Platforms and Graphication Tools for a mobile application: Simulation of Planetary Objects' Trajectories

Analuz Silva Silverio[1,*], Mario A. De Leo Winkler[2], Antonio Alarcón Paredes[1]

[1]Universidad Autónoma de Guerrero, Unidad Académica de Ingeniería, Chilpancingo, Mexico
sisianlight@gmail.com, aalarcon@uagro.mx
[2]University of California, Physics and Astronomy Department, Riverside, USA
mario.deleo-winkler@ucr.edu

Abstract. During the development of mobile applications some of the first things that should be taken under consideration are the platform, code and development tools to be used. To do this, it is important to keep in mind compatibility of devices and operating systems, the focus audience, the project's budget, and other factors that end up being crucial for the successful completion of a project of this type. Because this particular case we want to develop applications for scientific outreach in astronomy, we investigate and present trends in developing mobile applications as well as propose a number of tools for our particular project, themed on planetary objects in a bi-dimensional simulation of positions, as well as the charted and tabulated data.

Keywords: Physics computing, Mobile computing, Native app, Application software, Open source software, Software tools, Programming environments, Computer science education, Astronomy.

1 Introduction

In mobile applications' (apps) development, one of the first things to take into account is the type of application that is being developed. For this reason, it becomes necessary to know positive and negative aspects in the trends of apps development.

Nowadays, the guidelines of app development have a wider scope than the initial ones where only two paradigms existed: native apps and web apps. Native applications are developed on the native language of an operating system (OS). A web application consists of a website specifically optimized with an interface and a set of functions to be used in mobile devices. Recently, with the arose of HTML5, the concept of hybrid applications has emerged and comes with added features and functionalities related to direct hardware control and the possibility to be implemented in different OS. Those new features have made hybrid apps become a new trend of development.

We analyzed the possibility of having a greater exposure of astronomy through the use of mobile apps for scientific outreach where we reviewed currently available astronomical apps for devices running the Android OS and we defined through that review the astronomical themes that needed further development or improvement [1]. It is worth to mention that there are no works with a similar intention in Spanish and through this, we contribute to the creation of a scientific outreach tool for observers, amateur astronomers and public education in general.

The final objective of selecting a specific kind of tool for development is to create a set of apps focused to create outreach for astronomy in Spanish, under a GNU license, completely free charge for users and an open source code commented for its future use or improvement among the Spanish-speaking population interested in this subject. Our app will simulate the position of the major moons around the gaseous planets in our Solar System; such a simulation serves three purposes: it creates a useful tool for amateur astronomers to plan their telescopes/binocular observations, our app is paired with important information on the satellites and planets themselves creating a educational tool, the source code is commented and open sourced so it can be used as an educational tool for programmers of astronomical apps.

In this paper we discuss some of the trends in platforms and graphic tools for the development of mobile apps. The paper is organized as follows: Section 2 has a review of apps development guidelines, in which we explain a research we made for selecting the IDEs, frameworks, trends on graphics building, charting libraries, and the arguments of choosing the ones we used. In the Results Section, the products of using those tools are shown. Section 4 deals with the discussion of the proposed work, and finally, the conclusions and references are presented.

2 Method

We must mention that in every cases compatibility issues could occur, mainly related to the hardware of the device and the OS version, which is also a latent issue when developing apps.

2.1 Review of the Context on Mobile Applications' Development Trends

Describing the main characteristics of the paradigms of the guidelines of apps development is important for based on that information be able to select one of them.

Web applications. This type of apps properly constitutes a website, the programming languages are the same used in common websites (e.g. HTML, CSS, and JavaScript (JS)), so the content could be anything supported on a website. It can be reached by an URL in a web browser and optimized for hardware specifications of mobile devices [2]. Since a

browser allows the creation of a website shortcut on the desktop, the web apps are not truly installed, so the "installation" step is the creation of a shortcut.

Adapting a website to a mobile format is a good solution in many cases, although there are some drawbacks such as the lack of accessing the device's hardware, publishing on official application stores (at least before HTML5) and the poor response time and user experience in contrast with common web navigation.

