PWA
PROGRESSIVE WEB APPS

Angular

ANGULAR

PWA

ANGULAR

- Las Aplicaciones Web Progresivas (o *PWAs* de "*Progressive Web Apps*") son aplicaciones que utilizan tecnología web (HTML, CSS y JS) pero que se comportan como aplicaciones "nativas" en los dispositivos móviles.
- Los pilares del diseño de PWA:
  - Capacidades extendidas: las aplicaciones web tienen un conjunto de capacidades que no paran de crecer, acercándose cada vez más a las de las aplicaciones nativas.
  - Confiables. Las PWA deben ser rápidas y seguras, proporcionando una experiencia de usuario agradable. Deben poder funcionar incluso sin conexión.
  - Instalable. Debe poder "sacarse" del navegador para ejecutarse de manera independiente.
- Una aplicación PWA tiene las siguientes características:
  - Uso de un contexto seguro (HTTPS).
  - Utiliza "service workers".
  - Dispone de un archivo manifest.

ANGULAR

- Ventajas de las PWA:
  - Multiplataforma.
  - Rapidez. En la carga y la ejecución.
  - Actualización permanente.
  - Segura.
  - Indexable y enlazable.
  - Funciona *offline*.
  - Acceso al hardware.
  - Experiencia de usuario similar a las aplicaciones nativas.

ANGULAR

- Algunos ejemplos aplicaciones PWA:
  - Telegram
  - Dev.to
  - Twitter.
  - 2048 (videojuego)
  - Starbucks
  - Trivago
  - Tinder

ANGULAR

- Instalación:
  - **ng add @angular/pwa**
    - Convierte una aplicación Angular en una PWA (Progressive Web App) agregando automáticamente todo lo necesario:
      - Instala el paquete @angular/service-worker
      - Activa el Service Worker en angular.json
      - Crea el archivo ngsw-config.json
      - Agrega el manifest.webmanifest
      - Agrega los íconos de instalación PWA
      - Registra el Service Worker en app.module.ts (si aplica)

ngsw-config.json recoge los archivos que se guardan en caché para que la app funcione offline

ANGULAR

- APP SHELL:
  - Es la estructura básica de la app.
  - Se carga inicialmente.
  - Contiene la UI y el esquema de navegación principal.
  - Permite utilizar la aplicación antes de que esté completamente disponible.
  - Mejora la percepción de velocidad.

ANGULAR

- APP SHELL:
  - Agregar App Shell:

```
ng generate app-shell
CREATE src/main.server.ts (300 bytes)
CREATE src/app/app.config.server.ts (527 bytes)
CREATE src/app/app.routes.server.ts (174 bytes)
CREATE src/app/app-shell/app-shell.spec.ts (566 bytes)
CREATE src/app/app-shell/app-shell.ts (208 bytes)
CREATE src/app/app-shell/app-shell.css (0 bytes)
CREATE src/app/app-shell/app-shell.html (25 bytes)
UPDATE angular.json (2576 bytes)
UPDATE tsconfig.app.json (464 bytes)
UPDATE package.json (1238 bytes)
UPDATE src/app/app.config.ts (541 bytes)
```

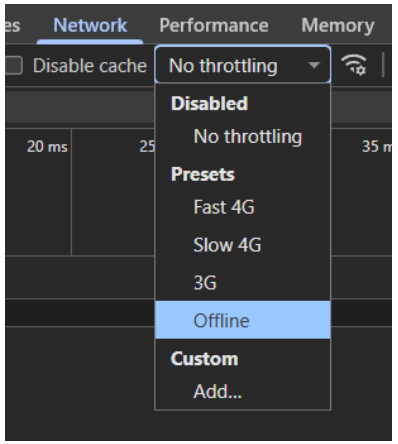
ANGULAR

- SERVICE WORKER
  - Añadir un service worker a una aplicación Angular es uno de los pasos para convertirla en una PWA.
  - En su forma más simple, es un script que se ejecuta en el navegador y administra el almacenamiento en cache de una aplicación.
  - Funciona como un proxi de red, interceptando todas las peticiones HTTP salientes y decidiendo cómo responderlas. Proporciona independencia de la red.
  - El service worker es el elemento que primero se carga de una aplicación.
  - El service worker permanece activo aún con la pestaña del navegador cerrada.
  - El Service Worker se añade al proyecto cuando se ejecuta el comando **ng add @angular/pwa**

