se solventaron las dificultades técnicas asociadas al procesamiento de videos de larga duración y se optimizó notablemente el tiempo de ejecución del algoritmo, comprobado mediante una recolección de datos de tiempo de ejecución, se implementó un archivo temporal además se cambió la forma de mostrar el frame del video, se eliminó la funcionalidad de reproducir el video para minimizar el uso de recursos.

Tabla 18

Video	Duración	Tiempo de procesamiento	
		Prototipo Sprint 4	Prototipo Sprint 5

ecovianaVideo.avi	00:00:20	00:02:30	00:00:22
GUABIRAOKINAWA.mp4	00:36:12	01:45:19	00:12:45

Nota: Datos tomados durante el test al prototipo. Fuente: Elaboración propia

Por otro lado, respecto a la interfaz gráfica, se simplificó el diseño visual, se amplió el área destinada a la previsualización del video y se integró una barra de progreso que proporciona retroalimentación en tiempo real sobre el estado del proceso. Adicionalmente, se añadieron etiquetas temporales a la imagen resultante para facilitar su interpretación cronológica, de esta manera el usuario podrá tener una referencia temporal y consultar partes específicas del video procesado.

Tabla 19

Actualización del Sprint backlog del sprint 5

HU	DESCRIPCIÓN	ESTADO
HU-5	Optimización del procesamiento	Completado
HU-6	Mejorar la interfaz grafica	Completado

Nota: Actualización de la columna estado al finalizar el sprint 5 Fuente: Elaboración Propia

3.5.7.4.Sprint Retrospective

A la conclusión de este sprint, se tiene como resultado un prototipo robusto, escalable, intuitivo y optimizado para su uso. Según las pruebas realiza ahora le prototipo procesa los videos 3 veces más rápido que antes. La obtención de estas métricas se realizó mediante una función para guardar el nombre de la video duración del mismo y el tiempo que el prototipo tardaba en procesarlo.

Tabla 20

Product Backlog del proyecto al terminar el sprint 4

N° DE HISTORIA	DESCRIPCIÓN	PRIORIDAD	SPRINT	ESTADO
HU-1	Desplegar la técnica de rebanado en un prototipo	Alta	1	Completado
HU-2	Desarrollar una interfaz gráfica para	Alta	2	Completado

	el prototipo			
HU-3	Migración y optimización en MATLAB	Media	3	Completado
HU-4	Desarrollo nueva interfaz gráfica con Spyder y Tkinter	Alta	4	Completado
HU-5	Optimización del procesamiento	Alta	5	Completado
HU-6	Mejorar interfaz gráfica	Alta	5	Completado

Nota: Actualización de la columna “Estado” del Product Backlog del proyecto al terminar el sprint

5 Fuente: Elaboración propia

Dado que los objetivos previstos para este sprint se cumplieron en su totalidad, se procedió a la actualización del product backlog. Con el fin de solventar las nuevas observaciones, se determinó la creación de un quinto sprint enfocado exclusivamente en tareas de optimización y pruebas, (Ver)

3.6.EVALUACIÓN TÉCNICA

Para la evaluación técnica se optó por una evaluación cualitativa mediante grupo focal. El objetivo de este tipo de evaluación es recolectar las percepciones y experiencias, a diferencia de los métodos cuantitativos (como encuestas), no se busca obtener datos estadísticamente representativos, sino explorar la riqueza de los significados que los profesionales del área vial de la empresa atribuyen a su experiencia.

La conformación de los grupos focales, así como la designación del moderador y los participantes, estuvo a cargo de la gerencia de ECOVIANA S.R.L. Para la evaluación, se entregó el sistema desarrollado a la empresa, el cual fue utilizado durante las sesiones de análisis. Los resultados de este proceso se encuentran reflejados en la carta de aval correspondiente (ver Anexo H: Cartas de ECOVIANA S.R.L.).

El desarrollo de la técnica de rebanado de video para la generación de imágenes 2D de la distribución de señales de tránsito ha demostrado ser técnicamente viable y efectiva para potenciar el software de análisis de videos de ECOVIANA S.R.L. La evaluación se centró en la reducción de información irrelevante contenida en los videos, la precisión en la identificación de señales de tránsito, el ahorro de tiempo en el análisis y la confiabilidad de los resultados frente al sesgo del observador.

Los grupos focales y pruebas funcionales realizadas dentro de la empresa mostraron que el prototipo permite extraer imágenes 2D relevantes de los videos, facilitando la interpretación y validación de los inventarios viales. Además, se evidenció que la integración del modelo de rebanado con el software existente mejora significativamente la eficiencia del proceso, al reducir el tiempo de supervisión manual y estandarizar los resultados. La documentación técnica generada proporciona una guía clara para futuras mejoras y una base sólida para la implementación definitiva del sistema.

3.7.EVALUACIÓN ECONÓMICA

Dadas las características del proyecto se realiza la evaluación económica tomando en cuenta el costo de estudio, el costo de las pruebas y el costo del esfuerzo basado en el tiempo empleado.

Para determinar el costo de estudio se considera el precio mensual del plan de internet utilizado durante la etapa de investigación, este plan contempla tanto móvil como conexión Wifi.

Tabla 21

Tabla descriptiva de gastos

Descripción	Cantidad	Costo mensual (Bs)	Costo total (Bs)
Plan de internet wifi e internet móvil ilimitado	11 meses	309	3399

Nota: Descripción de los costos de internet. *Fuente:* Elaboración propia y datos obtenidos en la web (Tigo, s.f.)

La contabilización del trabajo dedicado se expone en la siguiente tabla

Tabla 22

Contabilización de trabajo dedicado en horas

Descripción	Tiempo dedicado (en horas)
Investigación sobre el análisis espacio-temporal	560
Creación del modelo matemático	280
Creación del modelo computacional	280
Desarrollo del prototipo (incluyendo investigación de librerías)	210
Total	1330 horas

Nota: Descripción del tiempo trabajado *Fuente:* Elaboración propia

Para la evaluación económica basada en el esfuerzo y tiempo dedicado se consideró como base un sueldo de Bs.- 8000 al mes para el rol de analista o ingeniero en simulación/machine learning, considerando el tiempo de desarrollo de la investigación de 1330 horas, se obtiene un total de Bs76,016.20. como se muestra a continuación.

