**📋 SIPAT-LARAVEL: DOCUMENTACIÓN DE LÓGICA DEL SISTEMA**

**🏗️ ARQUITECTURA GENERAL**

**Patrón Arquitectónico**

**MVC (Model-View-Controller)** con arquitectura en capas:

* **Modelos**: Eloquent ORM para abstracción de datos
* **Controladores**: Lógica de negocio y coordinación
* **Vistas**: Blade templates con Bootstrap 5
* **Servicios**: Lógica especializada reutilizable
* **Comandos**: Tareas automatizadas y programadas

**Estructura de Directorios Clave**

app/

├── Console/Commands/ # Comandos artisan personalizados

├── Http/Controllers/ # Controladores de aplicación

│ ├── Api/ # Controladores para API REST

│ └── ... # Controladores web

├── Models/ # Modelos Eloquent

├── Services/ # Servicios especializados

└── Imports/ # Importadores de datos

resources/views/ # Plantillas Blade

├── dashboard/ # Vistas del dashboard

├── conductores/ # Gestión de conductores

├── validaciones/ # Sistema de validaciones

├── notificaciones/ # Centro de notificaciones

├── reportes-avanzados/ # Reportes especializados

└── layouts/ # Layouts base

database/

├── migrations/ # Esquemas de base de datos

└── seeders/ # Datos iniciales

**🧠 LÓGICA DE NEGOCIO PRINCIPAL**

**1. GESTIÓN DE CONDUCTORES**

**Modelo Conductor (app/Models/Conductor.php)**

**Responsabilidades:**

* Almacenar información personal y profesional
* Calcular métricas de rendimiento
* Gestionar estados operativos
* Validar disponibilidad para turnos

**Estados del Conductor:**

DISPONIBLE → Puede ser asignado a turnos

DESCANSO → En periodo de descanso obligatorio/voluntario

VACACIONES → En periodo vacacional

SUSPENDIDO → Temporalmente inactivo

**Métricas Calculadas:**

* **dias\_acumulados**: Días consecutivos trabajados sin descanso
* **eficiencia**: Porcentaje de eficiencia operativa (0-100)
* **puntualidad**: Porcentaje de puntualidad (0-100)
* **score\_general**: Media de eficiencia y puntualidad

**Reglas de Negocio:**

1. Un conductor con ≥6 días acumulados debe ir a descanso
2. Eficiencia <80% genera validación de advertencia
3. Puntualidad <85% genera validación de seguimiento

**Controlador ConductorController**

**Flujo de Operaciones:**

1. **Creación**: Validación → Almacenamiento → Auditoría → Validaciones automáticas
2. **Actualización**: Validación → Cambios → Auditoría → Notificaciones
3. **Activación/Descanso**: Cambio de estado → Reset de métricas → Logs

**2. SISTEMA DE VALIDACIONES**

**Modelo Validacion (app/Models/Validacion.php)**

**Tipos de Validaciones:**

DESCANSO\_001 → Descanso obligatorio por días acumulados

EFICIENCIA\_002 → Baja eficiencia del conductor

PUNTUALIDAD\_003 → Baja puntualidad del conductor

TURNO\_VACIO\_004 → Turno sin asignar

DESCANSO\_PROGRAMADO → Descanso manual programado

**Severidades:**

INFO → Informativa, no requiere acción inmediata

ADVERTENCIA → Requiere atención, no crítica

CRITICA → Requiere acción inmediata

**Estados de Validación:**

PENDIENTE → Creada, esperando resolución

RESUELTO → Validación atendida y cerrada

IGNORADO → Validación descartada manualmente

**Lógica de Validaciones Automáticas**

**Comando:** php artisan sipat:validaciones

**Flujo de Ejecución:**

1. **Consulta de Conductores Críticos**:
2. Conductor::where('dias\_acumulados', '>=', 6)
3. ->where('estado', '!=', 'DESCANSO')
4. ->get();
5. **Verificación de Validaciones Existentes**:
   * Evita duplicar validaciones pendientes
   * Mantiene integridad del sistema
6. **Creación de Validaciones**:
   * Generación automática de mensaje descriptivo
   * Asignación de severidad según criticidad
   * Registro en auditoría
7. **Notificaciones**:
   * Creación automática de notificaciones
   * Categorización por tipo y severidad

**3. SISTEMA DE RUTAS CORTAS**

**Modelo RutaCorta (app/Models/RutaCorta.php)**

**Propósito**: Gestionar rutas de corta distancia con optimización de recursos

**Atributos Clave:**

* **codigo\_ruta**: Identificador único de la ruta
* **origen/destino**: Puntos de inicio y fin
* **duracion\_horas**: Duración estimada del recorrido
* **ingreso\_estimado**: Proyección de ingresos
* **estado**: PROGRAMADA, EN\_CURSO, COMPLETADA, CANCELADA
* **es\_consecutiva**: Indica si es parte de una secuencia

