Máster Universitario en Ingeniería Informática

Interfaces de Usuario Multimodales

Curso 2025/2026

# Proyecto: Compra-Inteligente


Universidad de Jaén

**Integrantes**

**Oscar Reimar Cedeño Delgado**

**Rubén Prieto García**

# Módulo de Interacción Gestual

## 1. Objetivos del Proyecto

El propósito principal del proyecto **Compra-Inteligente** es el desarrollo de una *Single Page Application* (SPA) de comercio electrónico multimodal que simula una experiencia de supermercado online. Desde la perspectiva de la interfaz de usuario no táctil, el objetivo se centra en la implementación de un sistema de **detección de gestos manuales** mediante visión artificial.

Este sistema debe permitir al usuario realizar el flujo completo de compra sin contacto físico con el periférico de entrada, abarcando:

- Exploración de un catálogo de 50 productos alimenticios.
- Filtrado de productos por categorías (snacks, drinks, dairy, cereals) y clasificación Nutriscore.
- Gestión del carrito de compras (añadir, modificar cantidad y eliminar).
- Finalización del proceso mediante un flujo de *checkout* simulado.

La interacción se realiza mediante la librería MediaPipe HandLandmarker, procesando la señal de vídeo en tiempo real para interpretar comandos gestuales específicos.

## 2. Stack Tecnológico

Para la implementación del módulo de gestos y la infraestructura base de la aplicación, se ha seleccionado la siguiente arquitectura técnica:

- **Lenguaje de Programación:** JavaScript ES6+ (ECMAScript Modules nativos).
- **Motor de Visión Artificial:** MediaPipe Tasks Vision (GestureRecognizer) v0.10.8, cargado dinámicamente vía CDN.
- **Renderizado de Interfaz:** Manipulación directa del DOM con *template strings*, sin uso de frameworks reactivos (React/Vue/Angular).
- **Estilos:** TailwindCSS (vía CDN) complementado con CSS vanilla para efectos de profundidad y animaciones.

- **Gestión de Estado:** Arquitectura centralizada basada en el patrón *Observer* (store.js).
- **Infraestructura:** Aplicación 100% *client-side*, sin *backend* (arquitectura *Serverless* para la lógica de presentación), utilizando datos estáticos en formato JSON.

# 3. Especificación Funcional: Detección y Lógica de Gestos

El módulo gestures.js orquesta la interpretación de *landmarks* manuales y la ejecución de lógica de negocio. El sistema implementa un cursor virtual que refleja la posición de la mano y reconoce los siguientes comandos con sus respectivas reglas de negocio:

## 3.1. Diccionario de Gestos Soportados

- **Thumb_Up (👍):** Acción de añadir una unidad del producto activo al carrito o incrementar su cantidad si ya existe.
- **Thumb_Down (👎):** Acción de decrementar la cantidad de un producto o eliminarlo del carrito.
- **Open_Palm (✋):** Comando de cancelación o retorno. Permite navegar hacia atrás en la máquina de estados (Checkout $\rightarrow$ Cart $\rightarrow$ Browse).
- **Victory (✌):** Comando de confirmación y avance. Permite la transición positiva entre estados (Browse $\rightarrow$ Cart $\rightarrow$ Checkout $\rightarrow$ Finalizar Compra).
- **Closed_Fist (✊):** Acceso a la vista de detalles del producto actualmente enfocado.
- **Italian (□):** Gesto personalizado implementado mediante algoritmo manual. Añade 2 unidades del producto simultáneamente.
    - *Lógica de detección:* Verifica orientación vertical, agrupación de dedos en un centroide reducido y distancia de extensión específica (anti-puño).

- **Swipe Left/Right:** Navegación lateral entre productos en la vista de *Coverflow*.

## 3.2. Mecanismos de Control y Estabilidad

Para garantizar la usabilidad y prevenir falsos positivos, se han implementado los siguientes controles técnicos:

- **Cooldowns Independientes:** Cada gesto posee un temporizador de enfriamiento variable (500ms - 1500ms) para evitar ejecuciones múltiples involuntarias.
- **Histéresis Dinámica (Dynamic Hysteresis):** Para la detección de *swipes*, se aplica un umbral base de 0.15 para la activación y un umbral estricto de 0.35 para contra-movimientos, reduciendo el "ruido" posicional.
- **Zero-Velocity Unlock:** Sistema que resetea la histéresis tras detectar ~10 *frames* de estabilidad manual, facilitando la ejecución de gestos consecutivos.

