

# Aplicaciones móviles no nativas

Por Jose Pablo Monge Bonilla

Algunos pensarán que es sumamente costoso, en cuestiones de dinero y tiempo, además de lento, el proceso de desarrollar dos o más versiones de la misma aplicación puede tornarse difícil, por desconocimiento y demás.

Cuando los sistemas operativos móviles empezaron a surgir, el proceso sí era así, y era justamente por eso que la cantidad de aplicaciones era limitada, e inclusive percibíamos que muchas de ellas existían para un sistema operativo, en lugar de todos los más usados.

Los tiempos cambian, y día con día innovan y vemos salir nuevas aplicaciones y nuevos sistemas operativos, pero también, nuevas maneras de desarrollo, las cuales nos permiten llegar a todos los sistemas operativos móviles que desee, o al menos los que consideremos necesarios para llevar a cabo nuestra idea.

Es así, como nacen los “*frameworks*” para desarrollo de aplicaciones móviles no nativas, muchos accesibles para todos los desarrolladores a nivel mundial, otros más restringidos. Antes de empezar a hablar más profundamente del tema, definamos que es un framework de este tipo: un framework para desarrollo de aplicaciones móviles no nativas, es una plataforma, la cual (por lo general) no utiliza el lenguaje nativo de los sistemas operativos móviles, y que nos permite realizar una lógica de negocio única, la cual es empaquetada para Android, iOS, Windows Phone, etc. Por lo general, utilizan lenguajes para desarrollo de páginas web, como HTML5, CSS3 y JavaScript, pero también existen plataformas que utilizan C++, Delphi, entre otros.

Con una instalación sencilla, y una configuración casi nula, estos frameworks han llegado para quedarse, brindando oportunidad de figurar en el ámbito móvil a desarrolladores sin importar su nivel de experiencia en alguna tecnología, y es justamente por eso, que utilizan lenguajes más comunes y conocidos por la comunidad de programadores mundial.

Podríamos concluir que para nivel personal, o para emprendedores es una herramienta más que satisfactoria cierto? Pero bueno, dejemos por un momento de lado esa perspectiva, y utilicemos la perspectiva de negocio, de las grandes compañías de desarrollo a nivel mundial, o las que han ingresado al mundo móvil.

¿Cuál es el valor agregado de estos frameworks a mi negocio como tal? La respuesta es simple, reducción de costos. Se sabe, que un desarrollador de aplicaciones nativas es caro, y que dependiendo de la organización, son recursos dedicados no aprovechados al cien por ciento, así que se convierten en un gasto. Un gasto necesario, tal vez, pero al fin y al cabo, un gasto. Es más barato, mucho más barato mantener tres desarrolladores web, que inclusive a dos desarrolladores nativos, uno para Android y otro para iOS (citando un ejemplo).

Entonces, ¿Podemos concluir, que la solución a mis problemas son estos frameworks? Respuesta simple de nuevo, no. Si bien es cierto me reduce el costo, y hace la implementación más rápida, no siempre es la solución más acertada, realmente depende del tipo de proyecto que vamos o que pretendemos desarrollar. PhoneGap por ejemplo, presenta limitaciones en la brújula, manejo de ciertos archivos y de medios; Unity, presenta limitaciones en el uso de realidad aumentada, entre otros.

Todo gira en torno a la necesidad. Hay ocasiones donde una aplicación de este tipo es genial, se adapta al cien por ciento a lo que deseo desarrollar, hasta se vuelve más sencillo, ya hay controles específicos y demás; hay otras ocasiones donde si funcionan, pero hasta cierto punto, así que se realiza una aplicación al mejor estilo de Frankenstein, utilizando el framework y código nativo; a como hay ocasiones donde simplemente no se puede, y ese no puede convertirse en un no rotundo.

Es interesante la evolución de las tecnologías móviles, y como las mismas se han ido adaptando y mejorando de gran manera a las necesidades presentadas, sin embargo, nunca se deben de preferir a éstas, contra el desarrollo nativo. Ya hablé de ventajas, las cuales son muy diversas, es hora de mencionar las desventajas, que aunque son pocas relativamente hablando, son muy importantes de recalcar.

Tal vez el más importante de recalcar, es la cuestión del desempeño de la aplicación, ya que nadie desea una aplicación lenta, o que nunca responda, o se cierre a cada momento de ejecución, y esto, es un aspecto a tener mucho en cuenta al realizar una aplicación no nativa. No es cierto que al hacer la aplicación nativa esto no va a suceder, pero es más fácil de controlar, por la buena comunicación entre el lenguaje nativo (valga la redundancia) y el dispositivo.

Para concluir, podemos decir con seguridad que las aplicaciones nativas son las más robustas y de mejor desempeño, sin embargo, todo se apega a la necesidad presentada. No siempre vale la pena desarrollar cualquiera de las dos, hay veces, donde es mejor que sean no nativas, hay ocasiones donde es mejor que sean nativas, repito, es cuestión de necesidad.