Tabla 23

Costo por hora de trabajo en bolivianos

Horas trabajadas	Pago por hora (en Bs)	Total (en Bs)
1330	57.14	76016.20

Nota: Costo por hora de trabajo en bolivianos considerando salario de Bs8000 *Fuente:* Elaboración propia

El costo de las herramientas utilizadas en el proyecto se detalla en la siguiente tabla.

Tabla 24

Costo de las herramientas

Herramienta	Descripción	Costo (en Bs)
Jupyter Notebook	Desarrollo del modelo computacional	0
Streamlit	Desarrollo del prototipo	0
Vscode	IDE para codificación del prototipo	0
Git y GitHub	Control de versiones	0
	TOTAL	0

Nota: Costos de las herramientas utilizadas en el proyecto. *Fuente:* Elaboración propia.

La mayoría de las herramientas son open source, lo cual ayuda a reducir los gastos además de regirse a la norma de que las entidades públicas (como la ABC) deben usar herramientas de este tipo.

Una vez recabados todos los costos del proyecto, se obtiene un costo final y total de Bs79.415,2

CAPITULO 4.

CONCLUSIONES Y RECOMENDACIONES

4.1.CONCLUSIONES

El presente proyecto demuestra que es viable desarrollar una técnica de rebanado de video capaz de generar imágenes 2D de la distribución de señales de tránsito a partir de grabaciones de carreteras, contribuyendo a potenciar el software de análisis de videos de ECOVIANA S.R.L. Esta técnica permite reducir la información irrelevante contenida en los videos, optimizando el proceso de inventariación de señales y disminuyendo el sesgo del observador, lo que fortalece la confiabilidad de los estudios técnicos realizados por la empresa.

El análisis de los procesos actuales de supervisión vial permitió identificar los puntos críticos en la gestión de videos, especialmente la excesiva cantidad de información redundante y la dependencia de un análisis manual prolongado. Estos hallazgos descritos dejan en evidencian la necesidad de implementar soluciones automatizadas que aumenten la eficiencia y precisión del inventario de señales de tránsito.

El diseño y desarrollo del modelo de rebanado de video proporcionó una metodología sistemática para la extracción automatizada de imágenes 2D relevantes, aplicando herramientas de procesamiento de imágenes y análisis espaciotemporal. La creación del prototipo funcional confirmó la viabilidad técnica del modelo, demostrando que es posible integrar la técnica en el software existente y reducir significativamente el tiempo requerido para la supervisión manual de los videos.

La evaluación cualitativa mediante grupos focales mostró que la técnica mejora la eficiencia y confiabilidad del análisis de señales de tránsito, facilitando la interpretación de la información y respaldando la toma de decisiones técnicas. Asimismo, la documentación detallada de la metodología, los algoritmos y los resultados obtenidos descrita en el marco aplicativo, en subtítulos prototipo y evaluación técnica proporciona una base para futuras mejoras, escalabilidad y aplicación práctica de la técnica dentro de la empresa.

4.2.RECOMENDACIONES

Para mejorar los resultados se recomienda trabajar en dos direcciones:

Mejorar la captura de datos: Utilizar dispositivos mecánicos que permitan estabilicen la captura del video para evitar las vibraciones de la cámara; Usar una cámara más adecuada; Considerar una captura estéreo con dos cámaras; Añadir georreferenciación al video.

Mejorar el preprocesamiento y post procesamiento: Utilizar herramientas software de edición avanzada de videos para estabilizar los videos capturados; Desarrollar herramientas de post procesamiento para que la imagen bidimensional se acerque más a un mapa temático, mejorando la visualización de los datos.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- Sermanet, P., & LeCun, Y. (2011). Traffic Sign Recognition with Multi-Scale Convolutional Networks. En International Joint Conference on Neural Networks, San Jose, California, USA, July 31 – August 5, 2011.
- Alarcón Vargas, A. J. (2023). Detección de la señalización de tránsito vertical con redes neuronales convolucionales basadas en bloques residuales. En publicación
- Heike, Schaumberg. (2022). Temporal Data. Synthesis Lectures on Data Management, 85-102. doi: 10.1007/978-3-031-27177-9_5
- ¿Quiénes somos? - ABC. (n.d.). Retrieved from https://www.abc.gob.bo/?page_id=391
- Aaker, D. A., & Day, G. S. (2007). *Marketing Research*. Hoboken, NJ: John Wiley & Sons.
- ABC; APIA XXI Ingenieros y Arquitectos Constructores. (2018, Septiembre 13). MANUALES TÉCNICOS PARA EL DISEÑO DE CARRETERAS.
- Amazon Web Services. (n.d.). *¿Qué es python?* Retrieved Mayo 4, 2024, from <https://aws.amazon.com/es/what-is/python/#:~:text=Python%20es%20un%20lenguaje%20de,ejecutar%20en%20muchas%20plataformas%20diferentes>.
- Atlassian. (2025, Noviembre 11). *¿Qué es scrum? Una guía para el marco de trabajo ágil*. Retrieved from Atlassian.com.
- Aurora. (2024, Junio 28). *Matplotlib: La herramienta esencial para visualización en Data Science*. Retrieved from <https://iddigitalschool.com/bootcamps/matplotlib-la-herramienta-esencial-para-visualizacion-en-data-science/>
- Ayo-Vaughan, S. (2023, Octubre 17). Retrieved from A Brief History of Data Analytics: <https://adunni.org/2023/10/17/brief-history-of-data/>

