



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE QUINTANA ROO

DIVISIÓN DE CIENCIAS, INGENIERÍA Y TECNOLOGÍA

**APLICACIONES WEB PROGRESIVAS (PWA)
COMO ALTERNATIVA A LAS APLICACIONES
MÓVILES NATIVAS**

TRABAJO MONOGRÁFICO
PARA OBTENER EL GRADO DE

INGENIERA EN REDES

PRESENTA
YURIDIA MIMI ARROYO BANDALA

SUPERVISORES
M.T.I. MELISSA BLANQUETO ESTRADA
M.T.I. VLADIMIR VENIAMIN CABAÑAS VICTORIA

DR. JAIME SILVERIO ORTEGÓN AGUILAR

DR. JAVIER VÁZQUEZ CASTILLO

M.S.I. LAURA YÉSICA DÁVALOS CASTILLO

CHETUMAL QUINTANA ROO, MÉXICO, JULIO DE 2023





UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE QUINTANA ROO

DIVISIÓN DE CIENCIAS, INGENIERÍA Y TECNOLOGÍA

TRABAJO MONOGRÁFICO TITULADO

"APLICACIONES WEB PROGRESIVAS (PWA) COMO ALTERNATIVA A LAS APLICACIONES
MÓVILES NATIVAS"

ELABORADO POR
YURIDIA MIMI ARROYO BANDALA

BAJO SUPERVISIÓN DEL COMITÉ DEL PROGRAMA DE LICENCIATURA Y
APROBADO COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE:

INGENIERA EN REDES

SUPERVISORA:

COMITÉ DE SUPERVISIÓN

M.T.I. MELISSA BLANQUETO ESTRADA

SUPERVISOR:

M.T.I. VLADIMIR VENIAMIN CABAÑAS VICTORIA

SUPERVISOR:

DR. JAIME SILVERIO ORTEGON AGUILAR

SUPERVISOR SUPLENTE:

DR. JAVIER ALFONSO CASTILLO

SUPERVISORA SUPLENTE:

MS.I. LAURA JESSICA DAVALOS CASTILLO



CHETUMAL, QUINTANA ROO, MÉXICO, JULIO DE 2023



RESUMEN

El contenido de esta investigación monográfica está dividido en 7 capítulos donde cada uno de ellos describe y adjunta elementos que nos ayudan a entender de forma clara, descriptiva y gráfica las Aplicaciones Web Progresivas (PWA) de manera que entre más capítulos se avancen más información tendremos sobre las PWA.

En el capítulo 1 se plantea el problema por el cual se optó por el tema y se describen los objetivos generales y específicos que caracterizan esta monografía, así como el alcance de ésta.

En el capítulo 2 podemos encontrar las distintas definiciones que se encuentran para las PWA, así como antecedentes históricos de como surgieron estas nuevas tecnologías usadas en las aplicaciones.

Seguidamente, el capítulo 3 expone las funciones y capacidades de las PWA y sus características en cuanto a los procesos que sigue para funcionar adecuadamente.

El capítulo 4 explica e ilustra los componentes tecnológicos principales de una PWA: el archivo manifiesto, los Service Worker, y el protocolo seguro para una PWA.

En el capítulo 5 se muestran gráficos comparativos entre una PWA y una APP nativa; así como, las ventajas y desventajas de ésta.

En el capítulo 6 se describen casos de éxito donde se han implementado aplicaciones web progresivas en diferentes empresas.

Por último, en el capítulo 7 conocerá la conclusión a la que llevó este estudio y a qué empresas está dirigida una PWA y los problemas que es capaz de resolver.

AGRADECIMIENTOS

En primera instancia, agradezco a Dios por permitirme llegar a este día tan esperado y por haberme acompañado en todo momento que sentí desfallecer, por estar conmigo en los buenos y malos momentos que enfrenté.

A mis padres Alberto Arroyo Roano y Minerva Bandala Olmedo que me apoyaron económicamente en todo momento aun en los momentos más complicados que enfrentamos y me motivaron a seguir adelante con mis estudios para ser una mujer independiente.

Quiero expresar mi agradecimiento a la MTI. Melissa Blanqueto Estrada mi tutora en este proyecto que confió en mí y me animó a elaborar esta monografía y sabiendo que a veces interfieren factores que nos debilitan y que nos consumen tiempo al realizar el trabajo, pero ella pacientemente me esperó y guío durante este duro trabajo corrigiendo mis errores y mostrándome el camino para superar este último obstáculo antes de graduarme.

También quiero agradecer al MTI. Vladimir Veniamin Cabañas Victoria por su apoyo durante toda mi estancia en la universidad pues siempre resolvió mis dudas cuando acudía en su ayuda, aunque no era mi tutor asignado siempre pude acudir a él.

Agradezco al Dr. Javier Vazquez Castillo que fue mi tutor durante todo mi periodo de estudios y quien me oriento cada semestre en cuanto a las materias de mi mapa curricular.

DEDICATORIA

Dedico esta monografía a mis amados padres Alberto Arroyo Roano y Minerva Bandala Olmedo ya que me trajeron a este mundo lleno de obstáculos y complicaciones, pero donde también hay victorias cuando uno se propone a conseguir las; a ellos que me motivaron cada día a superarme siendo su hija mayor.

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN.....	1
1.1 PLANTEAMIENTO DEL PROBLEMA.	2
1.2 JUSTIFICACIÓN.....	2
1.3 OBJETIVOS.....	3
1.3.1 <i>Objetivo General</i>	3
1.3.2 <i>Objetivos Específicos</i>	3
1.4 ALCANCES Y LIMITACIONES.....	3
CAPÍTULO 2. ANTECEDENTES DE LAS APLICACIONES WEB PROGRESIVAS	5
2.1. LAS APLICACIONES Y LOS DISPOSITIVOS MÓVILES	5
2.2 DEFINICIÓN DE APLICACIONES NATIVAS.....	5
2.3 DEFINICIÓN DE APLICACIONES WEB.....	8
2.4 DEFINICIÓN DE APLICACIONES WEB PROGRESIVAS.....	10
CAPÍTULO 3 FUNCIONAMIENTO DE UNA APLICACIÓN WEB PROGRESIVA (PWA)	13
3.1 ARCHIVO DE MANIFIESTO	13
3.1.1 <i>El formato JSON</i>	14
3.2 FUNCIÓN POR MEDIO DE LOS SERVICE WORKER.....	14
3.3 PROCESO DEL FUNCIONAMIENTO DE LA PWA.....	16
3.4 PROCESO PARA LA INICIALIZACIÓN DE UNA PWA.....	17
CAPÍTULO 4 COMPONENTES TECNOLÓGICOS DE UNA PWA.....	18
4.1 ARCHIVO MANIFIESTO.....	18
4.1.1 <i>Estructura De Los Elementos Para El App Manifiesto</i>	19
4.2 SERVICE WORKER.....	22
4.2.1 <i>Función de un Service Worker</i>	23
4.2.2 <i>Cómo agregar un Service Worker</i>	25
4.2.3 <i>Ciclo de vida de un Service Worker</i>	26
4.3 PROTOCOLO SEGURO PARA LAS APLICACIÓN WEB PROGRESIVA.....	28
4.3.1 <i>Proceso de HTTPS</i>	29
CAPÍTULO 5 APLICACIONES WEB PROGRESIVAS VS APLICACIONES MÓVILES NATIVAS	30

5.1 VENTAJAS Y DESVENTAJAS QUE NOS BRINDAN LAS PWA.....	30
5.2. BUSCADORES COMPATIBLES CON LAS PWA.....	32
5.3 COMPARACIÓN DE LAS VENTAJAS DE UNA PWA Y UNA APLICACIÓN MÓVIL NATIVA.....	32
CAPÍTULO 6 POSIBILIDADES QUE BRINDAN LAS APLICACIONES WEB PROGRESIVAS (PWA).	34
6.1 CASOS DE ÉXITO.....	34
CAPÍTULO 7 CONCLUSIONES	36
BIBLIOGRAFÍA	38

ÍNDICE DE FIGURAS

FIGURA 1. EJEMPLO DE APLICACIONES NATIVAS EN ANDROID E IOS.....	8
FIGURA 2 EJEMPLO DE APLICACIÓN WEB ABIERTA DESDE UN NAVEGADOR SAFARI.	9
FIGURA 3 EJEMPLO DE UNA APLICACIÓN WEB ABIERTA DESDE EL NAVEGADOR GOOGLE CHROME.	10
FIGURA 4 COMPARATIVA CON CARACTERÍSTICAS ENTRE UNA APP NATIVA Y UNA PWA.....	11
FIGURA 5 ARQUITECTURA DEL FUNCIONAMIENTO DE UN SERVICE WORKER	15
FIGURA 6. PROCESO DESCRIPTIVO DE UNA PWA.....	16
FIGURA 7 EJEMPLO DE LA ESTRUCTURA DE UN ARCHIVO MANIFIESTO	20
FIGURA 8 DESCRIPCIÓN BÁSICA DEL SERVICE WORKER	23
FIGURA 9. FUNCIONAMIENTO DE UN SERVICE WORKER (ELABORACIÓN PROPIA).	24
FIGURA 10. CICLO DE VIDA DE UN SERVICE WORKER	27
FIGURA 11. GRÁFICO DESCRIPTIVO DEL PROTOCOLO HTTPS.	29
FIGURA 12. BUSCADORES QUE SOPORTAN PWA	32

ÍNDICE DE TABLAS

TABLA-1. REQUERIMIENTOS DE APLICACIONES NATIVAS (GALDAMEZ, THOMAS, PESADO, & LISANDRO, 2013).....	7
TABLA 2. CARACTERÍSTICAS BÁSICAS DE UN MANIFIESTO JSON	13
TABLA 3 TABLA DESCRIPTIVA DE LOS ELEMENTOS DEL MANIFIESTO	20
TABLA 4 PRINCIPALES DIFERENCIAS ENTRE PROTOCOLÓ HTTP Y HTTPS	28
TABLA 5. VENTAJAS Y DESVENTAJAS QUE ENCONTRAMOS EN LAS PWA.....	30
TABLA 6. CARACTERÍSTICAS QUE BRINDA UNA PWA ANTE UNA APLICACIÓN NATIVA.....	33

CAPÍTULO 1. INTRODUCCIÓN

La presente investigación tiene como objeto de estudio las aplicaciones web progresivas (PWA) ante las aplicaciones móviles nativas. Y como su nombre lo indica es un tipo de aplicación que progresivamente mejora y se adapta a las necesidades y actualizaciones de la tecnología actual. El término PWA resulta de la abreviatura de las siglas en ingles *Progressive Web Apps*.

Una PWA adapta características de una aplicación nativa con las características de una aplicación web, es una convergencia de ambas brindando así características y ventajas únicas. Por ejemplo, una aplicación web progresiva no requiere instalación en el dispositivo y se comparten por un enlace o código QR.

El principal objetivo por el cual se investiga este tema es debido a que no muchas personas ni empresas están familiarizadas con las aplicaciones web progresivas, por lo que no saben de sus ventajas y alcances que nos brindan, por ejemplo, que son más económicas que una aplicación web nativa. Así pues, para poder dar a conocer este tipo de aplicaciones y mostrar cómo funcionan se abordaron temas como; procesos que siguen las PWA, cuáles son sus componentes tecnológicos, cuáles son sus características, sus ventajas, como nos beneficia y porque es una mejor alternativa ante otro las aplicaciones nativas. Por esta razón, se encuentra información donde se comparan las aplicaciones nativas convencionales con las aplicaciones web progresivas.