ANGULAR

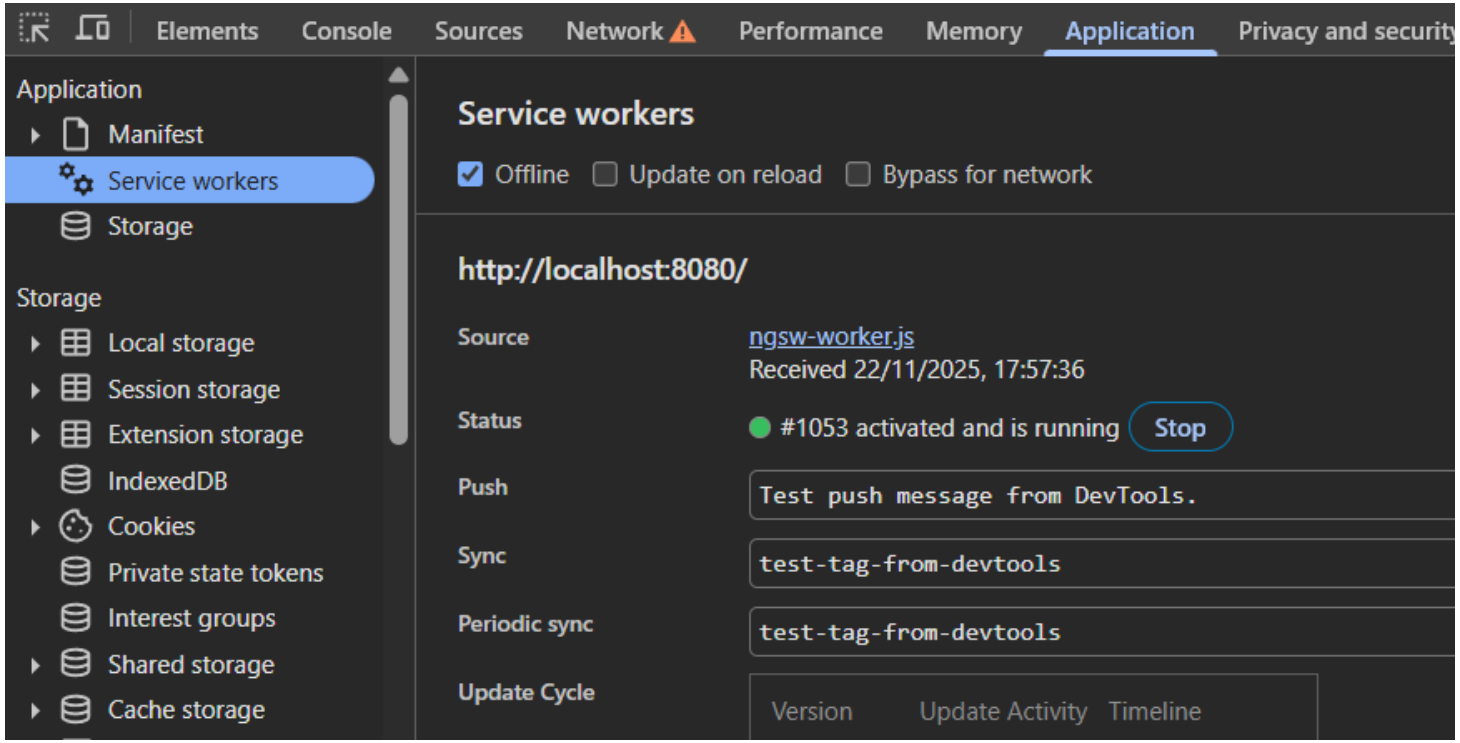
- **ENTORNO DE PRUEBAS:**
  - Construida la app con:
  - `ng build --configuration production`
  - Levantar un servidor Angular:
  - `ng serve --configuration=production`
  - Alternativa: Levantar un servidor local (-c-1 --> --cache -1 --> SIN CACHÉ)
  - `npx http-server -p 8080 -c-1 dist/`*`nombre-app`*`/browser/`
  - Desconectar la red en el navegador:

Soporte service worker



ANGULAR

- **ENTORNO DE PRUEBAS:**
  - Monitorización del service worker:



ANGULAR

- PWA FALLBACK
  - Es la página que se muestra cuando no hay conexión (modo offline) y la ruta indicada no se encuentra en la caché.
  - Por defecto, es la entrada **index** del fichero ngsw-config.json:
    - "index": "/index.html“.
  - Otras estrategias más complejas:
    - Crear un componente específico para mostrar un mensaje amigable.
    - Configurar el routing de Angular para que cuando se vaya a index.html se verifique si hay conexión. Si no la hay, se muestra el componente de notificación de offline.

ANGULAR

- Compilación:
  - Eliminar estás líneas de angular.json para que no se construya el Proyecto con SSR:
    - `"server": "src/main.server.ts",`
    - `"outputMode": "static"`
  - Crear el "build":
    - `ng build --base-href /carpeta/ --configuration production`

ANGULAR

- Fichero manifest.webmanifest
  - Contiene información sobre la aplicación.
  - Se construye automáticamente cuando se agrega @angular/pwa a la aplicación.
  - Se encuentra en la carpeta public.
  - https://developer.mozilla.org/es/docs/Web/Progressive_web_apps/Manifest

ANGULAR

- Uso como aplicación nativa:
  - Framework IONIC → Dispositivos móviles.
    - https://ionicframework.com/
  - Capacitor de IONIC → Proporciona acceso al hardware del dispositivo.
    - https://capacitorjs.com/
  - Electron → Aplicaciones de escritorio.
    - https://www.electronjs.org/
  - TWA → Trusted Web Activities:
    - https://developer.android.com/develop/ui/views/layout/webapps/trusted-web-activities?hl=es-419

ANGULAR

USO DE LIGHTHOUSE

ANGULAR

“Lighthouse es una herramienta automatizada de código abierto que te ayudará a mejorar la calidad de las páginas web. Puedes ejecutarla en cualquier página web, ya sea pública o que requiera autenticación. Realiza auditorías de rendimiento, accesibilidad, SEO y mucho más.

Puedes ejecutar Lighthouse en DevTools de Chrome, desde la línea de comandos o como un módulo de Node. Si le proporcionas una URL a Lighthouse para que la audite, ejecutará una serie de auditorías en la página y, luego, generará un informe sobre el rendimiento de la página. Usa las auditorías que fallaron como indicadores para mejorar la página. Cada auditoría tiene una referencia que explica por qué es importante, así como cómo corregirla.

También puedes usar Lighthouse CI para evitar regresiones en tus sitios.”

https://developer.chrome.com/docs/lighthouse

ANGULAR

- Instalación:
  - npm install -g lighthouse
- Ejecución desde línea de comandos:
  - lighthouse <url>
- Información sobre las métricas:
  - https://developer.chrome.com/docs/lighthouse/overview?hl=es-419

ANGULAR

- Enlaces de interés:
  - Apps web progresivas:
    - https://web.dev/explore/progressive-web-apps?hl=es-419
  - Angular Service Workers & PWAs
    - https://angular.dev/ecosystem/service-workers
  - Appshell:
    - https://angular.dev/ecosystem/service-workers/app-shell
  - Lighthouse:
    - https://developer.chrome.com/docs/lighthouse/overview?hl=es-419
    - https://chromewebstore.google.com/detail/lighthouse/blipmdconlkpinefehnmjammfjpmpbjk