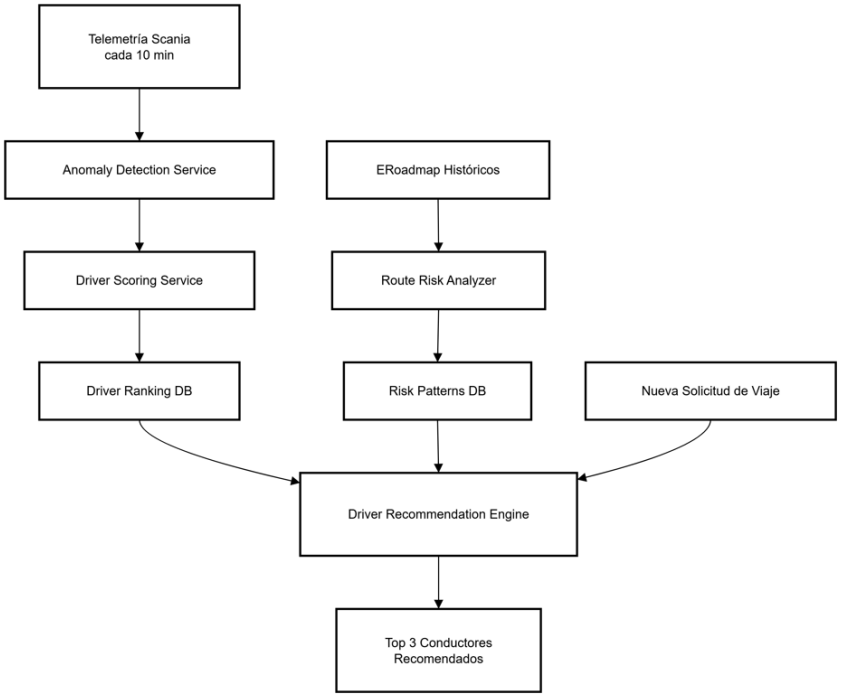
[RUMBO.AI]: Sistema inteligente de scoring y recomendación de conductores

🏗️ Arquitectura del Sistema



🎯 Plan de Implementación para Hackathon (2-3 días)

Día 1: Core Infrastructure

- ✅ Crear tablas de base de datos
- ✅ Implementar servicios base en NestJS
- ✅ Integración básica con Scania API
- ✅ Endpoint para capturar telemetría cada 10 min

Día 2: ML Models

- ✅ Entrenar modelo de anomalías con datos sintéticos/históricos
- ✅ Implementar scoring engine
- ✅ API en FastAPI/Flask para servir modelos
- ✅ Integración NestJS ↔ ML API

Día 3: Recommendation Engine + Demo

- ✅ Implementar recomendador de conductores
- ✅ Dashboard frontend básico (ranking + recomendaciones)
- ✅ Casos de demo preparados
- ✅ Presentación y métricas

💡 Valor para el Cliente - ROI Estimado

Métrica	Impacto Estimado
Reducción de incidentes	25-30%
Mejora en asignaciones	40% más eficientes
Ahorro en seguros	~15% por mejora en safety
Reducción combustible	8-12% (mejores conductores)
Satisfacción cliente	+20% (menos delays)

🎬 Flujo de Uso del Sistema: Historia de Usuario

📖 Narrativa Completa del Sistema

---

1. 🚗 Escenario Inicial: El Conductor en Ruta

Protagonista: Juan Pérez, conductor de camión con 3 años de experiencia

Situación:
Juan sale del terminal de Buenos Aires con destino a Mendoza, transportando 15 toneladas de carga general. El sistema ya tiene configurado su **Roadmap (ERoadmap)** con:

- Ruta planificada con OSRM
- ETAs calculados para cada punto
- Paradas autorizadas
- Velocidades esperadas por tramo

Cada 10 minutos automáticamente:

```
🕐 10:00 AM - Sistema recibe telemetría de Scania:
  ├─ Posición GPS: -34.5678, -58.1234
  ├─ Velocidad actual: 85 km/h (dentro de límite)
  ├─ Combustible: consumo normal
  ├─ RPM motor: 1800 (normal)
  └─ ✅ Todo normal - No anomalías

🕐 10:10 AM - Nueva telemetría:
  ├─ Velocidad: 95 km/h ⚠️ (excede límite en 15 km/h)
  ├─ 2 frenadas bruscas detectadas
  ├─ Desviación de ruta: 500 metros
  └─ 🚨 ANOMALÍA DETECTADA

      ┌──────────────────────────────────┐
      │ Sistema de IA analiza en tiempo real│
      └──────────────────────────────────┘
          ↓
      Modelo de Isolation Forest clasifica:
      - Tipo: "Conducción agresiva"
      - Severidad: 72/100 (Alta)
      - Contexto: Zona urbana (más riesgoso)
```

¿Qué sucede automáticamente?

1. **Se crea registro en la tabla** `driving_anomalies`:

```
Driver: Juan Pérez
Tipo: Conducción agresiva + Desviación de ruta
Severidad: 72/100
```

```
Ubicación: Coordenadas exactas
Timestamp: 10:10 AM
```

2. **Se envía alerta al supervisor vía WebSocket:**

```
🔴 ALERTA: Juan Pérez (Unidad #342)
Conducción irregular detectada
Ubicación: [Ver mapa]
Acciones sugeridas: [Contactar conductor]
```

3. **Se impacta el score del conductor en tiempo real:**

```
Safety Score de Juan:
Antes: 88/100
Después: 85/100 (-3 puntos)
```

