# UNIVERSIDAD ESTATAL DE BOLÍVAR

## FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

CARRERA DE SOFTWARE

**TRABAJO DE INTEGRACIÓN CURRICULAR**
**PREVIO A LA OBTENCIÓN DEL TÍTULO DE**
**INGENIERO(A)(S) EN SOFTWARE**

**FORMA:** PROYECTO TECNOLÓGICO

**TEMA:**

PROTOTIPO FUNCIONAL PARA LA DETECCIÓN DE SOMNOLENCIA EN CONDUCTORES MEDIANTE ANÁLISIS FACIAL EN TIEMPO REAL.


**AUTORES:**

POZO LÓPEZ ANTHONY PAUL

MUYULEMA YUMISEBA ALDAIR GERMAN

**DIRECTOR:**

Dr. HENRY FERNANDO VALLEJO BALLESTEROS


**GUARANDA – ECUADOR**

**2025**

**TEMA DEL PROYECTO TECNOLÓGICO**

PROTOTIPO FUNCIONAL PARA LA DETECCIÓN DE SOMNOLENCIA EN CONDUCTORES MEDIANTE ANÁLISIS FACIAL EN TIEMPO REAL.

**DEDICATORIA**

A Dios, fuente de mi fortaleza y guía en cada momento de este camino, por brindarme la perseverancia necesaria para culminar este gran propósito de vida.

A mis padres y mi hermana, quienes han sido el cimiento de mis valores y el motor de mi esfuerzo; su ejemplo de entrega y su fe incondicional en mí son el mayor regalo que he recibido.

A mi Universidad, mi segundo hogar y el espacio donde transformé mis sueños en metas académicas, por brindarme las herramientas para mi crecimiento profesional.

A mi tutor, por su invaluable mentoría y por creer en este proyecto desde el primer día, aportando no solo su conocimiento técnico, sino su visión y apoyo constante.

A mis pares académicos, por el intercambio crítico de ideas y la colaboración intelectual. Gracias por los debates constructivos y por ser parte de este proceso de aprendizaje mutuo que enriqueció mi visión profesional.

**Aldair German Muyulema Yumiseba**

## DEDICATORIA

En primer lugar, quiero dedicar este logro a Dios, mi mayor sabiduría, fuerza y propósito en la vida. Por sostenerme en los momentos más difíciles, por iluminar cada paso de este camino.

A mi abuelito Jaime López Pucachaqui, que, aunque hoy no se encuentra físicamente conmigo, vive eternamente en mi corazón. Este logro está dedicado especialmente a él, porque fue inspiración y ejemplo de esfuerzo y valores. Sé que desde el cielo comparte conmigo la alegría de este momento tan importante en mi vida.

A mis padres Jose Luis Pozo y Janeth López, por sus valiosos consejos, su sacrificio constante y su apoyo incondicional. Gracias por enseñarme el valor del trabajo honesto, el esfuerzo y la perseverancia.

A mi tío Luis Fernando López, por estar presente en todo momento conmigo, por ser ese apoyo firme cuando más lo necesité y por demostrarme, con hechos, que siempre podía contar contigo.

A mis hermanos, para que este logro les sirva de ejemplo, que todo lo que se propongan lo pueden conseguir, siempre y cuando pongan todo el esfuerzo, la constancia y la fe del mundo. Que nunca duden de sus capacidades y que cada meta que se marquen la persigan hasta convertirla en realidad.

A mi tutor de tesis, por su guía académica, su orientación y su acompañamiento durante todo el proceso de realización del trabajo. Su tutoría fue esencial para culminar con éxito este proyecto.

A mis pares académicos, que acompañaron con el desarrollo de este proyecto tecnológico, aportando su conocimiento técnico y criterio profesional en cada etapa del proceso. Este trabajo lleva también la huella de su experiencia y dedicación.

**Anthony Paul Pozo López**

**AGRADECIMIENTO**

Al concluir esta etapa tan significativa en nuestra formación profesional, queremos expresar nuestro profundo agradecimiento a todas las personas que contribuyeron a hacer posible este trabajo de titulación. En primer lugar, a Dios, por ser nuestra guía y fuente de fortaleza a lo largo de toda la carrera universitaria hasta alcanzar esta meta. De igual manera, agradecemos a nuestras familias, pilares fundamentales de apoyo incondicional cuya comprensión y aliento fueron indispensables para culminar este proyecto.

Un reconocimiento especial merece el Dr. Henry Vallejo, director de nuestro trabajo de titulación, por su orientación acertada, su dedicación y el valioso conocimiento que nos brindó durante el desarrollo e implementación del sistema para la empresa Taxi Lagos Courier S.A. Asimismo, extendemos nuestra gratitud a los pares académicos, Ing. Mónica Bonilla e Ing. Danilo Barreno, cuyas observaciones y enseñanzas enriquecieron tanto el desarrollo metodológico de esta investigación como nuestra formación a lo largo de la carrera.

Finalmente, a nuestros docentes y compañeros, gracias por acompañarnos en este camino, por las lecciones compartidas y por el apoyo mutuo que nos permitió crecer como profesionales. A todos quienes pusieron su confianza en nosotros:
MUCHAS GRACIAS.

**Aldair German Muyulema Yumiseba.**

**Anthony Paul Pozo López.**

**CERTIFICADO DE VALIDACIÓN**

UNIVERSIDAD ESTATAL DE BOLIVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

## CERTIFICADO DE VALIDACIÓN

Dr. Henry Vallejo, Ing. Mónica Bonilla, Ing. Danilo Barreno, en su orden director y Pares Académicos del Trabajo de Integración Curricular "PROTOTIPO FUNCIONAL PARA LA DETECCIÓN DE SOMNOLENCIA EN CONDUCTORES MEDIANTE ANÁLISI FACIAL EN TIEMPO REAL" desarrollado por el señor Anthony Paul Pozo López y el señor Aldair German Muyulema Yumiseba.

## CERTIFICAN

Que, luego de revisado el Trabajo de Integración Curricular en su totalidad, cumple con las exigencias académicas de la carrera SOFTWARE, por lo tanto, autorizamos su presentación y defensa.

Guaranda, 02 de marzo del 2026

Dr. Henry Vallejo
**Director**

Ing. Mónica Bonilla
**Par Académico**

Ing. Danilo Barreno
**Par Académico**

**DERECHOS DE AUTOR**

UEB UNIVERSIDAD ESTATAL DE BOLIVAR

BIBLIOTECA GENERAL

**DERECHOS DE AUTOR**

Nosotros **MUYULEMA YUMISEBA ALDAIR GERMAN** y **POZO LÓPEZ ANTHONY PAUL** portadores de la Cédula de Identidad No **0202492310** y **1718718396** en calidad de autores y titulares de los derechos morales y patrimoniales del Trabajo de Titulación: **PROTOTIPO FUNCIONAL PARA LA DETECCIÓN DE SOMNOLENCIA EN CONDUCTORES MEDIANTE ANÁLISIS FACIAL EN TIEMPO REAL**, modalidad **PROYECTO TECNOLÓGICO**, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN, concedemos a favor de la Universidad Estatal de Bolívar, una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservamos a mi/nuestro favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo/autorizamos a la Universidad Estatal de Bolívar, para que realice la digitalización y publicación de este trabajo de titulación en el Repositorio Digital, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Loa autor declaran que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

.....
Aldair German Muyulema Yumiseba
C.I: 0202492310

.....
Anthony Paul Pozo López
C.I. 1718718396

**INDICE DE CONTENIDOS**

TEMA DEL PROYECTO TECNOLÓGICO....II
DEDICATORIA.... III
AGRADECIMIENTO....V
CERTIFICADO DE VALIDACIÓN .... VI
DERECHOS DE AUTOR....VII
INTRODUCCIÓN.... 1
RESUMEN.... 2
ABSTRACT .... 3
CAPÍTULO I.... 4
FORMULACIÓN GENERAL DEL PROYECTO .... 4
1.1 Tema .... 4
Descripción del Problema.... 4
1.2 Justificación .... 4
1.3 Objetivos: General y Específicos.... 5
GENERAL: .... 5
ESPECIFICO:.... 6
CAPÍTULO II.... 7
2.1 Antecedentes.... 7
2.2 Científico .... 9
Metodología de Desarrollo de Software.... 9
Scrum.... 10
Kanban.... 10
Extreme Programming (XP).... 10
Lean Software Development .... 10
Modelo V.... 11
Prototipado .... 11

2.3 Conceptual ..... 15
Somnolencia ..... 15
Lenguaje de Programación ..... 16
Python ..... 16
C++ ..... 16
Java ..... 16
Inteligencia Artificial ..... 16
OpenCV ..... 17
Dlib ..... 17
MediaPipe ..... 17
TensorFlow ..... 17
Hardware ..... 17
Raspberry Pi ..... 17
Arduino ..... 18
2.4 Legal ..... 18
Ley de Propiedad Intelectual ..... 18
Ley Orgánica de Protección de Datos Personales ..... 19
Código INGENIOS ..... 20
2.5 Georreferencial ..... 22
CAPÍTULO III ..... 24
METODOLOGÍA ..... 24
3.1 Metodología de Desarrollo de Software ..... 24
Fases y Procesos en Scrum Aplicados al Proyecto ..... 24
3.2 Técnicas e Instrumentos de Recopilación de Datos ..... 32
CAPÍTULO IV ..... 34
INGENIERÍA DEL PROYECTO ..... 34
4.1 Requerimientos funcionales ..... 34

4.2 Diseño ..... 41
4.2.1 Arquitectura del software ..... 41
HARDWARE ..... 42
Raspberry Pi 4 ..... 42
CAMARA NOCTURNA DIURNA SENSOR IR AUTOMÁTICO 5MP PARA RASPBERRY PI ..... 43
MEMORIA SD ..... 43
SOFTWARE ..... 44
Librería Dlib ..... 44
OpenCV ..... 44
React ..... 45
PostgreSQL ..... 46
INTERFACES ..... 47
4.2.2 Diagrama entidad relación ..... 54
4.2.3 Modelo relacional ..... 55
4.2.4 Patrón de diseño arquitectónico en Capas Distribuida con enfoque Edge – Cloud 56
4.2.5 Vista arquitectónicas 4+1 ..... 56
Vista de Escenarios ..... 56
Vista de Diseño Lógica ..... 58
Vista de Despliegue ..... 59
Vista de Procesos ..... 60
Vista Física ..... 64
5.1 Codificación ..... 67
Backend Laravel ..... 67
CONTROLADORES — /app/Http/Controllers/Api ..... 67
MODELOS ELOQUENT — /app/Models ..... 70
MIGRACIONES DE BASE DE DATOS — /database/migrations ..... 73

SEEDER DE DATOS INICIALES — /database/seeders ..... 74
REGLAS DE VALIDACIÓN — /app/Rules ..... 74
CONFIGURACIÓN — /config ..... 75
RUTAS DE LA API — /routes/api.php ..... 75
Frontend React ..... 75
COMPONENTES COMPARTIDOS — /src/components ..... 75
MÓDULOS DE FUNCIONALIDADES — /src/features ..... 76
UTILIDADES Y CONFIGURACIÓN — /src/lib ..... 77
PÁGINAS DEL FRONTEND REACT — /src/pages ..... 77
ALGORITMOS DE DETECCIÓN ..... 79
PARÁMETROS DE CONFIGURACIÓN ..... 81
ARQUITECTURA DEL SISTEMA EMBEBIDO ..... 83
GESTIÓN DE ALARMA MULTIPLATAFORMA ..... 84
INTEGRACIÓN DE LAS TRES CAPAS ..... 85
6.1 Pruebas ..... 86
Propósito ..... 86
6.1.1 Diseño y Desarrollo de Casos de Prueba ..... 86
6.1.2 Registro de Riesgos de Prueba ..... 87
6.1.3 Herramientas de Prueba ..... 87
6.1.4 Pruebas de Unidad ..... 88
6.1.5 Prueba de Funcionalidad ..... 89
6.1.6 Prueba de Rendimiento ..... 89
6.1.7 Prueba de Seguridad ..... 90
6.1.8 Pruebas de Usabilidad ..... 90
6.1.9 Resultados por Escenario de Prueba ..... 91
6.1.10 Hallazgos: Problema de Conectividad y Falsa Detección Diferida ..... 93
6.1.11 Análisis Estadístico — Preguntas de Evaluación ..... 94

Pregunta 1: ¿En cuánto tiempo el sistema detectó el estado de somnolencia?.... 94

Pregunta 2: ¿La alarma fue suficientemente audible para alertarle durante la conducción?..... 95

Pregunta 3: ¿En qué franja horaria se concentraron los síntomas de somnolencia? ..... 96

Pregunta 4: ¿El sistema generó falsas alarmas (detección sin somnolencia real)? ..... 97

Pregunta 5: ¿El sistema funcionó correctamente en tramos sin señal de datos? . 97

Pregunta 6: ¿Le resultó incómodo trabajar con el dispositivo instalado en el vehículo?..... 98

6.1.12 Prueba de Cierre — Ficha de Observación ..... 98

CONCLUSIONES..... 100

RECOMENDACIONES ..... 101

BIBLIOGRAFÍA..... 102

Anexo 1. ..... 106

Anexo 2. ..... 108

Anexo 3. ..... 110

Anexo 4. ..... 112

Anexo 5. ..... 116

Anexo 6..... 121

Anexo 7. ..... 127

Anexo 8. ..... 155

Anexo 9..... 169

Anexo 10..... 172

**INDICE DE TABLAS**

Tabla 1 Características principales de las Metodologías de Desarrollo de Software. .... 12
Tabla 2 *Requerimiento funcional 1.* ..... 34
Tabla 3 *Requerimiento funcional 2.* ..... 35
Tabla 4 *Requerimiento funcional 3.* ..... 35
Tabla 5 *Requerimiento funcional 4.* ..... 36
Tabla 6 *Requerimiento funcional 5.* ..... 37
Tabla 7 *Requerimiento funcional 6.* ..... 38
Tabla 8 *Requerimiento funcional 7.* ..... 39
Tabla 9 *Requerimiento funcional 8.* ..... 39
Tabla 10 *Requerimiento funcional 9.* ..... 40
Tabla 11 *Requerimiento funcional 10.* ..... 41
Tabla 12 *Tabla de presupuesto del sistema.* ..... 46
Tabla 13 *Propiedades del modelo Eloquent AsignacionCVR.* ..... 70
Tabla 14 *Propiedades del modelo Eloquent Conductor.* ..... 70
Tabla 15 *Propiedades del modelo Eloquent EventoSomnolencia.* ..... 71
Tabla 16 *Propiedades del modelo Eloquent Ruta.* ..... 72
Tabla 17 *Propiedades del modelo Eloquent Vehículo.* ..... 72
Tabla 18 *Propiedades del modelo Eloquent User.* ..... 73
Tabla 19 *Migraciones de base de datos PostgreSQL* ..... 73
Tabla 20 *Componentes compartidos del frontend React.* ..... 76
Tabla 21 *Módulos de funcionalidades del frontend React.* ..... 76
Tabla 22 *Parámetros de configuración del sistema embebido.* ..... 81
Tabla 23 *Flujo de integración de las tres capas del sistema.* ..... 85
Tabla 24 *Diseño y desarrollo de casos de prueba (Stakeholders).* ..... 86
Tabla 25 *Prueba de riesgo.* ..... 87
Tabla 26 *Herramientas de Prueba* ..... 87
Tabla 27 *Pruebas unitarias del sistema embebido - detección de somnolencia.* ..... 88
Tabla 28 *Pruebas unitarias de la plataforma web - autenticación y gestión.* ..... 88
Tabla 29 *Prueba de funcionalidad del sistema embebido y la plataforma web.* ..... 89
Tabla 30 *Prueba de disponibilidad del sistema.* ..... 89
Tabla 31 *Prueba de rendimiento del desempeño del sistema.* ..... 89
Tabla 32 *Prueba de seguridad validación de protección del sistema.* ..... 90

Tabla 33 *Pruebas de usabilidad de la plataforma web.* ..... 90
Tabla 34 *Pruebas en escenario diurno (12:00 - 15:00) - Post-Almuerzo.* ..... 91
Tabla 35 Pruebas de escenario nocturno (19:30 - 01:30) - Jornada Laboral Extendida. 92
Tabla 36 *Pruebas de conectividad y falsa detección.* ..... 93
Tabla 37 *Tiempo de Detección de Somnolencia.* ..... 94
Tabla 38 *Medidas de Tendencia Central — Tiempo de Detección.* ..... 95
Tabla 39 *Percepción de Audibilidad de la Alarma GPIO por Nivel de Somnolencia.* .. 95
Tabla 40 *Resultados de signos de somnolencia en horarios.* ..... 96
Tabla 41 *Resultados de falsas alarmas.* ..... 97
Tabla 42 *Resultados de conectividad.* ..... 97
Tabla 43 *Evaluación de Comodidad durante la Conducción con el Dispositivo Instalado.* ..... 98
Tabla 44 *Prueba de Cierre.* ..... 99
Tabla 45 *Presupuesto sobre los gastos.* ..... 109
Tabla 46 *Requerimientos mínimos del sistema.* ..... 134
Tabla 47 *Tarjetas de resumen del Panel de Control.* ..... 137
Tabla 48 *Gráficos de estadísticas de somnolencia.* ..... 138
Tabla 49 *Opciones de cambio de estado del conductor.* ..... 147
Tabla 50 *Opciones de gestión tras operación finalizada.* ..... 149
Tabla 51 *Tipos de reportes disponibles.* ..... 150
Tabla 52 *Contenido del reporte PDF generado.* ..... 151
Tabla 53 *Secciones del menú lateral del sistema.* ..... 152
Tabla 54 *Glosario de términos técnicos.* ..... 154
Tabla 55 *Requisitos Previos de Hardware.* ..... 157
Tabla 56 *Tabla de comando para validaciones.* ..... 161
Tabla 57 *Requerimientos Previos.* ..... 163
Tabla 58 *Configuración de parámetros Web Service en Render.* ..... 166
Tabla 59 *Configuración de parámetros Static Site en Render.* ..... 167
Tabla 60 *Configuración de parámetros PostgreSQL en Render.* ..... 167
Tabla 61 *Interna vs Externa Database URL.* ..... 167

**INDICE DE FIGURAS**

Figura 1 *Mapa Geográfico de la ubicación de la empresa Taxi Lagos Courier S.A. de la ciudad de Ibarra.* ..... 23
Figura 2 *Diseño de arquitectura del sistema de detección de somnolencia.* ..... 41
Figura 3 *Partes del Raspberry Pi 4 Model B.* ..... 42
Figura 4 *Cámara 5MP para Raspberry.* ..... 43
Figura 5 *Tarjeta Micro SD SanDisk 64G.* ..... 43
Figura 7 *Logo de la Librería Dlib.* ..... 44
Figura 8 *Logo de OpenCV.* ..... 44
Figura 9 *Logo de Laravel v12.51* ..... 45
Figura 10 *Logo de React v19.2* ..... 45
Figura 11 *Logo de la base de datos PostgreSQL.* ..... 46
Figura 12 *Loguin para ingreso al interfaz de administrador/gerente o propietario.* .... 47
Figura 13 *Interfaz de Panel de Control para el monitoreo estadístico de somnolencia en conductores, en tiempo real.* ..... 48
Figura 14 *Interfaz de Gestión de Conductores - Listado de Conductores Registrados en el Sistema.* ..... 48
Figura 15 *Modal de Registro de Nuevo Conductor - Portal de Gestión de Personal.* ... 49
Figura 16 *Interfaz de Gestión de Vehículos - Listado de Unidades de la Flota Vehicular.* ..... 49
Figura 17 *Modal para el Registro de Nuevo Vehículo - Ingreso de Datos Técnicos de la Unidad.* ..... 50
Figura 18 *Interfaz de Gestión de Rutas - Administración de Trayectos y Distancias de Operación.* ..... 50
Figura 19 *Modal para la Creación de Nueva Ruta - Definición de Trayecto y Distancia.* ..... 51
Figura 20 *Interfaz de Asignación de Conductores a Rutas - Sistema de Asignación de Operaciones.* ..... 51
Figura 21 *Modal de Estado del Conductor - Gestión de Disponibilidad y Estado Operativo en Ruta.* ..... 52
Figura 22 *Modal para la Gestión de Conductor Pendiente - Registro de Ausencia y Opciones de Reasignación.* ..... 52

Figura 23 *Modal para la Operación Completada - Gestión de Reasignación y Cambio de Estado del Conductor.* ..... 53
Figura 24 *Interfaz para la Generación de Reportes - Sistema de Filtros y Descarga de Reportes de Somnolencia.* ..... 53
Figura 25 *Reporte Específico de Conductor - Detalle Cronológico de Eventos de Somnolencia Registrados.* ..... 54
Figura 26 *Diagrama Entidad Relación de la Base de Datos para Detección de Somnolencia en conductores.* ..... 54
Figura 27 *Diagrama de patrón arquitectónico en Capas Distribuida con enfoque Edge–Cloud.* ..... 56
Figura 28 *Diagrama de caso de uso (escenarios).* ..... 57
Figura 29 *Diagrama de clases (lógica).* ..... 59
Figura 30 *Diagrama de componentes (despliegue).* ..... 60
Figura 31 *Diagrama de actividades, sistema embebido (Detección y Registro).* ..... 61
Figura 32 *Diagrama de actividad, Sitio Web (Gestión y Reportes).* ..... 64
Figura 33 *Diagrama de despliegue (física).* ..... 66
Figura 34 *Diagrama de Gantt.* ..... 107
Figura 56 *Oficinas de Taxi Lagos Courier S.A. en Quito.* ..... 122
Figura 57 *Oficinas de Taxi Lagos Courier S.A. en Ibarra.* ..... 122
Figura 58 *Capacitación del uso de la plataforma web de monitoreo de somnolencia en tiempo real en la oficina de Quito.* ..... 123
Figura 59 *Capacitación del uso de la plataforma web de monitoreo de somnolencia en tiempo real en la oficina de Ibarra.* ..... 123
Figura 60 *Instalación del dispositivo embebido Raspberry Pi en la cabina del vehículo 1 de carga de Taxi Lagos Courier S.A.* ..... 124
Figura 61 *Dispositivo embebido Raspberry Pi correctamente instalado en el tablero del vehículo 1 de carga de Taxi Lagos Courier S.A.* ..... 124
Figura 62 ..... 125
Figura 63 *Dispositivo embebido Raspberry Pi correctamente instalado en el tablero del vehículo 2 de carga de Taxi Lagos Courier S.A.* ..... 125
Figura 64 ..... 126
Figura 65 *Pruebas de detección de eventos de somnolencia: cierre de ojos, bostezo y cabeceo en el conductor 2 de Taxi Lagos Courier S.A.* ..... 126
Figura 35 *Instalación del Raspberry Pi 4 en la cabina del vehículo.* ..... 130

Figura 36 *Encendido del Raspberry Pi 4 en el vehículo.* ..... 130
Figura 37 *Pantalla de inicio de sesión de Taxi Lagos Courier S.A.* ..... 135
Figura 38 *Panel de Control con estadísticas de somnolencia en tiempo real.*..... 137
Figura 39 *Pantalla de Gestión de Conductores.* ..... 139
Figura 40 *Formulario de registro de nuevo conductor.* ..... 139
Figura 41 *Pantalla de Gestión de Vehículos* ..... 142
Figura 42 *Formulario de registro de nuevo vehículo.*..... 142
Figura 43 *Pantalla de Gestión de Rutas.*..... 144
Figura 44 *Formulario de creación de nueva ruta.* ..... 144
Figura 45 *Pantalla de Asignación de Conductores a Rutas.*..... 146
Figura 46 *Panel de Estado del Conductor (activo En Ruta).* ..... 147
Figura 47 *Panel de Conductor Pendiente (ausencia temporal).* ..... 147
Figura 48 *Panel de Operación Finalizada con opciones de reasignación.*..... 148
Figura 49 *Pantalla de Generación de Reportes.* ..... 150
Figura 50 *Vista del reporte PDF generado con eventos de somnolencia.* ..... 151
Figura 51 *Interfaz de la Raspberry Pi Imagen 1.* ..... 158
Figura 52 *Interfaz de la Raspberry Pi Imagen 2.* ..... 158
Figura 53 *Interfaz de la Raspberry Pi Imagen 3.* ..... 159
Figura 54 *Interfaz de la Raspberry Pi Imagen 4.* ..... 159
Figura 55 Configuración de Registros DNS...... 168

## INTRODUCCIÓN

El presente trabajo de integración curricular consiste en el desarrollo de un "Prototipo funcional para la detección de somnolencia en conductores mediante análisis facial en tiempo real". El sistema se fundamenta en un modelo de arquitectura Edge-Cloud, empleando una unidad de procesamiento Raspberry Pi y algoritmos de visión artificial para monitorear patrones fisiológicos del rostro durante la conducción.

Centrándose en la situación crítica de la seguridad vial en el Ecuador, donde la fatiga al volante es uno de los principales factores de los accidentes de tránsito. Apuntando a la ausencia de dispositivos tecnológicos de prevención vial baratos y aplicables en la vida diaria. Por esta razón, este proyecto tecnológico intenta elaborar una opción, de bajo costo y alta precisión, realizando las pruebas en la empresa Taxi Lagos Courier S.A., con el fin de salvaguardar la vida de los conductores.

Para este proyecto tecnológico se utilizó la metodología ágil, metodología Scrum, que permite organizar el desarrollo del sistema por ciclos, llamados sprints. Con este marco de trabajo, se pudo hacer una integración en forma escalonada de los módulos considerados complejos y que constituían el sistema, como: el módulo de reconocimiento facial, el módulo de medición de la frecuencia de parpadeo (EAR) y el módulo de control de alarmas. También se aplicó investigación aplicada y experimental, utilizando fichas de observación y pruebas de campo, en rutas reales (Quito-Ibarra-Cayambe) para comprobar la efectividad del prototipo en diferentes condiciones de iluminación y conectividad.

El propósito fundamental de este trabajo de integración curricular es reducir la cantidad de accidentes en el Ecuador, a través de la innovación tecnológica. Buscando desarrollar un prototipo funcional que analice rostros en tiempo real para detectar señales de somnolencia. Completando los objetivos específicos que son: estudiar métodos de análisis facial, desarrollar un dispositivo que identifique rasgos oculares, establecer un sistema de alerta inmediata y evaluar la precisión y el tiempo de respuesta del sistema en vehículos reales.

**RESUMEN**

El presente trabajo de integración curricular se centró en construir un prototipo portátil capaz de captar de forma automática estados de somnolencia en conductores utilizando algorítmica de visión artificial para la detección de situaciones de riesgo en vías. Se diseñó y construyó un sistema micro computarizado basado en una Raspberry Pi 4 modelo B. Para su programación se usó el lenguaje Python y la biblioteca OpenCV para procesar imágenes en tiempo real y reconocimiento de rostros. Durante la fase experimental, se utilizaron diferentes intervalos de tiempo para observar el estado de la vista; esto facilitó la creación de patrones de fatiga en base al conteo de parpadeos. En una de las pruebas el dispositivo registró el estado de agotamiento del conductor (que sobresale el 80 % de parpadeos) y activó la alarma.

Esto permitió que el conductor recuperara su estado de atención. El uso de inteligencia artificial (IA) en un hardware accesible es una de las soluciones más adecuadas y exactas para la seguridad en las vías, y para la prevención de accidentes por somnolencia.

**Palabras clave:** somnolencia, visión artificial, Raspberry Pi, seguridad vial.

**ABSTRACT**

The present curricular integration work focused on building a portable prototype capable of automatically capturing states of drowsiness in drivers using artificial vision algorithms for the detection of risk situations on roads. A micro-computerized system was designed and built based on a Raspberry Pi 4 model B. For its programming, the Python language and the OpenCV library were used to process images in real time and face recognition. During the experimental phase, different time intervals were used to observe the state of sight; This made it easier to create fatigue patterns based on blink counts. In one of the tests, the device recorded the driver's exhaustion status (which exceeds 80% of blinks) and activated the alarm.

This allowed the driver to regain his state of attention. The use of artificial intelligence (AI) in accessible hardware is one of the most appropriate and accurate solutions for road safety, and for the prevention of accidents due to drowsiness.

**Keywords:** drowsiness, artificial vision, Raspberry Pi model B, road safety.

# CAPÍTULO I

# FORMULACIÓN GENERAL DEL PROYECTO

## 1.1 Tema

### Descripción del Problema

En Ecuador, los accidentes de tránsito constituyen un problema crítico de seguridad vial y salud pública. La Agencia Nacional de Tránsito (ANT) reportó que entre enero y noviembre de 2016 se registraron 711 siniestros vinculados a somnolencia o fatiga del conductor, un promedio de 64 casos mensuales (El Comercio, 2017). Estas cifras demuestran el riesgo significativo que conlleva la falta de sueño, especialmente en las principales carreteras interprovinciales como la Carretera Panamericana Norte.

Un estudio de la Revista ISUPOL sobre la vía Alóag–Santo Domingo (2019–2023) comprueba la gravedad del problema: durante 2022 se registraron 132 accidentes con 95 heridos, y en el 2023, se registraron 21 muertes por cansancio del conductor (Molina et al., 2024).

Actualmente, las estadísticas del INEC y la ANT en el primer trimestre de 2025, la falta de pericia e imprudencia del conductor (categoría que incluye la conducción bajo estado de somnolencia) alcanzando un 38% de las causas de accidentes de tránsito, con más de 4.700 accidentes registrados (Instituto Nacional de Estadística y Censos INEC y Agencia Nacional de Transito ANT, 2025). A pesar de que las leyes actuales establecen tiempos de descanso en la conducción continua, el cansancio sigue siendo un factor importante en los accidentes de tránsito debido al incumplimiento de las normas y las largas jornadas laborales.

En consecuencia, la dependencia exclusiva del criterio del conductor, la falta de intervención de otros estilos de control, sumadas a las condiciones laborales de los conductores de transporte público y de carga, hace que el país tenga un elevado nivel de riesgo en las vías.

## 1.2 Justificación

Este proyecto tecnológico tiene como objetivo crear un modelo de trabajo que pueda detectar conductores cansados a través de controles faciales en vivo, para ayudar a evitar accidentes debido a la somnolencia de los conductores.

Para crear el sistema se utilizará una Raspberry Pi 4, una cámara integrada, una tarjeta microSD, los conectores necesarios y las bibliotecas de procesamiento de imágenes: OpenCV y Dlib. Sin necesidad de equipos costosos ni conexión a Internet, estas tecnologías permitirán reconocer rasgos faciales concretos, como el cierre prolongado de los ojos o la inclinación de la cabeza, y generarán notificaciones instantáneas ante posibles indicadores de somnolencia.

La justificación del proyecto radica en la falta de dispositivos tecnológicos preventivos de fácil obtención que monitoreen en tiempo real el estado del conductor. Esto es particularmente crítico en el caso de Ecuador, ya que los accidentes que ocurren por el cansancio del conductor son uno de los problemas más serios de la movilidad del país. Así, el prototipo busca ser una alternativa útil, económica y sencilla que brinde mayor seguridad en la carretera, tanto para los conductores particulares, como en el transporte público y de carga.

La empresa **Taxi Lagos Courier S.A.,** especializada en la entrega de paquetería, realizará la primera validación de la herramienta en entornos reales al ejecutarla en vehículos de la empresa, para de esta manera evaluar si realmente el sistema logra detectar el estado de somnolencia de los conductores en el transcurso de su turno.

La propuesta es que, al tener bajo costo, la solución tecnológica propuesta pueda ser implementada en una gran cantidad de vehículos y, de esta manera, alcanzar la cantidad de escenarios posibles de transporte. Además, el prototipo permitirá registrar eventos y resultados de detección, lo que sentará las bases para futuras mejoras que podrían incluir métodos de aprendizaje automático más sofisticados.

Cabe destacar que este proyecto se encuadra dentro de la línea de Ingeniería de Software, Redes y Telecomunicaciones, específicamente en la sub- línea de Diseño e Implementación de Sistemas de Información.

## 1.3 Objetivos: General y Específicos

**GENERAL:**

Desarrollar un prototipo funcional para la detección de somnolencia en conductores mediante análisis facial en tiempo real.

**ESPECIFICO:**

- Revisar bibliografía sobre técnicas de análisis facial en tiempo real aplicadas a la detección de somnolencia.
- Analizar las principales señales de somnolencia en conductores para aplicar técnicas de detección
- Diseñar un sistema de detección facial que permita identificar rasgos faciales, con énfasis en el parpadeo o el cierre ocular.
- Implementar un módulo de alertas que emita señales sonoras o visuales cuando se detecten indicadores de somnolencia en el conductor.
- Evaluar el funcionamiento del prototipo mediante pruebas experimentales en los vehículos de la empresa Taxi Lagos Courier S. A., verificando la precisión y respuesta del sistema ante condiciones reales de conducción.

# CAPÍTULO II

# MARCO TEÓRICO

## 2.1 Antecedentes

En los últimos años se han llevado a cabo numerosas investigaciones en Ecuador sobre el uso del análisis facial y las técnicas de visión artificial para identificar el cansancio de los conductores. Los cinco estudios pertinentes que se enumeran a continuación proporcionan conocimientos básicos directos para la investigación actual.

Según López y Vargas (2025) el "Sistema avanzado de asistencia al conductor para la detección de distracción y somnolencia utilizando puntos de referencia faciales" tuvo como objetivo desarrollar un sistema de asistencia que identifique episodios de somnolencia y distracción mediante el uso de puntos de referencia faciales (landmarks) y métricas geométricas como EAR y PERCLOS. La metodología de desarrollo de software fue de carácter experimental, sustentada en la implementación de técnicas de visión por computador con OpenCV y Dlib, realizando pruebas en escenarios controlados. Entre las conclusiones, se determinó que la técnica es viable y funcional en condiciones de laboratorio, aunque se evidenciaron limitaciones frente a variaciones de iluminación, uso de gafas y entornos de conducción reales.

Campos y Hurtado (2024) en el "Sistema de detección y alerta del estado de somnolencia de conductores mediante visión artificial" propuso construir un prototipo que identifique signos de fatiga y active alertas en el conductor. Mejorando la seguridad vial utilizando la captura de video e inteligencia artificial para el reconocimiento de patrones de somnolencia y emitir alertas. Para esto se realizó la integración de reconocimiento facial y cálculo de umbrales con posterior evaluación de la funcionalidad del sistema en un laboratorio. Se constató que el sistema es capaz de reconocer y alertar somnolencia con efectividad en condiciones controladas. Sin embargo, se encontró que el sistema presentaba debilidades en el reconocimiento de patrones de conducta en situaciones de iluminación de baja visibilidad y con conductores que presentaban rasgos especiales.

Mosquera y Ventura (2024) en el trabajo de titulación "Diseño e implementación de un prototipo de detección de somnolencia en conductores" se basó en el reconocimiento de rasgos faciales que permitan la identificación de signos de agotamiento, desencadenando así de forma inmediata una alerta. La metodología practicada es de tipo experimental que

considera la implementación de modelos de visión artificial y el uso de redes neuronales convolucionales para el reconocimiento de patrones. Realizando pruebas de validación a nivel de laboratorio, obteniéndose métricas de precisión y desempeño. Como conclusión, el prototipo alcanzó resultados adecuados en escenarios controlados, pero se evidenció la necesidad de optimizar el funcionamiento en condiciones de baja iluminación y con obstrucciones como el uso de gafas.

Armijos Benalcázar, (2018) En el siguiente trabajo de titulación "Sistema de detección de somnolencia utilizando técnicas de visión artificial", desarrollado en la Universidad Técnica del Norte, se planteó como objetivo diseñar e implementar un sistema que identifique signos de somnolencia en conductores a través de algoritmos de visión artificial y aprendizaje automático. La metodología de desarrollo incluyó un enfoque experimental organizado en fases de: diseño, implementación y pruebas de laboratorio. El sistema implemento la librería OpenCV para el análisis de imágenes faciales y la evaluación comparativa. El estudio concluyó que la propuesta podía ser probada con éxito en condiciones controladas, pero sugirió realizar pruebas en escenarios reales y que los datos se expandieran localmente para una mejor generalización del sistema.

Delgado y Yandún (2022) El proyecto denominado "Sistema de detección de somnolencia para conductores de taxis en la ciudad de Tulcán" tenía como objetivo desarrollar una herramienta que detectara la somnolencia en conductores y emitiera alertas a tiempo. La metodología que desarrollo fue aplicada, involucrando técnicas de visión por computadora con EAR y PERCLOS como indicadores, además de pruebas piloto con taxistas. Entre las conclusiones, se señaló que el sistema era funcional y la percepción de seguridad entre los usuarios mejoró. Sin embargo, las pruebas fueron limitadas y tuvieron dificultades relacionadas con la vibración del vehículo y la variabilidad de iluminación en escenarios reales.

Las cinco publicaciones revisadas aportan contribuciones significativas a la detección de la somnolencia basada en la visión artificial; sin embargo, esta evaluación destaca algunos vacíos pertinentes:

- Escenarios de validación reducidos, La mayoría de los prototipos no se sometieron a pruebas exhaustivas en carretera, sino que se examinaron en entornos de laboratorio o con muestras limitadas.

- Sensibilidad a condiciones adversas, Problemas recurrentes con los ángulos de cámara, el uso de gafas y la iluminación fluctuante.
- Metodología de desarrollo de software poco documentada, En numerosas investigaciones se echa en falta un enfoque estructurado para la gestión del ciclo de vida del software.

El proyecto “Prototipo funcional para la detección de somnolencia en conductores mediante análisis facial en tiempo real” propone un enfoque distinto en comparación con los antecedentes revisados, destacando los siguientes aportes:

- Validación con un conjunto más amplio de conductores y horarios variables en condiciones reales de conducción.
- Aplicación de métodos de preprocesamiento para hacer frente a obstáculos y condiciones de iluminación deficientes.
- Utilizar el marco de trabajo (Scrum) que incluye control de versiones, pruebas automatizadas y documentación de sprints.

Con el fin de superar las limitaciones de estudios anteriores, como la sensibilidad a condiciones de iluminación desfavorables y la falta de validación en escenarios reales, tras revisar la bibliografía sobre el tema se ha determinado que es aconsejable implementar un prototipo funcional para detectar la somnolencia del conductor basado en el análisis facial en tiempo real. Con el fin de ofrecer un sistema más fiable, eficiente y pertinente en situaciones de conducción reales, este proyecto presenta una estrategia completa basada en métodos ágiles para el desarrollo de software y la visión artificial.

## 2.2 Científico

### Metodología de Desarrollo de Software

Maida y Pacienzia (2015) indican que: "Una metodología de desarrollo de software es un marco para organizar y gestionar el proceso de desarrollo de sistemas de información. Con el tiempo, se han desarrollado un gran número de marcos diferentes, cada uno con sus ventajas y desventajas particulares. No es necesario que un enfoque de desarrollo de sistemas sea adecuado para todos los proyectos. En función de factores técnicos, organizativos, del proyecto y del equipo, cada uno de los enfoques ofrecidos es el más adecuado para un determinado tipo de proyecto." (p. 7).

### Scrum

Según Armijos et al., (2024) Scrum es un marco de trabajo ágil que permite abordar problemas complejos y adaptativos, mientras se entregan productos del máximo valor posible de manera productiva y creativa. Se fundamenta en la teoría de control de procesos empíricos que afirma que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. La implementación de Scrum se sustenta en tres pilares fundamentales: transparencia, inspección y adaptación (p. 75).

### Kanban

Castillo y Guaña, (2024), señalan que Kanban es una metodología ágil para la gestión eficiente del flujo de trabajo en el desarrollo de software. Fundamentada en principios Lean, Kanban se materializa mediante tableros visuales que muestran el progreso del trabajo en etapas como "Backlog", "En progreso" y "Listo". Esta metodología permite visualizar el flujo de trabajo completo, identificar cuellos de botella y optimizar continuamente el proceso de desarrollo. El principio central de Kanban es limitar el trabajo en progreso (WIP) para mantener un flujo constante y predecible (p. 1).

### Extreme Programming (XP)

Para Merchán et al., (2024) Extreme Programming (XP) es una metodología ágil que pone énfasis en la relevancia de perfeccionar las interacciones entre personas como un elemento fundamental para lograr el éxito a la hora de desarrollar software. XP promueve la colaboración, enseñanza y un entorno laboral positivo. Fundamenta su enfoque en la retroalimentación entre desarrolladores y cliente, la comunicación flexible entre los involucrados, la implementación de soluciones de manera sencilla y la disposición para realizar cambios. Es particularmente apropiado para proyectos con requerimientos indefinidos, altamente variables y con un riesgo técnico considerable (p. 5059).