- Bcnvision. (2023, Julio 7). *Las 10 mejores aplicaciones de visión artificial en la industria*. Retrieved from <https://bcnvisiongroup.com/es/b205/las-10-mejores-aplicaciones-de-vision-artificial-en-la-industria>
- Beatty, J. (2012). *The Radon Transform and the Mathematics of Medical Imaging*. Colby College.
- Bresenham, J. E. (1965). Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4, 25-30.
- Carrión, J. (2017). *Diferencia entre dato información y conocimiento*.
- Chen, X. H. (2023). *Spatial-Temporal Data Analysis in Nonlinear System*. IntechOpen. doi:10.5772/intechopen.105709
- Churchill, G. A., & Iacobucci, D. (2005). *Marketing Research: Methodological Foundations*. Mason, OH: Thomson/South-Western.
- Contributors, A. C. (n.d.). *Pillow*. Retrieved abril 11, 2024, from <https://pillow.readthedocs.io/en/stable/>
- Cossío, D. N. (2008). CAUSAS DE MUERTE MÁS FRECUENTES EN ACCIDENTES DE TRÁNSITO REGISTRADOS EN LA MORGUE DEL HOSPITAL DE CLÍNICAS DE LA CIUDAD DE LA PAZ EN LA GESTIÓN 2008. La Paz, Bolivia.
- DataCamp. (2024, Abril 29). *Tutorial de Python: Streamlit*. Retrieved from <https://www.datacamp.com/es/tutorial/streamlit>
- Dhar, S. (2016). A novel method for edge detection in a gray image based on human psychovisual phenomenon and Bat algorithm. *International Conference on Advancement of Computer Communication and Electrical Technology*. Murshidabad.
- Doran, C., & Lasenby, A. (2003). *Geometric Algebra for Physicists*. Cambridge.
- Doran, C., & Lasenby, A. (2003). *Geometric Algebra for Physicists*. Cambridge.

Ecoviana S.R.L. (2024). Perfil Corporativo. La Paz.

Editorial Etecé. (2025, Septiembre 25). *Números enteros*. Retrieved from Concepto:
<https://concepto.de/numeros-enteros/>

Gálvez Ortiz, S. A. (2017). DESARROLLO DE UN ALGORITMO DE STITCHING PARA SECUENCIAS DE IMÁGENES CON AMPLIOS MOVIMIENTOS DE CÁMARA. Santiago de Chile, Chile.

García Ruíz, J. M., & Reyes Mora, S. (2016). La Matemática en la que Sustenta la Tomografía Axial Computarizada. Retrieved from
<http://repositorio.utm.mx:8080/jspui/handle/123456789/159>

Gaul, W., Opitz, O., & Schader, M. (2000). *Data Analysis. Scientific Modeling and Practical Application*. Springer Berlin, Heidelberg.

Gillis, A. S., Lutkevich, B., & Hughes, A. (2024, mayo). *Data Management*. Retrieved from TechTarget:
<https://www.techtarget.com/searchdatamanagement/definition/database>

Gonzalez, R. C., & Woods, R. E. (2017). *Digital Image Processing*. Pearson.

Hartshorne, R. (2013). *Geometry: Euclid and Beyond*. Springer Science & Business Media.

Henrik Abellsson, A. (2006, Febrero 11). *File:Bresenham.png*. Retrieved from
<https://commons.wikimedia.org/wiki/File:Bresenham.png>

Jupyter, T. (2015). *Componentes de Jupyter Notebook*. Retrieved from
<https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>

Kinnear, T. C., & Taylor, J. R. (1996). *Marketing Research: An Applied Orientation*. Orlando, FL: Dryden Press.

Laruta Espejo, J. E. (2018, Diciembre). Aprendizaje fin a fin para la conducción autónoma de vehículos domésticos usando visión artificial y redes neuronales convolucionales. La Paz, Bolivia.

- Lia. (2023, Junio 19). *Importancia de la visualización para Big Data*. Retrieved from Hexagon Data: <https://www.hexagondata.com/es/monetizacion-datos/visualizacion-para-big-data/>
- Magdalene, W., Wan, Y., Ching, Z., & Goh, X. (2019). Basic Electromagnetic Theory. In *Analyzing the Physics of Radio Telescopes and Radio Astronomy* (pp. 68-77).
- Marques Barbosa, J. L. (1995). *Geometría euclidiana plana* (Cuarta ed.).
- Marr, D. (2010). *Vision: A Computational Investigation*. The MIT Press.
- Matplotlib development team. (n.d.). *Documentacion*. Retrieved from <https://matplotlib.org/>
- Meno, O. (2023, Septiembre 25). *¿Qué es la metodología Scrumban y cómo utilizarla?* Retrieved from dev.to web site: <https://dev.to/oroscoloyamena/que-es-la-metodologia-scrumban-y-como-utilizarla-3933>
- Merali, Z. (2015). Optics: Super vision. *Nature*, 158-160.
- Merino, C. (n.d.). Retrieved from La Visualización de elementos mejora la forma de aprender: <https://www.pucv.cl/uuaa/ciencias/noticias/la-visualizacion-de-elementos-mejora-la-forma-de-aprender#:~:text=La%20visualizaci%C3%B3n%20es%20una%20capacidad,como%20producto%20de%20esta%20percepci%C3%B3n.>
- Mision y Visión - ABC. (n.d.). Retrieved from https://www.abc.gob.bo/?page_id=393#
- Open CV. (2019, Marzo 6). *Open CV*. Retrieved from About: <https://opencv.org/about/>
- Oracle. (2020, Noviembre 24). *Oracle*. Retrieved from *¿Qué es una base de datos?:* <https://www.oracle.com/ar/database/what-is-database/>
- Palacio, M. (2024). *Scrum Master*.

