**Universidad Tecnológica Santa Catarina**



**Título: Actividad Investigación**

**Nombres:** Enrique Alejandro Galván Montes

María Esmeralda García Hernandez

Ricardo Covarrubias Herrera

Joan Emiliano García Arcos

**Grupo:** IDGS09A

Table of Contents

[App Nativas. 2](#_Toc198839262)

[Características Clave de las Aplicaciones Híbridas 3](#_Toc198839263)

[PWA (Progressive Web App). 3](#_Toc198839264)

[Características de las PWA. 4](#_Toc198839265)

[Herramientas de ejecución y desarrollo para las PWA 5](#_Toc198839266)

[Frameworks para el desarrollo de PWAs 5](#_Toc198839267)

[1. Angular 5](#_Toc198839268)

[2. Vue.js 6](#_Toc198839269)

[3. Ionic 6](#_Toc198839270)

[4. React 7](#_Toc198839271)

[5. Svelte 7](#_Toc198839272)

[6. PWA Builder 8](#_Toc198839273)

[Herramientas y tecnologías de ejecución en PWAs: 9](#_Toc198839274)

[Navegadores compatibles con las pwa, poner el nombre del navegador y la versión 10](#_Toc198839275)

[Navegadores Compatibles con PWA 10](#_Toc198839276)

[1. Google Chrome 10](#_Toc198839277)

[2. Microsoft Edge (Chromium) 11](#_Toc198839278)

[3. Mozilla Firefox 11](#_Toc198839279)

[4. Safari 11](#_Toc198839280)

[5. Opera 11](#_Toc198839281)

[6. Samsung Internet 11](#_Toc198839282)

[Consideraciones Adicionales 11](#_Toc198839283)

[Ventajas y Desventajas de las Aplicaciones Web Progresivas (PWA) y Aplicaciones Móviles Nativas 12](#_Toc198839284)

[1. Aplicaciones Web Progresivas (PWA) 12](#_Toc198839285)

[Ventajas de las PWA 12](#_Toc198839286)

[Desventajas de las PWA 12](#_Toc198839287)

[2. Aplicaciones Móviles Nativas 13](#_Toc198839288)

[Ventajas de las Aplicaciones Nativas 13](#_Toc198839289)

[Desventajas de las Aplicaciones Nativas 13](#_Toc198839290)

[Comparación General: PWA vs Nativas 13](#_Toc198839291)

# App Nativas.

Las aplicaciones nativas son programas diseñados y desarrollados específicamente para un sistema operativo particular, como iOS o Android. A diferencia de las aplicaciones web o las híbridas, las nativas aprovechan al máximo las capacidades y características únicas de cada plataforma. Por eso, el desarrollo de aplicaciones multiplataforma es cada vez más demandado.

**Características de las apps nativas**

Los beneficios de las apps nativas son uno de sus puntos fuertes:

* **Rendimiento optimizado:** la velocidad y la capacidad de respuesta son características distintivas, ya que las apps nativas están diseñadas para aprovechar al máximo los recursos del dispositivo.
* **Acceso total a las funcionalidades del dispositivo:** desde la cámara y el micrófono hasta las notificaciones push, las apps nativas pueden aprovechar todas las funciones del dispositivo, ofreciendo una experiencia integrada.
* **Experiencia de usuario consistente:** el diseño y la interfaz se adaptan perfectamente al entorno del sistema operativo, proporcionando una experiencia de usuario coherente y familiar.
* **Optimización para tiendas de aplicaciones:** las apps nativas están diseñadas teniendo en cuenta los requisitos y pautas de las tiendas de aplicaciones, lo que facilita su descubrimiento y distribución.

**App Hibrida.**

Una aplicación móvil híbrida combina los elementos de una aplicación nativa (una aplicación desarrollada para una plataforma específica como Android o iOS) y una aplicación web (una aplicación a la que se puede acceder en internet a través de un navegador).

Está construida con lenguajes y tecnologías de desarrollo front-end populares como HTML5, JavaScript y CSS, lo que proporciona funcionalidad multiplataforma.

En otras palabras, los desarrolladores no tienen que crear código separado para Android e iOS. Pueden escribir código para una aplicación móvil una vez y al mismo tiempo adaptarse a múltiples plataformas.