1.1 Planteamiento Del Problema.

En los últimos años, el avance tecnológico ha crecido de manera exponencial, la demanda por la nueva tecnología y el diseño de nuevas plataformas está provocando la adaptación de nuevos procesos tecnológicos en los sistemas computacionales como los dispositivos móviles. Según (Thomas, Lisandro, Corbalán, & Cáseres, 2018) “El desarrollo de aplicaciones para dispositivos móviles plantea nuevos desafíos originados en las características únicas de esta actividad”, que tienen un gran crecimiento en el ámbito tecnológico.

La posibilidad de elegir una aplicación PWA ante una aplicación móvil nativa es resultado de la adaptación de las necesidades de los usuarios. Como bien se sabe las aplicaciones móviles nativas están instaladas en nuestros propios dispositivos móviles y ocupan un porcentaje considerable entre las aplicaciones de internet en la actualidad, sin embargo, estas aplicaciones nativas se desarrollan de manera específica para su plataforma y sistema operativo; un ejemplo de ello serían las aplicaciones para Android o iOS que se crean con su propio lenguaje de programación; Android usa el lenguaje Java e iOS utiliza el lenguaje Swift por lo que no son compatibles entre sí las apps Android y Apple; en otras palabras las aplicaciones creadas para Android no se pueden ejecutar en dispositivos iOS. Las PWA son una nueva forma de aplicaciones con las cuales podemos superar las limitaciones en las apps nativas; por ejemplo: menor tamaño, mayor rapidez en las actualizaciones, mayor capacidad de acceso por ser una aplicación para plataforma web, entre otras más. Por lo que las PWA son aplicaciones que no se instalan en nuestros dispositivos, sino que se accede a ellas utilizando algún navegador web para ver el contenido.

1.2 Justificación

El mercado de aplicaciones móviles no es exclusivo de un solo fabricante. las tecnologías que están surgiendo en la actualidad han impulsado el aumento en el desarrollo de nuevas aplicaciones vinculadas con el acceso a internet, evidenciando la importancia de consolidarse como empresas en este mercado, para ellos es

importante estar actualizando constantemente sus diferentes sistemas, tanto en el ahorro de espacio, velocidad o tiempo y sobre todo la diversificación en los diferentes sistemas en aplicaciones.

Actualmente existen diferentes tipos de aplicaciones móviles, entre las que se destacan: las aplicaciones nativas, aplicaciones híbridas y las aplicaciones progresivas. Esta investigación sobre aplicaciones web progresivas PWA puede apoyarnos en distinguir con mayor claridad las potencialidades que nos brindan, abordar qué caracteriza a la tecnología asociada a las aplicaciones web progresivas, sus ventajas y sus desventajas respecto a las aplicaciones móviles nativas y cuáles son las tendencias próximas.

1.3 Objetivos.

1.3.1 Objetivo General.

Describir las aplicaciones web progresivas (PWA), sus componentes tecnológicos y contrastarlas con las aplicaciones móviles nativas.

1.3.2 Objetivos Específicos.

- Definir qué es una aplicación web progresiva.
- Explicar cómo funciona una PWA.
- Identificar cuáles son los componentes tecnológicos de una aplicación web progresiva.
- Comparar las aplicaciones web progresivas respecto a las aplicaciones móviles nativas y exponer cuáles son las ventajas y riesgos de usarlas.
- Distinguir las posibilidades de las aplicaciones web progresivas.

1.4 Alcances y Limitaciones.

Alcances

Estudio y análisis del tema aplicaciones web progresivas (PWA) ante las aplicaciones nativas. Se toma en consideración investigaciones hechas

anteriormente y se contrastan con información más actual para comparar con las aplicaciones nativas actuales y demostrar las ventajas y oportunidades que nos brindan.

Limitaciones

La principal limitante de esta investigación es que la información sobre PWAs no está tan difundida y requiere mayor énfasis para crear aplicaciones que lleguen a potenciales consumidores de estas aplicaciones.

CAPÍTULO 2. ANTECEDENTES DE LAS APLICACIONES WEB PROGRESIVAS

2.1. Las aplicaciones y los dispositivos móviles

Una aplicación nunca deja de ser un software; el cual se diseña para cumplir diversas funciones dentro de un sistema. Según Cuello & Vittone (2013)

“Podemos decir que las aplicaciones son para los móviles lo que los programas son para los ordenadores de escritorio [...] Hubo un cambio grande con el ingreso de iPhone al mercado, ya que con él se generaron nuevos modelos de negocio que hicieron de las aplicaciones algo rentable, tanto para desarrolladores como para los mercados de aplicaciones, como App Store, Google Play y Windows Phone Store”(p.14-15).

Antes de abordar las aplicaciones web progresivas se definirá qué es una aplicación nativa y qué es una aplicación web para tratar de explicar la nueva generación de aplicaciones web llamada PWA puesto que se combinan los beneficios de una aplicación nativa con las aplicaciones web.

2.2 Definición de aplicaciones nativas

Una aplicación nativa es aquella que ha sido desarrollada y optimizada para un sistema operativo específico como podría ser Android o iOS. Una vez desarrollado una aplicación debe ser colocada en una app store conocida como tienda de aplicaciones la cual debe ser específica para cada sistema operativo de los dispositivos.

Una aplicación nativa tiene la capacidad de acceder y adaptarse al 100% con las características y funcionalidades que nos brinda el teléfono inteligente como la cámara, el GPS (sistema de geoposicionamiento), grabaciones de sonido etc.

Es importante aclarar que el desarrollo de una aplicación nativa genera un mayor costo puesto que si se desea utilizar una aplicación para distintos sistemas

operativos se debe realizar una nueva versión para cada sistema operativo, incrementando el costo del desarrollo.

Según (Ramírez Vique, 2013) algunas ventajas de las aplicaciones nativas se describen a continuación:

- Las novedades llegan primero a este tipo de aplicaciones.
- Permiten las notificaciones Push.
- Permiten recibir valoraciones.
- Se encuentran en las tiendas de aplicaciones de la plataforma para la cual fueron desarrolladas.
- Utilizan las funciones propias del dispositivo.

Algunas desventajas de las aplicaciones nativas se describen a continuación:

- Son costosas si se quieren para diferentes plataformas.
- Es necesario que tengan una licencia de pago por cada plataforma que se utiliza.
- No será aceptable en ningún otro sistema operativo diferente para el que se creó dicha aplicación.

Este tipo de aplicaciones se consideran de mayor potencial pues aprovechan el potencial máximo que nos brindan los dispositivos en los que se ejecutan dando una mejor experiencia al usuario ejemplo al usar la cámara del dispositivo o el uso de ubicación entre otros más.

A continuación, la Tabla-1 nos presenta los datos obtenidos para la creación de las aplicaciones nativas según su plataforma.

Tabla-1. Requerimientos de aplicaciones nativas (Galdamez, Thomas, Pesado, & Lisandro, 2013)

Aplicación nativa para Android	Aplicación nativa para iOS
<ul style="list-style-type: none">▪ <i>Requiere de un JDK (Java Development Kit) software que integra herramientas para crear código en el lenguaje de programación Java.</i>▪ <i>Android SDK (Software Development Kit). Proporciona librerías y herramientas necesarias para construir, probar y depurar aplicaciones Android.</i>	<ul style="list-style-type: none">▪ <i>Está basado en un modelo propietario,</i>▪ <i>Implica contar con una Apple Mac con OS X con Xcode instalado.</i>▪ <i>Lenguaje principal Objective C. Xcode, o el lenguaje especificado por Apple.</i>

Retomando lo anterior ahora tenemos claro que las aplicaciones nativas se descargan de su respectiva tienda de aplicaciones como en este caso App Store y Play Store. A continuación, se muestran dos capturas de pantalla (Figura 1) de diferentes celulares donde cada uno tiene diferente sistema operativo con aplicaciones nativas diseñadas para su propio sistema operativo.

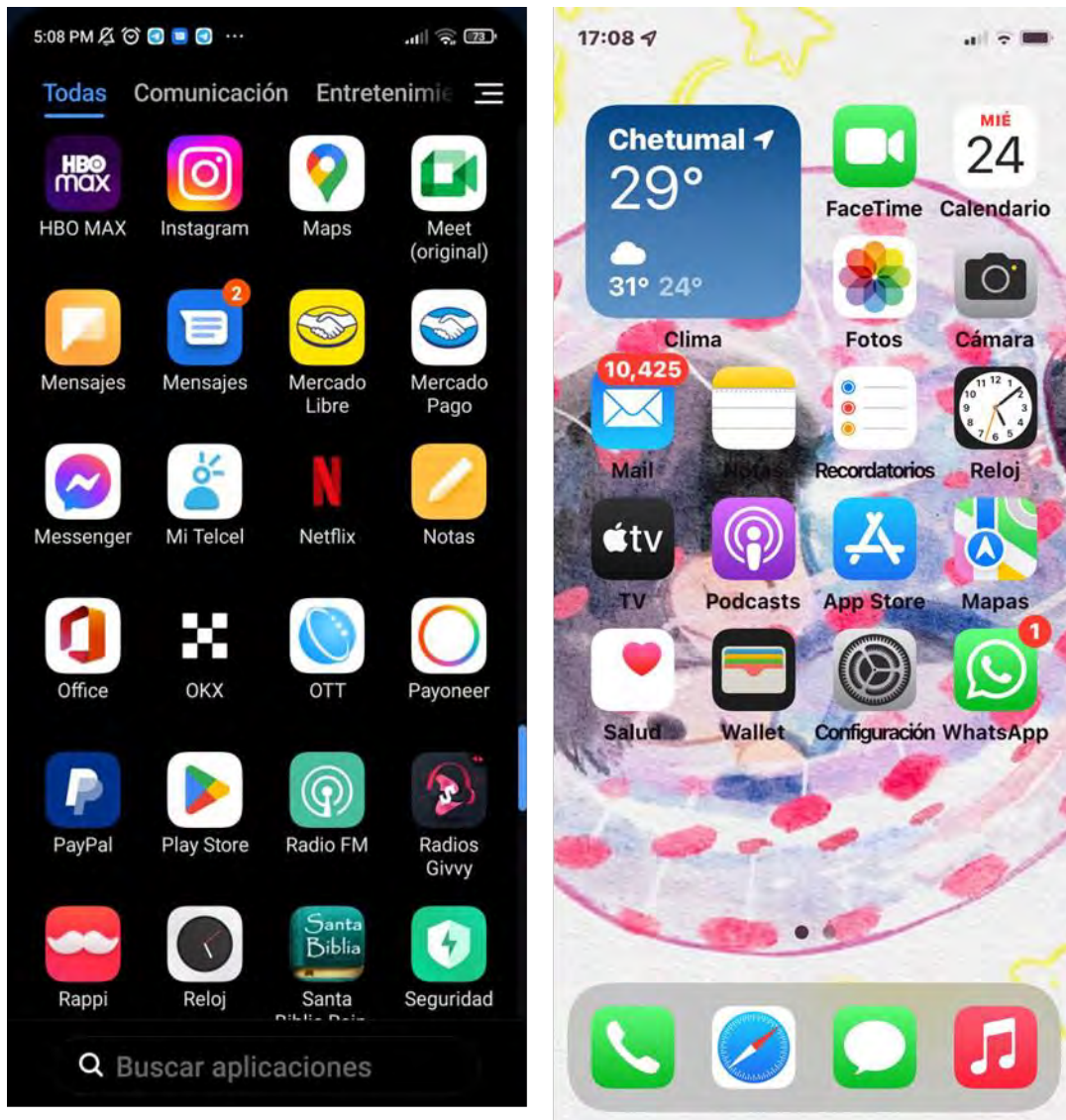


Figura 1. Ejemplo de aplicaciones nativas en Android e iOS.

