

Motor de Responsabilidad Inteligente con Marcus

Integración Python + IA

José Armando Son Rojas

28 de abril de 2025

Seguros Bolívar

Solución:

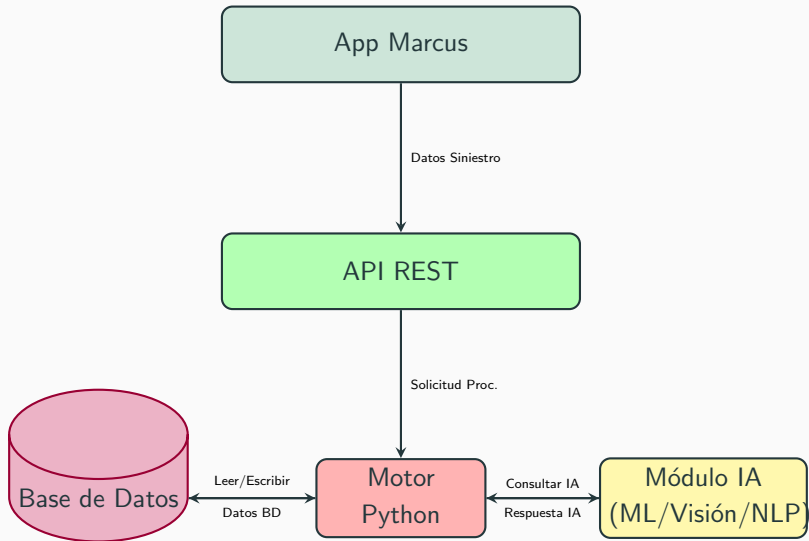
- Motor Python + IA
- Integración con Marcus

Una nueva era en la resolución de choques simples.

Nuestra Solución:

- Motor de decisión en Python.
- Potenciado con Inteligencia Artificial.
- Integración completa con la App Marcus Conductores.
- Enfocado en agilidad y precisión.

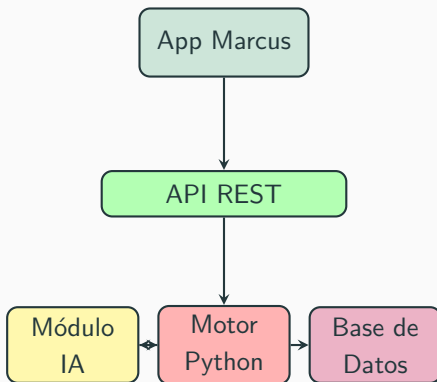
Arquitectura del Sistema



Flujo de datos simplificado:

1. **App Marcus:** Recoge info inicial (fotos, placas, ubicación).
2. **API REST:** Recibe y valida la solicitud.
3. **Motor Python:** Orquesta el proceso, consulta reglas.
4. **Módulo IA:** Analiza evidencias, consulta patrones, apoya predicción/negociación.
5. **Base de Datos:** Almacena/recupera info siniestros, pólizas.

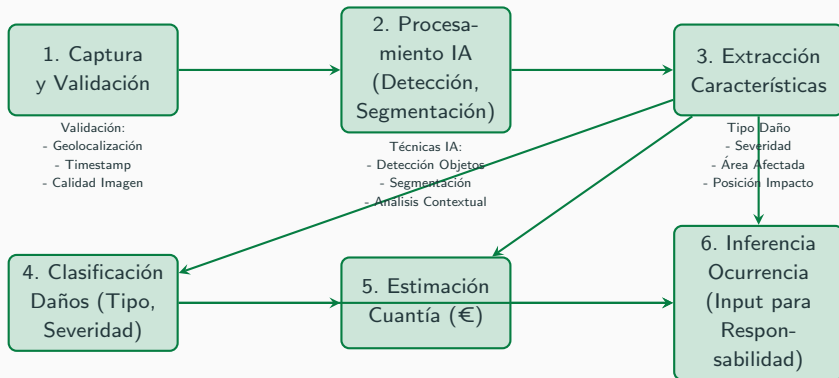
Arquitectura del Sistema



Flujo de datos:

1. App recoge información
2. API procesa la solicitud
3. Motor Python determina responsabilidad
4. IA valida patrones históricos

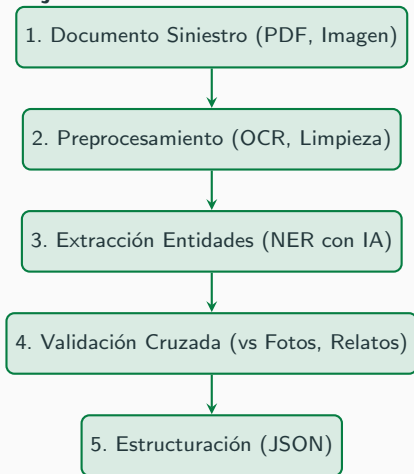
Análisis Avanzado de Evidencias Visuales (Visión IA)



Modelo de Visión Multimodal

Arquitectura (ej: Transformers Vision) que procesa imágenes (daños, escena) y texto asociado para contexto completo. Entrenada con +120,000 casos de siniestros y datos de reparación para **estimar cuantía** y **apoyar la determinación de ocurrencia**.

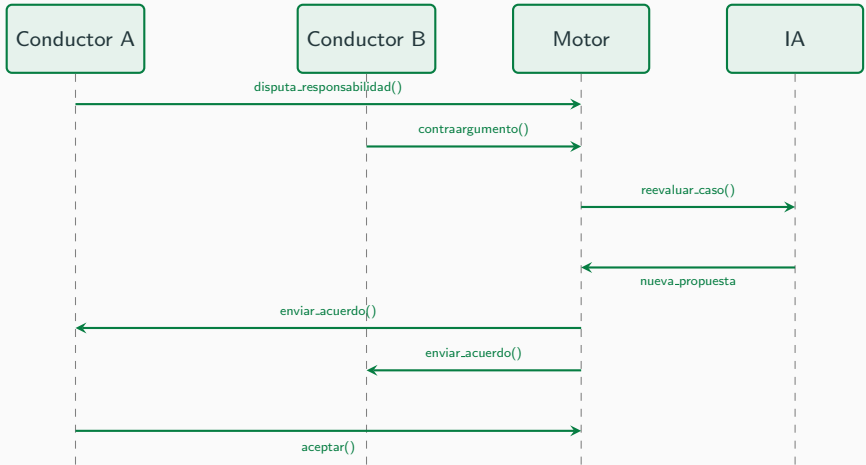
Flujo NLP:



Ejemplo Salida JSON (Simplificado):

```
{
  "siniestro_ref": "SIN-2025-0458",
  "documento_tipo": "Informe Policial",
  "fecha_doc": "2025-03-15",
  "ubicacion_reportada": {
    "direccion": "Calle 85 #15-23",
    "localidad": "Chapinero"
  },
  "vehiculos_doc": [
    {
      "placa_doc": "ABC123",
      "conductor_doc": "Juan Pérez",
      "danos_descritos": "Impacto frontal der.",
      "version_doc": "Vehículo 1 manifiesta que V2 invade carril..."
    },
    {
      "placa_doc": "XYZ789",
      "conductor_doc": "María Gómez",
      "danos_descritos": "Abolladura lateral izq.",
      "version_doc": "Vehículo 2 indica que V1 excedía velocidad..."
    }
  ],
  "hipotesis_autoridad": "Falta de precaución V2 al girar.",
  "consistencia_con_fotos": "Alta", // Resultado validación
  "confianza_extraccion": 0.95 // Score del modelo NLP
}
```