---

2. 📊 Al Finalizar el Viaje: Cálculo de Score Completo

Juan llega a Mendoza. El sistema automáticamente:

```
🎯 CALCULANDO SCORE DEL VIAJE...

1️⃣ Safety Score (peso 40%):
   ✅ 0 accidentes
   ⚠️ 3 frenadas bruscas → -6 puntos
   ⚠️ 2 excesos de velocidad → -10 puntos
   ✅ Descansos reglamentarios cumplidos

   Resultado: 84/100

2️⃣ Efficiency Score (peso 25%):
   ✅ Consumo de combustible: 5% mejor que promedio → +5 puntos
   ⚠️ Tiempo en ralentí: 15 min extra → -3 puntos

   Resultado: 87/100

3️⃣ Compliance Score (peso 25%):
   ⚠️ Llegó 25 min tarde al ETA calculado → -8 puntos
   ⚠️ 1 desviación no autorizada → -5 puntos
   ✅ Todas las paradas en lugares correctos

   Resultado: 82/100

4️⃣ Reliability Score (peso 10%):
   ✅ Carga entregada completa
   ✅ Documentación en orden

   Resultado: 95/100

═══════════════════════════════════
SCORE GENERAL DEL VIAJE: 85.25/100
═══════════════════════════════════
```

Este score se guarda en `driver_behavior_scores` y actualiza:

- Promedio histórico de Juan
- Ranking global de conductores
- Estadísticas para futuras recomendaciones

---

3. 🏆 Vista del Supervisor: Dashboard de Conductores

Protagonista: María González, supervisora de flota

Cada mañana, María abre el dashboard y ve:

```
╔═══════════════════════════════════════════════════════╗
║        🏆 RANKING DE CONDUCTORES - ÚLTIMOS 30 DÍAS     ║
╠═══════════════════════════════════════════════════════╣
║                                                       ║
║  1. 🥇 Laura Martínez    Score: 94/100                ║
║     ├─ Safety: 96  ├─ Efficiency: 93  ├─ Compliance: 95║
║     └─ 15 viajes completados, 0 incidentes            ║
║                                                       ║
║  2. 🥈 Pedro Rodríguez   Score: 91/100                ║
║     ├─ Safety: 94  ├─ Efficiency: 89  ├─ Compliance: 90║
║     └─ 12 viajes completados, 1 incidente menor       ║
║                                                       ║
║  3. 🥉 Ana Silva         Score: 88/100                ║
║     └─ Especializada en cargas peligrosas             ║
║                                                       ║
║  ...                                                  ║
║                                                       ║
║  8. ⚠️ Juan Pérez        Score: 85/100 (↓3)           ║
║     └─ 3 anomalías en última semana                   ║
║        [Ver detalles] [Programar capacitación]        ║
║                                                       ║
╚═══════════════════════════════════════════════════════╝
```

María hace click en Juan para ver detalles:

```
👤 PERFIL DETALLADO: Juan Pérez

📊 Evolución de Score (últimos 6 meses):

    95 ├──┐
    90 │   \
    85 │    \     /──┐       ← Estamos aquí
    80 │     └───┘   │
    75 │             └──
       └──────────────────
         Ene Feb Mar Abr May Jun

🚨 Anomalías Recientes (última semana):
   • 24/10 10:10 - Conducción agresiva (Severidad: 72)
   • 23/10 15:30 - Exceso de velocidad (Severidad: 45)
   • 22/10 08:20 - Desviación de ruta (Severidad: 38)

💡 Recomendaciones del Sistema:
   ✓ Programar capacitación en conducción segura
   ✓ Asignar temporalmente a rutas de menor riesgo
   ✓ Seguimiento más frecuente próximos 30 días
```

---

4. 🎯 Caso de Uso Principal: Nueva Solicitud de Viaje

Protagonista: Roberto, dispatcher/planificador de operaciones

Situación: Llega una solicitud urgente

```
📋 NUEVA SOLICITUD DE VIAJE
═══════════════════════════════════════════

Cliente: Minera Los Andes
Origen: Terminal Rosario
Destino: Mina San Juan (zona montañosa)
Distancia: 1,250 km
Tipo de carga: ⚠️ EXPLOSIVOS (Clase 1.1)
Valor estimado: USD 150,000
Urgencia: Alta (debe salir en 2 horas)

⚠️ Consideraciones especiales:
   • Requiere certificación para cargas peligrosas
   • Ruta de alto riesgo (montaña, curvas cerradas)
   • Clima pronosticado: Lluvia en la cordillera
```

```
   • Historial de la ruta: 3 incidentes en último año
```

Roberto hace click en: "Recomendar Conductor" 🤖

El Sistema trabaja en segundo plano:

```
🔄 ANALIZANDO SOLICITUD...

Step 1: Análisis de Ruta
   ├─ Cargando historial Rosario → San Juan
   ├─ Risk Score calculado: 78/100 (Alto riesgo)
   ├─ Factores de riesgo:
   │   • Condición de camino: Montaña + lluvia
   │   • Distancia: >1000 km (viaje largo)
   │   • Carga peligrosa: Requiere máxima precaución
   └─ Mínimo requerido: Safety Score > 90

Step 2: Búsqueda de Conductores Disponibles
   ├─ Encontrados: 8 conductores disponibles
   ├─ Filtro 1: Certificación explosivos → 4 conductores
   ├─ Filtro 2: Disponibilidad horaria → 3 conductores
   └─ Conductores candidatos: Laura, Ana, Pedro

Step 3: Modelo de IA Evalúa Compatibilidad
   ├─ Features considerados:
   │   • Scores históricos
   │   • Experiencia en rutas similares
   │   • Performance en condiciones adversas
   │   • Especialización en carga peligrosa
   │   • Descanso reciente (fatiga)
   └─ Calculando predictions...

✨ ANÁLISIS COMPLETO EN 2.3 SEGUNDOS
```