# Características Clave de las Aplicaciones Híbridas

* **Compatibilidad Multiplataforma:** Las aplicaciones híbridas funcionan en múltiples plataformas, eliminando la necesidad de desarrollar y mantener aplicaciones separadas para iOS y Android.
* **Desarrollo Rápido:** Utilizando un solo código base, los desarrolladores pueden crear aplicaciones híbridas más rápidamente en comparación con las aplicaciones nativas, lo que permite lanzamientos más rápidos al mercado.
* **Acceso a Características del Dispositivo:** Las aplicaciones híbridas pueden acceder a características del dispositivo, como la cámara, el GPS y las notificaciones, utilizando plugins, brindando una experiencia de usuario completa.
* **Experiencia de Usuario Consistente:** Al utilizar una interfaz de usuario común, las aplicaciones híbridas ofrecen una experiencia coherente en todas las plataformas, lo que ayuda a construir una marca sólida y atractiva.
* **Mantenimiento Sencillo:** Las actualizaciones se pueden realizar de forma centralizada, lo que simplifica el proceso de mantenimiento y garantiza que todos los usuarios estén utilizando la última versión de la aplicación.
* **Costos Reducidos:** Al evitar la necesidad de desarrollar aplicaciones separadas para cada plataforma, las aplicaciones híbridas pueden ahorrar significativamente en costos de desarrollo y recursos humanos.
* **Escalabilidad y Flexibilidad:** Las aplicaciones híbridas son altamente escalables y pueden adaptarse fácilmente a las crecientes demandas del usuario y a los cambios en el mercado, brindando flexibilidad a los emprendedores para evolucionar con sus negocios.

# PWA (Progressive Web App).

Las PWA se definen comúnmente como las Apps que reúnen lo mejor de las aplicaciones web y de las nativas, incluso llegando a ser entendidas como un punto medio o una forma evolucionada.

# Características de las PWA.

**Responsive:** Estas Apps deben adaptarse automáticamente a cualquier formato, navegador o dispositivo (con los consecuentes cambios de medidas y resolución), y más considerando la naturaleza mobile de las mismas.

**Actualizada:** Siempre mostrarán su última versión al usuario, con el empleo de actualizaciones automáticas, de manera constante e instantánea y sin necesidad de descargarlas. Esto es posible gracias al empleo de Service Workers y porque no deja de ser una web App, independiente de la publicación (y todo el proceso de revisión e instalación por parte del usuario que conlleva) en los markets de aplicaciones.

**Segura:** Se usa siempre el protocolo seguro HTTPS que, además, es necesario para la instalación del Service Worker. Esto posibilita asegurar que el acceso sea seguro y que el contenido servido no haya sido sujeto a manipulaciones. Se emplean tecnologías como TLS para el cifrado web.

**Rápida:** Una PWA tiene la velocidad, tanto de carga como de navegación, optimizada. Esto permite que los contenidos se muestren al usuario prácticamente al instante, ya que se apoyan en el almacenamiento en la caché.

**Offline:** Una PWA debe permitir el acceso, ya sea de manera parcial o incluso total, a pesar de que no haya conexión a Internet (o esta sea de baja calidad).

**Multiplataforma:** La tecnología utilizada contempla su ejecución en diversos dispositivos, sistemas operativos y navegadores. Esto, además de ser clave a la hora de ofrecer una experiencia de usuario satisfactoria y alcanzar a más público potencial, supone facilidades para los desarrolladores y permite abaratar costes, puesto que no se requieren programaciones diferenciadas (algo que sí ocurre con las Apps nativas).

**Indexable y enlazable**: El contenido de una PWA es rastreable e indexable, de forma que pueda aparecer como resultado en un buscador. Además, esta se puede compartir mediante una URL, con la posibilidad de que la otra persona la utilice sin necesidad de instalarlo.

**Con acceso directo:** Las webs a las que se acceda desde el navegador que dispongan de una versión PWA suelen informar al usuario, invitándole a “añadirla a su pantalla de inicio”. Estas aplicaciones se pueden utilizar desde el navegador, pero también se pueden instalar en el dispositivo. Esta instalación no requiere de una “descarga” tal y como la conocemos, sino que se basa en la inclusión de un acceso directo en la pantalla de inicio o escritorio de nuestro dispositivo. Este se muestra como un icono más, prácticamente idéntico al de cualquier App nativa.