Native applications. These applications are developed on native languages of each operating system (OS). The programming languages for this purpose are Java (Android), Objective C (iOS), C# and Visual Basic .NET (Windows), and C++ (BlackBerry) [3].

Nevertheless, native apps are the option with the best implementation [3], it depends on a good app design at all levels; according to the above, the development of native apps may grant full access to mobile device hardware, push notifications, publishing the app at Google Play, App Store, or Windows Phone Store [3], and data synchronization for offline use.

The drawbacks are mainly found in economic and time consumption terms, such as maintenance, functionality and OS updates which are frequently required. In addition, the impossibility to reuse code among different OS programming languages makes necessary that a single developer needs to know different programming languages or to have a bureau of skilled programmers for each OS and for some less-used languages among developers (compared to web languages).

Hybrid applications. For developing a hybrid application, a web programming language-based framework (such as HTML5, CSS3 or JS) is needed. In these apps, the OS integration level depends on the selected framework, and the OS itself. These apps let the developer have access to the hardware of the mobile device and, often, to some OS libraries. However this method has still not reached the same response time and experience as native apps. Nonetheless, they have been evolving to more complex processes where, in some cases, they could act like a web app and, in other cases, natively.

Hybrid applications are a very economic option and could be very useful to get a great number of users in a variety of different platforms and devices.

2.2 Selection of the Type of Application

In order to develop an astronomy app, and due to the highest percentage of users it has, the Android OS could be selected. But if we choose to develop a native app, it would run only on Android devices [1].

On the other hand, the major problem with web and hybrid apps is their high energy consumption when calling JS, Canvas, and some other graphical resources. However, this

drawback is overcome by the automatic release of the resources while the device is not using the application.

So, there is no clear advantage between the native and hybrid apps, but the trend is to use even more hybrid HTML5 apps over the native ones. Nonetheless, it is important to note the remarkable use of HTML5 in app development, as can be seen in Fig. 1; it is one of the preferred platforms for developers due to its versatility, new features and capabilities [4].

The selection of the application type has to do with the app's functions, and the use we must give it. Based on that, in this project we chose to develop a hybrid app, since the graphical resource consumption to represent a planetary simulation will not be as exhaustive as for a game app, for example. On the other hand, since hybrid applications have become easy to develop due to the use of some tools such as PhoneGap Developer App [5], we think the development time and resources needed, as well as the project costs will be minimum. Moreover, the app will not be limited to Android users only, and the contents could even be part of a website, making our app a wide and very helpful tool for scientific outreach.

2.3 Selecting the Integrated Development Environment (IDE)

Most of the IDEs have a smart code autocomplete function, a class navigator, object finder, class hierarchical diagrams for object-oriented software, and probably also an integrated version control system.

For a long time, one of the tools most used for developing web pages (and now apps) was Adobe Dreamweaver, but its use requires a purchased licence.

Aptana Studio is an alternative for developing websites that has become very popular. It is an open source IDE based on Eclipse, which can run under Windows, Mac and Linux, provides support for languages like PHP, Python, Ruby, CSS, Ajax, HTML, Adobe AIR, and the use of plugins for new languages is also possible. Its main characteristics are a code assistant for HTML and JS, Ajax libraries (jQuery, prototype, scriptaculous, Ext JS, dojo, YUI, and Spry, etc.), data base tools, syntax verifier, compatibility with Eclipse plugins or extensions (more than 1000).

Thus, Aptana Studio seems to be a very versatile IDE for the needs of our project. It also has a very good compatibility with most used frameworks for app development. So that, in this work we select Aptana Studio as our IDE.

