## 🔶 FASE 1: Inicio y contexto

🗣️ “Hola Carlos, como comentaste el primer día, tenemos libertad total y no teníamos que seguir una hoja de ruta estricta. Por eso nos organizamos internamente como si fuéramos un pequeño equipo técnico.”

🗣️ “En un principio intentamos trabajar con WordPress usando LocalWP, pero surgieron incompatibilidades técnicas que nos hicieron replantearlo.”

“LocalWP es ideal si haces una web típica de empresa con textos e imágenes, pero cuando quieres meter un visor 3D y controlarlo a medida, lo más eficaz y flexible es trabajar con código nativo, como hemos hecho con Vite + Three.js.”

✅ Muestra si quieres brevemente algún resto de esa etapa: instalación fallida, captura, carpeta abandonada.

🗣️ “Decidimos entonces empezar desde cero con código puro (HTML, CSS, JS), para tener más control y flexibilidad, sobre todo pensando en el visor 3D que nos pediste.”

## 🔶 FASE 2: Primer prototipo funcional (estructura general)

🗣️ “En los primeros días nos centramos en construir la estructura básica de la web: diseño del index, distribución de carpetas, sistema de navegación y sistema para cargar modelos 3D.”

### ✅ El diseño en Figma

🗣️ “Antes de pasar directamente al código, hicimos una primera propuesta visual en Figma para organizarnos. No es un diseño finalista, pero nos sirvió como guía para entender mejor la estructura y jerarquía del visor.”

“En vez de usar WordPress, que nos dio problemas, montamos un entorno local más profesional con Node.js y Vite, que nos permite usar modules, importar Three.js correctamente, y hacer pruebas sin necesidad de servidor backend por ahora.”

✅ Abre index.html y explica brevemente su simplicidad: diseño ligero, acceso al visor.

🗣️ “A nivel de código, estructuramos desde el principio el visor (viewer.html) en base a un contenedor Three.js, con módulos independientes como scene.js, viewerStyle.js, y sus propios estilos.”

## 🔶 FASE 3: Carga de modelos y visor

🗣️ “Una vez montada la base, nuestro objetivo fue permitir al usuario **arrastrar un modelo GLB o GLTF** y que este se cargue de forma automática en el visor.”

✅ Demuestra el flujo de carga con un modelo.

🗣️ “Aquí se usa sessionStorage para mantener el modelo cargado entre páginas sin necesidad de servidor.”

✅ (Si procede) Abre scene.js → función loadModel() y autoLoadFromSession() (solo nombrarlas por encima).

## 🔶 FASE 4: Controles visuales y menú

🗣️ “Lo siguiente fue añadir un menú lateral donde se pueden controlar propiedades del modelo: color, metalness, roughness…”

✅ Abre menú de opciones en vivo. Muestra sliders y cómo afectan al modelo.

🗣️ “Guardamos estos valores también en sessionStorage para mantener consistencia si se recarga.”

✅ (Opcional) Nombra actualizarModelo() y restaurarMaterialesOriginales().

## 🔶 FASE 5: Añadiendo interacción avanzada

🗣️ “Queríamos que no fuera solo visual, sino también **interactivo**. Por eso añadimos...”

* ✅ **Coordenadas de cámara** (parte inferior)
* ✅ **Helpers visuales** (cuadrícula, ejes, luz)
* ✅ **Cambio de fondo HDRI** con Three.js + RGBELoader
* ✅ **Rotación con teclas Q y E** y **slider de rotación sincronizado**

🗣️ “Intentamos que el usuario tenga libertad para observar el modelo en cualquier entorno o posición.”

✅ Muestra cambios en vivo de HDRIs, el uso del slider, y rotación con teclado.

## 🔶 FASE 6: Diseño y accesibilidad

🗣️ “Además del código, cuidamos la parte visual: que sea claro, intuitivo, y usable en distintos tamaños de pantalla.”

✅ Muestra el estilo oscuro, el botón de volver, el diseño responsive del panel lateral, etc.

🗣️ “También explicamos en pantalla ciertas acciones clave, como la posibilidad de mover el modelo con SHIFT + clic.”

## 🔶 CIERRE: visión y próximos pasos

🗣️ “Todo esto lo hemos desarrollado en apenas unos días. Nos motiva mucho lo que estamos aprendiendo y queremos seguir avanzando.”

🗣️ “Ahora mismo tenemos una base muy sólida sobre la que podríamos implementar el siguiente gran paso: las **capas activables del modelo**, como nos propusiste inicialmente.”

✅ Muestra brevemente la estructura de carpetas (/js, /styles, /assets, etc.) y el uso de Vite para cargar módulos de forma ligera.

## 🔸 OPCIONAL: si tienes tiempo

📂 Muestra el archivo Diario de trabajo.docx o SESION I.docx como prueba documental de las sesiones diarias y el seguimiento.