

# Aplicaciones Web Progresivas Enfocadas en el Uso y Optimización de Cache

Rocío A. Rodríguez, Pablo M. Vera, M. Roxana Martínez, Claudia G. Alderete, Mariano G. Dogliotti

CAETI - Centro de Altos Estudios en Tecnología Informática  
Universidad Abierta Interamericana (UAI)  
Montes de Oca 745, Ciudad Autónoma de Buenos Aires, Argentina

{rocioandrea.rodriguez, pablomartin.vera, roxana.martinez, claudia.alderete,  
mariano.dogliotti}@uai.edu.ar

## RESUMEN

La gran inserción de dispositivos móviles pone en foco el análisis de las soluciones diseñadas para los mismos. Los usuarios móviles descargan gran cantidad de aplicaciones y de hecho la tendencia indica que prefieren las aplicaciones móviles por sobre los sitios web móviles, este efecto se irá reduciendo con el desarrollo de PWA (Aplicaciones Web Progresivas). Las PWA permiten tener un ícono de acceso, poder trabajar offline utilizando información almacenada internamente en el dispositivo, pueden ofrecer notificaciones; una PWA bien desarrollada dará el mismo aspecto que una aplicación móvil instalada en el dispositivo. La ventaja de las PWAs es que si se optimizan bien los esquemas de cache, permitirá disminuir el consumo de datos. Sobre este tema de optimización y esquemas de uso de cache se centra el presente proyecto.

**Palabras clave:** Web Móvil, Dispositivos Móviles, PWA, Cache

## CONTEXTO

Esta línea de I+D forma parte de los proyectos radicados en el Centro de Altos Estudios en Tecnología Informática (CAETI) de la Universidad Abierta Interamericana (UAI). En este proyecto participan docentes y alumnos tanto de sede Centro como de la Castelar (ambas en la provincia de Buenos Aires). El proyecto cuenta con financiamiento asignado y una duración de 2 años.

## 1. INTRODUCCIÓN

La tecnología móvil ocupa un papel primordial en la vida de las personas hasta el punto por el cual en la actualidad el teléfono móvil combinado con internet es el principal medio por el cual se accede a la información. Una gran parte de los usuarios están acostumbrados a instalar y utilizar distintas aplicaciones en sus dispositivos aun cuando la versión web esté adaptada para ser utilizada correctamente en la pantalla reducida. Según el reporte global que analiza el estado de las tecnologías móviles creado por la consultora Comscore, "más del 80% de los minutos de uso en los dispositivos móviles, tomando en cuenta todos los mercados, son utilizados en apps". Particularmente en Argentina según este mismo informe el porcentaje de tiempo de uso de apps asciende al 94% [1].

Es esta preferencia de los usuarios que obliga a los desarrolladores a construir aplicaciones nativas, teniendo en su mayoría como foco los dos sistemas operativos más populares actualmente que son Android y IOS. Tener dos versiones de una misma aplicación hace que se multiplique el esfuerzo de desarrollo y más aún el de mantenimiento y distribución de nuevas versiones y mejoras. Una solución son los entornos de desarrollo híbrido que permiten programar en un único lenguaje y generar apps para ambos sistemas operativos. Pero este enfoque tiene el inconveniente de que la parte web queda por otro lado por lo que aún hay un doble trabajo. Para solucionar

este problema nacen las aplicaciones web progresivas.

Para acercarse más a las aplicaciones web de las nativas “un nuevo concepto ha surgido en los últimos años denominado Aplicaciones Web Progresivas (PWA por sus siglas en inglés). Una PWA es una aplicación web que utiliza las últimas tecnologías disponibles en los navegadores para ofrecer en dispositivos móviles una experiencia lo más parecida posible a la de una aplicación nativa. Los objetivos que persiguen las PWA son: lograr el mayor rendimiento posible en dispositivos móviles, que la aplicación cargue de manera casi instantánea, que la interfaz de usuario se parezca lo máximo posible a una nativa, que se pueda trabajar sin conexión (offline first) y que se puedan enviar notificaciones a los usuarios, como en una aplicación nativa” [2].

“Las aplicaciones web progresivas son una evolución natural de las aplicaciones web que difuminan la barrera entre la web y las aplicaciones, pudiendo realizar tareas que generalmente solo las aplicaciones nativas podían llevar a cabo. Algunos ejemplos son las notificaciones, el funcionamiento sin conexión a Internet o la posibilidad de probar una versión más ligera antes de bajarte una aplicación nativa de verdad” [3].

Resulta importante tener un enfoque integral que conjugue aspectos de hardware y software. “Una de las características importantes de la gran mayoría de los desarrollos móviles es su corta duración. Esto se debe a factores como la gran competencia en el sector, los cambios en el mismo con la aparición, casi constante, de novedades tanto software como hardware” [4].

