BITÁCORA — Día 1 · Proyecto GestureControl

Fecha: *[coloca la fecha de hoy]*
Responsable: *[tu nombre]*
Proyecto: *GestureControl — Control por gestos con cámara y ML (MediaPipe / TensorFlow.js)*

1. Inicio del proyecto

Hoy se definió que el proyecto final será **GestureControl**, un sistema de control por gestos usando cámara + inteligencia artificial, ejecutado en navegador (web), compatible con móviles/laptops.

Decisiones clave de diseño:

- La app será web para poder correr rápido sin instalar nada.
- Usaremos **TensorFlow.js + MediaPipe Hands** como modelo de detección.
- Tendremos 3 gestos principales:
    - **Puño**
    - **Mano abierta**
    - **Apuntar**

- Cada gesto realizará una acción configurable:

    - *Play/Pause*
    - *Next*
    - *Prev*

- Se incluirá una **interfaz gráfica**, bitácora interna, logs y calibración.

2. Configuración inicial del proyecto

Se creó el proyecto en una carpeta local:

```
GestureControl/
 index.html
 app.js
 style.css
 manifest.json
```

Decidimos correr un servidor local usando:

```
npx http-server web -p 8080
```

---

3. Primeros problemas detectados

⚠ ***Error 1: "npx no se reconoce como un comando"***

- No estaba instalado Node.js.
- Se instaló Node.js desde la página oficial.
- Después ya funcionaron `node`, `npm`, `npx`.

Error 2: "La ejecución de scripts está deshabilitada (PSSecurityException)"

- PowerShell no permitía ejecutar scripts (`npx.ps1`).
- Se arregló ejecutando:

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned

Después de eso el servidor local funcionó correctamente.

4. Problemas al iniciar la aplicación web

Tras abrir la app en el navegador aparecieron múltiples errores:

Error en consola:

Init error TypeError: i.loadGraphModel is not a function

→ Significa que TensorFlow.js no estaba cargando correctamente la función necesaria para MediaPipe.

Error: "handPoseDetection is not defined"

→ El script del modelo **no se estaba cargando** antes de `app.js`.

Error: Requested texture size [0x0] is invalid

→ La cámara no estaba lista y la imagen regresaba tamaño 0x0.

Manifest / íconos no encontrados

icon-192.png 404
favicon.ico 404

→ No afecta el funcionamiento, solo advertencias.

5. Corrección del `index.html`

Se reconstruyó el archivo para garantizar:

- Orden correcto de carga.
- Inclusión de TensorFlow bundle completo.
- Carga del backend WebGL.
- Carga del modelo correcto de MediaPipe Hands.

Se dejó el HTML final así:

- `tf.min.js`
- `tf-backend-webgl.min.js`
- `hand-pose-detection.min.js`
- `app.js` (al final)

Esto garantiza que las librerías estén definidas antes de inicializar el detector.

6. Depuración y análisiS

Para verificar que el navegador estuviera cargando las librerías, ejecutamos en consola:

typeof handPoseDetection
typeof tf
tf.loadGraphModel

Esto es fundamental para asegurarse que TensorFlow.js y MediaPipe están accesibles.

7. Cámara no detectada

Apareció esta alerta desde `requestCamera()`:

No se pudo acceder a la cámara. Revisa permisos y que la página esté en localhost/https.

Se identificaron posibles causas:

- Falta de permisos.
- El navegador no estaba en HTTPS (localhost sí funciona).
- Cámara usada por otra app.
- Resolución inválida.

La cámara finalmente se activó, aunque con retraso.

8. Gestos no detectados

Aunque el video funcionó, la app no reconocía la mano.

Posibles causas analizadas:

- Modelo no cargado correctamente.
- `handPoseDetection.createDetector()` devolvió error.
- Keypoints no estaban siendo generados.
- El código estaba llamando la API incorrecta de MediaPipe.

Esto llevó a reescribir la sección de carga del modelo.

9. Avance: Revisión profunda del script de MediaPipe

Se descubrió que:

- La versión usada de MediaPipe Hands en CDN era incompatible.
- El modelo “lite” o “full” necesita configurarse correctamente.
- La API del detector cambió entre versiones.

Por eso surgió el error:

handPoseDetection is not defined

Se corrigió el CDN a:

https://cdn.jsdelivr.net/npm/@tensorflow-models/hand-pose-detection@0.0.7/dist/hand-pose-detection.min.js

10. Próximos pasos identificados

1. Confirmar si `handPoseDetection.createDetector()` funciona desde consola.
2. Probar el modelo MediaPipe con un ejemplo mínimo.
3. Agregar logs visuales cuando se detecten manos.
4. Ajustar heurísticas de detección de gestos.
5. Optimizar parámetros (distancias, suavizado, cooldown).
6. Agregar acciones funcionales (reproducir audio/video, control de PPT, etc.).

RESUMEN DEL DÍA

Hoy se avanzó muchísimo:

✔ Se definió el proyecto GestureControl

✔ Se configuró el entorno

✔ **Se instaló y configuró Node.js**

✔ **Se logró correr el servidor local**

✔ **Se crearon index/app/style/manifest**

✔ **Se realizó la primera versión del código completo**

✔ **Se detectaron los principales errores del modelo**

✔ **Se corrigió el `index.html` completamente**

✔ **Se avanzó en depuración y pruebas del detector**

Aunque seguimos con el error final de `handPoseDetection`, ya está mucho más claro el origen y la solución está encaminada.

Estado actual del proyecto

Módulo	Estado
Interfaz web	✔ 100% funcional
Cámara	✔ Carga pero a veces falla
Carga de TensorFlow	✔ Correcta
Carga de MediaPipe Hands	❌ Fallando
Detección de keypoints	❌ No funcionando
Gestos	❌ No funcionando
Log / Bitácora interna	✔ Completa
Acciones configurables	✔
Toast UI	✔
Manifest	✔ (con warnings de iconos)

Conclusión

Aunque hubo muchos errores técnicos, **se avanzó de forma enorme**.
Se corrigió todo lo relacionado con:

- entorno
- Node/npm
- servidor
- cámara
- index.html
- orden de scripts
- estructura de archivos
- preparación del detector