

Multi-Platform Mobile Application Development Analysis

Lisandro Delía

Nicolás Galdámez

Pablo Thomas

Leonardo Corbalán

Patricia Pesado



Agenda



1. **Introduction**
2. Multi-Platform Mobile Applications
3. Case Study
4. Conclusions
5. Future Work

Introduction

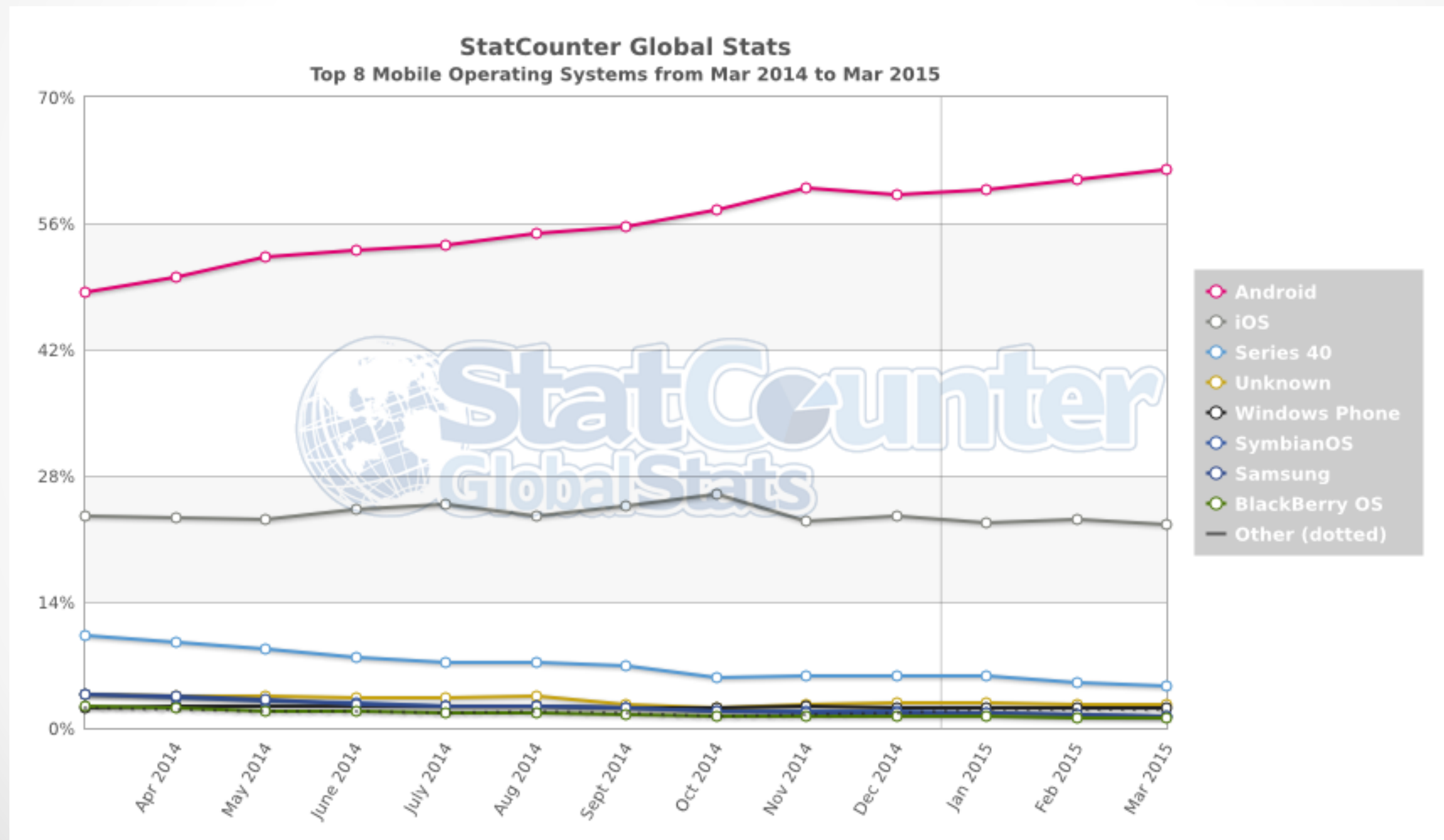


- The rise of mobile devices in recent years has an impact on almost every aspect of our lives.
- There is currently, a large number of operating systems:
 - Android
 - iOS
 - Windows Phone
 - BlackBerry - RIM
 - Firefox OS
 - Tizen
 - Ubuntu Touch OS
 - etc...



