State of the Tech Report

Bruno Cardoso
bruno.f.cardoso@tecnico.pt
ist172619

Abstract

The purpose of this report is to analyze and to detail the different available options on mobile development. This research will assist on choosing the best work tool for the final project.

1 Introduction

WebApp, Native, Hybrid The purpose of adding the Web Application concept is just to properly understand whether it should be taken into consideration. From previous researchs, using different Apps, these apps have evolved from some basic responsive websites, to fully developed Mobile Applications, where most of the times an "expert level user" is unable to distinguish between a web, native of hybrid. Due to this, the WebApp solution will be further analysed during this report. Building a Native application is what every developed should do, as these are the favourite option in terms of speed and hardware usage, yet, they usually tend to take the longest to develop, and are the hardest to update. This poses as an extra problem when there are 2 different major platforms to develop. Lastly, Hybrid applications are those that are developed using a framework, that generally use/allow standard web technologies like HTML, CSS and Javascript to pack and deploy an application into a mobile device. These are then uploaded to the different Application Stores like Apple Store of Play Store, using a webview approach at runtime.

https://mlsdev.com/blog/56-how-to-choose-the-right-development-platform-for-a-mobile-app

https://www.logicroom.co/how-to-choose-the-best-mobile-app-development-framework/

https://techbeacon.com/approaches-platforms-tools-sorting-out-your-mobile-development-options

http://www.businessofapps.com/guide/cross-platform-mobile-app-development/

2 The final goal

The application itself The results from this research will determine the best option for this project. Since the project is at a very early stage, there are a lot of requirements to be discussed. This adds extra difficulty to the task, as it is a major decision, that is hard to change at a later stage of development. As I write this report, there is only a simple idea of the requirements for the application, such as the ability to collect data using Passive Wifi Tracking, to be "energy efficient" and not drain batteries, but also to be fast and easy to use, although this last one really depends on the implementation itself more than the means to it.

3 Things to consider

Target Audience there are 2 large groups for the target audience, the first one are the students, for the summer ticket that has an exclusively low price for those that are studying in Funchal. The other group is the one for the Tourists. Different requirements, same goal.

Supported Devices This app will mainly be available for the majority of Mobile devices, as there is a possibility of developing a web application, browser acessible. Although, from my understanding, the most valuable platforms still are Android and iOS.

Access current data ? This is just an extra idea, but, since this project includes a big data component, the app should be able to access external data using several api's, as long as this is not a deal breaker in regards of speed or data consumption.

4 Cross-Platform Development Tools

Xamarin Every one in the industry is familiar to Xamarin. The platform has recently received a $54m investment, expanding its platform and reach daily. Xamarin is one of the leading solutions for cross-platform development. Xamarin is definitely one of the biggest players for mobile development today. Some of its key features include native user interfaces (keeping the user familiar to what would be the result of an app natively developed), platform specific capabilities using a native API, where it would be easier to add platform specific feature like the iBeacon and Android Fragments. Then there are tools to share the final result into pretty much all mobile development platforms, and there is an option to keep the application up to date using an IDE and their Xamarin Component Store. Lastly, it allows backend integration like Microsoft Azure, Parse and SAP.

PhoneGap PhoneGap is another well known cross platform development brand. It is now property of Adobe, and it is based on the Apache Cordova project, an open source project.

PhoneGap still is free nowadays, while Adobe is developing an enterprise version of it. This is still in beta stage, and it will presumably be payware. While Xamarin is a platform, with a suit of different tools that work with each other, PhoneGap works in a different way, as developers have to write the application using HTML, CSS and JavaScript, then deploy it across all platforms. Just as Xamarin, PhoneGap deploys without losing the features of a native app. Is is a cloud based developer tool that is built on top of a framework, that includes all of this, without requiring SDK's or Compilers. Another advantage of using a cloud-based service like this one is that all apps are developed using the most current SDK for each platform. In regards to the platforms, PhoneGap also includes Windows Phone and webOS. Lastly, it includes a dedicated desktop app, and it has a big community with third party tools that might be handy at a later stage of development.

5 Native development

Native development would always be the best option, but it lacks on several different areas. For a start, using a Cross-Platform Development Tool, in order to update the application, one would only have to update one, and then deploy to the different platforms. Then, there is the learning curve for each paltform, together with the different number of platforms to deploy, choices would have to be made, where one would have to choose to deploy to less platforms, and to learn different programming languages, as Android is mostly Java, while iOS is now using its own language Swift. If we would like to include WP, then C* would be another language to learn. This, for now, does not seem to be my best option.

6 Unity

Unity, in regard to mobile development, is something really new for me. Apart from finding out that a couple of very interesting mobile phone games are fully developed using unity, the only time I was in contact with Unity was for the Desktop games and applications. The main advantage is that Unity has a really good engine, both in regards to Graphics, Physics and scripting, as it is a really well developed tool. Not being a Cross-Platform development tool, Unity is developed over the basic SDK tools for iOS and Android, meaning that it would require special attention for the mobile deployment at the end. I find unity to be a really interesting tool, but at the same time, I do believe that it is more focused on a fully gaming app. What I mean is, the final app will have gamification, but will not use most of the tools that Unity offers.

7 React Native

This was a personal recommendation from a friend that previously developed for mobile using React Native. React Native is developed using JavaScript and React. It shares the design of React, and the UI is composed from declarative components. The final app is a real mobile application that is completely different from an app developed using Objective-C, Swift or Java. It also uses the same "native" building blocks of iOS and Android. Another big advantage is that React Native combines native code from the applications, meaning that when required, if a certain component is only available for a certain platform, it is convertible to the application. Lastly, although not really important for the final decision, React Native is already used by Facebook, Instagram or even Tesla, for their mobile apps.

8 MapBox

MapBox is not a development tool, yet, it might be an essencial part of the final application, at it is highly customizable, versatile, as it includes a complex API and SDK, but also as the final result is appealing. Definitely a platform to incorporate on the final application.

9 Decision?

The question mark at the title of this section details that, although there are several very good development tools, without specifying a very basic sketch of the requirements for the final application, I am unable to make a decision that best suits the final Application. With this in mind, and after researching the previously mentioned tools, I am very keen on further exploring React Native, for its high versatility, easy to adapt native code for each app, and good community/documentation.

As mentioned, I still have to further research, and to discuss the possibility of using React Native for the project with Prof. Nuno and Prof. Catia, but at the time I write this report, I do believe it is a good development tool, and a very recent one that uses new technology.