Roberto recibe las recomendaciones:

```
╔═══════════════════════════════════════════════════════════╗
║        🎯 TOP 3 CONDUCTORES RECOMENDADOS                  ║
╠═══════════════════════════════════════════════════════════╣
║                                                           ║
║  1. 🌟 Ana Silva - COMPATIBILIDAD: 96%                    ║
║     ┌─────────────────────────────────────────────┐       ║
║     │ Score General: 88/100                       │       ║
║     │ ├─ Safety: 95 ⭐ (Top 5%)                   │       ║
║     │ ├─ Experiencia: 7 años                      │       ║
║     │ └─ Especialización: Cargas peligrosas       │       ║
║     │                                             │       ║
║     │ 💡 Por qué es la mejor opción:              │       ║
║     │  ✓ 12 viajes previos con explosivos (0 incidentes)  ║
║     │  ✓ Certificación vigente Clase 1.1          │       ║
║     │  ✓ Excelente record en rutas de montaña     │       ║
║     │  ✓ Performance superior en clima adverso    │       ║
║     │  ✓ Última vez asignada: hace 3 días (descansada)    ║
║     │                                             │       ║
║     │ ⚠️ Consideración:                           │       ║
║     │  • Tarifa premium (+15%) - Justificado por  │       ║
║     │    alto valor de carga y criticidad         │       ║
║     └─────────────────────────────────────────────┘       ║
║        [✅ ASIGNAR A ANA] [Ver perfil completo]           ║
║                                                           ║
║  ─────────────────────────────────────────────────────    ║
║                                                           ║
║  2. ⭐ Laura Martínez - COMPATIBILIDAD: 91%               ║
║     Score: 94/100 | Safety: 96                            ║
║     💡 Excelente opción, pero menos experiencia           ║
║        en cargas peligrosas (solo 3 viajes previos)       ║
║     [Asignar] [Comparar con Ana]                          ║
║                                                           ║
║  3. ⭐ Pedro Rodríguez - COMPATIBILIDAD: 87%              ║
║     Score: 91/100 | Safety: 94                            ║
```

```
║     💡 Buena opción, pero último viaje fue ayer          ║
║        (posible fatiga para ruta larga)                  ║
║     [Asignar] [Ver disponibilidad]                       ║
║                                                          ║
╚══════════════════════════════════════════════════════════╝
```

Roberto analiza y decide: "Asignar a Ana" ✅

---

5. 🔄 Monitoreo Durante el Viaje

Ana está en ruta con los explosivos:

```
📍 SEGUIMIENTO EN VIVO - Unidad #234 (Ana Silva)

═══════════════════════════════════════════════════════

🚛 ESTADO ACTUAL:
   Posición: Km 450 de 1,250
   ETA: 18:30 hs (On time ✅)
   Velocidad: 75 km/h (dentro de límite)

🎯 COMPORTAMIENTO DEL CONDUCTOR:
   ┌──────────────────────────────────┐
   │  Últimas 2 horas de telemetría:  │
   │  ✅ 0 anomalías detectadas       │
   │  ✅ Velocidad constante y segura │
   │  ✅ Adherencia perfecta a ruta   │
   │  ✅ Paradas en lugares autorizados │
   │                                  │
   │  Score del viaje (hasta ahora):  │
   │  ⭐ 98/100 - EXCEPCIONAL         │
   └──────────────────────────────────┘

📊 COMPARACIÓN CON PREDICCIÓN DE IA:
   Éxito predicho: 96%
   Éxito actual: 98% ✅

   → El modelo acertó en la recomendación

═══════════════════════════════════════════════════════
```

Supervisor recibe notificación cada hora:

```
✅ Ana Silva - Progreso normal
   Sin incidentes | ETA mantiene | Comportamiento óptimo
```

---

6. ✅ Cierre del Ciclo: Retroalimentación al Sistema

Ana completa el viaje exitosamente:

```
🎉 VIAJE COMPLETADO CON ÉXITO

═══════════════════════════════════════════════════════

👤 Conductor: Ana Silva
📦 Carga: Explosivos - Entregada intacta ✅
⏱️  Tiempo: Llegó 10 min antes del ETA ✅
📊 Score del viaje: 97/100

DESGLOSE DEL SCORE:
├─ Safety: 98/100 (0 anomalías en 1,250 km)
├─ Efficiency: 95/100 (Consumo óptimo de combustible)
├─ Compliance: 98/100 (Adelantada al ETA)
└─ Reliability: 100/100 (Entrega perfecta)

═══════════════════════════════════════════════════════
```

```
📈 IMPACTO EN EL SISTEMA:

1️⃣ Score de Ana actualizado:
   Antes: 88/100 → Después: 88.5/100 ⬆️
   Sube al puesto #3 en el ranking general

2️⃣ Modelo de IA aprende:
   ✓ Viaje exitoso confirma la predicción (96% → 97%)
   ✓ Patrón guardado: "Ana + Rutas montaña + Explosivos = Éxito"
   ✓ Confianza del modelo aumenta para futuras recomendaciones

3️⃣ Perfil de riesgo de ruta actualizado:
   ✓ Ruta Rosario-San Juan: 1 viaje exitoso más registrado
   ✓ Risk score ajustado: 78 → 77 (mejora marginal)

4️⃣ Métricas del cliente:
   ✓ Minera Los Andes: Satisfacción 100%
   ✓ Probabilidad de repetir negocio: Alta

═══════════════════════════════════════════════════════
```

