

# IFTS24 - Proyecto Integrador

Reconocimiento Facial - Detección de Somnolencia



Profesor: Matias Barreto  
Alumno: Guimera Luis

# 1. Definición del Problema

## Área de Interés:

Seguridad vial / Salud pública

## Problema Detectado:

Uno de los principales factores de riesgo en accidentes de tránsito es la somnolencia o fatiga del conductor. Detectar signos tempranos de fatiga visual puede reducir significativamente estos eventos.

## Justificación:

Según organismos internacionales (como la OMS o la Agencia Nacional de Seguridad Vial), conducir con fatiga puede ser tan peligroso como hacerlo bajo efectos del alcohol. Una solución automatizada basada en visión por computadora puede actuar como sistema de alerta temprana para evitar siniestros.

## Objetivo General:

Desarrollar un sistema de detección temprana de somnolencia en conductores, basado en visión por computadora, que permita identificar signos de fatiga ocular mediante una webcam común.

## Objetivos Específicos:

- Detectar y rastrear ojos del conductor en tiempo real.
- Calcular el índice EAR (Eye Aspect Ratio).
- Generar una alerta visual y sonora ante riesgo de microsueños.
- Integrar una interfaz gráfica para facilitar el uso por cualquier persona.

# 2. Diseño de la Solución

## Enfoque Propuesto:

El sistema se basa en el análisis del parpadeo mediante el cálculo del EAR (Eye Aspect Ratio), una métrica derivada de las distancias entre puntos clave de los ojos. Este índice permite detectar cuándo los ojos están cerrados por un tiempo prolongado, lo que puede indicar somnolencia.

## Pasos de Implementación:

1. Captura de video desde webcam.
2. Detección facial usando MediaPipe Face Mesh.
3. Identificación de landmarks oculares.
4. Cálculo del EAR y análisis temporal.
5. Generación de alertas visuales/sonoras cuando el EAR cae por debajo del umbral durante cierto número de frames consecutivos.
6. Interfaz con Tkinter para facilitar la interacción.

## Herramientas y Librerías:

- **OpenCV:** procesamiento de imágenes y video.
- **MediaPipe:** detección facial y landmarks.
- **Scipy:** cálculo de distancias.
- **Tkinter:** interfaz gráfica.
- **playsound:** alarma sonora.
- **NumPy:** soporte de operaciones numéricas.
- **Google Colab:** desarrollo y ejecución colaborativa.

## 3. Implementación

Se desarrolló un script adaptado para poder ejecutarse en Google Colab con explicaciones comentadas en cada bloque y dividido por secciones (setup, funciones, ejecución)

Se integró con Gradio para poder utilizar la interfaz gráfica

## 4. Evaluación y Análisis

Incluiremos:

- Comprobación funcional del sistema con distintos usuarios.
- Evaluación de falsos positivos/negativos.
- Observaciones sobre velocidad de respuesta.
- Observación del rendimiento en distintas condiciones de luz o posiciones.

Limitaciones:

- Posible fallo con accesorios, por ejemplo lentes o mala iluminación.
- Dependencia de una cámara frontal bien posicionada.
- Falta de personalización por usuario (en esta versión).

Mejoras posibles:

- Ajuste dinámico del umbral EAR según cada persona.
- Incorporación de modelos CNN para detectar gestos faciales más complejos.
- Integración con sensores físicos o dispositivos móviles.
- Registro de eventos de somnolencia.