- Pham, T.-T. (2010). MODELE DE GRAPHE ET MODELE DE LANGUE POUR LA RECONNAISSANCE DE SCENES VISUELLES.
- Pirota, D. M. (2007). LA SEÑALIZACIÓN VIAL Y SU IMPACTO ACTUAL SOBRE EL PRINCIPIO DE CONFIANZA EN LA NORMALIDAD O SEGURIDAD DEL TRÁNSITO. *Carreteras: Revista técnica de la Asociación Española de la Carretera*, 100-102.
- Python Software Foundation. (n.d.). *tkinter — Python interface to Tcl/Tk*. Retrieved Octubre 4, 2025, from <https://docs.python.org/es/3.13/library/tkinter.html>
- Ramírez Giraldo, J. C., Arboleda Clavijo, C., & H. McCollough, C. (2008). Tomografía computarizada por rayos X:fundamentos y actualidad. *Revista Ingeniería Biomédica* .
- Ruiz Huaraz, C. B., Valenzuela Muñoz, A., & Valenzuela Ramos, M. R. (2023). *Introducción Metodología de Investigación*. Lima: Fondo Editorial.
- Sánchez Alberca, A. (2022, Mayo 12). *Aprende con Alf*. Retrieved from La librería Numpy: <https://aprendeconalf.es/docencia/python/manual/numpy/>
- Sánchez, A. (2022). *La librería Numpy*. Retrieved from <https://aprendeconalf.es/docencia/python/manual/numpy/>
- Sanjay, D., Maji, P., Harsh, M., & Washabaugh, K. (2020, Diciembre 18). *NVIDIA Technical Blog*. Retrieved from Developer Nvidia web site: <https://developer.nvidia.com/blog/deploying-a-scalable-object-detection-pipeline-the-inferencing-process-part-2/>
- Schaumberg, H. (2022). Temporal Data. *Synthesis Lectures on Data Management (Synthesis Lectures on Data Management)*, 85-102.
- Schimansky, T. (n.d.). *Official documentation and tutorial*. Retrieved from CustomTkinter: <https://customtkinter.tomschimansky.com/>

- Schwaber, K. (1996). SCRUM Development Process. In *Proceedings of the 10th Annual Conference on Object-Oriented Programming Systems, Languages, and Applications* (pp. 117-134). Springer. doi:10.1007/978-1-4471-0947-1_11
- Serna Ruiz, A., Ros García, F. A., & Rico Noguera, J. C. (2010). *Guía práctica de sensores*. Creaciones copyright SL.
- Significados, E. (2023, Octubre 2). *Qué es la Técnica*. Retrieved Junio 23, 2024, from Significados.com.: <https://www.significados.com/tecnica/>
- The MathWorks Inc. (n.d.). *Introducción a MATLAB*. Retrieved from <https://la.mathworks.com/help/matlab/getting-started-with-matlab.html>
- Tigo. (n.d.). *Planes Movil + Internet*. Retrieved Diciembre 13, 2024, from <https://www.tigo.com.bo/planes#internet-+-movil>
- Toro Bonilla, M. (2022). *Fundamentos de programación: PYTHON*. Sevilla: Universidad de Sevilla.
- Torrenti, Á. (2024, Abril 11). *Procesamiento de imágenes con OpenCV en Python*. Retrieved from <https://imaginaformacion.com/tutoriales/opencv-en-python>
- Urbina Lara, C. I. (2020). DETECCIÓN Y CLASIFICACIÓN DE GRIETAS EN ASFALTO MEDIANTE REDES CONVULCIONALES Y PROCESAMIENTO DE IMÁGENES. Santiago de Chile, Chile.
- Vías Bolivia. (2023, Noviembre 8). RESPETEMOS LAS SEÑALES DE TRÁNSITO. Retrieved from <https://www.facebook.com/ViasBolivia/posts/aprendeconv%C3%ADas-respetemos-las-se%C3%B1ales-de-tr%C3%A1nsitoinstamos-a-los-conductores-del-670462128547491/>
- Video Tramo La Paz - LLujturi. (n.d.).
- Wikipedia contributors. (n.d.). *Wikipedia*. Retrieved from Postulados de Euclides: https://es.wikipedia.org/wiki/Postulados_de_Euclides#Referencias

Xie, B., & Zhang, F. (2022). Design and Implementation of Data Mining in Information Management System. Chickballapur, India.

GLOSARIO

Glosario

- Discretizar:

Dividir un continuo en partes indivisibles o elementos discretos. En otras palabras, transformar algo continuo en algo contable.

- Formalización

Representar un sistema o concepto de manera formal y rigurosa, utilizando un lenguaje específico como las matemáticas o la lógica.

- Reconocimiento automático

La capacidad de un sistema informático de identificar y clasificar objetos, patrones o eventos de manera automática, sin intervención humana.

- Modelado conceptual

La creación de una representación abstracta de un sistema o concepto, utilizando diagramas, símbolos o lenguaje natural.

- Entorno de desarrollo

El software y las herramientas que se utilizan para escribir, compilar, depurar y ejecutar programas informáticos.

- Volumen de información tridimensional

Un volumen tridimensional es una proyección de una porción del mundo espaciotemporal cuatridimensional. Dentro de este proyecto, el volumen de información tridimensional se refiere al volumen creado con los recortes del video.

ANEXOS

Anexos

Anexo A

Carta dirigida a la entidad Administradora Boliviana de Carreteras (ABC)



La Salle
Universidad
Bolivia

Dirección: Av. Jorge Carrasco
esq. Las Palmas
N° 450 (Bolognia)
La Paz - Bolivia
Teléfono/Fax: (+591-2) 2723598
(+591-2) 2723588
www.ulasalle.edu.bo

La Paz, 27 de mayo de 2024
Cite ULS-DA-041/24

Señor
Ing. Mario Sandro Condori Flores
Subgerente de Construcciones a.i
Administradora Boliviana de Carreteras
Presente. -



MIEMBROS DE:
INTERNATIONAL ASSOCIATION
La Salle
UNIVERSITIES

CALIDAD CERTIFICADA
 

DECRETO SUPREMO N° 1079
RESOLUCIÓN MINISTERIAL N° 129/03.

Ref. Solicitud para realizar proyecto de grado en ABC

De nuestra mayor consideración:

A tiempo de saludarlo, permítame, mediante esta nota explicar algunos antecedentes para fundamentar la solicitud de referencia.

La Universidad La Salle de Bolivia, de acuerdo a su misión, se constituye en una institución de educación superior al servicio de la formación integral de jóvenes en las áreas de educación, investigación científica, tecnológica y la interacción social.

