

Un Análisis Experimental de Tipo de Aplicaciones para Dispositivos Móviles

Lisandro Delía¹, Nicolás Galdamez¹, Pablo Thomas¹, Patricia Pesado¹

¹ Instituto de Investigación en Informática LIDI. Facultad de Informática.
Universidad Nacional de La Plata. Argentina
{ldelia, ngaldamez, pthomas, ppesado}@lidi.info.unlp.edu.ar

Resumen. El auge de los dispositivos móviles ha generado nuevos desafíos para los ingenieros de software. Las capacidades técnicas ofrecidas, así como sus restricciones, plantean un escenario fértil, pero complejo. Existen diferentes alternativas de desarrollo de una misma aplicación para un dispositivo móvil. En este trabajo se presentan los enfoques de desarrollo de software existentes, sus características más destacadas, y un caso experimental que permite analizar las ventajas y dificultades de cada enfoque.

Palabras claves: dispositivos móviles, aplicaciones móviles nativas, aplicaciones móviles híbridas, aplicaciones móviles web.

1 Introducción

Los dispositivos móviles forman parte de la vida cotidiana y son cada vez más sofisticados, su poder de cómputo genera posibilidades hasta hace años no pensadas.

La creciente demanda de software específico para estos dispositivos ha generado nuevos desafíos para los desarrolladores, ya que este tipo de aplicaciones tiene sus características propias, restricciones y necesidades únicas, lo que difiere del desarrollo de software tradicional.

La computación móvil se puede definir como un entorno de cómputo con movilidad física. El usuario de un entorno de computación móvil será capaz de acceder a datos, información u otros objetos lógicos desde cualquier dispositivo en cualquier red mientras está en movimiento [1].

Las particularidades de este entorno incluyen: alto nivel de competitividad, tiempo de entrega necesariamente corto y la dificultad adicional de identificar los stakeholders y sus requerimientos.

Las aplicaciones se generan en un entorno dinámico e incierto. Generalmente, son pequeñas, no críticas, aunque no menos importantes. Están destinadas a un gran número de usuarios finales y son liberadas en versiones rápidas para poder satisfacer las demandas del mercado [2].

El desarrollo de aplicaciones móviles es, actualmente, un gran desafío, dado las demandas específicas y las restricciones técnicas de un entorno móvil [3], tales como dispositivos con capacidades limitadas, pero en evolución continua; varios estándares, protocolos y tecnologías de red, necesidad de operar sobre diferentes plataformas,

requerimientos específicos de los usuarios y las exigencias estrictas en tiempo del mercado.

Estos dispositivos tienen características físicas distintivas, entre las cuales se destacan su tamaño, peso, tamaño de pantalla, su mecanismo de ingreso de datos y su capacidad de expansión. Además, los aspectos técnicos, incluyendo el poder de procesamiento, espacio de memoria, autonomía de batería, sistema operativo, entre otros, tienen un rol esencial. Todas estas características deben ser cuidadosamente consideradas en el desarrollo de aplicaciones [4].

En la corta historia del desarrollo de software las plataformas de hardware y software han evolucionado en forma constante, pero nunca antes fue tan masivo el poder de cómputo que tienen las personas en sus manos, puntualmente a través del uso de dispositivos móviles. Esto conduce a nuevos desafíos y junto a ellos al crecimiento de la Ingeniería de Software como disciplina, acompañando esta evolución.

En este trabajo se presenta un estudio comparativo de tipos de aplicaciones para dispositivos móviles, a partir de un caso experimental desarrollado para la plataforma educativa WebUNLP [6]. En la sección 2 se detallan las características más salientes de los diferentes tipos de aplicaciones para dispositivos móviles. Posteriormente se presenta el proceso de desarrollo de los diferentes tipos de aplicaciones para el caso experimental utilizado. Finalmente, se expresan conclusiones y trabajos futuros.

2 Tipo de aplicaciones para dispositivos móviles

En los últimos años el mercado de los dispositivos móviles, en especial smartphones, ha mostrado un crecimiento notable tanto en Argentina como en todo el mundo. En particular, en nuestro país, las plataformas que más han crecido son Android e iOS [8] [9].