**Apariencia nativa:** La interfaz de usuario y, en general, la apariencia de una PWA es muy similar a la de las Apps nativas, tanto en estética como en la manera de interactuar y navegar por ella.

# Herramientas de ejecución y desarrollo para las PWA

Para empezar, hablaremos de las diferencias entre una herramienta de ejecución y de desarrollo:

|  |  |  |
| --- | --- | --- |
| **Concepto** | **Desarrollo** | **Ejecución** |
| **¿Qué es?** | Es el proceso de crear la PWA | Es el momento en que la PWA funciona en el navegador o dispositivo |
| **Objetivo** | Escribir el código, probar, depurar, diseñar | Usar la aplicación ya lista, como usuario final |
| **Herramientas típicas** | Frameworks, editores de código, CLI, testing, depuradores | Navegadores, dispositivos móviles, Service Workers |
| **Ejemplo** | Programar la lógica del carritoo en un ecommerce | Que un cliente use la PWA offline desde su celular |
| **Quien lo usa** | Desarrolladores o programadores | Usuarios finales (clientes, visitantes) |

# Frameworks para el desarrollo de PWAs

El artículo menciona varios frameworks y herramientas clave que los desarrolladores pueden utilizar para crear PWAs eficientes. Estos frameworks ofrecen soporte para la creación de aplicaciones rápidas, receptivas y confiables. A continuación, te doy una visión más detallada de los frameworks que se destacan:

## 1. Angular

* **Descripción**: Angular es un framework de JavaScript de código abierto que fue desarrollado por Google. Es ampliamente utilizado para crear aplicaciones web de una sola página (SPA) que son rápidas, dinámicas y fáciles de mantener.
* **Ventajas**:
  + **Desarrollo modular**: Angular permite trabajar con componentes modulares, lo que facilita la gestión y escalabilidad de aplicaciones grandes.
  + **Soporte para PWAs**: Angular ofrece herramientas como Service Workers y el módulo de PWA para integrar las características clave de una PWA, como el funcionamiento sin conexión.
  + **Comunicación bidireccional**: Con Angular, puedes manejar datos en tiempo real y con facilidad, lo que lo hace adecuado para aplicaciones de gran escala.
* **Ideal para**: Proyectos de gran envergadura que requieran una estructura robusta y soporte a largo plazo.

## 2. Vue.js

* **Descripción**: Vue.js es un framework de JavaScript progresivo que es especialmente conocido por su simplicidad, rendimiento rápido y facilidad de integración con otras bibliotecas.
* **Ventajas**:
  + **Curva de aprendizaje suave**: Vue es fácil de aprender y tiene una sintaxis sencilla, lo que lo hace perfecto para desarrolladores novatos.
  + **Componente reactivo**: Permite que la interfaz de usuario se actualice automáticamente cuando los datos cambian, lo que mejora la experiencia del usuario.
  + **Desarrollo flexible**: Vue es flexible y se puede utilizar tanto en proyectos pequeños como grandes. Puedes integrarlo progresivamente en una aplicación existente, lo que es perfecto para proyectos en evolución.
  + **PWA-friendly**: Vue también tiene herramientas y plugins como vue-pwa que facilitan la integración de características PWA como el uso de cachés y notificaciones push.
* **Ideal para**: Proyectos que necesitan una base simple pero potente y fácil de aprender.

## 3. Ionic

* **Descripción**: Ionic es un framework basado en tecnologías web como HTML, CSS y JavaScript. Se especializa en la creación de aplicaciones híbridas, pero también permite desarrollar PWAs.
* **Ventajas**:
  + **Enfoque nativo**: Ionic utiliza componentes de interfaz de usuario nativos para crear aplicaciones con un diseño que se asemeja a las aplicaciones móviles tradicionales.
  + **Compatibilidad multiplataforma**: Puedes crear aplicaciones para Android, iOS y la web con una sola base de código, lo que ahorra tiempo y recursos.
  + **Rendimiento**: A pesar de ser un framework híbrido, Ionic ofrece un rendimiento bastante bueno, especialmente para aplicaciones más sencillas.
  + **PWA-friendly**: Ionic tiene soporte incorporado para características clave de las PWAs, como la capacidad de trabajar sin conexión y la optimización para móviles.
* **Ideal para**: Desarrolladores que quieren crear aplicaciones con una experiencia nativa utilizando tecnologías web.

