```
const CACHE_NAME = "islandia-2026-v2";

const CORE = [
  "./",
  "./index.html",
  "./manifest.json",
  "./icon-192.png",
  "./icon-512.png"
];

// Para que se vea bien offline (CDN)
const EXTERNAL_ASSETS = [
  "https://cdn.tailwindcss.com",
  "https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css",

"https://fonts.googleapis.com/css2?family=Inter:wght@300;400;600;800&family=JetBrains+
Mono:wght@500&display=swap"
];

self.addEventListener("install", (event) => {
  event.waitUntil(
    caches.open(CACHE_NAME).then((cache) => cache.addAll(CORE))
  );
  self.skipWaiting();
});

self.addEventListener("activate", (event) => {
  event.waitUntil(
    caches.keys().then((keys) =>
      Promise.all(keys.map((k) => (k !== CACHE_NAME ? caches.delete(k) : null)))
    )
  );
  self.clients.claim();
});

self.addEventListener("message", async (event) => {
  const data = event.data || {};

  if (data.type === "CACHE_DAY") {
    const urls = Array.isArray(data.urls) ? data.urls : [];
    const cache = await caches.open(CACHE_NAME);

    const toCache = [...new Set([...CORE, ...EXTERNAL_ASSETS, ...urls])];

    for (const url of toCache) {
      try {
        await cache.add(url);
      } catch (e) {
        // Si algún CDN no deja cachear, no tumbamos todo
      }
    }

    event.source?.postMessage({ type: "CACHE_DAY_DONE" });
  }
});

// Cache-first
self.addEventListener("fetch", (event) => {
  event.respondWith(
    caches.match(event.request).then((cached) => {
      return cached || fetch(event.request).catch(() => cached);
    })
  );
});
```