# 4. Descripción del Diagrama de Flujo de Datos

Para la representación gráfica del sistema, el diagrama debe estructurarse en un modelo de **procesamiento en capas (Pipeline)**, detallando la transformación de la señal de vídeo en mutaciones de estado. La estructura textual del diagrama es la siguiente:

**Nivel 1: Capa de Captura y Pre-procesamiento**

- **Entrada:** Stream de vídeo en tiempo real (vía getUserMedia).
- **Proceso:** Bucle de renderizado mediante requestAnimationFrame (RAF loop).
- **Salida:** *Frames* individuales sincronizados para inferencia.

**Nivel 2: Motor de Inferencia (Machine Learning)**

- **Componente:** MediaPipe GestureRecognizer (v0.10.8).
- **Proceso:** Análisis de landmarks manuales sobre el frame actual.
- **Datos Generados:**
  - *Landmarks:* Coordenadas XYZ de los 21 puntos de la mano.

- *Classifications:* Etiqueta del gesto (ej: "Victory", "Thumb_Up") y puntuación de confianza.

**Nivel 3: Lógica de Interpretación y Filtrado (gestures.js)**

- **Sub-proceso A (Estabilización):** Aplicación de Histéresis Dinámica (umbrales 0.15/0.35) para vectores de movimiento y Zero-Velocity Unlock para reset de estados.
- **Sub-proceso B (Algoritmos Custom):** Detección geométrica manual para gestos no nativos (Gesto "Italian" mediante análisis de centroide y extensión).
- **Sub-proceso C (Gestión de Eventos):** Verificación de tiempos de *Cooldown* (500ms - 1500ms) para evitar rebotes.

**Nivel 4: Capa de Estado y Persistencia (store.js)**

- **Acción:** Disparo de métodos mutadores (addToCart, setMode, setFilter).
- **Reacción:** Notificación a suscriptores mediante patrón *Observer*.

**Nivel 5: Capa de Presentación (ui.js / DOM)**

- **Feedback Visual:** Renderizado del cursor virtual (espejo), actualización de contadores, y animaciones de confirmación (modal de *checkout*, notificaciones flotantes).

# 5. Dificultades Encontradas y Soluciones Técnicas

Durante la implementación del módulo de interacción gestual, se identificaron desafíos críticos relacionados con la estabilidad de la detección y la fluidez de la experiencia de usuario. A continuación, se detallan las problemáticas y las soluciones algorítmicas aplicadas:

## 5.1. Inestabilidad en la Navegación Vectorial (Efecto "Jitter")

La detección de movimientos de *Swipe* (desplazamiento lateral) presentaba inestabilidad debido al ruido natural en la captura de coordenadas de la mano, lo que generaba falsos positivos en cambios de dirección leves.

- **Solución Aplicada:** Implementación de un sistema de **Histéresis Dinámica**. Se establecieron umbrales diferenciados: un umbral base (0.15) para la activación inicial del movimiento y un umbral exigente (0.35) para los contra-movimientos. Esto filtra el ruido posicional y asegura que solo los gestos intencionales disparen la navegación.

## 5.2. Activaciones Involuntarias (Rebotes)

La alta tasa de refresco del reconocimiento de imágenes provocaba que un solo gesto (ej: *Thumb_Up*) fuera interpretado múltiples veces en milisegundos, añadiendo al carrito cantidades no deseadas de producto.

- **Solución Aplicada:** Desarrollo de un sistema de **Cooldowns Independientes**. Cada tipo de gesto posee un temporizador de bloqueo específico (entre 500ms y 1500ms) que impide la re-ejecución inmediata tras una detección exitosa, garantizando una interacción unitaria.

## 5.3. Bloqueo en Gestos Consecutivos

El sistema de histéresis estricto, aunque solucionaba la inestabilidad, ocasionalmente impedía realizar múltiples *swipes* rápidos si la mano no volvía a una posición neutra perfecta.

