

# Selección de proyecto Computación Visual

## AR Car Tuner & Configurator

### Integrantes:

- Michael Sebastian Caicedo Rosero
- Diego Leandro Rodriguez Diaz
- Gabriel Andres Anzola Tachak
- Sergio David Motta Romero
- Juan Diego Velasquez Pinzon

### 1. Definición

El proyecto AR Car Tuner & Configurator es una experiencia de Realidad Aumentada basada en la web (WebAR) que permite a los usuarios visualizar, personalizar y conducir un vehículo virtual (específicamente un Nissan 240SX) utilizando visión por computadora.

A diferencia de aplicaciones de AR convencionales que solo muestran objetos estáticos, este sistema integra tecnologías como A-Frame, Three.js y AR.js para el renderizado y tracking de marcadores (Hiro), junto con MediaPipe Hands para el reconocimiento de gestos mediante Inteligencia Artificial. Esto permite una interacción multimodal:

- **Personalización (Tuning):** Cambio de color en tiempo real y materiales PBR con reflejos realistas.
- **Conducción Híbrida:** Control mediante joystick virtual en pantalla o mediante gestos manuales (mano abierta para acelerar, puño para frenar, signo de paz para reversa).
- **Audio Procedural:** Generación de sonido de motor sintetizado en tiempo real basado en las RPM y velocidad del vehículo virtual.

### 2. Justificación

Hemos decidido desarrollar esta aplicación porque representa un desafío técnico integral que abarca múltiples áreas de la computación visual moderna: renderizado 3D, realidad aumentada y visión artificial.

La implementación de control por gestos mediante el uso de inteligencia artificial justifica la innovación del proyecto, ya que elimina la dependencia de controles táctiles tradicionales, ofreciendo una experiencia más inmersiva y natural. Además, el uso de tecnologías web (TypeScript, Vite) asegura que la aplicación sea accesible desde cualquier dispositivo móvil o laptop con cámara sin necesidad de instalaciones complejas.

Este proyecto no solo demuestra el dominio de la superposición de objetos en entornos reales, sino que profundiza en la interacción física simulada (física arcade) y la síntesis de audio, aportando un valor académico significativo al combinar gráficos por computadora con procesamiento de imágenes en tiempo real.

### 3. Cronograma de 13 semanas con objetivos SMART

#### Semana 1: Planeación y Configuración del Entorno

- Definir roles específicos (Desarrollo 3D, Lógica IA, UI/UX).
- Configurar el entorno de desarrollo con Vite y TypeScript.
- Buscar y optimizar el modelo 3D de algún vehículo para web.

#### Semana 2: Fundamentos de AR

- Implementar la base del proyecto con A-Frame y AR.js.
- Lograr la detección estable del marcador Hiro.
- Visualizar el modelo 3D sobre el marcador correctamente.

### **Semana 3: Sistema de Personalización (Tuning)**

- Crear la interfaz de usuario (UI) flotante.
- Programar la lógica para el cambio de materiales y colores (PBR).
- Validar la calidad visual de los reflejos (HDR).

### **Semana 4: Físicas y Control Básico**

- Implementar sistema de físicas arcade (velocidad, aceleración, fricción).
- Desarrollar el Joystick Virtual para control táctil en pantalla.
- Pruebas de conducción básica en escritorio.

### **Semana 5: Integración de Visión Artificial (Parte 1)**

- Integrar la librería MediaPipe Hands en el proyecto.
- Lograr la detección de la mano y sus "landmarks" en tiempo real sobre la capa de AR.

### **Semana 6: Integración de Visión Artificial (Parte 2)**

- Programar la lógica de reconocimiento de gestos específicos:
  - Mano Abierta (Acelerar).
  - Puño Cerrado (Frenar).
- Vincular los gestos detectados con el sistema de físicas del coche.

### **Semana 7: Gestos Avanzados y Dirección**

- Implementar el gesto "Amor y Paz" para la reversa.
- Programar la detección de posición lateral de la mano para controlar la dirección (volante).
- Documentar la versión 1.0 del prototipo funcional con IA.

### **Semana 8: Audio Procedural**

- Implementar Web Audio API.
- Crear el sintetizador de motor que varíe el tono (pitch) según la velocidad/RPM.
- Añadir efectos de sonido (UI feedback).

### **Semana 9: Optimización y Rendimiento**

- Optimizar el renderizado para mantener FPS estables en móviles.
- Gestionar el uso de memoria (AR + Detección de manos es intensivo).
- Mejorar la respuesta de los controles para reducir latencia.

### **Semana 10: Pruebas de Usabilidad**

- Validar la experiencia de usuario (¿Es fácil conducir con la mano?).
- Ajustar sensibilidad de la detección de gestos e iluminación.
- Elaborar informe parcial de resultados técnicos.

### **Semana 11: Detalles Finales y Pulido**

- Mejorar la interfaz gráfica y los avisos visuales ("IA Activada").
- Asegurar compatibilidad entre distintos navegadores y dispositivos.
- Validar cumplimiento de objetivos SMART iniciales.

### **Semana 12: Documentación Final**

- Redactar el documento final (detallando el stack tecnológico: Three.js, MediaPipe, etc.).
- Grabar video de demostración del funcionamiento.
- Preparar repositorio para entrega.

### **Semana 13: Entrega y Sustentación**

- Entrega del repositorio completo y desplegable.
- Entrega del documento final.
- Presentación del proyecto funcionando en vivo.

## Referencias

- A-Frame. (s.f.). *A-Frame framework*. <https://aframe.io/>
- AR.js. (s.f.). *AR.js documentation*. <https://ar-js-org.github.io/AR.js-Docs/>
- Google. (s.f.). *MediaPipe Hands*. Google AI for Developers.  
[https://developers.google.com/mediapipe/solutions/vision/hand\\_landmarker](https://developers.google.com/mediapipe/solutions/vision/hand_landmarker)
- Three.js. (s.f.). *Three.js – JavaScript 3D library*. <https://threejs.org/>