El poder almacenar en cache el contenido web reduce considerablemente el consumo de datos, haciendo la aplicación más rápida y siempre disponible para el usuario final. Nicolas Gallagher un desarrollador de la red social Twitter en su artículo «How we built Twitter Lite» [5], indica que gracias a la

versión PWA consiguieron reducir el consumo de datos hasta fracciones irrisorias en comparación con sus aplicaciones nativas.

Las PWA permiten realizar una aplicación 100% desconectada, pero si los datos cambian dicho cache debe ser actualizado. Por lo que es necesario estudiar los distintos enfoques para el manejo de cache estableciendo la mejor estrategia para cada tipo de aplicación.

Los enfoques posibles son:

- Solo cache: Utilizado para páginas estáticas que no cambian y siempre serán levantadas de la cache
- Solo red: Por el contrario del enfoque anterior siempre se irá a buscar el contenido a la red, muy útil para contenidos dinámicos que siempre cambian.
- Cache y si falla red: Esquema combinado en el que se prioriza la cache y en caso de no encontrar allí el contenido se recurre a la red. Al recuperarlo adicionalmente puede ser cargado en cache para futuros accesos.
- Red y si falla cache: Esquema combinado en el que se prioriza la red y si el usuario no tuviese conectividad el contenido a mostrar es el de cache.
- Carrera entre cache y red: Con carrera se refiere a que se consulta en cache y red en simultáneo y lo que primero se consiga es lo que se muestra. Por ejemplo, si se cuenta con equipos de acceso lento al almacenamiento, pero con una excelente conexión a la red para recursos pequeños puede ser más rápida la red que la cache.
- Cache y después red: Se muestra al usuario rápidamente lo de cache y mientras se busca en la red si hay actualización del contenido.
- Contenido de reserva: Si no está disponible en cache y no hay acceso a la red, tener contenido de reserva para

mostrarle al usuario (página de sin conexión ó indicación por defecto).

Los componentes principales de una PWA serán un archivo de manifiesto, service worker, almacenamiento local y notificaciones. De hecho es el service worker el que se ocupará de trabajar con el manejo de esquemas de cache o del acceso a la red. A modo de ejemplo se presenta la figura 1 (tomada de [6]), puede observarse que se recurre al service worker el cual accede a cache y si el contenido no está disponible accede a la red para buscarlo y poder servirlo. Este esquema prioriza la cache, con la posibilidad de actualizar la cache en los casos que se requiera acceder a la red, lo cual está identificado con la línea punteada (que se le ha agregado a la figura del autor).

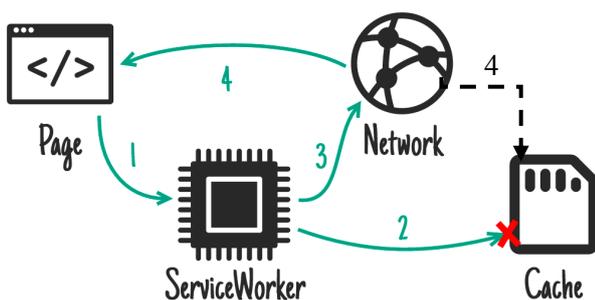


Figura 1. Esquema de acceso con prioridad a cache y luego red.

PWA hace que el usuario no distinga de si se trata de una aplicación nativa o una web, incluso pudiendo tener acceso al hardware disponible como GPS, vibración entre otros, lo que brinda una experiencia de uso más completa. “La brecha entre las capacidades de las aplicaciones nativas sobre las aplicaciones web se está reduciendo, especialmente impulsada por los estándares web de acceso al hardware” [7].

Actualmente existen diversos trabajos en el área de PWA entre ellos [8], [9] y [10] se enfocan en el manejo y optimización de cache. Otros se basan en distintas tecnologías y frameworks como [11], [12] y [13]. Por otra

parte [14] y [15] realizan una comparativa entre las PWA y las aplicaciones web tradicionales.

Estos trabajos son solo algunos ejemplos y muestra de que el estudio de las PWA está en pleno auge, por lo tanto resulta de suma importancia su estudio y comparativa con otras tecnologías existentes.

## 2. LÍNEAS DE INVESTIGACIÓN y DESARROLLO

Los ejes principales del trabajo son:

- Analizar las ventajas y desventajas de las PWA en comparación con las APP híbridas
- Analizar la evolución de los estándares para el manejo de hardware e información almacenada en el dispositivo mediante una PWA
- Profundizar en esquemas de Cache para manejar el versionado
- Comparar los distintos de entornos de desarrollo en contrapartida al desarrollo de PWA determinando la mejor opción según el tipo de APP a desarrollar.