2.3 Definición de aplicaciones web.

Una aplicación web es más sencilla y económica de crear puesto que para ejecutarse se necesita un navegador web Como Google, Firefox entre otros y no necesita ser instalada en el dispositivo. Cabe mencionar que su entorno no es amigable con dispositivos móviles. el lenguaje de programación para una aplicación web es HTML, CSS, JavaScript; una aplicación web no se encuentra en la tienda de los dispositivos como en la de App Store.

Algunas ventajas de las aplicaciones web son:

- No requiere instalación.
- Acceso a través de un navegador web.
- Es adaptable y fácil de utilizar.
- La aplicación puede estar en la nube,
- Es accesible para cualquier dispositivo que tenga un acceso a internet.
- Las aplicaciones web son multiplataforma.

En las Figura 2, y Figura 3, se puede observar que una aplicación web es ejecutable desde un buscador donde no interfiere el sistema operativo de los dispositivos.



Figura 2 Ejemplo de aplicación web abierta desde un navegador Safari.

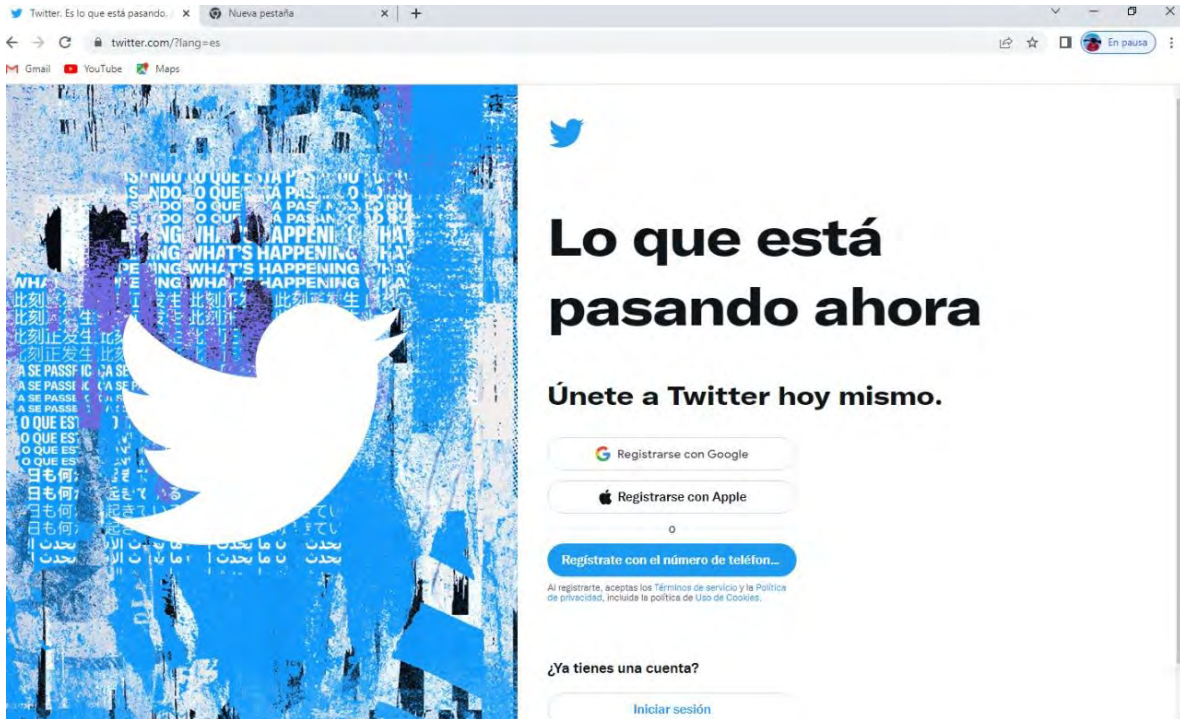


Figura 3 Ejemplo de una aplicación web abierta desde el navegador Google Chrome.

Definimos a una aplicación web como una extensión dinámica de un servidor web o un servidor de aplicaciones. Existen 2 tipos de aplicaciones web (Rondán Martínez, Valderas Aranda, & Pastor López, 2010):

- ✓ Aplicaciones orientadas a presentación, que generan páginas web interactivas en distintos lenguajes de marcado (HTML, XML, etc.) y contenido dinámico a las peticiones de los clientes.
- ✓ Aplicaciones orientadas a los servicios, que constituyen el punto final de un servicio web

2.4 Definición de Aplicaciones Web Progresivas.

Teniendo claro que es una aplicación nativa y que es una aplicación web podemos pasar a lo siguiente. Las **Aplicaciones Web Progresivas** (Progressive Web App – PWA por sus siglas en inglés) se considera una herramienta que está programada en lenguaje web como son HTML5, JavaScript y CSS, pero no se queda ahí pues tiene el comportamiento como una aplicación móvil nativa y a su vez como una aplicación web (“una aplicación combinada”) siendo esta multiplataforma pues se

pueden utilizar en diferentes dispositivos, son menos pesadas y se actualizan rápidamente, también se puede hacer uso de los mensajes Push.

La siguiente Figura 4 nos muestra las principales diferencias que nos ayudan a distinguir una PWA de una aplicación nativa.

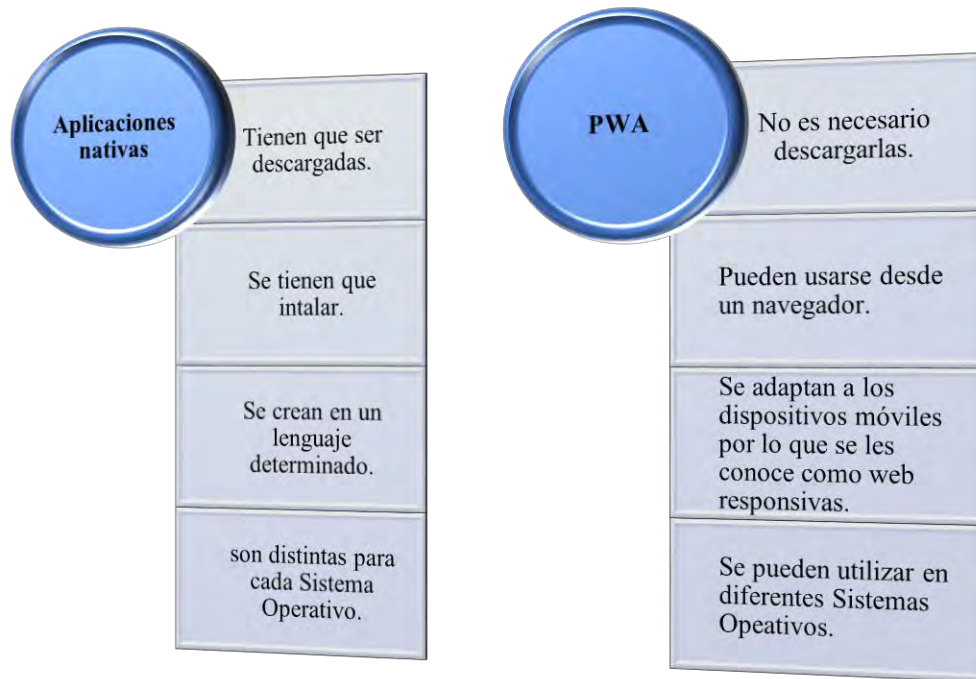


Figura 4 Comparativa con características entre una app nativa y una PWA.

Otra definición de PWA es la definida por Chrome Developers en su página oficial (Rachel, LePage, & Firtman, 2023) nos menciona que:

“Las aplicaciones web progresivas (PWA) son aplicaciones web creadas y mejoradas con API modernas para brindar capacidades, confiabilidad e instalación mejoradas mientras llegan a cualquier persona, en cualquier lugar, en cualquier dispositivo, todo con una base de código única”.

Por su parte, Google ha propuesto 5 bases que se deben cumplir en una PWA para poder considerarse realmente una aplicación web progresiva las cuales son:

- ✓ Funcionamiento offline; significa que debe recuperar la información obtenida del último acceso para poder consultarla cuando se desee;
- ✓ Respuesta a cualquier solicitud en menos de 5 segundos;
- ✓ Experiencia de uso similar a una aplicación nativa;

- ✓ Uso de diseños adaptables;
- ✓ Uso del protocolo HTTPS.

En la revista Computerworld (Raphael, 2021) nos define una aplicación web progresiva de la siguiente manera:

“La forma más sencilla de pensar en una aplicación web progresiva es como una mezcla entre un sitio web y un programa instalado localmente. Una PWA se conecta a su navegador y se basa en tecnologías web regulares, pero esta empaquetada de manera que hace que sea y se sienta más como una aplicación independiente tradicional”.

Una aplicación web progresiva es un híbrido que nos permite combinar lo mejor de las APPS (presencia y funcionalidad) y lo mejor de la web (rapidez, adaptabilidad y accesibilidad). Nos expone (Hernández & Hernández, 2018) que “esta tecnología ofrece al usuario una experiencia inmersiva en pantalla completa, de forma rápida, sin necesidad de conexión, y sin la instalación de APPS nativas”.

CAPÍTULO 3 FUNCIONAMIENTO DE UNA APLICACIÓN WEB PROGRESIVA (PWA)

Las PWA están pensadas en Mobile First pues esto quiere decir que esta creado con la idea de que se acceda a los sitios desde dispositivos móviles y tabletas. Las PWA progresivamente implementan características como notificaciones push, funciona sin conexión a internet usa características nativas del dispositivo, se actualiza constantemente, es atractiva para los usuarios es confiable pesa menos que una nativa y es más rápida su carga. Las PWA son aplicaciones que aprovechan el uso de APIs de las nuevas tecnologías Web; requieren una Web App Manifest y utilizan los Service Worker, entre otras.

3.1 Archivo de manifiesto

Podemos acceder a una PWA a través de la web, pero para acceder a través de un icono de nuestra pantalla las PWA hacen uso de un manifiesto que se explica a continuación.

El manifiesto de una aplicación está en formato JSON; que permite especificar iconos en la pantalla de inicio, es un archivo *manifest.json* que genera un acceso directo de la app web.

A continuación, se muestran algunas de las características que contiene dicho manifiesto.

Tabla 2. Características Básicas De Un Manifiesto JSON

Manifiesto de la aplicación formato JSON	Name
	Description
	Icons
	Start url
	Display
	Orientation
	Theme_color
	Bckground_color

En el capítulo 4 se abordarán más a fondo estas características de cada componente que se encuentran en el manifiesto JSON mencionadas en la tabla 2.

Ester Ribas citada en Vidal (2019) nos explica que:

“Uno de los elementos más importantes que conforman las PWA es el manifiesto de la aplicación y que está en formato JSON. Este, nos permite controlar cómo se verá nuestra aplicación en áreas donde normalmente los usuarios tienen aplicaciones nativas.”

3.1.1 El formato JSON

Por sus siglas en inglés Java Script Object Notation (JSON), que traducido quiere decir Notación de Objetos en JavaScript. Ahora bien, JSON es un formato basado en texto para representar datos estructurados en la sintaxis de objetos en JavaScript; pero JSON no es JavaScript.

A continuación (Mozilla, 2022) nos describe lo siguiente.

“Aunque JSON es muy parecido a la sintaxis de objeto literal de JavaScript, puede ser utilizado independientemente de JavaScript; y muchos entornos de programación poseen la capacidad de leer (convertir) y generar JSON.”

“Un objeto JSON puede ser almacenado en su propio archivo, que es básicamente sólo un archivo de texto con una extensión. json, y una MIME type (en-Us) de application/json.”

3.2 Función Por Medio De Los Service Worker

Los Service Worker son scripts, es decir, un documento de texto donde se coloca instrucciones u órdenes que posteriormente serán ejecutadas en segundo plano. Un Service Worker permite interceptar solicitudes de red, manejar mensajes push entre muchas tareas más.

