

# Construcción de Aplicaciones Móviles con Acceso al Hardware de los Dispositivos

Rocío Andrea Rodríguez<sup>1,2</sup>, Pablo Martín Vera<sup>1,2</sup>, María Roxana Martínez<sup>1</sup>,  
Luís Verbel de la Cruz<sup>1</sup>, Federico Ezequiel Vallés<sup>2</sup>

<sup>1</sup>CAETI, Centro de Altos Estudios en Tecnología Informática  
Facultad de Tecnología Informática, Universidad Abierta Interamericana (UAI)  
Av. Montes de Oca 745, Ciudad Autónoma de Buenos Aires, Argentina  
{ RocioAndrea.Rodriguez; PabloMartin.Vera; Maria.Martinez; Luis.Verbel }@uai.edu.ar

<sup>2</sup>GIDFIS, Grupo de Investigación, Desarrollo y Formación en Innovación de Software  
Departamento de Ingeniería e Investigaciones Tecnológicas  
Universidad Nacional de La Matanza (UNLaM)  
Florencio Varela 1903, San Justo, Buenos Aires, Argentina  
{ rrodriguez; pvera; fvalles }@ing.unlam.edu.ar

## Resumen

La alta inserción de los dispositivos móviles, entre ellos los teléfonos celulares, hace que surja la necesidad de construir aplicaciones planificadas para este tipo de dispositivos. Claramente hay limitaciones en cuanto a tamaño de pantallas, teclados reducidos o inexistentes (contando únicamente con teclados virtuales), etc. No obstante también estos dispositivos cuentan con una importante cantidad de sensores (Proximidad, NFC, etc.) junto con otros componentes (Micrófono, Cámara, etc); los cuales pueden ser accedidos desde las aplicaciones. Al momento de decidir si es conveniente crear una aplicación web o nativa, uno de los factores decisivos era si dicha aplicación necesitaba acceder a recursos de hardware; dado que desde la web no se contaba con esta posibilidad. Actualmente esta condición ha cambiado existen APIs que permiten acceder a gran parte del hardware desde una aplicación web. El objetivo de esta línea de I+D (Investigación y Desarrollo) es en primer lugar mostrar cómo se reduce la brecha

entre las aplicaciones web móviles y nativas; y por otra parte investigar el funcionamiento de las APIs que hacen esto posible.

**Palabras clave:** Dispositivos Móviles, Hardware, Aplicaciones Web, HTML 5, Aplicaciones Nativas, Android

## Contexto

Esta línea de I+D forma parte de los proyectos radicados en el Laboratorio de Algoritmos y Software del Centro de Altos Estudios en Tecnología Informática (CAETI) de la Universidad Abierta Interamericana (UAI). El proyecto cuenta con financiamiento asignado y una duración de 2 años siendo su fecha de inicio en Agosto del 2013. En el proyecto colabora también la Universidad Nacional de La Matanza.

## Introducción

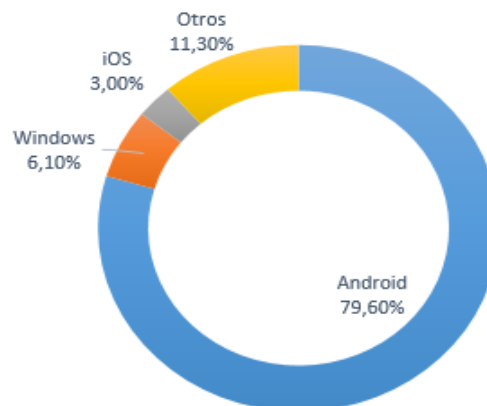
Actualmente los teléfonos celulares, de alta gama, traen un conjunto de

componentes y sensores que podrían ser utilizados en la construcción de aplicaciones más elaboradas. Entre los sensores actuales se encuentran: Acelerómetro; Giroscopio; Campo Magnético; Brújula; GPS; Proximidad; Luz ambiente; Cámara; Micrófono; Presión Atmosférica; Temperatura Ambiental; Humedad Relativa. Incluso los últimos equipos comienzan a incorporar sensores corporales, como por ejemplo medidor de pulsaciones. Esto permite generar aplicaciones innovadoras utilizando el hardware de los dispositivos.

En base al informe del 2014 disponible en la biblioteca física de la CNC (Comisión Nacional de Comunicaciones), es posible advertir que las compras de teléfonos celulares han cambiado, notándose una preferencia marcada por Smartphone por sobre otros teléfonos de gama media ó baja (ver figura 1). En el 2014 el 79% de las unidades adquiridas son Smartphones [1].