## 4. React

* **Descripción**: React es una biblioteca de JavaScript desarrollada por Facebook que se utiliza para crear interfaces de usuario interactivas. Es muy popular debido a su eficiencia y flexibilidad.
* **Ventajas**:
  + **Virtual DOM**: React usa un Virtual DOM para mejorar el rendimiento, lo que permite que las aplicaciones carguen rápidamente.
  + **Componentes reutilizables**: Permite crear componentes autónomos que pueden ser reutilizados en diferentes partes de la aplicación, lo que hace que el desarrollo sea más eficiente.
  + **PWA-friendly**: React puede integrarse con librerías como Workbox para permitir que las aplicaciones sean PWAs con capacidades de almacenamiento en caché y notificaciones push.
* **Ideal para**: Desarrolladores que buscan una biblioteca poderosa para crear interfaces de usuario reactivas y modernas.

## 5. Svelte

* **Descripción**: Svelte es un framework relativamente nuevo que ha ganado popularidad debido a su enfoque único. En lugar de ejecutar el código en el navegador, Svelte compila los componentes en código eficiente que se ejecuta de manera nativa en el navegador.
* **Ventajas**:
  + **Rendimiento excepcional**: Svelte elimina la necesidad de un framework pesado en el navegador, lo que se traduce en aplicaciones más rápidas.
  + **Sin virtual DOM**: A diferencia de otros frameworks, Svelte no utiliza un Virtual DOM, lo que mejora el rendimiento en tiempo de ejecución.
  + **Desarrollo rápido**: La sintaxis simple y la naturaleza reactiva de Svelte permiten a los desarrolladores crear aplicaciones rápidamente.
  + **PWA-friendly**: Svelte también puede integrarse con bibliotecas de PWA y ofrece un rendimiento increíble incluso en dispositivos con recursos limitados.
* **Ideal para**: Proyectos que necesitan un alto rendimiento y una experiencia de usuario fluida.

## 6. PWA Builder

* **Descripción**: PWA Builder es una herramienta que permite a los desarrolladores crear una PWA con facilidad. Esta herramienta genera el código necesario para crear una PWA desde una aplicación web existente.
* **Ventajas**:
  + **Fácil de usar**: La interfaz simple permite que incluso los desarrolladores con poca experiencia en PWAs puedan crear aplicaciones progresivas.
  + **Optimización para móviles**: Genera automáticamente una versión optimizada para dispositivos móviles, lo que es crucial para la adopción de PWAs.
  + **Generación automática de manifest y Service Workers**: Facilita la configuración de archivos críticos para una PWA.
* **Ideal para**: Desarrolladores que buscan una forma rápida y fácil de convertir una aplicación web en una PWA.

# Herramientas y tecnologías de ejecución en PWAs:

1. **Service Workers**
   * Son scripts que el navegador ejecuta en segundo plano, separados de la página web.
   * Permiten manejar la caché, controlar las solicitudes de red, y habilitar el funcionamiento sin conexión.
   * Son clave para que la PWA funcione rápido y pueda usarse offline.
2. **Manifest.json**
   * Es un archivo JSON que describe la aplicación: nombre, iconos, colores, modo pantalla, etc.
   * El navegador lo usa para saber cómo mostrar la app cuando se “instala” en el dispositivo (como un icono en la pantalla de inicio).
   * Facilita que la PWA se comporte como una app nativa.
3. **API de Notificaciones Push**
   * Permite que la PWA envíe notificaciones al usuario incluso cuando la app no está abierta.
   * Es parte del lado de ejecución y mejora la interacción con el usuario.
4. **API de Caché (Cache Storage API)**
   * Permite almacenar recursos estáticos y dinámicos localmente para que la PWA cargue rápido y funcione sin conexión.
   * Se usa junto con los Service Workers.
5. **API de Acceso a Hardware**
   * Incluye acceso a cámara, micrófono, geolocalización, acelerómetro, etc.
   * Permite que las PWAs utilicen funcionalidades del dispositivo en tiempo real.
6. **Modo pantalla completa / standalone**
   * Configurado a través del archivo manifest, hace que la PWA se ejecute sin la interfaz típica del navegador, pareciéndose más a una app nativa.