Tomando en cuenta la información obtenida de (Sunil Tandel & Jamadar, 2018) nos muestra que los trabajadores de servicio (Service Worker) realizan las siguientes funciones:

1. Almacenan en caché el Shell de la aplicación.
2. Actualizan el contenido en segundo plano.

3. Obtienen la identificación de notificaciones push del usuario para enviar la notificación.
4. Invalidan el caché cuando es necesario.
5. Emplean el protocolo HTTPS para garantizar la seguridad de la aplicación.

A continuación, se muestra en la Figura 5. la arquitectura de un Service Work tomando como base la obtenida de la publicación de (Rivera Moncayo, 2020).

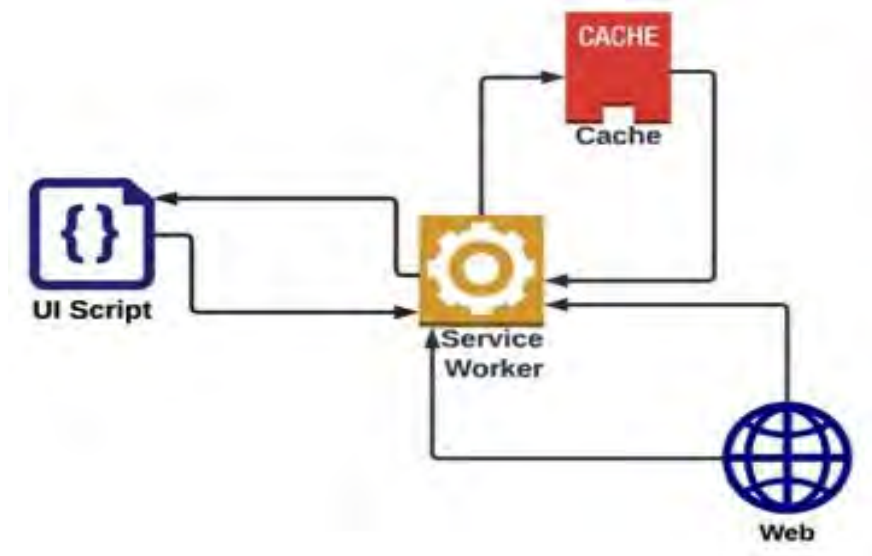


Figura 5 Arquitectura del Funcionamiento de un Service Worker

El objetivo del SW es actuar como intermediario entre la aplicación del usuario e internet; Los SW tienen las siguientes características (Rivera Moncayo, 2020) :

- Ejecutan su propio script en un contexto global.
- No está vinculado a una página web en particular.
- No puede realizar modificaciones a los elementos de la página web.
- Debe emplearse HTTPS.

3.3 Proceso del Funcionamiento De La PWA.

Para facilitar la comprensión del funcionamiento de las PWA se muestra a continuación la Figura 6 donde explica de manera general del proceso que se lleva a cabo.

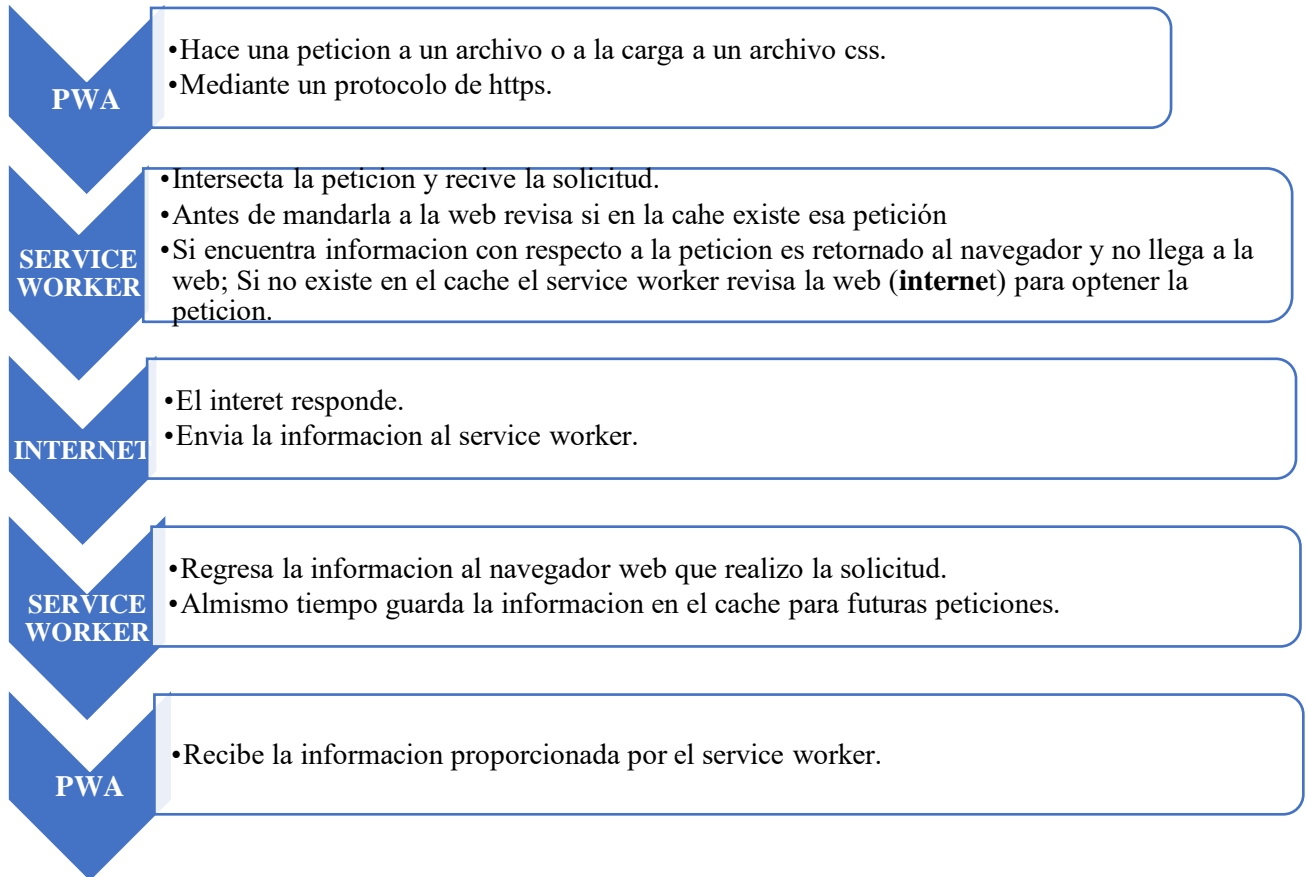


Figura 6. Proceso Descriptivo de una PWA

El proceso de una PWA es el que los usuarios no ven al ejecutar la aplicación, pero si obtienen los resultados que esperan.

3.4 Proceso para la inicialización de una PWA.

El proceso según (Lanza Ortega, 2019) consta de las etapas que se definen a continuación:

1. Inicia el proceso de descarga de la página web por parte del o los usuarios a través de un navegador.
2. Tomando en cuenta que no todos los navegadores permiten la ejecución de un Service Worker, parte del código JavaScript de la página de inicio incluirá la validación de la capacidad del navegador para proveer aplicaciones web progresivas.
3. Si se recibe una respuesta positiva entonces el navegador buscará en la ubicación solicitada.
4. Posteriormente se procederá a registrar.
5. Aquí se iniciará el proceso de instalación.
6. El comando *register* detectará si el Service Worker existía previamente en el navegador y actuará en consecuencia.

CAPÍTULO 4 COMPONENTES TECNOLÓGICOS DE UNA PWA

El artículo de (Lanza Ortega, 2019) nos describe que una aplicación PWA debe cumplir satisfactoriamente los siguientes tres criterios de referencia:

- Debe ejecutarse bajo HTTPS.
- Debe incluir un Web App Manifest.
- Debe hacer uso de los Service Workers (Mendoza, 2019).

Los componentes tecnológicos para una PWA se describen a continuación.

Las aplicaciones web deben de ejecutarse a través de una red segura ya que ayuda a establecer la red como un sitio confiable. Ahora bien, un Service Worker es un script que va a controlar las solicitudes que se le hagan a la red y el almacenamiento en el caché. Con los Service Worker se puede obtener webs más rápidas y experiencia fuera de línea es decir sin servicio a internet. El uso del archivo manifiesto es el que se encuentra en formato JSON el cual controla como se muestra la aplicación al usuario y hace que las aplicaciones PWA sean detectadas por los usuarios finales. (Mozilla, Aplicaciones Web Progresivas, 2022)(pp.)

4.1 Archivo Manifiesto.

La aplicación manifiesto que traducido quiere decir manifiesto de la aplicación es simplemente un JSON donde se encontrara información sobre la aplicación. La cual nos servirá para instalarla donde podremos tratarla como una aplicación nativa (Tapia J. L., Tesis, 2020) dice lo siguiente sobre la función principal de un archivo manifiesto.

“Su función principal es informar a los distintos dispositivos que vayan a instalar la aplicación web cuál es el aspecto que debe adoptar, es decir, nos dirá cuál es el icono que deberá mostrarse en la pantalla, si ocupará pantalla completa o dejará la barra de navegación y este tipo de variaciones”.

El comportamiento cuando inicializa la aplicación de acuerdo con el archivo de manifiesto es el siguiente (Alvarez, Edison , Yaguargos, & Santiago , 2020) :

- “Establecer el nombre corto de la aplicación y los distintos tamaños de imagen para que se adapten a diferentes dispositivos”.
- “Durante el proceso de descarga de recursos, establecer un estado intermedio para dinamizar el tiempo de carga”.
- “Establecer una configuración predeterminada en el navegador que evite transiciones bruscas”.
- “Proporcionar la información base y necesaria para que se ejecuten las características de una Aplicación Web Progresiva”.

4.1.1 Estructura De Los Elementos Para El App Manifiesto.

Las App manifestó que como ya se sabe es un archivo JSON el cual está compuesto por varios campos que en conjunto conforman el archivo manifiesto donde cada uno proporciona una parte de información necesaria. Como nos dice (Tapia J. L., Tesis, 2020) en su obra; algunos campos son obligatorios para que la instalabilidad sea posible y otros son optativos, pero cada uno proporciona una parte de información distinta.

```

{
  "short_name": "Clima",
  "name": "Clima: ¿ Necesito un Paraguas?",
  "description": "información del pronóstico del Tiempo",
  "icons": [
    {
      "src": "/images/icons-192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "/images/icons-512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": "/?source=pwa",
  "background_color": "#3367D6",
  "display": "standalone",
  "scope": "/",
  "theme_color": "#3367D6"
}

```

Figura 7 Ejemplo de la estructura de un archivo manifiesto

El contenido de la Tabla 3 muestran los elementos o campos del manifiesto y su descripción.

