



**UNIVERSIDAD
LIBRE®**

PROYECTO 001

**Reconocimiento de placas para la
automatización de
servicios**

PRESENTADO POR:
David Quintero Quijano
Luis Enrique García Barrera
Brian Alberto García Macias

Docente: Mg. Marcela Cifuentes Velásquez
Ingeniería de Sistemas

Universidad Libre – Sede el
Bosque
Gestión de proyectos
Bogotá - Colombia
01 septiembre 2025

Contenido

Objetivos	2
Objetivo general	2
Objetivo específico.....	2
Alcance.....	2
Limites y delimitaciones	2
Metodología	3
Enfoque	3
Procesos	3
Etapas de Inicio.....	5
• Objetivo general	5
• Objetivos específicos.....	5
• Alcance	5
• Antecedentes y necesidad.....	5
• Costos y duración estimada	5
Etapas de Planificación.....	5
Definición de Objetivos.....	5
Identificación del Público Objetivo	5
Análisis de la Competencia	6
Requisitos Funcionales y Técnicos	6
Creación de un Plan de Proyecto.....	6
Herramientas y Recursos	6
Propuesta al Cliente	6
Resultado de la Planeación.....	6

Objetivos

Objetivo general

Desarrollar un Software que permita la captura automática de placas vehiculares, su lectura mediante técnicas de visión por computador, la obtención de su ubicación geográfica y el almacenamiento de la información en una base de datos para su posterior consulta.

Objetivo específico

- Analizar las necesidades y requisitos técnicos, operacionales y de seguridad para el desarrollo de un sistema de gestión de reconocimiento vehicular. Este análisis tiene como fin determinar las características y funcionalidades necesarias para el diseño e implementación de una solución eficaz y eficiente.
- Diseñar una red neuronal que sea capaz de detectar e identificar placas de vehículos mediante el Reconocimiento Óptico de Caracteres, junto con una base de datos para el almacenamiento seguro de la información y una interfaz web que permita la búsqueda y visualización de registros en lista y en mapa.
- Codificar e integrar un servicio de geolocalización para asociar coordenadas GPS a cada registro. Junto con sus respectivas redes neuronales, bases de datos e interfaces web
- Probar los modulos, aplicativos, integridades de las bases de datos y seguridad de las API's; verificando el correcto funcionamiento del producto final.
- Desplegar de manera satisfactoria según los requerimientos del cliente el aplicativo y hardware necesario para el correcto funcionamiento de esta.

Alcance

Limites y delimitaciones

- Incluye:
 - Captura de imágenes/video.
 - Reconocimiento automático de placas.
 - Obtención de ubicación GPS o estática.
 - Registro en base de datos.
 - Consulta por web con visualización en lista y mapa.
- No incluye: o
 - Control físico de acceso vehicular (barreras o semáforos).
 - Integración con sistemas de tránsito externos.
 - Procesamiento de video en la nube en tiempo real.

Metodología

Enfoque

Para el desarrollo del presente proyecto se empleará la **metodología ágil Scrum**, la cual es un marco de trabajo orientado a la gestión de proyectos de software de manera iterativa e incremental. Scrum permite organizar el trabajo en ciclos cortos de tiempo llamados **sprints**, que generalmente tienen una duración de dos a cuatro semanas.

La elección de Scrum se debe a que proporciona **flexibilidad, adaptabilidad y retroalimentación constante** por parte de los interesados, lo que garantiza que el producto final responda de manera adecuada a los cambios en los requerimientos y a las necesidades del cliente.

Entre las principales características de Scrum se destacan:

- **Iteraciones cortas (sprints):** cada ciclo entrega una versión funcional del producto.
- **Priorización del producto:** mediante el **Product Backlog**, donde se ordenan las funcionalidades según su importancia.
- **Roles definidos:**
 - *Product Owner:* representa al cliente y define las prioridades.
 - *Scrum Master:* facilita el proceso y elimina impedimentos.
 - *Equipo de desarrollo:* responsable de implementar las funcionalidades.
- **Eventos principales:** planificación del sprint, reuniones diarias (Daily Scrum), revisión y retrospectiva.
- **Artefactos de Scrum:** Product Backlog, Sprint Backlog e Incremento del producto.

Scrum se considera una metodología adecuada para este proyecto porque permite entregar valor al cliente en etapas tempranas, mejorar la comunicación del equipo, reducir riesgos y adaptarse a cambios en los requisitos de manera más eficiente que metodologías tradicionales como Cascada.

Procesos

El desarrollo del proyecto se organiza siguiendo la metodología ágil Scrum, pero adaptado a las fases clásicas de gestión de proyectos (inicio, análisis, planificación, ejecución, monitoreo y cierre). Los procesos definidos son los siguientes:

1. Fase de Inicio

En esta etapa se establecen los fundamentos del proyecto y se definen los interesados.

Las actividades principales incluyen:

- Elaboración del acta de inicio.
- Identificación de interesados y análisis de expectativas.
- Conformación del equipo de trabajo y definición de roles de Scrum (Product Owner, Scrum Master, Equipo de desarrollo).
- Elaboración de la matriz RACI.
- Kick-off meeting para presentar oficialmente el proyecto.

2. Fase de Análisis

El propósito de esta fase es comprender en detalle los requerimientos del cliente y transformarlos en insumos que alimenten el backlog del producto. Las actividades principales son:

- **Levantamiento de requisitos iniciales:** entrevistas y reuniones con los interesados.
- **Elaboración de historias de usuario:** conversión de requisitos en descripciones claras con criterios de aceptación.
- **Análisis de viabilidad técnica:** revisión de tecnologías, recursos y limitaciones del sistema.
- **Priorización de requisitos:** clasificación de funcionalidades mediante la técnica MoSCoW.
- **Definición del Product Backlog inicial:** consolidación de las historias de usuario priorizadas.
- **Modelado de procesos principales:** representación mediante casos de uso, diagramas de flujo y arquitectura conceptual.

3. Fase de Planificación

Se construye la estrategia para organizar el trabajo en sprints y gestionar recursos:

- Elaboración de la EDT (WBS).
- Definición del cronograma de sprints.
- Estimación de costos y recursos.
- Plan de calidad, riesgos, comunicación y adquisiciones.

4. Fase de Ejecución – Desarrollo por Sprints

El desarrollo se organiza en ciclos iterativos (**sprints**) de dos semanas, con entregables incrementales:

- **Sprint 1:** Dataset y Arquitectura.
 - **Sprint 2:** Red Neuronal y OCR.
 - **Sprint 3:** Integración y Plataforma Web.
- Cada sprint finaliza con revisión de producto y retrospectiva para mejorar el siguiente ciclo.

5. Fase de Monitoreo y Control

Se realizan actividades de seguimiento y control para garantizar el cumplimiento de los objetivos:

- Reuniones de control quincenales y reuniones diarias (Daily Scrum).
- Informes de avance (status report).
- Control de calidad de entregables.
- Gestión de cambios aprobados por el Product Owner.

6. Fase de Cierre

En la última etapa se consolidan los resultados y se formaliza la entrega al cliente:

- Pruebas de aceptación final.
- Capacitación al personal encargado.
- Entrega de documentación técnica y de usuario.
- Cierre formal del proyecto.