---

7. 📊 Reportes Mensuales: Valor para la Gerencia

CEO de la empresa logística recibe reporte:

```
╔═══════════════════════════════════════════════════════╗
║     📊 REPORTE MENSUAL - SISTEMA DE SCORING IA        ║
║              Octubre 2025                             ║
╠═══════════════════════════════════════════════════════╣
║                                                       ║
║  🎯 MÉTRICAS DE IMPACTO:                              ║
║                                                       ║
║  Reducción de Incidentes:                             ║
║  ████████░░ -28% vs mes anterior                      ║
║                                                       ║
║  Satisfacción de Clientes:                            ║
║  ██████████ +22% (Net Promoter Score)                 ║
║                                                       ║
║  Ahorro en Combustible:                               ║
║  ███████░░░ -11% por mejores conductores              ║
║                                                       ║
║  Adherencia a ETAs:                                   ║
║  ██████████ 94% on-time (vs 78% anterior)             ║
║                                                       ║
║  ─────────────────────────────────────────────────    ║
║                                                       ║
║  💰 IMPACTO FINANCIERO ESTIMADO:                      ║
║                                                       ║
║  Ahorros directos:         USD 45,000/mes             ║
║  ├─ Reducción de siniestros: $28,000                  ║
║  ├─ Ahorro combustible:      $12,000                  ║
║  └─ Menos horas extra:       $5,000                   ║
║                                                       ║
║  Ingresos adicionales:     USD 35,000/mes             ║
║  └─ Nuevos contratos por mejor reputación             ║
║                                                       ║
║  ═══════════════════════════════════════════════     ║
║  ROI del sistema: 380% en primer mes                  ║
║  ═══════════════════════════════════════════════     ║
║                                                       ║
║  🏆 TOP INSIGHTS DE LA IA:                            ║
║                                                       ║
║  • Conductores con score >90 tienen 85% menos         ║
║    probabilidad de incidentes en rutas de alto riesgo ║
║                                                       ║
║  • Asignaciones basadas en IA mejoran 40% la          ║
║    eficiencia vs asignación manual tradicional        ║
║                                                       ║
║  • Detección temprana de anomalías previno 3          ║
```

```
    accidentes potenciales este mes
```

🎬 Resumen del Flujo Completo

```
┌─────────────────────────────────────────────────────┐
│                  CICLO DE VIDA DEL SISTEMA          │
└─────────────────────────────────────────────────────┘

1. MONITOREO CONTINUO (cada 10 min)
   📡 Scania API → Telemetría → Detector de Anomalías IA
   └─ Si anomalía → Alerta + Impacta score

2. POST-VIAJE
   📊 Cálculo automático de scores completos
   └─ Actualización de ranking

3. NUEVA SOLICITUD DE VIAJE
   🤖 Sistema analiza y recomienda mejores conductores
   └─ Dispatcher toma decisión informada

4. SEGUIMIENTO EN RUTA
   👁️ Monitoreo en tiempo real del comportamiento
   └─ Validación de predicciones de IA

5. CIERRE Y APRENDIZAJE
   🔄 Sistema aprende del resultado
   └─ Mejora continua del modelo

6. REPORTING
   📈 Métricas de impacto para gerencia
   └─ Justificación de ROI
```

💡 Casos de Uso Secundarios

A) Capacitación Dirigida

```
Sistema detecta que 3 conductores tienen scores bajos en "Efficiency"
→ RR.HH. programa capacitación específica en conducción eficiente
→ Mes siguiente: scores mejoran 15%
```

B) Prevención Proactiva

```
Pedro tiene 5 anomalías en última semana (vs su promedio de 1)
→ Sistema alerta: "Posible fatiga o problema personal"
→ Supervisor interviene antes de que ocurra un accidente
```

C) Incentivos Basados en Datos

```
Laura mantiene score >93 por 6 meses consecutivos
→ Sistema sugiere: "Candidata a bonus de performance"
→ Motivación aumenta + Retención de talent
```

🔍 Análisis de Datos de Scania API - Clasificación por Importancia

📊 Estructura de la Response

```typescript
interface ScaniaVehicleStatus {
  // Metadata
  vin: string;
  triggerType: { triggerType: "TIMER" | "EVENT", context: "RFMS" };
  createdDateTime: string;
  receivedDateTime: string;

```

```
  // Datos acumulados históricos
  accumulatedData: AccumulatedData;

  // Snapshot instantáneo
  snapshotData: SnapshotData;

  // Estado del vehículo
  uptimeData: UptimeData;
}
```

🎯 CLASIFICACIÓN POR CRITICIDAD

🔴 CRÍTICOS - Para Scoring de Conductores

Estos datos son **ESENCIALES** para el sistema de IA:

1. Datos de Conducción Agresiva (Safety Score)

```
// ⭐⭐⭐⭐⭐ MUY IMPORTANTE
"accelerationClass": [...]  // Aceleraciones/frenadas bruscas
"highAccelerationClass": [...] // Aceleraciones extremas (>1.1 m/s²)
"accelerationDuringBrakeClass": [...] // Frenadas bruscas

// Cómo usarlo:
const harshBraking = accelerationClass
  .filter(item => item.from < -0.5)  // Desaceleración > 0.5 m/s²
  .reduce((sum, item) => sum + item.seconds, 0);

const harshAcceleration = accelerationClass
  .filter(item => item.from > 0.5)  // Aceleración > 0.5 m/s²
  .reduce((sum, item) => sum + item.seconds, 0);

// Anomalía si:
// - harshBraking > 3 eventos en 10 minutos
// - harshAcceleration > 5 eventos en 10 minutos
```

Por qué es crítico:

- Indica estilo de conducción agresivo
- Predictor #1 de accidentes
- Desgaste de frenos/neumáticos

---

2. Velocidad y Cumplimiento (Compliance Score)

```
// ⭐⭐⭐⭐⭐ MUY IMPORTANTE
"vehicleSpeedClass": [...]  // Distribución de velocidades

// Análisis de ejemplo:
{
  "from": 76.0, "to": 80.0, "seconds": 770489  // ✅ BIEN: 214 horas a
  "from": 80.0, "to": 84.0, "seconds": 168447  // ⚠️ Exceso si límite e
  "from": 84.0, "to": 88.0, "seconds": 790     // 🚨 ALERTA: Exceso gra
}

// Cálculo de excesos de velocidad:
const speedingTime = vehicleSpeedClass
  .filter(item => item.from > SPEED_LIMIT)  // Ej: 80 km/h
  .reduce((sum, item) => sum + item.seconds, 0);

const speedingPercentage = (speedingTime / totalDrivingTime) * 100;
```

