

Mejorando la performance en Aplicaciones Web Progresivas mediante estrategias de utilización de la Cache

Rocío A. Rodríguez, Pablo M. Vera, M. Roxana Martínez, Franco Cifuentes, 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, pabломartin.vera, roxana.martinez, claudia.alderete, mariano.dogliotti}@uai.edu.ar

francoignacio.cifuentes@alumnos.uai.edu.ar

RESUMEN

Las aplicaciones web progresivas (PWA) han permitido trasladar las ventajas de las aplicaciones nativas (funcionamiento desconectado, acceso a hardware, uso de almacenamiento local, notificaciones, icono de acceso, etc...) a las aplicaciones web. Se cuenta entonces con aplicaciones que como toda solución web destacan por su portabilidad, pero con los beneficios y forma de uso de las aplicaciones nativas. En esta línea de investigación, el foco está puesto en el uso de almacenamiento interno en el dispositivo (cache) para generar estrategias que permitan optimizar esa cache disminuyendo la necesidad de consumir datos de la red. Para lo cual se diseñó una estrategia que une varios enfoques de cache existentes para generar una solución integral. Lo que permitirá comparar la performance de las soluciones existentes que utilizan una solución tradicional de manejo de cache, con la propuesta optimizada del equipo de investigación.

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 rápida expansión y adopción de los teléfonos móviles han generado cambios sociales y culturales en la sociedad, han modificado las formas de comunicación, de acceder a la información y las maneras en que los individuos se relacionan entre sí. La telefonía móvil ha producido una transformación en los ritos sociales de interacción. El uso de las TIC ha modificado la forma de trabajar, aprender, colaborar, jugar, pasar el tiempo y socializar de los individuos” [1]. Es por esto que resulta necesario que la web también de respuesta a nuevas necesidades, primeramente, a través del diseño adaptativo [2] en donde la misma solución podía ser visualizada sin dificultades en diversos dispositivos y actualmente derivando en las aplicaciones web progresivas (PWA).

Las PWA presentan diversas ventajas, por supuesto la portabilidad por ser aplicaciones web pero además incorporan las características de las aplicaciones nativas, entre lo que se destaca poder trabajar en forma desconectada y accediendo a la cache del dispositivo.

“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].

Para lograr esto debe modificarse la cache cuando los datos se modifican y no se cuenta con conexión al servidor para actualizar, también es necesario poder detectar sino existieron cambios y en ese caso usar la cache sin recurrir al servidor... es decir en cada situación habrá una estrategia para el acceso y recuperación de datos. Actualmente existen enfoques para el acceso a los datos: (1) Solo cache (2) Solo red (3) Cache y si falla red (4) Red y si falla cache (5) Carrera entre cache y red (6) Cache y después red (7) Contenido de reserva

Para una solución que es absolutamente estática se podría usar el esquema “sólo cache”. Para una solución que siempre es dinámica se podría usar “solo red”. Para una solución integral, empieza a ser importante establecer estrategias de uso de la cache en las distintas situaciones que pueden ocurrir. Por otra parte utilizar la cache del dispositivo en vez de descargar todo desde la red, permitirá disminuir el tráfico y consumo de datos. Nicolas Gallagher un desarrollador de la red social Twitter en su artículo “How we built Twitter Lite” [4], indica que gracias a la versión PWA consiguieron reducir el consumo de datos hasta fracciones irrisorias en comparación con sus aplicaciones nativas.

Para que una aplicación pueda tener las características previamente mencionadas es preciso tomar en cuenta diversas pautas al momento de su desarrollo. Resulta interesante un checklist provisto por google [5], en donde se destacan 8 pautas:

- Hosteado en https

- Usa diseño adaptativo y se visualiza correctamente en mobile y tablets
- Todas las páginas deben funcionar cuando se no se tiene conexión
- Debe tener metadatos para dar la opción “Agregar a la pantalla de Inicio” lo que permite “instalar” la aplicación.
- Debe tener un inicio rápido aún en redes lentas (<10 seg en redes 3g)
- Crossbrowsing (visualizarse correctamente independientemente del navegador).
- Los cambios de página deben ser rápidos
- Cada página debe tener su url y si es una app de una solo página con distintas vistas se debe poder reconstruir para permitir acceder directamente a cada vista.

Los componentes principales de una PWA son:

1. Archivo de Manifiesto
2. Service Worker
3. Almacenamiento Local
4. Notificaciones

En la figura 1, se presenta una captura de pantalla en donde puede observarse que, al ingresar por primera vez a la web, en la parte inferior la pantalla aparece un mensaje con el ícono de la aplicación en donde se ofrece agregarla a la pantalla principal.

Al instalar la aplicación se puede acceder desde un ícono a la misma y ya se no se muestra la barra del navegador (se ha elegido ocultarla) y en el caso de la pantalla de la derecha de la figura 2 se ha integrado la barra de título con la de estado con los íconos clásicos del sistema operativo. La forma final que se visualizará será configurada por los desarrolladores según un archivo de manifiesto, donde entre otras cosas se puede seleccionar: modo de visualización, orientación de la pantalla, íconos, color de fondo, color de tema, idioma, nombre corto para la aplicación...