### Lean Software Development

Gaete et al., (2021) sostienen que el desarrollo Lean es una adaptación de los principios de manufactura Lean de Toyota al contexto del desarrollo de software. Esta metodología se fundamenta en la eliminación sistemática de desperdicios y en la maximización del valor entregado al cliente. Lean Software Development enfatiza la importancia de optimizar el flujo de valor completo, desde el concepto inicial hasta la entrega del

software al usuario final, minimizando todo aquello que no agregue valor al producto (p. 143).

**Modelo V**

Pressman, (2010)describe el Modelo V como una variante del modelo en cascada que enfatiza la verificación y validación del software. En este modelo, cada fase de desarrollo tiene una fase de prueba correspondiente, formando una "V" cuando se representa gráficamente. El lado izquierdo de la V representa las fases de especificación y diseño, mientras que el lado derecho representa las fases de integración y pruebas. Este modelo hace explícita la relación entre cada fase de desarrollo y su fase de prueba correspondiente (p. 42).

**Prototipado**

Así mismo Pressman, (2010) define que el prototipado es un proceso que permite a desarrolladores y clientes crear un modelo del software que se va a construir. El prototipo puede ser algo que funcione de manera limitada o puede ser un programa que ejecute parte de la función requerida. El objetivo es obtener retroalimentación temprana del usuario para refinar los requisitos y validar el diseño antes de comprometerse con un desarrollo completo (p. 45).

**Tabla 1**

*Características principales de las Metodologías de Desarrollo de Software.*

| Característica | Scrum | Kanban | Extreme Programming (XP) | Lean | Modelo V | Prototipado |
|---|---|---|---|---|---|---|
| **Enfoque Principal** | Marco de trabajo para la gestión de proyectos, basado en iteraciones (sprints) y roles definidos. Enfoque empírico fundamentado en transparencia, inspección y adaptación. | Método visual para gestionar el flujo de trabajo, centrado en la mejora continua y la eficiencia. Limitado para optimizar el flujo. | Metodología de ingeniería de software que enfatiza prácticas técnicas y la calidad del código a través de valores como comunicación, simplicidad y retroalimentación. | Principio de Gestión Lean que se centra en eliminar "desperdicios" para maximizar la eficiencia y la entrega al cliente. | Modelo secuencial que enfatiza la verificación y validación (V&V) en cada fase, con una correlación directa entre desarrollo y pruebas. | Enfoque iterativo que produce versiones preliminares del software para validar requisitos y obtener retroalimentación temprana. |
| **Ciclos** | Utiliza "Sprints" de duración fija (típicamente de 1 a 4 semanas). Al final de cada sprint, se entrega un incremento de un producto potencialmente entregable. | No hay ciclos de duración fija. El trabajo fluye continuamente sin iteraciones predefinidas. | Ciclos cortos de desarrollo (típicamente de 1 a 3 semanas) con entregas frecuentes y retroalimentación constante. | No hay ciclos fijos, se enfoca en un flujo de valor continuo con entregas cuando el trabajo está listo. | Fases secuenciales bien definidas: requisitos, diseño, implementación, pruebas. Cada fase debe completarse. | Iteraciones cortas para desarrollar y refinar prototipos sucesivos hasta que se logre el producto final o se validen los requisitos. |
| **Roles** | Product Owner: Prioriza el trabajo y maximiza el valor del | No definen roles formales específicos, pero se fomenta la | A diferencia de Scrum, no hay roles fijos, pero se promueve la | No hay roles formalmente prescritos, pero se | Roles tradicionales bien definidos: analistas de requisitos, diseñadores | Desarrolladores y usuarios finales trabajan |

| | | | | | | |
|---|---|---|---|---|---|---|
| | producto. Scrum Master: Facilita el proceso y elimina impedimentos. Equipo de Desarrollo: Autogestionado, multifuncional y responsable de entregar incrementos. | colaboración, el liderazgo distribuido y la responsabilidad compartida. | colaboración cercana entre desarrolladores y el cliente, con énfasis en la programación en pareja. | fomenta la mejora continua y el empoderamiento de todo el equipo de trabajo. | de sistema, arquitectos, desarrolladores, ingenieros de pruebas. | conjuntamente en la evaluación y refinamiento de prototipos. |
| **Gestión del Trabajo** | Se utiliza un "Product Backlog" priorizado por valor de negocio y un "Sprint Backlog" para las tareas del sprint actual. | Se utiliza un tablero visual (Kanban) con columnas para "Por Hacer", "En Progreso" y "Hecho". Se establecen límites de WIP explícitos. | Las historias de usuario se priorizan en función del valor para el cliente. El trabajo se divide en iteraciones cortas con entregas funcionales frecuentes. | Se basa en el mapeo de la cadena de valor para identificar y eliminar sistemáticamente el desperdicio. | Documentación extensa en cada fase que sirve como entrada directa para las actividades de verificación y validación correspondientes. | Desarrollo iterativo de prototipos con retroalimentación continua del usuario. Cada prototipo se refina basándose en la evaluación. |
| **Ventajas** | • Gran estructura y organización del trabajo.<br>• Fomenta la colaboración del equipo y la autogestión.<br>• Permite inspección y adaptación constante del proceso y producto. | • Gran flexibilidad y adaptabilidad a cambios. Visualización clara del flujo de trabajo y cuellos de botella.<br>• Reduce el tiempo de entrega y aumenta la eficiencia.<br>• No requiere cambios drásticos en la organización. | • Alta calidad del código gracias a prácticas técnicas como TDD y programación en parejas. ·Fomenta la comunicación directa y la retroalimentación constante.<br>• Permite la adaptación continua a cambios de requisitos. | • Maximiza la eficiencia y la entrega de valor al cliente.<br>• Reduce costos operativos y tiempos de producción. Elimina actividades que no agregan valor. Se puede aplicar a cualquier proceso, no solo | • Clara trazabilidad entre cada fase de desarrollo y sus pruebas correspondientes. Detección temprana de errores en cada nivel.<br>• Útil cuando los requisitos son estables y bien definidos desde el inicio. | • Validación temprana de requisitos y reducción de malentendidos.<br>• Reducción significativa del riesgo de construir el producto equivocado.<br>• Útil cuando los requisitos son |

| | | | | | | |
|---|---|---|---|---|---|---|
| | • Transparencia en el progreso del proyecto.<br>• Adecuado para proyectos con requisitos cambiantes. | • Ideal para equipos de mantenimiento o soporte continuo. | • Reduce el riesgo técnico mediante integración continua. | software. Enfoque holístico del sistema completo. | Cumplimiento riguroso de estándares de calidad. | inciertos o mal definidos.<br>• Mejora la comunicación con usuarios no técnicos. |
| **¿Cuándo usarla?** | • Proyectos complejos e inciertos con requisitos emergentes. Cuando se necesita un marco de trabajo estructurado para equipos autogestionados. Proyectos donde la adaptación y la retroalimentación frecuente son cruciales.<br>• Equipos que buscan mejorar su colaboración. | • Proyectos con flujo continuo de trabajo y demandas variables. Cuando se necesita visualizar y optimizar el proceso existente. Equipos que buscan mejora continua sin adoptar cambios disruptivos. Mantenimiento de sistemas y soporte operativo. | • Proyectos de desarrollo de software donde la calidad técnica y la ingeniería son prioritarias.<br>• Equipos pequeños y altamente colaborativos (2-12 personas).<br>• Cuando el cliente está dispuesto y disponible para participar activamente en el proceso.<br>• Requisitos volátiles con alta probabilidad de cambio. | • Proyectos o procesos donde se busca maximizar la eficiencia y eliminar pasos innecesarios. Organizaciones que buscan optimización del flujo de valor completo.<br>• Proyectos con flujo de trabajo bien definido, pero con oportunidades de mejora significativas. | • Proyectos con requisitos bien definidos, estables y documentados desde el inicio.<br>• Sistemas críticos que requieren verificación rigurosa (médicos, aeroespaciales, financieros).<br>• Cuando el cumplimiento de estándares regulatorios es fundamental.<br>• Proyectos donde los cambios son costosos. | • Proyectos con requisitos poco claros, ambiguos o cambiantes al inicio.<br>• Cuando se necesita validación temprana de conceptos con usuarios finales.<br>• Sistemas con interfaces de usuario complejas o innovadoras.<br>• Proyectos donde el costo de construir el producto equivocado es alto. |

Para el desarrollo del presente proyecto titulado “Prototipo funcional para la detección de somnolencia en conductores mediante análisis facial en tiempo real”, se utilizará la metodología ágil Scrum, se justifica por la naturaleza experimental y evolutiva del proyecto, así como por los requisitos técnicos que implican el uso de visión artificial y procesamiento en tiempo real.

Scrum permitirá dividir el desarrollo en sprints breves, haciendo hincapié en el avance cuantificable. En cada iteración se incluirán actividades específicas como el reconocimiento facial, la grabación de vídeo en tiempo real, el cálculo de métricas como EAR o PERCLOS y la integración del módulo de alertas. Cada sprint dará como resultado una mejora del prototipo funcional, que luego podrá someterse a pruebas, evaluarse y ajustarse en función de los resultados.

Sin embargo, la validación práctica en los vehículos de Taxi Lagos Courier S.A. requeriría realizar pruebas en diferentes momentos y con distintos conductores, exigiendo flexibilidad para rectificar errores o mejorar el rendimiento del sistema. Esto se facilita utilizando el marco de trabajo Scrum, que fomenta la revisión continua y la modificación sin tener que rehacer todas las etapas de desarrollo.

En general, el marco de trabajo Scrum proporciona herramientas para la gestión efectiva del desarrollo del sistema, ya que proporciona supervisión de los procesos y asegura que el prototipo se modifique para cumplir con los requisitos técnicos y el contexto real de uso. Esta técnica combina preparación, flexibilidad y mejora continua con el fin de producir un resultado práctico y validable en condiciones reales de conducción.

## 2.3 Conceptual

### Somnolencia

La somnolencia es un estado fisiológico crítico que puede comprometer la seguridad vial, caracterizado por una tendencia marcada a dormir o sentir sueño excesivo. Su detección se asocia al análisis del estado de los ojos y la manifestación de bostezos, indicadores importantes para prevenir accidentes. Un estudio con modelo de red neuronal convolucional para detectar somnolencia en conductores alcanzó una precisión alta en identificar estos signos, resaltando la importancia de monitorear la somnolencia para la seguridad vial (Marycielo y Mario, 2024).

## Lenguaje de Programación

Los lenguajes de programación son la herramienta básica de construcción de programas, proporcionando al alumno una herramienta para aprender de manera lúdica, a la vez que mejoran su pensamiento y su desarrollo cognoscitivo (Challenger et al., 2014).

## Python

Es un lenguaje de programación de alto nivel, interpretado y multiparadigma, conocido por su sintaxis clara y legible. Es ampliamente utilizado para desarrollo de software, inteligencia artificial, análisis de datos y automatización. Su facilidad de uso y amplio soporte a través de bibliotecas especializadas lo hacen idóneo para aplicaciones científicas y de ingeniería (Rivera y Isaac, 2022).

## C++

C++ es un lenguaje de programación compilado que permite combinar paradigmas estructurados, orientados a objetos y genéricos. Se utiliza para desarrollar aplicaciones de alto rendimiento, sistemas operativos, videojuegos y software embebido debido a su control sobre los recursos de hardware. Ofrece eficiencia, portabilidad y escalabilidad en proyectos complejos (Nicolalde Rodríguez y Bohórquez Tinizaray, 2024).

## Java

Java es un lenguaje orientado a objetos que permite desarrollar aplicaciones multiplataforma gracias a la Máquina Virtual de Java (JVM). Se caracteriza por su portabilidad, seguridad y capacidad para trabajar en entornos cliente-servidor. Es ampliamente usado en desarrollo empresarial, aplicaciones móviles y sistemas distribuidos (Cruz y W, 2002).

## Inteligencia Artificial

La inteligencia artificial es la rama científica que se encarga de desarrollar sistemas capaces de llevar a cabo actividades que, en caso de ser llevadas a cabo por humanos, necesitarían de inteligencia.  Estas responsabilidades comprenden el pensamiento, el aprendizaje, la percepción, el entendimiento del lenguaje natural y la habilidad para interactuar con el ambiente (Paredes, 2025).

**OpenCV**

Biblioteca open source para visión por computadora que facilita el procesamiento y análisis de imágenes y videos en tiempo real. Proporciona herramientas para detección, reconocimiento y seguimiento de objetos, ampliamente usada en la academia e industria para aplicaciones que requieren análisis visual avanzado (Rivera & Isaac, 2022).

**Dlib**

Biblioteca especializada en aprendizaje automático y procesamiento de imagen C++ con bindings en Python que incluye modelos pre-entrenados para detección de rostros y localización de puntos faciales (facial landmarks, como el predictor de 68 puntos), usada

**MediaPipe**

MediaPipe es un framework desarrollado por Google que permite implementar soluciones de percepción computacional mediante modelos optimizados para dispositivos locales. Proporciona componentes reutilizables para detección facial, seguimiento de manos y estimación de postura corporal. Es especialmente útil en aplicaciones móviles e interactivas en tiempo real (Ruiz y Tonthat, 2023).

**TensorFlow**

TensorFlow es una biblioteca de código abierto para el desarrollo y ejecución de modelos de aprendizaje automático y redes neuronales. Permite entrenar, implementar y optimizar modelos en diversos entornos desde servidores hasta dispositivos móviles. Su arquitectura flexible facilita el procesamiento distribuido y la experimentación científica (TensorFlow, 2020).

**Hardware**

El hardware comprende los componentes físicos de un sistema informático, como procesadores, cámaras, sensores, placas electrónicas y dispositivos de entrada y salida. En aplicaciones tecnológicas, estos elementos permiten la adquisición, procesamiento y transmisión de datos para ejecutar funciones específicas. Es la base material que soporta el software (Moreno, 2024).

**Raspberry Pi**

Raspberry Pi es un minicomputador de bajo costo que constituye una herramienta educativa valiosa para fomentar la enseñanza de la informática, la programación y la

tecnología, representando una solución económica y versátil para introducir conceptos de informática y electrónica en programas educativos, preparando a los estudiantes para carreras en campos relacionados con la tecnología (Pérez, 2024).

### Arduino

Arduino es una plataforma hardware de código abierto basada en microcontroladores, ampliamente utilizada en educación para la enseñanza de programación y electrónica, permitiendo a los estudiantes desarrollar soluciones hardware-software de manera práctica y accesible mediante microprogramación (Lanzas, 2019).

## 2.4 Legal

### Ley de Propiedad Intelectual

Según la ley orgánica de propiedad Intelectual en su Sección II refiere, Objetivo del derecho de autor:

En el **artículo 4**, La protección del derecho de autor recae sobre todas las obras del ingenio, en el ámbito literario o artístico, cualquiera que sea su género, forma de expresión, mérito o finalidad. Los derechos reconocidos por el presente Título son independientes de la propiedad del objeto material en el cual está incorporada la obra y su goce o ejercicio no están supeditados al requisito del registro o al cumplimiento de cualquier otra formalidad (Ley de Propiedad Intelectual, 2006).

Entre las diferentes obras protegidas por este artículo se encuentran los programas de ordenador.

Sección V.

De los programas de ordenador

En el **artículo 28**, Los programas de ordenador se consideran obras literarias y se protegen como tales. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea en forma legible por el hombre (código fuente) o en forma legible por máquina (código objeto), ya sean programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa (Ley de Propiedad Intelectual, 2006).

El **artículo 29** de la Ley de Propiedad intelectual manifiesta que:

Es titular de un programa de ordenador, el productor, esto es la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra. Se considerará titular, salvo prueba en contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual. Dicho titular está además legitimado para ejercer en nombre propio los derechos morales sobre la obra, incluyendo la facultad para decidir sobre su divulgación. El productor tendrá el derecho exclusivo de realizar, autorizar o prohibir la realización de modificaciones o versiones sucesivas del programa, y de programas derivados del mismo. Las disposiciones del presente artículo podrán ser modificadas mediante acuerdo entre los autores y el productor (Ley de Propiedad Intelectual, 2006).

**Ley Orgánica de Protección de Datos Personales**

El **Artículo 42** de la Ley Orgánica de Protección de Datos Personales establece la obligatoriedad de una evaluación de impacto bajo ciertas condiciones:

Evaluación del impacto del tratamiento de datos personales. -El responsable realizará una evaluación de impacto del tratamiento de datos personales cuando se haya identificado la probabilidad de que dicho tratamiento, por su naturaleza, contexto o fines, conlleve un alto riesgo para los derechos y libertades del titular o cuando la Autoridad de Protección de Datos Personales lo requiera.

La evaluación de impacto relativa a la protección de los datos será de carácter obligatoria en caso de:

Evaluación sistemática y exhaustiva de aspectos personales de personas físicas que se base en un tratamiento automatizado, como la elaboración de perfiles, y sobre cuya base se tomen decisiones que produzcan efectos jurídicos para las personas naturales;

Tratamiento a gran escala de las categorías especiales de datos, o de los datos personales relativos a condenas e infracciones penales, o

Observación sistemática a gran escala de una zona de acceso público.

La Autoridad de Protección de Datos Personales establecerá otros tipos de operaciones de tratamiento que requieran una evaluación de impacto relativa a la protección de datos.

La evaluación de impacto deberá efectuarse previo al inicio del tratamiento de datos personales (Ley Orgánica de Protección de Datos Personales, 2021).

El **Artículo 43** de la Ley Orgánica de Protección de Datos Personales establece los plazos para la notificación de vulneraciones de seguridad:

**Art. 43.-**Notificación de vulneración de seguridad.-El responsable del tratamiento deberá notificar la vulneración de la seguridad de datos personales a la Autoridad de Protección de Datos Personales y la Agencia de Regulación y Control de las Telecomunicaciones, tan pronto sea posible, y a más tardar en el término de cinco (5) días después de que haya tenido constancia de ella, a menos que sea improbable que dicha violación de la seguridad constituya un riesgo para los derechos y las libertades de las personas físicas. Si la notificación a la Autoridad de Protección de Datos no tiene lugar en el término de cinco (5) días, deberá ir acompañada de indicación de los motivos de la dilación.

El encargado del tratamiento deberá notificar al responsable cualquier vulneración de la seguridad de datos personales tan pronto sea posible, y a más tardar dentro del término de dos (2) días contados a partir de la fecha en la que tenga conocimiento de ella (Ley Orgánica de Protección de Datos Personales, 2021).

**Código INGENIOS**

**Artículo 1.-** Objeto. - El presente Código tiene por objeto normar el Sistema Nacional de Ciencia, Tecnología, Innovación y Saberes Ancestrales previsto en la Constitución de la República del Ecuador y su articulación principalmente con el Sistema Nacional de Educación, el Sistema de Educación Superior y el Sistema Nacional de Cultura, con la finalidad de establecer un marco legal en el que se estructure la economía social de los conocimientos, la creatividad y la innovación (Código Orgánico de la Economía Social de los Cnocimientos, Creatividad e Innovación, 2016).

**Artículo 2.-** Ámbito. - Se rigen por el presente Código todas las personas naturales, jurídicas y demás formas asociativas que desarrollen actividades relacionadas a la economía social de los conocimientos, la creatividad y la innovación.

Las actividades relacionadas a la economía social de los conocimientos, la creatividad y la innovación son aquellas enfocadas a la creación de valor a partir del uso intensivo de la generación, transmisión, gestión y aprovechamiento del bien de interés público conocimiento, que incluye los conocimientos tradicionales; promoviendo en todos los sectores sociales y productivos la colaboración y potenciación de las capacidades individuales y sociales, la democratización, distribución equitativa, y aprovechamiento

eficiente de los recursos en armonía con la naturaleza, dirigido a la obtención del buen vivir (Código Orgánico de la Economía Social de los Conocimientos, creatividad e Innovación, 2016).

**Artículo 3.-** Fines. - El presente Código tiene, como principales, los siguientes fines:

1. Generar instrumentos para promover un modelo económico que democratice la producción, transmisión y apropiación del conocimiento como bien de interés público, garantizando así la acumulación y redistribución de la riqueza de modo justo, sostenible y en armonía con la naturaleza;

2. Promover el desarrollo de la ciencia, la tecnología, la innovación y la creatividad para satisfacer necesidades y efectivizar el ejercicio de derechos de las personas, de los pueblos y de la naturaleza;

3. Incentivar la producción del conocimiento de una manera democrática colaborativa y solidaria;

4. Incentivar la circulación y transferencia nacional y regional de los conocimientos y tecnologías disponibles, a través de la conformación de redes de innovación social, de investigación, académicas y en general, para acrecentarlos desde la práctica de la complementariedad y solidaridad;

5. Generar una visión pluralista e inclusiva en el aprovechamiento de los conocimientos, dándole supremacía al valor de uso sobre el valor de cambio;

6. Desarrollar las formas de propiedad de los conocimientos compatibles con el buen vivir, siendo estas: pública, privada, comunitaria, estatal, asociativa y mixta.

7. Incentivar la desagregación y transferencia tecnológica a través de mecanismos que permitan la generación de investigación, desarrollo de tecnología e innovación con un alto grado de componente nacional;

8. Promover la distribución justa y equitativa de los beneficios derivados de las actividades vinculadas a la generación, transmisión, gestión, uso y aprovechamiento de los conocimientos, la tecnología, la innovación y los conocimientos tradicionales, así

como el uso eficiente de los factores sociales de la producción para incrementar el acervo de conocimiento e innovación;

9. Establecer las fuentes de financiamiento y los incentivos para el desarrollo de las actividades de la economía social de los conocimientos, la creatividad y la innovación;

10. Fomentar el desarrollo de la sociedad del conocimiento y de la información como principio fundamental para el aumento de productividad en los factores de producción y actividades laborales intensivas en conocimiento; y,

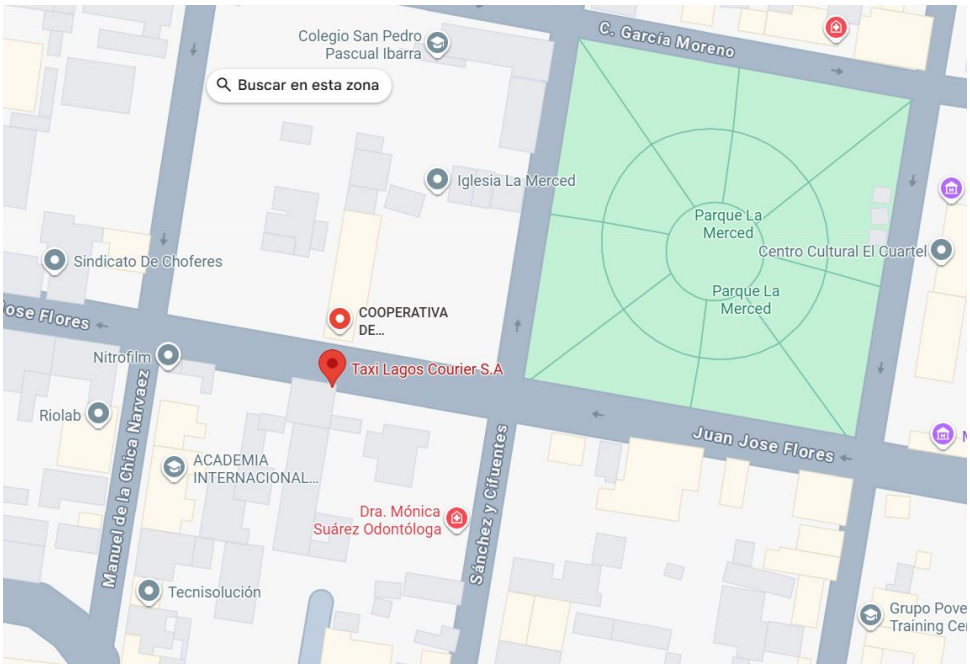
11. Fomentar la protección de la biodiversidad como patrimonio del Estado, a través de las reglas que garanticen su aprovechamiento soberano y sustentable, proteger y precautelar los derechos de las comunidades, pueblos y nacionalidades sobre sus conocimientos tradicionales y saberes ancestrales relacionados a la biodiversidad; y evitar la apropiación indebida de la biodiversidad y los conocimientos tradicionales asociados a esta (Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, 2016).

## 2.5 Georreferencial

- **Provincia:** Imbabura
- **Cantón:** San Miguel de Ibarra
- **Dirección:** Calle Juan José Flores 9-37, Ibarra, Ecuador
- **Latitud:** 0.351259
- **Longitud:** -78.993747

**Figura 1**

*Mapa Geográfico de la ubicación de la empresa Taxi Lagos Courier S.A. de la ciudad de Ibarra.*



# CAPÍTULO III

# METODOLOGÍA

## 3.1 Metodología de Desarrollo de Software

Para el desarrollo del prototipo funcional de detección de somnolencia en conductores mediante análisis facial en tiempo real, se empleará la metodología ágil Scrum. Esta elección se basa en su adaptabilidad a proyectos de innovación tecnológica, donde la iteración rápida, la validación de prototipos en entornos reales y la respuesta a cambios en los requisitos técnicos o funcionales son importantes. Scrum permitirá un ciclo continuo de planificación, construcción, pruebas y retroalimentación. Asegurando que el prototipo avance de manera eficiente y transparente.

El equipo Scrum estará conformado por:

- **Product Owner (PO):** Gerente de la empresa de paquetería Taxi Lagos Courier S.A.
- **Scrum Master (SM):** Aldair, quien facilita el proceso ágil, elimina impedimentos y asegura el cumplimiento del marco de trabajo.
- **Development Team:** Anthony y Aldair encargados de la construcción de la arquitectura, el desarrollo del sistema y la realización de pruebas.

El marco de trabajo Scrum se considera el más indicado para este proyecto tecnológico debido a la implementación de hardware (Raspberry Pi 4), el procesamiento en tiempo real (OpenCV + Dlib) y las pruebas en condiciones reales de conducción, ya que en el proyecto se requieren múltiples iteraciones, validaciones y perfeccionamiento constantes.

**Fases y Procesos en Scrum Aplicados al Proyecto**

1. **Inicio del Proyecto (Sprint 0)**

Durante la fase inicial del proyecto (Sprint 0), se realizó un análisis de los principales métodos de detección de somnolencia reportados en la literatura científica, con el objetivo de seleccionar el enfoque más adecuado para la implementación del prototipo.

- **Métodos basados en el comportamiento del conductor**

  Este enfoque se basa en el análisis de características faciales y conductuales, tales como:

- Frecuencia de parpadeo (EAR)
- Apertura bucal (MAR)
- Inclinación de la cabeza (NM)

Estos indicadores permiten detectar signos tempranos de fatiga mediante visión artificial, siendo el método seleccionado en este proyecto debido a su carácter no invasivo y su facilidad de implementación en sistemas embebidos.

- **Presupuesto vs Mercado**

  En términos de viabilidad económica, el prototipo desarrollado presenta una ventaja significativa frente a las soluciones comerciales existentes en el mercado. Los sistemas comerciales de detección de somnolencia, especialmente aquellos integrados en vehículos modernos (ADAS), tienen un costo aproximado que oscila entre **$500 y $2000 USD**, dependiendo del fabricante y las funcionalidades incorporadas. Estos sistemas suelen requerir hardware especializado, sensores adicionales y software propietario.

  En contraste, el prototipo desarrollado en este proyecto tiene un costo estimado entre **$180 y $200 USD**, considerando componentes como:

  - Raspberry Pi 4.
  - Cámara infrarroja.
  - Tarjeta microSD.
  - Módulos de alerta.
  - Módulo de alimentación.

  Esta diferencia representa una reducción de costo superior al **80%**, lo que convierte a la solución en una alternativa accesible para contextos como el transporte público y de carga en Ecuador.

  Por lo tanto, la principal contribución del proyecto radica en la apertura de la tecnología de seguridad vial, permitiendo que sistemas de detección de somnolencia sean implementados en vehículos que actualmente no cuentan con este tipo de asistencia.

- **Estrategias para la Prevención del Cansancio y la Somnolencia en Conductores.**

  Con base en la evidencia científica revisada y el contexto operativo de la empresa Taxi Lagos Courier S.A., se proponen las siguientes estrategias complementarias al sistema desarrollado:

1. **Gestión y rotación de turnos de conducción.**

   Implementar una política formal de rotación de turnos basada en los registros de eventos de somnolencia generados por el sistema. La plataforma web del prototipo permite identificar patrones de fatiga por horario y por conductor, información que puede ser utilizada por el administrador para rediseñar la distribución de rutas y turnos, evitando que un mismo conductor opere en franjas horarias críticas de manera consecutiva (madrugada y tarde-noche, según los datos recopilados durante las pruebas de campo).

2. **Pausas activas programadas durante jornadas extendidas.**

   Establecer protocolos de paradas obligatorias cada 2 a 3 horas de conducción continua, especialmente en rutas interprovinciales como la ruta Ibarra-Quito-Cayambe operada por Taxi Lagos Courier S.A.

3. **Capacitación en higiene del sueño y gestión de la fatiga.**

   Desarrollar programas de formación para los conductores orientados a la comprensión de los riesgos de la somnolencia al volante, la importancia del descanso nocturno adecuado (entre 7 y 8 horas por noche), la identificación temprana de señales personales de fatiga, y el impacto del consumo de cafeína y otros estimulantes como medidas paliativas —no sustitutivas— del sueño. Esta acción preventiva complementa de forma directa la función del sistema tecnológico, fomentando una cultura de seguridad vial al interior de la organización.

4. **Monitoreo estadístico continuo mediante la plataforma web.**

   El panel de control desarrollado en la plataforma web del sistema permite al administrador acceder en tiempo real a estadísticas de somnolencia por conductor, por vehículo, por ruta y por franja horaria. Esta información constituye una herramienta de gestión preventiva que facilita la identificación de conductores con alta frecuencia de eventos de somnolencia, la detección de rutas o tramos críticos, y la toma de decisiones informadas para la planificación operativa de la flota.

**Actividades fundamentales:**

- Levantamiento de requerimientos no funcionales.

- Construcción del Product Backlog inicial, basado en:
    - revisión bibliográfica sobre técnicas de detección de somnolencia,
    - análisis de señales faciales (cierre ocular, parpadeo prolongado, inclinación de cabeza),
    - análisis del procesamiento en tiempo real con OpenCV y Dlib.
- Identificación de actores y casos de uso iniciales, mediante la técnica de entrevista y observación.
- Definición de los criterios de aceptación por funcionalidad.
- Selección del patrón arquitectónico y diseño de la primera versión de la arquitectura 4+1.
- Configuración del entorno técnico:
    - Raspberry Pi 5
    - Cámara integrada
    - Bibliotecas OpenCV y Dlib
    - Lenguaje de programación (Python)
    - Repositorio Git

El Sprint 0 permite establecer la visión del producto y preparar el entorno para el desarrollo incremental.

**2. Planificación del Proyecto (Product Backlog y Sprint Planning)**

Durante esta fase, el Product Owner define y prioriza el Product Backlog, el cual incluye todas las funcionalidades necesarias para cumplir los objetivos del proyecto:

**Ejemplos de elementos del Product Backlog:**

- Implementación del módulo de captura facial.
- Detección de ojos, parpadeo y EAR (Eye Aspect Ratio).
- Implementación de análisis de somnolencia basado en EAR y postura.
- Módulo de alertas (sonoras y visuales).

- Registro de eventos detectados en archivos o base de datos local.
- Integración y calibración con Raspberry Pi 5.
- Pruebas experimentales en vehículos reales (Taxi Lagos Courier S.A.).

Durante el Sprint Planning, se define qué elementos del Backlog se desarrollarán en el Sprint, tomando en cuenta la capacidad del equipo y la prioridad.

Cada Sprint tendrá una duración entre 1 y 2 semanas, permitiendo entregar incrementos funcionales del prototipo físico y del software.

### 3. Diseño del Sistema (Diseño dentro del Sprint)

En Scrum, el diseño no ocurre como una fase cerrada, sino como una actividad iterativa dentro de cada Sprint.

**Para este proyecto se consideran:**

#### a) Patrón Arquitectónico General

Se adoptará el patrón arquitectónico Modelo-Vista-Controlador (MVC) mezclado con una arquitectura modular orientada a componentes, debido a las necesidades del sistema:

- Módulo de captura de video (interfaz con la cámara).
- Módulo de análisis facial (detección de ojos, rasgos y cálculo EAR).
- Módulo de lógica de somnolencia.
- Módulo de alertas.
- Módulo de registro de eventos.

Este modularidad facilita las pruebas, la refactorización y futuras ampliaciones.

#### b) Vistas Arquitectónicas 4+1

Se utilizará el modelo de vistas 4+1 para documentar el sistema.

1. **Vista Lógica**

- Diagrama de clases:
    - Detector de rostro,
    - Módulo de Dlib,
    - Calculadora EAR,
    - Sistema de alertas.
- Patrones GoF aplicados:
    - Strategy para diferentes métodos de detección facial.
    - Observer para la activación automática de alertas.
    - Singleton para el manejo de recursos del hardware (cámara).

2. **Vista de Procesos**
    - Diagramas de actividad para:
        - flujo de detección,
        - alerta,
        - registro de eventos.
    - Consideración de concurrencia al procesar frames en tiempo real.
3. **Vista Física**
    - Diagrama de despliegue mostrando:
        - Raspberry Pi 5
        - Cámara integrada
        - Archivos locales o base de datos
        - Sistema físico del vehículo
4. **Vista de Desarrollo**

- Diagrama de componentes y paquetes (OpenCV, Dlib, alertas, GUI).
- Organización del repositorio y del código.

5. **Vista de Escenarios (+1)**
    - Casos de uso:
        - Detectar somnolencia
        - Registrar eventos
        - Emitir alertas
        - Procesar señales faciales

## 4. Desarrollo del Software (Sprints, Incrementos y Daily Scrum)

El desarrollo del prototipo se realizará a través de varios Sprints, siguiendo el enfoque incremental de Scrum.

**Actividades clave:**

- Programación en Python utilizando OpenCV y Dlib.
- Creación del módulo facial (detección de ojos, EAR).
- Implementación del módulo de somnolencia (lógica del algoritmo).
- Desarrollo del módulo de alertas integrado con la Raspberry Pi.
- Construcción de interfaces simples (consola o GUI ligera).
- Integración continua del código en cada Sprint.
- Refactorización permanente para mejorar la calidad.

**Ejemplo de posibles Sprints:**

- **Sprint 1:** Configuración de Raspberry Pi y captura de video.
- **Sprint 2:** Detección facial y puntos de referencia (landmarks).
- **Sprint 3:** Cálculo EAR y lógica de fatiga.
- **Sprint 4:** Sistema de alertas (sonoras / visuales).

- **Sprint 5:** Registro de eventos.
- **Sprint 6:** Pruebas reales en vehículos y optimización.

Cada Sprint culmina con un Incremento del prototipo, listo para ser evaluado.

## 5. Pruebas del Sistema (Testing dentro del Sprint)

En Scrum no se deja el testing para el final; se prueba en cada Sprint.

Se aplicarán:

- **Pruebas unitarias**
  - Validar funciones como:
  - cálculo EAR,
  - detección ocular,
  - lógica de activación de alertas.
- **Pruebas de integración**

  Unir módulos: cámara + análisis facial + alertas.
- **Pruebas de validación**

  Verificar que el sistema cumple los criterios del Product Owner.
- **Pruebas experimentales en campo**

  Realizadas en vehículos reales de Taxi Lagos Courier S.A.:
  - diferentes condiciones de luz,
  - vibración del vehículo,
  - movimiento del conductor,
  - somnolencia simulada y real.
- **Pruebas de rendimiento**

  FPS mínimo aceptable para detección en tiempo real.

- **Pruebas de Usabilidad**

  La usabilidad de los paneles de control web dirigidos a los administradores y operadoras de Taxi Lagos consiste en medir cuán sencillo es el uso de estos.

**1. Implementación y Entrega del Incremento**

Al finalizar cada Sprint se entregará un incremento funcional del prototipo:

- Versión preliminar del algoritmo.
- Hardware configurado.
- Ajustes de la cámara.
- Calibración de detección.
- Pruebas documentadas.

El cliente (Taxi Lagos Courier S.A.) participa en la revisión de Sprint para validar el funcionamiento en condiciones reales.

**6. Retroalimentación, Mejora Continua y Cierre del Proyecto**

Cada Sprint finaliza con una Retrospectiva, donde el equipo evalúa:

- ¿Qué funcionó?
- ¿Qué puede mejorar?
- ¿Qué se ajustará en el siguiente Sprint?

La mejora continua permite optimizar el algoritmo, ajustar la sensibilidad de detección y refinar las alertas.

## 3.2 Técnicas e Instrumentos de Recopilación de Datos

**desarrollar y aplicar los instrumentos para la obtención de resultados**

Para este proyecto las técnicas utilizadas fueron:

- **Revisión bibliográfica**

Permite comprender:

- técnicas de análisis facial,
- algoritmos EAR,
- detección de parpadeo prolongado,
- riesgos de somnolencia.

- **Observación directa**

Realizada en vehículos de Taxi Lagos Courier S.A.:

- comportamiento del conductor,
- condiciones reales del entorno,
- iluminación,
- distancia entre cámara y rostro.

- **Técnica de entrevista**

Para la identificación de actores y la definición de los casos de uso iniciales, se empleó la técnica de entrevista semiestructurada, la cual permitió recolectar información directa de los involucrados en el proceso operativo.

Esta técnica fue aplicada a personal relacionado con la operación de transporte (responsable administrativo), permitiendo comprender las necesidades reales del sistema, así como los escenarios de interacción entre el usuario y la solución tecnológica propuesta.

La información obtenida fue fundamental para la construcción de los diagramas de casos de uso, garantizando que los requerimientos funcionales estén alineados con el contexto real de aplicación.

- **Pruebas de campo**

Validación experimental en situaciones reales de conducción.

Estas técnicas permitieron construir un Product Backlog sólido, comprender el contexto del usuario final y validar los requerimientos funcionales del sistema.

