

PWA

Origen de Pwa

PWA, abreviatura de Progressive Web App, es un concepto y enfoque de desarrollo web que combina lo mejor de las aplicaciones web y las aplicaciones móviles nativas. El concepto de PWA fue presentado por primera vez por Google en 2015, pero ha ganado popularidad en los últimos años.

Una PWA es una aplicación web que utiliza tecnologías web modernas para proporcionar una experiencia de usuario similar a la de una aplicación móvil nativa. Estas aplicaciones se pueden acceder a través de un navegador web, pero también se pueden instalar en el dispositivo del usuario y funcionar en modo desconectado.

Las PWA se caracterizan por las siguientes características:

1. **Responsive:** Las PWA están diseñadas para adaptarse y funcionar correctamente en diferentes dispositivos y tamaños de pantalla, ya sea en un teléfono móvil, una tableta o una computadora de escritorio.
2. **Conectividad y modo desconectado:** Las PWA pueden funcionar en modo desconectado o en condiciones de conectividad limitada. Esto se logra mediante el almacenamiento en caché de recursos y datos necesarios para el funcionamiento de la aplicación.
3. **Interfaz de usuario nativa:** Las PWA utilizan los estándares web modernos para crear interfaces de usuario atractivas y que se sientan como aplicaciones nativas. Esto incluye animaciones fluidas, navegación intuitiva y una experiencia de usuario similar a la de una aplicación móvil.
4. **Notificaciones push:** Las PWA pueden enviar notificaciones push al usuario, similar a las aplicaciones móviles, lo que permite una mayor interacción y compromiso con la aplicación.
5. **Acceso al hardware del dispositivo:** Las PWA tienen acceso limitado al hardware del dispositivo, lo que les permite utilizar características como la cámara, el micrófono, la geolocalización, etc., lo que las hace más poderosas y versátiles.
6. **Actualización automática:** Las PWA se actualizan automáticamente en segundo plano, lo que asegura que los usuarios siempre tengan la última versión de la aplicación sin necesidad de descargar o instalar actualizaciones manualmente.

El objetivo principal de las PWA es brindar a los usuarios una experiencia de aplicación móvil sin tener que desarrollar una aplicación nativa específica para cada plataforma. Al aprovechar las tecnologías web, las PWA son más rápidas de desarrollar, mantener y distribuir en comparación con las aplicaciones móviles tradicionales.

Además, las PWA ofrecen ventajas como una mayor visibilidad en los motores de búsqueda, una mayor tasa de retención de usuarios, una menor cantidad de espacio de almacenamiento requerido en los dispositivos y la capacidad de llegar a un público más amplio, ya que se pueden acceder a través de un navegador web sin la necesidad de descargar e instalar desde una tienda de aplicaciones.

En resumen, una PWA es una aplicación web progresiva que utiliza tecnologías web modernas para ofrecer una experiencia similar a la de una aplicación móvil nativa. Proporcionan una interfaz de usuario atractiva, funcionan en modo desconectado y tienen acceso a características nativas del dispositivo. Las PWA ofrecen una alternativa eficiente y efectiva para el desarrollo de aplicaciones móviles multiplataforma.

Ventajas de Pwa

1. **Accesibilidad y disponibilidad:** Las PWA se pueden acceder a través de un navegador web, lo que significa que los usuarios no tienen que descargar e instalar una aplicación desde una tienda de aplicaciones. Esto facilita el acceso a la aplicación y permite llegar a un público más amplio.
2. **Experiencia de usuario similar a la de una aplicación nativa:** Las PWA ofrecen una experiencia de usuario similar a la de una aplicación móvil nativa. Proporcionan una interfaz de usuario atractiva, interacciones suaves, navegación intuitiva y la capacidad de trabajar en modo desconectado.
3. **Actualización automática:** Las PWA se actualizan automáticamente en segundo plano, lo que asegura que los usuarios siempre tengan la última versión de la aplicación sin tener que descargar o instalar actualizaciones manualmente.
4. **Menor desarrollo y mantenimiento:** Las PWA se desarrollan utilizando tecnologías web estándar, lo que significa que se pueden utilizar las mismas habilidades y herramientas para el desarrollo web. Esto reduce el costo y el tiempo de desarrollo en comparación con el desarrollo de aplicaciones móviles nativas para diferentes plataformas.
5. **Mejora en el rendimiento:** Las PWA pueden aprovechar técnicas de almacenamiento en caché y carga rápida para mejorar el rendimiento de la aplicación. Esto significa que las páginas y recursos se pueden cargar más rápidamente, lo que brinda una mejor experiencia al usuario.