En esta línea de I+D se considera tanto el desarrollo de:

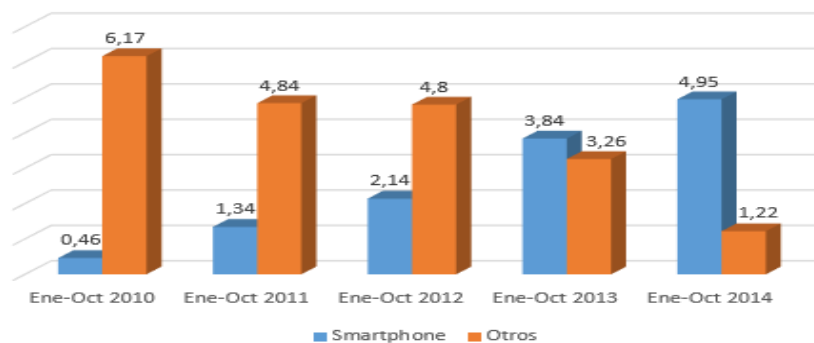
- Aplicaciones nativas, seleccionándose a ANDROID por ser el sistema operativo que utilizan más usuarios en argentina. Lo cual puede verse en la figura 2 (cuyos datos han sido obtenidos de la biblioteca física de la CNC (Comisión Nacional de Comunicaciones) [1].
- Aplicaciones web móviles: basadas en HTML 5.



**Figura 2.** Sistemas Operativos Móviles

Las aplicaciones nativas siempre han tenido como punto favorable la posibilidad de acceder al hardware de los dispositivos. Si bien es cierto que las aplicaciones web no cuentan con acceso total al hardware cada vez surgen más APIs que permiten acceder a distintos sensores y componentes. Entre ellos se puede mencionar a:

- Sensores de Movimiento: esta especificación provee eventos DOM (Document Object Model) para recuperar información que describe la orientación física y el movimiento del dispositivo. Se puede acceder a las coordenadas tri-dimensionales para ubicarse en el espacio registrando el evento `DeviceOrientation` en el objeto `window` del navegador mediante javascript.
- Estado de la batería: esta API se utiliza para determinar el estado de la



**Figura 1.** Cantidad de Ventas –Expresada en Millones de Unidades

batería del dispositivo. Mediante la creación del objeto `BatteryManager` es posible obtener diferentes atributos de la batería registrando eventos como: `onchargingchange` para obtener el estado de la carga, `onchargingtimechange` para saber el tiempo restante para que finalice la carga de la batería. Cuando la batería se está descargando y se quiere saber el tiempo restante útil, se puede usar el evento `onchargingtimechange`.

También se puede obtener rápidamente el nivel de carga mediante el evento `onlevelchange`. Saber el nivel de carga de la batería permitirá realizar aplicaciones que se adapten a esta condición para gastar menos batería o para realizar copia de seguridad de los datos para no perder información.

- **Sensores de proximidad:** con esta API se puede acceder al sensor de proximidad del equipo, el cual permite determinar si se encuentra cerca o no de otro objeto, generalmente el sensor de proximidad en los teléfonos móviles se encuentra sobre la pantalla y se utiliza para apagar la pantalla durante una llamada cuando el usuario lo acerca a su oído. Por lo que esta API nos permitirá detectar la presencia de un objeto cerca de dicho sensor. Se dispone de del evento `userproximity` que posee un atributo denominado `near` cuyo valor, cuando está en verdadero, indica la presencia de un objeto cercano y cuando está en falso es que no hay objetos cerca.
- **Sensores de luz ambiente:** permite medir el nivel de luz ambiente del lugar donde se encuentra el dispositivo. La API provee el evento

`deviceLight` que contiene una propiedad que representa el valor de luz ambiente medido en lux. Dicho valor puede variar entre dispositivos debido a los métodos de detección o la construcción de los sensores.

- **Vibración:** permite hacer que el dispositivo vibre por un tiempo determinado. Esto es útil para poder brindar feedback al usuario ante determinados eventos. Se utiliza directamente el método `vibrate` presente el objeto `navigator`. Como parámetro dicho evento recibe la cantidad de milisegundos que dura la vibración. Una sobrecarga de dicho método permite pasar tres parámetros indicando tiempo de una primer vibración, tiempo de espera y tiempo de una segunda vibración.
- **Presión atmosférica:** permite obtener el valor de la presión atmosférica mediante la utilización de un barómetro o sensores similares. Este valor puede utilizarse para detectar la altitud a la que se encuentra el dispositivo o para brindar o predecir condiciones climatológicas.
- **Sensor de Temperatura ambiente:** retorna el valor de la temperatura del lugar donde se encuentra el dispositivo medida en grados Celsius.
- **Sensor de humedad:** mide la humedad ambiente relativa en porcentaje.
- **Camara y micrófono:** permite acceder a la cámara y micrófono directamente desde el navegador, lo que permitirá por ejemplo realizar una aplicación de video conferencia sin necesidad de instalar una aplicación específica.
- **NFC [2]:** Esta API permite realizar comunicaciones NFC directamente desde el navegador. Algunas de las utilidades de la API [3] nombradas por la W3C son `Tap to` (tocar para):

Reproducir, Compartir, Controlar,  
Conectar, Leer, Escribir.