**Algoritmo de Asignación**:

1. Evaluar disponibilidad de conductores

2. Calcular compatibilidad (proximidad + experiencia)

3. Optimizar por eficiencia y puntualidad

4. Asignar conductor con mayor score

**Métricas de Optimización**:

* **Factor Proximidad (30%)**: Distancia entre origen del conductor y ruta
* **Factor Puntualidad (25%)**: Historial de puntualidad
* **Factor Eficiencia (25%)**: Rendimiento operativo
* **Factor Disponibilidad (20%)**: Disponibilidad horaria

**4. SISTEMA DE NOTIFICACIONES**

**Modelo Notificacion (app/Models/Notificacion.php)**

**Categorías de Notificaciones:**

VALIDACION → Relacionadas con validaciones del sistema

TURNO → Asignaciones y cambios de turnos

SISTEMA → Mantenimiento, backups, actualizaciones

**Ciclo de Vida**:

1. **Creación**: Automática desde validaciones o manual
2. **Distribución**: Mostrar en dashboard y centro de notificaciones
3. **Lectura**: Marcar como leída por usuario
4. **Limpieza**: Eliminación automática después de X días

**Integración con Validaciones**:

// Cada validación crítica genera notificación automática

Validacion::created(function($validacion) {

if ($validacion->severidad === 'CRITICA') {

Notificacion::crear(

'VALIDACION',

"Validación Crítica: {$validacion->tipo}",

$validacion->mensaje,

'CRITICA'

);

}

});

**5. SISTEMA DE AUDITORÍA**

**Modelo AuditoriaLog (app/Models/AuditoriaLog.php)**

**Eventos Auditados**:

* **CREATE**: Creación de registros
* **UPDATE**: Modificaciones
* **DELETE**: Eliminaciones
* **VIEW**: Accesos importantes

**Datos Capturados**:

- Usuario ejecutor

- Acción realizada

- Modelo afectado

- ID del registro

- Datos anteriores (UPDATE/DELETE)

- Datos nuevos (CREATE/UPDATE)

- IP y User Agent

- Timestamp preciso

**Integración Automática**:

// En cada modelo crítico

protected static function booted()

{

static::created(function ($model) {

AuditoriaLog::registrar('CREATE', $model, ...);

});

static::updated(function ($model) {

AuditoriaLog::registrar('UPDATE', $model, ...);

});

}

**6. SISTEMA DE REPORTES**

**Controlador ReporteAvanzadoController**

**Tipos de Reportes**:

1. **Reporte de Conductores**:
   * Análisis de rendimiento individual
   * Tendencias por período
   * Métricas comparativas
   * Identificación de conductores críticos
2. **Análisis de Validaciones**:
   * Distribución por tipo y severidad
   * Tiempo promedio de resolución
   * Tendencias diarias/semanales
   * Eficiencia del sistema de validaciones
3. **Eficiencia Operacional**:
   * KPIs principales del sistema
   * Cobertura de turnos
   * Ingresos por rutas cortas
   * Benchmarks de rendimiento
4. **Auditoría de Actividad**:
   * Logs de cambios por período
   * Actividad por hora/día
   * Análisis de patrones de uso

**🔄 FLUJOS DE PROCESOS CRÍTICOS**

**Flujo 1: Proceso de Validación Automática**

graph TD

A[Comando Scheduled] --> B[Consultar Conductores]

B --> C{¿Días >= 6?}

C -->|Sí| D[Verificar Validación Existente]

C -->|No| E[Evaluar Otras Métricas]

D --> F{¿Existe Pendiente?}

F -->|No| G[Crear Validación CRITICA]

F -->|Sí| H[Continuar con Siguiente]

G --> I[Crear Notificación]

I --> J[Registrar Auditoría]

E --> K[Fin del Proceso]

**Flujo 2: Asignación de Rutas Cortas**

graph TD

A[Nueva Ruta Corta] --> B[Obtener Conductores Disponibles]

B --> C[Calcular Score de Compatibilidad]

C --> D[Evaluar Proximidad Geográfica]

D --> E[Evaluar Eficiencia/Puntualidad]

E --> F[Calcular Score Final]

F --> G[Seleccionar Mejor Conductor]

G --> H[Asignar Ruta]

H --> I[Actualizar Estado Conductor]

I --> J[Registrar Auditoría]