# Navegadores compatibles con las pwa, poner el nombre del navegador y la versión

Las Aplicaciones Web Progresivas (PWA, por sus siglas en inglés) han revolucionado la forma en la que interactuamos con la web, combinando lo mejor de las aplicaciones nativas y las páginas web tradicionales. Su objetivo principal es proporcionar experiencias de usuario rápidas, fiables y atractivas, independientemente del navegador o dispositivo que se utilice. Esta tecnología depende en gran medida de la compatibilidad que los navegadores ofrezcan para funcionar adecuadamente.

## Navegadores Compatibles con PWA

|  |  |  |  |
| --- | --- | --- | --- |
| **Navegador** | **Versión mínima compatible** | **Sistemas operativos compatibles** | **Observaciones** |
| Google Chrome | 73 | Windows, macOS, Linux, Android, iOS | Soporte completo. Permite instalación y funcionalidades nativas. |
| Microsoft Edge | 79 (Chromium) | Windows, macOS, Android, iOS | Versión moderna basada en Chromium. Soporte completo. |
| Mozilla Firefox | 44 (Android) | Android, Windows, macOS, Linux | Soporte completo solo en Android. En escritorio, soporte muy limitado. |
| Safari | 11.1 | iOS, macOS | Soporte parcial. Sin notificaciones push, restricciones en almacenamiento local. |
| Opera | 46 | Windows, macOS, Linux, Android | Basado en Chromium. Soporte completo. |
| Samsung Internet | 5.0 | Android | Basado en Chromium. Compatible con PWA y optimizado para dispositivos Samsung. |

## 1. Google Chrome

Desde la versión 73, Google Chrome permite la instalación directa de PWA mediante el botón “Instalar” que aparece al detectar manifest.json y service workers. Compatible con Web App Manifest, Service Workers, almacenamiento local y push notifications.

## 2. Microsoft Edge (Chromium)

Desde la versión 79, Edge adoptó la base de Chromium, heredando todas las capacidades de soporte para PWA de Chrome. Además, permite instalación desde la Microsoft Store en algunos escenarios.

## 3. Mozilla Firefox

Firefox permite instalar PWA en Android desde la versión 44. En escritorio, el soporte fue experimental y finalmente retirado, aunque existen navegadores basados en Firefox como Floorp que han revivido esa funcionalidad.

## 4. Safari

Disponible a partir de iOS 11.3 y macOS 11.1. Aunque permite agregar PWA a la pantalla de inicio, no soporta completamente notificaciones push ni funcionalidades offline completas.

## 5. Opera

Opera, al estar basado en Chromium, ofrece el mismo nivel de compatibilidad que Chrome desde su versión 46. Compatible con todas las características esenciales de una PWA.

## 6. Samsung Internet

Este navegador está basado en Chromium y está optimizado para dispositivos Samsung. Desde la versión 5.0 soporta manifest.json, service workers y permite la instalación de PWA directamente desde el navegador.

# Consideraciones Adicionales

Instalabilidad: Los navegadores modernos permiten la instalación de PWA desde la interfaz del navegador si detectan correctamente el manifiesto y los service workers.

Almacenamiento: Algunas funcionalidades como almacenamiento offline, caché y acceso a sensores pueden variar dependiendo del navegador.

Seguridad: Las PWA requieren servir contenido vía HTTPS, lo cual es compatible en todos los navegadores modernos.

Limitaciones en iOS: Safari aún restringe funcionalidades como notificaciones push, almacenamiento persistente y acceso a fondo.

# **Ventajas y Desventajas de las Aplicaciones Web Progresivas (PWA) y Aplicaciones Móviles Nativas**

En el contexto del desarrollo móvil moderno, existen principalmente dos enfoques para llevar contenido y funcionalidades a los usuarios: las Aplicaciones Web Progresivas (PWA) y las Aplicaciones Móviles Nativas. Cada una presenta fortalezas y debilidades que deben evaluarse dependiendo del tipo de proyecto, los recursos disponibles y los objetivos a alcanzar. Este documento analiza de forma comparativa las ventajas y desventajas de ambas tecnologías.

# 1. Aplicaciones Web Progresivas (PWA)

Una PWA es una aplicación desarrollada con tecnologías web tradicionales como HTML, CSS y JavaScript, pero que incorpora funcionalidades que normalmente se encuentran en apps nativas, como trabajo offline, acceso al hardware y notificación push.