La Universidad La Salle en Bolivia reconoce el Proyecto de Grado como una modalidad de graduación. El mismo, consiste en un trabajo escrito que cumple con exigencias de la metodología científica, que describe un conjunto ordenado de recursos y acciones conducentes a generar propuestas aplicables a determinado proceso, unidad, organización, grupo, producto o servicio, para su mejora, innovación, modernización o desarrollo tecnológico. El Proyecto de Grado debe contener elementos propositivos que coadyuven a dar solución a problemas de las instituciones, organismos o empresas interesadas en el Proyecto.

La postulante Rosalia Reynaga Funes, estudiante de la carrera de Ing. de Sistemas, está elaborando un Proyecto que implica técnicas de visualización que permite - a partir de un video de carretera- generar una imagen bidimensional que contenga la distribución de las señales horizontales y verticales presentes en un escenario vial.

Otras investigaciones realizadas en nuestra universidad se han enfocado en el reconocimiento y clasificación de las señales de tráfico presentes, mediante el uso de Inteligencia Artificial. Sobre la base de estos avances, la postulante propone que se pueda mejorar la visualización de los resultados con la finalidad de coadyuvar en la elaboración de los Informes de Inspección de las Carreteras que la ABC realiza.

La investigación se limita al tramo Achica Ariba (Reten Rodaje) - Calamarca (Cr. Rt. A Sapahaqui), y los resultados se mostrarán para el punto crítico: Cruce La Paz - Llujturi. Los postulantes deben presentar una carta de aceptación de una institución interesada en el desarrollo del Proyecto de Grado. Los resultados del proyecto podrán ser utilizados por la Institución interesada.

Sobre la base de lo expuesto, solicitamos a su Autoridad que pueda colaborar con la carta de aceptación como institución interesada en el proyecto

En caso de aceptar la solicitud, solicitamos comunicarse con el Coordinador de la Carrera, Ing. Osamu Yokosaki al celular 62500895 y al teléfono de la interesada Rosalía Reynaga: 79518469.

Agradeciendo su atención, me despido de usted.

Atentamente,


M.Sc. Wilma Peñañel
Directora Académica
Universidad La Salle - Bolivia



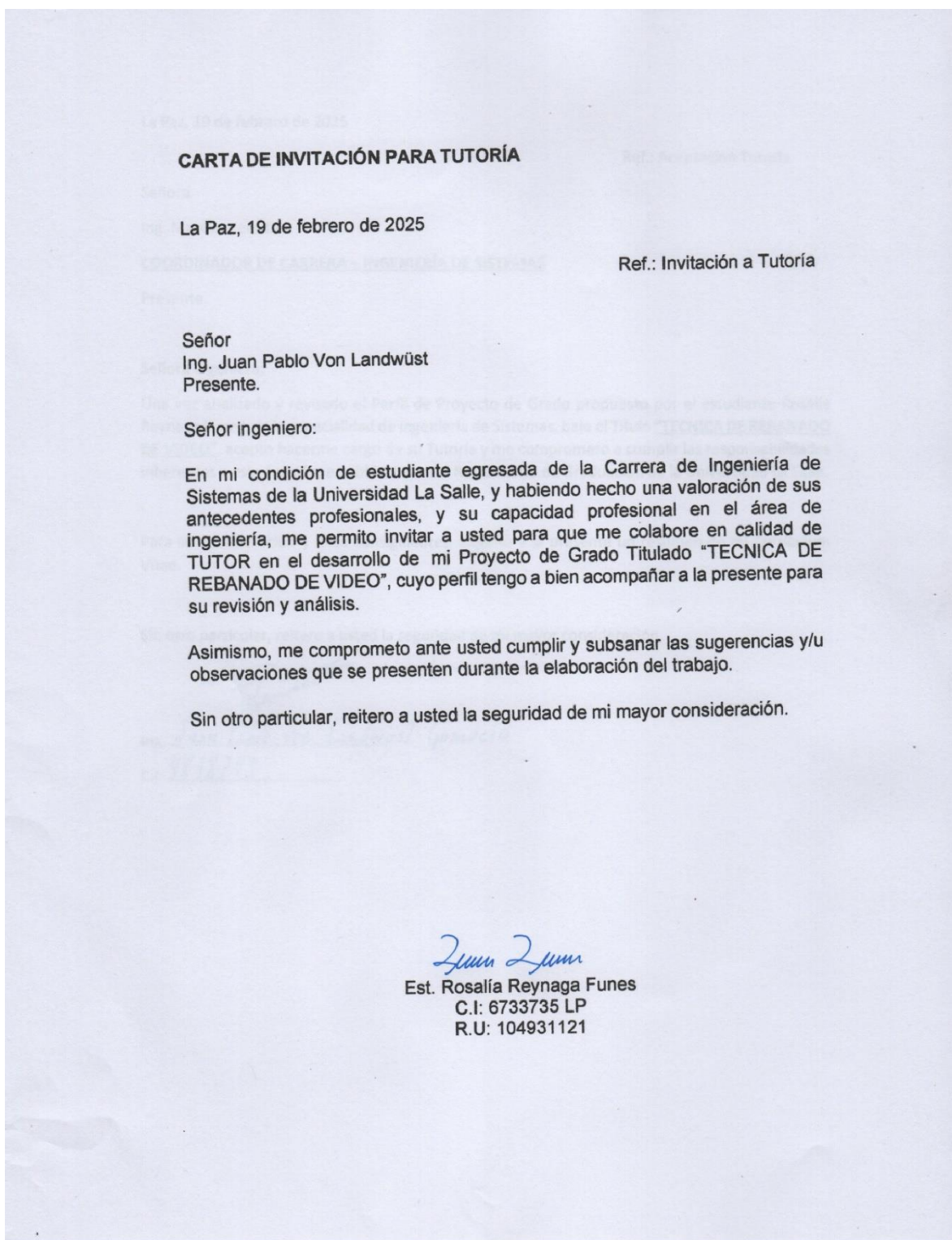
MIEMBROS DE:
INTERNATIONAL ASSOCIATION
La Salle
UNIVERSITIES



DECRETO
SUPREMO
Nº 1079

RESOLUCIÓN
MINISTERIAL
Nº 129/03

Cc/ Archivo

Anexo B: Carta de invitación de tutoría**Anexo C:** Carta de aceptación de tutoría

La Paz, 19 de febrero de 2025

Ref.: Aceptación Tutoría

Señora

Ing. Martha Heredia

COORDINADOR DE CARRERA – INGENIERÍA DE SISTEMAS

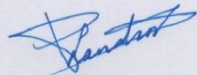
Presente.