Cada una de estas plataformas cuenta con una infraestructura de desarrollo particular.

El principal reto para los proveedores de aplicaciones es proporcionar soluciones para todas las plataformas, pero tiene un alto costo [11].

La solución ideal a este problema es crear y mantener una única aplicación para todas las plataformas. El desarrollo multiplataforma tiene como objetivo mantener la misma base de código para diversas plataformas. De esta forma el esfuerzo y costo de desarrollo se reduce notablemente.

A continuación se presentan tres enfoques para desarrollo de aplicaciones para dispositivos móviles: un enfoque nativo y dos enfoques multiplataforma (web e híbrido).

2.1 Aplicaciones Web

Las aplicaciones web para móviles son diseñadas para ser ejecutadas en el navegador del dispositivo móvil. Estas aplicaciones son desarrolladas utilizando HTML, CSS y JavaScript, es decir, la misma tecnología que la utilizada para crear sitios web.

Una de las ventajas de este enfoque es que los dispositivos no necesitan la instalación de ningún componente en particular, ni la aprobación de algún fabricante para que las aplicaciones sean publicadas y utilizadas. Solo se requiere acceso a internet. Además, las actualizaciones de la aplicación son visualizadas directamente en el dispositivo, ya que los cambios son aplicados sobre el servidor y están disponibles de inmediato. En resumen, es rápido y fácil de poner en marcha.

La principal ventaja de este tipo de aplicación es su independencia de la plataforma. No necesita adecuarse a ningún entorno operativo. Solo es necesario un navegador.

Por contrapartida, esto disminuye la velocidad de ejecución y podrían llegar a ser menos atractivas que las aplicaciones nativas. Además, podrían tener baja performance por problemas de conectividad. Finalmente este tipo de aplicaciones no pueden utilizar todos los elementos de hardware del dispositivo, como por ejemplo, cámara, GPS, entre otros.

2.2 Aplicaciones Nativas

Las aplicaciones nativas son aquellas que se conciben para ejecutarse en una plataforma específica, es decir, se debe considerar el tipo de dispositivo, el sistema operativo a utilizar y su versión.

El código fuente se compila para obtener código ejecutable, proceso similar que el utilizado para las tradicionales aplicaciones de escritorio.

Cuando la aplicación está lista para ser distribuida debe ser transferida a las App stores (tiendas de aplicaciones) específicas de cada sistema operativo. Estas tienen un proceso de auditoría para evaluar si la aplicación se adecúa a los requerimientos de la plataforma a operar. Cumplido este paso, la aplicación se pone a disposición de los usuarios.

La principal ventaja de este tipo de aplicaciones es la posibilidad de interactuar con todas las capacidades del dispositivo (cámara, GPS, acelerómetro, agenda, entre otras). Además no es estrictamente necesario poseer acceso a internet. Su ejecución es rápida, puede ejecutarse en modo background y notificar al usuario cuando ocurra un evento que necesite su atención.

Claramente estas ventajas se pagan con un mayor costo de desarrollo, pues se debe utilizar un lenguaje de programación diferente según la plataforma. Por ende, si se desea cubrir varias plataformas, se deberá generar una aplicación para cada una de ellas. Esto conlleva a mayores costos de actualización y distribución de nuevas versiones.

2.3 Aplicaciones Híbridas

Las aplicaciones híbridas combinan lo mejor de los dos tipos de aplicaciones anteriores. Se utilizan tecnologías multiplataforma como HTML, Javascript y CSS, pero se puede acceder a buena parte de las capacidades específicas de los dispositivos.

En resumen, son desarrolladas utilizando tecnología web y son ejecutadas dentro de un contenedor web sobre el dispositivo móvil.

Entre las principales ventajas de esta metodología se pueden mencionar la posibilidad de distribución de la aplicación a través de las tiendas de aplicaciones, la reutilización de código para múltiples plataformas y la posibilidad de utilizar las características de hardware del dispositivo.

Una de las desventajas es que, al utilizar la misma interfaz para todas las plataformas, la apariencia de la aplicación no será como la de una aplicación nativa. Finalmente la ejecución será más lenta que la ejecución en una aplicación nativa.