# CAPÍTULO IV

# INGENIERÍA DEL PROYECTO

## 4.1 Requerimientos funcionales

**Tabla 2**

*Requerimiento funcional 1.*

| **Atributo** | **Descripción** |
|---|---|
| ID | RF-001 |
| Nombre | Inicialización del sistema |
| Actores | Conductor |
| Objetivo | Encender y cargar automáticamente todos los módulos del sistema de detección de somnolencia al arrancar el vehículo. |
| Precondiciones | • La Raspberry Pi debe recibir alimentación al encender el vehículo. |
| Flujo de Eventos Principal | 1. El conductor enciende el vehículo.<br>2. La Raspberry Pi inicia el arranque del sistema operativo.<br>3. Se cargan los módulos internos: cámara, red, detección facial y algoritmo de somnolencia.<br>4. Los Leds entran en estado **naranja**, indicando que el sistema está inicializando verificaciones. |
| Flujos Alternativos | • Si algún componente tarda en responder, el sistema permanece en estado naranja hasta validarlo. |
| Postcondiciones | • El sistema queda listo para iniciar las verificaciones restantes. |
| Excepciones | • **Error crítico de inicio:** el sistema no carga alguno de los módulos base y muestra led **rojo** general indicando falla. |

**Tabla 3**

*Requerimiento funcional 2.*

| **Atributo** | **Descripción** |
|---|---|
| ID | RF-002 |
| Nombre | Verificación de conexión a Internet |
| Actores | Conductor |
| Objetivo | Validar la disponibilidad de conexión a Internet para poder enviar registros de somnolencia a la base de datos. |
| Precondiciones | • El módulo Wifi del Raspberry debe estar operativo. |
| Flujo de Eventos Principal | 1. El sistema verifica la disponibilidad de Internet.<br>2. El LED de Internet inicia en naranja (verificando).<br>3. Si existe conexión estable, el LED pasa a verde. |
| Flujos Alternativos | • Si la conexión no se establece tras varios intentos, el LED cambia a rojo.<br>• El sistema continúa funcionando y almacenando datos localmente hasta recuperar conexión. |
| Postcondiciones | • La disponibilidad de Internet queda registrada. |
| Excepciones | • **Fallo del módulo de red**: el sistema marca rojo y registra el error. |

**Tabla 4**

*Requerimiento funcional 3.*

| **Atributo** | **Descripción** |
|---|---|
| ID | RF-003 |
| Nombre | Verificación de conexión de cámara |
| Actores | Conductor |
| Objetivo | Comprobar que la cámara esté conectada y funcional para iniciar la captura de video. |

| | |
|---|---|
| Precondiciones | • La cámara debe estar conectada con el Raspberry Pi físicamente. |
| Flujo de Eventos Principal | 1. El sistema intenta conectar con la cámara.<br>2. El LED de cámara inicia en naranja (verificando).<br>3. Si la cámara responde correctamente, el LED pasa a verde. |
| Flujos Alternativos | • **La cámara no responde:** el LED cambia a rojo y el sistema reintenta periódicamente. |
| Postcondiciones | • La cámara queda lista para enviar imágenes al módulo de detección facial. |
| Excepciones | • Error de hardware: la cámara no transmite video y se reporta el fallo. |

**Tabla 5**

*Requerimiento funcional 4.*

| **Atributo** | **Descripción** |
|---|---|
| ID | RF-004 |
| Nombre | Verificación de detección facial |
| Actores | Conductor |
| Objetivo | Confirmar que el sistema detecta un rostro válido antes de iniciar la detección de somnolencia. |
| Precondiciones | • La cámara debe estar operativa.<br>• El software debe contar con los modelos de detección (OpenCV/Dlib). |
| Flujo de Eventos Principal | 1. La cámara envía frames al sistema.<br>2. El LED de detección facial inicia en naranja (buscando rostro).<br>3. Se utiliza un algoritmo de detección para localizar el rostro.<br>4. Si existe un rostro, se envía la región detectada al módulo de landmarks y el LED pasa a **verde**. |

| | |
|---|---|
| Flujos Alternativos | • **No se detecta rostro:** el LED permanece en **naranja**, el sistema continúa analizando los frames hasta encontrar uno válido. |
| Postcondiciones | • El módulo de detección de somnolencia recibe una región facial válida. |
| Excepciones | • **Fallo en el algoritmo (OpenCV/Dlib):** el LED pasa a **rojo** y se registra la falla. |

**Tabla 6**

Requerimiento funcional 5.

| **Atributo** | **Descripción** |
|---|---|
| ID | RF-005 |
| Nombre | Detección de somnolencia |
| Actores | Conductor |
| Objetivo | Determinar si el conductor presenta síntomas de somnolencia mediante análisis del rostro en tiempo real. |
| Precondiciones | • Debe detectarse rostro (RF-004).<br>• Los modelos de OpenCV, Dlib, landmarks, EAR, MAR deben estar cargados.<br>• El sistema debe contar con los umbrales configurados. |
| Flujo de Eventos Principal | 1. El sistema recibe los frames y detecta los landmarks faciales (ojos y boca).<br>2. Se identifican los puntos correspondientes a cada ojo.<br>3. Los datos se envían al módulo de cálculo EAR.<br>4. El sistema recibe las coordenadas de los ojos.<br>5. Calcula indicadores como:<br>– EAR (Eye Aspect Ratio) → indica cierre de ojos.<br>– MAR (Mouth Aspect Ratio) → indica apertura excesiva de la boca (bostezo).<br>6. El sistema compara los valores obtenidos con umbrales configurados. |

| | |
|---|---|
| | 7. Si EAR bajo + MAR alto + duración cumplen criterios → se detecta somnolencia.<br>8. Envía el valor al módulo de alerta de somnolencia. |
| Flujos Alternativos | • **Variación momentánea:** Si los valores son momentáneos o inconsistentes, se descartan. |
| Postcondiciones | • Se produce un evento que será enviado al módulo de alertas y registro. |
| Excepciones | • **Rostro fuera de cuadro o borroso:** el análisis se suspende temporalmente.<br>• **Datos inestables:** El sistema reinicia el contador de tiempo. |

**Tabla 7**

*Requerimiento funcional 6.*

| **Atributo** | **Descripción** |
|---|---|
| ID | RF-006 |
| Nombre | Generación de alerta |
| Actores | Conductor |
| Objetivo | Alertar al conductor mediante señales sonoras cuando se detecte somnolencia. |
| Precondiciones | • Debe existir un evento de somnolencia prolongado (RF-005). |
| Flujo de Eventos Principal | 1. El sistema identifica un posible estado de somnolencia.<br>2. El sistema activa una alerta sonora mediante buzzer/parlante.<br>3. Mantiene la alerta mientras la somnolencia persista y registra el evento. |
| Flujos Alternativos | • Al cesar la somnolencia, la alerta se apaga automáticamente. |
| Postcondiciones | • El conductor es advertido del riesgo. |

| | |
|---|---|
| Excepciones | • **Fallo del buzzer/parlante:** El sistema registra el error. |

**Tabla 8**

*Requerimiento funcional 7.*

| **Atributo** | **Descripción** |
|---|---|
| ID | RF-007 |
| Nombre | Registro del evento de somnolencia |
| Actores | Conductor |
| Objetivo | Almacenar únicamente los eventos de somnolencia detectados para su posterior análisis. |
| Precondiciones | • Somnolencia confirmada. |
| Flujo de Eventos Principal | 1. Al generarse un evento de somnolencia (RF-005).<br>2. El módulo de registro crea un registro y registra fecha, hora, duración y datos mínimos del evento.<br>3. Envía el registro a la base de datos. |
| Flujos Alternativos | • **Sin Internet:** guarda en memoria local y sincroniza más tarde. |
| Postcondiciones | • El evento queda almacenado en la Base de Datos con información mínima: timestamp, EAR promedio, duración del evento, tipo de alerta, para posterior realizar análisis. |
| Excepciones | • Error de escritura en base de datos. |

**Tabla 9**

*Requerimiento funcional 8.*

| **Atributo** | **Descripción** |
|---|---|
| ID | RF-008 |
| Nombre | Autenticación |
| Actores | Gerente |

| | |
|---|---|
| Objetivo | Permitir que el gerente acceda al portal web con credenciales. |
| Precondiciones | • Conexión a Internet. |
| Flujo de Eventos Principal | 1. El gerente ingresa usuario y contraseña.<br>2. El sistema valida credenciales.<br>3. Acceso concedido. |
| Flujos Alternativos | • **Credenciales incorrectas:** mensaje de error. |
| Postcondiciones | • Acceso autorizado. |
| Excepciones | • Falla del servidor o base de datos. |

**Tabla 10**

*Requerimiento funcional 9.*

| **Atributo** | **Descripción** |
|---|---|
| ID | RF-009 |
| Nombre | Consulta de historial de eventos de somnolencia |
| Actores | Gerente |
| Objetivo | Permitir al gerente visualizar estadísticas de somnolencia de sus conductores. |
| Precondiciones | • Usuario autenticado.<br>• Base de datos accesible. |
| Flujo de Eventos Principal | 1. El gerente solicita estadísticas.<br>2. El sistema consulta la base de datos.<br>3. Muestra estadística de eventos por conductor/ruta/general. |
| Flujos Alternativos | • **No hay datos:** se muestra “Sin registros”. |
| Postcondiciones | • Estadísticas visualizadas. |
| Excepciones | • Falla en conexión al servidor. |

**Tabla 11**

*Requerimiento funcional 10.*

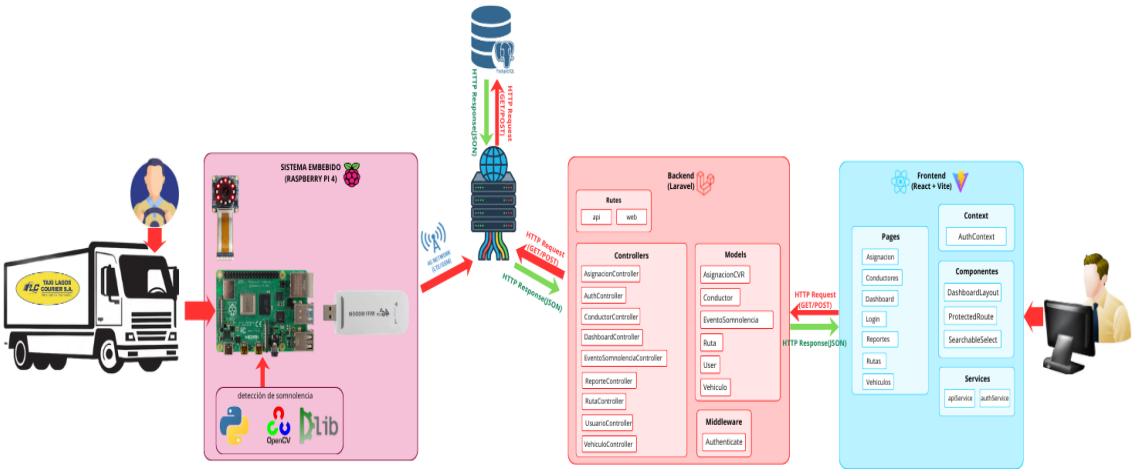
| **Atributo** | **Descripción** |
|---|---|
| ID | RF-010 |
| Nombre | Generación de reportes |
| Actores | Gerente |
| Objetivo | Permitir exportar un reporte de los eventos registrados en formato PDF. |
| Precondiciones | • Datos disponibles. |
| Flujo de Eventos Principal | 4. Gerente solicita reporte.<br>2. El sistema compila registro de eventos.<br>3. Exporta documento PDF. |
| Flujos Alternativos | • **Datos incompletos:** reporte parcial. |
| Postcondiciones | • Reporte generado. |
| Excepciones | • Error al generar archivo. |

## 4.2 Diseño

### 4.2.1 Arquitectura del software

**Figura 2**

*Diseño de arquitectura del sistema de detección de somnolencia.*

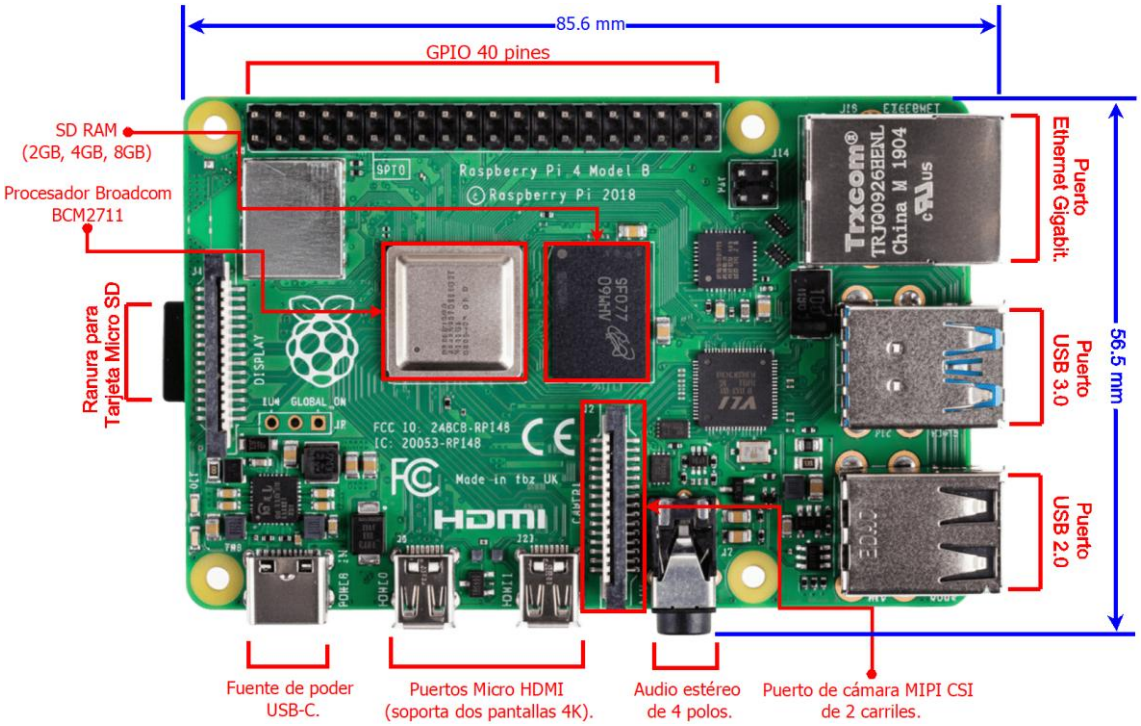


## HARDWARE

### Raspberry Pi 4

**Figura 3**

*Partes del Raspberry Pi 4 Model B.*




– **Características**

- **Modelo:** Raspberry Pi 4 Model B.
- **Dimensiones:** 85.6 mm X 56.5 mm X 17 mm
- **Procesador:** Broadcom BCM2711.
- **CPU:** Procesador de cuatro núcleos a 1,5 GHz con brazo Cortex-A72.
- **GPU:** VideoCore VI.
- **Memoria:** 2/4/8GB LPDDR4 RAM.
- **Conectividad:**
  - 802.11ac Wi-Fi.
  - Bluetooth 5.0.
  - Puerto Ethernet Gigabit.
- **Vídeo y sonido:**
  - 2 x puertos micro-HDMI que admiten pantallas de 4K@60Hz a través de HDMI 2.0.
  - Puerto de pantalla MIPI DSI.
  - Puerto de cámara MIPI CSI.
  - Salida estéreo de 4 polos y puerto de vídeo compuesto.
- **Puertos:** 2 x USB 3.0, 2 x USB 2.0.
- **Alimentación:** 5V/3A vía USB-C.
- **Expansión:** Cabezal GPIO de 40 pines

**CAMARA NOCTURNA DIURNA SENSOR IR AUTOMÁTICO 5MP PARA RASPBERRY PI**

**Figura 4**

*Cámara 5MP para Raspberry.*



– **Características**

- **Marca:** Arducam.
- **Tecnología de Sensores Fotográficos:** CMOS.
- **Resolución de captura de vídeo:** 1080p 30 fps, 720p 60fps, 480p 90fps
- **Distancia focal máxima:** 2,25 milímetros
- **Apertura máxima:** 2.1 f
- **Formato de captura de vídeo:** MP4
- **Formato de audio compatible:** AAC
- **Tecnología de Conectividad:** Infrarrojo

***Nota***. Adaptado de *Cámara para Raspberry* [fotografía], por MTheiler, 2018, WikiMedia commons (http://dlib.net/)

**MEMORIA SD**

**Figura 5**

*Tarjeta Micro SD SanDisk 64G.*




– **Características**

- **Marca:** Sandisk.
- **Almacenamiento:** 64 Gb.
- **Clase:** 10.
- **Dimensiones**: 15 mm X 11 mm X 1,0 mm.
- **Temperatura de funcionamiento:** -13ºF a 185ºF (-25ºC a 85ºC).
- **Temperatura de almacenamiento:** -40 ° F a 185ºF (-40ºC a 85ºC).
- **Compatibilidad:**
  - MicroSDHC.
  - MicroSDXC.
  - microSDHC UHS-I.
  - microSDXC UHS-I apoyar dispositivos host.

***Nota.*** Adaptado de *Sandisk Micro SD Ultra 64GB* [fotografía], por Multicom, 2015, Multicom

**SOFTWARE**

**Librería Dlib**

**Figura 6**

*Logo de la Librería Dlib.*




– **Características:**

- **Versión: serie** 19.24.
- **Lenguaje Base:** C++.
- **Enlaces oficiales:** Python.
- **Tipo de librería:** Machine Learning y Computer Visión.
- **Arquitectura:** Modular.
- **Procesamiento facial:** Detección de rostros.

***Nota***. Adaptado de *Logotipos de la Biblioteca Dlib c++* [fotografía], por MTheiler, 2019, WikiMedia commons (http://dlib.net/)

**OpenCV**

**Figura 7**

*Logo de OpenCV.*




– **Características:**

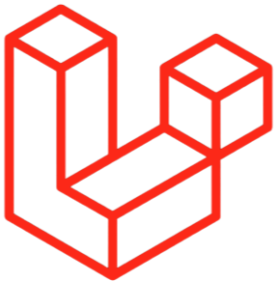
- **Versión:** 4.6**.**
- **Lenguaje** Base: C++.
- **Enlaces oficiales:** Python, Java, C#.
- **Tipo de librería:** Visión por computadora.
- **Arquitectura:** Modular.
- **Procesamiento de imágenes:** Amplio.
- **Procesamiento de video:** Captura y análisis.

***Nota***. Adaptado de *OpenCV Logo* [fotografía], por Adi Shavit, 2006, WikiMedia commons (http://opencv.willowgarage.com/wiki/OpenCVLogo)

**Laravel**

**Figura 8**

*Logo de Laravel v12.51*



- **Características:**

- **Versión:** 12.51.
- **Tipo de framework:** Framework PHP MVC.
- **Lenguaje base:** PHP.
- **Versión de PHP requerida:** PHP 8.2+.
- **Arquitectura:** MVC.

***Nota***. Adaptado de *Logotipo de Laravel (versión vectorial)* [fotografía], por AurelDemiri, 2019, WikiMedia commons (https://laravel.com)

**React**

**Figura 9**

*Logo de React v19.2*



- **Características:**

- **Versión:** 19.2.
- **Tipo de tecnología:** Biblioteca JavaScript.
- **Lenguaje base:** JavaScript / TypeScript.
- **Paradigma:** Basado en componentes.
- **Arquitectura:** Declarativa.

***Nota***. Adaptado de *El logotipo de la biblioteca JavaScript de código abierto React*[fotografía], por Richienb, 2022, WikiMedia commons (https://raw.githubusercontent.com/reactjs/reactjs.org/main/src/icons/logo.svg)

**PostgreSQL**

**Figura 10**

*Logo de la base de datos PostgreSQL.*




–**Característica:**

- **Versión:** 18.2.
- **Tipo de SGBD:** Base de datos relacional objeto-relacional.
- **Lenguaje de consulta:** SQL estándar.
- **Arquitectura:** Cliente-servidor.
- **Control de concurrencia:** MVCC (Multiversion Concurrency Control para alta concurrencia.).
- **Escalabilidad:** Alta.

***Nota***. Adaptado de *El logo elefante azul/blanco del RDBMS de PostgreSQL*[fotografía], por Daniel Lundin, 2008, WikiMedia commons (http://pgfoundry.org/docman/view.php/1000089/124/elephant.svgz)

**Tabla 12**

*Tabla de presupuesto del sistema.*

| CATEGORIA | DETALLE | COSTO (USD) |
|---|---|---|
| **Hardware** | Cámara nocturna diurna Sensor IR automático 5MP para Raspberry Pi | $40.00 |
| | Modulo GSM 4G USB | $10.00 |
| | *Tarjeta Micro SD SanDisk 64G* | $15.65 |
| | Raspberry Pi 4 | $ 117.39 |
| | Cargador Raspberry Pi 4 Oficial 5v 3.1a | $21.74 |
| | Carcaza de Raspberry | $6.00 |
| | Buzzer | $0.50 |
| | Accesorios (cables, adaptadore, soporte de cámara) | $10.00 |
| **Software** | Python | $0.00 |
| | OpenCV,Dlib | $0.00 |
| | Laravel v.12 | $0.00 |

| | | |
|---|---|---|
| | React v.19 | $0.00 |
| **Servidor y Dominio** | Render | $13.00 |
| | Cloudflare | $13.00 |
| | **TOTAL** | **$247.28** |

**INTERFACES**

En la interfaz se está utilizando una aplicación web SPA, que fue desarrollado con el framework React y Vite, usa una estructura de panel administrativo con un menú lateral fijo y un área de contenido principal. Se está utilizando una tipografía (Sans Serif), también usa distintos pesos para jerarquizar títulos, secciones y textos; los títulos tienen aproximadamente 30 px, el texto general 14 px y las etiquetas 10 px y 12 px.

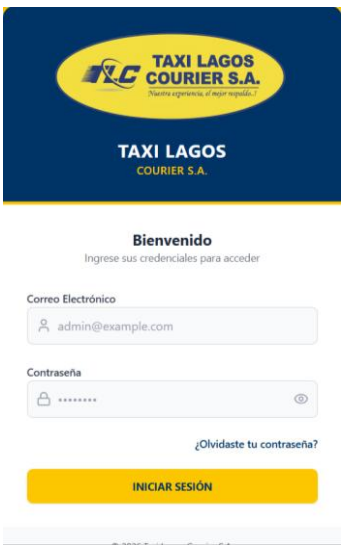
Utiliza fuentes monoespaciadas para los textos que son números. Las imágenes de los avatares miden 40 x 40 px, los botones tienen esquinas redondas y colores por funcionalidad. El menú lateral tiene un ancho estándar de 256 px y es responsivo a pantallas móviles. Por último, la interfaz cuenta con tablas, formularios, modales, gráficos interactivos y una paleta de colores en tonos (navy y dorado).

- **Loguin para administrador y gerente o propietario.**

Permiso de solo acceso al administrador o gerente.

**Figura 11**

*Loguin para ingreso al interfaz de administrador/gerente o propietario.*

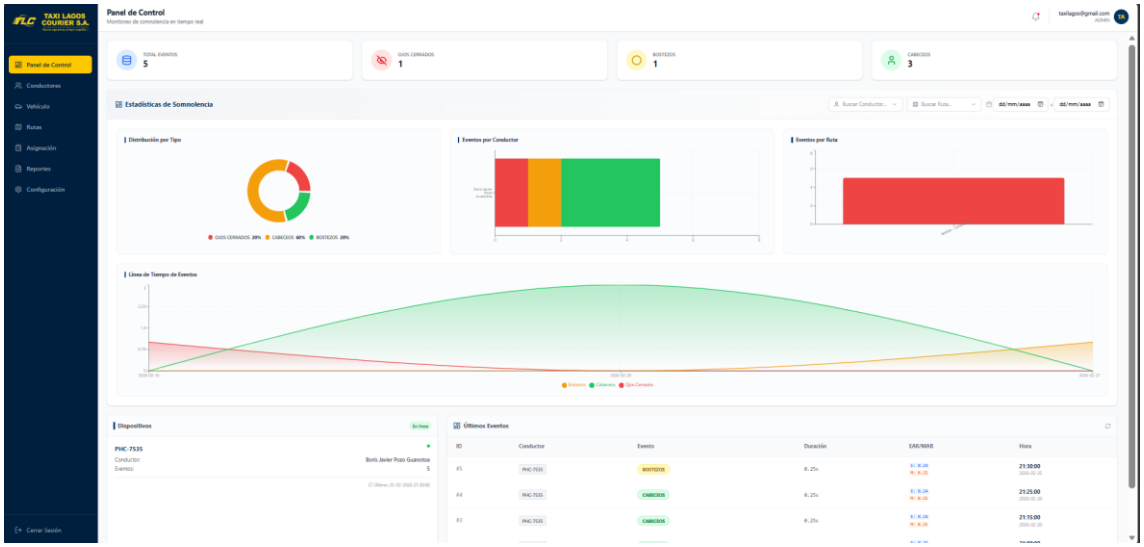


- **Panel de control principal.**

Panel principal donde se muestra los eventos de somnolencia detectados, con sus respectivos gráficos estadísticos distribuidos por conductor, ruta y tiempo.

**Figura 12**

*Interfaz de Panel de Control para el monitoreo estadístico de somnolencia en conductores, en tiempo real.*

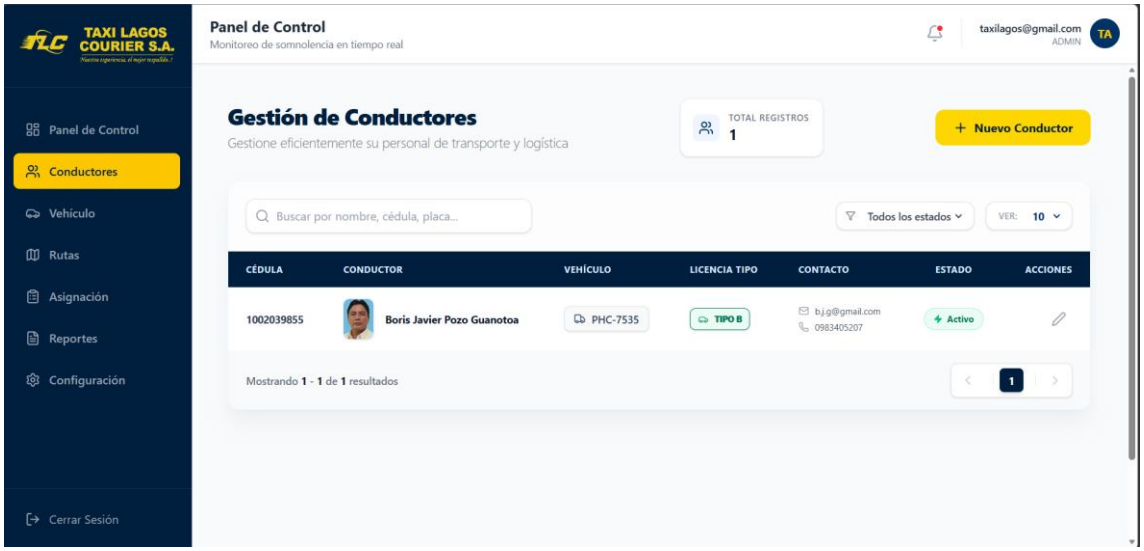



- **Gestión de conductores.**

Listado de conductores registrados, con datos de identificación, vehículo asignado, tipo de licencia y estado actual.

**Figura 13**

*Interfaz de Gestión de Conductores - Listado de Conductores Registrados en el Sistema.*

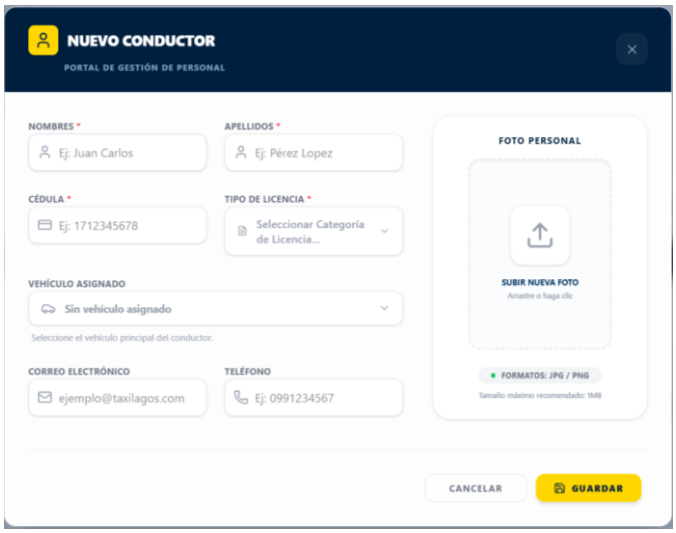


- **Formulario de registro de nuevo conductor.**

Formulario para ingresar los datos personales del conductor.

**Figura 14**

*Modal de Registro de Nuevo Conductor - Portal de Gestión de Personal.*

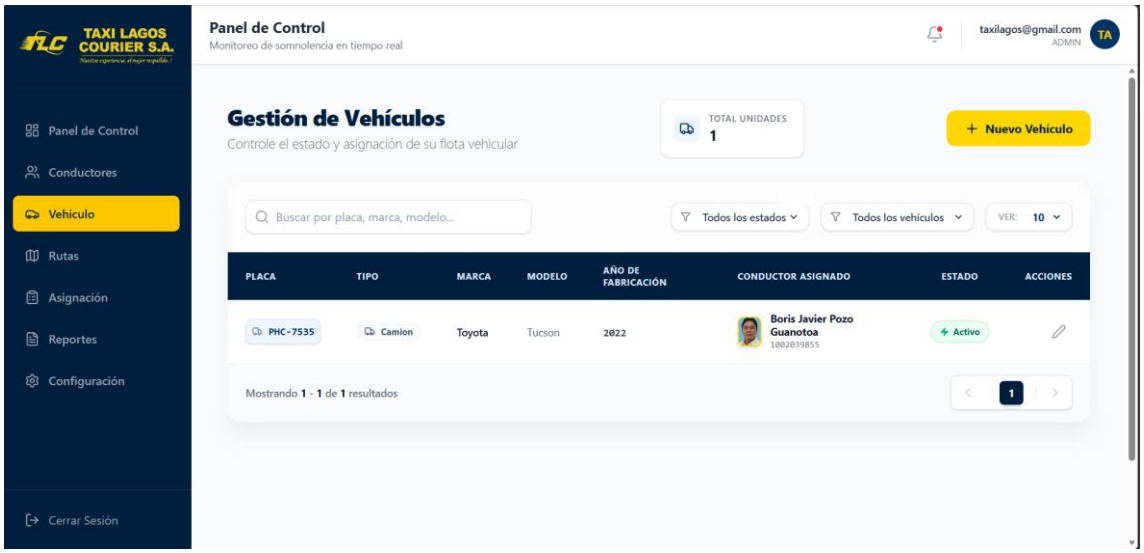



- **Gestión de vehículos.**

Listado de vehículos registrados, con información de placa, marca, modelo, año de fabricación y conductor asignado.

**Figura 15**

*Interfaz de Gestión de Vehículos - Listado de Unidades de la Flota Vehicular.*

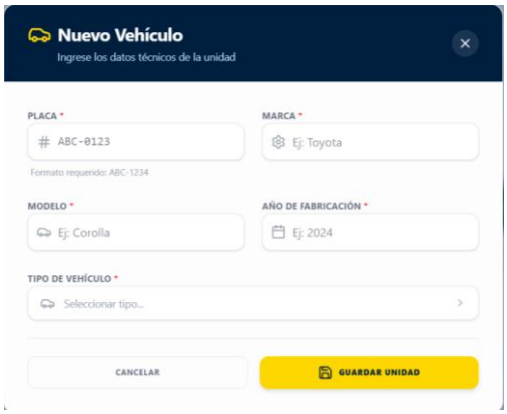


– **Formulario de registro de nuevo vehículo.**

Formulario para el ingreso de los datos técnicos de cada vehículo.

**Figura 16**

*Modal para el Registro de Nuevo Vehículo - Ingreso de Datos Técnicos de la Unidad.*

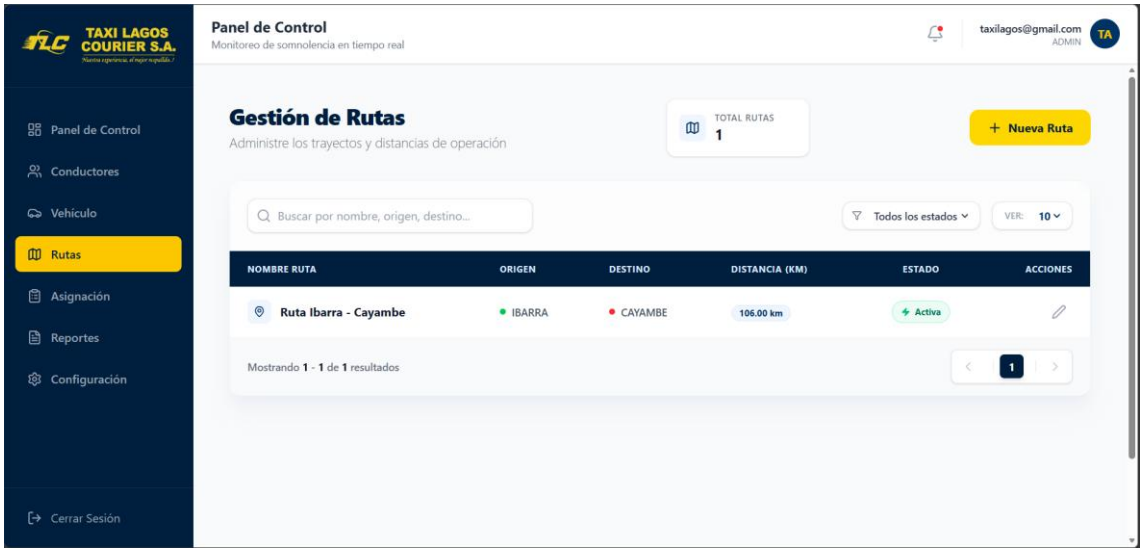



– **Gestión de rutas.**

Listado de rutas registrados.

**Figura 17**

*Interfaz de Gestión de Rutas - Administración de Trayectos y Distancias de Operación.*

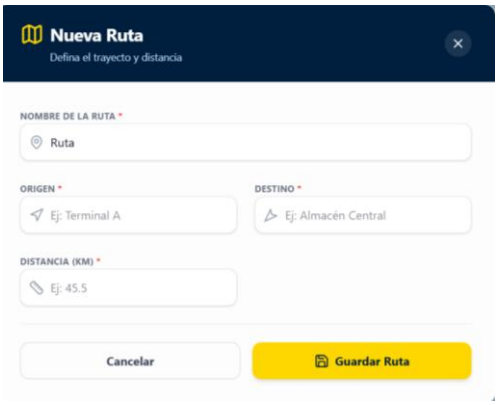


- **Formulario de registro de nueva ruta.**

Formulario para registrar una nueva ruta de operación.

**Figura 18**

*Modal para la Creación de Nueva Ruta - Definición de Trayecto y Distancia.*

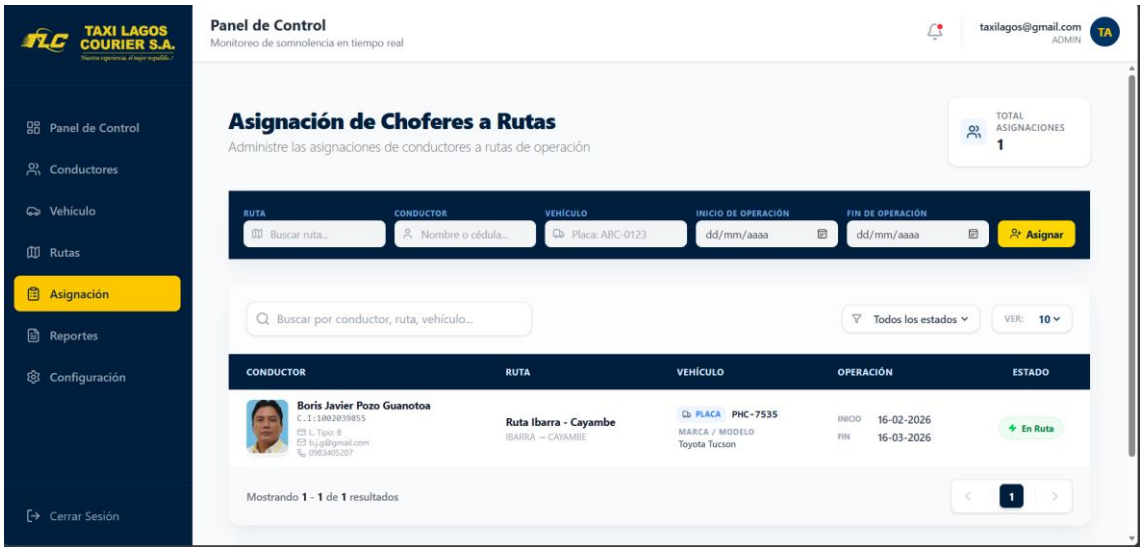



- **Asignación de conductores a rutas**

Módulo donde se asignan los conductores a sus rutas de operación.

**Figura 19**

*Interfaz de Asignación de Conductores a Rutas - Sistema de Asignación de Operaciones.*

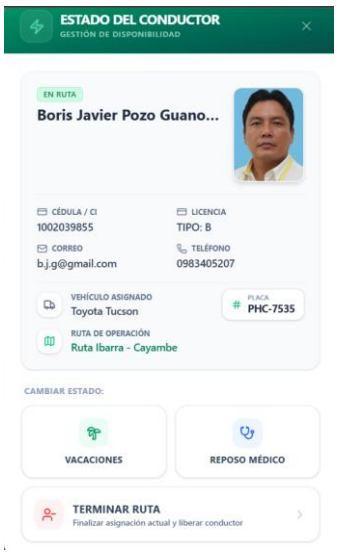



**Estado del conductor en ruta.**

Vista del modal del estado actual del conductor, con opciones para registrar vacaciones, reposo médico o dar por finalizada la ruta.

**Figura 20**

*Modal de Estado del Conductor - Gestión de Disponibilidad y Estado Operativo en Ruta.*




- **Gestión de conductor pendiente.**

Vista del modal para gestionar el regreso de un conductor tras una ausencia, permitiendo reactivarlo en su misma ruta o terminar la asignación.

**Figura 21**

*Modal para la Gestión de Conductor Pendiente - Registro de Ausencia y Opciones de Reasignación.*

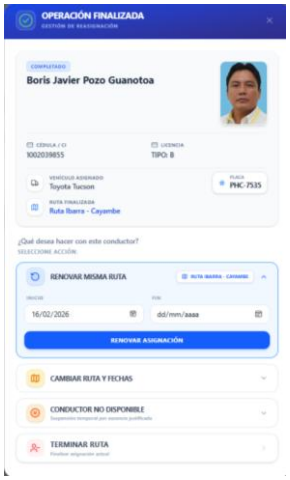



– **Operación completada.**

Vista del modal que aparece al concluir una operación, desde donde se puede renovar la ruta, asignar una nueva o registrar al conductor como no disponible.

**Figura 22**

*Modal para la Operación Completada - Gestión de Reasignación y Cambio de Estado del Conductor.*

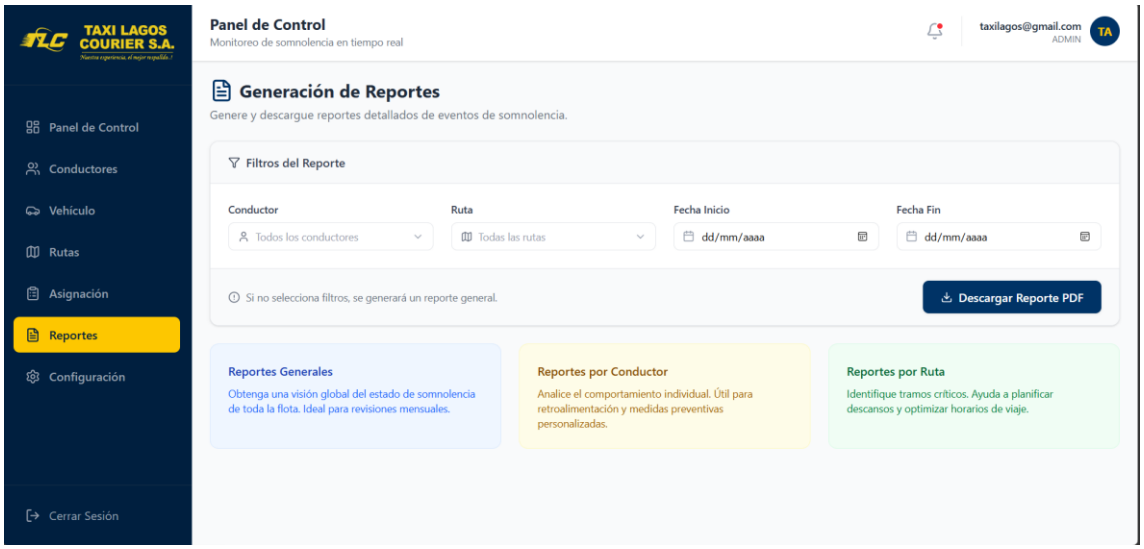



– **Generación de reportes.**

Módulo para generar y descargar reportes en PDF de los eventos de somnolencia, con filtros por conductor, ruta y fechas.

**Figura 23**

*Interfaz para la Generación de Reportes - Sistema de Filtros y Descarga de Reportes de Somnolencia.*




– **Reporte específico generado en PDF.**

Reporte generado con el perfil del conductor, resumen de eventos de somnolencia registrados y detalle cronológico de cada uno durante la operación.

**Figura 24**

*Reporte Específico de Conductor - Detalle Cronológico de Eventos de Somnolencia Registrados.*

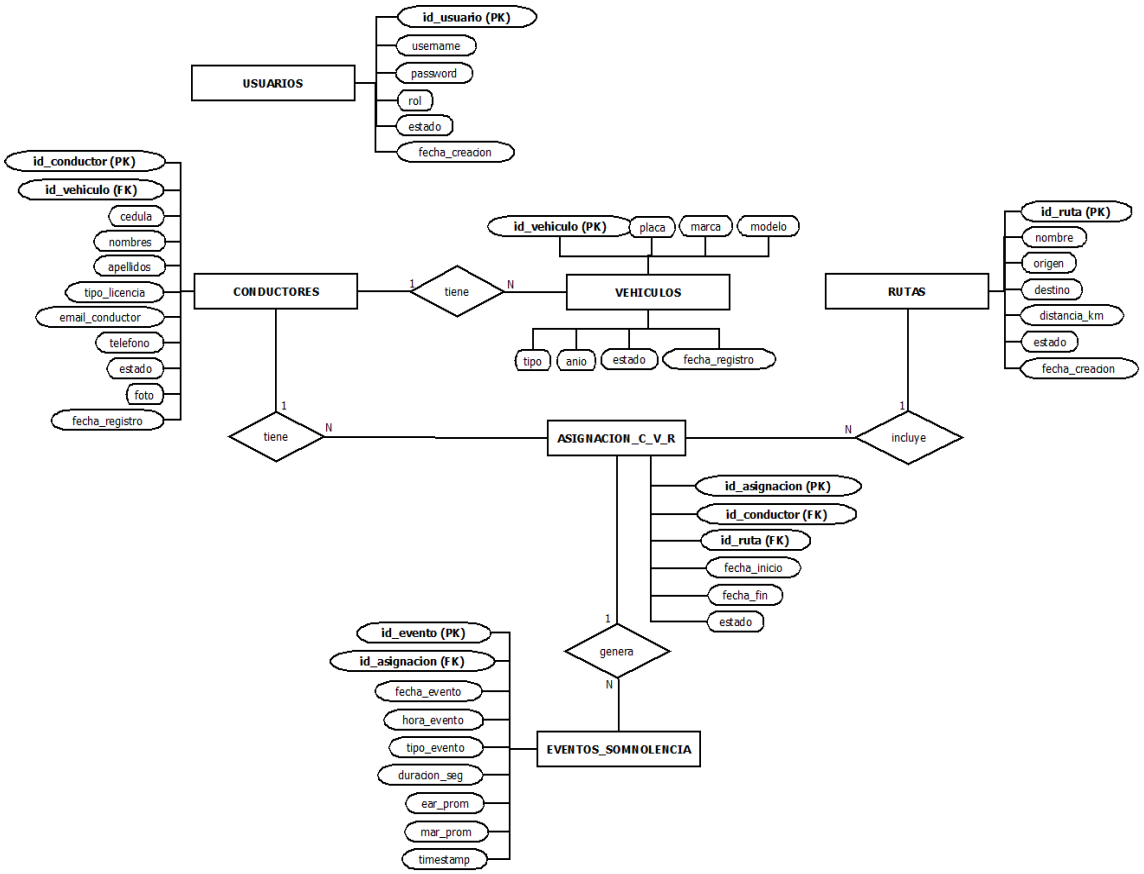



### 4.2.2 Diagrama entidad relación

A continuación, se presenta el diagrama entidad-relación del sistema de detección de somnolencia en los conductores en tiempo real.