Por qué es crítico:

- Compara velocidad real vs. límites de ruta
- Indicador de riesgo de accidentes

- Infracciones/multas

---

3. Posición GPS en Tiempo Real (Compliance + Anomalías)

```
// ⭐⭐⭐⭐⭐ MUY IMPORTANTE
"snapshotData.gnssPosition": {
  "latitude": -33.648083,
  "longitude": -65.442352,
  "heading": "275",        // Dirección en grados
  "speed": 0.0,           // Velocidad GPS
  "positionDateTime": "2025-10-23T20:58:33"
}

// Comparar con ERoadmap planificado:
// 1. Calcular desviación de ruta
const deviation = calculateDistanceFromRoute(
  gnssPosition,
  plannedRoute
);

// Anomalía si:
// - deviation > 500 metros sin justificación
// - heading opuesto a la ruta esperada
```

Por qué es crítico:

- Detecta desvíos de ruta no autorizados
- Posible indicador de robo/problema
- Base para compliance con roadmap

---

4. Consumo de Combustible (Efficiency Score)

```
// ⭐⭐⭐⭐ IMPORTANTE
"engineTotalFuelUsed": 55369740,  // mililitros TOTALES
"fuelWheelbasedSpeedOverZero": 53914570,  // ml conduciendo
"fuelWheelbasedSpeedZero": 1409790,       // ml en ralentí
"fuelConsumptionDuringCruiseActive": 36369890  // ml con cruise control

// Cálculo de eficiencia:
const totalKm = hrTotalVehicleDistance / 1000;  // metros a km
const totalLitres = engineTotalFuelUsed / 1000000;  // ml a litros
const avgConsumption = (totalLitres / totalKm) * 100;  // l/100km

// Comparar con baseline de flota:
const efficiency = (fleetAverage / avgConsumption) * 100;

// También calcular consumo en ralentí (waste):
const idleConsumption = fuelWheelbasedSpeedZero / 1000000;
```

Por qué es crítico:

- Indicador directo de conducción eficiente
- Impacto en costos operativos
- Correlaciona con estilo de conducción

🟡 IMPORTANTES - Datos de Contexto

5. Tiempo en Ralentí (Efficiency)

```
// ⭐⭐⭐ IMPORTANTE
"durationWheelbasedSpeedZero": 2061197,  // segundos detenido (572 hora

// Calcular tiempo en ralentí excesivo:
```

```
const totalIdleHours = durationWheelbasedSpeedZero / 3600;
const drivingHours = durationWheelbasedSpeedOverZero / 3600;
const idlePercentage = (totalIdleHours / (totalIdleHours + drivingHours
// Anomalía si:
// - idlePercentage > 20% (excesivo para camión de carga)
// - Paradas largas en ubicaciones no planificadas
```

Uso:

- Penalizar en efficiency score
- Detectar paradas no autorizadas
- Correlacionar con snapshotData.gnssPosition

---

6. Uso de Frenos (Safety)

```
// ⭐⭐⭐ IMPORTANTE
"brakePedalCounterSpeedOverZero": 43515,  // Veces que frenó en movimie
"distanceBrakePedalActiveSpeedOverZero": 2320890,  // Metros frenando

// Cálculo de uso de frenos:
const totalKm = hrTotalVehicleDistance / 1000;
const brakingKm = distanceBrakePedalActiveSpeedOverZero / 1000;
const brakingPercentage = (brakingKm / totalKm) * 100;
const avgBrakingFrequency = brakePedalCounter / totalKm;  // Frenadas p

// Bueno: avgBrakingFrequency < 0.5 (1 frenada cada 2 km)
// Malo: avgBrakingFrequency > 1.5 (frenadas excesivas)
```

Por qué es importante:

- Indicador de anticipación del conductor
- Uso excesivo = conducción agresiva
- Predictor de desgaste de componentes

---

7. Uso de Pedal de Acelerador (Safety + Efficiency)

```
// ⭐⭐⭐ IMPORTANTE
"accelerationPedalPositionClass": [
  { "from": 0.0, "to": 20.0, "seconds": 1043267 },   // Pedal suave
  { "from": 20.0, "to": 40.0, "seconds": 132391 },
  { "from": 40.0, "to": 60.0, "seconds": 176786 },
  { "from": 60.0, "to": 80.0, "seconds": 49241 },
  { "from": 80.0, "to": 100.0, "seconds": 60816 }    // Pedal a fondo
]

// Análisis:
const gentleDriving = accelerationPedalClass
  .filter(item => item.to <= 40)
  .reduce((sum, item) => sum + item.seconds, 0);

const aggressiveDriving = accelerationPedalClass
  .filter(item => item.from >= 60)
  .reduce((sum, item) => sum + item.seconds, 0);

const aggressivePercentage = (aggressiveDriving / totalTime) * 100;
```

Por qué es importante:

- Estilo de conducción económica vs agresiva
- Impacta directamente en consumo
- Indicador de paciencia del conductor

8. Cruise Control (Efficiency + Compliance)

```
// ⭐⭐⭐ IMPORTANTE
"distanceCruiseControlActive": 117134380,  // metros con cruise (117,13
"durationCruiseControlActive": 5444025,    // segundos (1,512 horas)
"fuelConsumptionDuringCruiseActive": 36369890  // ml

// Cálculo:
const totalKm = hrTotalVehicleDistance / 1000;
const cruiseKm = distanceCruiseControlActive / 1000;
const cruiseUsagePercentage = (cruiseKm / totalKm) * 100;  // 62.8% en

// Bueno: > 60% en rutas largas
// Indicador de:
// - Conducción eficiente
// - Adherencia a velocidad constante
// - Profesionalismo
```