3 Caso experimental: WebUNLP

3.1 Descripción del problema

WebUNLP es un entorno virtual de enseñanza y aprendizaje que permite a los docentes mediar sus propuestas educativas. Alumnos y docentes pueden encontrarse en ese espacio para compartir materiales de estudio, comunicarse y generar una experiencia educativa en forma virtual [6].

Actualmente, WebUNLP cuenta con una versión web enfocada a computadoras de escritorio o portátiles, pero no está adaptada para ser utilizada desde dispositivos móviles.

El desarrollo planteado en este trabajo consiste en extender WebUNLP, con la construcción de una aplicación móvil que permita acceder a determinadas funcionalidades del sistema a través de un dispositivo móvil. El enfoque propuesto incluye un análisis de la misma solución, comparando el desarrollo nativo, web e híbrido, a fin de establecer cuál de ellos es conveniente.

Como ocurre con cualquier desarrollo de software, la construcción de una aplicación móvil implica tener claramente definido su propósito y cuáles requerimientos debe cumplir. En particular, para el caso de software para dispositivos móviles resulta esencial tener objetivos más específicos que en su versión de escritorio [7].

Para el caso específico de WebUNLP se realizó un desarrollo incremental de una aplicación móvil, y esta primera versión se enfocó en una de sus herramientas de comunicación: la cartelera. Esta herramienta permite comunicar las novedades de un curso, como, por ejemplo, el cambio de horario de una cursada, recordar las fechas de entrega de un trabajo práctico, entre otras [6].

3.2 Análisis

Uno de los primeros interrogantes planteados fue la elección de la plataforma. En términos de mercado los sistemas operativos que predominan en Argentina son Android e iOS [8][9], con lo cual se decidió dar soporte a estos dos sistemas operativos.

Se analizaron los requerimientos funcionales y no funcionales de forma aislada a la plataforma y luego de forma específica para cada una de ellas.

A continuación, se presentan algunos requerimientos a cumplir por la aplicación:

- El usuario debe poder ingresar a la aplicación con las mismas credenciales que las utilizadas para acceder a la versión web.
- El usuario debe poder acceder a la cartelera de todos los cursos en los que participa, ya sea como docente o alumno.
- El usuario debe recibir una notificación en su dispositivo cuando una novedad es publicada en la cartelera. Este requerimiento no se puede cumplir en la versión web accesible desde computadoras de escritorio y/o portables.
- El usuario debe tener la misma experiencia de uso en todas las plataformas operativas
- La aplicación web existente debe estar sincronizada con la aplicación móvil a desarrollar, esto significa que cualquier cambio realizado desde la aplicación móvil debe verse reflejado en la versión web y viceversa.

3.3 Diseño

Para satisfacer los requerimientos planteados en el punto anterior, a excepción de la notificación, el diseño de la aplicación móvil web consiste en una réplica de lo ofrecido por WEBUNLP, adaptándose solamente la interfaz al tamaño de pantalla del dispositivo móvil.