Señora Ingeniera:

Una vez analizado y revisado el Perfil de Proyecto de Grado propuesto por el estudiante Rosalía Reynaga Funes, de la especialidad de Ingeniería de Sistemas, bajo el Título "**TECNICA DE REBANADO DE VIDEO**", acepto hacerme cargo de su Tutoría y me comprometo a cumplir las responsabilidades inherentes a esta función, establecidas en el Reglamento de Graduación de la Universidad La Salle.

Para su consideración y fines consiguientes, adjunto a la presente un resumen de mi Curriculum Vitae.

Sin otro particular, reitero a usted la seguridad de mi mayor consideración.


Ing. Juan Pablo von Landwüst Gumucio
C.I. 4878249

FORMULARIO DE CONTROL – PERFIL

Control Perfil			
Estudiante:	Rosalia Reynaga Funes		
Título:	Técnica de rebanado de video basado en algoritmo		
Revisión por:			
Tutor:	☐	Revisor 1:	☐
		Revisor 2:	☐
		IG:	☐
Fecha de entrega:	08/05/24	Fecha de devolución:	
Por favor verifique las fechas antes de la firma del formulario. A continuación, usted debe seleccionar la casilla que corresponda:			
Nº	DETALLE	PRESENTA OBSERVACIÓN	LA OBSERVACIÓN FUE SUBSANADA
1	Título	☐	☐
2	Introducción	☐	☐
3	Antecedentes	☒	☐
4	Planteamiento del problema	☒	☐
5	Identificación del problema	☒	☐
6	Formulación del problema	☒	☐
7	Objetivo general	☐	☐
8	Objetivos específicos	☐	☐
9	Justificaciones	☐	☐
10	Delimitaciones	☐	☐
11	Diseño Metodológico	☒	☐
12	Bibliografía	☐	☐
Nº	DETALLE	Cumple	No cumple
1	Cumplimiento de formato (Numeración, Tipo de letra, Párrafo, Pies de Página, Etc.)	☒	☐
2	Redacción, gramática y ortografía del trabajo	☒	☐
3	Relación Título - Problemática - Objetivos	☐	☒
4	Profundidad y Pertinencia del Trabajo	☒	☐

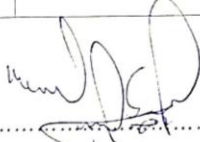
Observaciones: (Para ser llenado de forma Obligatoria)


 FIRMA

CONTROL DE MARCO APLICATIVO			
Estudiante	Reynaga Funes Rosalia		
Título	TÉCNICA DE REBANADO DE VIDEO BASADO EN ALGORITMOS DE PROCESAMIENTO DE IMÁGENES		
Revisado Por	Ing. K. Marlon Socz Hamani		
Tutor:	Revisor1:	Revisor2:	Docente TG:
Fecha de Entrega	26/09/2024	Fecha de Devolución	27/09/2024
Por favor verifique las fechas antes de la firma del formulario. A continuación, usted debe seleccionar la casilla que corresponda:			

Nº	DETALLE	PRESENTA OBSERVACION	LA OBSERVACIÓN FUE SUBSANADA
1	Planificación		
2	Exploratoria	✓	
3	Modelo Matemático	✓	

Nº	DETALLE	Cumple	No cumple
1	Cumplimiento de formato (Numeración, Tipo de letra, Párrafo, Pies de Página, Etc.)	☒	☐
2	Redacción, gramática y ortografía del trabajo	☒	☐
3	Profundidad y Pertinencia del Trabajo	☒	☐



 Ing. K. Marlon Socz M.

95/100

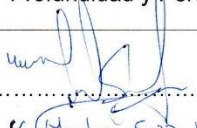
Observaciones	Recomendaciones
- No suele ser adecuado mostrar los códigos completos en el documento principal sino en anexos. - Enfocar el desarrollo matemático en los tópicos principales de tu investigación.	- Considerar la utilización de pseudo-código en vez de lenguaje de programación. - Considerar limitar los conceptos matemáticos o reubicarlos.

Anexo F Bitácoras de revisión

CONTROL DE MARCO APLICATIVO			
Estudiante	Reynaga Funes Rosalía		
Título	TÉCNICA DE REBANADO DE VIDEO BASADO EN ALGORITMOS DE PROCESAMIENTO DE IMÁGENES		
Revisado Por	Ing. Marlon Soza M.		
Tutor: 	Revisor1:	Revisor2:	Docente TG:
Fecha de Entrega	11/12/2024	Fecha de Devolución	11/12/2024
Por favor verifique las fechas antes de la firma del formulario. A continuación, usted debe seleccionar la casilla que corresponda:			

Nº	DETALLE	PRESENTA OBSERVACION	LA OBSERVACIÓN FUE SUBSANADA
1	Planificación		☒
2	Exploratoria		☒
3	Modelo Matemático		☒
4	Modelo Computacional		☒
5	Prototipo		☒
6	Evaluación Técnica		☒
7	Evaluación Económica		☒