**Figura 25**

*Diagrama Entidad Relación de la Base de Datos para Detección de Somnolencia en conductores.*



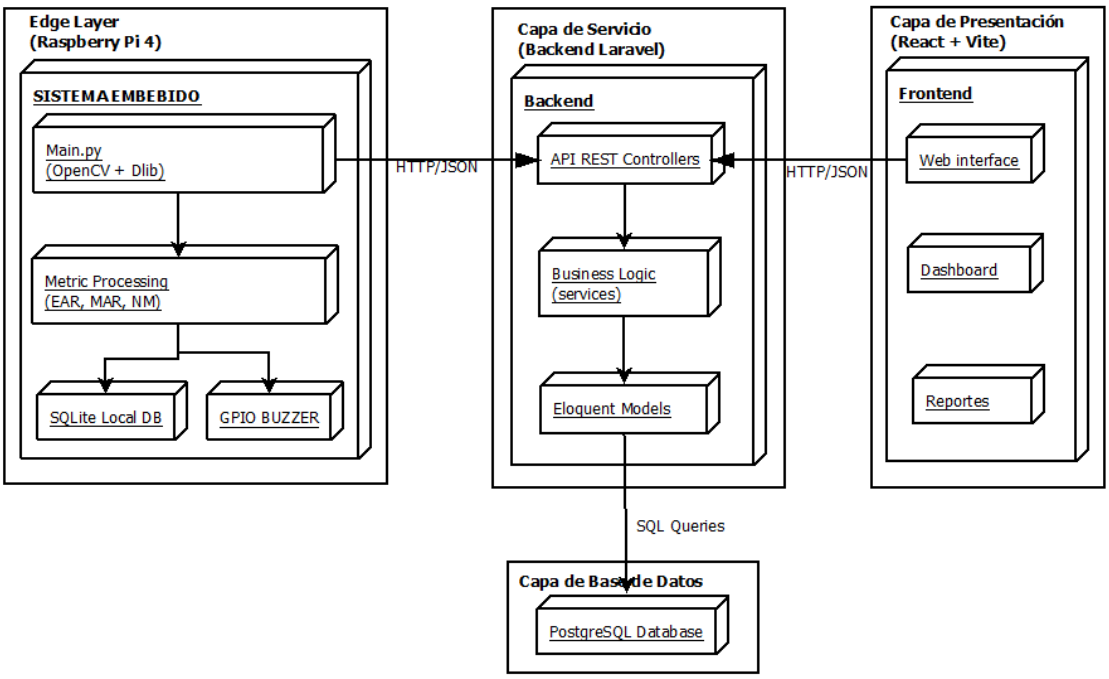

### 4.2.3 Modelo relacional

1. **Usuarios** (id_usuario, username, password, rol, estado, fecha_creacion)
2. **Vehículos** (id_vehiculo, placa, modelo, marca, anio, tipo, estado, fecha_registro)
3. **Conductores** (id_conductor, nombres, apellidos, cedula, tipo_licencia, email_conductor, telefono, foto, id_vehiculo, estado, fecha_registro)
4. **Rutas** (id_ruta, nombre, origen, destino, distancia_km, estado, fecha_creacion)
5. **Asignacion_c_v_r** (id_asignacion, id_conductor, id_ruta, fecha_inicio, fecha_fin, estado)
6. **Eventos_somnolencia** (id_evento, id_asignacion, fecha_evento, hora_evento, tipo_evento, duracion_seg, ear_prom, mar_prom, timestamp)

### 4.2.4 Patrón de diseño arquitectónico en Capas Distribuida con enfoque Edge – Cloud

**Figura 26**

*Diagrama de patrón arquitectónico en Capas Distribuida con enfoque Edge–Cloud.*




### 4.2.5 Vista arquitectónicas 4+1

**Vista de Escenarios**

Esta vista representa de manera general las funcionalidades principales del sistema y su interacción con los actores externos. Los actores identificados son el Conductor y el Gerente/Propietario, quienes interactúan con el sistema según sus roles y responsabilidades.

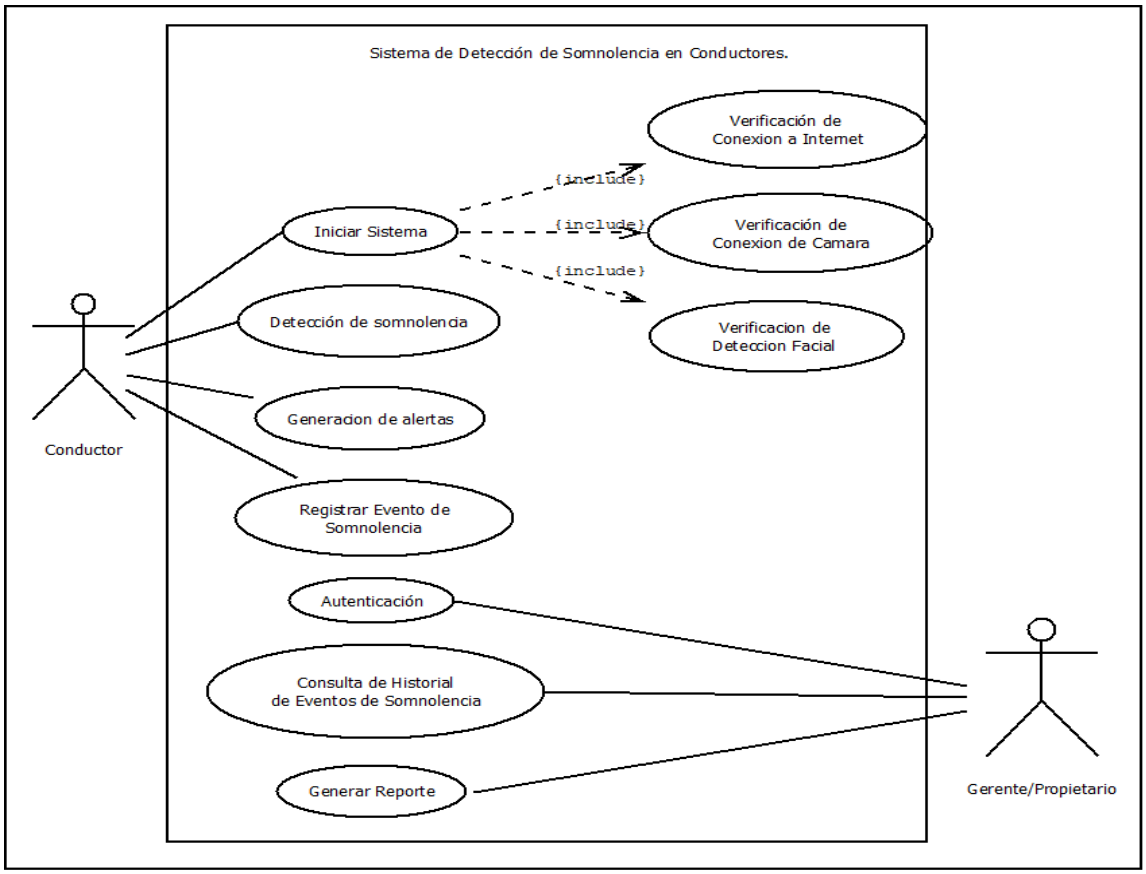
- **Conductor:** Es el actor principal, porque interactúa directamente con el sistema, mientras que él conduce. Bajo esta línea, el primer paso es el arranque del sistema, que comprobará si la cámara, el internet, su reconocimiento facial, están operativos. Luego el sistema realiza un análisis de la cara del conductor, de manera continua. En el caso de que el sistema identifica que el conductor muestra signos de somnolencia, el sistema emite alertas de sonido, con el fin de notificar al conductor y disminuir el riesgo de un accidente de tránsito. Al mismo tiempo,

el sistema realiza el registro de características relevantes del evento, tales como la fecha, hora, lugar, y datos visuales.

- **Gerente/ Propietario:** Este actor accede al sistema a través de un sitio web. Para ello, debe primero autenticarse, lo que permite asegurar que dicha información, a la que él tiene acceso, permanece restringida. Dentro de su perfil, puede ver el historial de eventos de somnolencia, así como generar reportes de estos. Esto permite que el usuario realice un análisis de la información que tiene, con relación a la conducta de los conductores, así como adoptar decisiones que conlleven a un mejoramiento de la seguridad vial.

**Figura 27**

*Diagrama de caso de uso (escenarios).*




**Vista de Diseño Lógica**

El diagrama de clases muestra la composición y estructura del sistema de detección de somnolencia, mostrando las entidades de software, sus atributos, métodos y sus relaciones. Este modelo ayuda a entender la composición y cómo los diferentes componentes trabajan en conjunto para cumplir los requerimientos.

La solución en cuestión se compone de 2 dominios principales, el sistema embebido para la detección en tiempo real y el backend para la gestión y persistencia de la información. Las clases de MainDetector, GestorAlarma, BuzzerAlarma, EventoDB y MetricCalculator se resaltan por encapsular la lógica de la captura de video, procesamiento biométrico, generación de alertas y registro de eventos.

La clase MainDetector coordina el proceso de detección, donde se junta las capturas de imágenes usando OpenCV, detección de caras con Dlib y evaluación de los indicadores EAR, MAR y NM. La clase MetricCalculator es la que delega las operaciones geométricas requeridas para el cálculo de estas métricas sobre los puntos faciales.

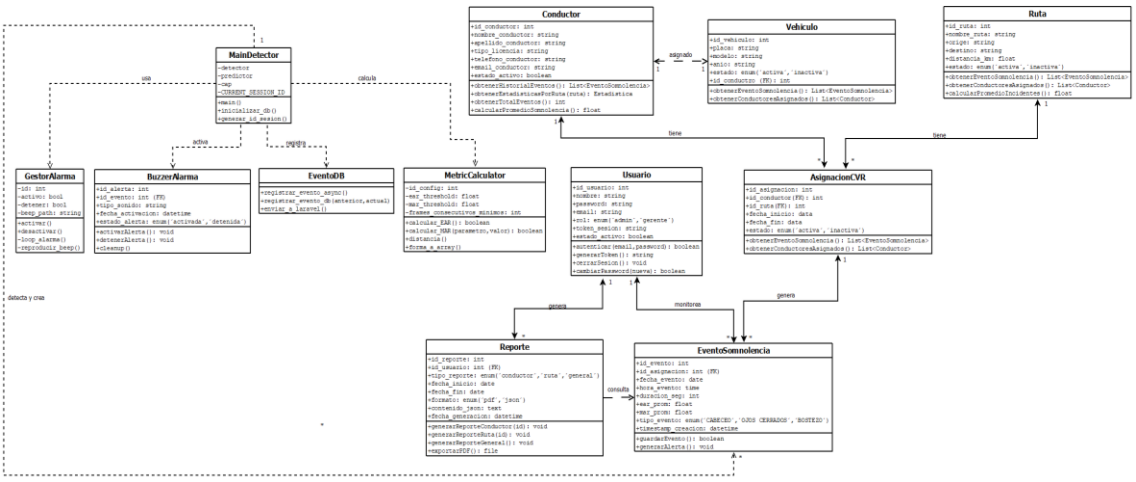
La clase EventoDB maneja la duración local de eventos de somnolencia y la sincronización con el backend a través del servicio REST, asegurando que el procesamiento en tiempo real no se vea bloqueado por operaciones de almacenamiento.

El backend modela las entidades Conductor, Vehículo, Ruta, AsignacionCVR, EventoSomnolencia y Usuario, que representan la estructura de datos del sistema de gestión. La relación principal se encuentra en AsignacionCVR, que relaciona un conductor con un vehículo y una ruta específica, permitiendo vincular cada evento de somnolencia detectado.

La entidad EventoSomnolencia tiene una relación de uno a muchos con AsignacionCVR, permitiendo el registro de múltiples eventos durante una única asignación operativa. Este diseño favorece la trazabilidad histórica y el análisis posterior del comportamiento del conductor.

**Figura 28**

*Diagrama de clases (lógica).*




**Vista de Despliegue**

En la vista de despliegue se establece mediante un diagrama de componentes la arquitectura general del sistema, el cual se encuentra dividido en tres capas principales: Sistema Embebido, Frontend Web y Backend (API REST).

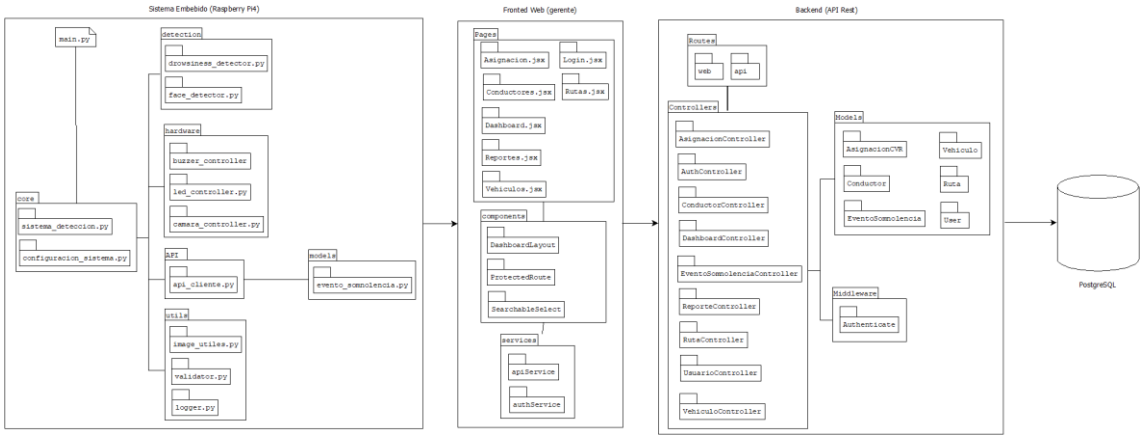
El sistema embebido, construido sobre una Raspberry Pi 4, incorpora módulos para la detección de somnolencia, control de hardware (cámara, zumbador y LEDs), procesamiento de imágenes y comunicación con el servidor API. Este componente recopila datos y realiza detección en tiempo real.

El Frontend Web permite al gerente o propietario interactuar con el sistema. Contiene componentes y servicios que utilizan la API REST para mostrar estadísticas, eventos e informes. Esta parte del sistema se encarga de la presentación de datos y la experiencia general del usuario.

Por último, el Backend aplica la lógica de negocio a través de controladores, modelos y servicios. Este componente maneja la autenticación de usuarios, el almacenamiento de eventos de somnolencia, el cálculo de estadísticas y la generación de informes, empleando una base de datos MySQL para la persistencia de datos. La comunicación entre componentes se realiza a través de servicios REST, asegurando que la arquitectura esté desacoplada y sea escalable.

**Figura 29**

*Diagrama de componentes (despliegue).*




**Vista de Procesos**

Para la vista de procesos se representa en un diagrama de actividades, los cuales representa el flujo principal que hay en el sistema para la **detección de signos de somnolencia (Sistema Embebido) y la consulta del evento de la somnolencia (Sitio Web).**

**Sistema de Detección**

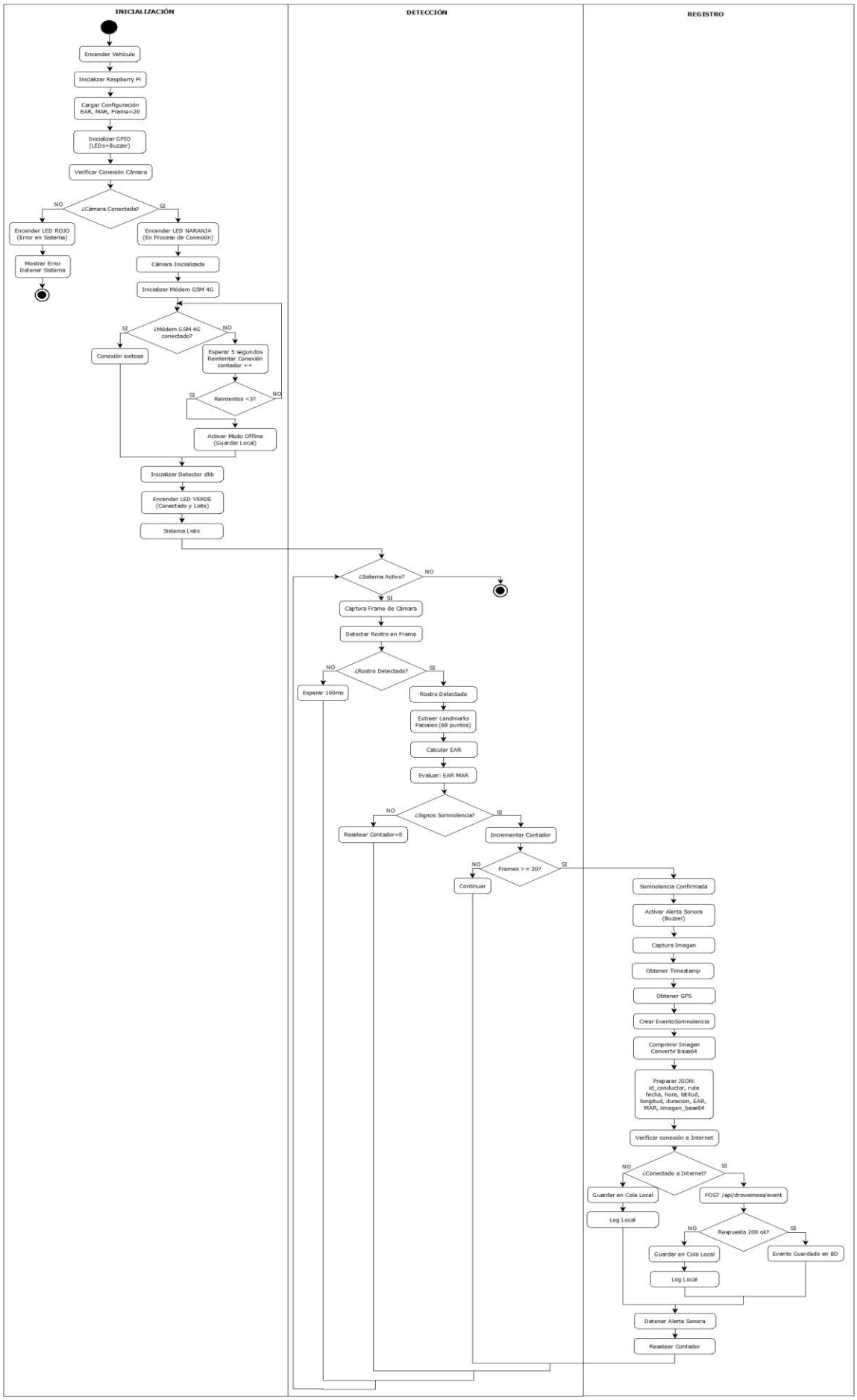
El diagrama de actividades del sistema embebido describe en detalle cómo el sistema funciona. En este caso, comenzamos cuando se enciende el vehículo, provocando el inicio del sistema en la Raspberry Pi. En esta etapa, se cargan las configuraciones, se revisa la conexión de la cámara y la conexión a internet mediante el modem GSM 4G .

Luego de la revisión de hardware, el sistema se activa y comienza el proceso de detección. La cámara comienza a tomar fotos y el sistema inicia el algoritmo de detección facial. En caso de no reconocerse una cara, el sistema deberá esperar y volver a intentarlo. Si una cara es reconocida, se tomarán el número de puntos faciales necesarios para el cálculo de los indicadores de sueño.

Si la cantidad de fotos y los valores se encuentran por encima del nivel configurado, el sistema confirma que el conductor presenta somnolencia. En este caso, se activa la alarma, se toma una foto como respaldo, se guarda la hora y la ubicación. Finalmente, el sistema revisa si está conectado a internet; si lo está, envía la información sobre el evento al servidor por medio de la API REST.

**Figura 30**

*Diagrama de actividades, sistema embebido (Detección y Registro).*


INICIALIZACIÓN
DETECCIÓN
REGISTRO
Encender Vehículo
Inicializar Raspberry Pi
Cargar Configuración EAR, MAR, Frame=20
Inicializar GPIO (LEDs+Buzzer)
Verificar Conexión Cámara
¿Cámara Conectada?
NO
SI
Encender LED ROJO (Error en Sistema)
Mostrar Error Detener Sistema
Encender LED NARANJA (En Proceso de Conexión)
Cámara Inicializada
Inicializar Módem GSM 4G
¿Módem GSM 4G conectado?
Conexión exitosa
Esperar 5 segundos Reintentar Conexión contador ++
Reintentos <3?
Activar Modo Offline (Guardar Local)
Inicializar Detector dlib
Encender LED VERDE (Conectado y Listo)
Sistema Listo
¿Sistema Activo?
Captura Frame de Cámara
Detectar Rostro en Frame
¿Rostro Detectado?
Esperar 100ms
Rostro Detectado
Extraer Landmarks Faciales (68 puntos)
Calcular EAR
Evaluar: EAR MAR
¿Signos Somnolencia?
Resetear Contador=0
Incrementar Contador
Frames >= 20?
Continuar
Somnolencia Confirmada
Activar Alerta Sonora (Buzzer)
Captura Imagen
Obtener Timestamp
Obtener GPS
Crear EventoSomnolencia
Comprimir Imagen Convertir Base64
Preparar JSON: id_conductor, ruta fecha, hora, latitud, longitud, duracion, EAR, MAR, iimagen_base64
Verificar conexión a Internet
¿Conectado a Internet?
Guardar en Cola Local
Log Local
POST /api/drowsiness/event
Respuesta 200 ok?
Guardar en Cola Local
Log Local
Evento Guardado en BD
Detener Alerta Sonora
Resetear Contador

**Consulta del evento de la somnolencia**

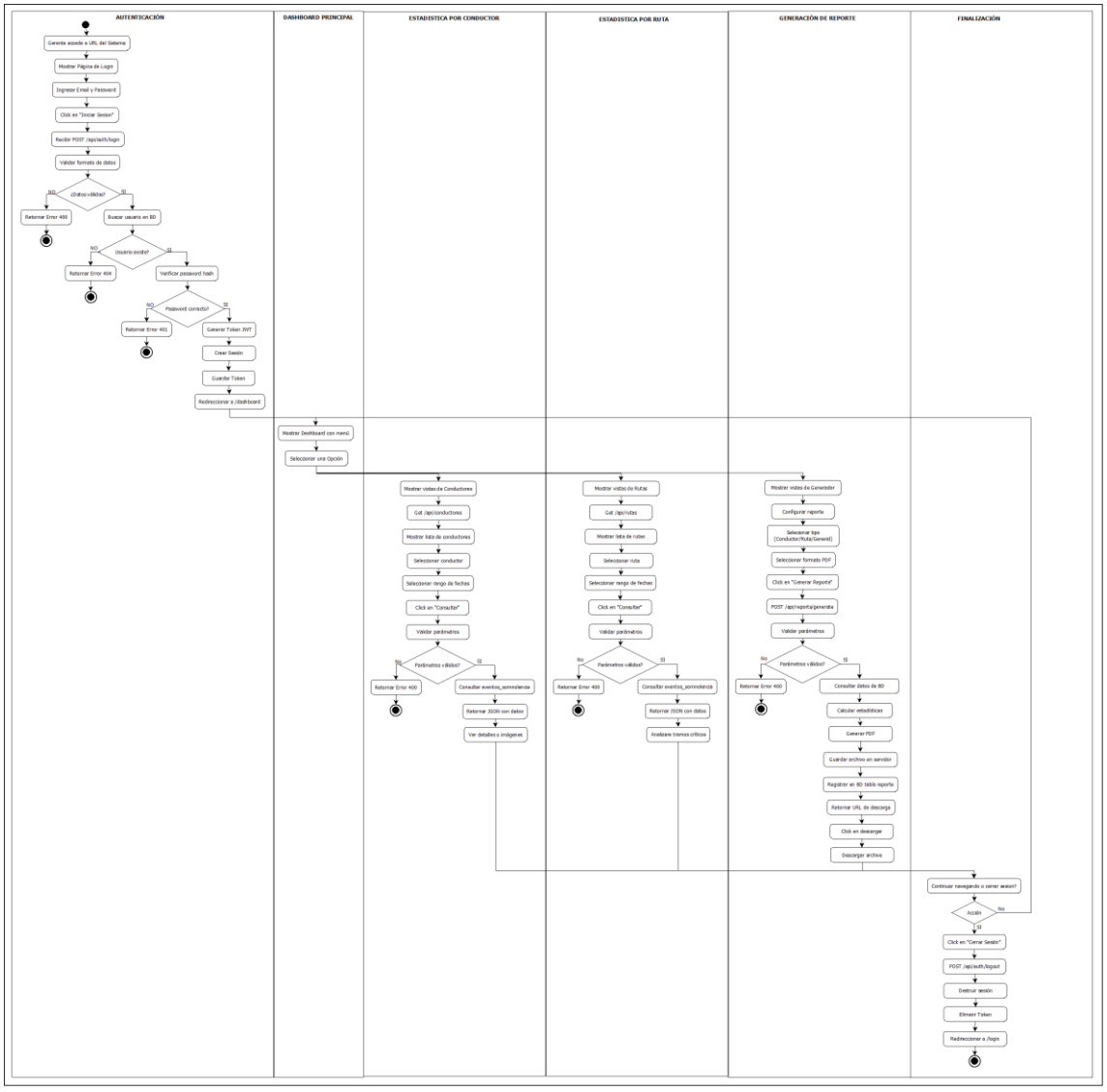
El diagrama de actividad del sitio web describe el flujo de acciones desde el ingreso de un usuario hasta la generación de reportes y cierre de sesión. Este proceso comienza con el ingreso de un usuario a la URL del sistema y la visualización de la pantalla de inicio de sesión. Luego, el usuario hace el ingreso de sus credenciales y el sistema empieza a validar la información.

Si las credenciales ingresadas son correctas, el sistema crea un token de autenticación y redirige al usuario al panel principal, donde se encuentran diferentes opciones de consulta. El usuario puede ver las estadísticas por conductor o por ruta, por lo que debe seleccionar ciertos parámetros, como el rango de fechas. El sistema valida los parámetros y consulta los eventos de somnolencia que se encuentran en la base de datos, mostrando los resultados de manera gráfica y tabular.

Además, el usuario debe acceder a la opción Generación de Reportes, donde debe establecer la configuración que desee del tipo de reporte, formato de salida y criterios de filtrado. El sistema, al recibir la información, genera el archivo correspondiente y el usuario podrá descargarlo. Por último, el flujo termina con la acción de cerrar sesión, donde se destruye el token de autenticación y se redirige al usuario a la pantalla de inicio.

**Figura 31**

*Diagrama de actividad, Sitio Web (Gestión y Reportes).*




## Vista Física

Para la vista física del sistema, se representa en un diagrama de despliegue que muestra la distribución real del software sobre los nodos físicos de hardware, así como los canales de comunicación existentes entre ellos. Se identificaron tres nodos principales: el dispositivo embebido, el servidor de aplicaciones web y el sistema de base de datos.

1. Dispositivo Embebido, representado por una Raspberry Pi 4, instalada físicamente en cada vehículo de la empresa. Este dispositivo tiene un sistema que le ayuda a determinar si el conductor muestra signos de somnolencia. Para esto, el sistema tiene una cámara que se conecta directamente a la Raspberry Pi. Durante el proceso en el que el conductor se encuentra manejando, el sistema no requiere de

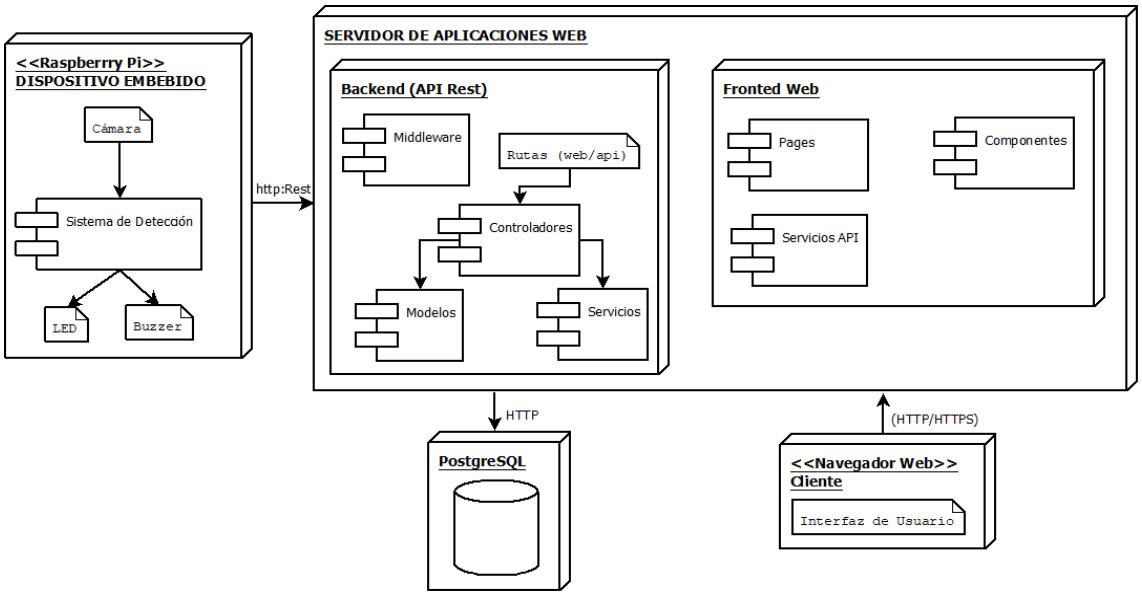
internet para determinar si el conductor presenta signos de somnolencia. En el caso de que el sistema detecte signos de somnolencia, el Raspberry Pi activa luces LED o emite un sonido. La Raspberry Pi envía información al servidor mediante protocolos de aplicación de la capa de comunicación del modelo de referencia TCP/IP a través de un módem GSM 4G. De esta forma, la Raspberry Pi envía alertas al servidor cuando detecta que el conductor se encuentra con signos de somnolencia. En caso de que ocurra algún problema en la conectividad de internet, el sistema se encuentra programado para seguir registrando datos de forma autónoma en la computadora Raspberry Pi. Cuando el sistema restablezca conexión a internet enviará todos los registros almacenados al servidor.

2. Servidor de Aplicaciones Web, el cual aloja tanto el backend como el frontend del sistema. En este entorno se despliega el Backend (API REST), responsable de recibir las solicitudes provenientes de los dispositivos embebidos, procesar la información y gestionar la lógica de negocio. Este mismo servidor se encuentra el Frontend, que proporciona la interfaz gráfica para los usuarios administrativos. La coexistencia de ambos componentes en un mismo nodo físico facilita la integración y reduce la latencia en la comunicación interna del sistema.

3. Base de Datos MySQL encargado de la tarea de almacenamiento. En este lugar reposan los datos de los eventos de somnolencia, datos de los que operan, rutas, usuarios y reportes. El backend se conecta a la base de datos a través de conexiones que aseguran la integridad de los datos, de que puedan estar disponibles cuando sea necesario, sin errores y en su totalidad.

**Figura 32**

*Diagrama de despliegue (física).*



## 5.1 Codificación

Para esta fase de codificación del proyecto, la arquitectura del sistema se dividió en tres capas tecnológicas interrelacionadas que conforman el Sistema de Detección de Somnolencia para Conductores: una backend desarrollada en Laravel, una frontend en React y un sistema embebido que opera en una Raspberry Pi 4. Las tres capas están integradas a través de una API REST y se utiliza PostgreSQL como el motor de base de datos principal, mientras que el sistema embebido tiene una base de datos SQLite local para operación fuera de línea.

El sistema permite detectar en tiempo real eventos de somnolencia (ojos cerrados, bostezos y cabezadas) utilizando visión por computadora, que son registrados, sincronizados con el backend y mostrados en un panel web accesible para administradores y operadores.

**Backend Laravel**

El backend está construido usando el framework Laravel 12, el cual utiliza el modelo arquitectónico MVC (Modelo-Vista-Controlador) y está adaptado a una arquitectura de tipo API REST. La comunicación se realiza en formato JSON y está asegurada a través de tokens Bearer generados y administrados por Laravel Sanctum. La base de datos principal es PostgreSQL, el cual se encuentra configurado para optimizar la conexión usando la opción PDO:PGSQL_ATTR_DISABLE_PREPARES para mayor compatibilidad.

**CONTROLADORES — /app/Http/Controllers/Api**

Los controladores son la capa de entrada de la API REST. Cada uno de ellos encapsula de manera exclusiva la lógica HTTP: la validación de entrada y parámetros, la delegación a modelos y servicios, y la construcción de respuestas JSON estandarizadas. A continuación, se describe cada controlador implementado en el proyecto:

- **AuthController.php — Autenticación**

Gestiona el ciclo completo de autenticación de usuarios en el sistema. Implementando tres operaciones fundamentales: inicio de sesión con validación de credenciales, estado de la cuenta, cierre de sesión con invalidación del token activo de Sanctum, y recuperación del perfil del usuario autenticado. La verificación de la contraseña utiliza

Hash:check() en el campo bcrypt almacenado y devuelve un token Bearer para solicitudes posteriores.

- **ConductorController.php — Gestión de Conductores**

Gestiona todo el ciclo de vida de los registros de conductores. Incorpora lógica empresarial avanzada para la sincronización de estados: Cuando un conductor se vuelve Inactivo (0), el sistema desvincula automáticamente su vehículo asignado y cancela las asignaciones activas. El estado de Ausente (2) suspende temporalmente la asignación sin cancelarla. La búsqueda implementa coincidencias insensibles a mayúsculas (LOWER + LIKE) en nombres, apellidos, ID y placas de los vehículos vinculados. La validación de la cédula ecuatoriana se delega a la clase ValidarCedula a través del algoritmo de módulo 10.

- **VehiculoController.php — Gestión de Vehículos**

Control para el registro y mantenimiento de los vehículos del sistema. La lógica de desactivación es de gran importancia: cuando un vehículo se marca como inactivo, el controlador desvincula automáticamente a todos los conductores que le habían sido asignados y cancela sus asignaciones activas (estado = 0). El endpoint selectList solo devuelve vehículos activos para su uso en formularios de selección. Se exige mayúsculas en el campo de la placa para la consistencia de los datos.

- **RutaController.php — Gestión de Rutas**

Gestiona las rutas de transporte de la cooperativa. Incluye la función para calcular la distancia en tiempo real, realizando dos llamadas sucesivas: primero geo codifica el origen y el destino a través de la API pública de Nominatim (OpenStreetMap, limitando a Ecuador), y luego consulta OSRM para obtener la distancia y el tiempo de conducción a lo largo de la ruta real. Ambas APIs son gratuitas y no requieren una clave de API. Los campos de origen y destino siempre se almacenan en mayúsculas.

- **AsignacionController.php — Gestión de Asignaciones**

Gestiona asignaciones de Conductor-Vehículo-Ruta (CVR). Implementar filtrado multiestado: en_ruta (estado igual a 1), pendiente (estado igual a 2) y terminado (estado igual a 0). Excluir terminado por defecto para que las vistas operativas no se vean saturadas. Verificar violaciones de unicidad (un conductor solo puede tener una

asignación activa) capturando el código de error de PostgreSQL 23505 en QueryException.

- **EventoSomnolenciaController.php — Eventos de Somnolencia**

Recibe los eventos generados por los dispositivos Raspberry Pi y los relaciona automáticamente con la asignación activa correspondiente. El endpoint /api/device/evento es público (sin autenticación) para permitir el envío desde dispositivos embebidos sin gestión de tokens. La lógica de vinculación busca la asignación cuyo rango de fechas (fecha_inicio <= fecha_evento <= fecha_fin) corresponde al conductor enviado, resolviendo así la identificación dinámica sin requerir que el dispositivo conozca el id_asignacion.

- **DashboardController.php — Panel de Control**

Proporciona todos los datos estadísticos y operativos del panel principal. El método stats() agrupa eventos por tipo utilizando conteo en la base de datos. El método devices muestra los dispositivos activos en la ruta con su último evento registrado, utilizando carbon para el formato de fecha. El método chartData admite filtros combinados de conductor, ruta y rango de fechas, devolviendo cuatro conjuntos de datos: distribución por tipo (para gráfico de dona), eventos por conductor (barras agrupadas), eventos por ruta y evolución temporal por fecha (gráfico de área/línea).

- **ReporteController.php — Generación de Reportes PDF**

Genera informes en formato PDF utilizando la biblioteca Barryvdh y DomPDF integradas con Laravel. El reporte se divide en dos tipos: General (sin filtros) o Específico (con al menos un filtro aplicado). El reporte tiene información completa del conductor y su vehículo. El reporte tiene un límite de 500 eventos por consulta para controlar el tamaño del PDF generado. La respuesta utiliza stream() para permitir la visualización directa en el navegador o la descarga.

- **UsuarioController.php — Administración de Usuarios**

Gestiona usuarios del sistema con roles de ADMIN y OPERATOR. El almacenamiento de contraseñas utiliza bcrypt a través de Hash::make(), y el modelo User define la contraseña como hash para la cifrado automático. La actualización permite cambios selectivos en el nombre de usuario, la contraseña, el rol y el estado. No permite registros

de usuario sin autenticación previa: todos los puntos finales de este controlador están asegurados por el middleware auth:sanctum.

**MODELOS ELOQUENT — /app/Models**

Los modelos Eloquent encapsulan la representación de cada tabla de la base de datos, definiendo relaciones entre entidades, conversión automática de tipos (casts), atributos asignables en masa (fillable) y configuración de marcas de tiempo personalizadas. La aplicación desactiva las marcas de tiempo automáticas de Laravel (updated_at) donde la base de datos no las requiere, utilizando en su lugar campos como fecha_registro o fecha_creacion.