Comandos e Instalación

Antes de nada, hay 3 comandos. Este primer comando sirve para servir a cualquier aplicación web estática (Ya sea angular, react etc)

```
npm install http-server -g
```

```
http-server dist
```

Y este comando es el más común de vue:

```
npm run serve
```

Vue3

En vue se hace lo siguiente:

1. Si se crea un proyecto nuevo, configurarlo manualmente para que tenga el PWA.
2. Si ya esta creado ewl proyecto, se ejecuta el comando

```
vue add pwa
```

3. Se añade el siguiente codigo al registreServiceWorker.js en caso de que se quiera generar automaticamente, en caso de que necesita personalizarlo, se crea el manifest.json

```
module.exports = {
  pwa: {
    name: 'Mi PWA', (Nombre de la App)
    themeColor: '#4DBA87', (Color de fondo de la barra de navegación )
    msTileColor: '#000000', (Color del texto del titulo )
    appleMobileWebAppCapable: 'yes', (Se puede ejecutar como una
    aplicación independiente en dispositivos móviles con sistema operativo
    iOS)
    appleMobileWebAppStatusBarStyle: 'black',
    (Estilo de la barra de estado en dispositivos móviles con sistema
    operativo iOS )
    // configuración específica del manifest
    manifestOptions: {
      start_url: '/',
      display: 'standalone',
      icons: [
        {
          src: './img/icons/android-chrome-192x192.png',
          sizes: '192x192',
          type: 'image/png',
        },
        {
          src: './img/icons/android-chrome-512x512.png',
          sizes: '512x512',
          type: 'image/png',
        },
      ],
    },
  },
};
...

```

4. Si se quiere personaliza, se crea el manifest.json y se añade de la misma manera

```
{
  "name": "Planner",
  "short_name": "Planner",
  "theme_color": "#000000",
  "appleMobileWebAppCapable": "yes",
  "appleMobileWebAppStatusBarStyle": "black",
  "icons": [
    {
      "src": "./img/icons/android-chrome-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "./img/icons/android-chrome-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    },
    {
      "src": "./img/icons/android-chrome-maskable-192x192.png",
      "sizes": "192x192",
      "type": "image/png",
      "purpose": "maskable"
    },
    {
      "src": "./img/icons/android-chrome-maskable-512x512.png",
      "sizes": "512x512",
      "type": "image/png",
      "purpose": "maskable"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "background_color": "#000000"
}
```

5. Si se hace de la primera manera, se añade esto en el html (o .vue)

```
<link rel="manifest" href="<%= BASE_URL %>manifest.json">
```

Si se hace con el manifest.json, no hace falta.

En un proyecto basico

1. Se crea el manifest.json y el servide-worker.js

2. En el manifest.json, se debe añadir lo siguiente (el icono es obligatorio porque si no no funciona):

```
{
  "name": "Mi PWA",
  "short_name": "Mi PWA",
  "start_url": ".",
  "display": "standalone",
  "icons": [
    {
      "src": "android-chrome-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    }
  ]
}
```

3. En el serve-worker.js se añade lo siguiente

```
self.addEventListener("fetch", function (event) {
  event.respondWith(
    caches.match(event.request).then(function (response) {
      return response || fetch(event.request);
    })
  );
});
```

4. En el index.html se añade el link de la siguiente manera

```
...
<link rel="manifest" href="manifest.json" />
...
```

Angular

1. Una vez creado el proyecto, se instala la dependencia (el nombre del proyecto esta en angular.json/defaultProject)

```
ng add @angular/pwa
```

2. Al igual que en vue, se modifica el manifest.webmanifest dependiendo de lo que uno quiera

3. Se debe compilar así en caso de que se tenga ya un servidor https o en localhost.

```
ng build --prod
```