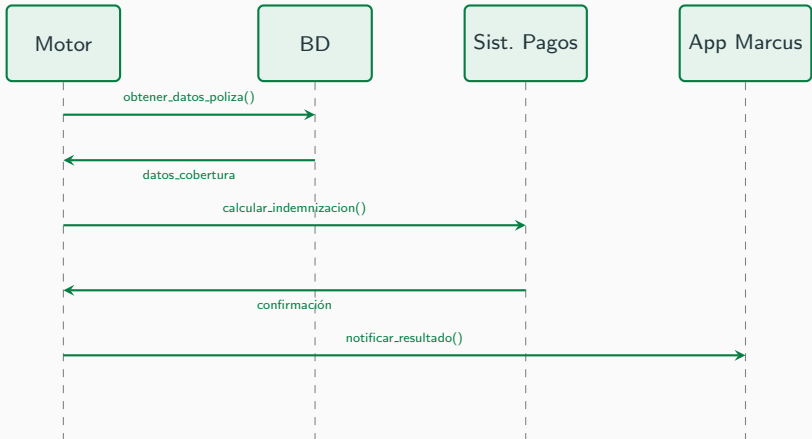
Flujo de Negociación de Responsabilidad



Algoritmo de Negociación

Basado en teoría de juegos y análisis histórico de casos similares.

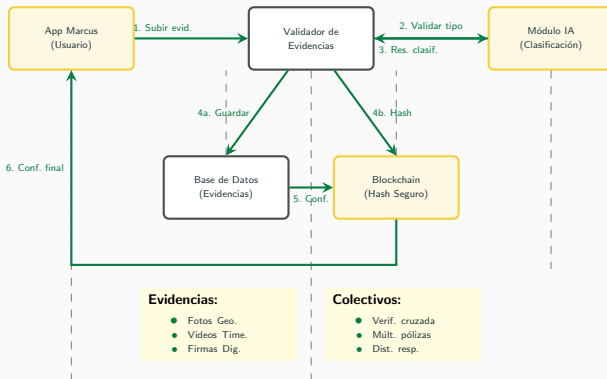
Flujo de Cálculo de Indemnizaciones



Fórmula de Indemnización

$$\text{Monto} = (\% \text{Resp.} \times \text{CostoRep.}) \times \text{FactorCob.}$$

Gestión de Evidencias Digitales



Garantías de Integridad

Blockchain:	Registro inmutable hashes
IA:	Detección fraudes y validación cruzada
Auditoría:	Acceso regulatorio a evidencias

Código del Motor de Responsabilidad

```
class MotorResponsabilidad:
    def procesar_siniestro(self, datos):
        # 1. Validar evidencias
        evidencias = self.validar_evidencias(datos['fotos']) # Simplificado
        # 2. Determinar responsabilidad
        if datos.get('disputa', False):
            resultado = self.negociar_responsabilidad(
                datos.get('conductores', []), evidencias)
        else:
            resultado = self.ia.predecir_responsabilidad(evidencias)

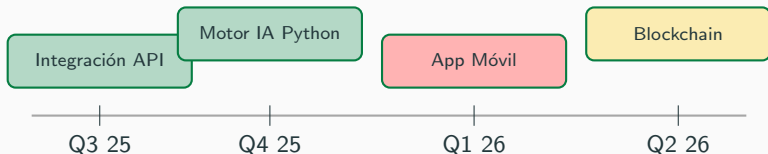
        # 3. Calcular indemnización (Simplificado)
        indemnizacion = self.calcular_indemnizacion(
            resultado.get('responsabilidad', [0,0]),
            datos.get('danos', 0), datos.get('tipo_seguro', 'Individual'))

        # 4. Generar acta digital (Simplificado)
        acta = self.generar_acta(datos, resultado, indemnizacion)

        final_result = {**resultado}
        final_result['indemnizacion'] = indemnizacion
        final_result['acta'] = acta
        return final_result

    # --- Métodos auxiliares (placeholders) ---
    def validar_evidencias(self, fotos): return fotos
    def negociar_responsabilidad(self, c, e): return {'responsabilidad': [50, 50]}
    def calcular_indemnizacion(self, r, d, t): return d * 0.5
    def generar_acta(self, d, r, i): return "Acta Generada"
    # self.ia = ModuloIA() # Necesita __init__
```

Roadmap y Futuras Funcionalidades



Próximos pasos:

- Modelos predictivos para fraude
- Integración con más aseguradoras
- Smart contracts para pagos automáticos

.El motor de responsabilidad de Marcus transforma la gestión de IA, negociación automatizada y cálculo preciso de indemnizaciones”