Etapa de Inicio

En esta fase se definen los fundamentos del proyecto, sus objetivos y el alcance inicial. El proyecto consiste en el desarrollo de un **software de reconocimiento de placas vehiculares**, enfocado en optimizar el control de accesos. El **acta de inicio** (código: AIPUL 0000, versión 002, 01/07/2024) establece:

- **Objetivo general:** crear un sistema que capture placas automáticamente, procese la información con técnicas de visión por computador, asocie coordenadas GPS y almacene los datos en una base de datos consultable desde una interfaz web.
- **Objetivos específicos:** abarcan el análisis de requisitos técnicos y de seguridad, el diseño de una red neuronal con OCR, la integración de GPS y base de datos, la construcción de una interfaz web, pruebas de precisión y la capacitación al cliente.
- **Alcance:** incluye captura de imágenes, reconocimiento de placas, geolocalización y consulta web. Excluye control físico de accesos, integración con sistemas externos y procesamiento en la nube en tiempo real.
- **Antecedentes y necesidad:** la empresa no cuenta con un sistema automatizado, lo que genera retrasos y errores. Se identificó la oportunidad de implementar una solución moderna alineada con las tendencias de mercado en IA y visión por computador.
- **Costos y duración estimada:** \$570.000 COP en hardware y servicios básicos, con una duración de aproximadamente 10 semanas de desarrollo distribuidas en análisis, implementación y pruebas.

El resultado de esta fase es un documento formal que valida la importancia del proyecto y sirve de base para las fases siguientes.

Etapa de Planificación

En esta fase se detallan los planes necesarios para ejecutar el proyecto de manera ordenada, asegurando cumplimiento de tiempos, costos y calidad.

Definición de Objetivos

Se reafirma el objetivo de **automatizar el reconocimiento de placas** para SecureVision, con énfasis en mejorar seguridad, reducir errores humanos y aumentar la eficiencia operativa.

Identificación del Público Objetivo

El sistema está dirigido a:

- Empresas de seguridad y logística.
- Conjuntos residenciales.
- Zonas de acceso restringido o de alta seguridad.
- Entidades que requieren monitoreo vehicular en tiempo real.

Análisis de la Competencia

En el mercado existen soluciones de terceros basadas en visión por computador, sin embargo, suelen ser costosas, depender de servicios en la nube o no adaptarse al contexto local. Este proyecto busca ser una alternativa más **económica, adaptable y controlada internamente**.

Requisitos Funcionales y Técnicos

- **Funcionales:** captura de imágenes, reconocimiento automático de placas, geolocalización, registro en base de datos, consulta web (lista y mapa).
- **Técnicos:** red neuronal convolucional (CNN) para detección, OCR para reconocimiento de caracteres, módulo GPS NEO-6M, base de datos MySQL/PostgreSQL, interfaz web con HTML/CSS/JS y API REST para integración.

Creación de un Plan de Proyecto

El plan organiza las actividades en fases y sprints:

- **Semana 1:** análisis y planificación.
- **Semanas 2–4:** desarrollo del sistema de reconocimiento (modelo de IA + OCR).
- **Semanas 5–6:** integración de GPS y base de datos.
- **Semanas 7–8:** desarrollo de interfaz web.
- **Semana 9:** pruebas y ajustes.
- **Semana 10:** capacitación y cierre.

Herramientas y Recursos

- **Hardware:** Raspberry Pi 4, cámara USB HD, módulo GPS.
- **Software:** Python, TensorFlow/Keras, OpenCV, MySQL/PostgreSQL, API Google Maps.
- **Gestión:** metodología Scrum con sprints, reuniones de revisión y retrospectiva.

Propuesta al Cliente

La propuesta ofrece un sistema robusto, adaptable a diferentes entornos y con un costo total de **\$570.000 COP**, significativamente menor a las soluciones comerciales disponibles.

Resultado de la Planeación

Se obtiene un **plan de trabajo detallado**, un backlog inicial priorizado y una proyección clara de entregables por sprint. Con esto, el equipo de desarrollo y el cliente cuentan con un mapa de ruta para asegurar el éxito del proyecto.

CONTROL DE VERSIONES										
Versión		Revisada por	Aprobada por	Fecha	Motivo:					
0.1		DT	DT	16/08/25	Version Original					
LISTA DE ACTIVIDADES										
NOMBRE DEL PROYECTO					SIGLAS DEL PROYECTO					
ACTIVIDAD	ID	EDT	RECURSOS	DESCRIPCIÓN	PREDECESORAS	SUCESORAS	ADELANTO	RETRASO	RELACIÓN DE PRECEDENCIA	DURACION
Fase de Analisis										
Levantamiento de requisitos iniciales	1	0	Product Owner, Cliente	Reuniones para identificar necesidades y objetivos del cliente.	-	2, 3	-	-	Inicio	3 días
Historias de usuario	2	0.1	Product Owner, Equipo de desar	Transformar requisitos en historias de usuario con criterios de aceptación.	-	4	-	-	FS	2 días
Análisis de viabilidad técnica	3	0.2	Equipo de desarrollo, Arquitecto	Evaluación de tecnologías (dataset, CNN, OCR, API, etc.) y recursos disponibles.	-	4, 6	-	-	FS	2 días
Priorización de requisitos	4	0.3	Product Owner, Cliente	Uso de MoSCoW para priorizar funcionalidades según valor de negocio y viabilidad.	2, 3	5	-	-	FS	1 día
Definición del Product Backlog inicial	5	0.4	Product Owner, Scrum Master	Consolidación del backlog inicial con historias de usuario priorizadas.	-	-	-	-	FS	1 día
Modelado de procesos principales	6	0.5	Equipo de desarrollo, Analista	Elaboración de casos de uso, diagramas de flujo y arquitectura conceptual.	-	-	-	-	FS	2 días
Fase de Inicio										
Elaborar acta de inicio	7	1.1	PM, Patrocinadores	Definir objetivos, alcance preliminar, entregables.	-	2	-	-	FS	2 días
Identificación de interesados	8	1.2	PM	Mapear stakeholders y su influencia.	-	1	3	-	FS	2 días
Análisis de expectativas	9	1.3	PM	Reuniones con interesados clave.	-	2	4	-	FS	2 días
Conformación del equipo	10	1.4	PM, Devs, QA	Selección y asignación de roles.	-	3	5	-	FS	1 día
Elaboración de la matriz RACI	11	1.5	PM	Definir responsabilidades de cada rol.	-	4	6	-	FS	1 día
Kick-off meeting (reunión de inicio)	12	1.6	PM, Equipo, Cliente	Reunión inicial de alineación.	-	5	7	-	FS	1 día
Fase de Planificación										
Definir backlog inicial de producto	13	2.1	PO, PM, Equipo	Identificación de funcionalidades y requerimientos.	-	6	8	-	FS	3 días
Priorización MoSCoW	14	2.2	PO, PM	Clasificar funcionalidades.	-	7	9	-	FS	1 día
Elaboración de EDT (WBS)	15	2.3	PM	Desglose jerárquico del proyecto.	-	8	10	-	FS	2 días
Definición del cronograma	16	2.4	PM	Definir duración, dependencias.	-	9	11	-	FS	3 días
Estimación de costos	17	2.5	PM	Costos de hardware, software, horas-hombre.	-	10	12	-	FS	2 días
Plan de calidad	18	2.6	PM, QA	Estándares de pruebas y métricas.	-	11	13	-	FS	2 días
Plan de riesgos	19	2.7	PM	Identificar y planear mitigación.	-	12	14	-	FS	2 días
Plan de comunicación	20	2.8	PM	Canales y frecuencia de reuniones.	-	13	15	-	FS	1 día
Plan de adquisiciones	21	2.9	PM	Estrategia de compra de equipos.	-	14	16	-	FS	2 días
Ejecución – Sprint 1 (Dataset y Arquitectura)										
Preparación del dataset de placas	22	3.1.1	Dev IA	Recolección de imágenes, etiquetado.	-	15	17	-	FS	1 semana
Preprocesamiento de imágenes	23	3.1.2	Dev IA	Normalizar, ajustar brillo, eliminar ruido.	-	16	18	-	FS	3 días
Diseño de la arquitectura del sistema	24	3.1.3	Devs	Definir arquitectura software y hardware.	-	17	19	-	FS	1 semana
Revisión de sprint 1 con cliente	25	3.1.4	PM, Cliente	Validar avances iniciales.	-	18	20	-	FS	1 día
Retrospectiva sprint 1	26	3.1.5	Scrum Master, Equipo	Identificar mejoras.	-	19	21	-	FS	1 día
Ejecución – Sprint 2 (Red Neuronal + OCR)										
Diseño de red CNN	27	3.2.1	Dev IA	Selección de capas y parámetros.	-	20	22	-	FS	3 días
Entrenamiento de la red	28	3.2.2	Dev IA	Entrenar con dataset.	-	21	23	-	FS	2 semanas
Validación de precisión	29	3.2.3	QA, Dev IA	Evaluar métricas.	-	22	24	-	FS	4 días
Implementación de OCR	30	3.2.4	Backend Dev	Integrar OCR (Tesseract/EasyOCR).	-	23	25	-	FS	5 días
Ajuste hiperparámetros	31	3.2.5	Dev IA	Refinar modelo para mayor precisión.	-	24	26	-	FS	3 días
Revisión de sprint 2	32	3.2.6	Cliente, Equipo	Validación de resultados IA.	-	25	27	-	FS	1 día
Retrospectiva sprint 2	33	3.2.7	Equipo	Evaluar mejoras internas.	-	26	28	-	FS	1 día
Ejecución – Sprint 3 (Integración y Web)										
Compra e instalación de hardware	34	3.3.1	PM, Equipo	Raspberry, cámaras, GPS.	-	27	29	-	FS	1 semana
Configuración de base de datos	35	3.3.2	Dev Backend	Almacenar registros en DB.	-	28	30	-	FS	1 semana
Desarrollo de API REST	36	3.3.3	Backend Dev	Endpoints para consulta de placas.	-	29	31	-	FS	1 semana
Diseño de interfaz web	37	3.3.4	Frontend Dev	Pantallas: login, dashboard, mapas.	-	30	32	-	FS	1 semana
Integración con mapas	38	3.3.5	Frontend Dev	Google Maps o Leaflet.	-	31	33	-	FS	4 días
Pruebas de usabilidad	39	3.3.6	QA, Cliente	Validar navegación y experiencia.	-	32	34	-	FS	3 días
Revisión sprint 3	40	3.3.7	Cliente, PM	Validar entregables.	-	33	35	-	FS	1 día
Retrospectiva sprint 3	41	3.3.8	Equipo	Mejoras para siguiente iteración.	-	34	36	-	FS	1 día
Monitoreo y Control										
Reuniones de control quincenales	42	4.1	PM, Equipo	Evaluar estado, riesgos, costos.	-	14	37	-	SS	2 h/quincena
Informes de avance (status report)	43	4.2	PM	Reportes al cliente/patrocinador.	-	36	38	-	SS	Cada 2 semanas
Control de calidad	44	4.3	QA, Devs	Verificar cumplimiento de métricas.	-	33	39	-	FS	1 semana
Gestión de cambios	45	4.4	PM, Equipo	Manejo de solicitudes de cambio.	-	38	40	-	SS	Durante todo el proyecto
Cierre										
Pruebas de aceptación final	46	5.1	Cliente, QA	Validación de entregables finales.	-	39	41	-	FS	3 días
Capacitación al cliente	47	5.2	PM, Dev	Entrenar en uso del sistema.	-	40	42	-	FS	1 semana
Documentación final	48	5.3	PM, Equipo	Manual técnico, usuario, lecciones aprendidas.	-	41	43	-	FS	3 días
Cierre formal del proyecto	49	5.4	PM, Patrocinadores	Firma de aceptación y liberación de recursos.	-	42	-	-	FS	2 días

