

# Reducing the Gap between Native and Web Applications

Rocío A Rodríguez, Pablo M Vera, María R Martínez, Claudia Pons, Federico E Valles, Luis Verbel de La Cruz

Interamerican Open University (UAI)  
Higher education center in information technology  
City of Buenos Aires, Buenos Aires, Argentina  
{RocioAndrea.Rodriguez; PabloMartin.Vera; RoxanaMaria.Martinez} @uai.edu.ar

National University of La Plata  
LIFIA – Research and Education Laboratory on Advance Computing  
La Plata, Buenos Aires, Argentina  
cpons@lifia.info.unlp.edu.ar

National University of La Matanza  
Department of Engineering and Technological Research  
San Justo, Buenos Aires, Argentina  
{rrodriguez, pvera, fvalles}@ing.unlam.edu.ar

**Abstract.** New web standards are emerging day after day. Some of them are crucial in the improvement of web application for mobile devices. From visualization issues to hardware access, all this new tools can help developers to create native like applications on web environments. HTML5 and CSS3 with css media queries are some examples that are already available to use. Some other standards are still in progress but nearly to arrive. This paper shows existing and future standards for mobile web and shows implementation examples of some of them.

**Keywords.** WEB MOBILE, NATIVE MOBILE, HTML 5, CSS3

## 1. Introduction

Software development process includes several steps, data designing, defining the When an application for a mobile device must be develop, the big question is: should we create a native or a web application? Of course the answer is not simple. It is relative to the requirements of that application. Anybody will say that Native applications are more powerful than web applications, and it's true. But new standards are emerging to reduce that gap. Table 1 shows the capabilities of each type of application.

**Table 1. Native App vs Web App**

	Native App	Web App
Hardware Access	Full	Limited (but growing)
Connectivity	Online and offline	Mostly Online but can be offline
Portability	Only on the same operating system and restricted by the supported version.	Can be used on any device with any operation system supporting web standards
Graphics	Can use all hardware capabilities to create graphical stunning applications like games.	Limited to the visualization supported by the browser.
Look and Feel	Access to native controls allows creating standard applications.	With the use JavaScript frameworks, native controls can be simulated on the browser giving the user the feel of being using a native application.

## 2. Mobile Web Applications

This section will show some strategies for creating more powerful mobile web applications using some new standards.

### 2.1. Off Line Capabilities

HTML5 includes a new way of handling cache for web application using a manifest file. This file can control the way each file is treated. Manifest file includes three sections:

- Cache Manifest: Files to be cached
- Network: Files to be always downloaded from the server
- Fallback: Files to be shown if no connection if available

An example cache manifest can be:

```
CACHE MANIFEST
# 2012-02-21 v1.0.0
theme.css
logo.gif
main.js
NETWORK:
login.asp
FALLBACK:
/html5/offline.html
```

Line starting with a # is a comment. It's a good practice to include the update date of the file so each time the file change the server will download all pages and replace

it's cache. So comments can be used to force the download of new version of the pages.

AppCache can be used to create web applications that works completely offline, but with the limitation that if the user clears the browser cache it will also lose access to the application until being online again.

## 2.2. Storage

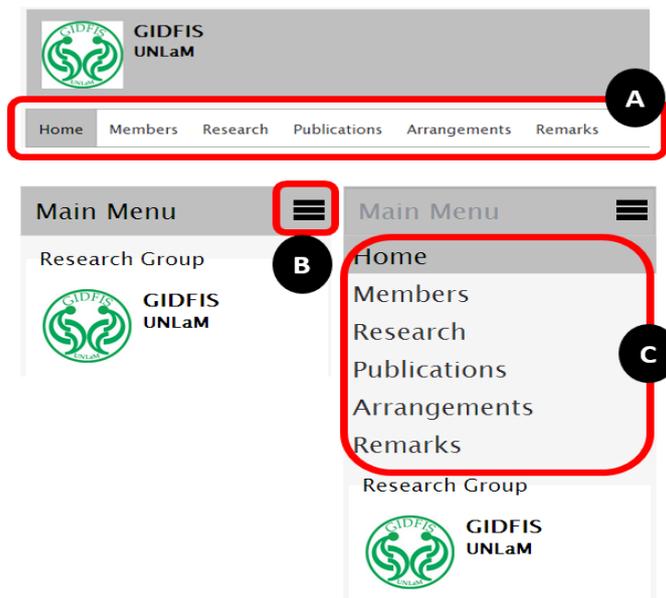
Web applications usually store data in the cloud using some server side scripting. But some applications needs to store data locally. HTML5 includes new storage capabilities with WebStorage feature.