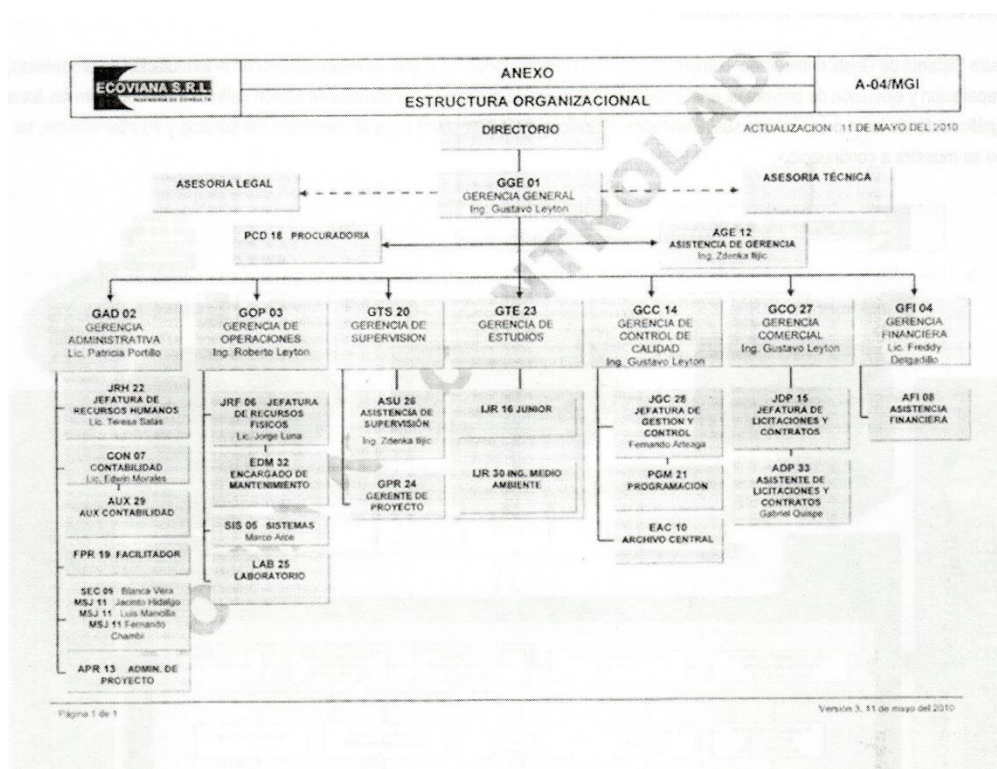
Nº	DETALLE	Cumple	No cumple
1	Cumplimiento de formato (Numeración, Tipo de letra, Párrafo, Pies de Página, Etc.)	☒	☐
2	Redacción, gramática y ortografía del trabajo	☒	☐
3	Profundidad y Pertinencia del Trabajo	☒	☐

Ing.  Soza M.

100/100

Observaciones	Recomendaciones

Anexo G: Organigrama de la empresa ECOVIANA S.R.L



Anexo H: Cartas de ECOVIANA S.R.L**Anexo I: Carta de avala de resultados de parte de ECOVIANA S.R.L**



La Paz, 20 de abril de 2025

Señores
Universidad La Salle
Carrera de Ingeniería de Sistemas
Presente.-

REF.: CARTA DE AVAL DE RESULTADOS DEL PROYECTO DE GRADO

Distinguidas autoridades:

Nos es grato dirigirnos a ustedes para informar lo siguiente:

- La postulante **Rosalía Reynaga Funes** ha concluido el desarrollo de su Proyecto de Grado titulado **"TÉCNICA DE REBANADO DE VIDEO"** en nuestra institución, iniciado el 3 de febrero de 2025 y finalizado el 20 de abril de 2025.
- La técnica desarrollada por la postulante presenta un enfoque innovador de segmentación de video que mejora la visualización y el análisis de señales de tráfico en escenarios viales.
- El trabajo proporciona una sólida base teórica que permite el diálogo transdisciplinario entre diversas áreas de la ingeniería, particularmente en lo relacionado con modelos espaciotemporales y aplicaciones de inteligencia artificial en el ámbito de la seguridad vial.
- Las pruebas realizadas en nuestra institución han permitido obtener, a partir de los videos, imágenes que **reducen significativamente la redundancia de datos**, facilitando una mejor descripción de la distribución de señales en el entorno vial. Esto contribuye directamente a la eficiencia en los procesos de supervisión de infraestructura vial.

Por lo expuesto, **expresamos nuestra conformidad con el trabajo desarrollado** y otorgamos el **aval correspondiente** para que la postulante Rosalía Reynaga Funes, estudiante de la carrera de Ingeniería de Sistemas de la Universidad La Salle – Bolivia, continúe con los trámites administrativos respectivos ante dicha Casa de Estudios.

Sin otro particular, nos despedimos con nuestras mejores consideraciones.