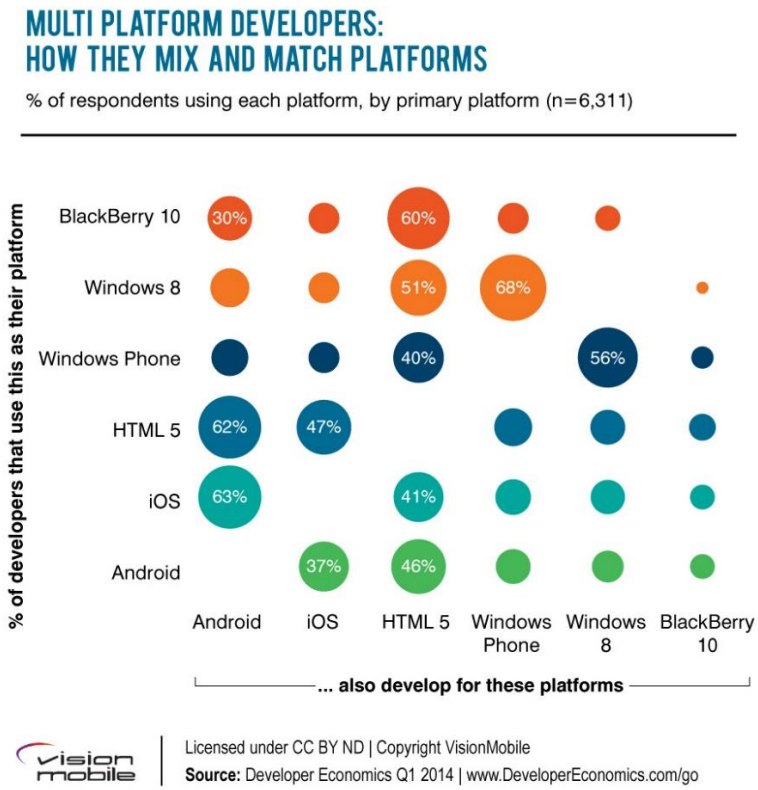

Fig. 1. Multi Platform Developers: How They Mix And Match Platforms. The third column shows how HTML5 is one of the preferred platforms for developers to match with other platforms due to its versatility, new features and capabilities [4].

2.4 Frameworks and Libraries

Seeking among the most used frameworks to create apps for Android and iOS with HTML5 [6], we found those for developing web apps and hybrid apps that require no financial cost to use or it is not requested unless the developer will profit from the end product. Those frameworks are:

Chrome Dev Editor. Created by Google to develop web apps and Chrome Apps, it can be used from devices using Chrome OS (Chromebooks) or the Chrome browser. This allows to create projects with templates such as Web JS apps and combining them with Polymer, Dart apps and Chrome Apps, for desktop and mobile, which are easily executable in Android and even packable in an executable Android application package (APK) or

Web Starter Kit. It includes Dart compiler and launches an embedded service to implement projects from the local machine. It is also able to directly upload the Chrome Apps to the Web Store [7].

PhoneGap. Being one of the best known, PhoneGap is the preferred framework, and also, the one that concentrates the greatest number of developers worldwide [8]. Based on Apache Cordova license, it allows packaging code as an application for iOS, Android, Blackberry and Windows Phone. PhoneGap allows programming through its API native SDK functionalities. For distribute a project among different platforms, a native application for each one is built with the help of Phonegap, and using the same source code it generates the executable for the different OS [6].

Icenium. This framework removes the concern of each platform's SDK. Through its own IDE creates apps using web standards (such as HTML5 and CSS) and publishes them directly to the app stores through a simple step by step. This is the main advantage of Icenium compared to PhoneGap. Due to its environment Icenium Mist can be accessed from any browser and makes the publication by selecting a desired platform (Android or iOS). Unlike PhoneGap, this framework is not available for free [6].

Appcelerator. Just as Icenium, it has its own IDE, allowing code common functionality for different platforms. One of the most interesting qualities of Appcelerator is the easy connection of applications with different data sources using their customized connected services with node.js. Although it is required to work within the platform and use its own functionalities. Appcelerator offers features that easily integrate information from multiple services into applications [6].