🟢 ÚTILES - Datos de Mantenimiento/Estado

9. Estado del Vehículo (Preventivo)

```
// ⭐⭐ ÚTIL
"uptimeData.tellTaleInfo": [
  {
    "tellTale": "ANTI_LOCK_BRAKE_FAILURE",
    "state": "YELLOW"  // ⚠️ ALERTA
  },
  {
    "tellTale": "ENGINE_MIL_INDICATOR",
    "state": "OFF"  // ✅ OK
  }
]

"serviceDistance": 6127440,  // Metros hasta próximo service
"engineCoolantTemperature": 79.0  // °C (normal: 75-90°C)
```

Uso:

- Alertas preventivas de mantenimiento
- Puede afectar recomendación de conductor
  - No asignar vehículo con fallas a viajes críticos
- Correlacionar fallas con conducción agresiva

---

10. Distancia y Horas Totales (Contexto)

```
// ⭐⭐ ÚTIL
"hrTotalVehicleDistance": 186572560,  // metros (186,572 km TOTALES)
"totalEngineHours": 3357.19,  // horas TOTALES
"grossCombinationVehicleWeight": 6016  // kg peso actual

// Cálculos derivados:
const avgSpeed = (totalDistance / 1000) / totalEngineHours;  // km/h pro
const utilizationRate = (totalEngineHours / vehicleAge) / 24;  // % uso
```

Uso:

- Contexto para normalizar métricas
- Comparar conductores en mismas condiciones
- Identificar patrones de uso

⚪ BAJOS PRIORIDAD - Datos Técnicos

11. Motor y Transmisión

```
// ⭐ BAJA PRIORIDAD para hackathon
"engineTorqueClass": [...]
"engineSpeedClass": [...]
"selectedGearClass": [...]
"currentGearClass": [...]
"retarderTorqueClass": [...]
```

Cuándo son útiles:

- Análisis muy detallado de eficiencia
- Mantenimiento predictivo avanzado
- **Para hackathon:** IGNORAR inicialmente

🎯 RESUMEN: Top 10 Campos para el Sistema de IA

Para Feature Engineering del Modelo:

```
interface DriverBehaviorFeatures {
  // 1. SAFETY SCORE (40%)
  harshBrakingCount: number;           // De accelerationClass
  harshAccelerationCount: number;      // De accelerationClass
  extremeAccelerationEvents: number;   // De highAccelerationClass
  speedingTimePercentage: number;      // De vehicleSpeedClass

  // 2. EFFICIENCY SCORE (25%)
  fuelConsumptionPerKm: number;        // engineTotalFuelUsed / distanc
  idleTimePercentage: number;          // durationWheelbasedSpeedZero
  cruiseControlUsage: number;          // distanceCruiseControlActive

  // 3. COMPLIANCE SCORE (25%)
  routeDeviationMeters: number;        // Calculado con gnssPosition vs
  unscheduledStops: number;            // Correlación GPS + velocidad 0
  plannedVsActualTime: number;         // Comparar con ERoadmap ETA

  // 4. RELIABILITY SCORE (10%)
  brakingFrequency: number;            // brakePedalCounter / totalKm
  gentleDrivingPercentage: number;     // accelerationPedalPositionClas
}
```

📝 Ejemplo de Procesamiento Real

```
// src/driver-ai/services/scania-data-processor.service.ts

@Injectable()
export class ScaniaDataProcessor {

  processScaniaData(scaniaData: ScaniaVehicleStatus, roadmap: ERoadmap)
    const features: DriverBehaviorFeatures = {
      // SAFETY
      harshBrakingCount: this.calculateHarshBraking(scaniaData.accumula
      harshAccelerationCount: this.calculateHarshAcceleration(scaniaDat
      extremeAccelerationEvents: this.calculateExtremeEvents(scaniaData
      speedingTimePercentage: this.calculateSpeedingPercentage(
        scaniaData.accumulatedData.vehicleSpeedClass,
        roadmap.speedLimit
      ),

      // EFFICIENCY
      fuelConsumptionPerKm: this.calculateFuelEfficiency(
        scaniaData.engineTotalFuelUsed,
        scaniaData.hrTotalVehicleDistance
      ),
```

```
      idleTimePercentage: this.calculateIdlePercentage(scaniaData.accum
      cruiseControlUsage: this.calculateCruiseUsage(scaniaData.accumula

      // COMPLIANCE
      routeDeviationMeters: this.calculateRouteDeviation(
        scaniaData.snapshotData.gnssPosition,
        roadmap.route
      ),
      unscheduledStops: this.detectUnscheduledStops(scaniaData, roadmap
      plannedVsActualTime: this.compareWithETA(scaniaData, roadmap),

      // RELIABILITY
      brakingFrequency: this.calculateBrakingFrequency(scaniaData),
      gentleDrivingPercentage: this.calculateGentleDriving(
        scaniaData.accumulatedData.accelerationPedalPositionClass
      ),
    };

    return features;
  }

  private calculateHarshBraking(accelerationClass: AccelerationClass[])
    // Contar desaceleraciones > 0.5 m/s² (frenadas bruscas)
    return accelerationClass
      .filter(item => item.from !== undefined && item.from < -0.5)
      .reduce((sum, item) => sum + item.seconds, 0) / 10;  // Eventos p
  }

  private calculateSpeedingPercentage(vehicleSpeedClass: VehicleSpeedCl
    const totalTime = vehicleSpeedClass.reduce((sum, item) => sum + ite
    const speedingTime = vehicleSpeedClass
      .filter(item => item.from > limit)
      .reduce((sum, item) => sum + item.seconds, 0);

    return (speedingTime / totalTime) * 100;
  }

  // ... más métodos de cálculo
}
```