WebStorage allows keeping large volumes of information without affecting the website performance in contrast to cookies where the information travels from server to client on each request. There are two bags to store data: LocalStorage and SessionStorage.

- SessionStorage: keep data while the http session is alive.
- LocalStorage: keep data permanently on the device or until the user explicitly deletes browser data.

## 2.3. Visualization

A key feature of HTML is the ability to adapt to screen of the device being used, from relative units to complex css, it allows creating a unique site that will fit different types of screens. CSS Media queries allows adapting the interface regarding the screen resolution so for example if the site is being used in a tablet it could show more controls if it's being used in a cell phone. This is called Responsive Design [4], [12]. Some frameworks like jquery mobile [9] includes responsive controls available to be used. Figure 1 shows an example of a responsive menu. Section A shows how the menu is expanded in high resolution screens. Section B shows that in a narrower screen the menu will be collapsed and an icon will appear. Section C shows the expanded options of the menu when clicking the icon.



**Fig. 1.** Example of responsive menu built with CSS media queries and its visualization regarding the screen resolution

There are several ways of using CSS Media Queries [12]:

- Including a query in the link to a CSS file

```
<link rel= "stylesheet" type= "text/css"
  media= "screen and (max-device-width: 480px)"
  href= "customstyle.css" />
```

In this case the CSS will be applied to the page only if the screen width is lower than 480 pixels.

- Including queries inside the CSS file

```
@media screen and (max-device-width: 480px)
{
  .column
  { float: none;}
}
```

- Using an import directive including the query

```
@import url ("customstyle.css") screen and (max-
device-width: 480px);
```

### 2.4. Hardware and Information Access

Web applications cannot access all hardware of the mobile devices, but new tools are emerging to be able to do so. A clear example is geolocation capability of HTML5. It allows getting the user current position by means of wireless network and the GPS if it's available. But a set of new standards are being develop to access device hardware like:

- Motion Sensors
- Battery Status
- Proximity Sensors
- Ambient Light Sensors
- Humidity Sensors
- Camera and Microphone
- NFC

A detailed explanation and current status of each of the features can be found at [13].

Other features are being developed that will allow accessing user information available in the mobile device like Address book and Calendar Data [14].

### 2.5. Look and Feel

Web applications can look like native applications thanks to numerous JavaScript frameworks already available. For example IUI [8] imitates Iphone user interface (figure 2).