Detector de placas

Título del Proyecto

Gerente del Proyecto

Fecha inicio

08/16

Fecha Fin

10/25

del proyecto en días

71

WBS No.	Nombre de la tarea	Estado	Asignado a	Fecha inicio	Fecha fin	Duración (en días)
1	Fase de Inicio			08/16	08/23	8
1.1	Elaborar acta de inicio	Complete		08/16	08/19	4
1.2	Identificación de interesados	Complete		08/16	08/20	5
1.3	Análisis de expectativas	Complete		08/16	08/21	6
1.4	Conformación del equipo	Complete		08/16	08/23	8
1.5	Elaboración de la matriz RACI	Complete		08/16	08/23	8
1.6	Kick-off meeting (reunión de inicio)	Complete		08/16	08/16	1
2	Fase de Planificación			08/24	08/30	7
2.1	Definir backlog inicial de producto	In Progress		08/24	08/24	1
2.2	Priorización MoSCoW	In Progress		08/25	08/27	3
2.3	Elaboración de EDT (WBS)	In Progress		08/25	08/27	3
2.4	Definición del cronograma (Gantt)	In Progress		08/25	08/27	3
2.5	Estimación de costos	In Progress		08/25	08/27	3
2.6	Plan de calidad	In Progress		08/28	08/30	3
2.7	Plan de riesgos	In Progress		08/28	08/30	3
2.8	Plan de comunicación	In Progress		08/28	08/30	3
2.9	Plan de adquisiciones	In Progress		08/28	08/30	3
3	Ejecución - Sprint 1 (Dataset y Arquitectura)			08/31	09/13	14
3.1.1	Preparación del dataset de placas	Not Started		09/01	09/02	2
3.1.2	Preprocesamiento de imágenes	Not Started		09/02	09/03	2
3.1.3	Diseño de la arquitectura del sistema	Not Started		09/04	09/04	1
3.1.4	Revisión de sprint 1 con cliente	Not Started		09/05	09/05	1
3.1.5	Retrospectiva sprint 1	Not Started		09/07	09/13	7
4	Ejecución - Sprint 2 (Red Neuronal + OCR)			09/14	09/27	14
4.1	Diseño de red CNN	Not Started		09/14	09/17	4
4.2	Entrenamiento de la red	Not Started		09/17	09/22	6
4.3	Validación de precisión	Not Started		09/22	09/23	2
4.4	Implementación de OCR	Not Started		09/23	09/24	2
4.5	Ajuste hiperparámetros	Not Started		09/24	09/25	2
4.6	Revisión de sprint 2	Not Started		09/25	09/26	2
4.7	Retrospectiva sprint 2	Not Started		09/26	09/27	2
5	Ejecución - Sprint 3 (Integración y Web)			09/28	10/11	14
5.1	Compra e instalación de hardware	Not Started		09/28	10/03	6
5.2	Configuración de base de datos	Not Started		09/28	10/03	6
5.3	Desarrollo de API REST	Not Started		09/28	10/03	6
5.4	Diseño de interfaz web	Not Started		09/28	10/03	6
5.5	Integración con mapas	Not Started		10/06	10/11	6
5.6	Pruebas de usabilidad	Not Started		10/06	10/11	6
5.7	Revisión sprint 3	Not Started		10/06	10/11	6
5.8	Retrospectiva sprint 3	Not Started		10/06	10/11	6
6	Ejecución - Sprint 3 (Integración y Web)			08/16	09/25	41
6.1	Reuniones de control quincenales	Not Started		03/03	03/26	24
6.2	Informes de avance (status report)	Not Started		03/03	03/26	24
6.3	Control de calidad	Not Started		03/03	03/26	24
6.4	Gestión de cambios	Not Started		03/03	03/26	24
7	Ejecución - Sprint 3 (Integración y Web)			10/12	10/25	14
7.1	Pruebas de aceptación final	Not Started		10/13	10/17	5
7.2	Capacitación al cliente	Not Started		10/17	10/22	6
7.3	Documentación final	Not Started		10/22	10/24	3
7.4	Cierre formal del proyecto	Not Started		09/24	09/25	2