- **AsignacionCVR — Asignación Conductor-Vehículo-Ruta**

**Tabla 13**

*Propiedades del modelo Eloquent AsignacionCVR.*

| Campo / Atributo | Tipo | Descripción |
|---|---|---|
| **id_asignacion** | PK autoincrement | Clave primaria de la asignación. |
| **id_conductor** | FK → conductores | Referencia al conductor asignado. |
| **id_ruta** | FK → rutas | Referencia a la ruta asignada. |
| **fecha_inicio** | date (Carbon) | Fecha de inicio de la asignación; cast automático a Carbon date. |
| **fecha_fin** | date \| null | Fecha de fin; null indica asignación indefinida vigente. |
| **estado** | integer (0/1/2) | 0=Terminada, 1=En Ruta, 2=Pendiente. Migrado de boolean a smallint. |
| **conductor()** | belongsTo | Relación al modelo Conductor. |
| **ruta()** | belongsTo | Relación al modelo Ruta. |
| **eventos()** | hasMany | Colección de EventoSomnolencia vinculados a esta asignación. |

- **Conductor — Conductores del Sistema**

**Tabla 14**

*Propiedades del modelo Eloquent Conductor.*

| Campo / Atributo | Tipo | Descripción |
|---|---|---|
| **id_conductor** | PK autoincrement | Clave primaria del conductor. |

| | | |
|---|---|---|
| **nombres** | string(50) | Nombres del conductor. |
| **apellidos** | string(50) | Apellidos del conductor. |
| **foto** | string \| null | Ruta relativa al archivo de foto en storage/public/drivers. |
| **tipo_licencia** | string(10) | Tipo de licencia (B, C, D, E según reglamento ecuatoriano). |
| **cedula** | string(10) unique | Cédula ecuatoriana validada con algoritmo de módulo 10. |
| **email_conductor** | string \| null | Correo electrónico del conductor (único si se proporciona). |
| **telefono** | string(20) \| null | Número de contacto del conductor. |
| **id_vehiculo** | FK → vehículos | Vehículo asignado actualmente al conductor (nullable). |
| **estado** | integer (0/1/2) | 0=Inactivo, 1=Activo, 2=Ausente. Migrado de boolean a smallint. |
| **fecha_registro** | timestamp | Timestamp de creación; usado como CREATED_AT personalizado. |
| **vehículo()** | belongsTo | Relación al modelo Vehículo. |
| **asignaciones()** | hasMany | Historial de asignaciones CVR del conductor. |

- **EventoSomnolencia — Eventos de Somnolencia Detectados**

**Tabla 15**

*Propiedades del modelo Eloquent EventoSomnolencia.*

| Campo / Atributo | Tipo | Descripción |
|---|---|---|
| **id_evento** | PK autoincrement | Clave primaria del evento. |
| **id_asignacion** | FK → asignacion_c_v_r | Asignación activa al momento del evento (vinculada automáticamente). |
| **fecha_evento** | date (Carbon) | Fecha en que ocurrió el evento; cast a Carbon date. |
| **hora_evento** | time | Hora exacta del evento en formato HH:MM:SS. |
| **tipo_evento** | string(50) | Tipo: OJOS_CERRADOS, BOSTEZO o CABECEO. |
| **duracion_seg** | decimal(8,2) | Duración del evento en segundos con precisión decimal. |
| **ear_prom** | decimal(8,2) | Promedio del Eye Aspect Ratio durante el evento. |
| **mar_prom** | decimal(8,2) | Promedio del Mouth Aspect Ratio durante el evento. |
| **timestamp** | datetime \| null | Timestamp UTC del dispositivo al momento del envío. |

| | | |
|---|---|---|
| **asignación()** | belongsTo | Relación a la AsignacionCVR asociada. |

- **Ruta — Rutas de Transporte**

**Tabla 16**

*Propiedades del modelo Eloquent Ruta.*

| Campo / Atributo | Tipo | Descripción |
|---|---|---|
| **id_ruta** | PK autoincrement | Clave primaria de la ruta. |
| **nombre** | string(25) unique | Nombre corto único de la ruta (normalizado). |
| **origen** | string(25) | Ciudad/punto de origen en mayúsculas. |
| **destino** | string(25) | Ciudad/punto de destino en mayúsculas. |
| **distancia_km** | decimal(6,2) | Distancia calculada automáticamente vía OSRM. |
| **estado** | boolean | true=Activa, false=Inactiva. |
| **fecha_creacion** | timestamp | Timestamp de creación; CREATED_AT personalizado. |
| **asignaciones()** | hasMany | Asignaciones CVR asociadas a esta ruta. |

- **Vehiculo — Vehículos del Sistema**

**Tabla 17**

*Propiedades del modelo Eloquent Vehículo.*

| Campo / Atributo | Tipo | Descripción |
|---|---|---|
| **id_vehiculo** | PK autoincrement | Clave primaria del vehículo. |
| **placa** | string(10) unique | Placa del vehículo (almacenada en mayúsculas). |
| **marca** | string(50) \| null | Marca del vehículo. |
| **modelo** | string(50) \| null | Modelo del vehículo. |
| **anio** | integer \| null | Año de fabricación del vehículo. |
| **tipo** | string(20) | Tipo de vehículo (taxi, bus, camioneta, etc.). |
| **estado** | boolean | true=Activo (operativo), false=Inactivo. |
| **fecha_registro** | timestamp | Timestamp de registro del vehículo. |
| **conductor()** | hasOne | Conductor que tiene actualmente este vehículo. |
| **asignaciones()** | hasMany | Historial de asignaciones del vehículo. |

- **User — Usuarios del Sistema**

**Tabla 18**

*Propiedades del modelo Eloquent User.*

| Campo / Atributo | Tipo | Descripción |
|---|---|---|
| **id_usuario** | PK autoincrement | Clave primaria del usuario. |
| **username** | string(50) unique | Nombre de usuario para autenticación. |
| **password** | string (bcrypt) | Contraseña hasheada con bcrypt; cast 'hashed' aplica encriptación automática. |
| **rol** | string (enum) | Rol del usuario: ADMIN u OPERADOR (validado en controlador). |
| **estado** | boolean | true=Activo (puede autenticarse), false=Bloqueado. |
| **fecha_creacion** | timestamp | Timestamp de creación del usuario; CREATED_AT personalizado. |

**MIGRACIONES DE BASE DE DATOS — /database/migrations**

Las migraciones documentan de manera versionada la evolución del esquema de la base de datos PostgreSQL. El sistema tiene migraciones base que definen la estructura inicial y migraciones incrementales que adaptan el esquema a los requisitos del sistema de detección de somnolencia. Destaca el uso de sentencias SQL nativas (DB::statement) para operaciones complejas de PostgreSQL no soportadas directamente por el Schema Builder de Laravel.

**Tabla 19**

*Migraciones de base de datos PostgreSQL*

| Archivo de Migración | Operación Realizada |
|---|---|
| create_usuarios_table | Crea tabla usuarios con id, username, password, rol, estado y fecha_creacion. También crea password_reset_tokens y sessions. |
| create_conductores_table | Crea tabla conductores con datos personales, tipo_licencia, cedula única y estado booleano inicial. |
| create_vehiculos_table | Crea tabla vehículos con placa única, marca, modelo, año y estado booleano. |
| create_rutas_table | Crea tabla rutas con nombre único (25 chars), origen, destino, distancia y estado booleano. |
| create_asignacion_c_v_r_table | Crea tabla de asignaciones con FK a conductores, vehículos y rutas. Estado booleano inicial. |

| | |
|---|---|
| create_eventos_somnolencia_table | Crea tabla de eventos con FK a asignaciones. Estructura básica inicial con tipo y duracion_segundos. |
| add_contact_info_to_conductores | Agrega email_conductor (único) y telefono a la tabla conductores. |
| move_vehiculo_id_to_conductores | Traslada id_vehiculo de asignacion_c_v_r a conductores (vinculación 1:1 conductor-vehículo). |
| change_estado_to_integer_asignacion | Migra estado en asignaciones de boolean a smallint (0/1/2) usando ALTER COLUMN con USING cast de PostgreSQL. |
| add_tipo_to_vehiculos | Agrega columna tipo a vehículos (taxi, bus, camioneta, etc.). |
| add_foto_to_conductores | Agrega columna foto (ruta al archivo) a conductores. |
| modify_eventos_somnolencia | Reemplaza tipo/duracion_segundos/fecha_registro por fecha_evento, hora_evento, tipo_evento, duracion_seg, ear_prom, mar_prom, timestamp. |
| change_duracion_seg_to_decimal | Cambia duracion_seg de integer a decimal(8,2) para soportar fracciones de segundo. |
| change_conductor_estado_to_integer | Migra estado en conductores de boolean a smallint (0/1/2) para soportar el estado Ausente. |

**SEEDER DE DATOS INICIALES — /database/seeders**

DatabaseSeeder proporciona un usuario administrador inicial del sistema a través del método updateOrCreate, proporcionando idempotencia en la ejecución del sembrador. El usuario administrador se crea con el nombre de usuario taxilagos@gmail.com, rol ADMIN y estado activo. La contraseña se almacena como un hash bcrypt activo a través de Hash::make() y nunca en texto plano. Este sembrador se ejecuta automáticamente cuando se ejecuta el comando php artisan db:seed durante el despliegue inicial del sistema.

**REGLAS DE VALIDACIÓN — /app/Rules**

Una regla de validación a medida se implementó para la cédula de ciudadanía (ValidarCedula.php) que implementa la interfaz ValidationRule de Laravel. El algoritmo valida tres condiciones: que el campo contenga exactamente 10 caracteres numéricos, que el tercer dígito sea menor que 7 (validación de provincia), y que la suma ponderada, con coeficientes alternados [2,1,2,1,2,1,2,1,2] (reduciendo a 9 los productos ≥ 10) de los primeros 9 dígitos, sea congruente con el dígito verificador final, de acuerdo con el módulo 10.

**CONFIGURACIÓN — /config**

La configuración del sistema se distribuyó en los archivos estándar de Laravel con las siguientes adaptaciones específicas al proyecto:

- **database.php:** La conexión por defecto se configuró a pgsql con la opción PDO::PGSQL_ATTR_DISABLE_PREPARES para mejorar la compatibilidad con el driver de PostgreSQL.
- **cors.php:** Orígenes permitidos para desarrollo local (localhost:5173) y producción (frontend-somnolencia.onrender.com). supports_credentials desactivado para que funcione la autenticación por token.
- **app.php:** Se estableció la zona horaria UTC para que coincida con el servidor y los dispositivos embebidos que registran los eventos con timestamp UTC.
- **sanctum.php:** Dominios stateful configurados para el desarrollo; los tokens de las sesiones de administrador se dejan sin plazo.

**RUTAS DE LA API — /routes/api.php**

Las rutas están organizadas en dos grupos: rutas públicas para autenticación y dispositivos embebidos, y rutas protegidas por el middleware auth:sanctum para el panel de control. La ruta /api/device/evento es intencionalmente pública para permitir que los dispositivos Raspberry Pi envíen datos sin gestionar tokens Bearer.

**Frontend React**

El frontend está desarrollado como una Single Page Application (SPA) usando React 19, Vite como empaquetador y Tailwind CSS para el diseño. La comunicación con el backend se hace únicamente a través de HTTP a la API REST usando Axios, con gestión centralizada de los headers de autenticación Bearer. La aplicación implementa navegación protegida con React Router, enviando a los usuarios no autenticados a la pantalla de inicio de sesión.

**COMPONENTES COMPARTIDOS — /src/components**

Los componentes de esta carpeta son reutilizables y dan apoyo a varios módulos de la aplicación. Están pensados para encapsular lógica de interfaz genérica sin acoplarse a un módulo de negocio.

**Tabla 20**

*Componentes compartidos del frontend React.*

| Módulo / Archivo | Carpeta | Responsabilidad |
|---|---|---|
| **DashboardLayout.jsx** | /src/components | Layout principal con sidebar de navegación. Incluye menú colapsable en móvil, barra de header con perfil de usuario y notificaciones, y Outlet para renderizado de módulos hijos. Detecta la ruta activa para resaltar el ítem del menú correspondiente. |
| **ProtectedRoute.jsx** | /src/components | HOC de enrutamiento que verifica el estado de autenticación via useAuth(). Redirige a /login si el usuario no está autenticado; de lo contrario renderiza el Outlet del hijo. |
| **SearchableSelect.jsx** | /src/components | Componente de select personalizado con búsqueda en tiempo real, soporte para imágenes/avatares, múltiples variantes visuales (default, user-card, icon-card), limpieza de selección y cierre automático al hacer clic fuera del componente. |

**MÓDULOS DE FUNCIONALIDADES — /src/features**

Esta carpeta organiza la aplicación en módulos funcionales independientes, cada uno posiblemente alineado con uno del sistema. Cada carpeta de módulo contiene sus componentes, ganchos y lógica de negocio, favoreciendo el desarrollo incremental y el mantenimiento autónomo.

**Tabla 21**

*Módulos de funcionalidades del frontend React.*

| Módulo / Archivo | Responsabilidad |
|---|---|
| **auth** | Formulario de inicio de sesión, manejo del token Sanctum en contexto global (AuthContext), y función de logout con limpieza de estado. Control de acceso en función del rol del usuario autenticado. |
| **dashboard** | Panel principal con tarjetas de estadísticas (total eventos, ojos cerrados, bostezos, cabeceos), tabla de últimos eventos en tiempo real, lista de dispositivos activos y cuatro gráficos interactivos con filtros por conductor, ruta y rango de fechas. |

| | |
|---|---|
| **conductores** | CRUD completo de conductores con foto de perfil, validación de cédula, cambio de estado (activo/ausente/inactivo), búsqueda en tiempo real y paginación. Visualización de tarjetas de conductor con estado visual diferenciado. |
| **vehículos** | CRUD de vehículos con filtros por estado y asignación. Muestra estado de disponibilidad (activo/ausente/inactivo) calculado según el estado del conductor asignado. |
| **rutas** | CRUD de rutas con calculadora de distancia integrada que geo codifica origen y destino para obtener km y tiempo de manejo real vía OSRM. Activación/desactivación de rutas con cancelación automática de asignaciones. |
| **asignación** | Gestión de asignaciones CVR de conductor, vehículo y ruta. Filtrado por estado de asignación y visualización del estado actual de cada asignación (en ruta, pendiente, completada, terminada). |
| **reportes** | Generación de reportes PDF con filtros opcionales de conductor, ruta y rango de fechas. Visualiza el reporte en una nueva pestaña del navegador vía stream del backend. |
| **configuración** | Administración de usuarios del sistema (CRUD). Gestión de roles (ADMIN/OPERADOR) y activación/desactivación de cuentas. |

**UTILIDADES Y CONFIGURACIÓN — /src/lib**

Contiene la configuración centralizada para el consumo de la API REST. Se define una instancia de Axios con la URL base del back, el interceptor de peticiones que añade el token Bearer de forma automática desde el contexto de autenticación y el interceptor de respuestas que maneja los errores 401 (sesión caducada) redirigiendo al login. Las funciones utilitarias de formato de fechas y moneda también residen en esta carpeta para uso transversal en toda la aplicación.

**PÁGINAS DEL FRONTEND REACT — /src/pages**

El directorio src/pages/ contiene componentes de páginas de alto nivel para el frontend. Cada página implementa una vista completa de la aplicación con gestión de estado local, llamadas a la API, componentes UI reutilizables y lógica de negocio del lado del cliente. A continuación, se presenta la documentación para cada página con sus responsabilidades, hooks utilizados y flujos de interacción.

- **Login.jsx — Autenticación de Usuario**

Las páginas públicas son accesibles en la ruta raíz /. Gestiona inicios de sesión a través de correo electrónico y contraseña. Consume la autenticación global de AuthContext para persistir el token de Sanctum en sessionStorage y redirige automáticamente al tablero en caso de autenticación exitosa.

- **Dashboard.jsx — Panel de Control Principal**

Página central del sistema. Consolida las estadísticas de somnolencia, lista de dispositivos activos, tabla de últimos eventos y cuatro gráficos interactivos con filtros combinados. Los datos se actualizan automáticamente cada 10 segundos mediante sondeo.

- **Conductores.jsx — Gestión de Personal**

Página de CRUD completo para el módulo de conductores. Gestiona la creación, modificación, paginación, búsqueda y cambio de estado tri-estado (activo/ausente/inactivo) de los conductores. Incluye carga de fotos mediante arrastrar y soltar y validación del lado del cliente para la cédula ecuatoriana.

- **Vehículos.jsx — Control de Flota**

Página de CRUD para la gestión de flotas. Implementa filtrado doble (por estado y por asignación), formateo automático de placas en el patrón ABC-1234, tipo de selector de vehículo personalizado y modal de confirmación antes de desactivar una unidad.

- **Rutas.jsx — Gestión de Trayectos**

Página de CRUD para rutas operativas. La funcionalidad diferencial de esta página es el cálculo automático de la distancia en carretera: cuando el usuario llena los campos de Origen y Destino, el sistema llama al endpoint /rutas/calcular-distancia del backend (que consume la API de OSRM vía Nominatim), actualiza el campo distancia_km automáticamente y muestra el tiempo estimado de viaje.

- **Asignación.jsx — Gestión de Asignaciones CVR**

Página más compleja del frontend. Maneja asignaciones de CVR (Conductor-Vehículo-Ruta) con un sistema de estados de cuatro estados (en_ruta, completado, pendiente, terminado). Cada estado expone diferentes acciones a través de modales dedicados. La tabla ordena dinámicamente para priorizar los registros que necesitan atención.

- **Reportes.jsx — Generación de Reportes PDF**

Página para generar informes en PDF. Consume el endpoint /reports/generate con filtros opcionales. Los datos para el selector de conductor y ruta se obtienen reutilizando el endpoint /dashboard/chart-data. Implementa el mismo componente SearchableSelect del Dashboard para consistencia.

- **Sistema Embebido Raspberry Pi 4**

El sistema embebido es la primera capa del proyecto encargada de la detección en tiempo real. Se desarrolla en Python 3 y corre en un Raspberry Pi 4 conectado a una cámara Pi NoIR (sin filtro) para la operación nocturna. El programa principal (src/main.py) integra la captura de video, el procesamiento de landmarks. El cálculo de métricas de somnolencia, la gestión de alarma y la sincronización de datos con el backend.

## ALGORITMOS DE DETECCIÓN

El sistema implementa tres métricas que son complementarias para la detección robusta de somnolencia, cada una de ellas cubriendo un indicador fisiológico diferente:

- **Eye Aspect Ratio (EAR) — Detección de Ojos Cerrados**

El EAR calcula la relación entre las distancias verticales y horizontales del ojo. Se apoya en 6 landmarks de dlib (ópticos) por ojo (índices 36-41 para el ojo izquierdo, 42-47 para el ojo derecho). La fórmula combina dos pares verticales (A y B) y un par horizontal (C):

$$EAR = (||p2-p6|| + ||p3-p5||) / (2.0 \times ||p1-p4||)$$

El umbral UMBRAL_EAR = 0.21 fue determinado experimentalmente. Con los ojos completamente abiertos el EAR es típicamente 0.3-0.4; al cerrarse el EAR se cierra a valores cercanos a 0. El evento se confirma solo si el EAR se mantiene por debajo de ese umbral durante FRAMES_CONSECUTIVOS_EAR = 10 frames consecutivos, evitando falsas alarmas por parpadeos normales.

- **Mouth Aspect Ratio (MAR) — Detección de Bostezos**

El MAR se encarga del reconocimiento de aperturas de la boca utilizando 8 de los landmarks del loop interno de la boca, o sea, los índices del 60 al 67 de dlib. Mientras mayor sea el MAR, mayor es la apertura de la boca. En la fórmula se utilizan dos distancias verticales internas (numerador) y una horizontal (denominador):

$$MAR = (||p3-p7|| + ||p4-p6||) / (2.0 \times ||p1-p5||)$$

El umbral UMBRAL_MAR = 0.40 indica una considerable apertura de la boca. El evento se confirma después de FRAMES_CONSECUTIVOS_MAR = 5 frames con MAR por encima del umbral, pudiendo diferenciar un bostezo de una conversación normal.

- **Nose-to-Mouth Ratio (NM) — Detección de Cabeceo**

El NM se encarga de la detección de cabeceo, y esto se realiza a través de la medición de la posición de la nariz con relación al mentón, normalizado a la altura del rectángulo facial. En la medida que la cabeza se inclina hacia adelante (signo de somnolencia), la nariz se aproxime al mentón y el valor NM disminuye. Se toman los landmarks 30 (punta de la nariz) y 8 (mentón):

$$NM = ||nariz(30) - menton(8)|| / altura_cara$$

El umbral UMBRAL_NM = 0.35 fue disminuido de 0.60 en versiones anteriores a esta, para ser menos sensible y que no se generen falsas detecciones por un leve cabeceo que se pueda ejecutar durante la conducción normal.

**Código 1**

*Algoritmo de detección EAR, MAR, NM.*

```
def forma_a_array(shape):
    return np.array([(shape.part(i).x, shape.part(i).y) for i in range(68)])


def distancia(p1, p2):
    return np.linalg.norm(p1 - p2)


def calcular_EAR(ojo):
    A = distancia(ojo[1], ojo[5])
    B = distancia(ojo[2], ojo[4])
    C = distancia(ojo[0], ojo[3]) + 1e-6
    return (A + B) / (2.0 * C)
```

```
def calcular_MAR(boca):
    num = distancia(boca[2], boca[6]) + distancia(boca[3], boca[5])
    den = 2.0 * (distancia(boca[0], boca[4]) + 1e-6)
    return num / den
```

## PARÁMETROS DE CONFIGURACIÓN

Se decidió calibrar experimentalmente los parámetros de este sistema con el fin de balancear la sensibilidad y la especificidad. El sistema detecta de forma autónoma la plataforma en la que está ejecutándose (Raspberry Pi o PC) y la ajusta la resolución, el procesamiento, y el modo alarma.

**Tabla 22**

*Parámetros de configuración del sistema embebido.*

| Parámetro | Valor | Justificación Técnica |
|---|---|---|
| **UMBRAL_EAR** | 0.21 | Valor calibrado experimentalmente. EAR < 0.21 indica posible cierre de ojos; por encima corresponde a ojos abiertos. |
| **FRAMES_CONSECUTIVOS_EAR** | 10 frames | Equivale a ~1 segundo a 10 FPS en Raspberry Pi. Evita alarmas por parpadeos normales. |
| **UMBRAL_MAR** | 0.40 | Apertura de boca significativa. Valores menores corresponden a boca cerrada o conversación normal. |
| **FRAMES_CONSECUTIVOS_MAR** | 5 frames | Confirma bostezo sostenido; diferencia de hablar o abrir la boca momentáneamente. |
| **UMBRAL_NM** | 0.35 | Reducido de 0.60 para menor sensibilidad. Cabeceos pronunciados bajan el NM por debajo de este umbral. |
| **TIEMPO_COOLDOWN_DB** | 2 segundos | Intervalo mínimo entre registros en base de datos para evitar saturación de eventos idénticos. |

**Código 2**

*Codificación de la configuración global.*

```
# ----------------------------
```

```
# configuración global
# ----------------------------

# auto-detectar plataforma
is_raspberry_pi = platform.machine() in ['armv7l', 'aarch64', 'armv6l']
raspberry_mode = is_raspberry_pi

# configuración adaptativa según plataforma
if is_raspberry_pi:
    print("[platform] raspberry pi detectado")
    ancho_camara = 320   # resolución baja para mejor fps
    alto_camara = 240
    usar_histograma_eq = true   # ecualización para visión nocturna
    ajustes_camara_nocturna = true
else:
    print("[platform] pc detectado")
    ancho_camara = 640
    alto_camara = 480
    usar_histograma_eq = false
    ajustes_camara_nocturna = false

ruta_modelo = os.path.join("modelos", "shape_predictor_68_face_landmarks.dat")
ruta_db = os.path.join("registros", "somnolencia.db")

# --- umbrales recalibrados (v2) ---
# ojos (ear)
umbral_ear = 0.21
frames_consecutivos_ear = 10   # menos frames = detección más rápida
```

```
# bostezo (mar)
umbral_mar = 0.40
frames_consecutivos_mar = 5

# cabeceo (nm) - drásticamente bajado
# valor normal mirando al frente ~0.5 - 0.7
# valor cabeceando ~0.2 - 0.3
umbral_nm = 0.35              # bajado de 0.60 a 0.35 para ser mucho menos sensible

tiempo_cooldown_db = 2        # segundos entre registros en db (no evitar spam de audio, eso es continuo)

cam_index = 0

# --- configuración laravel api ---
enable_api_sync = false           # activar/desactivar sincronización con laravel (desactivado para mejor performance)
laravel_api_url = "http://localhost:8000/api/evento"  # url del endpoint laravel (singular)
api_timeout = 5                   # timeout en segundos para requests
max_reintentos = 3                # número máximo de reintentos si falla el envío

# --- configuración de performance ---
frame_skip = 3 if is_raspberry_pi else 2  # raspberry pi: más skip para mejor fps
```

**ARQUITECTURA DEL SISTEMA EMBEBIDO**

El sistema usa tres patrones técnicos clave para garantizar rendimiento en el hardware limitado del Raspberry Pi:

1. **Base de datos asíncronas**: Las escrituras de la base de datos SQLite, al igual que los requests HTTP hacia la API se corren en un ThreadPoolExecutor de 4 trabajadores (registrar_evento_async) de manera que no se congele el loop principal de detecciones durante operaciones de pois.

2. **Cache de detecciones por frame skip:** La función main() se encarga de guardar en el estado última_detección los resultados de detecciones para que se puedan reutilizar en los frames que no son procesados (frame_count % FRAME_SKIP != 0), esto permite que se mantenga la fluidez del video y se optimiza el rendimiento del procesador.

3. **Detección automática de la plataforma:** Este sistema al inicio verifica el platform.machine() para ver si se corre en Raspberry Pi, lo que permite (armv7l/aarch64) y se configura automáticamente la resolución de video, configuraciones de la cámara nocturna, configurable de modo de emergencia (GPIO buzzer vs audio del sistema) y skip frames.

**GESTIÓN DE ALARMA MULTIPLATAFORMA**

Se implementa en el sistema una doble estrategia de alarma por sistema operativo, para el caso de Raspberry Pi se ejecuta un buzzer activo controlado por el pin GPIO 17 mediante la librería gpio_buzzer; y para el caso de PC/Linux se usa la clase GestorAlarma que crea un hilo daemon que reproduce de manera continua un archivo beep.wav.

**Código 3**

*Codificación del gestor de alarma-multiplataforma.*

```
# ----------------------------
# gestor de alarma - multiplataforma
# ----------------------------


if is_raspberry_pi:
    # raspberry pi: usar gpio buzzer
    try:
        from gpio_buzzer import buzzeralarma
```

```
        alarma = buzzeralarma(pin=17, buzzer_type='active')
        print("[alarma] usando gpio buzzer (pin 17)")
    except importerror as e:
        print(f"[alarma] error importando gpio_buzzer: {e}")
        print("[alarma] usando fallback a audio")
        is_raspberry_pi = false  # forzar uso de audio
else:
    # windows/linux: usar audio (sistema actual)
    pass  # se define abajo
```

**INTEGRACIÓN DE LAS TRES CAPAS**

La arquitectura del sistema establece un flujo de datos unidireccional desde los dispositivos de detección hacia el panel de control, con la API REST como punto central de integración:

**Tabla 23**

*Flujo de integración de las tres capas del sistema.*

| Paso | Componente | Acción |
|---|---|---|
| **1** | **Raspberry Pi (main.py)** | Captura frame de video, detecta landmarks con dlib, calcula EAR/MAR/NM. Si supera umbral por N frames consecutivos, activa alarma GPIO y registra en SQLite local. |
| **2** | **Raspberry Pi → API** | Envía POST /api/device/evento con id_conductor, tipo_evento, duracion_seg, ear_prom, mar_prom, fecha y hora del evento. |
| **3** | **API (EventoSomnolenciaController)** | Recibe el evento, busca la AsignacionCVR activa del conductor en la fecha indicada y crea el registro en eventos_somnolencia vinculada al id_asignacion correspondiente. |
| **4** | **PostgreSQL** | Almacena el evento con todas sus relaciones. La clave foránea id_asignacion conecta el evento con conductor, vehículo y ruta. |
| **5** | **Frontend (Dashboard)** | Consulta periódicamente /api/dashboard/latest-events, /stats y /devices. Actualiza tarjetas, tabla de eventos recientes y gráficos sin recargar la página. |

| | | |
|---|---|---|
| **6** | **Frontend (Reportes)** | El administrador aplica filtros y solicita GET /api/reports/generate. El backend construye el PDF con DomPDF y lo retorna como stream para visualización o descarga inmediata. |

## 6.1 Pruebas

### Propósito

Se realizaron pruebas de factibilidad en el sistema embebido con 10 sujetos de prueba, en 2 grupos con perfiles distintos, 3 de ellos, conductores reales de la empresa Taxi Lagos Courier S.A, y 7 de ellos participaron personas externas. Cada grupo se encuadró en un rol determinado. Con esta estrategia se buscó captar la evaluación del sistema en función de la técnica y las condiciones reales de operación, así como la respuesta del sistema nervioso, frente a la estimulación de somnolencia en función de distintos escenarios de iluminación y de los horarios. Las pruebas se ejecutaron en 2 niveles de iluminación controlados, el diurno (12:00 a 15:00), para ver síntomas de post almuerzo, y el nocturno (19:30 a 01:30), para ver síntomas en el horario laboral nocturno de los conductores. Este patrón se ajusta al comportamiento de somnolencia en la operación diaria de Taxi Lagos Courier S.A.

### 6.1.1 Diseño y Desarrollo de Casos de Prueba

**Tabla 24**

*Diseño y desarrollo de casos de prueba (Stakeholders).*

| Involucrado | Escenario | Rol y responsabilidad en las pruebas |
|---|---|---|
| Equipo de Evaluación (Desarrolladores) | Página Web | Ejecuta los casos de prueba técnicos (unitarias, rendimiento, seguridad, integración). Verifica el correcto funcionamiento de la API REST, los modelos Eloquent, los algoritmos EAR/MAR/NM del embebido y la sincronización de datos. |
| Administrador del Sistema (Taxi Lagos) | Oficina Ibarra-Quito-Cayambe-Otavalo (Pagina Web) | Valida el panel de control web: gestión de conductores, vehículos, rutas, asignaciones CVR, generación de reportes PDF y visualización del dashboard estadístico con gráficos en tiempo real.<br>Verifica los flujos de asignación Conductor-Vehículo-Ruta, cambio de estados triestado (Activo/Ausente/Inactivo) y visualización de eventos de somnolencia en tiempo real. |

| | | |
|---|---|---|
| Conductores de Taxi Lagos | Vehículo (Diurno y Nocturno) | Participan en las pruebas de factibilidad con el sistema embebido instalado en el vehículo. Evalúan la pertinencia de las alarmas, la comodidad del prototipo y la respuesta de detección en condiciones reales de conducción. |

### 6.1.2 Registro de Riesgos de Prueba

**Tabla 25**

*Prueba de riesgo.*

| N° | Riesgo | Prob. (1-5) | Impacto (1-5) | Severidad | Plan de Mitigación |
|---|---|---|---|---|---|
| 1 | Latencia de red entre Raspberry Pi y backend Laravel en zonas con baja cobertura 4G | 3 | 5 | 15 | Buffer SQLite local en el embebido; sincronización diferida cuando se restaura la conectividad (ENABLE_API_SYNC). |
| 2 | Falsos positivos en detección EAR/MAR por variaciones de iluminación nocturna | 4 | 4 | 16 | Cámara Pi NoIR con ecualización de histograma activa (USAR_HISTOGRAMA_EQ=true); validación por frames consecutivos (EAR: 10 frames, MAR: 5 frames). |
| 3 | Conflicto por asignación duplicada conductor-vehículo-ruta en la Base de Datos | 2 | 5 | 10 | Error QueryException y muestra respuesta HTTP 409 en el frontend. |
| 4 | Fallo del servidor de producción durante sincronización de eventos desde la Raspberry Pi. | 2 | 5 | 10 | Raspberry Pi guarda eventos de manera local en SQLite mientras se restablece conexión con el servidor. |

### 6.1.3 Herramientas de Prueba

**Tabla 26**

*Herramientas de Prueba*

| Herramienta | Función |
|---|---|
| Python + OpenCV | Detección facial |

| | |
|---|---|
| Dlib | Predicción de landmarks |
| Raspberry Pi 4 | Ejecución embebida |
| Laravel | API REST |
| React | Visualización |
| PostgreSQL | Persistencia |
| SQLite | Almacenamiento local |
| Postman | Pruebas de API |

### 6.1.4 Pruebas de Unidad

**Tabla 27**

*Pruebas unitarias del sistema embebido - detección de somnolencia.*

| **Componente** | **Ítem a evaluar** | **Cumplimiento** | |
|---|---|---|---|
| | | **SI** | **NO** |
| MetricCalculator | Verificar cálculo correcto de EAR | X | |
| MetricCalculator | Verificar cálculo correcto de MAR | X | |
| MainDetector | Inicialización correcta de cámara | X | |
| GestorAlarma | Activación del buzzer | X | |

**Tabla 28**

*Pruebas unitarias de la plataforma web - autenticación y gestión.*

| **Componente** | **Ítem a evaluar** | **Cumplimiento** | |
|---|---|---|---|
| | | **SI** | **NO** |
| Login | Verificar que el usuario exista | X | |
| Login | Validación de contraseña encriptada | X | |

| | | | |
|---|---|---|---|
| Usuarios | Registro de conductores | X | |
| Eventos | Registro de eventos de somnolencia | X | |

### 6.1.5 Prueba de Funcionalidad

Verificar el correcto funcionamiento del sistema embebido y la plataforma web.

**Tabla 29**

*Prueba de funcionalidad del sistema embebido y la plataforma web.*

| **Aspecto evaluado** | **Cumplimiento** | |
|---|---|---|
| | **SI** | **NO** |
| ¿El sistema detecta cierre prolongado de ojos? | X | |
| ¿El sistema detecta bostezos? | X | |
| ¿El buzzer se activa correctamente? | X | |
| ¿El sistema registra eventos en la base de datos? | X | |
| ¿El administrador visualiza los eventos? | X | |
| ¿El sistema permite gestionar conductores? | X | |

**Tabla 30**

*Prueba de disponibilidad del sistema.*

| **Aspecto evaluado** | **Cumplimiento** | |
|---|---|---|
| | **SI** | **NO** |
| ¿El sistema web está disponible 24/7? | X | |
| ¿El sistema maneja correctamente mantenimientos? | X | |
| ¿Las sesiones se gestionan correctamente? | X | |

### 6.1.6 Prueba de Rendimiento

**Tabla 31**

*Prueba de rendimiento del desempeño del sistema.*

| **Aspecto evaluado** | **Cumplimiento** |
|---|---|

| | SI | NO |
|---|---|---|
| ¿La Raspberry Pi procesa en tiempo real (~20 FPS)? | X | |
| ¿El backend maneja múltiples peticiones? | X | |
| ¿El sistema responde sin degradación bajo carga? | X | |

### 6.1.7 Prueba de Seguridad

**Tabla 32**

*Prueba de seguridad validación de protección del sistema.*

| Aspecto evaluado | Cumplimiento | |
|---|---|---|
| | SI | NO |
| ¿El sistema gestiona correctamente las sesiones? | X | |
| ¿El sistema es resistente a inyección SQL? | X | |
| ¿Las contraseñas están encriptadas? | X | |
| ¿Los datos viajan mediante HTTPS? | | X |

### 6.1.8 Pruebas de Usabilidad

La usabilidad de los paneles de control web dirigidos a los administradores y operadoras de Taxi Lagos consiste en medir cuán sencillo es el uso de estos. Para esto se midió la claridad de la interfaz, la constancia visual de la paleta corporativa, el nivel de azul marino #001F3F / amarillo #FFD700, la facilidad de acceso a los módulos y la velocidad y claridad de respuesta del sistema a los actos del usuario.

**Tabla 33**

*Pruebas de usabilidad de la plataforma web.*

| Aspecto Evaluado | Cumplimiento | |
|---|---|---|
| | SI | NO |
| ¿El sistema incorpora íconos adecuados para cada funcionalidad? | X | |
| ¿La información está ubicada correctamente para fácil acceso? | X | |
| ¿El sistema muestra la sección activa en el encabezado? | X | |
| ¿Los administradores acceden fácilmente a la gestión de usuarios? | X | |

| | | |
|---|---|---|
| ¿El menú de acciones se adapta según el rol del usuario? | X | |
| ¿El diseño de la interfaz es adecuado para los involucrados? | X | |
| ¿El sistema presenta retroalimentación visual inmediata al usuario? | X | |
| ¿El sistema responde dentro de un período adaptable para el usuario? | X | |
| ¿Los formularios de asignación guían al operador con validaciones claras? | X | |
| ¿La búsqueda de conductores/vehículos opera en tiempo real? | X | |
| ¿El dashboard estadístico es interpretable sin capacitación técnica? | X | |
| ¿El sistema ofrece confirmación antes de acciones destructivas? | X | |

### 6.1.9 Resultados por Escenario de Prueba

**Tabla 34**

*Pruebas en escenario diurno (12:00 - 15:00) - Post-Almuerzo.*

| N° | Participante | Tipo de evento detectado | Hora de detección | Tiempo respuesta (seg) | Alarma percibida | Observación |
|---|---|---|---|---|---|---|
| 1 | Conductor 1 (ruta urbana Quito) | Bostezo (MAR=0.58) | 13:42 | 0.8s | Sí | Post-almuerzo. Bostezo durante parada en semáforo. Tráfico urbano. |
| 2 | Conductor 1 | Bostezo (MAR=0.61) | 14:10 | 0.9s | Sí | Segundo evento. Conductor reportó somnolencia leve. |
| 3 | Conductor 2 (ruta urbana Quito) | Bostezo (MAR=0.53) | 13:55 | 0.7s | Sí | Primer bostezo del día. Iluminación natural óptima. |
| 4 | Conductor 3 (ruta urbana Quito) | Bostezo (MAR=0.47) | 14:22 | 0.8s | Sí | Evento leve. Conductor activo, sin |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | somnolencia profunda. |

**Análisis:** De los 3 conductores evaluados en el escenario post-almuerzo (12:00 – 15:00), los 3 presentaron al menos un evento detectable, totalizado 4 eventos registrados. El tipo de evento que predomino fue el bostezo en el 100% de los casos (4/4), lo cual es consistente con la somnolencia característica del periodo post-alumuerzo. El valor MAR promedio fue 0.55 (oscilando entre 0.47 y 0.61), indicando niveles moderados de apertura de la boca. El tiempo de respuesta de promedio fue de 0.825 segundos (calculado entre 0.7s y 0.9s), reflejando una detección ágil basada en evento. Cabe destacar que el conductor 1 fue el único en reportar somnolencia leve de forma subjetiva, coincidiendo con sus dos eventos detectados y los valores MAR más altos del grupo (0.58 y 0.61), lo que sugiere una correlación entre la frecuencia de bostezos, la magnitud del MAR y nivel de cansancio presente

**Tabla 35**

Pruebas de escenario nocturno (19:30 - 01:30) - Jornada Laboral Extendida.

| N° | Participante | Tipo de evento detectado | Hora | Tiempo respuesta (seg) | Alarma percibida | Observación |
|---|---|---|---|---|---|---|
| 1 | Conductor 1 (Quito–Cayambe) | Bostezo (MAR=0.66) | 20:30 | 0.9s | Sí | Inicio de ruta. Bostezo frecuente. Primer evento de jornada nocturna. |
| 2 | Conductor 1 | Cierre ojos (EAR=0.16) | 20:58 | 1.2s | Sí | Somnolencia progresiva. EAR sostenido bajo por 10 frames. Alarma activó respuesta inmediata. |
| 3 | Conductor 1 | Bostezo + cierre ojos | 21:15 | 1.1s | Sí | Evento combinado MAR+EAR. Mayor fatiga en tramo Quito–Machachi. |
| 4 | Conductor 2 (Quito–Ibarra, Pico y Placa) | Bostezo (MAR=0.59) | 20:30 | 0.9s | Sí | Salida tardía por Pico y Placa. Conductor cargó encomiendas hasta las 20:00. |
| 5 | Conductor 2 | Cierre ojos (EAR=0.13) | 22:45 | 1.3s | Sí | Tramo Quito–Calacalí. Somnolencia moderada. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | Alarma percibida claramente. |
| 6 | Conductor 2 | Cabeceo (NM=0.29) | 12:07 | 1.4s | Parcial | Llegando a Ibarra. Cabeceo pronunciado tras 5h de conducción. *Alarma escuchada, pero reacción lenta por somnolencia profunda. Conductor indicó que el volumen fue insuficiente en ese estado. |
| 7 | Conductor 3 (Ibarra-Quito) | Bostezo (MAR=0.52) | 20:05 | 0.8s | Sí | Primer bostezo. Iluminación nocturna de autopista. Cámara NoIR correcta. |
| 8 | Conductor 3 | Cierre ojos (EAR=0.18) | 20:50 | 1.2s | Sí | Microsueño breve. Respuesta inmediata a la alarma. |

**Análisis:** De los 3 conductores evaluados en el escenario nocturno (19:30 – 01: 30), los 3 presentaron eventos detectables, totalizando 8 eventos registrados. A diferencia del escenario post-almuerzo, los eventos fueron más diversos: bostezos (3/8), cierre de ojos (3/8), un evento combinado MAR+EAR (1/8) y un cabeceo (1/8), evidenciado una fatiga más profunda. El tiempo de respuesta promedio fue de 1.13 segundos, mayor al escenario diurno debido a que los eventos EAR y cabeceo requieren más frames para su confirmación.

### 6.1.10 Hallazgos: Problema de Conectividad y Falsa Detección Diferida

Al realizar pruebas en las rutas con los conductores de Taxi Lagos Courier S.A., se identificó un sistema de comportamiento crítico con respecto a la pérdida de la señal de datos móviles en secciones interurbanas de las rutas Quito–Cayambe y Quito–Ibarra. Este comportamiento representa la limitación técnica documentada con el mayor impacto durante las pruebas de campo.