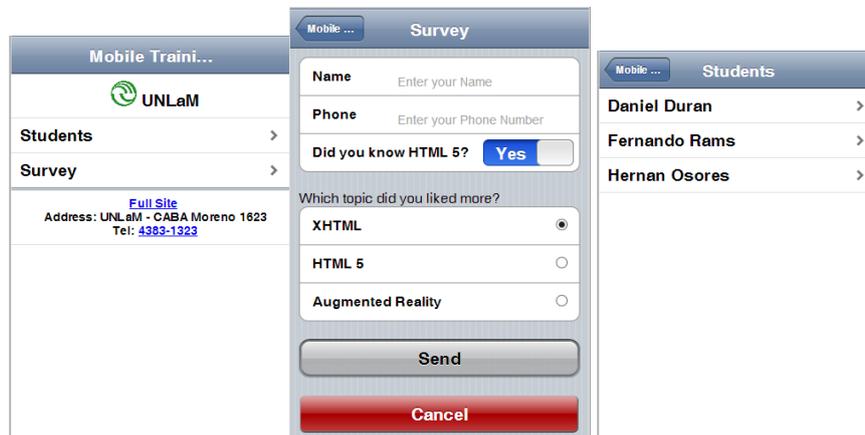


Fig. 2. Examples of a web application developed with IUI

Another usefull framework is JQuery Mobile [9] wich includes a custom framework for mobile applications, with several controls, including responsive design. Figure 3 shows some web pages made with jquery mobile.

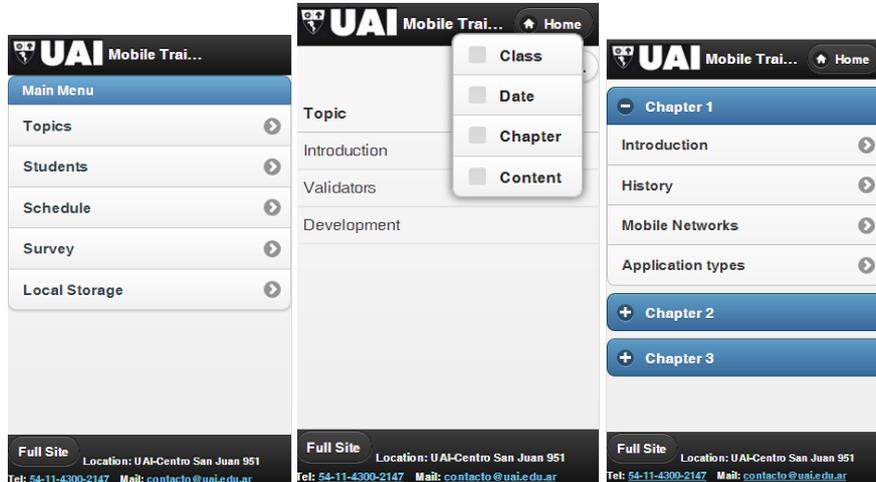


Fig. 3. Examples of a web application developed with JQuery Mobile

Several other javascript frameworks are available some are more powerful than others, but all of them use javascript and css to generate a rich user interface.

### 3. A Sample Application

Interamerican Open University (UAI) is organized in ten faculties: Architecture, Computer Science, Business Studies, Law and Political Science, Development and Educational Research, Medicine and Health Sciences, Human Movement and Sport Psychology and Human Relations, Information Technology, Tourism and Hospitality. It also offers postgraduates, and open courses are offered to the Community. University has sixteen locations in Argentina, placed at Buenos Aires City, Buenos Aires Province and Santa Fe province.

It's important for students knowing where to go for administrative procedures, taking a course or borrowing a book. A student during the carrier will probably need going to different locations (grade course at one location, postgraduate at another location and research internship in a third location). So having a mobile application capable of knowing user current position and the route to reach a desire location is very useful.

Main goal was building an application with the following functionalities:

- Mark each location in a map
- Show user's current position
- Keep position update when the user moves
- Routing Calculation
- Route Information

Thanks to hardware advances new mobile devices can execute javascript very quickly, so Javascript Google Maps API v3 [6] was choose to draw the maps. This API also allows adding marker and calculating routes.

Google API has a parameter to enable geolocation using device sensors (GPS, mobile networks). This parameter is set when referencing google js file specifying the option `sensor=true`.

To determine user's current position, google has another API called Google Maps Geolocation API [5] but this API can be replaced by the new geolocation feature of HTML5. User location can be determined by using the function `navigator.geolocation.getCurrentPosition`.

Keeping user position when moving can be also made with HTML5. It includes the function `navigator.geolocation.watchPosition` which track user location changes in an asynchronous way. This function has a parameter to enable High Accuracy of geo location that will force to use the device GPS to get a more accurate position. To save battery two buttons were added to the application to start tracking current position and cancel the tracking anytime by the user.