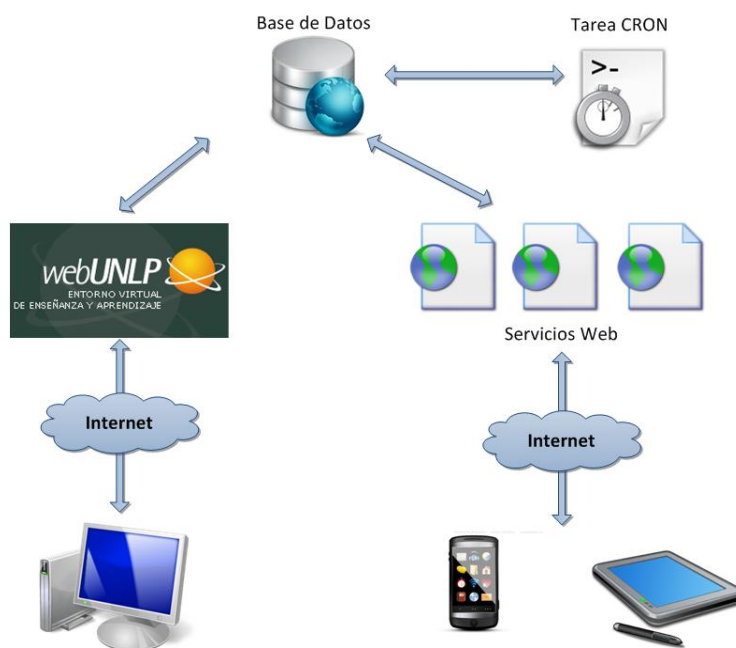


Figura 1 - Arquitectura genérica para aplicación nativa e híbrida

Para el diseño de una aplicación móvil nativa y una híbrida, la situación es más compleja. En la figura 1 se presenta la arquitectura genérica de todos los componentes que participan en este escenario de desarrollo. En ella se observa el acceso desde una PC a WebUNLP, y el acceso desde dispositivos móviles (celular y tablet) a la información de WebUNLP, mediante servicios web. Para el desarrollo de los servicios web se diseñó una API (Application Programming Interface) Restful dada su simpleza, escalabilidad e interoperabilidad [14] [15].

Asimismo existe una tarea programada (Cron) de ejecución intermitente a intervalos regulares de tiempo, la cual notifica a los dispositivos móviles correspondientes cuando se crea una novedad en la cartelera de WebUNLP. El Cron identifica el sistema operativo del dispositivo a notificar y genera dicha notificación.

En cuanto a la interfaz gráfica, se planteó un diseño independiente de la plataforma para analizar los aspectos de usabilidad de la aplicación, y luego se realizaron los ajustes necesarios para cada tipo de aplicación.

En las interfaces diseñadas la navegación es en serie, en el orden en que se presentan en el mockup [13] de la figura 2.

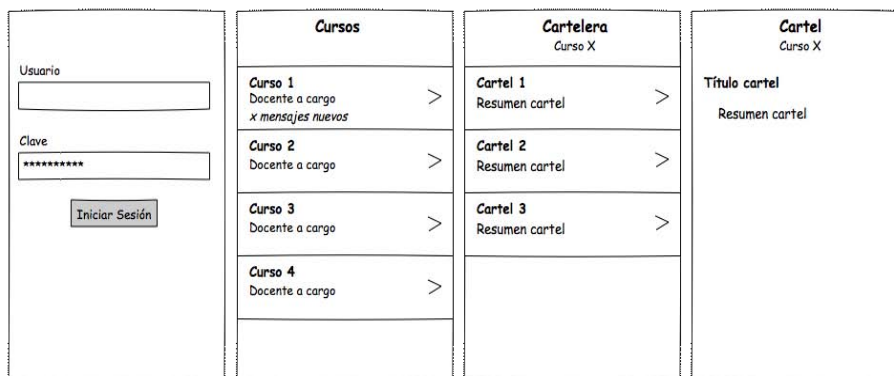


Figura 2 - Mockup independiente del tipo de aplicación

3.4 Desarrollo

3.4.1 Aplicación Nativa para Android

El desarrollo de aplicaciones para la plataforma Android requiere disponer de un JDK (Java Development Kit) y su entorno de programación, conocido como Android SDK (Software Development Kit). Este último provee librerías y herramientas necesarias para construir, testear y depurar aplicaciones para Android.

El desarrollo de la aplicación de WebUNLP para Android se realizó según las convenciones adoptadas por su comunidad, porque aplica buenas prácticas en la construcción de las interfaces y la lógica detrás de ellas, el acceso a datos y a servicios web. Para cumplir el requerimiento de las notificaciones en el dispositivo correspondiente, el Cron genera la notificación mediante el uso del servicio GCM

(Google Cloud Messaging) [16]. En la figura 3, se presentan las interfaces de la aplicación desarrollada.