Introduction

Top mobile operating systems in the world





Introduction

The natural scenario is developing **native** applications

- Development is oriented towards a particular platform.
- Full access to device capabilities: GPS, camera, accelerometer, etc.
- Native user interfaces.
- There is no reuse of code between platforms
- If we need to target to more than one platform, will be expensive:
 - the development process
 - the maintenance process
 - the distribution process

Agenda



1. Introduction
2. **Multi-Platform Mobile Applications**
3. Case Study
4. Conclusions
5. Future Work

Multi-Platform Mobile Applications



In this paper,

- current multi-platform development approaches are analysed,**
- their most significant features through a case study are discussed.**

Mobile Web Applications



- No installation is necessary. The device's browser is used. Internet connection is required for operation.
- Updates can be performed at any time and be available immediately.
- The development process is simple: it uses standard technologies such as HTML, JavaScript and CSS.
- Less attractive and simpler than a native application.
- Response times are lower due to client-server interaction.
- It is not possible to access to all device features.

Hybrid Applications



- It uses web technologies (HTML5, JavaScript and CSS) as mobile web apps, but with the ability to have full access to device capabilities.
- Hybrid apps are hosted inside a native application that utilizes a mobile platform's WebView.
- They can be distributed through app stores for installation.
- Greater performance than a mobile website but slower than a native.
- The user experience is affected by not using native components on the interface.

Interpreted Applications



- The application code is mostly translated into native code while the rest is interpreted at runtime.
- They are implemented independently of platforms using different technologies and languages, such as Java, Ruby and JavaScript, among others.
- Native user interfaces are obtained.
- On the downside, there is a total dependence on the development framework.

Applications Generated by Cross-Compilation



- The projects are compiled natively creating a specific high-performance version for each target platform.
- The user interface uses native controls on each platform, creating apps that are indistinguishable from a native application.
- It is possible to find different levels of code reuse between platforms, depending on the development environment used (approximately 75%).

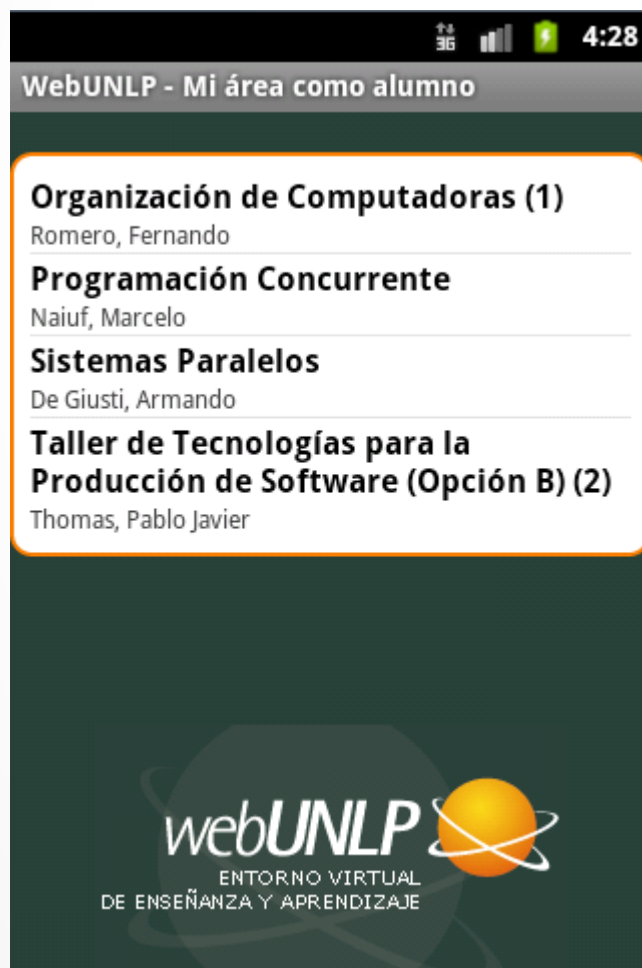
Agenda



1. Introduction
2. Multi-Platform Mobile Applications
3. **Case Study**
4. Conclusions
5. Future Work

Case Study

Native Application (Android)

