



Mobile Web Apps

Nicolás Serrano, Josune Hernantes, and Gorka Gallardo

With smartphones being the primary handheld device for more than a billion people, mobile Web apps are a necessity in both technical and commercial fields. There are several approaches to developing mobile Web apps, but given the fast speed of mobile software evolution, in which the leading companies become marginal in months and new gadgets continually appear, it's crucial to understand the basic technologies. Authors Nicolás Serrano, Josune Hernantes, and Gorka Gallardo examine current development approaches that can enhance the decision-making process. I look forward to hearing from both readers and prospective column authors about this column and the technologies you want to hear more about. —*Christof Ebert*

WHEN WE TALK about designing a new application, we're usually talking about building a Web application—that is, an application that runs in a browser. In the past, the tendency was to begin by creating desktop applications specifically for an OS like Windows or Unix. But ultimately, most developers want to make the applica-

tion work for mobile devices.¹ The decision-making process for creating mobile apps is complex owing to an

Mobile Web and Apps

The idea of a *mobile Web app* has different meanings for different people.

others think of a native app built for a specific mobile OS that connects to the Web to send or receive data. To differentiate between these concepts, we can use a scale ranging from standard to native Web applications. By definition, native apps run at native speed, whereas hybrid and mobile apps run on top of additional layers, which consumes computing resources and can decrease the app's speed.

Here, we look at standard, responsive, and mobile Web as a single category because their technologies as well as their pros and cons are very similar. We define them separately in this section to highlight that they are in fact different solutions. For example, some users and developers like responsive design, whereas others prefer an ad hoc option for specific devices (see the sidebar for examples). Even mobile websites (for example, Facebook or Google

There isn't a single solution for choosing which mobile application type will be best for every situation.

For some, it refers to an application that runs on the Web and has been designed to appear correctly on a mobile device;

tion work for mobile devices.¹ The decision-making process for creating mobile apps is complex owing to an



EXAMPLES OF MOBILE APPS

Openbravo is an open source enterprise resource planning (ERP) system (<http://demo.openbravo.com/retail>).¹ It's a Web app, but it's also designed to be used on a desktop computer. A segment of Openbravo market is retail business, and for them, the company has designed a new interface for accessing the ERP's point-of-sale functionality from a mobile device. The approach has been to use adaptive design so the same application can be seen on any device. For this design, the company used the Enyo framework (<http://enyojs.com>). In Figure A1, we see a window when it's accessed from a device with a width of more than 800 pixels. In Figure A2, we see the same function when accessed from a smaller device. In this latter case, the user can change the view via touch gestures (or the upper corner buttons for nontactile screens).

Ludei is a company that develops gaming apps by using HTML5, CSS, and JavaScript. In this way, it develops a Web application as well as apps for different platforms. An example of this is the ibasket game app (www.ludei.com/games/ibasket), available for the Web, iOS, Android, Chrome, Facebook, and Amazon. The designers used the CocoonJS platform (www.ludei.com/tech/cocoonjs) to make it work like a native app.

Reference

1. N. Serrano and J.M. Sarriegi, "Open Source Software ERPs: A New Alternative for an Old Need," *IEEE Software*, vol. 23, no. 3, 2006, pp. 94–97.