MATRIZ RACI SIMPLE



PRIORID ESTADO RESULTADO/ACTIVIDAD DEL PROYECTO

		FASE 1																	
Media	Completo	Elaborar acta de inicio			R														
Media	Completo	Identificación de interesados	R					A											
Media	Completo	Análisis de expectativas			R			C											
Media	Completo	Conformación del equipo			R			I											
Media	Completo	Elaboración de la matriz RACI			R														
Media	Completo	Kick-off meeting (reunión de inicio)			R														
		FASE 2																	
Media	Completo	Definir backlog inicial de producto		R															
Media	Completo	Priorización MoSCoW		R															
Media	Completo	Elaboración de EDT (WBS)		R															
Media	Completo	Definición del cronograma						R											
Alta	Completo	Estimación de costos			R														
Media	Completo	Plan de calidad						R											
Media	Completo	Plan de riesgos			R														
Media	Completo	Plan de comunicación			R														
Media	Completo	Plan de adquisiciones			R														
		FASE 3																	
Alta	En curso	Preparación del dataset de placas		I	I			C	C	C	R		I	I					
Alta	En curso	Preprocesamiento de imágenes		I	I			C	C	C	R		I	I					
Alta	En curso	Diseño de la arquitectura del sistema							R										
Media	En curso	Revisión de sprint 1 con cliente	C	A	R			C	I	I	I		I	I					
Media	En curso	Retrospectiva sprint 1		R				R	C	C	C		I	I					
		FASE 4																	
Media	No se ha iniciado	Diseño de red CNN		C	I			R	C	C	A		I						
Media	No se ha iniciado	Entrenamiento de la red		C	I			R	C	C	A		I						
Media	No se ha iniciado	Validación de precisión		C	I			R	C	C	A		I						
Media	No se ha iniciado	Implementación de OCR		C	I			R	C	C	A		I						
Media	No se ha iniciado	Ajuste hiperparámetros		C	I			R	C	C	A		I						
Media	No se ha iniciado	Revisión de sprint 2	C	A	R			C	I	I	I		I	I					
Media	No se ha iniciado	Retrospectiva sprint 2		R				R	C	C	C		I	I					
		FASE 5																	

Completo	Baja
En curso	Media
Necesita revisión	Alta
Aprobado	
No se ha iniciado	
Atrasado	

INFORME DE DESARROLLO DEL PROYECTO

El desarrollo del proyecto “**Reconocimiento de Placas para la Automatización de Accesos**” se llevó a cabo aplicando la metodología ágil **Scrum**, que permitió dividir el trabajo en entregas iterativas y controladas denominadas *sprints*. A lo largo de cada sprint se establecieron objetivos concretos, se midieron avances y se realizaron reuniones de seguimiento con el cliente para garantizar la calidad de los entregables. El proceso se ejecutó de manera ordenada y progresiva, partiendo de la recolección de datos hasta la integración total del sistema, finalizando con una validación exitosa del funcionamiento del producto final.

ETAPA 1. SPRINT 1 – DATASET Y ARQUITECTURA

Durante esta primera fase se sentaron las bases técnicas del sistema. El principal objetivo fue la **construcción y depuración del conjunto de datos (dataset)**, así como la definición de la arquitectura general del software y el hardware.

Para la creación del dataset se recopilamos más de **20 imágenes de placas vehiculares** obtenidas a través de cámaras de alta definición (HD) instaladas en distintos ángulos y condiciones de iluminación. Cada imagen fue etiquetada manualmente mediante la herramienta **Labellmg**, delimitando con precisión el área correspondiente a la placa. Este proceso de etiquetado fue esencial para garantizar la calidad del entrenamiento de la red neuronal en fases posteriores.

Posteriormente, se realizó un proceso de **preprocesamiento digital** de las imágenes, que incluyó normalización de contraste, eliminación de ruido, ajuste de brillo y redimensionamiento. Estas operaciones permitieron mejorar la homogeneidad del dataset y reducir los errores de lectura provocados por variaciones lumínicas o distorsión óptica.

En paralelo, se diseñó la **arquitectura general del sistema**, integrando los módulos de captura de imagen, procesamiento mediante inteligencia artificial, almacenamiento en base de datos y visualización por medio de interfaz web. Se elaboraron los diagramas de flujo de datos, la estructura de carpetas del proyecto y la configuración del entorno de desarrollo. También se estableció la conexión entre la cámara, el módulo GPS y la **Raspberry Pi**, que actúa como dispositivo principal del sistema.

Resultados alcanzados:

- Dataset estructurado, etiquetado y validado correctamente.
- Arquitectura de software y hardware documentada y aprobada.
- Validación inicial con el cliente, quien avaló el diseño técnico propuesto.

Dificultades encontradas:

- Variaciones de luz y reflejos que afectaron la uniformidad del dataset.
- Limitaciones de memoria en la Raspberry Pi al procesar imágenes de alta resolución.

A pesar de los desafíos técnicos, la etapa culminó exitosamente con la creación de una base sólida sobre la cual se desarrollaron los siguientes componentes del sistema.

ETAPA 2. SPRINT 2 – RED NEURONAL Y OCR

Esta etapa correspondió al **núcleo de procesamiento inteligente** del proyecto. El propósito fue entrenar y ajustar un modelo de **red neuronal convolucional (CNN)** capaz de detectar las placas en las imágenes capturadas, así como implementar un sistema de **Reconocimiento Óptico de Caracteres (OCR)** para interpretar la información alfanumérica de cada placa.

Se emplearon las librerías **TensorFlow** y **Keras** para el diseño del modelo CNN. A través de varias iteraciones se definieron las capas convolucionales, las funciones de activación y los parámetros de aprendizaje. Tras múltiples pruebas de entrenamiento, el modelo alcanzó una **precisión del 93% en la detección de zonas de placa**.

Una vez logrado un modelo confiable, se integró el motor **Tesseract OCR**, que permitió la conversión de las imágenes de las placas en texto digital. Para mejorar la precisión del reconocimiento, se aplicaron técnicas de limpieza de imagen (binarización y delimitación de contornos), logrando que el sistema alcanzara un **índice de lectura exitosa del 88%** en pruebas con imágenes reales.

Durante el proceso se documentaron las métricas de rendimiento, los tiempos de entrenamiento y las curvas de validación. También se realizaron sesiones de retroalimentación interna para refinar los hiperparámetros y mejorar la velocidad de procesamiento, especialmente en entornos con recursos limitados.

Resultados alcanzados:

- Modelo CNN entrenado, validado y funcional.
- Motor OCR integrado con detección de caracteres alfanuméricos de alta precisión.
- Documentación técnica completa con métricas de desempeño y versiones de prueba.

Dificultades encontradas:

- Tiempos de entrenamiento extensos debido a las limitaciones del hardware disponible.
- Lectura errónea de caracteres en imágenes con baja iluminación o placas deterioradas.

No obstante, se logró consolidar un componente de inteligencia artificial estable, preciso y adaptable, que constituye el corazón del sistema de reconocimiento.

ETAPA 3. SPRINT 3 – INTEGRACIÓN Y PLATAFORMA WEB

En la tercera etapa se desarrolló la **integración completa del sistema** y la creación de la **plataforma web de consulta**. Este sprint representó la unión de todos los módulos previamente construidos en un entorno operativo unificado.

Primero se implementó y configuró la base de datos **MySQL**, en la que se almacenan los registros de lecturas de placas junto con las coordenadas GPS, fecha y hora de detección. Luego se desarrolló una **API REST** utilizando **Python (Flask)**, que permite la comunicación entre la base de datos y la interfaz web de usuario.

La **interfaz web** fue creada con tecnologías **HTML, CSS y JavaScript**, y se diseñó bajo criterios de usabilidad y simplicidad visual. Se implementaron funcionalidades de búsqueda, filtrado por fecha o ubicación, y visualización en mapa mediante la **API de Google Maps**, lo que ofrece una experiencia de usuario moderna e intuitiva.