Tabla 3 Tabla descriptiva de los elementos del manifiesto (Foundation, 2022)

Propiedades	Descripción
background_color	Define el color de fondo deseado para la aplicación.
description	Proporciona una descripción general sobre lo que hace la aplicación.
dir	<p>Especifica la dirección del texto, junto con lang representa correctamente los idiomas que se escriben de derecha a izquierda.</p> <p>ltr: (izquierda) rtl: (derecha a izquierda) auto: (Unicode bidireccional)</p>
display	<p>Define el modo de visualización preferido para la aplicación web.</p> <p>fullscreen: Utiliza toda la pantalla disponible.</p>

	<p>standalone: La aplicación se mostrará como una app independiente, puede tener su propia ventana, icono, etc.</p> <p>minimal-ui: Se mostrará como una aplicación independiente, pero tendrá un mínimo de elementos de interfaz.</p> <p>browser: Se abrirá en una pestaña nueva del navegador o una ventana nueva, dependiendo del navegador y plataforma.</p>
icons	<p>Especifica un array de imágenes que pueden servir como iconos de aplicación en diferentes contextos.</p> <p>valores:</p> <p>sizes: Una cadena que contiene separados por espacios las dimensiones de la imagen.</p> <p>src: La ruta de acceso al archivo de imagen.</p> <p>type: Permite que un agente de usuario ignore rápidamente las imágenes de los tipos de medios que no admite.</p>
lang	<p>Especifica el idioma principal para los miembros name y short_name. Este valor es una cadena que contiene una sola etiqueta de idioma.</p>
name	<p>Proporciona un nombre legible por humanos para la aplicación, ya que está destinado a mostrarse al usuario.</p>
orientation	<p>Defina la orientación por defecto para todos los contextos de navegación de nivel superior de la aplicación web.</p> <p>valores: any, natural, landscape, landscape-primary, portrait, portrait-primary, y portrait-secondary</p>
prefer_related_applications	<p>Especifica un valor booleano que sugiere que el agente de usuario le indique al usuario que las aplicaciones específicas relacionadas están disponibles, y recomendadas sobre la aplicación web. Esto solamente debería ser usado si la aplicación nativa relacionada realmente ofrece algo que la aplicación web no puede hacer.</p>
related_applications	<p>Especifica una matriz de "objetos de aplicación" que representan aplicaciones nativas que la plataforma subyacente puede instalar o acceder a ella.</p> <p>valores:</p>

	platform: La plataforma en la que se encuentra la aplicación.
	url: La URL en la que se puede encontrar la aplicación.
	id: El ID utilizado para representar la aplicación en la plataforma especificada.
scope	Define el ámbito de navegación en el contexto de la aplicación web. Esto básicamente restringe qué paginas se pueden ver cuando se aplica el manifiesto. Si el usuario navega fuera del ámbito de la aplicación, continúa como en una web normal.
short_name	Proporciona un nombre corto para la aplicación.
start_url	Especifica la url que se carga cuando el usuario lanza la aplicación desde un dispositivo.
theme_color	Define el color por defecto para la aplicación.

Como nos dice el autor en su investigación. (Tapia J. L., Thesis, 2020) Si el manifiesto está presente en la raíz del proyecto y debe ser enlazado a su archivo correspondiente `<link rel="manifest" href="/manifest.json">`.

Chrome entre otros navegadores automáticamente activará un banner en la parte inferior de la pantalla para notificar al usuario de que puede instalar la aplicación. Si el usuario acepta, el icono que hemos definido en el manifiesto aparecerá en la pantalla de inicio el teléfono como si se tratara de una aplicación nativa (Yaguargos Castro, 2020).

4.2 Service Worker.

Como se sabe un Service Worker se considera un archivo con una secuencia de comandos creados que se ejecutan en segundo plano dentro de un navegador y escucha eventos aun cuando una página web está abierta o no lo está.

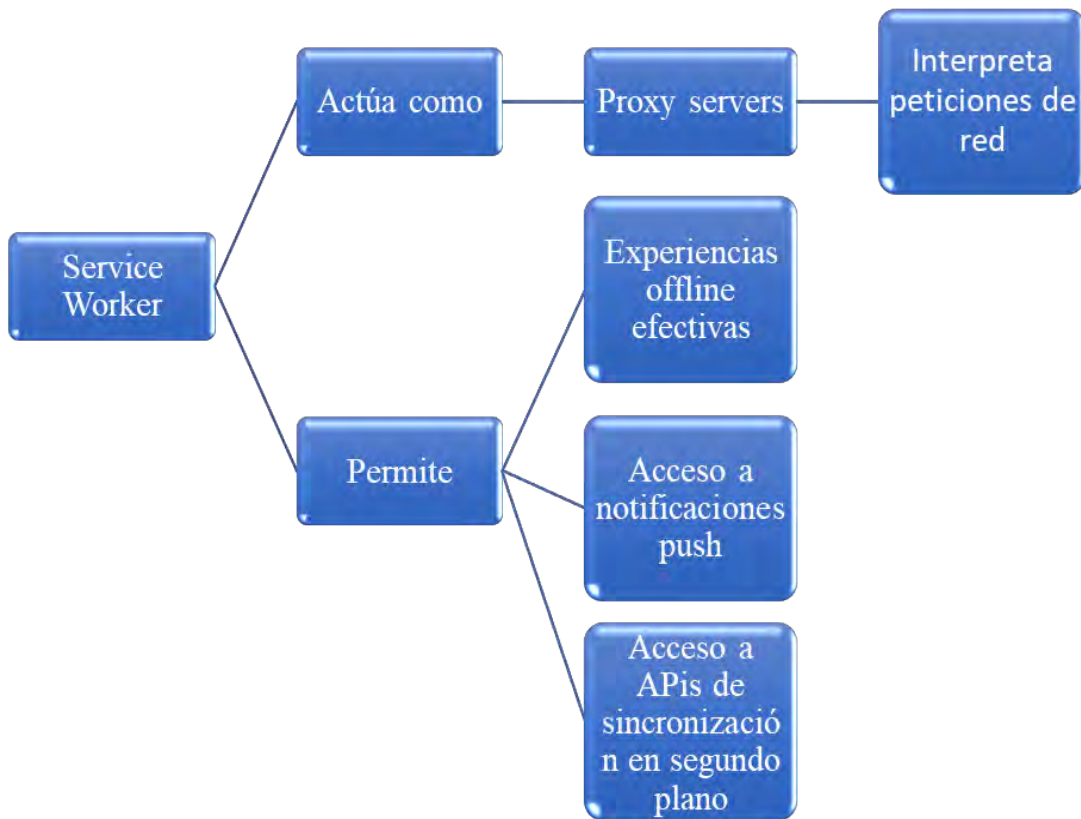


Figura 8 Descripción básica del Service Worker

4.2.1 Función de un Service Worker

El Service Worker como nos dice (Otero Carbón, 2020):

Es el encargado de garantizar que el contenido de la página se pueda ver offline mediante el almacenamiento en caché. A mayores de esa capacidad offline, el uso del Service Worker puede ser justificado también para proporcionar a los usuarios un sitio web más rápido y confiable.

Su funcionamiento puede describirse de la siguiente forma (Alvarez, Edison , Yaguargos, & Santiago , 2020):

1. El proceso de service worker se pone en ejecución una vez que se realice el inicio del navegador. En ese momento, se realiza el inicio.
2. A partir de ahí el service worker se mantiene en ejecución.

3. No quiere decir que tengan un tiempo de vida más largo, ya que se pueden producir algún tipo de modificaciones en los mismos mientras la página se mantiene abierta (Lanza Ortega, 2019).

Ahora bien, cómo funciona un service worker: el usuario accede a una página web por primera vez → el Service Worker comenzará a descargarse → se instala el SW (en esta etapa es cuando comienza a prepararse la caché para tener recursos).

En la Figura 9 se describe de manera ilustrativa el funcionamiento de Service Worker para facilitar la comprensión de este proceso descrito anteriormente.

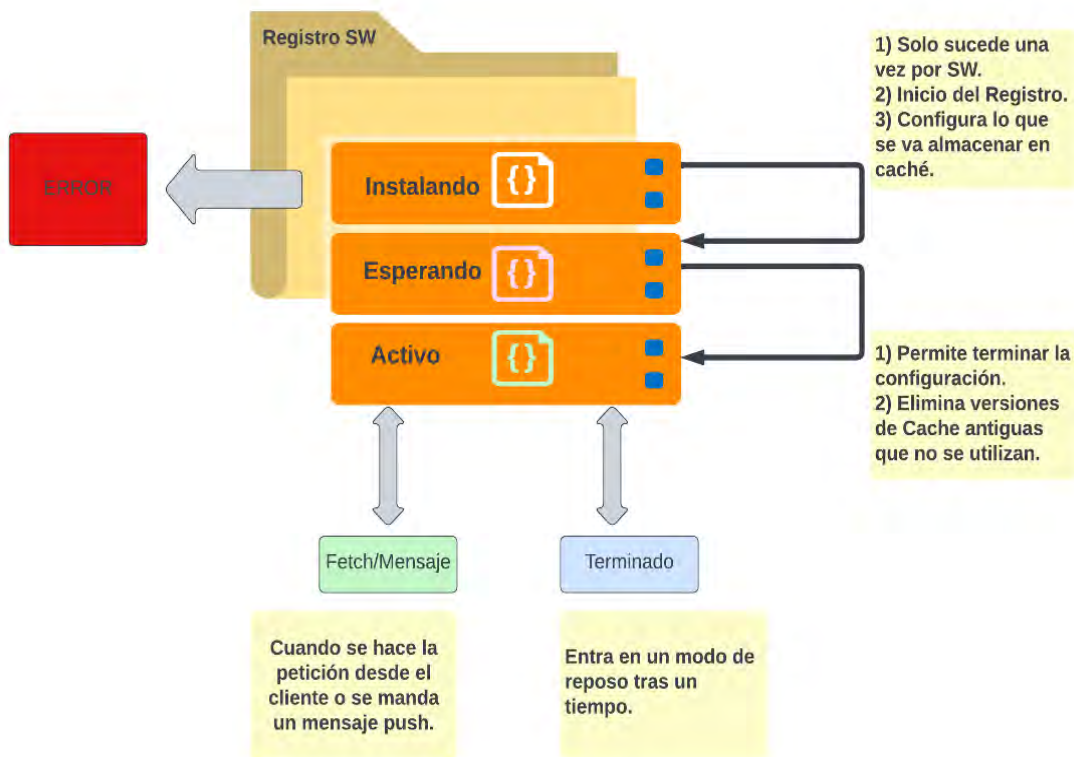


Figura 9. Funcionamiento de un Service Worker (elaboración propia).

Un Service worker es un script que permite interceptar los requests de la red y controlar los y también el almacenamiento del cache del browser. Como tienen acceso al manejo de los requests, pueden llegar a ser muy poderosos por lo que se tiene que manejar con mucho cuidado.

4.2.2 Cómo agregar un Service Worker

- ✓ Crear el archivo JavaScript del Service Worker.

Lo primero es crear el archivo *JavaScript sw.js* y lo agregamos la raíz del sitio web. Suele estar ubicado en el *index.html*. Existen diferentes librerías y servicios para generar código de Service Worker y facilitarnos, por ejemplo, el Workbox creado por Google (García de Zúñiga, 2018).

- ✓ Registrar el Service Worker.

Creado el archivo *sw.js*, registraremos el Service Worker con más código JavaScript. Este proceso puede tramitarse a través de una sola llamada a un método del API de los Service Worker del navegador. Debemos cerciorarnos de que el cliente soporta dicha tecnología (Yaguargos Castro, 2020).

Este es un ejemplo del código para registrar el Service Worker:

```
if ('serviceWorker' in navigator) {  
  window.addEventListener('load', function() {  
    navigator.serviceWorker.register('/sw.js').then(function(registration) {  
      // Si es exitoso  
      console.log('SW registrado correctamente');  
    }, function(err) {  
      // Si falla  
      console.log('SW fallo', err);  
    });  
  });  
}
```

- ✓ Service Worker registrado.

Verificar el registro correcto del Service Worker, mediante las herramientas de desarrollador, como por ejemplo las de Chrome Developer Tools.

Como nos muestra (web.dev); Para realizar la verificación de si un Service Worker está registrado se puede realizar lo siguiente en navegadores basados en Firefox y Chromium (Microsoft Edge, Google Chrome o Samsung Internet):

1. Abra las herramientas para desarrolladores, luego haga clic en la pestaña Aplicación.
2. En el panel izquierdo, seleccione Trabajadores del servicio.
3. Verifique que la URL del script del trabajador del servicio aparezca con el estado "Activado". (Aprenderá lo que significa este estado en la sección del ciclo de vida de este capítulo). En Firefox, el estado puede ser "En ejecución" o "Detenido".