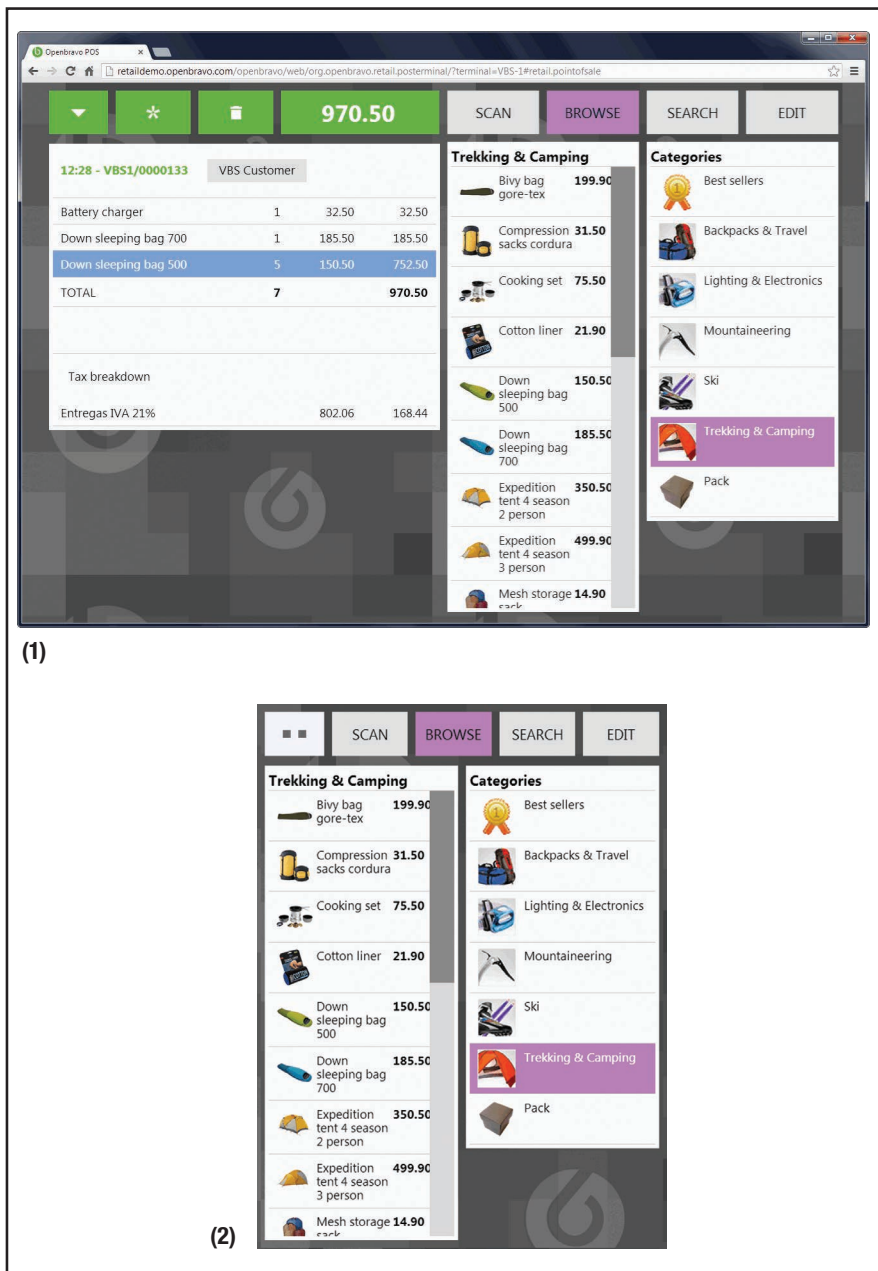


FIGURE A. The Enyo framework point of sale for (1) a browser and (2) a mobile device.

Docs) usually offer users the option of accessing a standard Web version rather than a mobile version; users will originally be directed to the mobile version, but will also see an option to view the standard version. Even though the standard version will result in a slower process for the smartphone, some users will choose it in order to have access to the site's full features.

Standard Web Apps

In this article, we use the term *Web applications* to refer to applications designed to work on desktop computer browsers. Essentially, they work with devices that have a browser. They can also work on mobile devices, on the condition that they don't rely on specific browser features that are unavailable on most mobile devices (such as Adobe Flash).

Responsive Web

Applications with a responsive Web design apply a different style (or set of styles, depending on device size) primarily based on cascading style sheets (CSS). The server can decide the design when serving the application, the design can be applied at the

Mobile Web

We use the term *mobile Web* to describe when there's either a specific website or logic execution for content to be delivered specifically to mobile devices. It usually has a better feel and touch than the responsive Web because it renders the user interface controls (such as buttons, selectors, and text-boxes) in a way that's similar to a native app. However, with mobile Web, there's still a need to maintain different sites.

Hybrid Apps

Hybrid apps are mobile Web applications packed into a native app. They behave like a native app in that they're installed from a Web store and have access to the same capabilities as a native app, but they're developed using the same tools used to develop Web applications—mainly, HTML5, CSS, and JavaScript.

Native Apps

The companies that build mobile OSs want apps that are specific to their own environments that can take full advantage of their particular features. This requires developing the app using that

fact that in addition to multiple existing OSs, new ones continually appear.

Technical Considerations

There isn't a single solution for choosing which mobile application type will be best for every situation. Several technical criteria can help us narrow down our options to select the most suitable development approach for a given situation.

Platforms and Version Support

First, we should consider the platforms and versions to be supported, including the range of devices, the development stack for each of them, the platform's browser capabilities, and our own development skills. If the goal is to build an app once with support for multiple platforms, a hybrid or a mobile Web approach is a better choice than a native app, which must be developed specifically for each mobile platform.

Device Capabilities

We also need to consider which device capabilities are needed. If the app needs access to the camera, a barcode scanner, the file system, or a Bluetooth peripheral, we're better off using a native or hybrid approach because they have direct access to these features. The latest browsers support hardware-accelerated animation properties, but they still can't fully use device capabilities such as native graphics or other device-specific interfaces included in the native device API.