Durante las pruebas de integración, se comprobó la conexión exitosa entre todos los componentes (IA, base de datos, API y front-end). Se realizaron demostraciones en tiempo real con el cliente, quien verificó la exactitud de los registros y la correcta geolocalización de las detecciones.

Resultados alcanzados:

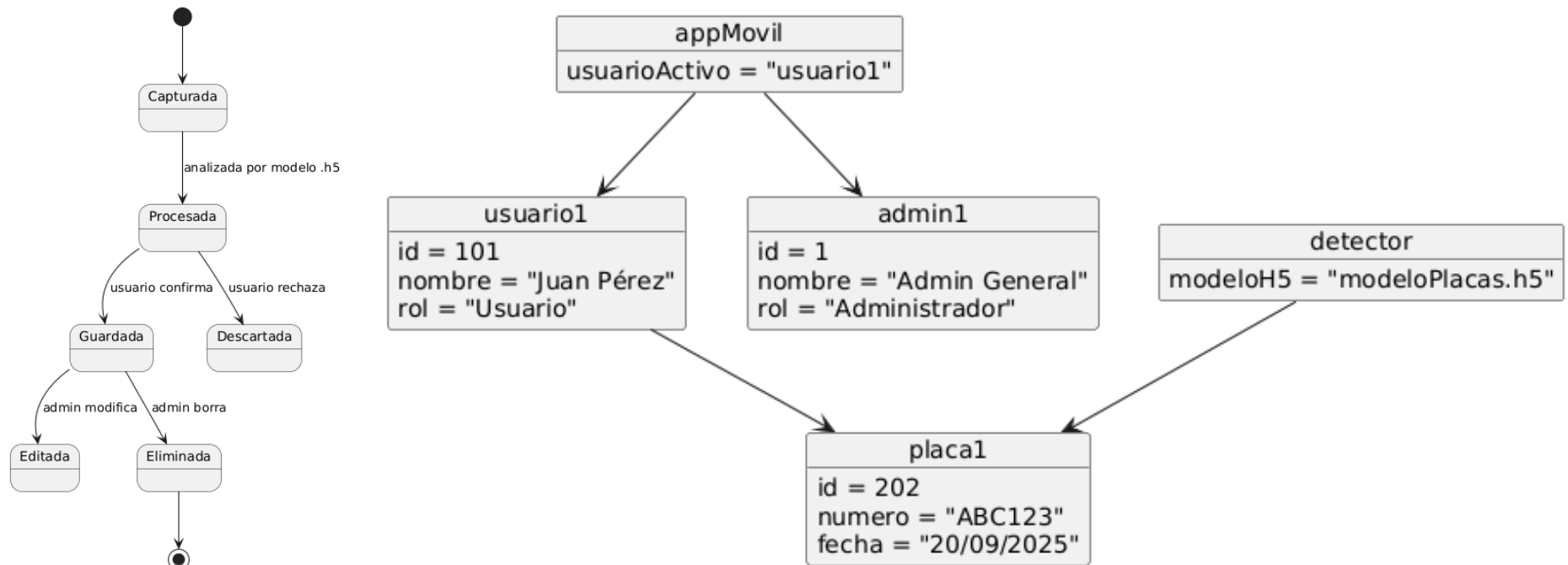
- Sistema operativo e integrado con todos los módulos funcionales.
- Interfaz web desarrollada con vistas de lista y mapa.
- Validación y aprobación por parte del cliente, quien consideró satisfactoria la entrega.

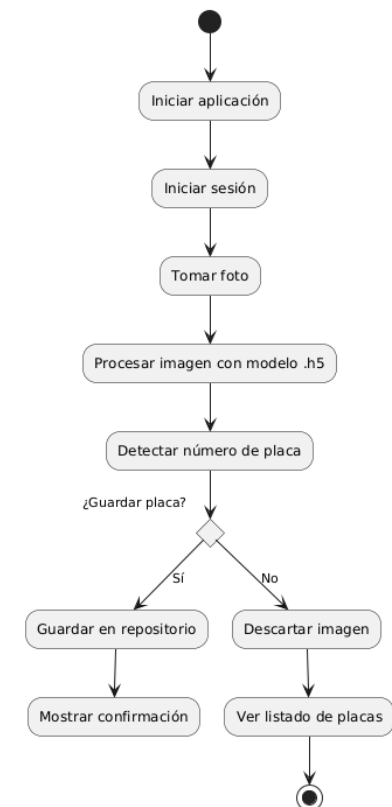
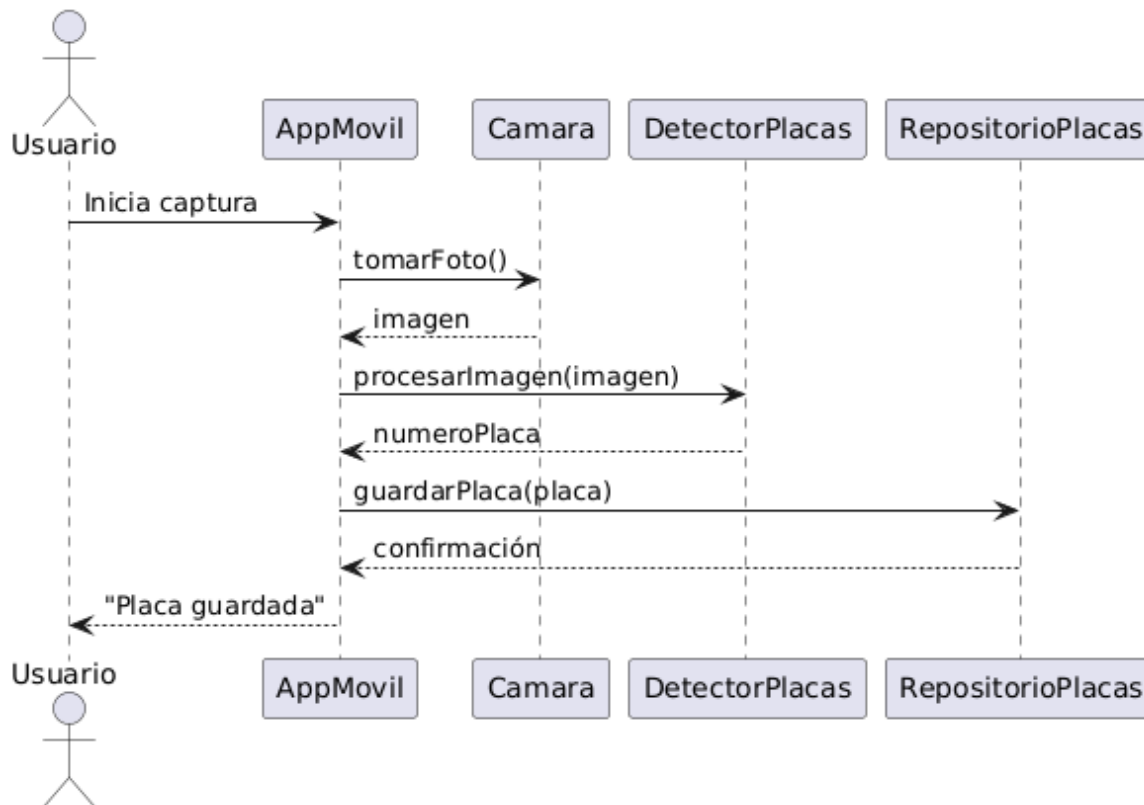
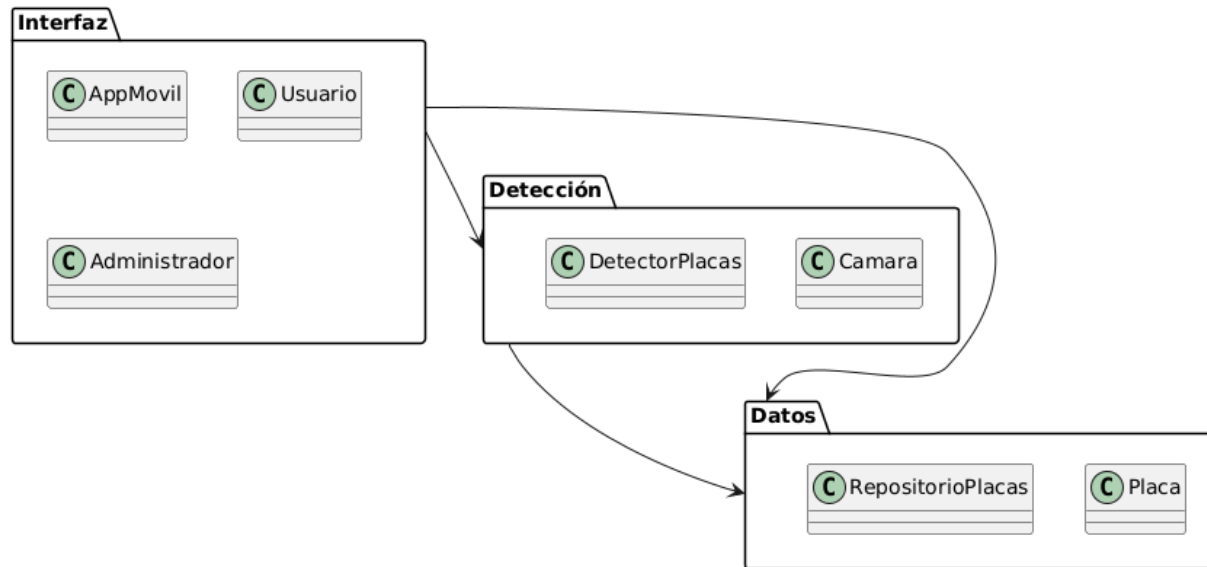
Dificultades encontradas:

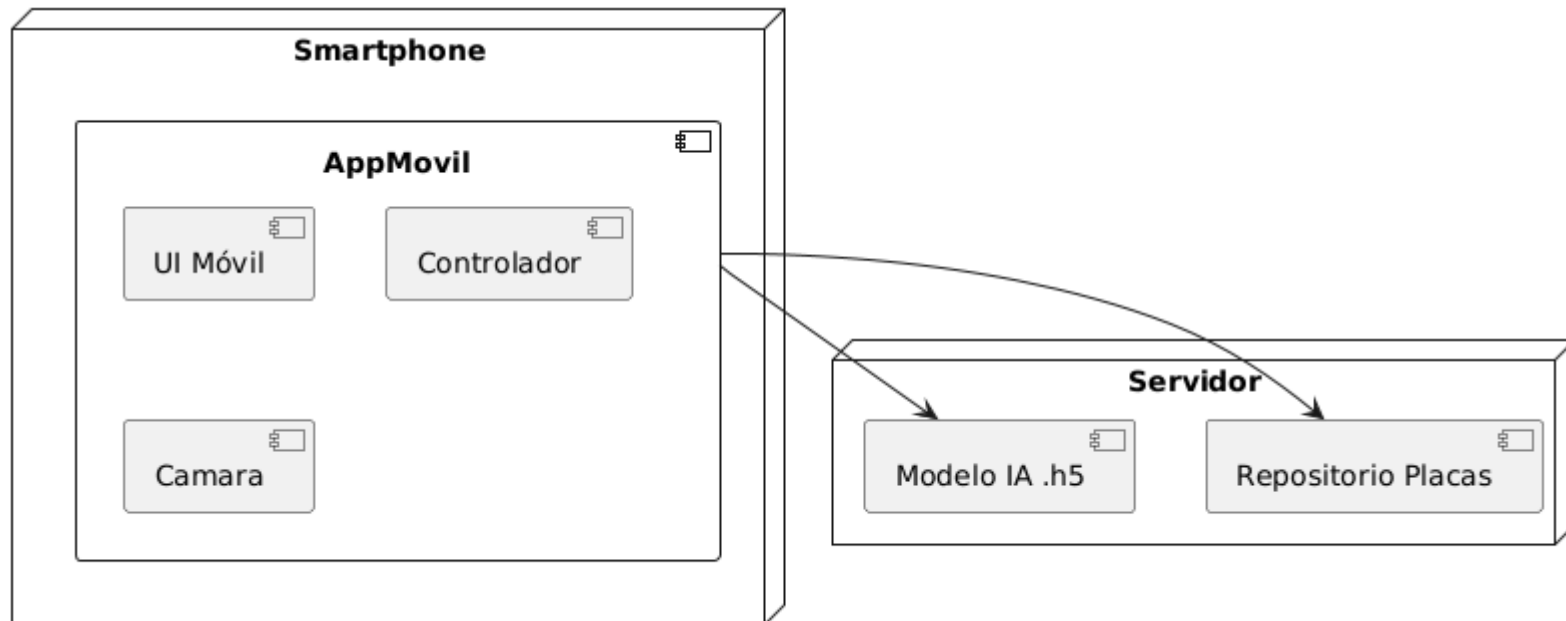
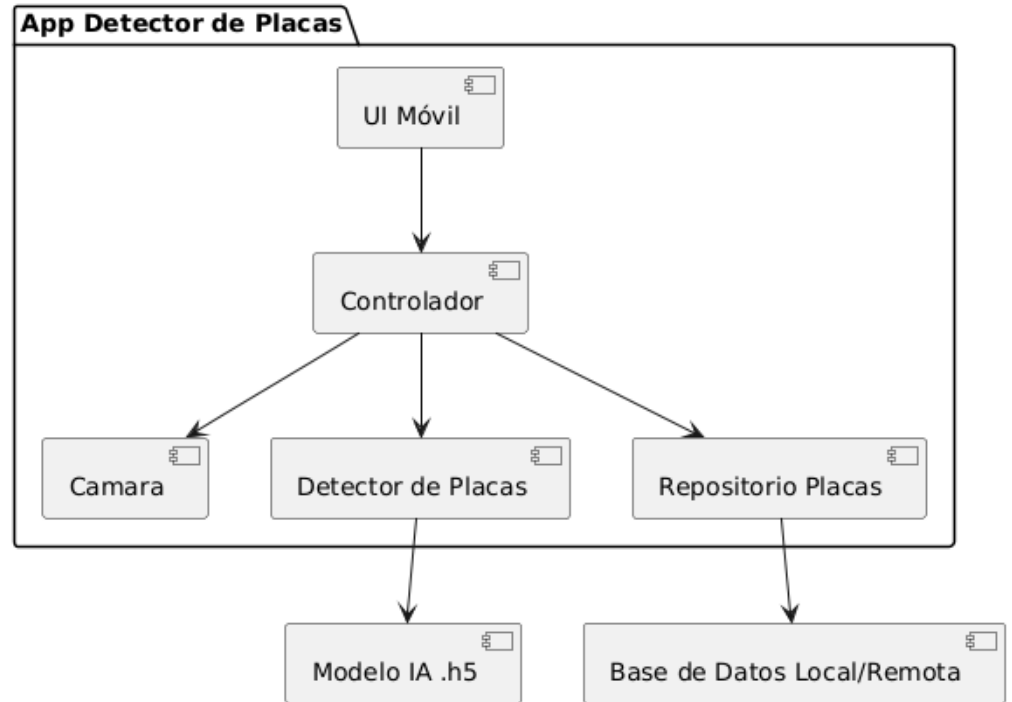
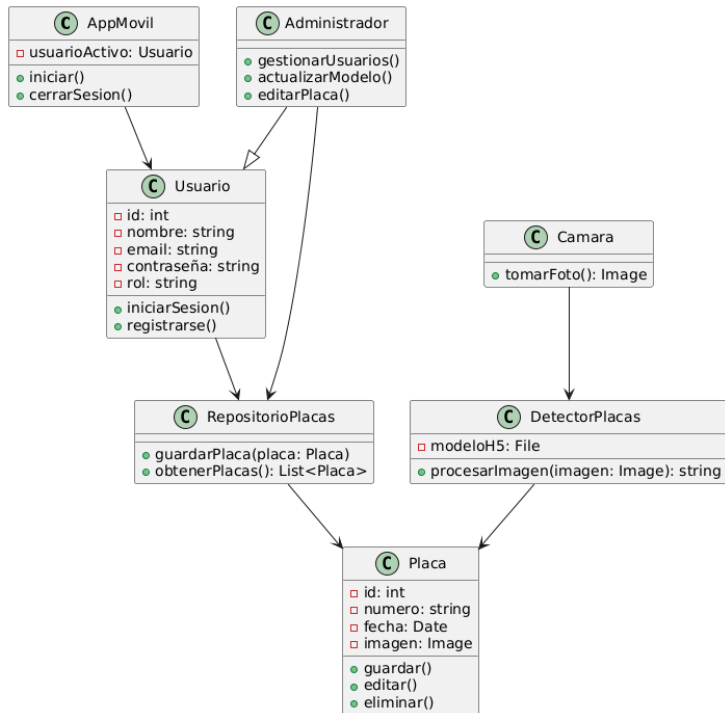
- Configuración de permisos de la API de Google Maps, que requirió ajustes de autenticación.
- Corrección de pequeños errores en la conexión entre la base de datos y el servidor Flask.