4.2.3 Ciclo de vida de un Service Worker

¿Cuándo empieza el ciclo de vida de un service worker? Este empieza en el momento exacto en el que un usuario navega a través de una página web desde su dispositivo es ahí cuando se ejecuta el proceso de registro; durante ese proceso en el que se realiza el registro el navegador descarga todo o necesario, analiza y ejecuta el service worker; pero si en algún momento en el que se está llevando el proceso anterior existiera o surgiera algún error entonces el registro se rebotara y el service worker será rechazado.

Es importante tener claro que un service worker es como un proxy que intercepta todas las peticiones, por ser “event-driven”, y será capaz de decidir cómo proceder. Hay muchas posibilidades: comprobar si tenemos la respuesta en caché, si no está ira a la red a buscarla, quizás ir directo a la red o solo a la caché. A continuación, se muestra en la Figura 10 la descripción del ciclo de vida de los SW tomada de las sigs. fuentes (Mozilla Corporation’s, 2022) y (Tapia J. L., 2020).

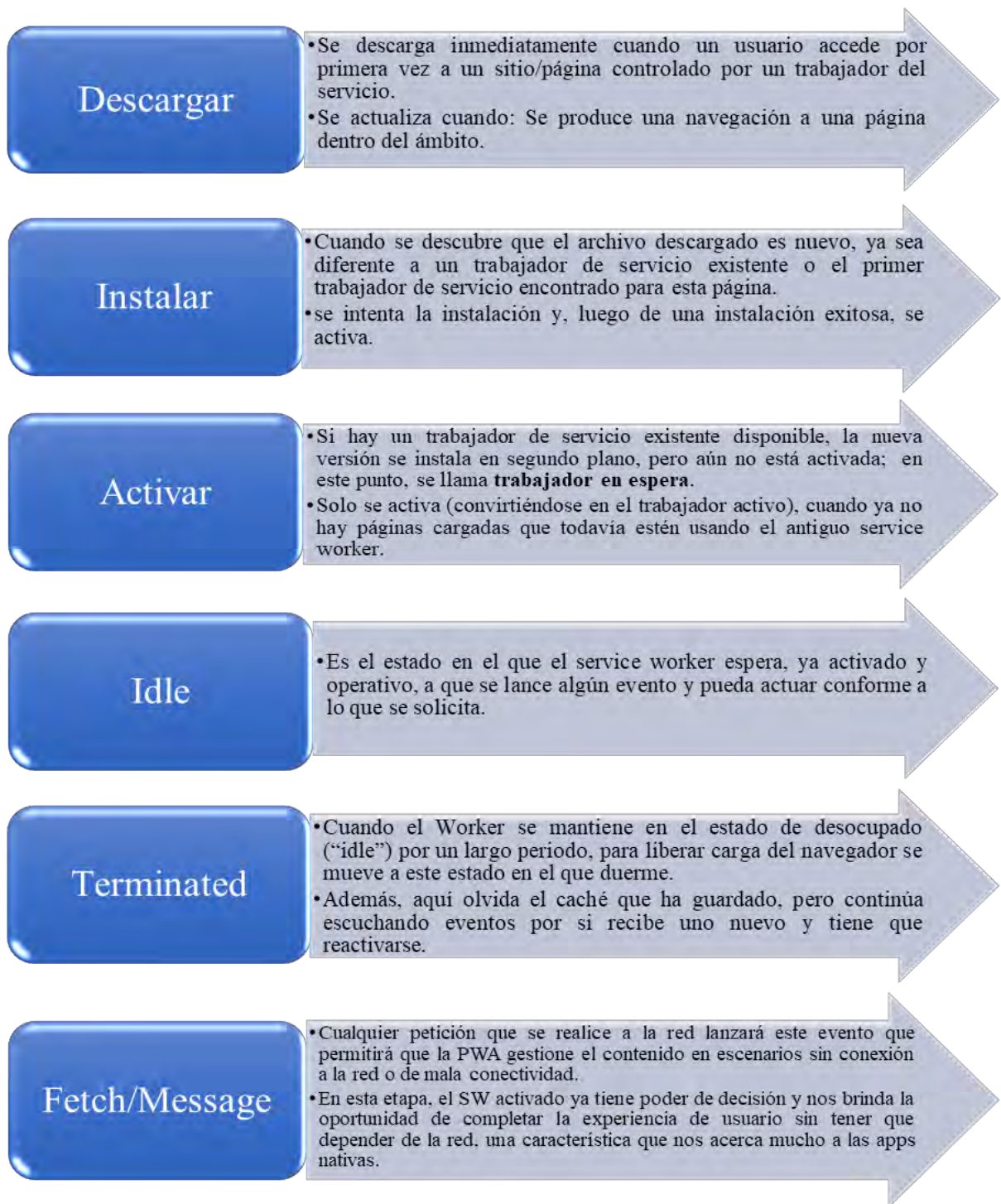


Figura 10. Ciclo de vida de un Service Worker

4.3 Protocolo seguro para las Aplicación web progresiva.

HTTP es un protocolo no seguro, lo que significa que la transmisión de información se hace de forma que cualquiera que tenga acceso al canal va a poder recuperarla. Para evitar esos problemas y resolver la inseguridad nació el HTTPS, el cual es un protocolo seguro, en el que se genera la confianza de la información pues no puede ser interceptada y decodificada por un tercero. Para conseguirlo emplea un sistema de cifrado SSL.

Una aplicación web se debe servir a raves de una red segura por eso se utiliza el protocolo HTTPS (HTTP Secure); normalmente utiliza SSL (en-US) o TLS (en-US).

HTTPS es esta conexión segura que permite a los clientes y servidores intercambiar datos confidenciales, elimina riesgos de violaciones de seguridad, manipulación de contenido, espionaje. Es importante tener en cuenta que para que la mayoría de las funciones con las que cuentan las PWA como lo son la geolocalización e inclusive los servicios *workers*; solamente podrán estar disponibles cuando la aplicación con la que se esta interactuando ha cargado mediante el protocolo HTTPS.

En la Tabla 4 se muestran las diferencias más claras que tiene el protocolo https ante el protocolo http como mejoría en la seguridad.

Tabla 4 Principales diferencias entre protocoló HTTP Y HTTPS

HTTP	HTTPS
No cifra la información	Cifra la información
Puerto por defecto 80	Puerto por defecto 443
Última versión (HTTP/2)	Basado en HTTP
No necesita tener un certificado instalado	Necesita certificado SSL

La forma más simple para saber si los datos introducidos en una web viajan de forma segura, es verificar en la sección de barra de direcciones de nuestro navegador si aparece **https://** lo que indicaría que es una web segura, o si aparecerá **http://** que significa que no cuenta con cifrado SSL. Para que se

produzcan transmisiones seguras, HTTPS debe estar habilitado en el servidor y el navegador del cliente debe estar habilitado con el protocolo SSL. Hoy en el mercado la mayoría de los navegadores ya están habilitados para el protocolo SSL.

4.3.1 Proceso de HTTPS

En resumen, el protocolo HTTPS se basa en la transmisión de los certificados TLS/SSL, que verifican que un determinado proveedor es quien dice ser, evitando que sea un tercero quiera hacerse pasar por el verdadero proveedor. En la imagen siguiente se muestra de manera grafica cual es el proceso que utiliza el protocolo HTTPS.

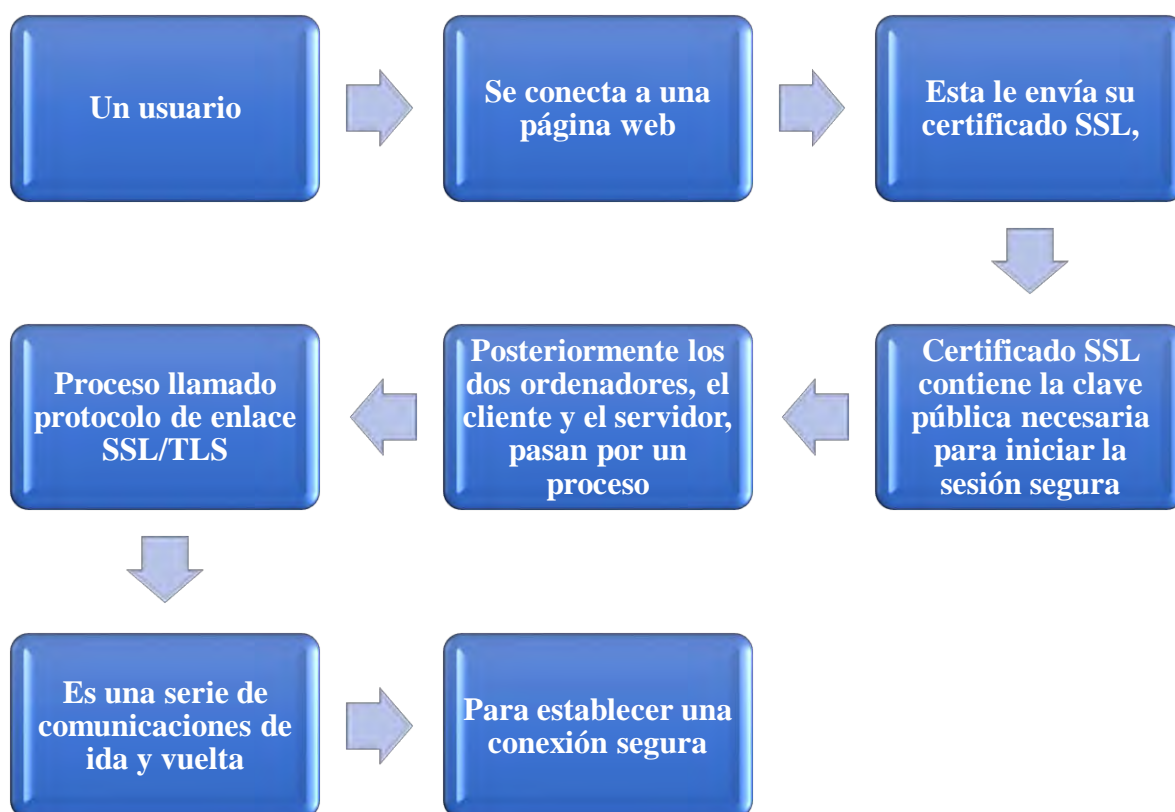


Figura 11. Gráfico descriptivo del protocolo HTTPS.

CAPÍTULO 5 APLICACIONES WEB PROGRESIVAS VS APLICACIONES MÓVILES NATIVAS

5.1 Ventajas y desventajas que nos brindan las PWA.

En la Tabla 5 se muestran a continuación las ventajas y desventajas de las PWA obtenidas al estudiar distintos trabajos de investigación documental, así como páginas web de estudios relacionadas a las PWA.

Tabla 5. Ventajas y desventajas que encontramos en las PWA.