For the development of this project we chose to work with PhoneGap and its direct tester app "PhoneGap Developer App" [5], as it proves easy to work offline, and to test applications without internet access. Because offline use is one of the objectives of this work, the possibility to use them in the middle of the field without suffering problems in the use of the application due to the absence of internet. On the other hand, as shown in Fig. 2, PhoneGap is the most sought tool in this tenor. It should be mentioned that if a migration to other frameworks is required, this is possible because they work under the same type of scheme.

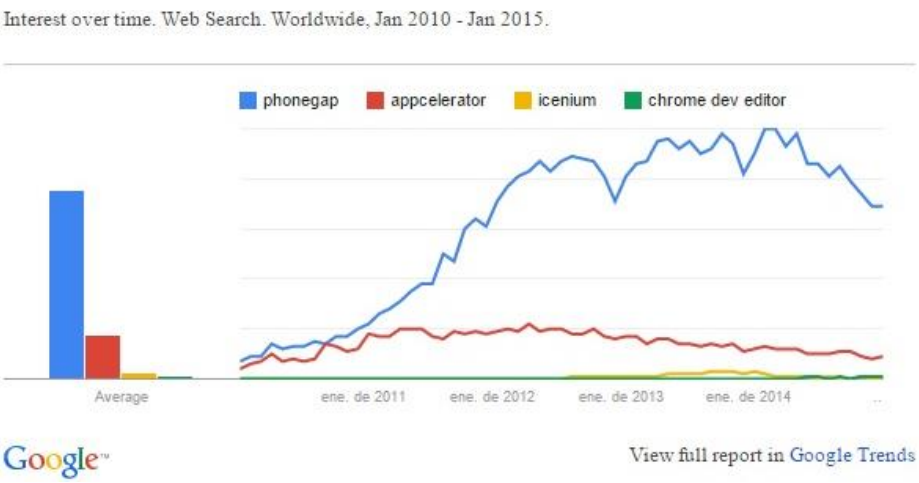

Fig. 2. Interest over time of frameworks for hybrid applications development [9].

Once we have chosen the framework we proceed to review trends of web graphics usage. HTML5 Canvas, SVG and Adobe Flash are trendy in Web graphic representations today. Arising from the creation of HTML5, Canvas, rendered through a bitmap, requires a single element to be added to the DOM, the canvas element in HTML5. Hence, we can use JS to access the context and then draw things. This offers a wide variety of options to be deployed, without making a page or app much heavier, making this element one of the most innovative resources of HTML5 [10].

Since SVG and VML require DOM elements for each form added to the document, the model document can quickly overflow for complex drawings, and the application can become slow or even crash. While Canvas is relatively new, it has proved to be very efficient and have a better performance mainly because its strongest point is the independence of complements for its handling. Then, if an application is rendered in 2D, it needs to be fast and does not require direct events on the canvas from the user interface, it is recommended to create an app using Canvas [10].

We have found several libraries to create charts, but few of them handled Canvas as its deployment element and, among those offering this option, few had a wide variety of charts or configuration possibilities. Through searches we performed, we found Highcharts was among the most popular and complete of the libraries. Highcharts is a graphics library written in JS, which provides an easy way to add interactive graphics to web sites or web applications. For this project we opted to use this library as it covers our developing needs with the following features [11]:

- It works in all modern browsers, on iOS and Android; it provides multitouch support and it is free for non-commercial uses.

- It is solely based on native browser technologies and doesn't require client side plugins like Flash or Java, it allows downloads of the source code and to make your own edits and offers a simple configuration syntax.
- Through a full API it can add, remove and modify series points, or modify axes at any time after chart creation and all options can be set individually, including reversing, styling and position, additionally, has datetime axis and offers inverted chart or reverse axis. Upon hovering the chart can display a tooltip text for information on each point following the movement of the mouse over the graph.
- Users can export the chart to PNG, JPG, PDF or SVG format at the click of a button, or print the chart directly from the web page.