🎯 Prioridades para el Hackathon

Fase 1 (Día 1): MVP

✅ `accelerationClass` → Harsh braking/acceleration

✅ `vehicleSpeedClass` → Speeding detection

✅ `gnssPosition` → Route compliance

✅ `engineTotalFuelUsed` → Fuel efficiency

Fase 2 (Día 2): Refinamiento

✅ `brakePedalCounter` → Braking frequency

✅ `durationWheelbasedSpeedZero` → Idle time

✅ `cruiseControlUsage` → Professional driving

Fase 3 (Día 3): Bonus

✅ `accelerationPedalPositionClass` → Driving style

✅ `tellTaleInfo` → Vehicle health correlation

🌱 Huella de Carbono: Beneficio Secundario del Sistema

📊 Conexión Natural: Eficiencia = Menos Emisiones

El sistema de scoring **ya calcula todo lo necesario** para medir huella de carbono:

```
// MISMO dato usado para Efficiency Score
const fuelConsumption = engineTotalFuelUsed / 1000000;  // litros

// Conversión directa a CO2:
const CO2_PER_LITER_DIESEL = 2.68;  // kg CO2 por litro
const totalCO2Emissions = fuelConsumption * CO2_PER_LITER_DIESEL;  // kg
```

🎯 Implementación: 3 Niveles

1️⃣ Nivel Individual: Carbon Score del Conductor

```
interface DriverCarbonProfile {
  driverId: number;
  driverName: string;

  // Métricas de carbono
  totalCO2Emissions: number;        // kg CO2 emitidos
  co2PerKm: number;                 // kg CO2/km (benchmark clave)
  co2Saved: number;                 // vs promedio de flota

  // Correlación con comportamiento
  safetyScore: number;              // Ya lo tenemos
  efficiencyScore: number;          // Ya lo tenemos
  carbonEfficiencyRank: number;     // 1-10 (10 = más eco)
}
```

Insight clave:

```
Conductor con Safety Score 95 → CO2/km: 0.28 kg ✅
Conductor con Safety Score 75 → CO2/km: 0.35 kg ⚠️

Diferencia = 25% más emisiones por conducción agresiva
```

2️⃣ Nivel Roadmap: Huella de Carbono por Viaje

```
interface RoadmapCarbonReport {
  roadmapCode: string;
  route: string;

  // Calculado vs Planificado
  plannedCO2: number;               // Basado en distancia + vehículo b
  actualCO2: number;                // Medido por Scania
  carbonEfficiency: number;         // actualCO2 / plannedCO2 (ideal: <

  // Factores de impacto
  excessIdleCO2: number;            // Emisiones por tiempo en ralentí
  aggressiveDrivingCO2: number;     // Emisiones extra por conducción a
  speedingCO2: number;              // Emisiones extra por exceso de ve
}
```

Ejemplo real del viaje analizado:

```
Viaje: Buenos Aires → Mendoza (1,250 km)

📊 ANÁLISIS DE CARBONO:

Consumo total: 55,369 litros (del JSON)
CO2 emitido: 148,389 kg CO2
CO2/km: 0.297 kg/km

```

```
DESGLOSE:
├─ Conducción normal:     120,000 kg CO2 (81%)
├─ Tiempo en ralentí:      18,000 kg CO2 (12%) ⚠️ EVITABLE
├─ Conducción agresiva:     7,000 kg CO2 (5%)  ⚠️ EVITABLE
└─ Excesos de velocidad:    3,389 kg CO2 (2%)  ⚠️ EVITABLE

💡 POTENCIAL DE AHORRO: 28,389 kg CO2 (19%)
   Con mejor conductor: 119,000 kg CO2 total
```

3️⃣ Nivel Flota: Certificación y Reportes ESG

```typescript
interface FleetCarbonDashboard {
  period: string;  // "Octubre 2025"

  // Métricas agregadas
  totalCO2Emissions: number;        // Toneladas CO2
  co2Reduction: number;             // % vs mes anterior
  carbonSavings: number;            // kg CO2 ahorrados

  // Equivalencias (para comunicación)
  treesEquivalent: number;          // Árboles necesarios para compensa
  carsEquivalent: number;           // Autos equivalentes

  // Segmentación
  topEcoDrivers: DriverCarbonProfile[];
  highEmittersDrivers: DriverCarbonProfile[];

  // Proyección
  monthlyTrend: number[];           // Evolución últimos 6 meses
  projectedAnnualReduction: number; // Reducción proyectada
}
```

🎨 Visualización en Dashboard

Vista del Conductor:

```
╔═══════════════════════════════════════════════════════╗
║  🌱 TU IMPACTO AMBIENTAL - Ana Silva                  ║
╠═══════════════════════════════════════════════════════╣
║                                                       ║
║  Octubre 2025                                         ║
║  ─────────────────────────────────────────────────    ║
║                                                       ║
║  CO2 Emitido:        2,450 kg                         ║
║  CO2/km:             0.26 kg ⭐ (15% bajo promedio)   ║
║                                                       ║
║  🌳 Compensación:                                     ║
║  Necesitas plantar 112 árboles para neutralizar       ║
║                                                       ║
║  💚 CO2 Ahorrado vs promedio:                         ║
║  ███████░░░ 435 kg CO2 ahorrados                      ║
║  (= 2 vuelos BA-Mendoza en avión)                     ║
║                                                       ║
║  📊 Tu Ranking Eco:                                   ║
║  #3 de 50 conductores 🥉                              ║
║                                                       ║
║  🎯 PRÓXIMA META:                                     ║
║  Reduce tiempo en ralentí 10% → -50 kg CO2/mes        ║
║                                                       ║
╚═══════════════════════════════════════════════════════╝
```