Ventajas	Desventajas
La programación de las PWA es más sencilla y económica.	Se encuentra en desarrollo la compatibilidad con navegadores y sistemas operativos móviles,
También permite realizar al mismo tiempo una página web tradicional y una aplicación multiplataforma	Aun no se sabe que tantas funciones nativas de los dispositivos puedan ser soportadas en el futuro por las PWA.
Son aplicaciones que se encuentran de manera fácil en cada navegador; y no son necesario descargar. Por lo que se pueden probar sin ocupar espacio en el dispositivo.	No todos los exploradores soportan las PWA en todas sus cualidades; lo que hace pensar si en el futuro los dispositivos como por ejemplo los iOS soportaran estas tecnologías.
El formato WebAPK permite instalarlo en el sistema operativo	Las WebAPK representan una tecnología que no muchos conocen.
Las PWA ya pueden hacer acceder a las funciones nativas de los dispositivos móviles, como las notificaciones push, la geolocalización, la cámara, e micrófono, entre otras	No se pueden usar aun todas las funciones nativas de los terminales como por ejemplo los contactos. Es incierto si en algún momento las PWA puedan integrar en su totalidad todas las funciones de la misma manera que usan las nativas.
No es necesario descargar actualizaciones porque las aplicaciones se actualizan automáticamente cada vez que se abren (si tienes conexión a internet).	No se sabe si las PWA se impondrán en el mercado, pero su situación es prometedora en el futuro.
Pueden utilizarse sin conexión a internet; lo que significa que funcionan en modo fuera de línea. Por medio de <i>service workers</i> que	Fuera de una conexión a internet la aplicación no se actualiza.

guarda y almacenan en la caché las funciones y también la información.	
Son responsivas pues deben de funcionar con cualquier dispositivo ya sea computadora, Tablet o teléfono inteligente.	Tiene mucho que ver con la compatibilidad de los buscadores.
Diseño fácil de manejar y entender ya que imita a las aplicaciones nativas es decir menús simplificados y fáciles de ubicar y con facilidad de interactuar en su entorno.	Es similar a las nativas su entorno grafico por lo que no es fácil distinguir sus diferencias de una app nativa a una PWA.
Una PWA es segura ya que usa siempre el protocolo HTTPS para mantener seguro los datos de los usuarios.	
Para compartir una PWA solo se requiere únicamente la URL para compartirla.	Si el dispositivo no tiene un buscador compatible no podrá ejecutar correctamente la aplicación PWA
Las PWA tienen un rendimiento mejorado; su velocidad de ejecución de las aplicaciones web progresivas mejora debido a que puede funcionar de manera eficiente, es similar a un sitio web, almacena en al caché y sirve imágenes y textos y otros contenidos.	

5.2. Buscadores compatibles con las PWA

Para ejecutar una PWA se necesitan dos requisitos: un buscador compatible con la aplicación y un servicio de PWA habilitado.

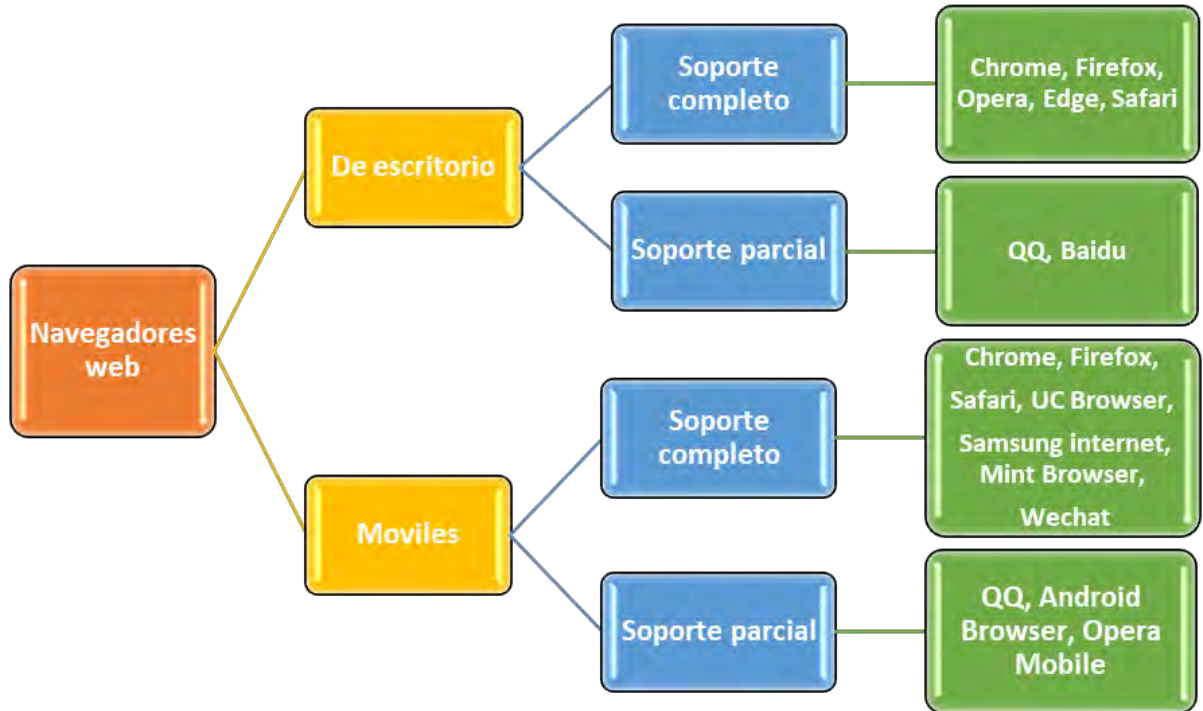


Figura 12. Buscadores que soportan PWA

5.3 Comparación de las Ventajas de una PWA y una aplicación móvil nativa.

Teniendo en cuenta que las aplicaciones web progresivas son por así decirlo la evolución de las aplicaciones web, juntando lo mejor de las aplicaciones más conocidas las apps nativas y las aplicaciones web dando como surgimiento a las PWA.

Las PWA asemejan características a las aplicaciones nativas, pero como todo tiene diferencias entre ellas. Una de las diferencias más marcadas entre las aplicaciones nativas y las PWA, es que las nativas conllevan mucho más trabajo de programación a comparación del desarrollo de una PWA, donde no se tienen que modificar para Android e iOS.

Ahora bien, en la siguiente Tabla 6 se muestran las características generales que diferencian a las PWA y las APP nativas.

Tabla 6. Características que brinda una PWA ante una Aplicación Nativa

Características	Aplicación Nativa	Aplicación Progresiva	Web
<i>Lenguaje</i>	Java, -C, NET	HTML, JavaScript	CSS,
<i>Funciones online</i>	SI	SI	
<i>Notificaciones Push</i>	SI	SI	
<i>Acceso a la pantalla de inicio</i>	SI	SI	
<i>No requiere descarga</i>	NO	SI	
<i>Necesita estar en una tienda de aplicaciones</i>	SI	NO	
<i>Conectable y compatible</i>	NO	SI	
<i>Patrocinado por Google</i>	NO	SI	
<i>Bajos requerimientos de datos</i>	SI	SI	
<i>No requiere actualizaciones</i>	SI	NO	
<i>Costo de mantenimiento</i>	Alto	Bajo	
<i>Tiempo de Desarrollo</i>	Alto	Bajo	

CAPÍTULO 6 POSIBILIDADES QUE BRINDAN LAS APLICACIONES WEB PROGRESIVAS (PWA).

Progressive web apps no solo es moda, sino que son un territorio técnico aun con mucho potencial que explorar, donde el termino progresivo hace referencia a las habilidades que presentan estas aplicaciones para adaptarse lo que significa que entre más capacidades aporte el dispositivo (terminal) y el navegador empleado para su uso, más capacidades puede ofrecer una PWA.

Las PWA podrían representarse como un rival para las ya conocidas aplicaciones nativas en unos años si Apple acepta el uso del formato PWA; por ahora Apple ha apostado su máximo en saca provecho del sistema de aplicaciones para las plataformas conocidas como iOS y MacOS y para su propia App Store, por lo que muestra menos interés en el proyecto de las PWA.

Se sabe que los navegadores de los mayores proveedores (Google, Mozilla, Microsoft, Opera) van a apostar intensamente por una mayor compatibilidad del formato abierto PWA con sus productos (IONOS, 2017). En unos años más se sabrá si las aplicaciones progresivas podrán competir realmente con las nativas, aunque las oportunidades pintan bien a la vista de los renombrados apoyos con los que cuenta.

6.1 Casos de éxito

Muchas empresas han implementado las PWA y han tenido éxito, por ejemplo:

- ✓ Trivago vio un aumento del 150 % para las personas que agregan su PWA a la pantalla principal. El aumento de la contratación llevó a un aumento del 97 % en las salidas a las ofertas de hoteles (Google, 2017)

- ✓ La PWA de Starbucks ha aumentado los usuarios activos diarios dos veces. Los pedidos en el escritorio son casi la misma velocidad que el móvil (Brunelle, 2018)
- ✓ Pinterest con un enfoque en el crecimiento internacional, decidieron crear una experiencia móvil, desarrollando una PWA (Oraá, 2019). Y sin duda los resultados fueron increíbles, Pinterest aumentó el tiempo que los usuarios invierten en su plataforma. Dato obtenido de Orbe innovación (Bernal,2022).

En una publicación realizada en tweet se dio a conocer la siguiente información con respecto a LYFT (Bidelman, 2021):

- ✓ *Los usuarios de dispositivos más antiguos realizan un 11 % más de viajes en lyft PWA que los usuarios nativos y hacen clic en "Instalar PWA" un 40 % más a menudo que en "descargar aplicación".*

Para dar contexto LYFT es un servicio de transporte similar a Uber, orientada a viajeros en el que usuarios locales utilizan sus coches para ganarse ofrecer servicios de transporte, ya que cuenta con su PWA. Así como éstos se encuentran muchos más casos de éxito para empresarios y creadores de aplicaciones web que implementan las PWA.

CAPÍTULO 7 CONCLUSIONES

Una aplicación web progresiva es aquella aplicación que ha surgido al tomar lo mejor de las características de una aplicación web y de las aplicaciones nativas unificándolas para así crear las aplicaciones web progresivas.

Las PWA son compatibles con los distintos sistemas operativos y no tienen necesidad de descargarse e instalarse en los dispositivos evitando así la saturación en el almacenamiento de los dispositivos; y aprovechan también los componentes tecnológicos de los dispositivos como la cámara, ubicación, calendario, el uso de los mensajes push entre otros.

En comparación con las Apps Nativas, las PWA son más ligeras, son compatibles con los distintos sistemas operativos por lo que el costo de creación disminuye significativamente, ya que no es necesario crear una aplicación para cada sistema operativo; otra de sus características sobresaliente es que se asemeja al entorno gráfico de una app nativa por lo que es muy difícil de distinguir las diferencias, también las PWA guardan la información para utilizarlas fuera de línea lo que es muy útil.

Una de las desventajas de las PWA es que no se encuentran en las tiendas de aplicaciones de los dispositivos por lo que para buscarlas hay que entrar directo en el buscador del dispositivo en donde se utilizará.

Para concluir con esta investigación se presentaron los antecedentes, características y estructura general de la tecnología de una PWA, que es otra opción para crear una aplicación de negocios, pues el costo de inversión es menor y está al alcance de cualquier sistema operativo. Estas aplicaciones se adaptan a los cambios constantes y adaptaciones de la tecnología debido a los Service Worker que facilitan su actualización. Por último, se destaca que una PWA es segura y

confiable ya que usa el protocolo HTTPS por lo que es la opción adecuada cuando se está iniciando en el desarrollo de aplicaciones para los negocios.