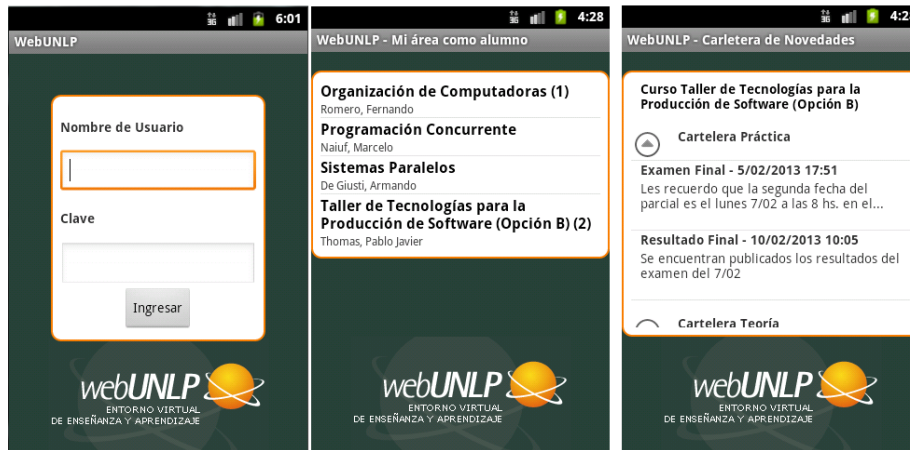


Figura 3 - Aplicación nativa para Android

3.4.2. Aplicación Nativa para iOS

La plataforma Apple iOS está basada en un modelo propietario, es por ello que el desarrollo de una aplicación nativa iOS implica contar con una Apple Mac corriendo OS X con Xcode instalado. El lenguaje de programación principal es Objective C. Xcode es el entorno de desarrollo de Apple para todos sus dispositivos y es el encargado de proporcionar el iOS SDK con las herramientas, compiladores y frameworks necesarios. Además, Xcode viene integrado con simuladores para dispositivos iOS (iPhones y iPads) que facilitan las etapas de prueba del sistema desarrollado.

Para los aspectos referentes a la interfaz gráfica y la interacción con el usuario, se siguieron las convenciones propuestas por Apple [10] para lograr una mejor integración de la aplicación con el sistema operativo y mejorar la experiencia del usuario.

Por último, para satisfacer el requerimiento de notificaciones, el Cron notifica al dispositivo iOS correspondiente mediante el uso del servicio APNs (Apple Push Notification Service) [17]. En la figura 4, se presentan las interfaces de la aplicación desarrollada.

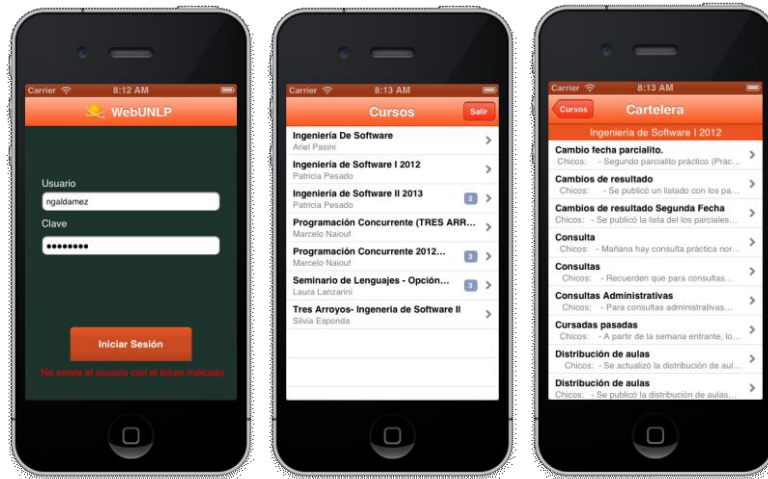


Figura 4 - Aplicación nativa para iOS

3.4.3 Aplicación Híbrida

Para la construcción de la aplicación WebUNLP híbrida se utilizó el framework PhoneGap [19], el cual permite desarrollar aplicaciones móviles que utiliza tecnologías comunes a todos los dispositivos: HTML5, CSS y Javascript.

Asimismo, se utilizó el framework Javascript denominado JQuery Mobile [20] para lograr interfaces con apariencia y comportamiento consistente a través de las diferentes plataformas móviles.

Con el objetivo de implementar el patrón de diseño MVC (Modelo, Vista, Controlador) se utilizó la librería Backbone.js [21].

Por último, para satisfacer el requerimiento de notificaciones, se utilizó el plugin Pushwoosh [18].