3 Results

For the development of this project we took some decisions related to specific requirements trying to give the astronomical theme we selected the widest possible exposure, for instance: free distribution, have the opportunity of development by a single developer and develop within a limited budget. During the development we verified that the simulations building, graphics in HTML5 Canvas and the Highcharts library offered the versatility needed for this project.

Fig. 3. Result of the bi-dimensional simulation of Jupiter made in HTML5 Canvas. In this capture the moons of Jupiter can be seen given a particular date in the text field below together with controls for changing the positions and speed of reproduction of animation.

In the deployment of the developed simulation, Canvas includes several manipulable JS functions allowing freedom of handling and an outstanding design. The functions used to generate the positions of the planetary objects were based on a code made in HTML [12], and the algorithms were explained in the book "Astronomical Algorithms" [13]. As shown in Fig. 3, we used buttons, sliders and text fields to redraw the Canvas element as the data is taken automatically from the system date or manually from a date entered, as well as the activation and de-activation of the animation and the increase or decrease of the speed of simulation. The way in which the positions of the objects are drawn just re-

quired to load the data in advance in the page before calling the canvas element so that no errors occur during the deployment.

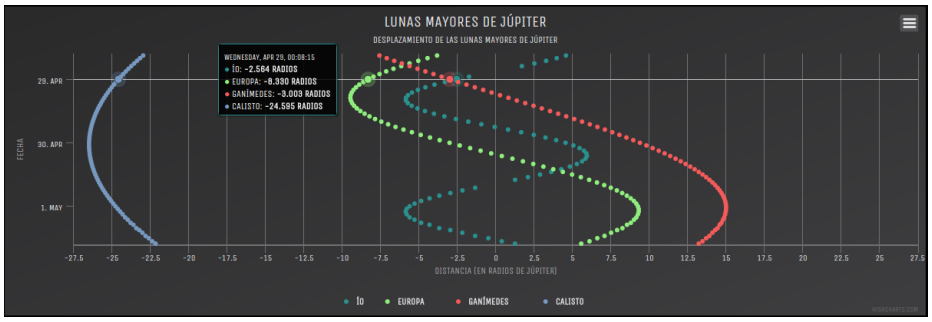

Fig. 4. Result of the chart made with Highcharts. In this image we can see a graphic generated for the positions of the trajectory of the Galilean satellites for the following 72 hours.

Highcharts, offered the possibility of translating charts with inverted axes required to capture the sinusoids formed by the changing position of the satellites with respect to time, with high quality graphics design, and a range of settings for units and options in the design of the chart, which is well adjusted to the needs of the project. As seen in Fig 4, the deployment was adjusted to the design needs of the project. The axes are inverted, a recurrent representation in apps that present satellites trajectories such as the app "Moons of Jupiter" [14]. Among the options worked in Highcharts was the setting of time units for showing the displacements during a time span of three days. Unlike other charts we tested, this one allows to draw points corresponding to each satellite, even when lacking the data to complete the sequence. This is important when a satellite is not visible (when it passes behind the planet), this allows the discontinuity in the display, making it a visible educational tool in the chart.

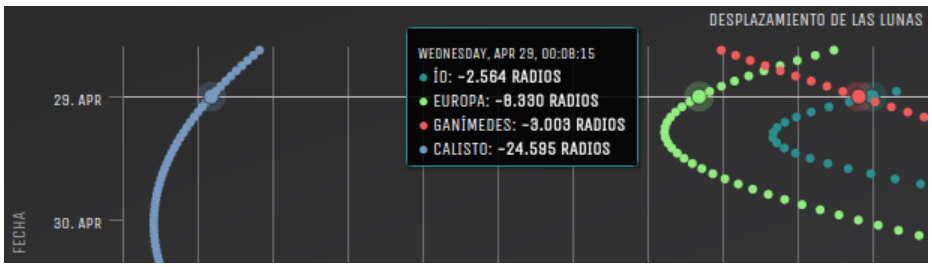

Fig. 5. Labels and points in the chart created with Highcharts. Upon hovering, the chart can display a tooltip text with information on each point and series.

