educación IT
CODE YOUR FUTURE

ReactJS Developer

Módulo 9

educación IT
CODE YOUR FUTURE

Progressive Web App

Progressive Web App

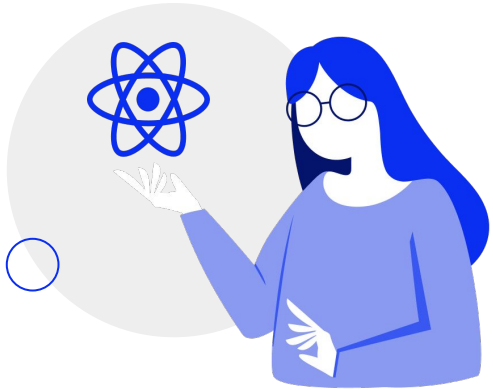
Una **Aplicación Web Progresiva** (*PWA* por sus siglas en inglés) es **un sitio web estático** (HTML/CSS/JS) que adopta, progresivamente, **ciertas características de la experiencia de usuario de una *app mobile***.

Las características más importantes son:

- **Se sirve bajo HTTPS**.
- Funciona ***offline*** (pre cacheo a través del *ServiceWorker*).
- Es **instalable**, permite acceder a ella directamente desde un ícono en el celular y tiene una interfaz en la que **no se muestran los controles del navegador**, simulando la experiencia de usuario de una *app mobile* real.

Pese a que desarrollar una PWA sea una cuestión puramente de **experiencia de usuario** y aunque no cuente c**on todas las características de una app nativa** (como acceso al sistema de archivos), debes saber que es una alternativa muy utilizada.

Muchos proyectos llegan a preferir el desarrollo de una PWA por encima de una app móvil nativa.



Veamos algunas de sus ventajas:

- **Es un desarrollo barato:** en comparación con el desarrollo de una app móvil nativa. Para desarrollar éstas últimas, necesitamos gente especializada en Java / Swing. Sin embargo, una PWA está desarrollada puramente en JS.
- **Es un producto ligero:** no requiere grandes instalaciones ni mucho espacio en el disco. Si bien es más limitada, ofrece una funcionalidad perfectamente válida para muchos proyectos.
- **Se adapta perfectamente a la mayoría de los casos:** en muchos productos web, no necesitamos acceder a los contactos. Piensa en un *eCommerce*, un *eLearning*, etc. No necesitas demasiados recursos del dispositivo móvil ni precisas aprovecharte de iOS o Android, sino que tienes todo lo necesario en tu web y en el navegador.

Algunas desventajas:

- **Son más limitadas**: no pueden acceder **al sistema de archivos, la lista de contactos ni guardar datos en una base de datos local como SQLite.**
- **Tienen un despliegue más complejo**: una app móvil nativa depende de ella misma y ya. Puede ser que tenga acceso a la red, pero no es un requisito imprescindible. Ahora bien, las PWA **necesitan un back-end publicado en otro servidor para obtener datos.**
- **No están disponibles en Play Store o App Store**: No pueden ser comercializadas individualmente como una app nativa. Son una vista diferente de una página web.

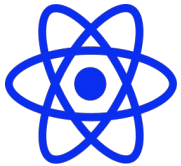
Crear un nuevo proyecto React + TS + PWA

Para crear una nueva PWA lo mejor es crear un proyecto **con React desde cero**. Si ya tienes un proyecto React y quieres convertirlo en PWA, puedes seguir estos mismos pasos y copiar los archivos propios de tu proyecto (tus componentes y los estáticos) a este nuevo creado.

Si bien *TypeScript* no es necesario, esta fórmula de React + TypeScript + PWA sería la utilizada para grandes proyectos.

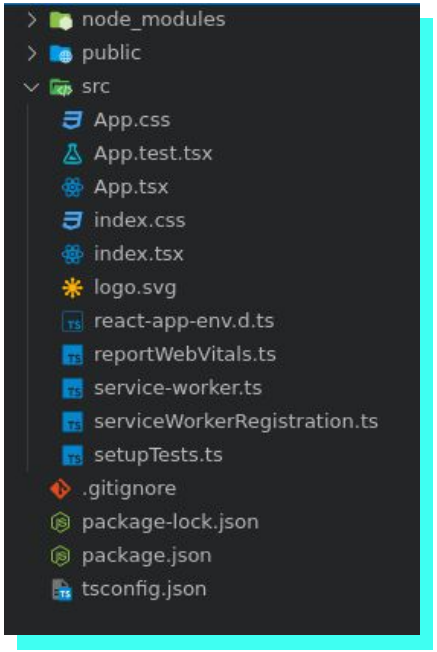
El comando es el siguiente:

```
npx create-react-app <NOMBRE DE TU APP>
--template cra-template-pwa-typescript
```



Estructura de archivos y carpetas

- Los cambios más importantes ocurrieron en la carpeta **/src**.
- El archivo **.d.ts** contiene **las declaraciones de tipos que serán exportadas luego**.
- El archivo **service-worker.ts** contiene **la lógica del *ServiceWorker* utilizando las librerías** workbox.
- El archivo **serviceWorkerRegistration.ts** es **el adaptador del *ServiceWorker* a la aplicación**.



Deploy a producción

1. Durante la etapa de desarrollo, tenemos el *service worker* deshabilitado, tal como se observa en el índex. Esto es así para que **no se almacenen en caché** (se *precacheen*) **los resultados de nuestras pruebas en nuestro navegador**.
2. Al momento de construir nuestro proyecto, debemos cambiar la llamada a ***unregister*** por la llamada a ***register***, tal como se indica en el archivo index.ts. Esto hará que nuestro *service worker* **comience a actuar.**
3. Luego, ejecutamos normalmente el comando **`npm run build`**.

educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!