En la figura 5, se presentan las interfaces de la aplicación desarrollada.

3.4.4 Aplicación Web

Finalmente, se desarrolló una aplicación web capaz de acceder a la cartelera de WebUNLP. La misma se encuentra disponible para cualquier dispositivo móvil que cuente con un navegador que soporte las características utilizadas para el desarrollo: HTML5, CSS3, Javascript.

Como la velocidad de transmisión/recepción de datos de un dispositivo móvil a través de WiFi, y en particular 3G, es inferior a la velocidad de una computadora de escritorio, la versión web de la cartelera de WebUNLP es liviana y gran parte de los requerimientos son implementados a través de Ajax [12] para evitar, ante algún cambio, la recarga de la página completa.

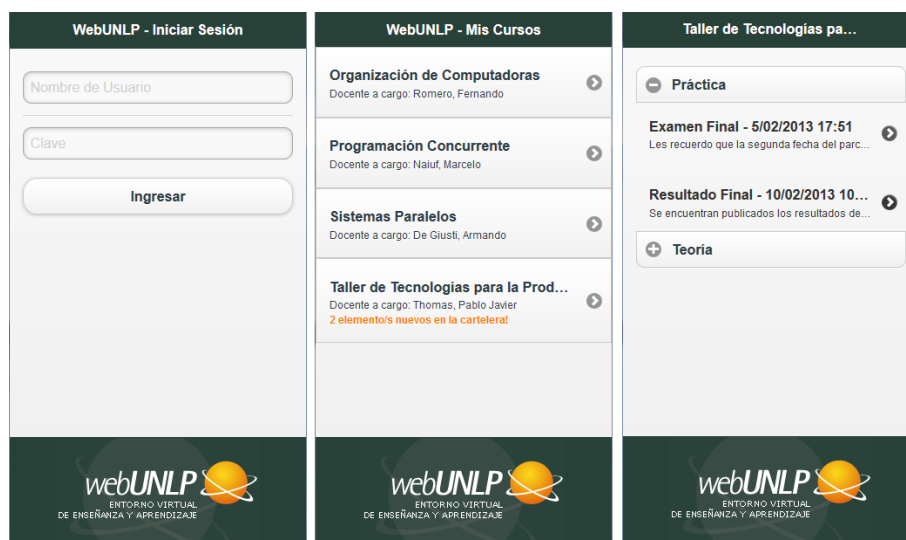


Figura 5 - Aplicación híbrida

4 Conclusiones

Inicialmente los dispositivos móviles fueron pensados y diseñados con un propósito especial. A través de los años, el crecimiento tecnológico ha permitido incorporar funcionalidades adicionales, lo que ha posibilitado expandir el marco de uso.

Actualmente, el poder de cómputo subyacente en una gran variedad de dispositivos móviles ha generado nuevas posibilidades, lo que constituye un reto para los ingenieros de software.

Es difícil establecer afirmaciones rotundas en este entorno de evolución vertiginoso y continuo. Es claro que, por el momento, existen tres posibilidades de desarrollo de una misma aplicación, y la incursión en cada una de ellas ha permitido establecer algunas conclusiones.

Seleccionar un dominio para desarrollar un caso experimental tiene sus particularidades. La novedad de disponer una aplicación móvil no es motivo suficiente que justifique tal desarrollo. Es necesario satisfacer un conjunto de requerimientos claramente planteados.

WebUNLP es un entorno virtual de enseñanza y aprendizaje utilizado por diversos cursos de grado y postgrado de la Universidad Nacional de La Plata. Por ende, la cantidad de usuarios beneficiados por acceder desde cualquier lugar a este entorno es importante.

De esta manera, se optó por elegir este espacio virtual para ser replicado en una versión móvil que no sólo permita acercarlo a sus usuarios, sino que además amplíe sus funciones, en este caso puntualmente para la cartelera.

Actualmente, se han desarrollado los tres tipos de aplicación móvil (web, nativa e híbrida) con el mismo propósito y para satisfacer el mismo conjunto de requerimientos.