## 3. RESULTADOS OBTENIDOS/ESPERADOS

Dado que este proyecto ha comenzado en el presente año aún no hay resultados obtenidos. Pero los resultados a obtenerse serán producto de analizar y probar los distintos esquemas de cache disponibles proponiendo la mejor forma de administrar la misma según el tipo de aplicación que se quiera realizar. No se aplicará el mismo esquema para un sitio estático que para un uno dinámico, pero a su vez, si el sitio tiene páginas estáticas se busca analizar qué pasa cuando se quiere incorporar una nueva versión por alguna modificación. El service worker al cambiar la versión hace

que las páginas estáticas se vuelvan a descargar lo que puede incurrir en gasto de tiempos y datos por un cambio menor. Uno de los objetivos es generar un esquema de manejo de cache que permita actualizar solo lo necesario para que el usuario no descargue datos que ya tiene. Esta optimización se realizará directamente desde el service worker trabajando con los eventos y métodos disponibles. Es importante que el usuario tenga disponible siempre la aplicación y que el desarrollador pueda actualizarla con un mínimo costo para el usuario cuando sea necesario.

En cuanto a los entornos de desarrollo híbridos se planea analizar los más difundidos actualmente y categorizarlos según sus características por ejemplo en los que usan web View, los que usan los controles nativos y los que renderizan sus propios controles.

#### 4. FORMACIÓN DE RECURSOS HUMANOS

El equipo está formado por 5 docentes: 2 de ellos doctores en Ciencias Informática graduados en la Universidad Nacional de La Plata (UNLP), 1 con maestría finalizada (UAI) con la cursada del doctorado finalizada y que se encuentra actualmente haciendo su tesis doctoral (UNLP) y 1 realizando actualmente su maestría en UAI. Esto implica que 4 de los 5 docentes que componen el grupo tienen estudios de posgrados finalizados o en progreso.

Este proyecto también cuenta con la participación de alumnos de grado y posgrado de la UAI (actualmente en el proyecto se encuentran vinculados 4 alumnos).

En el área de dispositivos móviles se encuentran en realización 4 tesis de maestría (2 en la UAI y 2 en UNLaM – Universidad Nacional de La Matanza), siendo directores de dichas tesis miembros del equipo de investigación.

#### 5. BIBLIOGRAFIA

- [1] Comscore, 2019 Report Global State of Mobile  
<https://www.comscore.com/Insights/Presentations-and-Whitepapers/2019/Global-State-of-Mobile>
- [2] Thomas Pablo, Delia Lisandro, Corbalán Leonardo, et al. “Tendencias en el desarrollo de Aplicaciones para Dispositivos Móviles”. Instituto de Investigación en Informática LIDI (III-LIDI) Facultad de Informática – Universidad Nacional de La Plata. Centro Asociado a la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC). Abril 2018.  
[https://digital.cic.gba.gov.ar/bitstream/handle/11746/8316/11746\\_8316.pdf?sequence=1&isAllowed=y](https://digital.cic.gba.gov.ar/bitstream/handle/11746/8316/11746_8316.pdf?sequence=1&isAllowed=y)
- [3] Ramírez Ivan (2018), ¿Qué es una aplicación web progresiva o PWA?.  
<https://www.xataka.com/basics/que-es-una-aplicacion-web-progresiva-o-pwa>
- [4] Vique, R. R. (2012). Métodos para el desarrollo de aplicaciones móviles.
- [5] N. Gallagher, “How We Built Twitter Lite,” Apr. 2017.  
[https://blog.twitter.com/engineering/en\\_us/topics/open-source/2017/how-we-built-twitter-lite.html](https://blog.twitter.com/engineering/en_us/topics/open-source/2017/how-we-built-twitter-lite.html)
- [6] Archibald Jake (2019), La guía de soluciones sin conexión.  
<https://developers.google.com/web/fundamentals/instant-and-offline/offline-cookbook/?hl=es>
- [7] Rodríguez Rocío A, Vera Pablo M, Martínez M. Roxana, Pons Claudia, Valles Federico E, de La Cruz Luis Verbel. “Reducing the Gap between Native and Web Applications”. SETESEC – Italia2014
- [8] Rojas, C. (2020). Caching Strategies. In Building Progressive Web Applications with Vue. js (pp. 67-81). Apress, Berkeley, CA.
- [9] Hajian, M. (2019). Advanced Angular Service Worker and Runtime Caching.

In Progressive Web Apps with Angular (pp. 107-139). Apress, Berkeley, CA.

- [10] Kim, H. G., Park, J. T., Choi, M. H., & Moon, I. Y. (2019). Design of Web Content Update Algorithm to Reduce Communication Data Consumption using Service Worker and Hash. *The Journal of Advanced Navigation Technology*, 23(2), 158-165.
- [11] Satyal, A. (2020). *Designing and Developing a Website with ReactJS: Progressive Web Application*.
- [12] Hajian, M. (2019). App Shell and Angular Performance. In *Progressive Web Apps with Angular* (pp. 169-199). Apress, Berkeley, CA.
- [13] Khan, A. I., Al-Badi, A., & Al-Kindi, M. (2019). Progressive Web Application Assessment Using AHP. *Procedia Computer Science*, 155, 289-294.
- [14] Said Tahirshah, F. (2019). Comparison between Progressive Web App and Regular Web App.
- [15] Kvist, J., & Mathiasson, P. (2019). *Progressive Web Apps and other mobile developing techniques: a comparison*.