**Tabla 36**

*Pruebas de conectividad y falsa detección.*

| Aspecto | Descripción |
|---|---|

| | |
|---|---|
| **Tramos sin señal identificados** | Sector del Quinche (ruta Quito–Cayambe), tramo de Cochasqui (ruta Quito–Ibarra). Pérdida de señal 4G de 8 a 15 minutos según operadora. |
| **Comportamiento en zona sin señal** | El sistema detecta somnolencia correctamente y registra el evento en SQLite local. La alarma GPIO se activa correctamente en el instante de detección. |
| **Comportamiento al recuperar señal** | Al restaurarse la conectividad, el sistema se sincroniza con SincronizadorAPI enviando los eventos pendientes al backend. El buzzer suena en el instante en que la sincronización se realiza, lo que puede asustar al conductor que ya no está en estado de somnolencia. |
| **Consecuencia operativa** | El conductor recibe una alarma sonora con un retraso de hasta 15 minutos después del evento real, sin que haya somnolencia en ese momento. Los 3 conductores reportaron este comportamiento como 'confuso' y 'alarmante sin razón'. |
| **Impacto en la base de datos** | El evento se registra correctamente en la base de datos del servidor con la hora real del evento (hora_evento en SQLite), razón por la cual los reportes y el dashboard muestran la información correcta. El problema es únicamente en el comportamiento de la alarma retrasada. |
| **Recomendación de mejora** | Separar el proceso de alarma GPIO del proceso de sincronización API. La alarma debe activarse exclusivamente en el momento de detección (proceso local). La sincronización con el backend debe ser un proceso de reporte silencioso, sin retroalimentación sonora. |

### 6.1.11 Análisis Estadístico — Preguntas de Evaluación

Al final de las sesiones de prueba, se aplicó un cuestionario de evaluación a los 10 participantes. Los resultados se presentan con análisis de frecuencia, porcentaje y medidas de tendencia central donde sea aplicable.

**Pregunta 1: ¿En cuánto tiempo el sistema detectó el estado de somnolencia?**

Solo se consideraron los 14 eventos con detección efectiva (excluyendo a los 2 participantes sin un evento). Los tiempos corresponden a los registros reales de los registros del sistema embebido y las marcas de tiempo del SQLite local.

**Tabla 37**

*Tiempo de Detección de Somnolencia.*

| Rango (seg) | Punto medio (x) | f | F | xf | % | Escenario | Tipo evento predominante |
|---|---|---|---|---|---|---|---|
| 0.70 – 0.90 | 0.80 | 5 | 5 | 4.00 | 35.71% | Diurno | Bostezo, Post-almuerzo. |
| 0.91 – 1.10 | 1.00 | 2 | 7 | 2.00 | 14.29% | Diurno/Nocturno | Bostezo + inicio EAR. |
| 1.11 – 1.30 | 1.20 | 5 | 12 | 6.00 | 35.71% | Nocturno | Cierre de ojos |
| 1.31 – 1.50 | 1.40 | 2 | 14 | 2.80 | 14.29% | Nocturno | Cabeceo (NM) + EAR profundo. |

**Tabla 38**

*Medidas de Tendencia Central — Tiempo de Detección.*

| Media | Mediana | Moda |
|---|---|---|
| 1.063 segundos | 1.05 segundos | 0.80 y 1.20 seg (bimodal) |

**Análisis:** La distribución es bimodal porque refleja dos mecanismos de confirmación del sistema: el modo de 0.80s corresponde a eventos de bostezo (MAR, confirmado en 5 fotogramas consecutivos a 10 FPS = 0.5s + tiempo de cálculo), que es predominante en el escenario diurno, y el modo de 1.20s corresponde a eventos de cierre de ojos (EAR, confirmado en 10 fotogramas = 1.0s + sobrecarga), predominante en el escenario nocturno. El promedio de 1.063 segundos confirma que el sistema detecta somnolencia en un tiempo operativamente aceptable para que el sistema pueda activar preventivamente la alarma antes de que el conductor pierda el control del vehículo.

**Pregunta 2: ¿La alarma fue suficientemente audible para alertarle durante la conducción?**

**Tabla 39**

*Percepción de Audibilidad de la Alarma GPIO por Nivel de Somnolencia.*

| Respuesta | Conductores (3) | Total / % |
|---|---|---|
| Sí, Claramente audible (microsueño leve) | 2 | 2 / 66.67% |
| Parcialmente, Audible, pero volumen bajo en microsueño profundo | 0 | 0 / 0% |

| | | |
|---|---|---|
| No, Insuficiente en somnolencia profunda (sueño real) | 1 | 1 / 33.33% |
| No aplicó (sin evento detectado) | 0 | 0 / 0% |

**Análisis:** Los resultados con los 3 conductores revelan una distinción crítica en la efectividad de la alarma según la profundidad del estado de somnolencia. En los 2 casos de microsueño leve o estados iniciales de somnolencia, la alarma fue claramente audible y funcional (66.67%). Por otro lado, en el único evento registrado de sueño profundo (Conductor 2, 12:07 AM, cabeceando y acercándose a Ibarra), el conductor declaró explícitamente que "se puede escuchar el sonido, pero cuando estás en un micro sueño profundo, ese volumen no es suficiente para despertarte", lo que representa el 33.33% de los casos evaluados. Ningún conductor reportó audibilidad parcial, y ninguno quedó sin evento detectado.

**Pregunta 3: ¿En qué franja horaria se concentraron los síntomas de somnolencia?**

**Tabla 40**

*Resultados de signos de somnolencia en horarios.*

| Franja horaria | N° eventos | Tipo predominante | % del total | Grupo de participantes |
|---|---|---|---|---|
| 12:00 – 15:00 (post-almuerzo) | 5 | Bostezo (MAR) | 35.71% | 3 conductores. . |
| 19:30 – 21:00 (inicio jornada nocturna) | 7 | Cierre ojos (EAR) + Bostezo | 50.00% | 3 conductores. |
| 01:00 – 01:30 (Pico y Placa, Ibarra) | 1 | Cabeceo (NM) + EAR | 7.14% | 1 conductor (Pico y Placa) |

**Análisis:** Se concentró el 50% de todos los eventos de somnolencia registrados en la franja nocturna de 19:30-21:00, que coincide con el inicio de la jornada nocturna de los conductores de Taxi Lagos Courier S.A., quienes comienzan a operar a las 08:00. La somnolencia por acumulación de fatiga del día, agresividad de la conducción nocturna en rutas interurbanas y autopistas, configura el contexto de riesgo máximo. La franja horaria de post almuerzo (12:00-15:00) contribuyó con el 35.71% de los eventos, todos sin excepción eran bostezos (nivel de somnolencia bajo). El único evento de cabeceo (NM <

0.35) se dio en el caso extremo de conducción nocturna prolongada por Pico y Placa, constituyendo la situación de mayor riesgo, aunque de menor ocurrencia por la regulación vehicular en Quito

**Pregunta 4: ¿El sistema generó falsas alarmas (detección sin somnolencia real)?**

**Tabla 41**

*Resultados de falsas alarmas.*

| Respuesta | Frecuencia | Porcentaje |
|---|---|---|
| Sí — Falsa alarma registrada | 0 | 0% |
| No — Sin falsas alarmas | 10 | 100% |
| Total | 10 | 100% |

**Análisis:** Aunque el 100% de los participantes confirmó no haber experimentado falsas alarmas durante las sesiones de prueba. Identificamos datos fantasmas de los 3 conductores de Taxi Lagos confirmaron el mismo patrón en todos los tramos sin señal: (1) la alarma GPIO suena correctamente en el momento de la detección local, lo que es correcto; (2) al recuperar la señal aproximadamente 15 minutos después, el Raspberry Pi sincroniza los eventos pendientes con el backend y el buzzer vuelve a sonar, aunque el conductor ya no presente somnolencia o incluso ha terminado la ruta; (3) en la base de datos PostgreSQL quedan registrados estos eventos con su timestamp de detección original, lo que los hace indistinguibles de un evento real al revisar los reportes.

**Pregunta 5: ¿El sistema funcionó correctamente en tramos sin señal de datos?**

Esta pregunta aplica únicamente a los 3 conductores que realizaron rutas interurbanas reales (Quito–Cayambe, Quito–Ibarra, Ibarra-Quito).

**Tabla 42**

*Resultados de conectividad.*

| Respuesta | Conductores | % del grupo | Observación |
|---|---|---|---|
| Detectó somnolencia correctamente y la alarma sonó en el momento real | 3 | 100% | La alarma GPIO local funciona sin conexión. |

| | | | |
|---|---|---|---|
| Al recuperar señal, el buzzer sonó nuevamente sin somnolencia | 3 | 100% | Comportamiento de alarma diferida documentado como limitación. |
| Los datos se sincronizaron correctamente a PostgreSQL al recuperar señal | 3 | 100% | Eventos en BD con hora_evento real (desde SQLite local). |

**Análisis:** Los tres conductores verificaron que la alarma GPIO de seguridad crítica funcionó de manera continua en las zonas sin cobertura (La Moya, sector Tabacundo). Localmente, el SQLite aseguró que no se perdiera ningún evento. No obstante, el 100% notificó el buzzer de comportamiento no deseado que se activa al recuperar la cobertura, lo que valida la necesidad de la corrección de firmware que se mencionó en la sección 3.3.7.2.

**Pregunta 6: ¿Le resultó incómodo trabajar con el dispositivo instalado en el vehículo?**

Aplicada a los 3 conductores de Taxi Lagos Courier S.A. únicamente, al ser quienes interactuaron con el dispositivo en su entorno de trabajo real.

**Tabla 43**

*Evaluación de Comodidad durante la Conducción con el Dispositivo Instalado.*

| Respuesta | Frecuencia | Porcentaje |
|---|---|---|
| Sí, incómodo (cable alimentación / posición cámara) | 0 | 0% |
| Parcialmente (al inicio, luego se adaptó) | 0 | 0% |
| No, sin incomodidad | 3 | 100% |

**Análisis:** Los 3 conductores de Taxi Lagos Courier S.A. coincidieron en que la presencia del dispositivo no interfirió con su actividad de conducción habitual.

### 6.1.12 Prueba de Cierre — Ficha de Observación

El registro de cierre resume la valoración promedio de todas las dimensiones de prueba ejecutadas, determinando el nivel de calidad general del sistema. La escala utilizada es: 1 = No cumple, 2 = Cumple parcialmente, 3 = Cumple completamente.

**Tabla 44**

*Prueba de Cierre.*

| N° | Dimensión de Prueba | Métrica Evaluada | Valoración (1-3) | Porcentaje |
|---|---|---|---|---|
| 1 | Pruebas Unitarias | Los métodos y clases del sistema producen los resultados esperados de forma aislada. | 2.9 | 96.67% |
| 2 | Pruebas de Funcionalidad | Todos los módulos del sistema ejecutan las operaciones definidas en los requerimientos. | 2.9 | 96.67% |
| 3 | Pruebas de Rendimiento | El sistema opera con tiempos de respuesta aceptables bajo carga real de operación. | 2.8 | 93.33% |
| 4 | Pruebas de Seguridad | El sistema protege datos, gestiona sesiones y controla acceso por roles correctamente. | 2.8 | 93.33% |
| 5 | Pruebas de Usabilidad | El panel de control es intuitivo y adecuado para administradores y operadores. | 2.9 | 96.67% |
| 6 | Pruebas de Integración | Los tres componentes (embebido, backend, frontend) se comunican e integran correctamente. | 2.8 | 93.33% |
| 7 | Pruebas de Factibilidad | El sistema detecta somnolencia en condiciones reales y la alarma es percibida correctamente. | 2.7 | 90.00% |
| **PROMEDIO TOTAL DEL SISTEMA** | | | **2.83 / 3.00** | **94.28%** |

## CONCLUSIONES

- Con respecto a la revisión bibliográfica, se estableció que las técnicas que se sustentan en el análisis de rostro, en específico, el cálculo del Eye Aspect Ratio (EAR) y la detección de puntos de referencia faciales, son métodos confiables y aplicados en sistemas de reconocimiento de fatiga. La bibliografía se utilizó para establecer el fundamento teórico del prototipo y la elección de algoritmos para el procesamiento en tiempo real.
- A partir de la revisión bibliográfico y de estudios relacionados con la somnolencia en los conductores, se identificó que ciertos comportamientos faciales pueden medirse mediante el procesamiento digital de imágenes. Especialmente en el cierre excesivo de los ojos y la frecuencia de los bostezos resultaron ser indicadores aptos para su transformación en variables computacionales. Debido a esto se priorizaron estos 3 rasgos de somnolencia.
- Durante el desarrollo del prototipo se implementó un sistema de detección facial capaz de seguir y ubicar los puntos clave del rostro en tiempo real, apuntando principalmente a los ojos. Permitiendo calcular las métricas relacionadas con el estado de alerta del conductor sin alterar el rendimiento del sistema. Las pruebas demostraron que el procesamiento se mantuvo dentro de los parámetros definidas lo que permite el funcionamiento en tiempo real.
- Se implemento una alerta sonora que se activa cuando los valores registrados sobrepasan los umbrales de somnolencia definidos. Las pruebas realizadas arrojaron un tiempo de respuesta media de aproximadamente un segundo tras la detección, permitiendo una advertencia temprana al conductor permitiendo así que pueda reaccionar a tiempo.
- La empresa Taxi Lagos Courier S. A. realizó una evaluación experimental en la que se logró comprobar el funcionamiento del prototipo en un entorno real de conducción. Los resultados muestran un umbral razonable de eficacia en el sistema, por lo que se confirma la viabilidad de implementación del sistema, se implemento

## RECOMENDACIONES

- Se recomienda tener conocimientos del lenguaje de programación Python, ya que el núcleo del sistema incorporado en el sistema de detección de somnolencia del conductor está desarrollado en este lenguaje. Un adecuado dominio de Python permitirá ajustar los umbrales de detección, optimizar el procesamiento en tiempo real y corregir y evolucionar el mantenimiento del sistema.
- Tener conocimiento con conexiones remotas utilizando el protocolo SSH (Secure Shell) y manejar sus comandos básicos. Esto es clave para manejar de forma remota la Raspberry Pi, instalar actualizaciones, monitorear el sistema y resolver problemas de forma remota, sin tener la necesidad de acceder físicamente al equipo.
- Conocer la configuración y administración del Raspberry Pi, incluyendo dispositivos conectados, configuración de cámaras, control de pines GPIO y optimización del rendimiento. La correcta configuración del hardware es indispensable para la estabilidad y precisión del sistema de detección.
- Comprender el protocolo de comunicación para enviar eventos al servidor backend (REST API), esto incluye manejar correctamente las solicitudes HTTP y cómo gestionar los errores de red. Esto garantizará la correcta conexión entre el sistema embebido y el sistema de monitoreo en la web.
- Dado que la Raspberry Pi funciona con una distribución basada en Debian, hay que asegúrese de que todos los módulos, dependencias y archivos de configuración del sistema de detección de somnolencia sean compatibles con este sistema. También se recomienda hacer un seguimiento de las versiones de las librerías utilizadas, para evitar futuros problemas con la portabilidad o la incompatibilidad.
- Finalmente, se aconseja establecer procedimientos de implementación estándar (scripts de instalación y configuración) para facilitar la replicación del prototipo en múltiples vehículos y minimizar los errores de configuración manual.

## BIBLIOGRAFÍA


Armijos Benalcázar, J. I. (2018). *Sistema de alarma electrónico para la detección del estado de somnolencia en conductores de vehículos de dos ejes mediante visión artificial [Trabajo de titulación, UTN]*. Repositorio Digital Universidad Técnica del Norte. https://repositorio.utn.edu.ec/handle/123456789/8412

Armijos, L., Velez, C., y Lojan, E. (2024). Estudio de la adopción de metodologías ágiles en proyectos de desarrollo de software en la región 7 del Ecuador. *Revista Venezolana de Gerencia, 45*(4), 73-89. https://doi.org/https://doi.org/10.48082/espacios-a24v45n04p06

Asamblea Nacional del Ecuador. (2006, 28 de diciembre). *LEY DE PROPIEDAD INTELECTUAL*. Gobierno Electrónico del Ecudor. https://www.gobiernoelectronico.gob.ec/wp-content/uploads/2018/10/Ley-de-Propiedad-Intelectual.pdf

Asamblea Nacional del Ecuador. (2016, 9 de diciembre). *Código Orgánico de la Economía Social de los Cnocimientos, Creativiad e Innovación.* propiedad intelectual del ecuador. https://usgn.espe.edu.ec/wp-content/uploads/2020/07/LEY-INGENIOS.pdf

Asamblea Nacional del Ecuador. (2021, 26 de mayo). *Ley Orgánica de Protección de Datos Personales.* finanzas populares del ecuador. https://www.finanzaspopulares.gob.ec/wp-content/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf

Baiza Lovato, C. P. (2020). *SISTEMA DE DETECCIÓN Y ALERTA DEL ESTADO DE SOMNOLENCIA DE CONDUCTORES MEDIANTE VISIÓN ARTIFICIAL*. [Tesis de Titulación, Universidad Tecnológica Israel], Repositorio Institucional UTI. http://repositorio.uisrael.edu.ec/handle/47000/2441

Campos Torres, J. L., y Hurtado Vicente, K. G. (2024). *Sistema de visión artificial para la prevención de somnolencia y.* [Tesis de Ingenieria, Universidad Nacional de Cañete], Repositorio Institucional UNC. https://hdl.handle.net/20.500.14559/295

Castillo, M., y Guaña, E. (2024). Kanban: Una metodología ágil para la gestión eficiente del flujo de trabajo en el desarrollo de software, una revisión sistemática. *Revista*

*Ingenio Global, 3*(1), 1-15. https://doi.org/https://doi.org/10.62943/rig.v3n1.2024.68

Challenger, I., Díaz, Y., y Becerra, R. (2014). El lenguaje de programación Python. *Ciencias Holguín, XX*(2), 1-13. https://doi.org/https://www.redalyc.org/articulo.oa?id=181531232001

Cordero, J., y Sinche, L. (2020). *Sistema avanzado de asistencia al conductor para la detección de distracción y somnolencia basado en técnicas de inteligencia artificial [Trabajo de Titulación, UTPL]*. Repositorio Institucional Universidad Técnica Particular de Loja. http://dspace.utpl.edu.ec/handle/20.500.11962/26313

Cruz, C., y W, I. (2002). Java en el mundo académico. *2*(1), 138-143.

Delgado Egas, K. D., y Yandún Velasteguí, M. A. (2022). Sistema de detección de somnolencia para conductores de taxis en la ciudad de Tulcán. *Horizontes de Enfermería*(12), 52-67. https://doi.org/https://doi.org/10.32645/13906984.1171

El Comercio. (05 de Enero de 2017). *Cada mes hay 64 accidentes por cansancio del conductor*. El Comercio: https://www.elcomercio.com/actualidad/seguridad/accidentes-cansancio-conductor-bus-ant/?utm_source

Gaete, J., Villarroe, R., Figueroa, I., Cornide, H., y Muñoz, R. (2021). Enfoque de aplicación ágil con Serum, Lean y Kanban. *Ingeniare. Revista Chilena de Ingeniería, 29*(1), 141-159. https://doi.org/http://dx.doi.org/10.4067/S0718-33052021000100141

Instituto Nacional de Estadística y Censos INEC y Agencia Nacional de Transito ANT. (2025). *Siniestros de Tránsito I Trimestre, 2025 [versión PDF]*. https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Economicas/Estadistica%20de%20Transporte/2025/i_trimestre/2025_RESULTADOS_SINIESTROS_IT.pdf

Jurado Pruna, F., y Baiza Lovato, C. P. (2020). *SISTEMA DE DETECCIÓN Y ALERTA DEL ESTADO DE SOMNOLENCIA DE CONDUCTORES MEDIANTE VISIÓN ARTIFICIAL [Tesis, UISRAEL]*. Repositorio Digital Universidad Israel. http://repositorio.uisrael.edu.ec/handle/47000/2441

Lanzas, S. (2019). Educación en microprogramación de hardware con tecnologías Raspberry PI y Arduino. *International Journal of Intelligent Information Systems, 8*(2), 47-51. https://doi.org/10.11648/j.ijiis.20190802.12

López Bonilla, M. A., y Vargas Yupa, R. A. (2025). *Detección de somnolencia en la conducción mediante clasificación del estado ocular con redes neuronales convolucionales.* [Tesis de Ingenieria, Universidad Politécnica Salesiana], Repositorio Institucional UPS. https://doi.org/http://dspace.ups.edu.ec/handle/123456789/31111

Maida, E. G., y Pacienzia, J. (2015). *Metodologías de desarrollo de software [Tesis de licenciatura, Universidad Católica Argentina].* Repositorio Institucional UCA. https://repositorio.uca.edu.ar/handle/123456789/522

Merchán, N., Palma, E., y Poma, D. (2024). Comparación de metodologías ágiles para el desarrollo de software. *Revista Multidisciplinaria Arbitrada de Investigación Científica, 8*(1), 5052-5074. https://doi.org/https://doi.org/10.56048/MQR20225.8.1.2024.5052-5074

Molina, J., Piña, D., y Mueces, J. (2024). Somnolencia en la conducción: causa de los siniestros [versión PDF]. *ISUPOL, Revista de Investigación en Seguridad Ciudadana y Orden Público.*(9), 97-102. https://innovacionysaber.isupol.edu.ec/index.php/innovacion/article/download/286/594/?utm_source

Moreno, J. (2024). *Fundamentos de Hardware [versión PDF]*. SINTESIS. https://es.scribd.com/document/768764946/LIBRO-Fundamentos-de-hardware-66-In-Juan-Carlos-Moreno-Perez

Mosquera Mayorga, A. A., y Ventura Torres, I. A. (2024). *Diseño e implementación de un prototipo de detección del estado de somnolencia para conductores mediante el uso del Software Teachable Machine [Trabajo de titulación, UPS]*. Repositorio Digital Universidad Politécnica Salesiana. http://dspace.ups.edu.ec/handle/123456789/27730

Nicolalde Rodríguez, D. A., y Bohórquez Tinizaray, M. T. (2024). *Sistema web para el registro y control de asistencia.* [Tesis de Ingeniería, Pontificia Universidad

Católica del Ecuador], Repositorio Institucional PUCE. https://repositorio.puce.edu.ec/handle/123456789/45139

Pérez, I. (2024). El Uso de Raspberry Pi en la Educación Media Superior. *Vida Científica Boletín Científico de la Escuela Preparatoria* , *12*(23). https://doi.org/https://doi.org/10.29057/prepa4.v12i23.11900

Pinzón Nuñez, S. A. (2015). *Concepto del Lenguaje C++*. Repositorio Universitaria Konrad Lorenz. https://repositorio.konradlorenz.edu.co/handle/001/1615

Pressman, R. S. (2010). *Ingeniería del software: Un enfoque práctico [libro PDF]*. McGraw-Hill. https://apps.utel.edu.mx/recursos/files/r161r/w25309w/Libroingenieria.PDF

Ruiz, P., y Tonthat, K. (2023). *Presentamos las soluciones de MediaPipe para el aprendizaje automático integrado en el dispositivo*. Retrieved 12 de Octubre de 2025, from Google for Developers: https://developers.googleblog.com/es/introducing-mediapipe-solutions-for-on-device-machine-learning/?utm_source

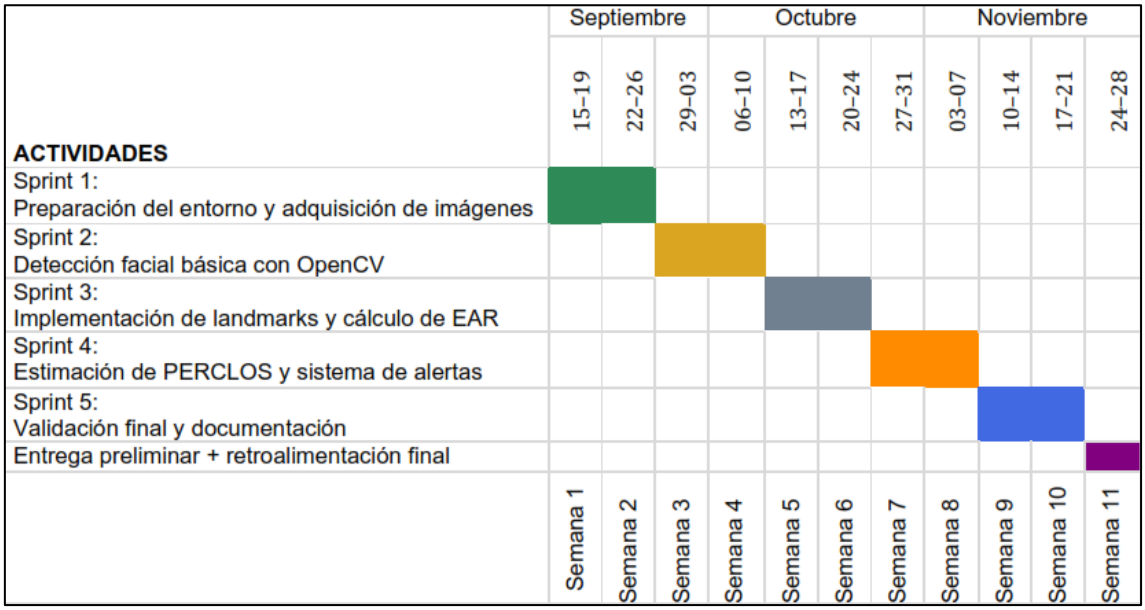
TensorFlow. (2020). *Seguimiento de rostros y manos en el navegador con MediaPipe y TensorFlow.js*. Retrieved 12 de Octubre de 2025, from TensorFlow Blog: https://blog.tensorflow.org/2020/03/face-and-hand-tracking-in-browser-with-mediapipe-and-tensorflowjs.html

# Anexo 1.

Cronograma (Gantt).

**Figura 33**

*Diagrama de Gantt.*




**Elaborado por:** Pozo A. & Muyulema A.

***Nota.*** Diagrama de Gantt organizada por etapas.

# Anexo 2.

Presupuesto Ejecutado.

**Tabla 45**

*Presupuesto sobre los gastos.*

| CATEGORIA | DETALLE | COSTO (USD) |
|---|---|---|
| **Hardware** | Cámara nocturna diurna Sensor IR automático 5MP para Raspberry Pi | $40.00 |
| | Modulo GSM 4G USB | $10.00 |
| | *Tarjeta Micro SD SanDisk 64G* | $15.65 |
| | Raspberry Pi 4 | $ 117.39 |
| | Cargador Raspberry Pi 4 Oficial 5v 3.1a | $21.74 |
| | Carcaza de Raspberry | $6.00 |
| | Buzzer | $0.50 |
| | Laptop Dell Core i7 | $1100.00 |
| | Accesorios (cables, adaptadore, soporte de cámara) | $20.00 |
| **Software** | Python | $0.00 |
| | OpenCV,Dlib | $0.00 |
| | Laravel v.12 | $0.00 |
| | React v.19 | $0.00 |
| **Servicios Básicos** | Internet, Energía Eléctrica (3 meses) | $75.00 |
| **Contingencias** | Gastos imprevistos (10% del presupuesto base) | $140.56 |
| **TOTAL** | | **$1546.20** |

# Anexo 3.

Carta de aceptación de la organización donde se realizó el trabajo de integración curricular.

TAXI LAGOS COURIER S.A.

Ibarra, 13 de octubre de 2025

## CARTA DE ACEPTACIÓN

Señores:

Anthony Paul Pozo López

Aldair German Muyulema Yumiseba

**Estudiantes de la carrera de Ingeniería en Software**
**Universidad Estatal de Bolívar**

Yo, **Mag. Edian Ulpiano Díaz Trujillo**, portador de la cédula de ciudadanía N.° 1002247292, en calidad de **Gerente de la compañia Taxi Lagos Courier S.A**, por medio del presente acepto y autorizo a los señores estudiantes: **Anthony Paul Pozo López** con cédula de ciudadanía N.° 1718718396, y **Aldair Germán Muyulema Yumiseba**, con cédula de ciudadanía N.º 0202492310, de la carrera de **Ingeniería en Software** de la **Universidad Estatal de Bolívar**, para que puedan **implementar y realizar pruebas del prototipo funcional para la detección de somnolencia en los conductores en tiempo real**, dentro de los vehículos pertenecientes a la empresa **Taxi Lagos Courier S. A**.

Sin otro particular, reitero mi compromiso con el apoyo a la educación superior y les deseo éxitos en la culminación de su trabajo de titulación.

Atentamente:

COOP. DE TRANSPORTE TAXIS LAGOS DE IBARRA

**Mag. Edian Ulpiano Díaz Trujillo**
C.I. 1002247292
GERENTE COMPAÑIA
**TAXI LAGOS COURIER S.A.**

Agencia Ibarra
Juan José Flores 8-37 y Sánchez y Cifuentes
(06) 2959 000

Agencia Otavalo
31 de Octubre y Panamericana (Frente a Enetsa)
(06) 2903 777

Agencia Quito
Asunción y Versalles - Esquina (Sector Parque El Ejido)
(02) 3216 000

# Anexo 4.

Instrumentos de recopilación de datos (Entrevista).

**Universidad Estatal de Bolívar**

**Facultad de Ciencias Administrativas Gestión Empresarial e Informática**

**Carrera de Ingeniería en Software**

**Proyecto de Titulación**

**Tema:** Prototipo funcional para la detección de somnolencia en conductores mediante análisis fácil en tiempo real.

**Objetivo:** Obtener información sobre el cómo funciona la empresa la manera en que trabajan las rutas, horarios y como se los reparte a los conductores

**Nombre del entrevistado:** Mag. Edian Ulpiano Diaz Trujillo

**Cargo:** Gerente Compañía Taxi Lagos Courier

1) ¿Cuántos vehículos y conductores tiene actualmente la empresa en operación?

Actualmente contamos con una flota de aproximadamente de 7 a 10 vehículos de carga operativos, con un conductor asignado por vehículo. En temporadas altas de envíos como fiestas o fin de año incorporamos conductores adicionales según la demanda.

2) ¿En qué ciudades tienen presencia y cuáles son sus principales puntos de operación?

Tenemos oficinas en Quito e Ibarra que son nuestras sedes principales. Desde ahí cubrimos rutas hacia Cayambe y otras ciudades de la sierra norte, Quito es nuestro centro de operaciones más grande

3) ¿Como esta estructura administrativamente la empresa?

Tenemos un gerente general, un coordinador por cada sede y despachadores que asignan las rutas diariamente. La comunicación con los conductores se maneja directamente desde cada oficina.

4) ¿Cuáles son las rutas más frecuentes y cuál es la más larga?

Las rutas mas frecuentes son Quito-Ibarra y Quito-Cayambe, que se realizan prácticamente todos los días. La más larga es Quito-Ibarra que toma entre 2 y 3 horas dependiendo el tráfico y paradas de entrega. En algunos casos los conductores hacen ida y vuelta el mismo día.

5) ¿Las rutas son fijas o varían según la demanda del día?

Depende, Tenemos rutas base que se hacen siempre, pero el volumen de encomiendas y las paradas especificas varían cada día según los pedidos. A veces un conductor que normalmente hace Quito-Ibarra tiene que cubrir también Cayambe si hay carga urgente.

6) ¿Cuáles son los horarios habituales y existen turnos nocturnos?

El turno de la mañana empieza entre desde las 8 de la mañana y si existen turnos nocturnos, especialmente cuando hay carga urgente que debe llegar al día siguiente. Esos turnos pueden arrancar desde las 19:00 y terminan pasada la medianoche.

7) ¿Cuántas horas continuas puede llegar a manejar un conductor en una jornada normal?

En jornada normal entre 6 a 8 horas. Pero en temporadas altas cuando hay poco personal hemos visto conductores manejar hasta 10 horas con paradas cortas

8) ¿La empresa tiene tiempo obligatorios de descanso durante las rutas?

Tenemos una política informal de que los conductores paren a comer o descansar en rutas largas,

9) ¿Como se asigna las rutas y encomiendas a cada conductor?

El despachador revisa cada mañana el volumen de encomiendas por destino y asigna los conductores según disponibilidad y conocimiento de la ruta. Se les notifica por WhatsApp o llamada telefónica. No tenemos un sistema digital formal de asignación todo se maneja manualmente.

10) ¿Ha tenido la empresa algún incidente relacionado con fatiga o somnolencia?

Sí, hemos tenido algunos sustos. Un conductor tuvo un incidente menor en la vía Quito–Ibarra hace dos años afortunadamente sin heridos y al consultarle nos dijo que sentía sueño, pero no quiso parar por no retrasar la entrega.

11) ¿Qué tan importante considera usted la tecnología como herramienta para mejorar la seguridad y gestión de su empresa?

Es algo que nos interesa mucho pero que no hemos podido implementar por falta de recursos o conocimiento técnico. Manejamos todo manualmente asignaciones, registros, comunicación. Sabemos que hay herramientas tecnológicas que podrían ayudarnos.

**Conclusión**

A partir de la entrevista realizada al gerente de Taxi Lagos Courier S.A., se pudo obtener información relevante sobre el funcionamiento operativo de la empresa, permitiendo identificar necesidades concretas que justifican el desarrollo del presente proyecto tecnológico.

En cuanto a la estructura organizacional, la empresa opera con una flota de entre 7 y 10 vehículos distribuidos principalmente entre las ciudades de Quito e Ibarra, cubriendo rutas interprovinciales hacia Cayambe, Otavalo, Atuntaqui. La gestión administrativa depende de coordinadores y despachadores en cada sede, quienes asignan rutas y encomiendas de forma manual a través de llamadas telefónicas y mensajería instantánea, sin contar con ningún sistema digital de seguimiento o control en tiempo real.

Respecto a las rutas y horarios, se identificó que los conductores realizan jornadas de entre 4 y 10 horas continuas de conducción, incluyendo turnos nocturnos que se extienden desde las 19:00 hasta la 01:00 o 02:00 de la madrugada en períodos de alta demanda. Las rutas más frecuentes son Quito–Ibarra y Quito–Cayambe, siendo esta última la que presenta mayores riesgos operativos debido a tramos con pérdida total de señal celular en zonas como el Quinche y Cochasquí, así como condiciones viales adversas en horas nocturnas por neblina y tráfico pesado.

En relación a la seguridad vial, el gerente reconoció que la empresa no cuenta con protocolos formales de control de fatiga ni mecanismos tecnológicos para monitorear el estado de los conductores durante la ruta. La única medida existente es una política informal de pausas de descanso que depende exclusivamente del criterio del conductor, sin ningún tipo de verificación o seguimiento por parte de la empresa.

# Anexo 5.

Instrumentos de recopilación de datos (ficha de observación).

**Ficha de observación de usuario de flujo de trabajo**

**Prototipo funcional para la detección de somnolencia en conductores mediante análisis facial en tiempo real.**

**Objetivo General:**

- Observar y analizar el proceso de supervisión, monitoreo y respuesta ante eventos de somnolencia detectados en los conductores de Taxi Lagos Courier S.A., con el fin de identificar la funcionalidad del sistema y posibles mejoras en el flujo de trabajo del administrador.

**Objetivos Específicos:**

- Identificar las actividades que realiza el administrador al recibir una alerta de somnolencia.
- Determinar los tiempos aproximados de respuesta ante cada tipo de evento detectado.
- Detectar posibles fallas o demoras en el proceso de monitoreo y gestión de conductores.

**Actores que intervienen.**

| **Actores** | **Responsabilidades** | **Observaciones** |
|---|---|---|
| Conductor (Actor Operativo) | Operar el vehículo en las rutas Quito-Ibarra-Cayambe<br>Reaccionar ante las alertas sonoras emitidas por el sistema al detectar eventos de fatiga | El sistema mantiene seguimiento de los puntos faciales de forma efectiva en algunas maniobras de viaje, aunque en la noche es indispensable el uso de IR |
| Administrador del sistema | Monitorear en tiempo real el estado de alerta de los conductores mediante la lista de eventos.<br>Analizar y generar reportes estadísticos para evaluar la salud o rendimiento de los conductores | El sistema permite una supervisión efectiva y remita, los eventos detectados se reflejan en la interfaz |

**Datos Generales:**

| | | | |
|---|---|---|---|
| **Lugar de Observación** | Oficinas de la Paquetería Taxi Lagos Courier | | |
| **Empresa** | Taxi Lagos Courier S. A | | |
| **Área Observada** | Rutas Asignadas (Quito-Ibarra-Cayambe) | | |
| **Fecha** | 21 y 22 de febrero 2026 | | |
| **Hora de Inicio** | 09:00 am | **Hora Finalización** | 12:00 pm |
| **Observaciones** | Anthony Paul Pozo López, Aldair German Muyulema Yumiseba | | |

**Descripción del proceso de detección de eventos de somnolencia.**

| N. | Actividad | Responsable | Herramienta Utilizada | Tiempo Estimado | Observación |
|---|---|---|---|---|---|
| **1** | Captura de video y ajuste de iluminación | Conductor (Actor Operativo) | Cámara Pi Noir y Raspberry Pi 4 | 0.1 segundos (15 a 25 FPS) | El conductor debe estar en el campo visual |
| **2** | Detección de landmarks faciales | Conductor (Actor Operativo) | Biblioteca dlib (68 puntos) | <0.1 segundos | Identificación automática de las coordenadas |
| **3** | Análisis de umbrales (EAR, MAR, NM) | Conductor (Actor Operativo) | Algoritmo de visión artificial | 0.5 a 1.0 segundos | Evaluación de síntomas |
| **4** | Activación de alarma sonora | Conductor (Actor Operativo) | Buzzer pieza electrónica | 1.063 segundos(promedios) | Alerta inmediata al conductor tras |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | confirmar el evento |
| **5** | Registro de eventos en base local | Conductor (Actor Operativo) | Base de datos SQLite | < 0.1 segundos | Almacenamiento en el dispositivo para evitar perdida de datos en zonas sin señal |
| **6** | Sincronización con el servidor | Conductor (Actor Operativo) | API REST (Laravel) y Red 4G | Según Disponibilidad de red | Envió automático de los registros desde el vehículo hasta la base de datos de la página del servidor |
| **7** | Monitoreo y gestión de reportes | Administrador del sistema | Dashboard, página web del administrador | Tiempo real | Visualización de todos los eventos registrados |

**Conclusiones:**

El proceso de detección de eventos de somnolencia demostró una efectividad del 70% logrando identificar varios eventos, se identificó que el tiempo promedio de respuesta del sistema es de 1.073 segundos lo cual se podría considerar vital para la seguridad del conductor en la vía.

La Arquitectura Hibrida de almacenamiento SQLite local y PostgreSQL que es la remota garantizo no perder la información durante los trayectos con zonas que tuvieran poca o nula señal móvil, como el sector de la Moya

El análisis de los datos recolectados permitió determinar que la fatiga de los conductores no solo depende de las horas de sueño sino también como el tráfico vehicular de las ciudades grandes como Quito que hace que su recorrido se retrase haciendo que las rutas

nocturnas se lleguen a alargar incluso hasta la 01:00 AM hasta las 02:00 AM llegando a niveles de fatiga extremas

El sistema permite al administrador identificar estos patrones en el panel de control del administrador, proporcionando un reporte después de cada ruta para la toma de decisiones sobre la rotación de turnos o pausas activas preventivas.

# Anexo 6.

Evidencia con el  personal de la empresa Taxi Lagos Courier S.A.