Atentamente,

Ing. 
GERENTE
ECOVIANA SRL

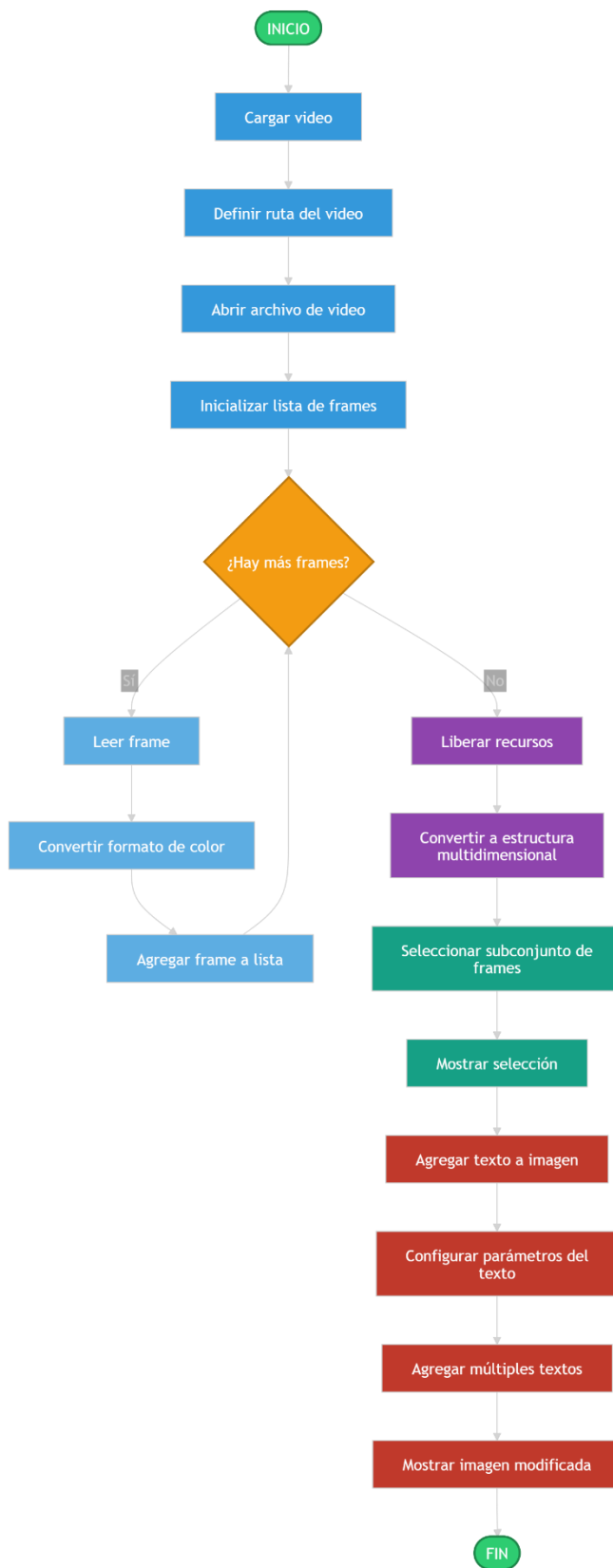


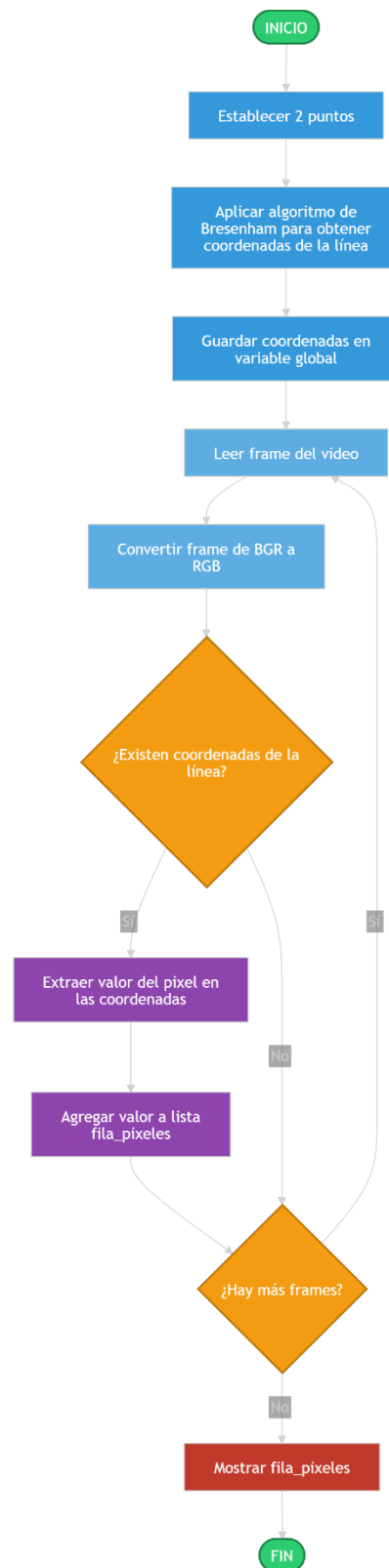
Av. Mario Mercado Bajo Llojeta Edif. Mirador Jardín N° 84 • Central Piloto: (591-2) 2500331
e-mail: correo@ecoviana.com.bo – www.ecoviana.com.bo • La Paz - Bolivia

```

1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 # Ruta de la imagen de entrada
6 input_image_path = 'framePrueba.jpg'
7
8 # Leer la imagen
9 image = cv2.imread(input_image_path)
10 if image is None:
11     print("Error al cargar la imagen.")
12     exit()
13
14 # Convertir la imagen de BGR a RGB para mostrarla correctamente en Matplotlib
15 image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
16
17 # Extraer dimensiones de la imagen
18 h, w = image.shape[:2]
19
20 # Puntos de referencia
21 src = np.float32([[50, h],
22                  [1000, h],
23                  [460+200+5, 700+150-55],
24                  [535+200+5, 700+150-55]])
25
26 # Puntos de destino
27 dst = np.float32([[50, h],
28                  [1000, h],
29                  [50, 100],
30                  [1000, 100]])
31
32 # Dibujar puntos de referencia en la imagen original (color rojo)
33 for point in src:
34     point_int = tuple(map(int, point))
35     cv2.circle(image_rgb, point_int, 15, (255, 0, 0), -1) # Rojo (BGR)
36
37 # Obtener la matriz de transformación
38 M = cv2.getPerspectiveTransform(src, dst)
39
40 # Aplicar la transformación de perspectiva
41 img_perspectiva = cv2.warpPerspective(image, M, (w, h), flags=cv2.INTER_LINEAR)
42
43 # Convertir la imagen transformada de BGR a RGB para mostrarla correctamente en Matplotlib
44 img_perspectiva_rgb = cv2.cvtColor(img_perspectiva, cv2.COLOR_BGR2RGB)
45
46 # Dibujar puntos de destino en la imagen transformada (color azul)
47 for point in dst:
48     point_int = tuple(map(int, point))
49     cv2.circle(img_perspectiva_rgb, point_int, 15, (0, 0, 255), -1) # Azul (BGR)
50
51 # Mostrar la imagen original con puntos y la imagen con perspectiva cambiada
52 plt.figure(figsize=(12, 6))
53
54 plt.subplot(1, 2, 1)
55 plt.imshow(image_rgb)
56 plt.title('Imagen Original con Puntos')
57
58
59 plt.subplot(1, 2, 2)
60 plt.imshow(img_perspectiva_rgb)
61 plt.title('Imagen con Perspectiva Cambiada')
62
63 plt.show()
64

```

Anexo K: Diagrama de flujo del algoritmo sprint 1

Anexo L: Diagrama de flujo**Anexo M:** Diagrama de flujo prototipo optimizado