BIBLIOGRAFÍA

- Mozilla Corporation's. (1 de Diciembre de 2022). *https://developer.mozilla.org/*. Obtenido de mdn web docs: https://developer.mozilla.org/es/docs/Web/API/Service_Worker_API
- Alvarez, M., Edison , H., Yaguargos, C., & Santiago , D. (2020). APLICACIÓN WEB PROGRESIVA (PWA) PARA LA AUTOMATIZACIÓN DE LOS PROCESOS DE GESTIÓN E INFORMACIÓN EN LIGA DEPORTIVA PARROQUIAL TOTORAS". *Trabajo de Titulación Modalidad: Proyecto de Investigación presentado previo a la. UNIVERSIDAD TÉCNICA DE AMBATO, Ambato, Ecuador*. Recuperado el 2 de Agosto de 2023, de <https://repositorio.uta.edu.ec/jspui/handle/123456789/31308>
- Bass, L., Clements, P., & Kazman, R. (2003). *Software architecture in practice*. Addison-Wesley Professional.
- Bernal, J. (1 de abril de 2022). *Orbe Innovación*. Obtenido de <https://www.orbesoftware.com.ar/>: <https://www.orbesoftware.com.ar/top-3-casos-de-progressive-web-apps-que-la-rompieron.html>
- Bernal, J. (1 de Abril de 2022). *Top 3 casos de Progressive Web Apps que la rompieron*. Obtenido de ORBE INNOVACIÓN: <https://www.orbesoftware.com.ar/top-3-casos-de-progressive-web-apps-que-la-rompieron.html>
- Bidelman, E. (5 de Octubre de 2021). *Twitter*. (Principal Ingeniero de Lift) Obtenido de <https://twitter.com/ebidel/status/1445425210119704578>
- Brunelle, D. (8 de mayo de 2018). *Tweter*. Obtenido de <https://twitter.com/davidbrunelle/status/993960071406080000>
- Cervantes, H. (s.f.). *Arquitectura de Software*. Obtenido de SG Buzz: <https://sg.com.mx/revista/27/arquitectura-software>
- Coates, P. (2010). *Programming Architecture*. Abingdon : Routledge.
- Cuello, J., & Vittone, J. (2013). *Diseñando apps para móviles*. Argentina: Catalina Duque Giraldo. Recuperado el 2022
- Foundation, M. (13 de Agosto de 2022). *MDN Web Docs*. Obtenido de <https://developer.mozilla.org/es/docs/Web/Manifest>
- Galdamez, N., Thomas, P., Pesado, P., & Lisandro, D. (2013). Un analisis experimental de tipo aplicaciones para dispositivos móviles. *WIS - X Workshop ingeniería de software / XVIII Congreso Argentino de Ciencias de la Computación* (pág. 776). La Plata Argentina: UNLP. Recuperado el 8 de Enero de 2022, de http://sedici.unlp.edu.ar/bitstream/handle/10915/32397/Documento_completo.pdf?sequence=1&isAllowed=y
- García de Zúñiga, F. (23 de 03 de 2018). *Service workers, servicios web más allá del navegador*. Obtenido de arsys: <https://www.arsys.es/blog/service-worker>
- Google, T. w. (septiembre de 2017). *Los próximos mil millones de usuarios: trivago adopta las aplicaciones web progresivas como el futuro de los dispositivos móviles*. Obtenido de Think with Google: <https://www.thinkwithgoogle.com/intl/en-gb/marketing-strategies/app-and-mobile/trivago-embrace-progressive-web-apps-as-the-future-of-mobile/>
- Hanmer, R. S. (2013). *Pattern-Oriented Software Architecture For Dummies*. Chichester: John Wiley & Sons, Ltd.

- Hernández, E. M., & Hernández, L. C. (2018). *Manual del comercio electrónico: técnicas, modelos, normativa y casos prácticos*. España: Editorial Marge Books.
- IONOS. (2017). <https://www.ionos.mx/>. Obtenido de <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/progressive-web-apps-pros-y-contras/>
- Juarez, I. (s.f.). *Resumen Fundamentos de Arquitectura de Software*. Obtenido de Platzi: <https://platzi.com/tutoriales/1247-arquitectura-software/9248-resumen-fundamentos-de-arquitectura-de-software/>
- Kotlin, C. (22 de Abril de 2021). *MVVM en Android con Kotlin, LiveData y View Binding – Android Architecture Components*. Obtenido de Curso Kotlin: <https://cursokotlin.com/mvvm-en-android-con-kotlin-livedata-y-view-binding-android-architecture-components/>
- Lanza Ortega, E. (2019). Trabajo final de grado. *Aplicación Web Progresiva (PWA) para la gestión de pagos de estacionamiento en superficie*. Universidad de Cantabria, Cantabria, Cantabria. Obtenido de <https://repositorio.unican.es/xmlui/bitstream/handle/10902/17089/420138.pdf?sequence=1&isAllowed=y>
- Leiva, A. (s.f.). *MVP para Android: Cómo organizar la capa de presentación*. Obtenido de Devexperto: <https://devexperto.com/mvp-android/>
- Martin, R. C. (2018). *Clean Architecture, A CRAFTSMAN'S GUIDE TO SOFTWARE STRUCTURE AND DESIGN*. Pearson Education, Inc.
- más, P. y. (s.f.). *Android: ¿Qué es MVC, MVP y MVVM?* Obtenido de Programación y más: [https://programacionymas.com/blog/android-mvc-mvp-mvvm#:~:text=Model%20View%20Presenter%20\(MVP\)%20y,enfoque%20es%20superior%20al%20otro.](https://programacionymas.com/blog/android-mvc-mvp-mvvm#:~:text=Model%20View%20Presenter%20(MVP)%20y,enfoque%20es%20superior%20al%20otro.)
- Mendoza, J. A. (10 de Octubre de 2019). *Milenium*. (C. ©. Templates, Productor) Recuperado el 18 de Julio de 2022, de <https://www.informaticamilenium.com.mx/es/temas/aplicacion-web-progresiva.html>
- Miguel, R. M. (2014). *Desarrollo de aplicaciones para Android*. Madrid : RA-MA Editoria.
- Mozilla. (13 de Agosto de 2022). <https://developer.mozilla.org/>. (©1998–2022 by individual mozilla.org) Recuperado el 22 de Agosto de 2022, de https://developer.mozilla.org/es/docs/Web/Progressive_web_apps
- Mozilla. (4 de Agosto de 2022). *mozilla.org*. (Mozilla, Ed.) Recuperado el 10 de Agosto de 2022, de <https://developer.mozilla.org/>: <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>
- Nudelman, G. (2013). *Android Design Patterns: Interaction Design Solutions for Developers*. Indianapolis: John Wiley & Sons, Inc.
- Oraá, P. (28 de marzo de 2019). *Pinterest llega a la Microsoft Store en forma de PWA*. Obtenido de <https://microsofters.com/153783/pinterest-microsoft-store-pwa/>
- Otero Carbón, N. (2020). Progressive Web App: Aplicación para la gestión diaria de entrenamientos para equipos de fútbol. *Trabajo Fin de Máster*. Universidad Internacional de La Rioja (UNIR), Leeds (Reino Unido). Recuperado el 2023, de <https://reunir.unir.net/handle/123456789/10692>
- Oussalah, M. C. (2014). *Software Architecture 1*. Great Britain: ISTE Ltd.
- Rachel, A., LePage, A. J., & Firtman, M. (2023). *Learn Progressive Web Apps*. Obtenido de web.dev: <https://web.dev/learn/pwa/>

- Ramírez Vique, R. (2013). *Métodos Para el Desarrollo de Aplicaciones Móviles*. (UOC, Ed.) Catalunya, España: Universidad Oberta de Catalunya. Recuperado el 24 de agosto de 2022, de http://190.57.147.202:90/jspui/bitstream/123456789/464/1/Tecnologia_y_desarrollo_en_dispositivos_moviles.pdf
- Raphael, J. (24 de junio de 2021). Las mejores aplicaciones web progresivas para la productividad. *Computerworld*. Recuperado el Agosto de 2022
- Rivera Moncayo, J. A. (2020). USO DEL E-HEALTH COMO ESTRATEGIA DE INNOVACIÓN ORIENTADA A LA ADMINISTRACIÓN Y ASIGNACIÓN DE TURNOS. *Ttrabajo de Titulación, Previo a la Obtención del Título de Ingeniero de Sistemas e Informática*. Universidad de las Fuerzas Armadas Innovacion para la Exelencia, Sangolqui, Ecuador. Recuperado el 24 de Agosto de 2022, de <http://repositorio.espe.edu.ec/bitstream/21000/21961/8/T-ESPE-043580.pdf>
- Rondán Martínez, D., Valderas Aranda, P. J., & Pastor López, Ó. (2010). *Aplicaciones Web: Un enfoque práctico* (Primera Edición ed.). (Alfaomega, Ed.) Madrid España: RA-MA. Recuperado el Agosto de 2022
- Sacristán, C. R. (2013). *Programación en Android*. Madrid: Ministerio de Educación y Formación Profesional de España.
- Samuel. (9 de Abril de 2019). *MVVM y DataBinding: Patrones de diseño de Android*. Obtenido de Stips: <https://stips.wordpress.com/2019/04/09/mvvm-y-databinding-patrones-de-diseno-de-android/>
- Semegn, A. D. (2011). *Software Architecture and Design for Reliability Predictability*. Newcastle: Cambridge Scholars Publishing.
- Sunil Tandel, S., & Jamadar, A. (9 de Septiembre de 2018). Impact of Progressive Web Apps on Web App Development. *IJIRSET*, 7, 9442. doi:10.15680/IJIRSET.2018.0709021
- Tapia, J. L. (2020). *Thesis*. Universitat Politècnica de Catalunya, Catalunya, Barcelona.
- Tapia, J. L. (2020). Desarrollo E Implementacion De Una Aplicación Web Progresiva (PWA). *A Degree Thesis Submitted to the Faculty of the Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona*. Universitat Politècnica de Catalunya, Barcelona, Catalunya. Obtenido de https://upcommons.upc.edu/bitstream/handle/2117/192433/Final_Degree_thesis.docx.pdf
- Tapia, J. L. (2020). *DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PROGRESIVA (PWA)*. Universitat Politècnica de Catalunya, Barcelona. Recuperado el 10 de Julio de 2022
- Tapia, J. L. (2020). Tesis. *DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PROGRESIVA (PWA)*. Universitat Politècnica de Catalunya, Catalunya, Barcelona.
- Tapia, J. L. (s.f.). *Desarroyo e Implementacion de una Aplicacion Prgresiva (PWA)*. Universidad Politecnica de Catalunya .
- Thomas, P., Lisandro, D., Corbalan, L., & Cáseres, G. (2018). Tendencias en el desarrollo de Aplicaciones para Dispositivos Móviles. *XX Workshop de Investigadores en Ciencias de la Computación*, 588-589.
- Vázquez, Á. V., & Valero, J. A. (2019). *Android: del diseño de la arquitectura al despliegue profesional*. México: Alfaomega Grupo Editor, S.A. de C.V.
- Vespa, L. (23 de Mayo de 2019). *Patrones Arquitectónicos en Android*. Obtenido de Medium: <https://medium.com/@vespasoft/patrones-arquitect%C3%B3nicos-en-android-ded39f7a2c10>

Vidal, M. (5 de Noviembre de 2019). *iebschool*. (©2022 IEBS Business School. Todos los derechos reservados.) Recuperado el 24 de Agosto de 2022, de *iebschool*:
<https://www.iebschool.com/blog/progressive-web-apps-analitica-usabilidad/#pwa>
web.dev. (s.f.). *Learn PWA!* Obtenido de *Web.dev*: <https://web.dev/learn/pwa/service-workers/>
Yaguargos Castro, S. D. (2020). APLICACIÓN WEB PROGRESIVA (PWA) PARA LA AUTOMATIZACIÓN DE LOS PROCESOS DE GESTIÓN E INFORMACIÓN EN LIGA DEPORTIVA PARROQUIAL TOTORAS". *Trabajo de Titulación*. UNIVERSIDAD TÉCNICA DE AMBATO, Ambato, Ambato, Ecuador. Recuperado el 2023, de
<https://repositorio.uta.edu.ec/bitstream/123456789/31308/1/t1712si.pdf>