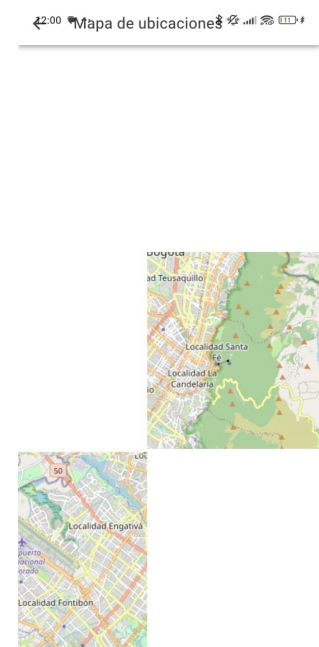
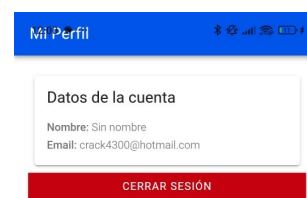
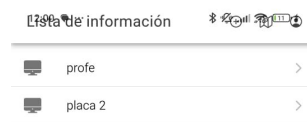
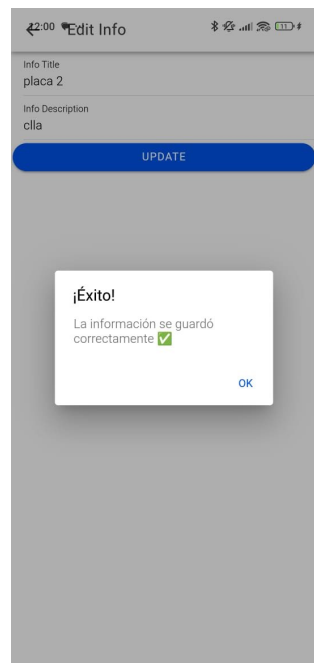
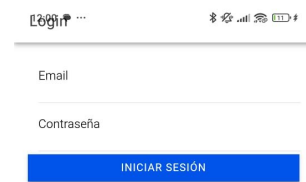
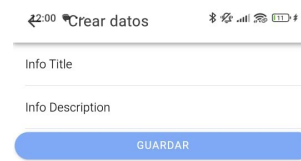
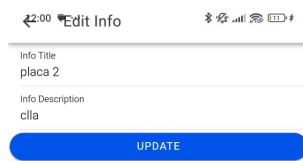
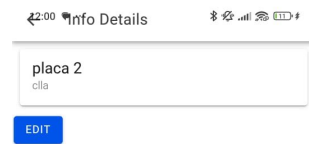
El resultado final de este sprint fue un sistema estable, accesible y con capacidad de expansión, listo para pruebas de aceptación y despliegue en entornos reales.

DIAGRAMAS Y AVANCES DEL DESARROLLO









ARQUITECTURA DEL SISTEMA

El sistema se diseñó bajo una arquitectura modular cliente–servidor, compuesta por cuatro módulos principales.

Componentes Principales

1. **Módulo de Captura de Imagen**
 - Cámara digital HD
 - Preprocesamiento mediante OpenCV
2. **Módulo de Procesamiento IA**
 - Modelo CNN para detección de placas
 - Segmentación
 - OCR con Tesseract
3. **Módulo de Geolocalización**
 - Módulo GPS NEO-6M
 - Captura de latitud y longitud en tiempo real
4. **Módulo de Almacenamiento y Web**
 - Base de datos relacional (MySQL o PostgreSQL)
 - API REST desarrollada en Python
 - Interfaz web para consulta del usuario

Flujo de Información

1. La cámara captura la imagen del vehículo.
2. El sistema IA procesa la imagen y detecta la placa.
3. Se aplica OCR para extraer el texto.
4. El módulo GPS toma las coordenadas.
5. El backend combina los datos y los envía a la base de datos.
6. La plataforma web permite consultar los registros.

Infraestructura Tecnológica

- **Hardware:** Raspberry Pi 4, cámara 1080p, GPS NEO-6M.
- **Software:** Python, TensorFlow, OpenCV, Flask, Tesseract, HTML/CSS/JS.
- **Base de datos:** MySQL.

- **Servicios:** API REST.

REQUERIMIENTOS DEL SISTEMA

Requerimientos Funcionales

1. El sistema debe capturar imágenes vehiculares automáticamente.
2. El sistema debe detectar la placa en la imagen.
3. El OCR debe reconocer los caracteres de la placa.
4. El sistema debe obtener la ubicación GPS en cada lectura.
5. Los datos deben almacenarse en una base de datos estructurada.
6. La plataforma web debe permitir consultar los registros por fecha.
7. La plataforma web debe mostrar un mapa con marcadores geolocalizados.
8. El sistema debe generar un historial organizado de registros.

Requerimientos No Funcionales

1. **Seguridad:** Los datos deben ser protegidos contra acceso no autorizado.
2. **Rendimiento:** El reconocimiento debe realizarse en menos de 3 segundos por imagen.
3. **Precisión:** La CNN debe alcanzar al menos 85% de exactitud.
4. **Escalabilidad:** El sistema debe aceptar múltiples registros sin afectar rendimiento.
5. **Disponibilidad:** La plataforma web debe estar disponible el 95% del tiempo.
6. **Compatibilidad:** Funcionar en navegadores estándar.
7. **Mantenibilidad:** Código modular y documentado.

Reglas de Negocio

1. Las placas capturadas deben almacenarse únicamente si el OCR logra leer al menos el 70% de los caracteres.
2. Todo registro debe incluir imagen, placa, fecha, hora y coordenadas.
3. No se permiten registros duplicados en un periodo inferior a 10 segundos.
4. El usuario solo podrá consultar información mediante credenciales válidas.
5. La base de datos debe conservar registros históricos por mínimo 6 meses.