- **Solución Aplicada:** Algoritmo de **Zero-Velocity Unlock**. El sistema monitorea la estabilidad de los *landmarks*; tras detectar aproximadamente 10 *frames* de estabilidad posicional, resetea forzosamente los acumuladores de histéresis,

"desbloqueando" el sistema para permitir un nuevo gesto de navegación inmediato.

## 5.4. Detección de Gestos Personalizados (Gesto "Italian")

La librería estándar de MediaPipe no incluye el gesto "Italian" (dedos juntos hacia arriba) en su modelo pre-entrenado.

- **Solución Aplicada:** Implementación de un algoritmo de **Geometría Computacional Manual**. Se desarrolló una lógica personalizada que verifica simultáneamente tres condiciones vectoriales: la orientación vertical de la mano, la agrupación de las puntas de los dedos en un centroide reducido y una distancia de extensión específica para diferenciarlo de un puño cerrado (*Closed_Fist*).

# 6. Restricciones y Requisitos del Sistema

Para la correcta ejecución del módulo de gestos, el sistema donde se implemente debe cumplir las siguientes restricciones técnicas:

## 6.1. Requisitos de Hardware (Dispositivo)

- **Cámara Web:** Es indispensable un dispositivo de captura de vídeo funcional accesible por el navegador.
- **Potencia de Procesamiento:** El dispositivo debe tener capacidad suficiente para ejecutar inferencia de ML en tiempo real (GPU/CPU con soporte de aceleración gráfica recomendada para evitar *lag* en el cursor).

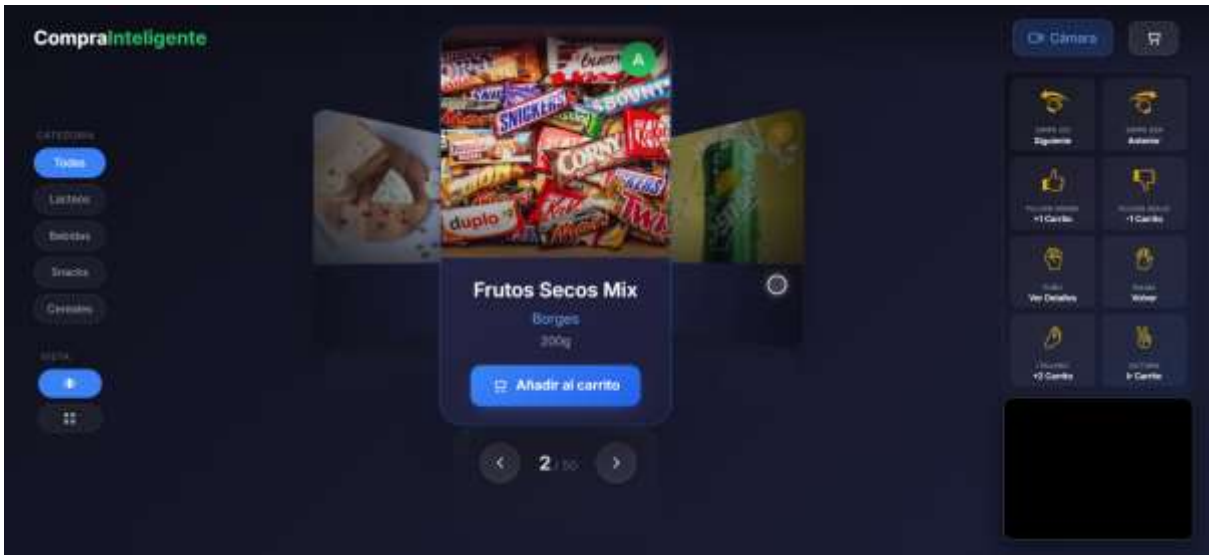
## 6.2. Requisitos de Software (Entorno)

- **Navegador Moderno:** Debe soportar **ES Modules** nativos, la API **MediaDevices** (getUserMedia) para acceso a la cámara y **WebAssembly (WASM)** para la ejecución del modelo MediaPipe.