Se destaca la gran simpleza de generar la versión web, dado que se utilizan las mismas herramientas tecnológicas que las usadas para el desarrollo de cualquier aplicación web tradicional. La principal diferencia en este sentido radica en la limitación de espacio en la pantalla del dispositivo. No obstante, el mayor contraste está dado en que no es posible acceder a todas las capacidades de hardware del dispositivo, lo que impidió implementar uno de los requerimientos, probablemente, el más interesante: la notificación de novedades de la cartelera de WebUNLP al usuario.

Por otra parte, las versiones nativas han cumplido todos los requerimientos. Aunque la mayor desventaja consiste en la no portabilidad, lo que implicó un desarrollo específico para la plataforma que se desee cubrir. En este trabajo se han presentado los desarrollos para Android e iOS, sistemas operativos más usados en Argentina, con un costo inherente mayor.

Finalmente, la versión híbrida ha logrado conjugar la simpleza del desarrollo web con el uso de todas las capacidades del dispositivo. Este tipo de enfoque pretende suplir las desventajas de los dos enfoques previos, hecho que lo posiciona como la elección prima facie, siempre condicionada por los requerimientos específicos a cumplir.

Como prueba de uso, recientemente se puso disponible la aplicación en su modo nativo, que logró captar la atención de los usuarios, traducida en el pedido de incorporación de nuevas funciones presentes en WebUNLP.

5 Trabajo futuro

Se pretende expandir a corto plazo algunos requerimientos cumplidos por WebUNLP que no se satisfacen en su versión móvil, como, por ejemplo, la mensajería y el foro, entre otros. Además, se espera incorporar nuevos requerimientos tales como un servicio de chat.

Desde el punto de vista del desarrollo, se analizarán alternativas para generar aplicaciones móviles híbridas mediante otros frameworks [22] [23].

Referencias

1. Talukder, A.K., Ahmed, H. and Yavagal,R.: *Mobile Computing, Technology, Applications, and Service Creation*. Second Edition. Tata McGraw-Hill. 2010.
2. Abrahamsson, P. *Mobile software development - the business opportunity of today*. Proceedings of the International Conference on Software Development, (pp. 20-23). 2005. Reykjavik.
3. Hayes, I. S. *Just Enough Wireless Computing*. Prentice Hall Professional Technical Reference . 2002. ISBN:0130994618
4. Abrahamsson P. et. al. , *Mobile-D: An Agile Approach for Mobile Application Development*. OOPSLA'04, Oct. 24–28, 2004, Vancouver, British Columbia, Canada.
5. Tracy, K.W., *Mobile Application Development Experiences on Apple's iOS and Android OS*, Potentials, IEEE, 2012

6. Sitio de WebUNLP. <http://webunlp.unlp.edu.ar>
7. Salmre, I. *Writing Mobile Code Essential Software Engineering for Building Mobile Applications*. Addison Wesley Professional, 2005
8. <http://gs.statcounter.com>
9. <http://www.mapbox.com/labs/twitter-gnip/brands/#4/-40.43/-63.62>
10. *iOS Human Interface Guidelines*,
<http://developer.apple.com/library/ios/#DOCUMENTATION/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html>
11. Raj R., Tolety S.B. *A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach*. India Conference (INDICON), 2012 Annual IEEE
12. <https://developer.mozilla.org/en/docs/AJAX>
13. <http://es.wikipedia.org/wiki/Mockup>
14. Richardson L., Ruby S., *RESTful Web Services*, O'Reilly Media, 2007.
15. <http://msdn.microsoft.com/es-es/magazine/dd315413.aspx?id0070023>
16. <http://developer.android.com/google/gcm/index.html>
17. http://developer.apple.com/library/mac/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html#apple_ref/doc/uid/TP40008194-CH100-SW9
18. <http://devgirl.org/2012/12/04/easy-phonegap-push-notifications-with-pushwoosh/>
19. <http://phonegap.com/>
20. <http://jquerymobile.com/>
21. <http://backbonejs.org/>
22. Digital Possibilities. Mobile Development Frameworks Overview <http://digital-possibilities.com/mobile-development-frameworks-overview/>
23. Markus Falk. Mobile Frameworks Comparison Chart, <http://www.markus-falk.com/mobile-frameworks-comparison-chart/>