Figure 4 shows the mobile web application running in different mobile devices with different operating systems.



Fig. 4. Sample Geolocation Web Application using Javascript and HTML5

As can be seen choosing a web application over native application easiness the development task by creating a unique app that fits several operation systems and screens sizes. In order to compare speed we also build an Android native application with the same requirements and notice no significant advantages on the speed on the application behavior.

In an application with giving requirements building a native application has no advantages over the mobile version. It will worth it if we need to use another sensors not available yet in the web environment

## 4. Performance

Once developed the web application, an Android Native application was also built in order to compare performance of both applications. Using the same smartphone response time of the two application were measure on Initial Map Loading and rendering and when calculating and showing a route.

### 4.1. Time Measuring

Next, functions used to measure times on both applications are explained:

1. On Android Native application a system function was used:  
`System.currentTimeMillis()`
  - For map initial loading : times are measure withing OnCreate event.
  - For showing route: times are measure starting on `onInfoWindowClick` event that fires when user clicks a marker and the final time is measure on `OnPostExecute` event.
2. For web application times are measure using javascript `new Date().getTime();` method

### 4.2. Results

From measure results it can be noticed that response time of both applications are really low, being faster the native application. However response time are low on both applications giving a good user experience. So response time is not a vital feature to choose the native application over the web application.

## 5. Related Work

Several authors are considering that web applications are getting closer to native applications, for example INTEL [7] analyses the development capabilities of the different mobile operation system and concludes that HTML5 is the best choice because it's common for all of them. So they recommend using it in conjunction with some JavaScript framework to develop a simple application targeting server devices. CHARLAND Andre, LEROUX Brian [1] compares web and native applications, remarking pros and cons of each technique. Remarking the increasing power of web applications also believing that web gaming will be also possible in mobile

environments thanks to the growing implementation of WebGL [11] (a 3D rendering environment for browsers).

CHRIST Adam [2] remarks the advantages of developing web applications over native applications. It also remarks the power of javascript frameworks to create the user interface of new web applications.

FLING Brian dedicates a chapter of his book “Mobile Design and Development: Practical Techniques for Creating Mobile Site and Web Apps” [3] to compare web and native applications. He remarks that if it’s possible for the porpoise of the solution web applications should be preferred over native applications: “I’m a big fan of native application and I feel that there are a lot of great innovative and market opportunities here, but mobile web apps are the only long-term viable platform for mobile content, services, and applications.”

## 6. Conclusions and Future Work

The gap between capabilities of native applications over web applications is narrowing, especially boosted by hardware access web standards. Besides developing a web application ensure it to be multi-platform in contrast to native applications that requires multiplying efforts for developing on each different platform for each mobile operating system. Thus also keeping updated web applications is simpler than maintaining native applications.

Regarding look and fell, there are several JavaScript frameworks that can help web applications to look like native applications, showing native like controls in a web page. Nowadays it begins to be complex distinguishing between a native a web application. Style sheets can now be conditionally applied with CSS Media Query, so the design can match the device screen creating a more usable interface, taking advance of bigger screens and giving a good user experience on small screens without having to develop several web sites.

We are convince that the future is on having applications in the cloud, where user can access then from different devices without losing previously loaded information. HTML5 allows keeping information on the device when it’s disconnected and sync it when internet connection is available.

All this features together, with hardware access capabilities, opens a wide range of opportunities of developing more powerful web applications.

Initially applications were almost entirely native, then began considering hybrid applications (those applications built in a common framework and then compiled in different pre-defined operating system, more powerful than web application but not as powerful as native applications). Nowadays different authors share our view that the future is in web applications.