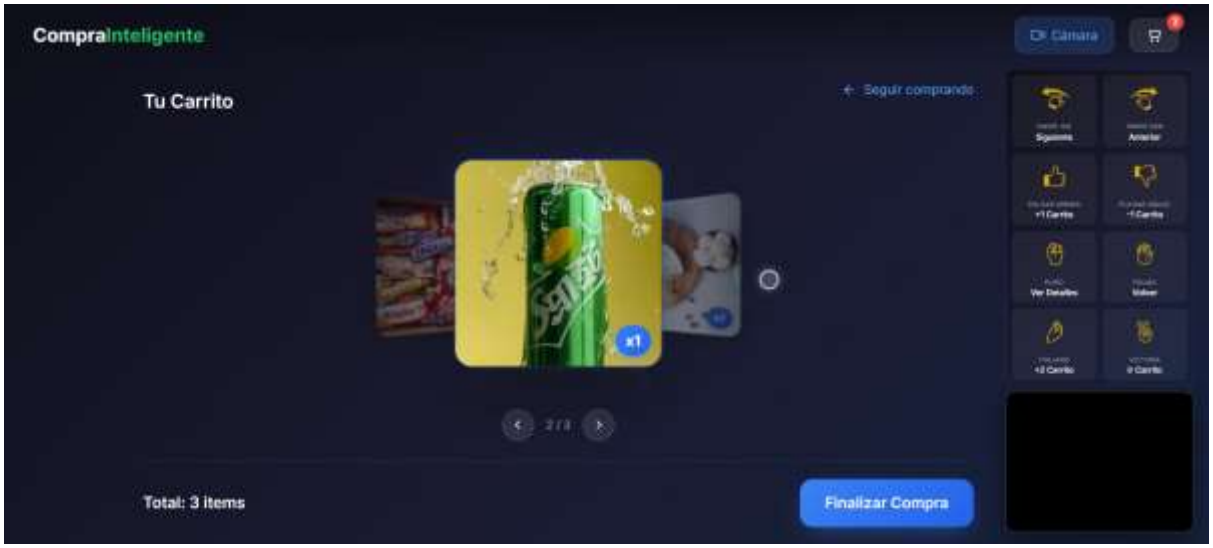
- **Conectividad:** Se requiere conexión a internet para la carga inicial de las librerías CDN y el modelo de visión (.task), aunque el sistema posee *fallbacks* limitados para modo *offline*.