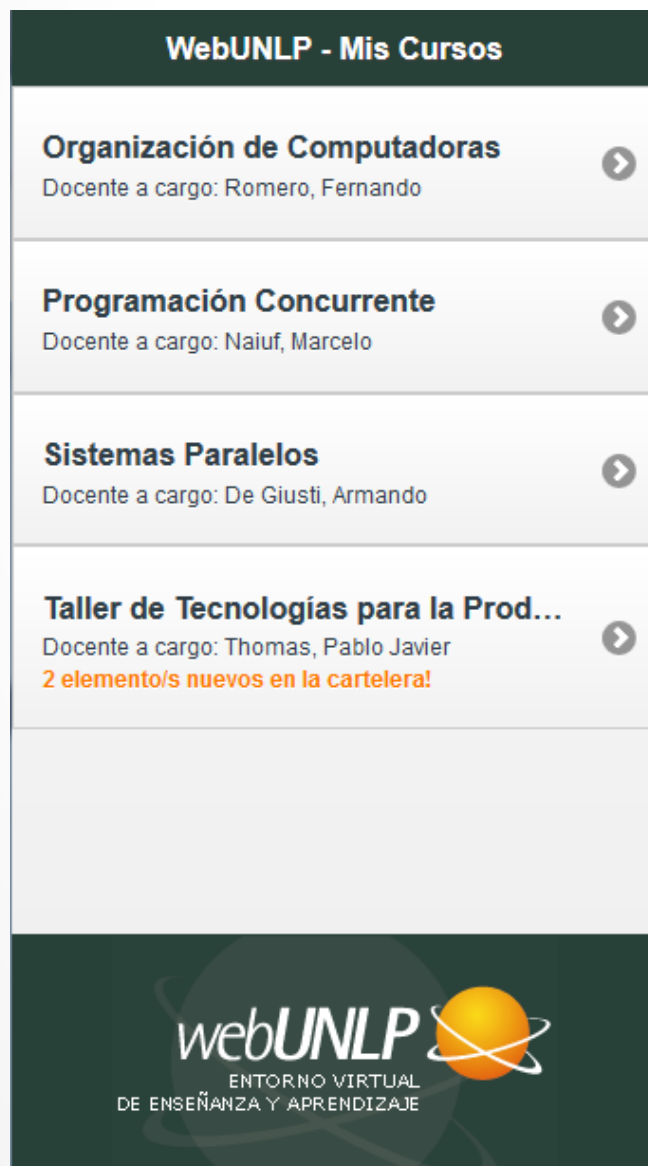


Native Application (iOS)



Case Study

Hybrid Application (Phonegap + JQueryMobile)