## Ventajas de las PWA

* Multiplataforma: Una sola base de código funciona en navegadores de escritorio y móviles.
* Sin necesidad de tiendas: No requiere publicación en App Store o Play Store; se instala directamente desde el navegador.
* Actualizaciones automáticas: Los usuarios siempre acceden a la última versión.
* Menor consumo de datos: Las PWA son más ligeras en tamaño y utilizan menos recursos.
* Desarrollo y mantenimiento más económico: No se necesita crear versiones separadas para cada sistema operativo.

## Desventajas de las PWA

* Acceso limitado al hardware: No todas las APIs del sistema están disponibles, especialmente en iOS.
* Restricciones en iOS: Limitaciones importantes como ausencia de notificaciones push o acceso a background sync.
* Sin presencia en tiendas: Puede afectar la visibilidad, ya que muchos usuarios buscan apps en tiendas oficiales.
* Menor rendimiento: Aunque ha mejorado mucho, el rendimiento sigue siendo inferior al de las apps nativas para tareas exigentes.

# 2. Aplicaciones Móviles Nativas

Las apps nativas se desarrollan usando lenguajes y herramientas específicas para cada sistema operativo, como Swift para iOS o Kotlin/Java para Android. Esto permite un control total sobre la interfaz y el rendimiento.

## Ventajas de las Aplicaciones Nativas

* Rendimiento superior: Aprovechan al máximo los recursos del dispositivo.
* Mejor integración con hardware: Acceso completo a GPS, cámara, sensores, bluetooth, etc.
* Notificaciones push sin restricciones: Total compatibilidad con los servicios de notificaciones del sistema.
* Experiencia de usuario personalizada: Diseños adaptados a cada sistema operativo.
* Presencia en tiendas: Mayor visibilidad y confianza para el usuario final.

## Desventajas de las Aplicaciones Nativas

* Costo elevado: Es necesario desarrollar una versión para cada plataforma.
* Tiempo de desarrollo mayor: Requiere más esfuerzo y equipos especializados.
* Actualizaciones dependientes de las tiendas: Cada actualización debe pasar por revisión.
* Mantenimiento complejo: Gestionar múltiples bases de código es más costoso y propenso a errores.

# Comparación General: PWA vs Nativas

|  |  |  |
| --- | --- | --- |
| **Característica** | **PWA** | **Aplicaciones Nativas** |
| Instalación | Desde navegador | Desde App Store / Play Store |
| Acceso a hardware | Limitado | Completo |
| Notificaciones push | Limitado en iOS | Completo |
| Experiencia de usuario | Generalizada | Específica y optimizada |
| Rendimiento | Medio | Alto |
| Costo de desarrollo | Bajo | Alto |
| Tiempo de desarrollo | Rápido | Lento |
| Mantenimiento | Centralizado | Individual por plataforma |

*Aplicaciones híbridas características | Tangram Consulting®*. (2023, 7 octubre). https://tangramconsulting.es/noticias/aplicaciones-hibridas-caracteristicas

*Apps web progresivas  |  web.dev*. (s. f.). web.dev. https://web.dev/learn/pwa/progressive-web-apps?hl=es-419

AppsFlyer. (2024, 18 febrero). *Aplicación híbrida*. AppsFlyer. https://www.appsflyer.com/es/glossary/hybrid-app/

Jardí, M. (2023, 27 septiembre). PWA vs. Native: ¿Aplicaciones web progresivas o nativas? *StrategicPlatform*. https://strategicplatform.com/articulos/aplicaciones-web-progresivas-o-aplicaciones-nativas

Mehta, A. (2024, 18 octubre). Top 10 Frameworks and Tools to Build Progressive Web Apps. *Appinventiv*. https://appinventiv.com/blog/top-pwa-development-frameworks/

Mora, S. L. (2022, 4 octubre). *Qué es PWA: características, tecnologías, ventajas y desventajas*. DIGITAL55. https://digital55.com/blog/que-es-pwa-ventajas-desventajas/

*Progressive web apps compatibility-firt.dev*. (2023, 9 agosto). https://firt.dev/notes/pwa/

Universidad Europea. (2024, 22 enero). *Qué son las apps nativas*. https://universidadeuropea.com/blog/apps-nativas/

Vidal, M. (2022, 2 noviembre). *¿Qué son las Progressive Web Apps? ¿Por qué son tan importantes?* Thinking For Innovation. https://www.iebschool.com/hub/progressive-web-apps-analitica-usabilidad/