**Flujo 3: Gestión de Estados de Conductor**

graph TD

A[Cambio de Estado] --> B{¿Tipo de Cambio?}

B -->|A DESCANSO| C[Reset Días Acumulados]

B -->|A DISPONIBLE| D[Verificar Condiciones]

B -->|A SUSPENDIDO| E[Crear Validación]

C --> F[Crear Validación Seguimiento]

D --> G{¿Cumple Requisitos?}

G -->|Sí| H[Actualizar Estado]

G -->|No| I[Mantener Estado Actual]

E --> F

F --> J[Notificar Cambio]

H --> J

I --> K[Crear Notificación Error]

**⚙️ CONFIGURACIÓN Y PARÁMETROS**

**Sistema de Configuración Avanzada**

**Modelo ConfiguracionSistema**

**Categorías de Configuración**:

1. **VALIDACIONES**:
   * dias\_maximos\_sin\_descanso: Control de descansos obligatorios
   * eficiencia\_minima\_conductor: Umbral de eficiencia
   * puntualidad\_minima\_conductor: Umbral de puntualidad
2. **ALERTAS**:
   * enviar\_notificaciones\_email: Activar/desactivar emails
   * hora\_ejecucion\_validaciones: Programación automática
   * retener\_notificaciones\_dias: Limpieza automática
3. **REPORTES**:
   * formato\_fecha\_reportes: Formato de fechas en reportes
   * generar\_reportes\_automaticos: Automatización
4. **GENERAL**:
   * nombre\_empresa: Branding en documentos
   * items\_por\_pagina: Paginación
   * zona\_horaria: Configuración temporal

**Método de Acceso**:

// Obtener valor de configuración

$valor = ConfiguracionSistema::obtenerValor('dias\_maximos\_sin\_descanso', 6);

// Establecer nuevo valor

ConfiguracionSistema::establecerValor('eficiencia\_minima\_conductor', 85);

**🔧 SERVICIOS ESPECIALIZADOS**

**CacheMetricasService**

**Propósito**: Optimizar consultas frecuentes mediante cache inteligente

**Métricas Cacheadas**:

* Dashboard principal (TTL: 5 minutos)
* Tendencias semanales (TTL: 1 hora)
* Conductores críticos (TTL: 15 minutos)

**Invalidación Automática**:

// Al modificar conductores o validaciones

CacheMetricasService::limpiarCacheMetricas();

**NotificacionService (Propuesto)**

**Responsabilidades**:

* Distribución inteligente de notificaciones
* Gestión de preferencias de usuario
* Integración con sistemas externos (email, SMS)

**📊 MÉTRICAS Y KPIs**

**KPIs Principales del Sistema**

1. **Cobertura de Turnos**:
2. (Conductores Disponibles / Turnos Requeridos) \* 100
3. **Eficiencia Promedio de Conductores**:
4. AVG(eficiencia) WHERE estado = 'DISPONIBLE'
5. **Tiempo Promedio de Resolución de Validaciones**:
6. AVG(DATEDIFF(fecha\_resolucion, created\_at)) WHERE estado = 'RESUELTO'
7. **Ingresos por Rutas Cortas**:
8. SUM(ingreso\_estimado) WHERE DATE(created\_at) = CURDATE()

**Métricas de Calidad del Sistema**

1. **Disponibilidad de Conductores**: % de conductores en estado DISPONIBLE
2. **Efectividad de Validaciones**: % de validaciones resueltas en <24h
3. **Satisfacción Operacional**: Score promedio de eficiencia + puntualidad
4. **Utilización de Rutas**: % de rutas completadas vs programadas

**🛡️ SEGURIDAD Y MANTENIMIENTO**

**Sistema de Backup Automático**

**Tipos de Backup**:

1. **Completo**: Estructura + datos (diario a las 2:00 AM)
2. **Datos**: Solo información (cada 6 horas)
3. **Estructura**: Solo esquemas (semanal)

**Comando de Backup**:

# Backup completo

php artisan sipat:backup --tipo=completo

# Solo datos

php artisan sipat:backup --tipo=datos

**Mantenimiento Preventivo**

**Tareas Automatizadas** (Domingos 3:00 AM):

1. Limpieza de notificaciones antiguas
2. Optimización de tablas de BD
3. Eliminación de logs de auditoría antiguos
4. Verificación de integridad del sistema
5. Limpieza de archivos temporales

**Comando de Mantenimiento**:

php artisan sipat:mantenimiento --forzar

**Logs de Auditoría y Trazabilidad**

**Retención de Datos**:

* Auditoría: 90 días
* Notificaciones: 30 días (configurable)
* Validaciones: Permanente
* Métricas: Permanente