# 7. Capturas de resultados y pruebas ejecutadas

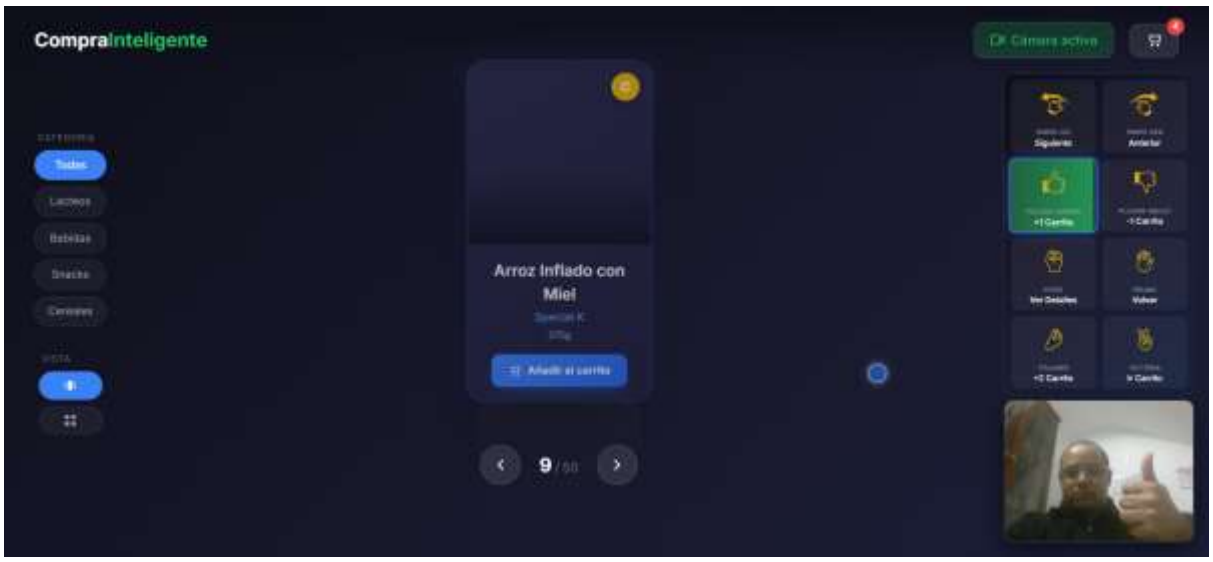
- **Pantalla principal del sistema**




- **Carrito de compras**

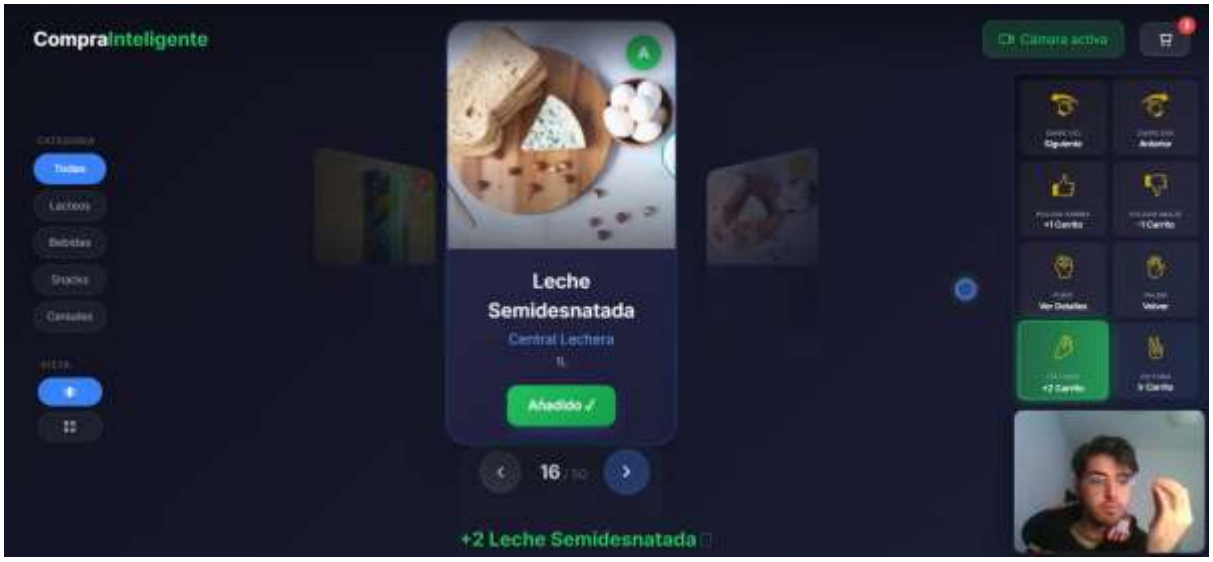


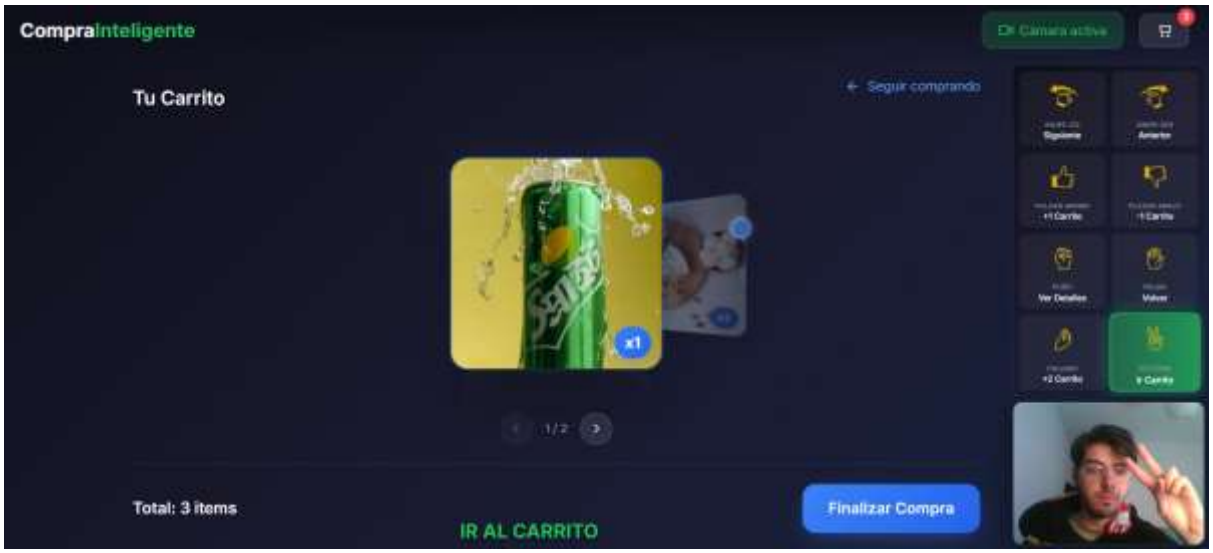
- **Gestos**

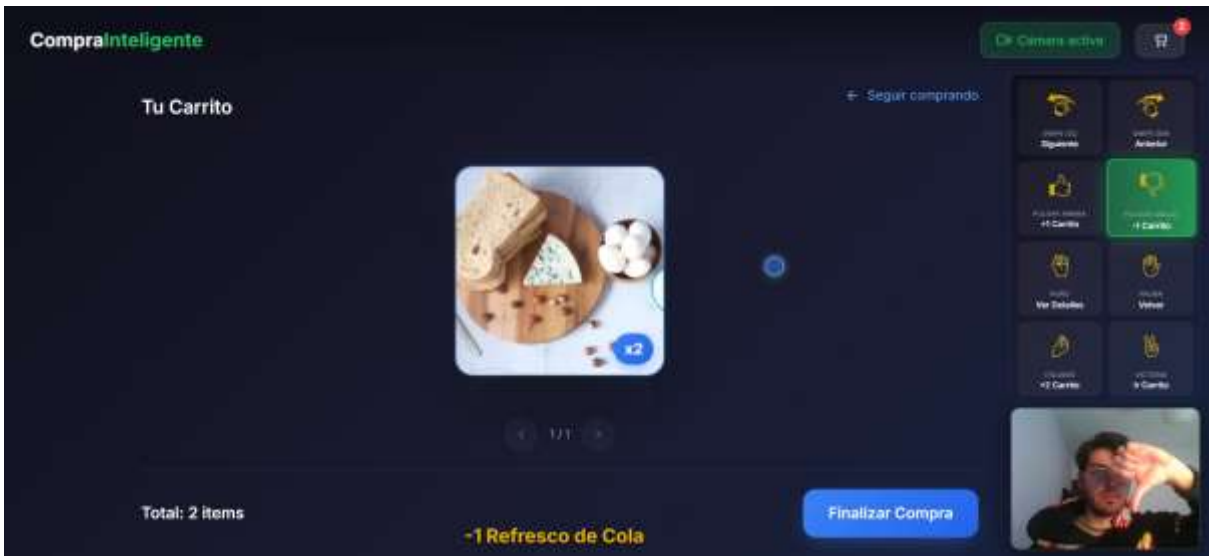

CompraInteligente
Arroz Inflado con Miel
Añadir al carrito
9 / 50


CompraInteligente
Volver
Barritas de Cereales
Hero
Añadir al carrito
Detalles: Barritas de Cereales


CompraInteligente
Barritas de Cereales
Hero
Añadir al carrito
10 / 50
VOLVER A TIENDA


CompraInteligente
Cámara activa
Leche
Semidesnatada
Central Lechera
Añadido ✓
16
+2 Leche Semidesnatada


CompraInteligente
Cámara activa
Tu Carrito
Seguir comprando
x1
Total: 3 items
IR AL CARRITO
Finalizar Compra


CompraInteligente
Cámara activa
Tu Carrito
Seguir comprando
x2
Total: 2 items
-1 Refresco de Cola
Finalizar Compra