## References

- [1] CHARLAND Andre, LEROUX Brian. “Web apps are cheaper to develop and deploy than native apps, but can they match the native user experience? - Mobile application Development: Web vs. native”. Vol. 54, No. 5. ACM (2011)  
<http://www.cemfarma.com/images/uploads/urun/pdf/bd1b62bc456d0240f58a8720fa66df1fd a24046aS249V3.pdf>
- [2] CHRIST Adam M. “Bridging the Mobile App Gap”. Sigma noblis, Volume 11, Number 1, pp.27-32 (2011)  
[http://www.noblis.org/media/2dd575c1-2de9-4d92-9bdb-f72ad9fb9a19/docs/SigmaDigEco2011\\_pdf](http://www.noblis.org/media/2dd575c1-2de9-4d92-9bdb-f72ad9fb9a19/docs/SigmaDigEco2011_pdf)
- [3] FLING Brian. “Mobile Design and Development: Practical concepts and techniques for Creating Mobile Sites and Web Apps”. O’Reilly. ISBN 978-0-596-15544-5. Chapter 9: “Mobile Web Apps Versus Native Applications”. pp. 143-150 (2011).
- [4] GARDNER Brett S. “Responsive Web Design: Enriching the User Experience”. Sigma noblis, Volume 11, Number 1, pp.13-19 (2011)  
[http://www.noblis.org/media/2dd575c1-2de9-4d92-9bdb-f72ad9fb9a19/docs/SigmaDigEco2011\\_pdf](http://www.noblis.org/media/2dd575c1-2de9-4d92-9bdb-f72ad9fb9a19/docs/SigmaDigEco2011_pdf)
- [5] GOOGLE. “The Google Maps Geolocation API”. (2013)  
<https://developers.google.com/maps/documentation/javascript/?hl=es>
- [6] GOOGLE. “Google Maps Javascript API V3 Reference” (2013)  
<https://developers.google.com/maps/documentation/javascript/reference?hl=en>
- [7] INTEL. “Bridging the Gap: from a Web App to a Mobile Device App”. HTML 5 DevCon (2013)  
[http://html5devconf.com/archives/april2013/slides/2013HTML5DevCon\\_Bridging-the-Gap.pdf](http://html5devconf.com/archives/april2013/slides/2013HTML5DevCon_Bridging-the-Gap.pdf)
- [8] IUI “iUI: User Interface Framework for Mobile Web Devices” (2013)  
<https://code.google.com/p/iui/>
- [9] JQUERY “jQuery Mobile: jQuery Mobile: Touch-Optimized Web Framework for Smartphones & Tablets” (2013)  
<http://jquerymobile.com/>
- [10] KAMEL BOULOS Maged N, WARREN Jeffrey, GONG Jianya, YUE Peng “Web GIS in practice VIII: HTML5 and the canvas element for interactive online mapping”. International journal of health geographic (2010)  
<http://www.biomedcentral.com/content/pdf/1476-072X-9-14.pdf>
- [11] KHROSOS Group. “WebGL Specification” Version 1.0.2 (2013)  
<https://www.khronos.org/registry/webgl/specs/1.0/>
- [12] MARCOTTE Ethan. “Responsive Web Design. A List Apart Magazine: Articles”. (2010).  
<http://www.princeton.edu/~mlovett/reference/A%20List%20Apart-Articles-Responsive%20Web%20Design.pdf>
- [13] W3C. “Standards for Web Applications on Mobile: current state and roadmap - Sensors and Hardware Integration” (2013)  
[http://www.w3.org/Mobile/mobile-web-app-state/#Sensors\\_and\\_hardware\\_integration](http://www.w3.org/Mobile/mobile-web-app-state/#Sensors_and_hardware_integration)
- [14] W3C. “Standards for Web Applications on Mobile: current state and roadmap – Personal Information Management” (2013)  
[http://www.w3.org/Mobile/mobile-web-app-state/#Personal\\_Information\\_Management](http://www.w3.org/Mobile/mobile-web-app-state/#Personal_Information_Management)