Highcharts also allowed us to show dynamic labels with the name, time and location of each satellite. The tooltip follows the user's moves of a mouse or touchpad over the graph, it sticks to the nearest point and makes it easy to read a point that is below another one. We can also configure the shape, size or color, as shown in Fig 5.

Another of the most interesting options is the generating if an instant chart capture in PNG, JPEG, PDF, SVG format or send it to print, as can be seen in Fig 6. This useful feature for users to save the chart at any given time, either for an interesting event reflected in the graph, or to save a record of the positions movement for various ends.

Fig. 6. Menu of options of capture in the chart. Users can export the chart to PNG, JPG, PDF or SVG format at the click of a button, or print the chart directly from the web page.

4 Discussion

From the results obtained during the time that we have been working with these tools, we can consider our choice of tools as appropriate. The search done among available tools yielded solid arguments to decide in favor of the best for graphics and simulations. The choices were tested and configured to meet the needs of our project. Some tools offered unexpected features that were quite useful and enriched the nature of the application, for example, capture options on charts generated by Highcharts.

Better tools are available, but they require some sort of license payment to use and this project is budget-restricted. Our selection has secured the best results for a public and free application which is focused in offering an educational tool on astronomy for the general Spanish speaking public.

5 Conclusions

For this project we choose some of the best free tools and open resources. By these and the publishing of our code under GNU license, we assure the possibility for its future use or improvement among the Spanish-speaking population interested in this issue by providing them our source code. By using these tools, we can conclude that this work provides useful features and solutions to the needs of the project, producing satisfactory results in order to develop a free app of the major moons of Jupiter, Saturn, Uranus and Neptune

and only fewer problems were encountered. We are creating more tools for outreach/education in astronomy that are not currently available in Spanish (the app and the open source code) which we hope will capture the interest of enthusiast, amateur astronomers and the general public.

References

1. Silva, A., De Leo, M., Cuevas, R.: Análisis del actual desarrollo de aplicaciones móviles sobre astronomía para dispositivos basados en Android. Tlamati Sabiduría. 5, 96–103 (2014).
2. Stark, J.: Building Android apps with HTML, CSS, and JavaScript. O'Reilly, Sebastopol, CA (2010).
3. Tipos de apps: nativas, híbridas y Web Apps, http://appio.es/tipos-de-apps/.
4. Voskoglou, C.: Getting your priorities straight, http://www.developereconomics.com/report/q1-2014-getting-your-priorities-straight/.
5. Brooks, Michael: PhoneGap Developer App, http://phonegap.com/blog/2014/04/23/phonegap-developer-app/, (2014).
6. Prego, I.: PhoneGap, Appcelerator e Icenium: Crea apps para iOS y Android con HTML5, CSS y Javascript | ivanprego.com, http://ivanprego.com/diseno-web/css/desarrollar-aplicaciones-para-ios-y-android-con-html5-css-y-javascript/, (2013).
7. Rodríguez, T.: Chrome Dev Editor, el IDE de Google para programar desde Chrome, http://www.genbetadev.com/herramientas/chrome-dev-editor-el-ide-de-google-para-programar-desde-chrome.
8. Michalakos, D., Vakulenko, M., Constantinou, A.: How can HTML5 compete with Native?, https://www.developereconomics.com/reports/can-html5-compete-native/.
9. Google Trends - Web Search interest - Worldwide, 2004 - present, https://www.google.com/trends/explore?q&hl=en-US.
10. Rowell, E.: Web Graphics Trends in 2013, http://www.html5canvastutorials.com/articles/web-graphics-trends-in-2013/.
11. Highcharts product, http://www.highcharts.com/products/highcharts.
12. akkana/scripts, https://github.com/akkana/scripts.
13. Meeus, J.: Astronomical Algoriths. Willmann-Bell, Inc., Richmond, Virginia (1998).
14. Moons of Jupiter - Aplicaciones de Android en Google Play, https://play.google.com/store/apps/details?id=pl.bizcalc.jupitergalileanmoons&hl=es.