User Experience

Later, we'll list different Web frameworks that provide libraries for mobile Web and hybrid apps to imitate native mobile interfaces and behavior. However, native apps still provide a richer, more compelling experience with a more responsive interface and superior interaction because users can open them faster and use device-specific

HTML5 is the standard
for Web development.

client level, or both can occur. The idea is to have a single source of content that renders differently based on a device's features. In addition to being a solution for mobile Web applications, it's also useful for other types of devices such as tablets, game consoles, and televisions.

provider's language and framework—for example, using Xcode with Objective-C for iOS and Eclipse with Java for Android. Thus, one project must be maintained for each OS, which, as you can imagine, leads to an increment in the development team, costs, and time. Moreover, developers must address the

TABLE 1 Criteria to consider when choosing a native, hybrid, or Web app approach.

Considerations	Native	Hybrid	Web
Effort of supporting platforms and versions	High	Medium	Low
Device capabilities access	Full	Full	Partial
User experience	Full	Full	Medium
Performance	Very high	Very high	High
Upgrade in the client	Needed	Needed	Not needed
Ease of publication/distribution	Medium	Medium	High
Approval cycle	Mandatory	In some cases	Not required
Monetization in app store	Available	Available	Not available

hand gestures. Moreover, mobile apps still face challenges in accessing a device's native features across all mobile browsers, which keeps users from achieving the same level of experience as when using a native app.

Hybrid apps offer an important trade-off in terms of the depth of experience. They allow the HTML code to have access to native APIs—although this comes at the cost of a nonnative user interface due to the Web technologies involved.

Performance

Performance is one of app developers' chief concerns. If the user interfaces are graphic-heavy or require excessive data processing, the mobile and hybrid app approaches have more difficulties in achieving high performance because they're running on top of additional layers, which consumes computing resources. At any rate, before completely developing any app, developers should test performance by using a preliminary prototype or testing similar existing apps.

Upgrade

In the native app development approach, developers need to keep in mind that they can't force app up-

grades, so they will likely be simultaneously serving different versions, adding complexity to back-end development and support. This also applies to hybrid apps when the code is in the local part of the app.

Nontechnical Considerations

Several nontechnical considerations can also help us identify the best solution for choosing a type of mobile application to develop. Table 1 summarizes both the technical and nontechnical criteria.

Distribution

Although mobile apps are easy to distribute, they can have discoverability issues, which means that users can have problems finding them outside the app store. In this case, a native or hybrid app is preferable. If we're targeting the consumer or gaming segments, our marketing expenses can be reduced considerably if we handle our distribution through the platform store. Nonetheless, as the volume of apps in these stores increases, efforts to gain visibility in the market are more important. If we need to limit the reach of our app—for example, if we're developing enterprise apps—we can use a private enterprise app store. However,

we must keep in mind that these stores have their limitations—for instance, we won't have the chance to influence the way the store is managed. Moreover, the store might have hundreds of thousands of applications competing for users' attention.

Approval Cycle

Mobile development can conflict with agile development methodologies. For an agile development approach, fast turnover and continuous user feedback are necessary. If we decide to follow the native or hybrid approach, we need to consider the approval process as part of our project. If there are no rejections, the approval time is usually minimal and the total time isn't long. There's also the possibility of having a license for a specific phone (for example, approval for only a specific version of an iPhone) or using a private enterprise app store (where you manage the site but don't have the visibility of the official store).

Monetization

In addition to the distribution advantage, if we base our development on the platform store, we can improve our conversion rates thanks to an easy, well-defined payment gateway. But

TABLE 2

Features of frameworks for mobile Web applications.

Frameworks	Target	Platforms					Hardware						Program- ming languages	License	
		iOS	Android	Windows Phone	Blackberry OS	Others	Camera	Accelerometer	Contacts	Gestures multitouch	Geolocation	Storage		Open source	Commercial
jQuery Mobile http://jquerymobile.com	Web, hybrid	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	No	HTML5, CSS, JavaScript	Yes	No
Sencha Touch www.sencha.com/products/touch	Web, hybrid	Yes	Yes	No	Yes	No	No	No	No	Yes	Yes	Yes	HTML5, CSS, JavaScript	Yes	Yes
jQT http://jqts.com	Web, hybrid	Yes	Yes	No	No	No	No	No	No	Yes	Yes	No	HTML5, CSS, JavaScript	Yes	No
Enyo http://enyojs.com	Web, hybrid	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes	No	No	HTML5, CSS, JavaScript	Yes	No
Lungo http://lungo.tapquo.com	Web, hybrid	Yes	Yes	No	Yes	Yes	No	No	No	No	Yes	No	HTML5, CSS, JavaScript	Yes	No
PhoneGap http://phonegap.com	Hybrid	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	HTML5, CSS, JavaScript	Yes	No
Titanium www.appcelerator.com	Native	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	JavaScript	Yes	Yes
Rhodes www.motorolasolutions.com/US-EN/RhoMobile+Suite/Rhodes	Native	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Ruby	Yes	Yes

these advantages come with a cost: the platform owner gets a significant chunk of the revenue (about 30 percent for iOS), so we need to carefully consider if we're better off outside of the platform.