Figura 1. Ejemplo Instalación de una PWA

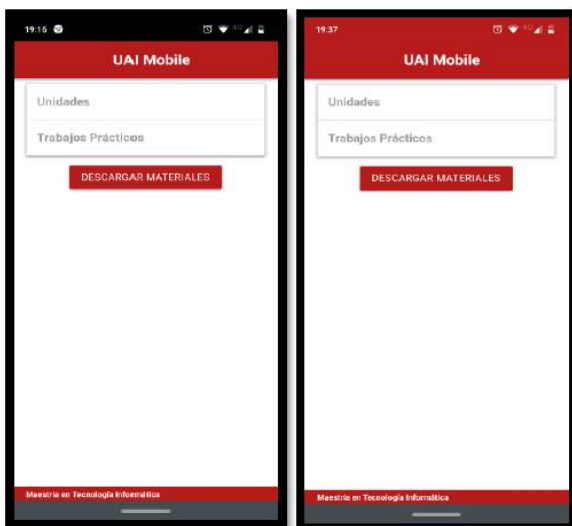


Figura 2. Ejemplo de Visualización de una PWA

Existen diversos artículos académicos que implementan PWA en una amplia variedad de temáticas, en menor medida hay trabajos en el área que se enfocan en el manejo y optimización de cache [6], [7], [8], [9] y [10].

El manejo de cache de forma optimizada permite mejorar la performance de las PWA, disminuyendo el consumo de datos de la red. En base a esta premisa se establecen las líneas de investigación y desarrollo para este año del proyecto de investigación.

2. LÍNEAS DE INVESTIGACIÓN y DESARROLLO

Los ejes principales del trabajo son:

- Analizar los esquemas de cache utilizados por las PWA actuales existentes en el mercado.
- Evidenciar mejora de performance optimizando el uso de la cache en las PWA existentes.
- Disponibilizar APIs que permitan implementar la estrategia de manejo de cache propuesta por el equipo de investigación

3. RESULTADOS OBTENIDOS/ESPERADOS

En el primer año del proyecto de investigación se realizó una propuesta de estrategia de manejo de cache que combina los esquemas existentes, dicha propuesta es desarrollada e implementada para la cual se definieron pruebas de funcionamiento.

Las pruebas planteadas fueron: (1) Activación de service worker y descarga de archivos, (2) Aumentar versión de caché en 1 y agregar nuevo archivo modificado (3) Aumentar versión de caché en 1 y no modificar arreglo de paginasModificadas (4) Aumentar versión de caché en 2 para simular salteo de versiones y agregar nuevo archivo modificado (5) Aumentar versión de caché en 2 para simular salteo de versiones y no agregar archivo al arreglo de paginasModificadas (6) Traer listado de clientes por primera vez (7) Traer listado de clientes con fecha de caché menor a fecha de actualizacion de la base de datos (8) Traer listado de clientes con fecha de caché

mayor a fecha de actualización de la base de datos (9) Traer listado de clientes sin servicio de internet.

Todas estas pruebas junto a sus resultados esperados fue lo que se validó en cuanto al funcionamiento de la estrategia de cache propuesta. En esta segunda etapa se comparará soluciones de terceros y su impacto al aplicar esta metodología integral de cache para evaluar la performance.

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 3 tesis de maestría (2 en la UAI y 1 en UNLaM – Universidad Nacional de La Matanza) y 1 tesina de grado (UAI), siendo directores de dichas tesis miembros del equipo de investigación.

5. BIBLIOGRAFIA

[1] Castro Rojas, S. "Ubicuidad y comunicación: los Smartphones." Chasqui: Revista Latinoamericana de Comunicación 118 (2012): 91-95.

[2] Rodríguez, R. A., Vera, P. M., Ramirez, M. A., Alderete, C. G., Conca, A. G., Dogliotti, M. G., & Zain, G. A. (2020). Análisis del Diseño Web Adaptativo Caso de estudio: Universidad Argentinas. Revista Abierta de Informática Aplicada (RAIA), (4), 51-62.

[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] 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

[5] Google Developers. "Progressive Web Apps" <https://developers.google.com/web/progressive-web-apps/>

[6] Rojas, C. (2020). Caching Strategies. In Building Progressive Web Applications with Vue.js (pp. 67-81). Apress, Berkeley, CA.

[7] Hajian, M. (2019). Advanced Angular Service Worker and Runtime Caching. In Progressive Web Apps with Angular (pp. 107-139). Apress, Berkeley, CA.

[8] 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.

[9] Gambhir, A., & Raj, G. (2018). Analysis of Cache in Service Worker and Performance Scoring of Progressive Web Application. 2018 International Conference on Communications (COMM), Communications (COMM), 2018 International Conference On, 01–06. <https://doi.org/10.1109/ICComm.2018.8430158>

- [10] Malavolta, I., Chinnappan, K., Jasmontas, L., Gupta, S., & Soltany, K. A. K. (2020, July). Evaluating the impact of caching on the energy consumption and performance of progressive web apps. In *Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems* (pp. 109-119).