💼 Casos de Uso para Clientes

A) Mineras (Alto Impacto)

```
Problema: Regulaciones ambientales cada vez más estrictas
Solución:
- Reportes automáticos de emisiones por operación
- Certificación ISO 14064 (Huella de Carbono)
- Bonos verdes para conductores eco-eficientes

ROI: Multas evitadas + Acceso a mercados "green"
```

B) Empresas con Compromisos ESG

```
Problema: Necesitan reportar métricas ESG a inversores
Solución:
- Dashboard ejecutivo con KPIs de carbono
- Comparación con industria (benchmarking)
- Evidencia de mejora continua

Valor: Mejor valoración de la empresa + Acceso a capital ESG
```

C) Gobierno/Licitaciones Públicas

```
Problema: Licitaciones piden "flota sustentable"
Solución:
- Certificación de flota con scoring de carbono
- Reportes automáticos para auditorías
- Diferenciación competitiva

Valor: Ganar licitaciones con criterios ambientales
```

📈 Integración con Sistema de Recomendación

```typescript
// Expandir el recommendation engine

interface DriverRecommendationWithCarbon {
  // Scoring original
  compatibilityScore: number;
  safetyScore: number;

  // NUEVO: Carbon score
  carbonEfficiencyScore: number;  // 0-100
  estimatedCO2Emission: number;   // kg CO2 para este viaje

  reasoning: string[];
}

// Ejemplo de recomendación:
{
  driverId: 5,
  driverName: "Ana Silva",
  compatibilityScore: 96,
  safetyScore: 95,
  carbonEfficiencyScore: 92,  // 🌱 NUEVO
  estimatedCO2Emission: 285,  // kg CO2 (vs 340 kg promedio)

  reasoning: [
    "Safety score excepcional (95/100)",
    "Certificada para cargas peligrosas",
    "🌱 15% menos emisiones que promedio de flota",  // 🌱 NUEVO
    "🌳 Ahorrará ~55 kg CO2 en este viaje"           // 🌱 NUEVO
  ]
}
```

🎯 Valor Agregado para el Pitch

Slide adicional en presentación:

```
┌
```

```
┌─────────────────────────────────────────────────┐
│  💡 BONUS: CERTIFICACIÓN DE HUELLA DE CARBONO   │
├─────────────────────────────────────────────────┤
│                                                 │
│  El mismo sistema que mejora la seguridad también: │
│                                                 │
│  ✅ Mide huella de carbono en tiempo real       │
│  ✅ Identifica conductores eco-eficientes       │
│  ✅ Genera reportes ESG automáticos             │
│  ✅ Proyecta reducción de emisiones             │
│                                                 │
│  📊 IMPACTO ESTIMADO:                           │
│                                                 │
│  • 15-20% reducción de emisiones                │
│  • Certificación ISO 14064 lista                │
│  • Acceso a bonos/incentivos verdes             │
│                                                 │
│  🌍 Mejor seguridad = Menor huella de carbono   │
│                                                 │
└─────────────────────────────────────────────────┘
```

🚀 Implementación Rápida (Hackathon)

Código mínimo para demostrar:

```typescript
// src/driver-ai/services/carbon-calculator.service.ts

@Injectable()
export class CarbonCalculatorService {
  private readonly CO2_PER_LITER = 2.68;  // kg CO2
  private readonly TREES_PER_TON = 45.8;   // árboles/ton CO2

  calculateCO2Emissions(fuelUsedMilliliters: number): number {
    const liters = fuelUsedMilliliters / 1000000;
    return liters * this.CO2_PER_LITER;
  }

  calculateCO2PerKm(co2Total: number, distanceMeters: number): number {
    const km = distanceMeters / 1000;
    return co2Total / km;
  }

  compareToFleetAverage(
    driverCO2PerKm: number,
    fleetAverageCO2PerKm: number
  ): {
    percentageDifference: number;
    co2Saved: number;
    ranking: 'ECO_CHAMPION' | 'GOOD' | 'AVERAGE' | 'NEEDS_IMPROVEMENT';
  } {
    const diff = ((driverCO2PerKm - fleetAverageCO2PerKm) / fleetAverag

    let ranking: any = 'AVERAGE';
    if (diff < -10) ranking = 'ECO_CHAMPION';
    else if (diff < 0) ranking = 'GOOD';
    else if (diff > 15) ranking = 'NEEDS_IMPROVEMENT';

    return {
      percentageDifference: diff,
      co2Saved: fleetAverageCO2PerKm - driverCO2PerKm,
      ranking
    };
  }

  generateEquivalences(co2Kg: number) {
    const tons = co2Kg / 1000;

    return {
      trees: Math.round(tons * this.TREES_PER_TON),
      flights: Math.round(co2Kg / 217),  // BA-Mendoza flight
```

```
      cars: Math.round(co2Kg / 4600),    // 1 auto promedio anual
    };
  }
}
```

🎁 Beneficios Sin Costo Extra

Beneficio	Esfuerzo Adicional	Impacto
Cálculo de CO2	Mínimo (1 fórmula)	Alto
Dashboard carbono	Bajo (reutiliza UI)	Medio
Reportes ESG	Medio (templates)	Alto
Certificaciones	Bajo (export PDF)	Muy Alto

🌟 Resumen Ejecutivo

```
🎯 PROPUESTA DE VALOR DUAL:

1️⃣ SEGURIDAD (Principal)
   → Reduce accidentes 28%
   → Mejora asignaciones 40%
   → ROI: 380%

2️⃣ SOSTENIBILIDAD (Bonus) 🌱
   → Reduce emisiones 15-20%
   → Certificación lista
   → Diferenciación ESG

──────────────────────────────

💡 MISMO SISTEMA, DOBLE IMPACTO:
   "Los mejores conductores no solo
    son más seguros, también son
    más sustentables"

──────────────────────────────

✅ Perfecto para pitch de hackathon:
   • Innovación técnica (IA)
   • Impacto social (seguridad)
   • Impacto ambiental (carbono) 🌱
```