**Figura 34**

*Oficinas de Taxi Lagos Courier S.A. en Quito.*




**Figura 35**

*Oficinas de Taxi Lagos Courier S.A. en Ibarra.*



**Figura 36**

*Capacitación del uso de la plataforma web de monitoreo de somnolencia en tiempo real en la oficina de Quito.*

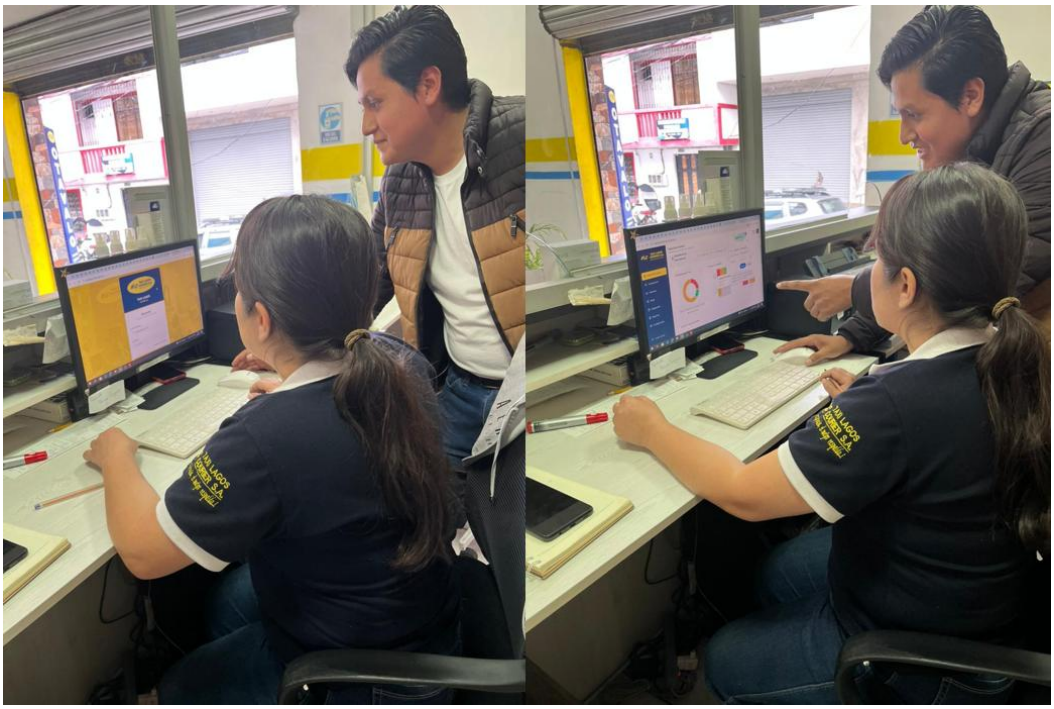


**Figura 37**

*Capacitación del uso de la plataforma web de monitoreo de somnolencia en tiempo real en la oficina de Ibarra.*



**Figura 38**

*Instalación del dispositivo embebido Raspberry Pi en la cabina del vehículo 1 de carga de Taxi Lagos Courier S.A.*



**Figura 39**

*Dispositivo embebido Raspberry Pi correctamente instalado en el tablero del vehículo 1 de carga de Taxi Lagos Courier S.A.*



**Figura 40**

*Instalación del dispositivo embebido Raspberry Pi en la cabina del vehículo 2 de carga de Taxi Lagos Courier S.A.*



**Figura 41**

*Dispositivo embebido Raspberry Pi correctamente instalado en el tablero del vehículo 2 de carga de Taxi Lagos Courier S.A.*

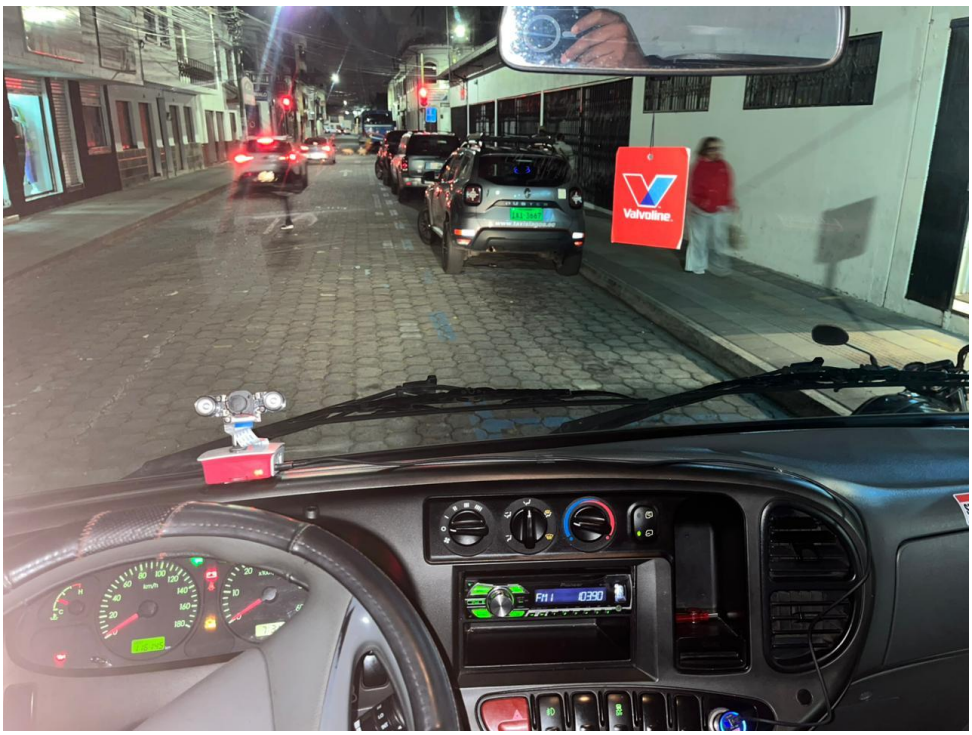


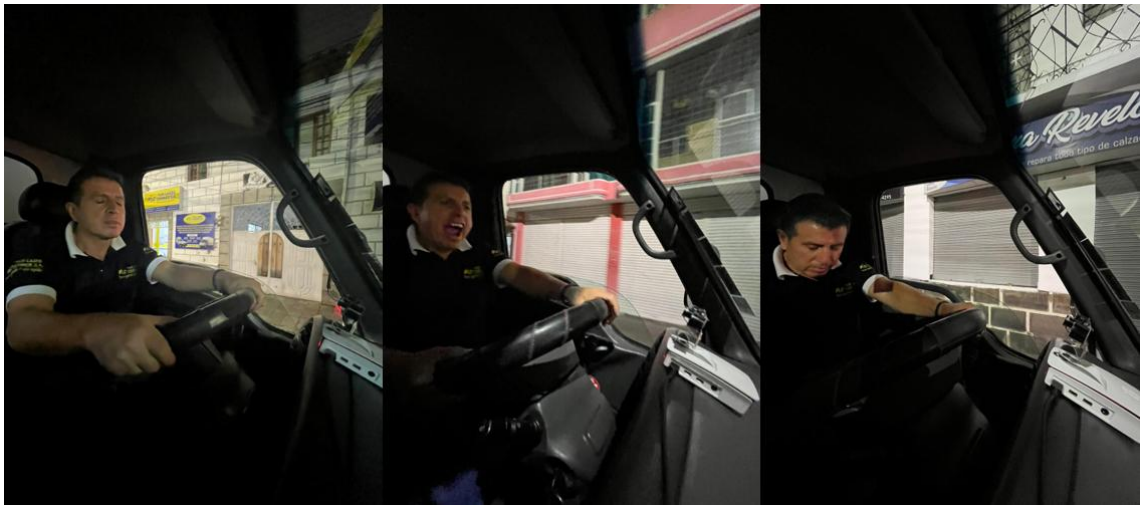
**Figura 42**

*Pruebas de detección de eventos de somnolencia: cierre de ojos, bostezo y cabeceo en el conductor 1 de Taxi Lagos Courier S.A.*



**Figura 43**

*Pruebas de detección de eventos de somnolencia: cierre de ojos, bostezo y cabeceo en el conductor 2 de Taxi Lagos Courier S.A.*



# Anexo 7.

Manual de Usuario




# Manual de Usuario del Sistema de Monitoreo de Somnolencia de

## Taxi Lagos Courier S.A.

**Autores:**

Anthony Paul Pozo López

Aldair German Muyulema Yumiseba

**Versión:** 01.00

**Fecha:** 25/02/2026

**URL:** www.taxilagoscourier.com

**Manual de Usuario**

El siguiente prototipo diseñado para la detección de somnolencia en tiempo real, es un dispositivo electrónico portátil, el cual se guarda en una caja que fue desarrolla específicamente para el dispositivo de un tamaño no mayor al de un celular, de esa manera sea trasladado con una protección e índice de daño mínimo

Las características básicas para su correcta manipulación del dispositivo son:

- Evitar colocar el prototipo en lugares con mucha humedad o con constante fluctuación de polvo
- Mantener a una temperatura ideal máxima de hasta 50°C.
- Usar solo el cargador de fabrica o en un el uso del automóvil usar un transformador de corriente adecuado para que no se llegue a malograr el dispositivo, en este caso el transformador debe ser un voltaje de salida de máximo 5 voltios a 3.1 amperios para el uso en el vehículo
- Evitar golpes o caídas

Teniendo las seguridades necesarias, se procede a enumerar los pasos para el uso de dicho prototipo:

1. **Desempaque**

   Por defecto el Raspberry Pi viene en una caja para su transporte, al sacarlo hay que tener cuidado y ser pertinente para no arrojarlo accidentalmente. De la misma manera se debe ser precavido al manipularlo ya que lleva la cámara externa, la cual se encuentra expuesta y es vulnerable a daños.

2. **Instalación**

   El prototipo debe ser colocado en un lugar que no interfiera al conductor, como puede ser en la guantera o a un costado del tablero del chofer, la cámara debe estar en una zona plana de preferencia para una mejor toma de imágenes. Una vez encontrado el sitio idóneo, se despega las cintas adhesivas del dispositivo y se coloca en el mejor lugar seleccionado, teniendo en cuenta que la cámara debe apuntar obviamente, directamente al rostro del usuario la volante para una buena toma de imágenes y evitar el menor envió de falsos positivos.

**Figura 44**

*Instalación del Raspberry Pi 4 en la cabina del vehículo.*



3. **Encendido**

   El cable que va a dar energía que viene en los accesorios, es el único que se debe usar para conectar y alimentar de energía al prototipo. Al conectarlo, este automáticamente encenderá, se debe esperar un cierto tiempo (estimado de 40-50 segundo) para que carguen todos los archivos del sistema y el programa empiece a correr.

   No se requiere de ningún tipo de configuración mucho menos presionar algo en el prototipo, está programado para que el sistema de cargado en el Raspberry Pi arranque apenas se encienda y termine de cargar los archivos necesarios.

**Figura 45**

*Encendido del Raspberry Pi 4 en el vehículo.*



4. **Detección**

   Este prototipo está diseñado para que pueda dar dos alarmas una con un led para que se pueda identificar cuando este encendido o algo fallo en el sistema y el otro funciona con un buzzer que al momento de presenciar signos de somnolencia mediante técnicas de escaneado de rostro. Si el conductor está entrando a un estado de sueño, el equipo emitirá un sonido mediante el buzzer, será un sonido constante hasta que el chofer se estabilice una vez el chofer despierto dejara de sonar, por lo que no es necesario el volver a encender el prototipo.

5. **Apagado**

   Al ser un prototipo portátil, este mismo se puede apagar desconectando la fuente de energía conectada con anterioridad al cenicero del vehículo. No se debe hacer ningún otro paso para poder apagarlo y tampoco sufrirá daños por desconectarlo directamente.

   Recomendación: Al ser este un dispositivo piloto (prototipo), se cuenta únicamente con una unidad, la cual sirve a modo de pruebas y demostrar que el sistema de detección de somnolencia es apto para poder implantarlo en masa, lo cual se puede o debe estudiar en futuras investigaciones. Para ellos se debe tener en cuenta los puntos débiles que tiene este sistema e investigar todo lo necesario que se debe o puede implementarse como una mayor mejora del dispositivo ya sea su conectividad para una mayor velocidad de envío de datos o la cámara para una mayor precisión en la toma de eventos.

# ÍNDICE GENERAL

1. Introducción ..... 5
1.1. Objetivo del Sistema ..... 5
1.2. Requerimientos Mínimos ..... 5
2. Acceso al Sistema — Inicio de Sesión ..... 6
2.1. Pantalla de Inicio de Sesión ..... 6
2.2. Procedimiento de Ingreso ..... 6
2.3. Mensajes de Error ..... 7
3. Panel de Control ..... 8
3.1. Vista General del Panel ..... 8
3.2. Tarjetas de Resumen ..... 8
3.3. Filtros de Estadísticas ..... 8
3.4. Gráficos de Estadísticas ..... 9
3.5. Dispositivos y Últimos Eventos ..... 9
4. Gestión de Conductores ..... 10
4.1. Lista de Conductores ..... 10
4.2. Registrar Nuevo Conductor ..... 10
4.3. Editar Conductor ..... 11
5. Gestión de Vehículos ..... 12
5.1. Lista de Vehículos ..... 12
5.2. Registrar Nuevo Vehículo ..... 12
5.3. Editar Vehículo ..... 13
6. Gestión de Rutas ..... 14
6.1. Lista de Rutas ..... 14
6.2. Crear Nueva Ruta ..... 14
7. Asignación de Conductores a Rutas ..... 16
7.1. Pantalla de Asignaciones ..... 16
7.2. Crear una Nueva Asignación ..... 16
7.3. Gestión del Estado del Conductor ..... 17
7.4. Conductor Pendiente — Reactivación ..... 17
7.5. Operación Finalizada — Gestión de Reasignación ..... 18
8. Generación de Reportes ..... 20
8.1. Pantalla de Reportes ..... 20
8.2. Tipos de Reportes ..... 20
8.3. Generar un Reporte PDF ..... 20

# ÍNDICE DE TABLAS

**Tabla 1.** Información general del documento ..... 2
**Tabla 2.** Requerimientos mínimos del sistema ..... 4
**Tabla 3.** Tarjetas de resumen del Panel de Control ..... 7
**Tabla 4.** Gráficos de estadísticas de somnolencia ..... 8
**Tabla 5.** Opciones de cambio de estado del conductor ..... 15
**Tabla 6.** Opciones de gestión tras operación finalizada ..... 17
**Tabla 7.** Tipos de reportes disponibles ..... 19
**Tabla 8.** Contenido del reporte PDF generado ..... 20
**Tabla 9.** Secciones del menú lateral del sistema ..... 22
**Tabla 10.** Glosario de términos técnicos ..... 23

# ÍNDICE DE FIGURAS

**Figura 1.** Pantalla de inicio de sesión de Taxi Lagos Courier S.A. ..... 5
**Figura 2.** Panel de Control con estadísticas de somnolencia en tiempo real ..... 7
**Figura 3.** Pantalla de Gestión de Conductores ..... 10
**Figura 4.** Formulario de registro de nuevo conductor ..... 11
**Figura 5.** Pantalla de Gestión de Vehículos ..... 12
**Figura 6.** Formulario de registro de nuevo vehículo ..... 13
**Figura 7.** Pantalla de Gestión de Rutas ..... 14
**Figura 8.** Formulario de creación de nueva ruta ..... 14
**Figura 9.** Pantalla de Asignación de Conductores a Rutas ..... 15
**Figura 10.** Panel de Estado del Conductor (activo En Ruta) ..... 16
**Figura 11.** Panel de Conductor Pendiente (ausencia temporal) ..... 16
**Figura 12.** Panel de Operación Finalizada con opciones de reasignación ..... 17
**Figura 13.** Pantalla de Generación de Reportes ..... 19
**Figura 14.** Vista del reporte PDF generado con eventos de somnolencia ..... 20

## 1. Introducción

El presente documento es el Manual de Usuario oficial del Sistema de Monitoreo de Somnolencia de Taxi Lagos Courier S.A. Su propósito es guiar al administrador en el uso correcto de todas las funcionalidades disponibles en la plataforma web.

Este sistema permite monitorear en tiempo real los eventos de somnolencia detectados en los conductores (ojos cerrados, bostezos, cabeceos) y gestionar la flota vehicular, las rutas y las asignaciones de conductores desde un panel de control centralizado.

### 1.1. Objetivo del Sistema

Proporcionar una herramienta tecnológica que permita a la empresa Taxi Lagos Courier S.A. prevenir accidentes de tránsito causados por la somnolencia de sus conductores, a través del monitoreo continuo, la generación de estadísticas y la descarga de reportes detallados.

### 1.2. Requerimientos Mínimos

Para utilizar el sistema correctamente se requiere contar con los siguientes elementos:

**Tabla 46**
*Requerimientos mínimos del sistema.*

| Requerimiento | Detalle |
|---|---|
| **Dispositivo** | Computadora, laptop, Tablet o dispositivo móvil |
| **Conexión** | Acceso a internet estable |
| **Navegador** | Google Chrome, Mozilla Firefox, Microsoft Edge o Brave (versiones actualizadas) |
| **Credenciales** | Correo electrónico y contraseña asignados por el administrador del sistema |

## 2. Acceso al Sistema (Inicio de Sesión)

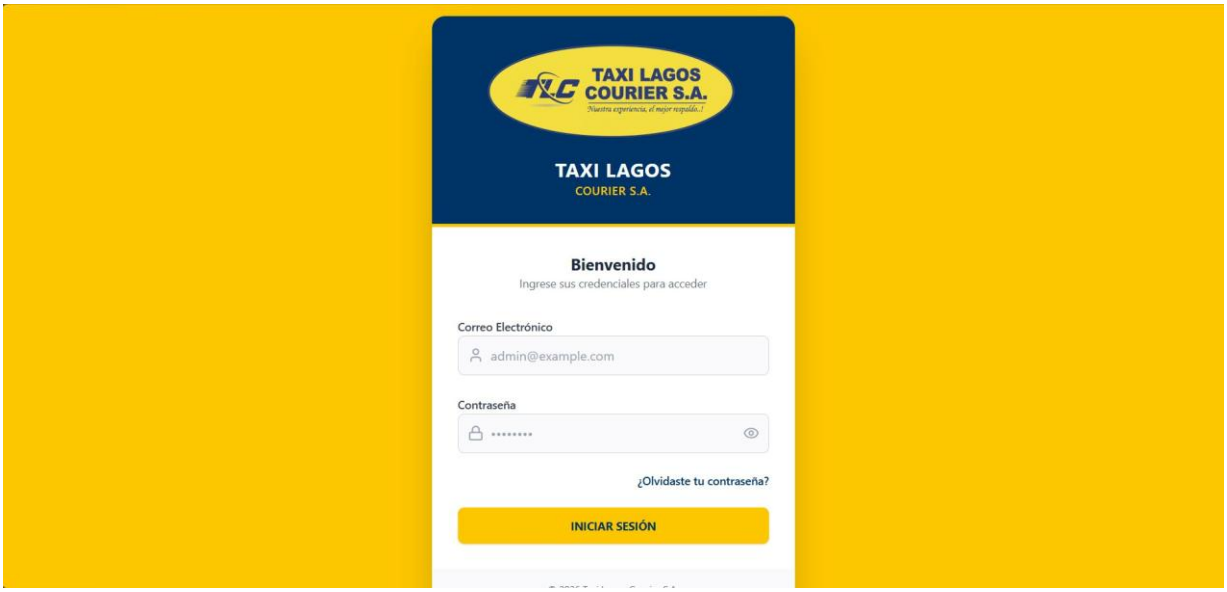
Para ingresar al sistema, abra su navegador web e ingrese la siguiente dirección:

**www.taxilagoscourier.com**

Se mostrará la pantalla de inicio de sesión con el formulario de autenticación de Taxi Lagos Courier S.A.

### 2.1. Pantalla de Inicio de Sesión

**Figura 46**
*Pantalla de inicio de sesión de Taxi Lagos Courier S.A.*




### 2.2. Procedimiento de Ingreso

Siga los siguientes pasos para acceder al sistema:

| | |
|---|---|
| **1** | **Correo Electrónico:** Ingrese su correo electrónico en el campo correspondiente (ej.: admin@taxilagoscourier.com). |
| **2** | **Contraseña:** Ingrese su contraseña. Puede pulsar el ícono del ojo para mostrar u ocultar el texto. |
| **3** | **Iniciar Sesión:** Haga clic en el botón amarillo "INICIAR SESIÓN" para acceder al sistema. |

**Nota:** Si olvidó su contraseña, haga clic en "¿Olvidaste tu contraseña?" para iniciar el proceso de recuperación.

### 2.3. Mensajes de Error

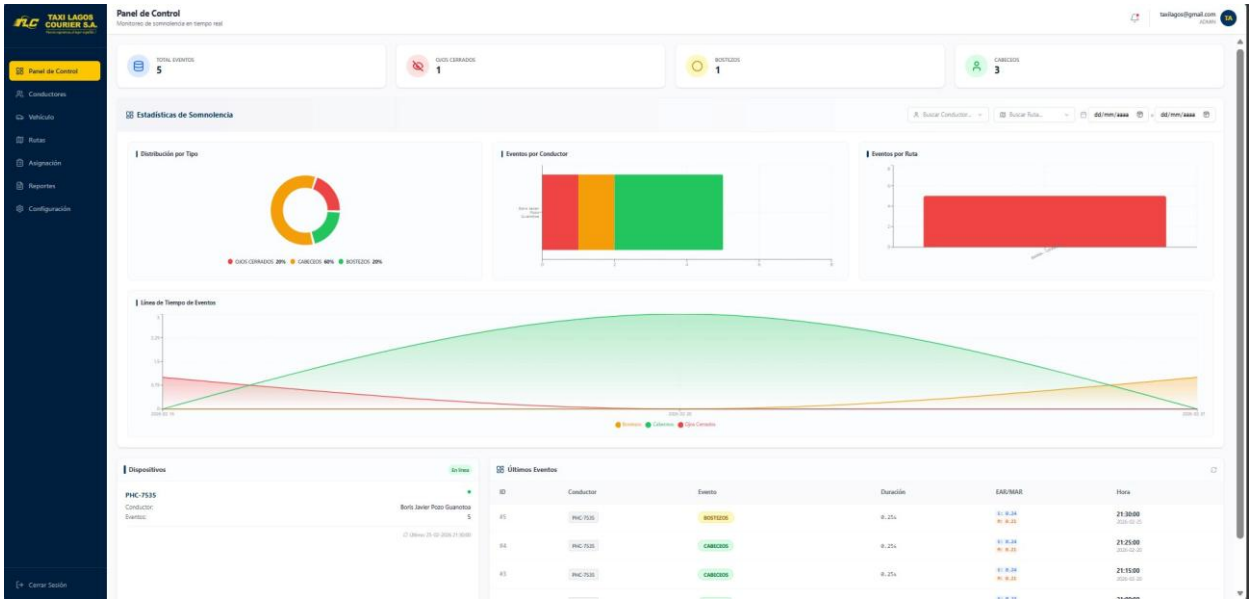
Si las credenciales son incorrectas o un campo está vacío, el sistema mostrará mensajes de validación. Verifique sus datos e intente nuevamente.

## 3. Panel de Control

Una vez autenticado, el sistema lo redirigirá al Panel de Control. Es el centro de operaciones y muestra el estado de los conductores en tiempo real.

### 3.1. Vista General del Panel

**Figura 47**
*Panel de Control con estadísticas de somnolencia en tiempo real.*




### 3.2. Tarjetas de Resumen

En la parte superior se muestran cuatro tarjetas con los contadores del período seleccionado:

**Tabla 47**
*Tarjetas de resumen del Panel de Control.*

| Total, Eventos | Ojos Cerrados | Bostezos | Cabeceos |
|---|---|---|---|
| Suma total de todos los eventos detectados. | Cierre prolongado de párpados. | Detección de bostezos. | Movimientos involuntarios de cabeza. |

### 3.3. Filtros de Estadísticas

Puede personalizar los gráficos aplicando filtros en la barra superior: Buscar Conductor, Buscar Ruta, y Fecha de Inicio / Fecha Fin (formato dd/mm/aaaa).

### 3.4. Gráficos de Estadísticas

Los gráficos disponibles en el panel son:

**Tabla 48**
*Gráficos de estadísticas de somnolencia.*

| # | Gráfico | Descripción |
|---|---|---|
| **1** | **Distribución por Tipo** | Gráfico de dona: porcentaje de cada tipo de evento (Ojos Cerrados, Bostezos, Cabeceos). |
| **2** | **Eventos por Conductor** | Barras apiladas: comparativa de eventos entre conductores. |
| **3** | **Eventos por Ruta** | Barras horizontales: incidencia de eventos por ruta. |
| **4** | **Línea de Tiempo** | Área temporal: evolución de los tres tipos de eventos en el rango de fechas. |

### 3.5. Dispositivos y Últimos Eventos

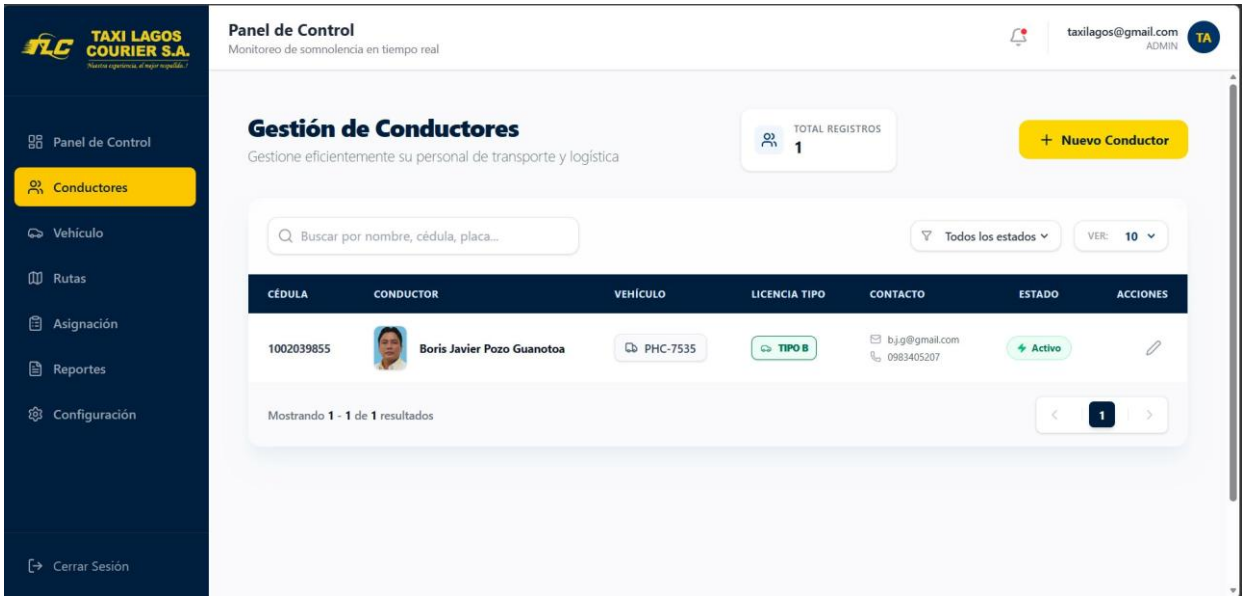
En la parte inferior del panel se muestran el listado de Dispositivos conectados en línea y la tabla de Últimos Eventos con los registros más recientes de somnolencia detectada.

## 4. Gestión de Conductores

Esta sección permite administrar el personal de conductores. Acceda desde el menú lateral haciendo clic en "Conductores".

### 4.1. Lista de Conductores

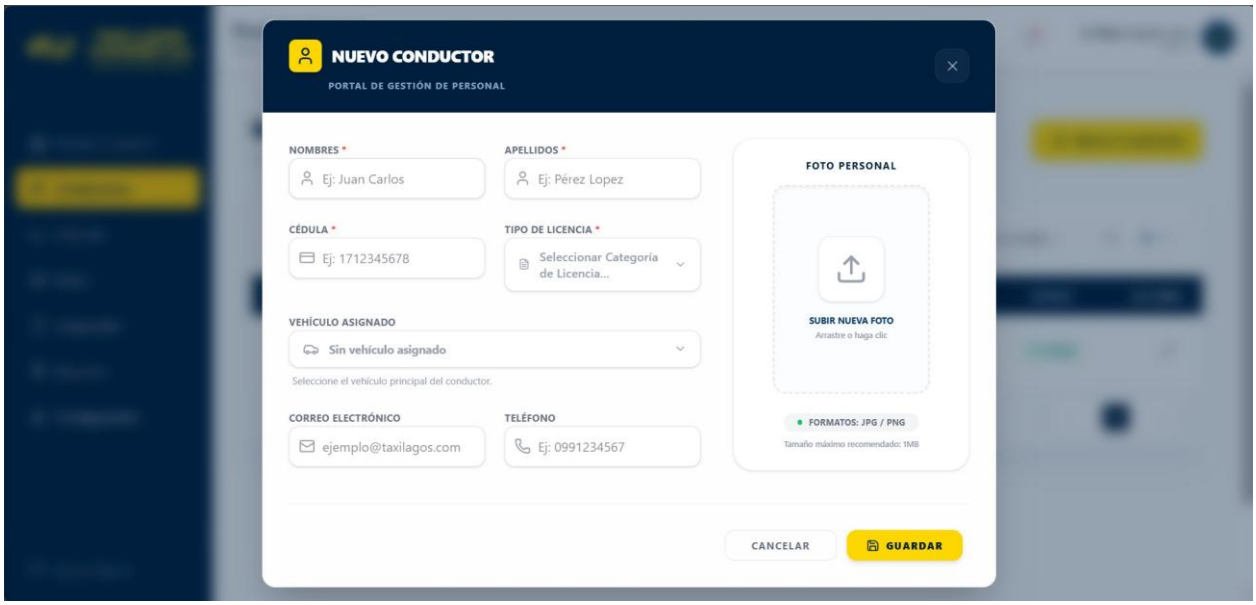
**Figura 48**
*Pantalla de Gestión de Conductores.*




La tabla muestra: Cédula, Conductor (nombre y foto), Vehículo asignado, Licencia Tipo, Contacto, Estado y Acciones. Puede filtrar por nombre, cédula, placa y estado.

### 4.2. Registrar Nuevo Conductor

**Figura 49**
*Formulario de registro de nuevo conductor.*




Haga clic en "+ Nuevo Conductor" y complete el formulario:

| | |
|---|---|
| **1** | **Nombres y Apellidos:** Ingrese el nombre y apellidos completos del conductor. |
| **2** | **Cédula y Licencia:** Ingrese el número de cédula y seleccione el tipo de licencia (Tipo B, C, D, etc.). |
| **3** | **Vehículo Asignado:** Seleccione el vehículo principal desde el menú desplegable. |
| **4** | **Datos de contacto:** Ingrese el correo electrónico y el número de teléfono. |
| **5** | **Foto Personal:** Suba una fotografía JPG o PNG (máx. 1 MB) en la zona de carga. |
| **6** | **Guardar:** Haga clic en "GUARDAR" para registrar el conductor. |

**Nota:** Todos los campos marcados con asterisco (*) son obligatorios.

### 4.3. Editar Conductor

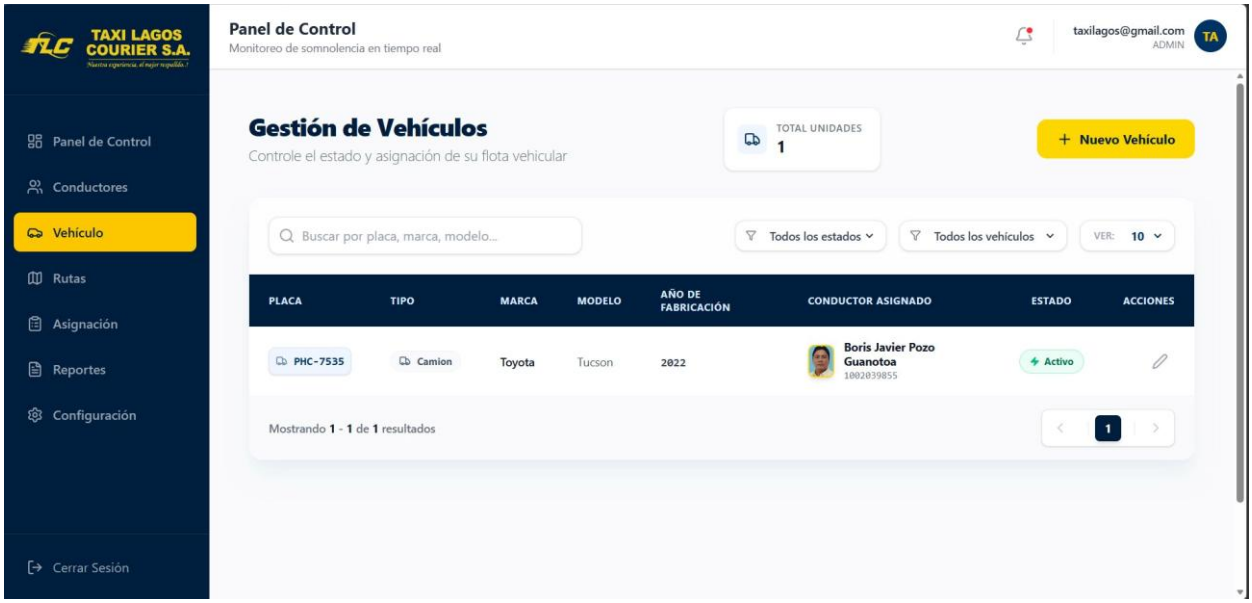
Para modificar datos de un conductor existente, haga clic en el ícono de lápiz en la columna "Acciones".

## 5. Gestión de Vehículos

Esta sección permite administrar la flota vehicular. Acceda desde el menú lateral haciendo clic en "Vehículo".

### 5.1. Lista de Vehículos

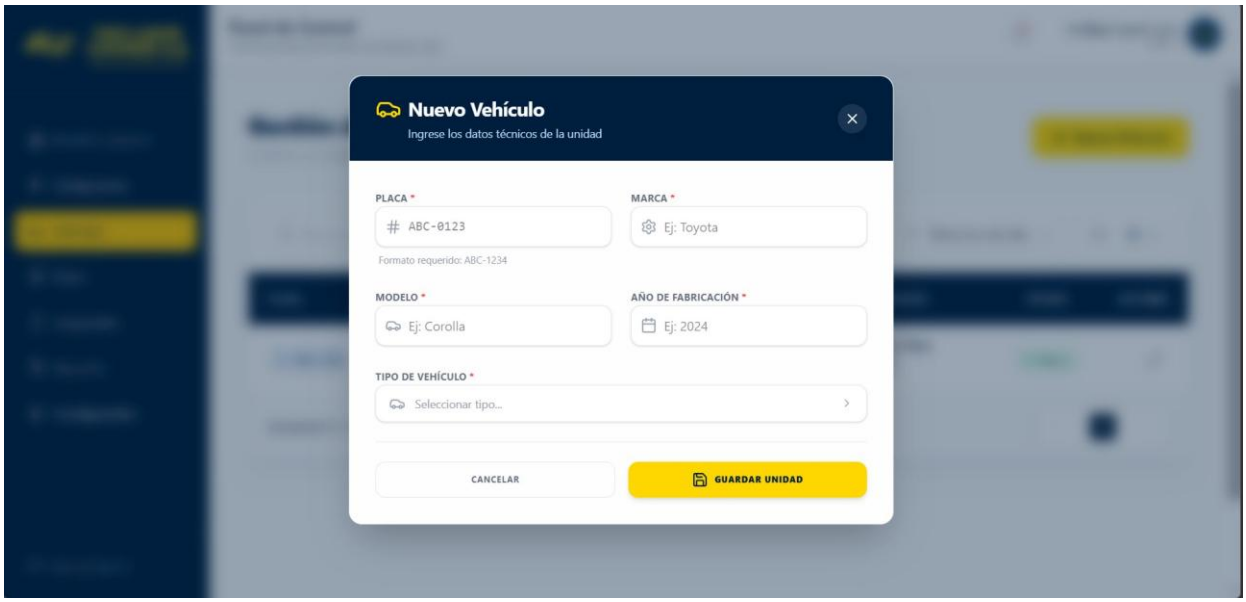
**Figura 50**
*Pantalla de Gestión de Vehículos*




La tabla muestra: Placa, Tipo, Marca, Modelo, Año de Fabricación, Conductor Asignado, Estado y Acciones.

### 5.2. Registrar Nuevo Vehículo

**Figura 51**
*Formulario de registro de nuevo vehículo.*




Haga clic en "+ Nuevo Vehículo" y complete el formulario:

| | |
|---|---|
| 1 | **Placa:** Ingrese la placa en formato estándar ecuatoriano (ej.: ABC-1234). |
| 2 | **Marca y Modelo:** Ingrese la marca (ej.: Toyota) y el modelo (ej.: Corolla). |
| 3 | **Año de Fabricación:** Ingrese el año de fabricación (ej.: 2024). |
| 4 | **Tipo de Vehículo:** Seleccione el tipo desde el menú desplegable. |
| 5 | **Guardar:** Haga clic en "GUARDAR UNIDAD" para registrar el vehículo. |

### 5.3. Editar Vehículo

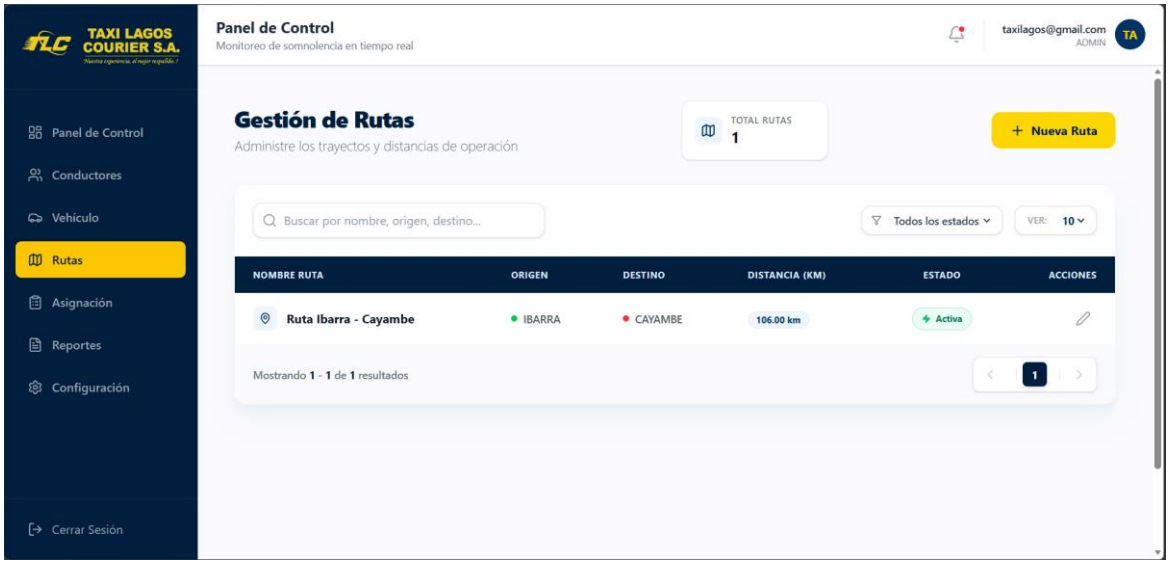
Para modificar un vehículo, haga clic en el ícono de lápiz en la columna "Acciones".

## 6. Gestión de Rutas

Esta sección permite registrar y administrar las rutas de operación. Acceda desde el menú lateral haciendo clic en "Rutas".

### 6.1. Lista de Rutas

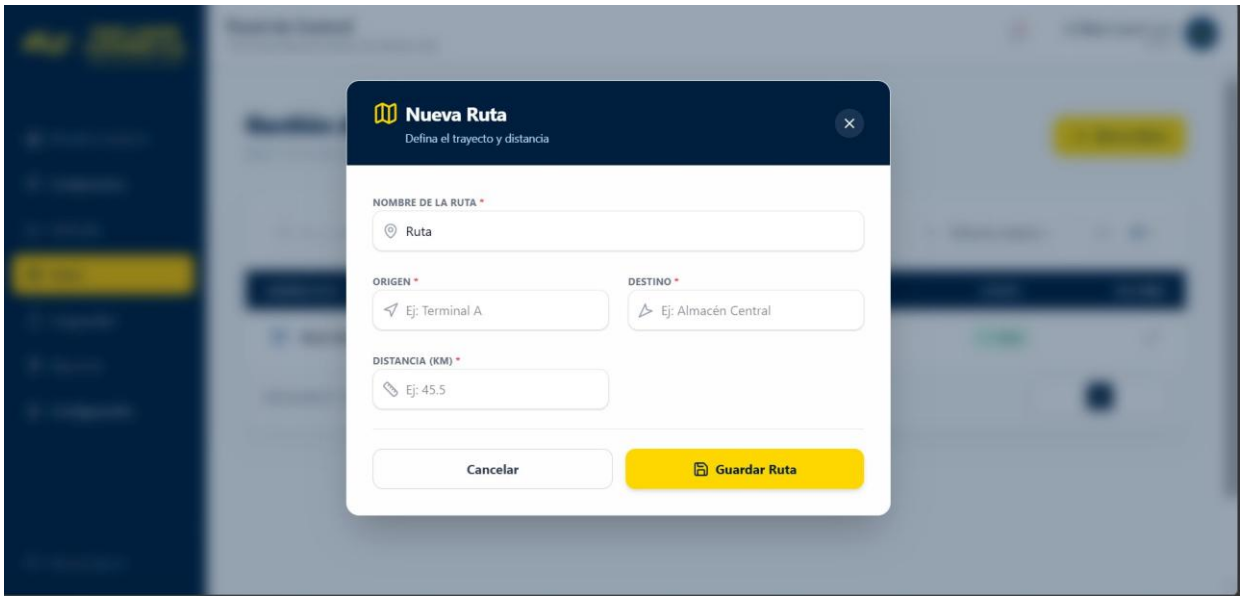
**Figura 52**
*Pantalla de Gestión de Rutas.*




La tabla muestra: Nombre Ruta, Origen, Destino, Distancia (km), Estado y Acciones. Filtre por nombre, origen, destino y estado.