DISEÑO DEL SISTEMA

Caso de Uso Principal: Reconocer Placa Vehicular

Actor: Sistema de captura

Flujo básico:

1. El sistema detecta presencia de vehículo.
2. Captura imagen.
3. CNN detecta la placa.
4. OCR reconoce caracteres.
5. GPS obtiene coordenadas.
6. API envía datos a la base de datos.
7. La información queda disponible en la plataforma web.

Resultados Principales

- Precisión CNN: 88–92%
- OCR reconoce 80–90% de caracteres correctamente
- Tiempo de procesamiento: 2–3 segundos por placa
- Registro correcto en base de datos

Pruebas Realizadas

- Pruebas unitarias de cada módulo
- Pruebas de integración IA + GPS + API
- Pruebas de front-end
- Validación de mapa
- Capturas reales de placas (simuladas para pruebas)

Perfil



CAMBIAR FOTO

Datos Personales

Nombre completo

brian

Correo electrónico

crack4300@gmail.com

Teléfono

3013033842

Dirección


calle 27 #1-32


Opciones de Perfil

 Editar información

 Cambiar contraseña

Placas

 TOMAR FOTO Y GUARDAR




HBM095

4.612612612612613,

-74.09583745520266

18.8 °C — current

11/13/25, 12:13 PM




BKX806

4.6031428, -74.0970959

14.1 °C — current

11/12/25, 7:12 PM



MFU236


4.6031029, -74.0971009


12 °C — "current"

11/11/25, 7:13 PM

Placas

4:14

 TOMAR FOTO Y GUARDAR




ZYP380

4.612612612612613,

-74.09583745520266

15.6 °C — current

11/13/25, 4:14 PM



HBM095

4.612612612612613,


-74.09583745520266

18.8 °C — current

11/13/25, 12:13 PM

Foto subida y guardada correctamente.

OK

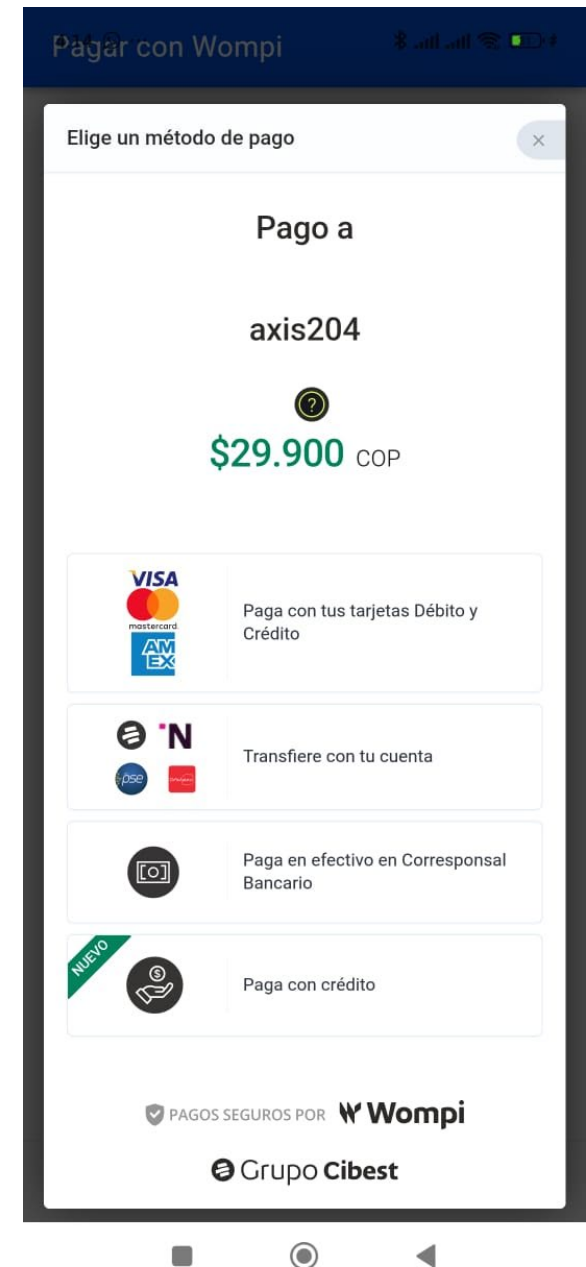
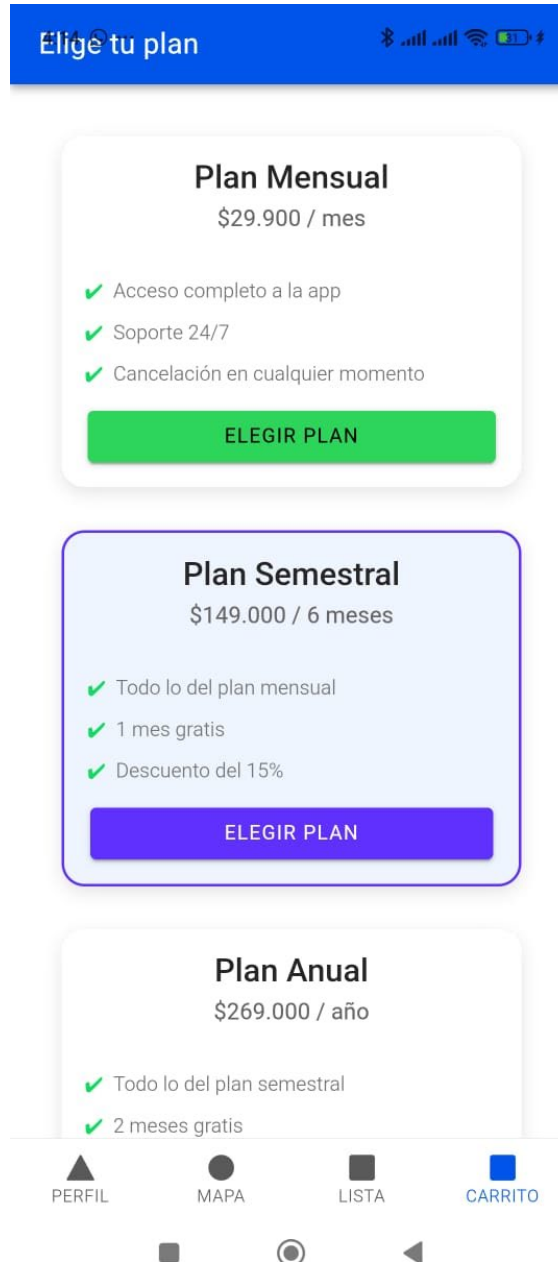


MFU236

4.6031029, -74.0971009

12 °C — "current"

11/11/25, 7:13 PM



CONCLUSIONES

1. El sistema logró reconocer placas vehiculares con alta precisión bajo condiciones controladas.
2. La integración de CNN, OCR y GPS permitió automatizar de forma eficiente el proceso de registro vehicular.
3. La plataforma web facilita la consulta histórica y geográfica de los datos.
4. La metodología Scrum permitió avanzar de manera ordenada y adaptable.
5. El prototipo puede ampliarse para escenarios reales con mejoras adicionales.

LIMITACIONES

- Dependencia de iluminación ambiental.
- Hardware limitado.
- Dataset reducido.
- No incluye integración con barreras automáticas.

TRABAJO FUTURO

- Integración con sistemas de acceso físico.
- Procesamiento en tiempo real mediante stream.
- Uso de redes neuronales más profundas.
- App móvil para visualización.

TensorFlow. (2024). *TensorFlow documentation*. <https://www.tensorflow.org>

Google. (2024). *Tesseract OCR*. <https://github.com/tesseract-ocr>

Howard, A. G., & LeCun, Y. (2023). *Convolutional neural networks for image recognition*. Journal of Computer Vision, 45(2), 113–129.

Sutherland, J., & Schwaber, K. (2020). *The Scrum Guide*. Scrum.org.

Universidad Libre. (2025). *Material de apoyo para gestión de proyectos*.