## Líneas de Investigación, Desarrollo e Innovación

Dentro de la presente línea de investigación se analizan los siguientes temas:

- Uso de APIs para la generación de aplicaciones web móviles con acceso al hardware (geolocalización, uso de NFC para personalizar configuraciones, realidad aumentada y manejo de interfaces por medio de sensores).
- Comparación de performance entre aplicaciones nativas y aplicaciones web en un mismo equipo (es decir con igual condición de hardware).
- Metodología de testeo de Aplicaciones Web

## Resultados y Objetivos

Los resultados obtenidos están vinculados al conocimiento del manejo del hardware de los dispositivos en aplicaciones web, por ello en el ámbito de Geolocalización se centró inicialmente el estudio. En base principalmente al GPS y el conocimiento adquirido en cuanto al manejo de la API de Google Maps se generó una aplicación web móvil que permite mostrar la ubicación de las sedes de la UAI (Ver figura 3). Luego se construyó otra aplicación con las mismas prestaciones para ANDROID, con la finalidad de poder comparar el rendimiento de ambas soluciones nativa vs web. Esta aplicación incluye: Marcadores para las localizaciones de UAI (sedes, anexos, dependencias, etc); Posición Actual del Usuario; Actualización de la Posición del Usuario;

Cálculo de Rutas e Información de la Ruta. Un primer resultado de la comparativa ha sido que “Si bien los tiempos de espera son despreciables en ambos casos, en cuanto a la carga inicial la aplicación en HTML 5 demora el doble que la nativa pero luego los tiempos de cálculo de rutas son inferiores en la solución web que en la nativa”[4]



**Figura 3.** Vista de la Aplicación Web en distintos dispositivos móviles

Esto ha sido un primer paso que permitió comenzar a trabajar en circuitos turísticos para exteriores diseñando una aplicación móvil que permita en un ambiente colaborativo proveer información a los usuarios y ofrecerles circuitos predefinidos los cuales puedan ser valorados y comentados por los usuarios de la misma.

Otro resultado importante de este proyecto ha sido en materia de realidad aumentada, utilizando la cámara del dispositivo móvil es posible añadir información de contexto al entorno del usuario. Por medio del framework VUFORIA [5] se comenzó a trabajar en la identificación de patrones y la presentación de modelos 3D en el lugar de ellos (ver figura 4). Esto es el paso inicial, hacia el trabajo de folletos aumentados. El objetivo es poder aumentar el folleto añadiendo más información a las personas interesadas accediendo a modelos 3D, videos explicativos, ilustraciones, etc.



**Figura 4.** Objeto sobre un patrón reconocido por la aplicación

El poderío de los teléfonos celulares de alta gama (smarphone), permite un gran abanico de aplicaciones que requieren el acceso al hardware y actualmente esto es posible realizarlo ya sea desde aplicaciones nativas como desde la web por medio de HTML 5. Se hicieron pruebas de las distintas APIs para acceder al hardware. Y se comprobó que si bien estas funcionan al realizar pruebas con dispositivos diferentes (distintas marcas) un sensor arroja valores distintos por ejemplo el sensor de proximidad (ver tabla 1). Esto implica que no hay una normalización y la mayoría de las APIs están aún en etapas de desarrollo.

**Tabla 1.** Ejemplos de Valores Arrojadados por el Sensor de Proximidad

Dispositivo	Valor Referencia	
	Cerca	Lejos
LG Nexus 4	0	5
Motorola MotoX	3	100
Motorola MotoG	3	100
Motorola Razer	3	100
Motorola Spice KeyXT316	0	9
Samsung Galaxy Si9003GT	0	5
Samsung S3 Mini	0	5
Samsung S4	0	8
Samsung S4 Mini	0	5

También es importante destacar que hay un bajo soporte por parte de los navegadores, siendo Mozilla Firefox el único que soporta todas las APIs.

## Formación de Recursos Humanos

El equipo de trabajo está compuesto por: 3 Docentes, 2 Becarios, 1 Pasante.

Dada la importancia de la temática, en esta área de investigación se encuentran vinculadas 1 tesis doctoral de uno de los miembros del equipo de investigación y 4 tesis de maestría:

- 1 tesis de Doctorado (UNLP)
- 3 tesis de Maestría (UAI)
- 1 tesis de Maestría (UNLaM)

## Referencias

1. Atlas y Anuario de las Comunicaciones (2014). Año XIX, Numero 228, Diciembre del 2014. Editorial Convergencia Telemática.
2. R. Rodríguez, P. Vera, M. R. Martínez, D. Giuliane - "Context Aware Applications on Mobile Environments – Engaged by the use of NFC". International Conference on Multimedia, Scientific Information and Visualization for Information Systems and Metrics. 29-31 de Enero de 2014. Maribour, Eslovenia.
3. W3C. Web NFC API (2014). <http://www.w3.org/TR/nfc/>
4. Rodríguez, R. A., Vera, P. M., Martínez, M. R., & Verbel de La Cruz, L. (2014). Aprovechamiento del hardware de los dispositivos móviles para la construcción de nuevas aplicaciones. XVI Workshop de Investigadores en Ciencias de la Computación.
5. QualcommVuforia. <https://www.qualcomm.com/products/vuforia>