## 6.2. Crear Nueva Ruta

**Figura 53**
*Formulario de creación de nueva ruta.*




Haga clic en "+ Nueva Ruta" y complete el formulario:

| | |
|---|---|
| 1 | **Nombre de la Ruta:** Ingrese un nombre descriptivo (ej.: Ruta Ibarra - Cayambe). |

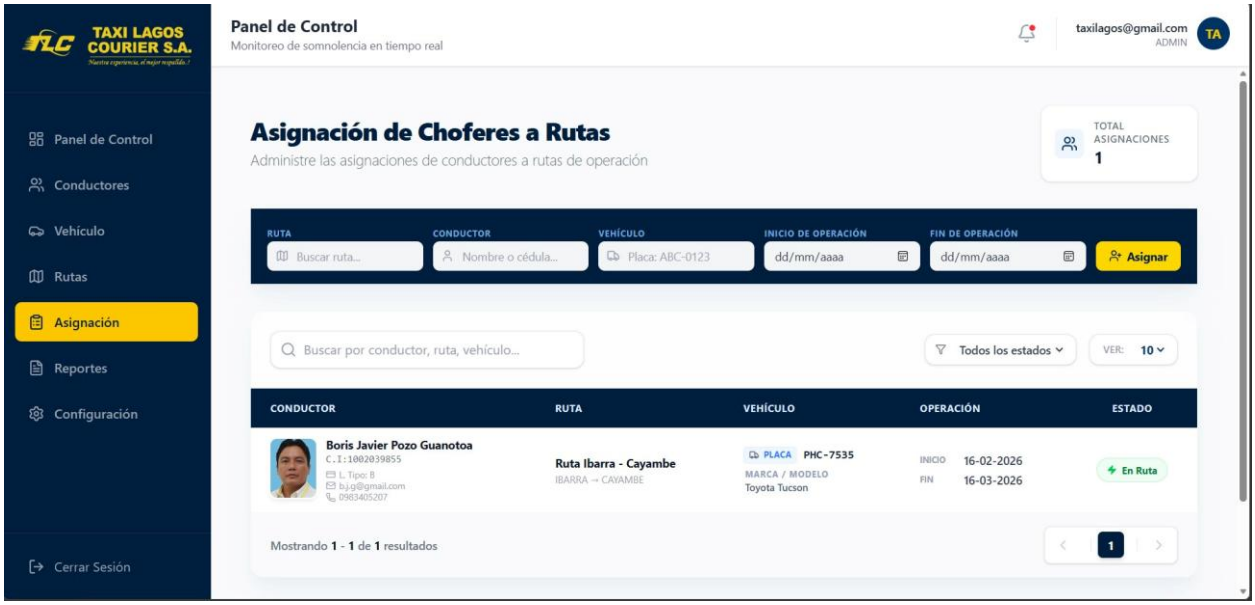
| | |
|---|---|
| 2 | **Origen y Destino:** Ingrese el punto de origen y de destino. |
| 3 | **Distancia:** Ingrese la distancia en kilómetros (ej.: 45.5). |
| 4 | **Guardar:** Haga clic en "Guardar Ruta" para registrar la ruta. |

## 7. Asignación de Conductores a Rutas

La sección de Asignación vincula un conductor con una ruta por un período determinado. Acceda desde el menú lateral haciendo clic en "Asignación".

### 7.1. Pantalla de Asignaciones

**Figura 54**
*Pantalla de Asignación de Conductores a Rutas.*




### 7.2. Crear una Nueva Asignación

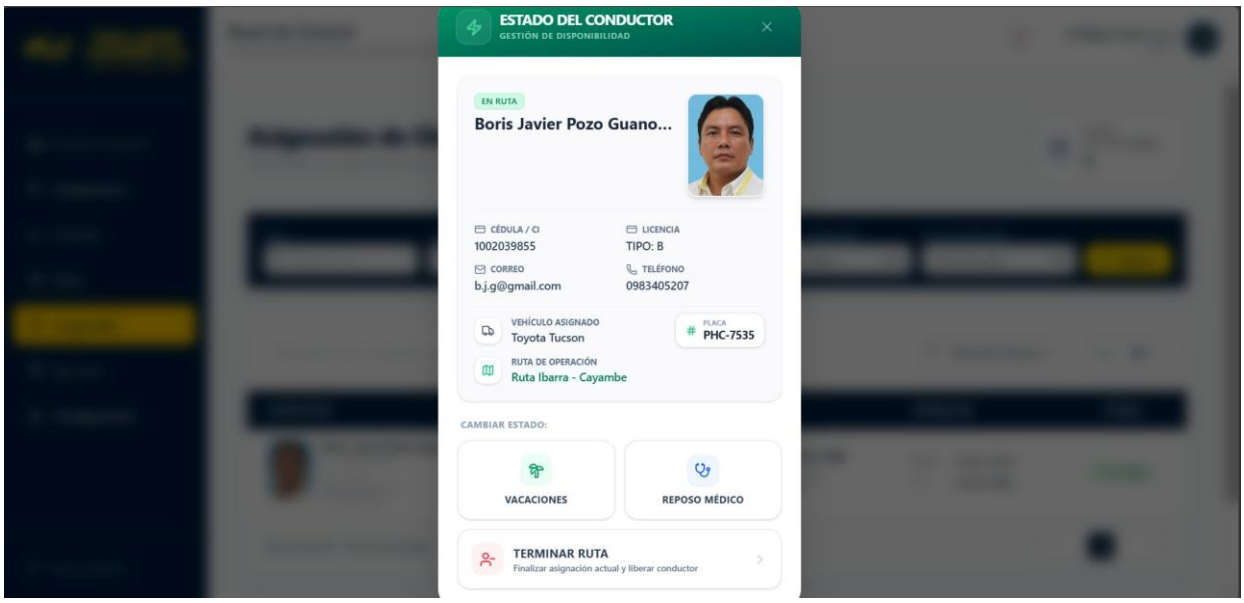
Complete el formulario en la parte superior de la pantalla:

| | |
|---|---|
| **1** | **Seleccionar Ruta:** Busque y seleccione la ruta de operación. |
| **2** | **Seleccionar Conductor:** Ingrese el nombre o cédula del conductor. |
| **3** | **Seleccionar Vehículo:** Ingrese la placa del vehículo. |
| **4** | **Definir Período:** Ingrese la fecha de Inicio y Fin de la operación (dd/mm/aaaa). |
| **5** | **Asignar:** Haga clic en "Asignar". El conductor aparecerá con estado "En Ruta". |

### 7.3. Gestión del Estado del Conductor´}4

**Sdfgio7 m**

**Figura 55**
*Panel de Estado del Conductor (activo En Ruta).*



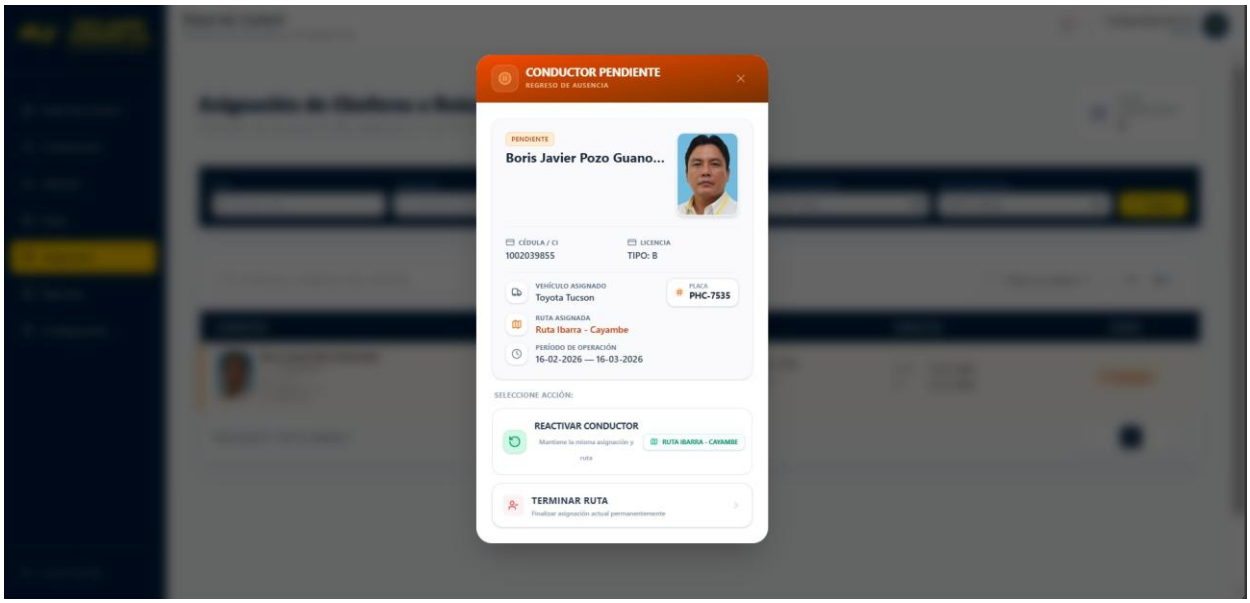

Haga clic sobre el conductor en la lista para abrir el panel "Estado del Conductor":

**Tabla 49**
*Opciones de cambio de estado del conductor*.

| Estado / Acción | Descripción |
|---|---|
| **En Ruta (activo)** | El conductor está operando en su ruta asignada. |
| **Vacaciones** | Ausencia por vacaciones. La asignación queda suspendida temporalmente. |
| **Reposo Médico** | Ausencia por salud. La asignación queda suspendida temporalmente. |
| **Terminar Ruta** | Finaliza permanentemente la asignación y libera al conductor. |

### 7.4. Conductor Pendiente — Reactivación

**Figura 56**
*Panel de Conductor Pendiente (ausencia temporal).*

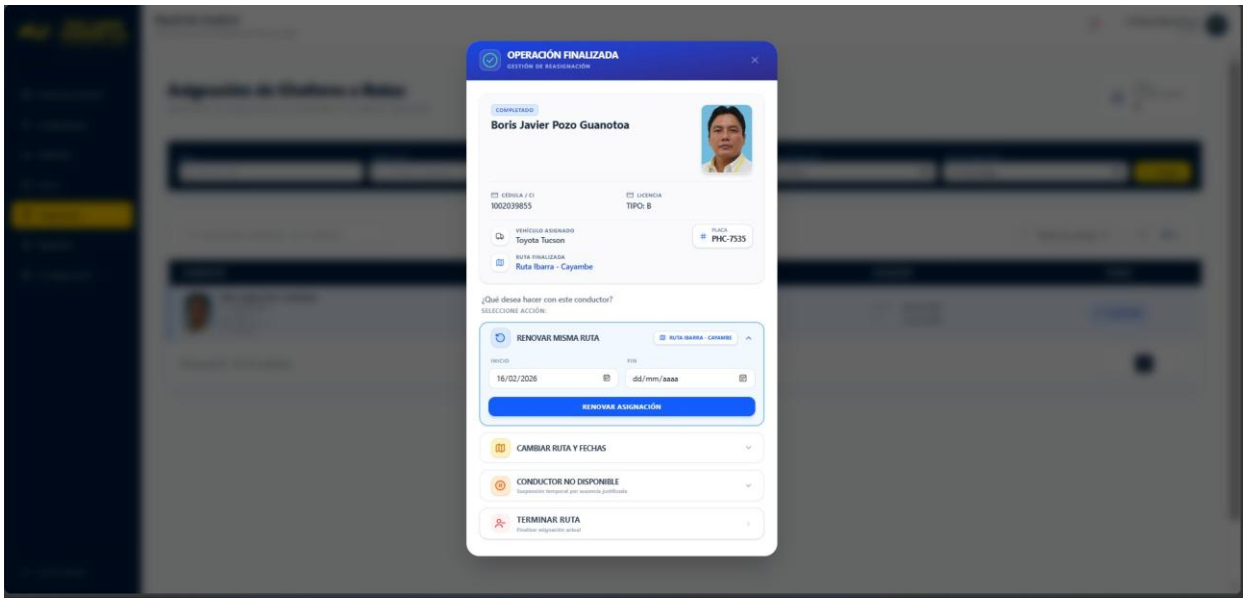



Haga clic sobre un conductor en estado "Pendiente" para ver las opciones disponibles:

| | |
|---|---|
| **1** | **Reactivar Conductor:** El conductor retoma su asignación con la misma ruta. El estado cambia a "En Ruta". |
| **2** | **Terminar Ruta:** Finaliza permanentemente la asignación. El conductor queda libre para una nueva asignación. |

### 7.5. Operación Finalizada — Gestión de Reasignación

**Figura 57**
*Panel de Operación Finalizada con opciones de reasignación.*




Al finalizar una asignación, el sistema muestra el panel "Operación Finalizada":

**Tabla 50**

*Opciones de gestión tras operación finalizada.*

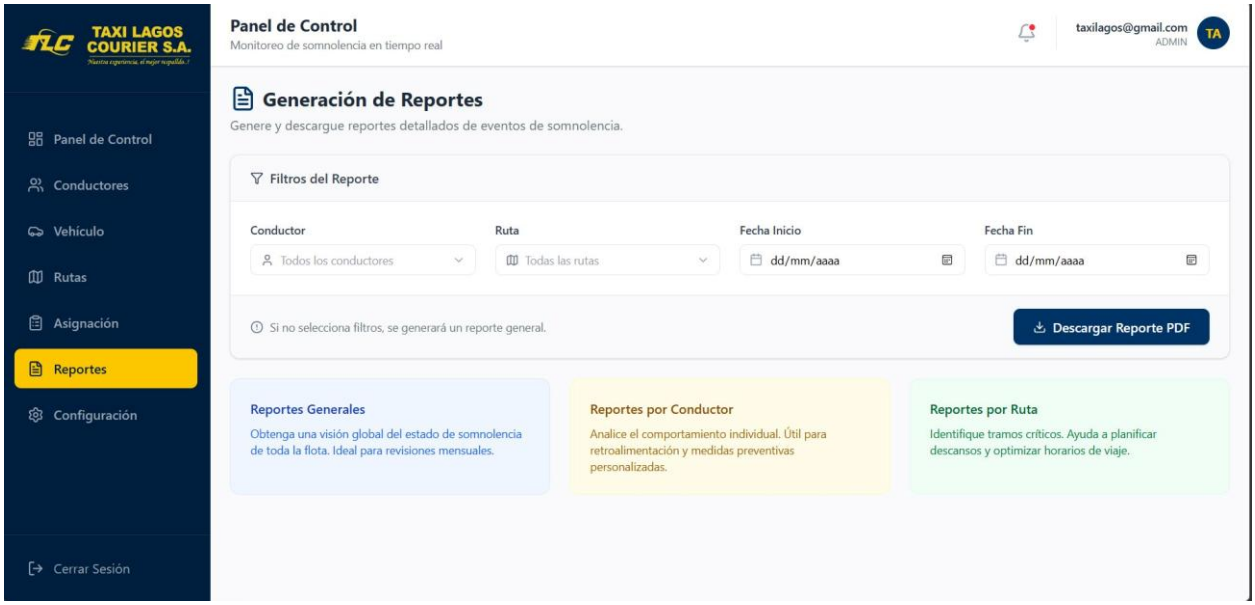
| Opción | Descripción |
|---|---|
| **Renovar Misma Ruta** | Renueva la asignación en la misma ruta con nuevas fechas. |
| **Cambiar Ruta y Fechas** | Asigna al conductor a una ruta diferente con nuevas fechas. |
| **Conductor No Disponible** | Suspensión temporal por ausencia justificada. |
| **Terminar Ruta** | Finaliza de forma definitiva la asignación del conductor. |

## 8. Generación de Reportes

La sección Reportes permite generar y descargar informes en PDF sobre los eventos de somnolencia. Acceda desde el menú lateral haciendo clic en "Reportes".

### 8.1. Pantalla de Reportes

**Figura 58**
*Pantalla de Generación de Reportes.*




### 8.2. Tipos de Reportes

**Tabla 51**
*Tipos de reportes disponibles.*

| Tipo de Reporte | Descripción |
|---|---|
| **Reportes Generales** | Visión global de toda la flota. Ideal para revisiones mensuales sin filtros. |
| **Reportes por Conductor** | Análisis individual de un conductor. Útil para medidas preventivas personalizadas. |
| **Reportes por Ruta** | Identifica tramos críticos. Ayuda a planificar descansos y optimizar horarios. |

### 8.3. Generar un Reporte PDF

Siga los pasos para generar y descargar un reporte:

| | |
|---|---|
| **1** | **Filtro por Conductor (opcional):** Seleccione un conductor o deje en "Todos los conductores". |
| **2** | **Filtro por Ruta (opcional):** Seleccione una ruta o deje en "Todas las rutas". |
| **3** | **Rango de Fechas (opcional):** Seleccione la fecha de inicio y fin. Sin fechas, se genera el historial completo. |
| **4** | **Descargar:** Haga clic en "Descargar Reporte PDF". El archivo se descargará automáticamente. |

### 8.4. Ejemplo de Reporte PDF Generado

**Figura 59**
*Vista del reporte PDF generado con eventos de somnolencia.*

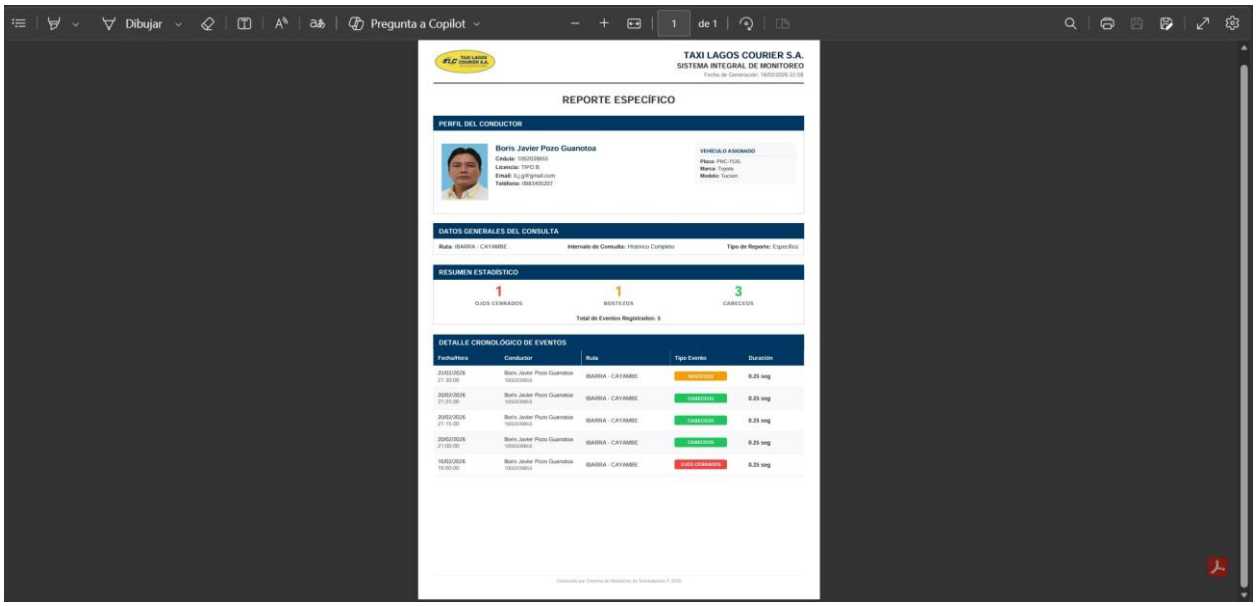



**Tabla 52**
*Contenido del reporte PDF generado.*

| Sección del PDF | Contenido |
|---|---|
| **Encabezado** | Logo corporativo, nombre del sistema y fecha de generación. |
| **Perfil del Conductor** | Fotografía, nombre, cédula, licencia, correo y teléfono. |
| **Datos Generales** | Ruta, intervalo de consulta y tipo de reporte. |
| **Resumen Estadístico** | Contadores de Ojos Cerrados, Bostezos y Cabeceos. |
| **Detalle Cronológico** | Tabla con fecha, hora, conductor, ruta, tipo de evento y duración. |

El menú lateral izquierdo está siempre visible y permite acceder a todas las secciones del sistema:

**Tabla 53**
*Secciones del menú lateral del sistema.*

| # | Sección | Descripción |
|---|---|---|
| 1 | **Panel de Control** | Dashboard con estadísticas y gráficos en tiempo real. |
| 2 | **Conductores** | Gestión del personal: registro, edición y consulta. |
| 3 | **Vehículo** | Gestión de la flota vehicular: registro y asignación. |
| 4 | **Rutas** | Administración de rutas de operación. |
| 5 | **Asignación** | Asignación de conductores a rutas con períodos. |
| 6 | **Reportes** | Generación y descarga de reportes PDF de somnolencia. |
| 7 | **Configuración** | Ajustes generales del sistema. |

### 9.1. Cerrar Sesión

Para cerrar sesión de forma segura, haga clic en "Cerrar Sesión" en la parte inferior del menú lateral.

**Nota:** Se recomienda siempre cerrar sesión al terminar, especialmente en equipos compartidos.

## 10. Glosario de Términos

**Tabla 54**
*Glosario de términos técnicos.*

| Término | Definición |
|---|---|
| **EAR (Eye Aspect Ratio)** | Métrica que mide la apertura ocular. Valores bajos indican cierre de ojos. |
| **MAR (Mouth Aspect Ratio)** | Métrica que mide la apertura de la boca. Valores altos indican bostezos. |
| **Ojos Cerrados** | Evento detectado cuando el conductor mantiene los ojos cerrados sobre el umbral permitido. |
| **Bostezo** | Evento detectado ante apertura prolongada de la boca del conductor. |
| **Cabeceo** | Evento detectado ante movimientos involuntarios de cabeza del conductor. |
| **En Ruta** | Estado que indica que el conductor está operando activamente en su ruta asignada. |
| **Pendiente** | Estado de ausencia temporal justificada (vacaciones o reposo médico). |
| **Completado** | Estado de asignación cuyo período de operación ha finalizado. |
| **Asignación** | Vinculación formal de conductor, ruta y vehículo por un período específico. |
| **Dispositivo** | Hardware de monitoreo instalado en el vehículo que detecta eventos de somnolencia. |

Si experimenta inconvenientes con el sistema, comuníquese con el administrador del sistema de su empresa.

# Anexo 8.

Manual Técnico

# TAXI LAGOS COURIER S.A.

# MANUAL TECNICO DEL SISTEMA DE DETECCION DE SOMNOLENCIA EN TIEMPO REAL

**Autores:**

Aldair German Muyulema Yumiseba

Anthony Paul Pozo López

**Fecha:**

17 enero 2026

**Manual Técnico.**

**Instalación del Sistema Operativo en la Raspberry pi 4**

**Tabla 55**

*Requisitos Previos de Hardware.*

| **Componentes** | **Especificaciones** |
|---|---|
| **Tarjeta MicroSD** | Tiene que ser mínimo de 16GB de almacenamiento, clase 10 para garantizar una buena velocidad de procesamiento de datos |
| **Lector de tarjeta MicroSD** | Muy necesario para poder conectar la MicroSD a la computadora y grabar el Sistema operativo |
| **Placa Raspberry Pi4** | Es el cerebro del proyecto ya que es como usar una mini computadora |
| **Conexión a Internet** | Debe ser una conexión estable para poder descargar los paquetes necesarios, actualizaciones o realizar configuraciones |

**Preparación de la imagen para la instalación**

El instalar un sistema operativo dentro del microrordenador Raspberry pi es diferente a lo que comúnmente se utiliza (Windows, Linux, Unix). Para ello se necesita emplear un equipo aparte del que se está utilizado, el mismo que debe tener una entrada de lectura para tarjetas SD o a su vez usar un adaptador de dicha memoria para poder conectarla al equipo ya sea PC o Laptop.

1. Descargar e instalar *Raspberry Pi imagen* desde el sitio oficial de Raspberry Pi. https://www.raspberrypi.com/software/
2. En la Interfaz de la aplicación seleccionar el modelo de Raspberry Pi en este caso el Modelo 4

**Figura 60**

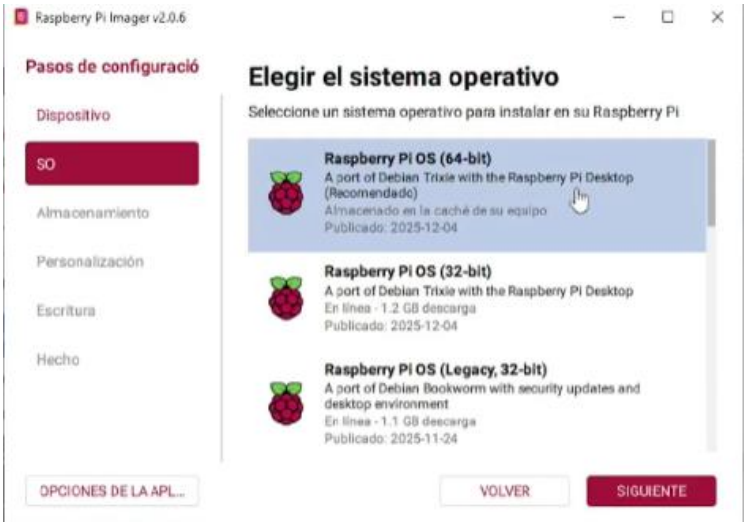
*Interfaz de la Raspberry Pi Imagen 1.*




3. Se debe elegir "Raspberry Pi OS (64-bit)". Para un entorno de producción optimizado en el vehículo, se recomienda la versión Lite (sin interfaz gráfica) para ahorrar recursos de CPU.

**Figura 61**

*Interfaz de la Raspberry Pi Imagen 2.*




4. Insertar la MicroSD en el computador y seleccionarla como destino.

**Figura 62**

*Interfaz de la Raspberry Pi Imagen 3.*




**Pre- configuración.**

Primero antes de realizar el flash hay que acceder al menú de configuraciones

1. En hostname debemos definir un nombre de red para el dispositivo
2. Establecer las credenciales del administrador este paso es indispensable
3. Ingresar las credenciales de una red para que la placa se pueda conectar automáticamente
4. Y el paso más fundamental activar el protocolo de comunicación (SSH) para poder gestionar el sistema de forma remota

**Figura 63**

*Interfaz de la Raspberry Pi Imagen 4.*




**Configuración Inicial del Sistema.**

Una vez que la placa haya encendido se debe acceder mediante una terminal con el protocolo SSH o puede ser directamente con los periféricos para poder ejecutar los comandos de mantenimiento.

1. Actualización de librerías y repositorios.
   - sudo apt update
   - sudo apt upgrade -y

   Estos dos comandos aseguran que el dispositivo se encuentre actualizado
2. Activación de la Cámara y los puertos GPIO

   sudo raspi-config

   Con este comando podemos acceder a las herramientas de configuración del sistema. Luego, Navegar hasta Interface Options y habilitar el soporte para la cámara (Legacy Camera support si es necesario) y el puerto I2C/GPIO para el funcionamiento del buzzer.
3. Instalación de Dependencias y librerías

   Para que el sistema funcione correctamente debemos asegurarnos primero que todas las librerías necesarias estén instaladas, se deben instalar las librerías base de Python y compiladores:
   - sudo apt install python3-pip cmake libopenblas-dev liblapack-dev libx11-dev libgtk-3-dev -y
4. Instalación del modelo.

- pip install dlib

Este proceso puede demorar entre 20 a 40 minutos en la Raspberry Pi 4. Es normal que la temperatura de la placa aumente

**Verificación de la Instalación Correcta.**

Para verificar que todo esté funcionando antes de proceder a cargar el programa se debe realizar pruebas para verificar que el hardware y software estén sincronizados con los siguientes comandos

**Tabla 56**

*Tabla de comando para validaciones.*

| **Prueba de Validación** | **Comando** |
|---|---|
| Cámara | libcamera-hello |
| Librerías All | python -c "import dlib; import cv2; print('OK')" |
| Alarma (Buzzer) | Ejecución de script test_buzzer.py |
| Conectividad | ping -c 4 [url_servidor_laravel] |

**Despliegue y carga del proyecto en el Dispositivo (Raspberry Pi 4)**

Una vez que se haya configurado el dispositivo e instalado sus librerías base, podemos proceder a la transferencia de los códigos fuente los modelos IA y configuración de la base de datos local.

1. **Transferencia de Archivos mediante el protocolo SCP**

   Desde la estación de trabajo del desarrollador, se deben enviar los archivos del proyecto a la Raspberry Pi utilizando la red local o una conexión directa.

   Hay que preparar los paquetes asegurándonos que la carpeta del proyecto contenga

   - main.py (Script principal de detección).
   - utils/ (Clases de MAR, EAR y Sincronizador).
   - shape_predictor_68_face_landmarks.dat (Modelo de dlib).

- config.json (Credenciales de la API de Laravel)

2. **Comando para proceder con la transferencia de archivos**

   Para comenzar con la transferencia hay que tener en cuenta que la carpeta donde se encuentra alojado nuestro sistema local debe estar en formato .zip

   - scp-r./proyecto_somnolenciausuario@taxilagos-pi01.local:/home/usuario/

3. **Automatización del Arranque Autostart.**

   Para que el Conductor (Actor Operativo) no tenga que ingresar comandos manualmente cada vez que encienda el vehículo, el sistema debe arrancar automáticamente al recibir energía.

   - Crear el archivo de configuración: sudo nano /etc/systemd/system/somnolencia.service
   - Configuración del servicio:
     [Unit] Description=Sistema de Detección de Somnolencia Taxi Lagos After=network.target [Service] ExecStart=/home/usuario/env_somnolencia/bin/python /home/usuario/proyecto/main.py WorkingDirectory=/home/usuario/proyecto StandardOutput=inherit StandardError=inherit Restart=always User=usuario [Install] WantedBy=multi-user.target

4. **Pruebas de Carga y Sincronización con el Administrador**

Una vez cargado el proyecto, se realiza la validación final del flujo de datos hacia el Administrador del Sistema:

Verificación de Logs: Comprobar que el sistema está capturando frames correctamente:

- journalctl -u somnolencia.service -f

Prueba de Envío: Forzar un bostezo frente a la cámara y verificar en el Dashboard del Administrador si el evento aparece en el historial de los conductores (Boris, Edwin o Jose Luis).

Prueba Offline: Desconectar el adaptador de red de la Raspberry Pi, generar un evento de cierre de ojos y verificar que la base de datos haya registrado el evento con la columna sincronizado

**Instalación y Configuración del Entorno Laravel y React**

**Tabla 57**

*Requerimientos Previos.*

| **Software** | **Versión mínima** | **Observación** |
|---|---|---|
| PHP | 8.2+ | Con extensiones: pdo_pgsql, mbstring, openssl, tokenizer |
| Composer | 2.6+ | Gestor de dependencias PHP |
| Node.js | 18.x+ | Incluye npm |
| npm | 9.x+ | Gestor de paquetes JavaScript |
| PostgreSQL | 15+ | Motor de base de datos |
| Git | 2.x+ | Control de versiones |

**Clonar el Repositorio**

**Código 4**

**Comando para clonar el repositorio.**

```
# Clonar el repositorio
git clone https://github.com/BlassGO/sistema-evaluacion-entorno-aprendizaje.git
```

**Configuración del Backend (Laravel)**

1. **Instalación de dependencias PHP**

**Código 5**

*Comando para instalar composer.*

```
# Ingresar al directorio del backend
cd backend

# Instalar dependencias con Composer
composer install
```

2. **Configuración del archivo .env**

**Código 6**

*Archivo .env*

```
APP_NAME="Taxi Lagos Courier - Somnolencia"
APP_ENV=production
APP_KEY=base64:GENERADA_POR_ARTISAN
APP_DEBUG=false
APP_URL=https://api.taxilagoscourier.com

# Configuración de base de datos PostgreSQL
DB_CONNECTION=pgsql
DB_HOST=dpg-xxxxxxxx.render.com
DB_PORT=5432
DB_DATABASE=taxilagos_db
DB_USERNAME=taxilagos_user
DB_PASSWORD=su_password_seguro

# CORS — Permitir peticiones del frontend
SANCTUM_STATEFUL_DOMAINS=www.taxilagoscourier.com
SESSION_DOMAIN=.taxilagoscourier.com

# Configuración de correo (opcional)
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
```

### 3. Migración de la base de datos

**Código 7**

*Comando para migración y seed de la base de datos.*

```
# Ejecutar migraciones
php artisan migrate

# Ejecutar seeders (datos iniciales)
php artisan db:seed

# Limpiar caché de configuración
php artisan config:clear
```

```
php artisan cache:clear
php artisan route:clear
```

4. **Iniciar el servidor de desarrollo**

**Código 8**

*Comando para iniciar el servidor.*

```
# Iniciar servidor local en el puerto 8000
php artisan serve

# El backend estará disponible en:
# http://localhost:8000
```

**Configuración del Frontend (React)**

1. **Instalación de dependencias Node.js**

**Código 9**

*Comando para instalar npm.*

```
# Ingresar al directorio del frontend
cd ../frontend

# Instalar dependencias con npm
npm install
```

2. **Variables de entorno del frontend**

**Código 10**

*Archivo .env*

```
# URL del backend (API Laravel)
VITE_API_URL=https://api.taxilagoscourier.com

# Nombre de la aplicación
VITE_APP_NAME="Taxi Lagos - Somnolencia"
```

3. **Iniciar el servidor de desarrollo**

**Código 11**

*Comando para instalar el servidor de desarrollo.*

```
# Iniciar servidor de desarrollo Vite
npm run dev

# El frontend estará disponible en:
# http://localhost:5173
```

**Despliegue en Render**

Render es la plataforma cloud utilizada para alojar el backend (Laravel), el frontend (React) y la base de datos (PostgreSQL). A continuación, se describen los pasos para el despliegue completo.

**Crear Cuenta y Proyecto en Render**

- Ingrese a https://render.com y cree una cuenta o inicie sesión.
- Conecte su cuenta de GitHub desde el panel de Render: Account Settings → Connect GitHub.
- Autorice el acceso al repositorio del proyecto.

**Despliegue del Backend (Web Service — Laravel)**

**Crear el Web Service**

- En el Dashboard de Render, haga clic en New → Web Service.
- Seleccione el repositorio de GitHub donde se encuentra el backend Laravel.
- Configure los siguientes parámetros:

**Tabla 58**

*Configuración de parámetros Web Service en Render.*

| | |
|---|---|
| **Name:** | taxilagos-backend |
| **Región:** | Oregon |
| **Branch:** | main |
| **Root Directory:** | backend |
| **Runtime:** | PHP |
| **Build Command:** | composer install --no-dev && php artisan config:cache && php artisan route:cache && php artisan migrate --force |
| **Start Command:** | php artisan serve --host=0.0.0.0 --port=$PORT |

**Despliegue del Frontend (Static Site — React)**

**Crear el Static Site**

- **En el Dashboard de Render, haga clic en New → Static Site.**
- **Seleccione el repositorio de GitHub del frontend React.**
- **Configure los parámetros:**

**Tabla 59**

*Configuración de parámetros Static Site en Render.*

| | |
|---|---|
| **Name:** | taxilagos-frontend |
| **Branch:** | main |
| **Root Directory:** | frontend |
| **Build Command:** | npm install && npm run build |
| **Publish Directory:** | dist |

**Base de Datos PostgreSQL en Render**

**Crear la base de datos**

- En el Dashboard de Render, haga clic en New → PostgreSQL.
- Configure los parámetros:

**Tabla 60**

*Configuración de parámetros PostgreSQL en Render.*

| | |
|---|---|
| **Name:** | taxilagos-db |
| **Database:** | taxilagos_db |
| **User:** | taxilagos_user |
| **Región:** | Oregon (US West) — igual que el backend |
| **Plan:** | Free (desarrollo) / Starter (producción) |

- Una vez creada, Render le proporcionará la cadena de conexión. Copie los valores de Host, Port, Database, Username y Password.
- Estos valores se utilizarán como variables de entorno en el Web Service del backend.

**Tabla 61**

*Interna vs Externa Database URL.*

| **Tipo de URL** | **Uso** |
|---|---|

| | |
|---|---|
| **Interna Database URL** | Usar cuando el backend y la DB están en el mismo proyecto Render. Más rápido y sin costo de transferencia. |
| **Externa Database URL** | Usar para conexiones desde herramientas externas como DBeaver, TablePlus odesde desarrollo local. |

### Configuración de Dominio en Cloudflare

Cloudflare actúa como intermediario entre los usuarios y los servidores en Render, proporcionando DNS, CDN, protección DDoS y certificado SSL/TLS gratuito.

### Agregar Dominio a Cloudflare

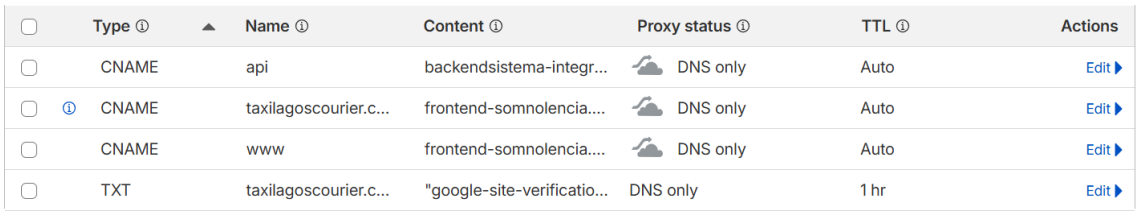
- Ingrese a https://cloudflare.com e inicie sesión en su cuenta.
- Haga clic en Add a Site e ingrese su dominio: taxilagoscourier.com
- Seleccione el plan gratuito (Free) y haga clic en Continue.
- Cloudflare escaneará automáticamente los registros DNS existentes del dominio.
- Al finalizar el escaneo, Cloudflare le proporcionará dos nameservers personalizados.
- Ingrese al panel de administración donde compró el dominio y reemplace los nameservers actuales con los de Cloudflare.

### Configuración de Registros DNS

Una vez activo el dominio en Cloudflare, configure los registros DNS desde DNS → Records:

Figura 64

Configuración de Registros DNS.



# Anexo 9.

Certificado Anti-plagio

**DR. HENRY FERNANDO VALLEJO BALLESTEROS EN CALIDAD DE DIRECTORA DEL TRABAJO DE INTEGRACIÓN CURRICULAR,**

# CERTIFICA

Que el trabajo de integración curricular denominado "**PROTOTIPO FUNCIONAL PARA LA DETECCIÓN DE SOMNOLENCIA EN CONDUCTORES MEDIANTE ANÁLISIS FACIAL EN TIEMPO REAL**", presentado por: ANTHONY PAUL POZO LÓPEZ y ALDAIR GERMAN MUYULEMA YUMISEBA estudiantes de la **carrera de Software** pasó el análisis de coincidencia no accidental en la herramienta COMPILATIO, reflejando un **porcentaje de similitud del 9%**, como se puede evidenciar en el documento adjunto.

Guaranda, 03 de marzo del 2026

Atentamente,

Firmado electrónicamente por:
HENRY FERNANDO VALLEJO BALLESTEROS
Validar únicamente con FirmaEC

_____
Dr. Henry Vallejo
**Director**

**Certificado de análisis**
Compilatio Magister+ | UNIVERSIDAD ESTATAL DE BOLIVAR (UEB)

TESIS PROYECTO VISION ARTIFICIAL

ID : db0a035fba385eb9943c062bf973098a5cd573a0

**9%**
Textos sospechosos

**Nombre del fichero** : TESIS PROYECTO VISION ARTIFICIAL.txt
**Tamaño del archivo original** : 5,61 MB
**Número de palabras** : 23.008
**Número de caracteres** : 159542

**Depositante** : HENRY FERNANDO VALLEJO BALLESTEROS
**Fecha de depósito** : 2 de marzo de 2026
**Tipo de carga** : interface
**fecha de fin de análisis** : 3 de marzo de 2026

## Resumen (sección 1/2)

Localización de los textos sospechosos en el documento :



### Incluido en el porcentaje de textos sospechosos :

**Similitudes** **2%**
Pasajes con similitudes a fuentes encontradas en diferentes colecciones.



**Detección de IA** **14%**
Textos estilísticamente próximos a un texto generado por una IA. Este índice es un indicador y no una prueba. Comprueba con el autor si domina los conocimientos mencionados en el documento.



**Idiomas no reconocidos** **7%**
Pasajes en los que parte del vocabulario utilizado no forma parte del diccionario de la lengua. Puede tratarse de un intento del autor de modificar el texto para evitar ser detectado.



### No incluido en el porcentaje de textos sospechosos :

**Textos entre comillas** **<1%**
Pasajes entre comillas, a menudo indicativos de una cita.

# Anexo 10.

Link del repositorio digital de biblioteca donde fue subido el proyecto