

AEV3 Tienda PWA (Frontend)

2º DAM – Sostenibilidad aplicada al sistema productivo



The image displays two side-by-side screenshots of a web application interface, likely a Progressive Web Application (PWA) for a store. Both screenshots have a dark header bar at the top with the text "Tienda Demo" and "Galería de productos con imágenes y videos". A "Ver Carrito" button is located in the top right corner of both.

Left Screenshot: This version shows a simple product grid. It includes a search bar labeled "Buscar productos...". Below the search bar are three items: "Producto 1" (a black camera), "Producto 2" (a red and white striped object), and a "Video del Producto" card featuring a video thumbnail of a camera with controls like play/pause, volume, and a progress bar showing "0:00". Each item has a price (\$25.00 or \$45.00) and an "Agregar al carrito" button.

Right Screenshot: This version is more polished. It includes a search bar labeled "Buscar productos...". The product cards are arranged in a grid with a larger margin. The "Video del Producto" card is styled as a video player with a play button, a timestamp of "0:00", and volume controls. The overall layout is cleaner and more modern.

Etapa
HTML_Inicial
PWA_Inicial
Tienda_01_PWA
Tienda_02_Carrito
Tienda_03_Buscador

Qué agregas
Solo HTML/CSS básico
manifest + sw.js + registro
Convertir tienda en PWA real
script.js + carrito.html + localStorage
Input + búsqueda en script.js

Qué aprendes
Estructura de tienda
Cómo funciona una PWA
Offline + instalación
Persistencia
Interactividad avanzada



- index.html: una **galería estática** de productos (imágenes/vídeos, títulos, precios...).
- style.css: estilos básicos de la tienda (layout, tarjetas de producto, cabecera...).
- script.js: prácticamente vacío, solo un console.log("Galería cargada").

👉 Es una **web normal**, sin PWA, sin carrito, sin buscador. Tu objetivo aquí es entender la estructura HTML y los estilos, porque todo lo demás se va a ir construyendo encima de esto.

Tienda Demo

Galería de productos con imágenes y videos

Producto 1
Producto 1
\$25.00

Producto 2
Producto 2
\$45.00

Video del Producto
\$30.00



Objetivo: aprender los ingredientes mínimos de una PWA.

1.Crear un archivo **manifest.json** con:

1. nombre, short_name
2. start_url
3. display (standalone)
4. Iconos

2.Agregar a **index.html**:

1. <link rel="manifest" href="manifest.json">
2. <meta name="theme-color">

3.Crear un archivo **sw.js** con:

1. Un array de archivos a cachear
2. Evento `install` para cachear esos archivos
3. Evento `fetch` para servir desde cache o red

4.Registrar el service worker en **index.html**.

✓ Resultado

Una **PWA muy simple** que puede:

- instalarse en el móvil o escritorio
- funcionar offline mostrando al menos una página

```

PWA_Inicial > JS sw.js > ...
1  const CACHE_NAME = "mi-pwa-v1";
2  const urlsToCache = ["index.html", "manifest.json", "icons/icon-192.png"];
3
4  self.addEventListener("install", (event) => {
5    event.waitUntil(
6      caches.open(CACHE_NAME).then((cache) => cache.addAll(urlsToCache))
7    );
8  });
9
10 self.addEventListener("fetch", (event) => {
11   event.respondWith(
12     caches
13       .match(event.request)
14       .then((response) => response || fetch(event.request))
15   );
16 });

```



Objetivo: convertir la tienda HTML en una PWA real.

1.Copiar el HTML y CSS de **HTML_Inicial** como base.

2.Añadir:

1. <link rel="manifest">
2. <meta name="theme-color">
3. registro de sw.js

3.Ajustar manifest.json para que represente la tienda:

1. name: "Mi Tienda PWA"
2. icons de tienda

4.Incluir en sw.js estos archivos:

1. index.html
2. style.css
3. manifest.json
4. imágenes o logos

Resultado

Tu tienda ahora es una **PWA instalable**, con la galería de productos funcionando offline (según caché).

The screenshot illustrates the process of turning a standard HTML website into a Progressive Web Application (PWA). At the top, a browser window displays a prompt to 'Install app' for 'Tienda Demo' from '127.0.0.1:5500'. A red arrow points to the 'Install' button. Below the prompt, the main content area shows the 'Tienda Demo' PWA, which features a dark header with the title 'Tienda Demo' and a subtitle 'Galería de productos con imágenes y videos'. The main content area displays two product cards: 'Producto 1' (\$25.00) and 'Producto 2' (\$45.00), each with a small image and a price. Below the products is a video player titled 'Video del Producto' with a price of \$30.00. To the right of the main content, there is a sidebar with the following options: 'App info' (with the URL 127.0.0.1:5500), 'Copy URL', 'Open in Brave', and 'Uninstall Tienda Demo...'. The overall layout is clean and modern, typical of a PWA user interface.