**🔌 API REST**

**Endpoints Principales**

**Conductores**:

GET /api/v1/conductores → Lista paginada

POST /api/v1/conductores → Crear conductor

GET /api/v1/conductores/{id} → Detalle específico

PUT /api/v1/conductores/{id} → Actualizar

DELETE /api/v1/conductores/{id} → Eliminar

GET /api/v1/conductores-metricas → Métricas agregadas

**Validaciones**:

GET /api/v1/validaciones → Lista de validaciones

POST /api/v1/validaciones/ejecutar → Ejecutar validaciones manuales

**Sistema**:

GET /api/v1/sistema/estado → Estado del sistema

**Formato de Respuesta Estándar**

{

"success": true,

"data": { ... },

"meta": {

"total": 150,

"current\_page": 1,

"last\_page": 8

},

"message": "Operación exitosa"

}

**🎯 ALGORITMOS CLAVE**

**Algoritmo de Compatibilidad para Rutas**

function calcularCompatibilidad($conductor, $turno) {

$score = 0;

// Factor proximidad (30%)

if ($conductor->origen === $turno->origen\_conductor) {

$score += 30;

}

// Factor puntualidad (25%)

$score += ($conductor->puntualidad / 100) \* 25;

// Factor eficiencia (25%)

$score += ($conductor->eficiencia / 100) \* 25;

// Factor disponibilidad (20%)

if ($conductor->estaDisponiblePara($turno->hora\_salida)) {

$score += 20;

}

return round($score, 2);

}

**Algoritmo de Priorización de Validaciones**

function calcularPrioridadValidacion($validacion) {

$prioridad = 0;

// Severidad (40%)

$prioridad += match($validacion->severidad) {

'CRITICA' => 40,

'ADVERTENCIA' => 25,

'INFO' => 10

};

// Antiguedad (30%)

$diasPendiente = $validacion->created\_at->diffInDays(now());

$prioridad += min(30, $diasPendiente \* 5);

// Tipo de validación (30%)

$prioridad += match($validacion->tipo) {

'DESCANSO\_001' => 30,

'EFICIENCIA\_002' => 20,

'PUNTUALIDAD\_003' => 15,

default => 10

};

return min(100, $prioridad);

}

**📈 TENDENCIAS Y ANÁLISIS**

**Análisis Predictivo (Futuro)**

**Métricas Predictivas Planificadas**:

1. Predicción de necesidad de descansos
2. Proyección de eficiencia de conductores
3. Optimización automática de rutas
4. Detección temprana de patrones problemáticos

**Integración con Business Intelligence**

**Datos Exportables**:

* Métricas históricas de conductores
* Tendencias de validaciones
* Análisis de rutas más eficientes
* Patrones de utilización del sistema

**🔄 CICLO DE VIDA DEL SISTEMA**

**Flujo Diario Típico**

06:00 → Ejecución automática de validaciones

06:30 → Generación de notificaciones críticas

07:00 → Inicio de operaciones (asignación de turnos)

12:00 → Validaciones de medio día

18:00 → Análisis de rendimiento del día

02:00 → Backup automático completo

03:00 → Mantenimiento semanal (domingos)

**Estados del Sistema**

1. **OPERATIVO**: Funcionamiento normal
2. **MANTENIMIENTO**: Durante tareas de mantenimiento
3. **BACKUP**: Durante procesos de respaldo
4. **ALERTA**: Cuando hay validaciones críticas pendientes

**📋 CONCLUSIONES Y RECOMENDACIONES**

**Fortalezas del Sistema**

1. **Automatización Inteligente**: Validaciones y mantenimiento automáticos
2. **Trazabilidad Completa**: Auditoría de todas las operaciones críticas
3. **Escalabilidad**: Arquitectura preparada para crecimiento
4. **Flexibilidad**: Sistema de configuración robusto
5. **Monitoreo Proactivo**: Identificación temprana de problemas

**Áreas de Mejora Futuras**

1. **Machine Learning**: Predicción de patrones y optimización automática
2. **Integración Externa**: APIs con sistemas de terceros
3. **Mobile App**: Aplicación móvil para conductores
4. **Real-time Dashboard**: Dashboard en tiempo real con WebSockets
5. **Gamificación**: Sistema de puntuación y recompensas

**Mantenimiento Recomendado**

1. **Monitoreo Diario**: Revisión de validaciones críticas
2. **Análisis Semanal**: Review de métricas de rendimiento
3. **Optimización Mensual**: Ajuste de parámetros según tendencias
4. **Auditoría Trimestral**: Revisión completa del sistema
5. **Actualización Semestral**: Incorporación de nuevas funcionalidades

**Este documento debe ser revisado y actualizado cada vez que se implementen cambios significativos en el sistema.**