Frameworks

Although it's possible to develop a mobile Web, hybrid, or native app without a framework,² using frameworks can simplify and reduce the development process effort. There are frameworks

available for just about any development option (see Table 2 for a list).³

Most frameworks are for HTML5 development. However, if we choose to develop for a hybrid or native approach with a cross-platform development frame in mind, we have different options that we should consider.

Native

We can develop a native app using a cross-platform approach but with just

one codebase for all devices. We can select our platform on the basis of our preferred development language—for example, JavaScript (Appcelerator's Titanium; www.appcelerator.com) or Ruby (Rhodes; www.motorolasolutions.com/US-EN/RhoMobile+Suite/Rhodes).

Hybrid

For a hybrid approach, Phonegap (<http://phonegap.com>) is the predominant technology and can be used with HTML5

frameworks. We might also consider Sencha mobile packaging (www.sencha.com/products/touch), which complements the Sencha development stack but only supports iOS and Android. Nonetheless, it could be a good option when developing with Sencha tools.

HTML5

To select an HTML5 framework for our project, in addition to considering device support, we would need to try each of the frameworks to choose the one that best fits our development practices.

There are plenty of options in the HTML5 framework landscape (see Table 2). There will be incremental improvements as more features are added and more devices are supported. However, this trend is making the framework's codebase bigger, which affects performance and bandwidth consumption. We expect to see some specialization in frameworks' capabilities, with different approaches to overcome this issue. (For example, JQuery Mobile can select just the necessary modules for a project.)

Although the mobile device market share is now essentially a duopoly of iOS and Android (smaller platforms on the market include BlackBerry, Windows Phone, and Symbian), the shorter life span of devices means that significant changes in mobile devices and mobile OSs can occur. One example is Firefox OS. This new platform is backed by major mobile network operators, and its development is based on HTML5, so it seems that it will be easier to attain the network externalities that any new device needs to attract a large enough customer base (apart from the compulsory Angry Birds app).

For native and hybrid apps, we always need to consider the risk that

comes with the lack of control over the platform. We can lose access to our consumers based on the platform owners' policies, which can unilaterally restrict access to our app. In the case of hybrid apps, this can be even more important because access restriction can be triggered if platform owners feel that their revenue streams are at risk. This was the case when Apple launched its App Store's subscription service for printed media, and major players replaced their native apps with HTML5 versions to avoid the loss of customer data and revenues (www.reuters.com/article/2011/08/31/us-apple-ft-idUSTRE77U1O020110831). It's possible that this situation could repeat itself and that each platform owner will ban all apps developed using a specific framework on the basis of economic or security-related issues.

There are different alternatives for developing mobile Web applications, but most of them converge in their use of HTML5. Individual developers and many companies say and show that HTML5 is the standard for Web development. As a new standard, a hot topic for developers has been its maturity. Although it might be early to bet on HTML5 as the go-to tool for developing mobile Web applications, it seems to be a safe bet. We as developers just need to master the technology we are using. ☞

References

1. Y. Chen et al., "Adapting Web Pages for Small-Screen Devices," *IEEE*

Internet Computing, vol. 9, no.1, 2005, pp. 50–56.

2. R. Nixon, *HTML5 for iOS and Android*, McGraw-Hill, 2011.
3. R. Rodger, *Beginning Mobile Application Development in the Cloud*, Wrox, 2011.

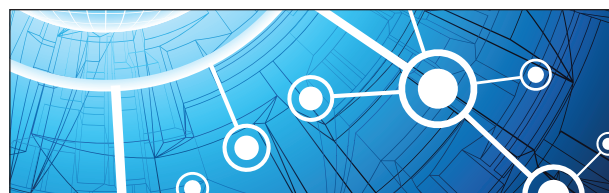
NICOLÁS SERRANO is a professor of computer science and software engineering at the University of Navarra's School of Engineering. Contact him at nserrano@tecnun.es.

JOSUNE HERNANTES is a professor of computer science and software engineering at the University of Navarra's School of Engineering. Contact her at jhernantes@tecnun.es.

GORKA GALLARDO is a professor of information systems at the University of Navarra's School of Engineering. Contact him at ggallardo@tecnun.es.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.



IEEE Open Access

Unrestricted access to today's groundbreaking research via the IEEE Xplore® digital library

IEEE offers a variety of open access (OA) publications:

- Hybrid journals known for their established impact factors
- New fully open access journals in many technical areas
- A multidisciplinary open access mega journal spanning all IEEE fields of interest

► Discover top-quality articles, chosen by the IEEE peer-review standard of excellence.

Learn more about IEEE Open Access
www.ieee.org/open-access



IEEE