Objetivo: añadir un carrito persistente con localStorage.

1.Añadir un archivo script.js.

2.Crear funciones:

1. addToCart (name, price, image)
2. Guardar los productos en localStorage
3. Leer el carrito desde localStorage

3.Crear carrito.html que:

1. muestra los productos guardados
2. calcula el total
3. incluye botón “Vaciar carrito”

4.En index.html, añadir botón a cada producto:

```
<button onclick="addToCart('Nombre', 9.99, 'ruta.jpg')">Añadir al carrito</button>
```

5.Actualizar sw.js para cachear:

1. carrito.html
2. script.js
3. imágenes si quieres que el carrito opere offline

Resultado

Tu tienda:

- añade productos al carrito
- guarda todo en localStorage
- puede mostrar el carrito en una página separada
- sigue siendo PWA

Key	Value
cart	[{"name": "Producto 1", "price": 25, "image": "assets/producto1.jpg"}, {"name": "Producto 2", "price": 45, "image": "assets/producto2.jpg"}]



Objetivo: **añadir un buscador de productos en tiempo real.**

1.Añadir un <input> en index.html:

```
<input id="searchInput" type="text" placeholder="Buscar..." oninput="searchProducts()">
```

2.En script.js, crear la función:

```
function searchProducts() {
  const text = document.getElementById("searchInput").value.toLowerCase();
  const products = document.querySelectorAll(".product");

  products.forEach(product => {
    const name = product.querySelector("h3").textContent.toLowerCase();
    product.style.display = name.includes(text) ? "block" : "none";
  });
}
```

No necesitas modificar manifest ni service worker, solo si agregas nuevos archivos.

✓ Resultado

Tu tienda PWA ahora tiene:

- Búsqueda en tiempo real
- Carrito persistente
- Instalación y funcionamiento offline

The screenshot shows the header of the PWA with the title 'Tienda Demo' and a subtitle 'Galería de productos con imágenes y videos'. Below the header is a search bar containing the text 'vide'. A navigation bar at the bottom includes a 'Ver Carrito' button.

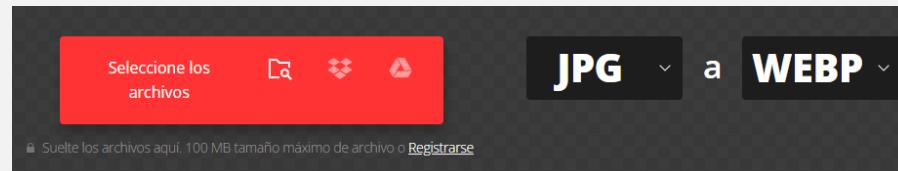
The screenshot displays a product card for a black Fujifilm camera with a lens. A video player overlay is visible, showing a preview of the camera. The video controls include a play button, a timestamp '0:00', a volume icon, and a full-screen icon. Below the video, the text 'Video del Producto' and the price '\$30.00' are shown, along with a 'Agregar al carrito' button.

The screenshot shows the footer of the PWA with a green bar containing the text 'Ver Carrito'.



Formato	Peso	Calidad	Transparencia	Animación	Compatibilidad	Uso recomendado
WebP	⭐ Muy ligero	Alta	Sí	Sí	Muy alta (todos los navegadores modernos)	Fotos, gráficos, web en general
AVIF	⭐⭐ Ultra ligero	⭐ Alta–muy alta	Sí	Sí	Media–alta	Máxima optimización de peso y calidad
JPEG/JPG	Medio	Alta (fotos)	No	No	Total	Fotografías tradicionales
PNG	Alto	Muy alta (sin pérdida)	Sí	No	Total	Logos, iconos, gráficos con transparencia
SVG	Muy ligero	Perfecta (vector)	Sí	Sí (limitado)	Total	Logos, iconos, ilustraciones escalables
GIF	Alto	Muy baja	Sí (limitado)	Sí	Total	Animaciones simples (pero WebP recomendado)

<https://convertio.co/es/jpg-webp/>





Formato	Peso	Calidad	Códecs comunes	Compatibilidad	Ventajas	Desventajas	Uso recomendado
MP4 (H.264)	Medio	Alta	H.264 / AAC	⭐ Total	Compatible con todo	No es el más ligero	Videos para cualquier navegador
WebM	⭐ Ligero	Alta	VP8/VP9/AV1	Muy alta (menos en Safari antiguo)	Archivo más pequeño, moderno	Compatibilidad no absoluta	Web optimizada, videos ligeros
Ogg/Theora (OGV)	Medio	Media	Theora / Vorbis	Media	Software libre	Poco usado, menor calidad	Alternativa abierta, poco común

<https://www.freeconvert.com/es/video-converter>

