

## CROSS-PLATFORM WEB DEVELOPMENT EXPLORING THE OFFLINE FUNCTIONS

Data de entrega dos originais à  
redação em: 14/02/2016

e recebido para diagramação  
em: 30/11/2016

Tiago Paiva de Castro<sup>1</sup>

Juliano Z. Blanco<sup>2</sup>

*Besides the quality and productivity, in many cases cross-platform is a requirement in software development. Therefore, it is necessary to consider the differences and limitations between devices, not only mobile but non-mobile as well. Moreover, the software developer should deal with details about adequate behavior to technological evolution, like offline function and data synchronization with the server. There are a few solution to consider in cross-platform development available on the market, some of them being: cross-platform hybrid development and responsive web development. Among the solutions for cross-platform development the responsive web presents the highest range with the lowest cost. However, mobile devices do not have access to Internet at all times. To ensure the use of the software even without connectivity, it is important to consider functions that can operate offline. Therefore the project aims to evaluate and quantify the mechanisms for local data storage and synchronization with the server to allow the use of web software in places without Internet connection.*

**Keywords:** *Cross Platform. Responsiveness. Local Storage.*

*Além da qualidade e da produtividade, em muitos casos a plataforma cruzada é um requisito no desenvolvimento de software. Portanto, é necessário considerar as diferenças e limitações entre dispositivos, não apenas móveis, mas também não móveis. Além disso, o desenvolvedor de software deve lidar com detalhes sobre o comportamento adequado à evolução tecnológica, como a função offline e sincronização de dados com o servidor. Existem algumas soluções a considerar no desenvolvimento de plataformas cruzadas disponíveis no mercado, algumas delas sendo: desenvolvimento híbrido multi-plataforma e desenvolvimento web responsivo. Entre as soluções para o desenvolvimento de plataformas cruzadas, a Web responsiva apresenta a maior gama com o menor custo. No entanto, os dispositivos móveis não têm acesso à Internet em todos os momentos. Para garantir o uso do software mesmo sem conectividade, é importante considerar funções que podem operar off-line. Portanto, o projeto visa avaliar e quantificar os mecanismos de armazenamento de dados locais e sincronização com o servidor para permitir o uso de software da web em locais sem conexão com a Internet.*

**Palavras-chave:** *Cross Platform. Capacidade de resposta. Armazenamento local.*

<sup>1</sup>System analysis and development

<sup>2</sup>Professor do Curso Técnico em Análise e Desenvolvimento.de Sistemas - Instituto Federal de São Paulo, Câmpus Piracicaba.

## 1 INTRODUCTION

Nowadays there are all sorts of mobile devices and highly available network to access the Internet, allowing software to be used virtually everywhere. Of all devices, the most utilized ones are smartphones, Tablets, Multimedia Centers among other ones. All of them being able to install and run software.

The constant growth of mobile devices had generates a great demand for cross-platform software, that being, a software system that can be used in various operational systems. This situation bring the developers some challenges, besides of quality and productive issues, the development team have to consider the individual characteristics of each device, like screen size and resolution or the internet connection system.

However, nowadays some solutions for these difficulties have being available: the native development, hybrid development and web with responsiveness. The native form corresponds to the development directed to each platform using a specific programming language, for example Java and swift being used for android and IOS respectively. This kind of development uses local databases as SQLite for persistency, which allows for offline use. The hybrid method consists in using some languages for web development, (HTML5, JavaScript and CSS3) automatically create versions for the main platforms, even the usage of SQLite is possible. However, the hybrid form can boost the productivity in comparison with the native form; both of them, the development team have to create a web version or a desktop app for the computer's users.

Web responsive development is also an alternative for the native and hybrid development methods, using HTML5, CSS and JavaScript. The term responsiveness (Marcotte 2010) was created in 2010 and is responsible of the software interface adaptation for different screen sizes, and orientation. These characteristics have made the web development the more cost effective option. However, it offers less mobility because the offline operations is an underexplored area.

Motivated by this idea, this project aims to evaluate and qualify the local storage mechanisms available for web applications. With the intention of enabling offline use of cross-platform software based in web, as already occurs in software developed through other forms of cross-platform development.

## 2 METHODS AND MATERIALS

To reach our goal in this research, we have defined a five steps method. As presented below:

First step was to research the available technologies for local storage in web development. So the web software have a similar behavior of hybrid and native apps in the offline use. These apps should use the caching concept. The caching system is a repository contained within the browsers to reduce data traffic on the network, the processing load of the servers and to allow that the web-based software can be operated without connectivity. There are some caching technologies that stores data on the client side. We can highlight some of them like LocalStorage, WebSql and IndexDB.

LocalStorage is the one with the higher compatibility with the browsers. The storage consist in the content to be stored and a key to identify it. The total capacity, given by W3C is

5 MegaBytes per site, allowing only text data. The IndexedDB is an API focused on data storage based on content and key as well, but it differs from LocalStorage because it accepts different kinds of data like: float, string, int and array. WebSql is another HTML5 resource for storage, but it was discontinued by W3C.

For this work's purpose, the chosen technology was the LocalStorage, because of browser compatibility, the great amount of documentation on the Internet and provided by W3C. In addition, it is easier to implement, improving the productivity and lowering error occurrence.

The third step was the study of LocalStorage official documentation provided by W3C. Allowing is to confirm that this technology did not provide a mechanism to synchronize automatically with the server. Therefore, an implementation of a data communication structure was necessary, along with a function to identify the internet availability. To do this, the JavaScript function "navigator.onLine" responsible for connectivity monitoring was used. And another function was developed for sending the data, stored on the client side, to the server as soon as a connection was detected.

The definition of the project scope was the fourth step. The software to be developed consists in a web app to create and store video game events, with some informations for example, the date and time, title of the game and platform to be used (Consoles or PC). One of the functions is to allow these events to be create in mobile devices or a computer regardless of Internet access.

The fifth step was the technology viability analyses with experimentation. Some of the criteria were the browser compatibility, storage capacity and syncing aspects like safety and stability.

### 3. RESULTS

The experiments were done in personal computers and mobile devices with a few of the more popular browsers in their latest implementation. Windows Linux and Mac were used on the desktop tests an Android 5.1 device and an iPhone with iOS 8 were used on the mobile tests. The Table 1 and 2 show the results of the compatibility test, which were vastly important because allowed the viability and use restrictions to be accounted.

Most browser are compatible with LocalStorage. Therefore, it is a viable solution as for desktop as mobile use. The data storage in cache as defined by W3C is 5MB, allowing up to 421 entries of the used model (event registration). This quantity considered sufficient for its purpose.

The data stay on the client as long as the developer allow it, by using the JavaScript parameter "max-age=#" followed by the seconds of permanency. The data synchrony is done by the following formula: to the first access the user have to have internet access, after that in case the device is offline all the entries are saved on the device memory. Right after the device software detects an Internet connection the events data are saved on the server where the registered users can participate of the event.

Browser	Version	Compatibility with WebStorage	Browser	Version	Compatibility with WebStorage
Google Chrome	45.0.2454.85 m +	compatible	UC	10.7.0.636	compatible
Mozilla FireFox	40.0.3 +	compatible	Dolphin	11.4.20	compatible
Opera	31.0.1889.174 +	compatible	Opera	30.0.1856.95530	compatible
Microsoft Internet Explorer	11.0.10240.16431 +	compatible	FireFox	40.0.3	compatible
Microsoft Edge	20.10240.16384.0 +	compatible	Opera Mini	11.0.1912.95711	non compatible
Apple Safari	8.0.7 (10600.7.12) +	compatible	Safari	ND	non compatible
			Google Chrome	45.0.2454.84	Compatible
<b>Table 1: Desktop browsers</b>			<b>Table 2: Mobile browsers</b>		

## REFERENCES

B. Leroux A. Charland. “**Mobile application development web vs. native**”. in: (2011), pp. 49–53.

MARCOTTE, Ethan. **A List Apart, Responsive Web Design**. Retrieved June, v. 9, p. 2013, 2010.

CAELUM **O que é Java**. 2010. Available em: <http://www.caelum.com.br/apostila-java-orientacao-objetos/o-que-e-java/#2-3-maquina-virtual>. Accessed: 8/3/2015

Apple. **The Swift Approach**. url: <https://developer.apple.com/swift/blog/>.

BIDELMAN, Eric. **Guia Básico sobre o uso do Cache do Aplicativo**. 2010. Available: <<http://www.html5rocks.com/pt/tutorials/appcache/beginner/>>. Accessed: 6 may 2015.

KITAMURA, Eiji. **Working with quota on mobile browsers: A research report on browser storage**. 2014. Available: <[http://www.html5rocks.com/en/tutorials/offline/quota-research/?redirect\\_from\\_locale=pt](http://www.html5rocks.com/en/tutorials/offline/quota-research/?redirect_from_locale=pt)>. accessed: 02 abr. 2015.

Kushal Mehta e Jasmine Jha. “**Web Cache Technique Responsive Web Design**”. In: (2014) Suyesh Amatya e Arianit Kurti. “Cross-platform mobile development: challenges and opportunities”. Em: ICT Innovations 2013. Springer, 2014, pp. 219–229.

MEHTA, Nikunj et al. **Indexed Database API**. 2008. Disponível em: <http://www.w3.org/TR/IndexedDB/>. Acesso em: 22/07/2015

KESTEREN, Anne van; HICKSON, Ian. **Anne van Kesteren**. Offline Web Applications. Disponível em: <http://www.w3.org/TR/offline-webapps/>. Acesso em: 10/3/2015