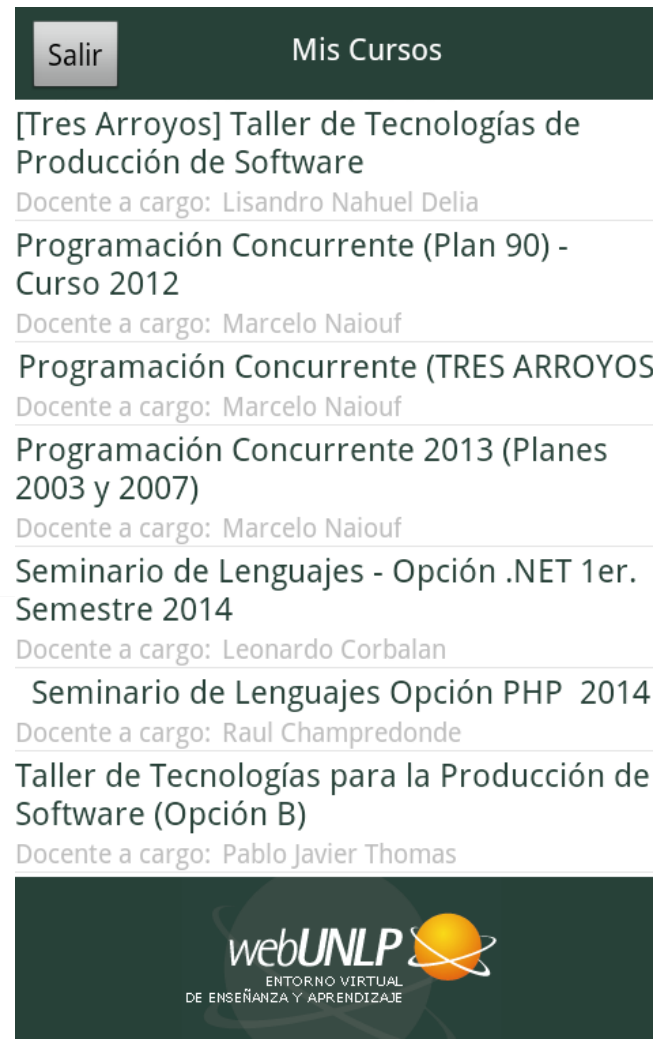


Hybrid Application (Sencha Touch)



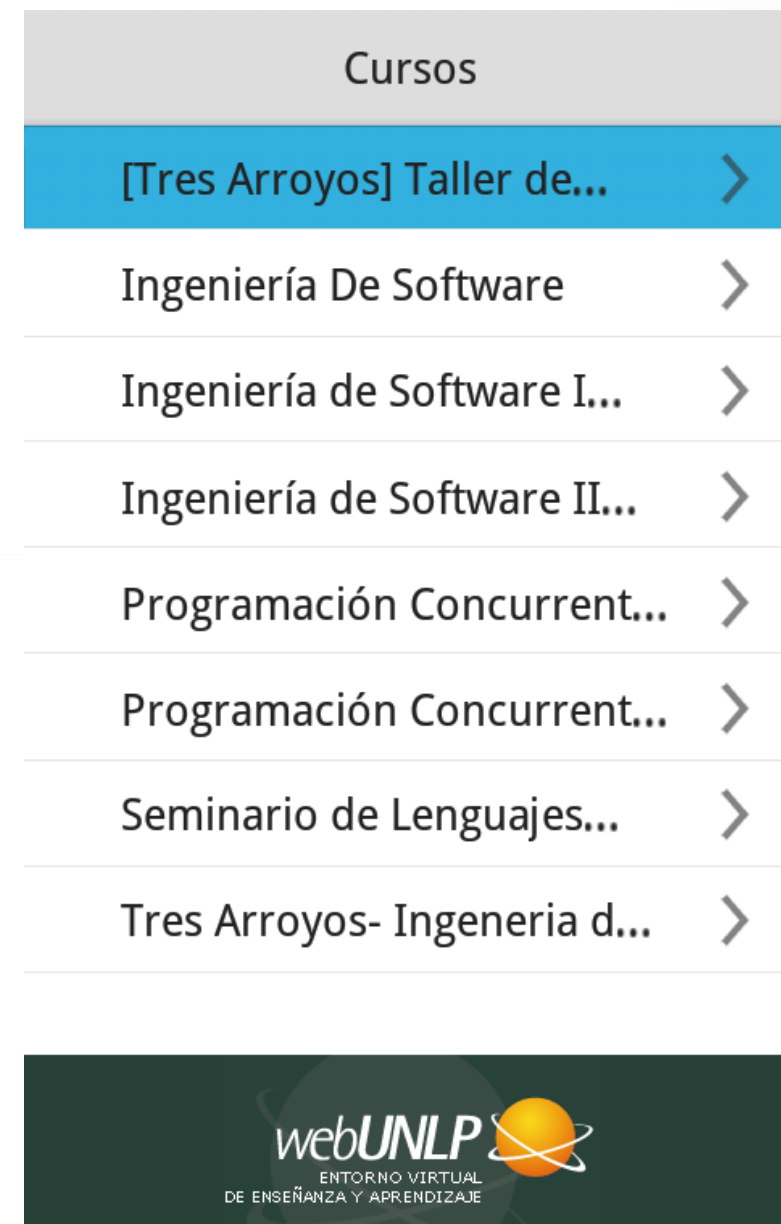
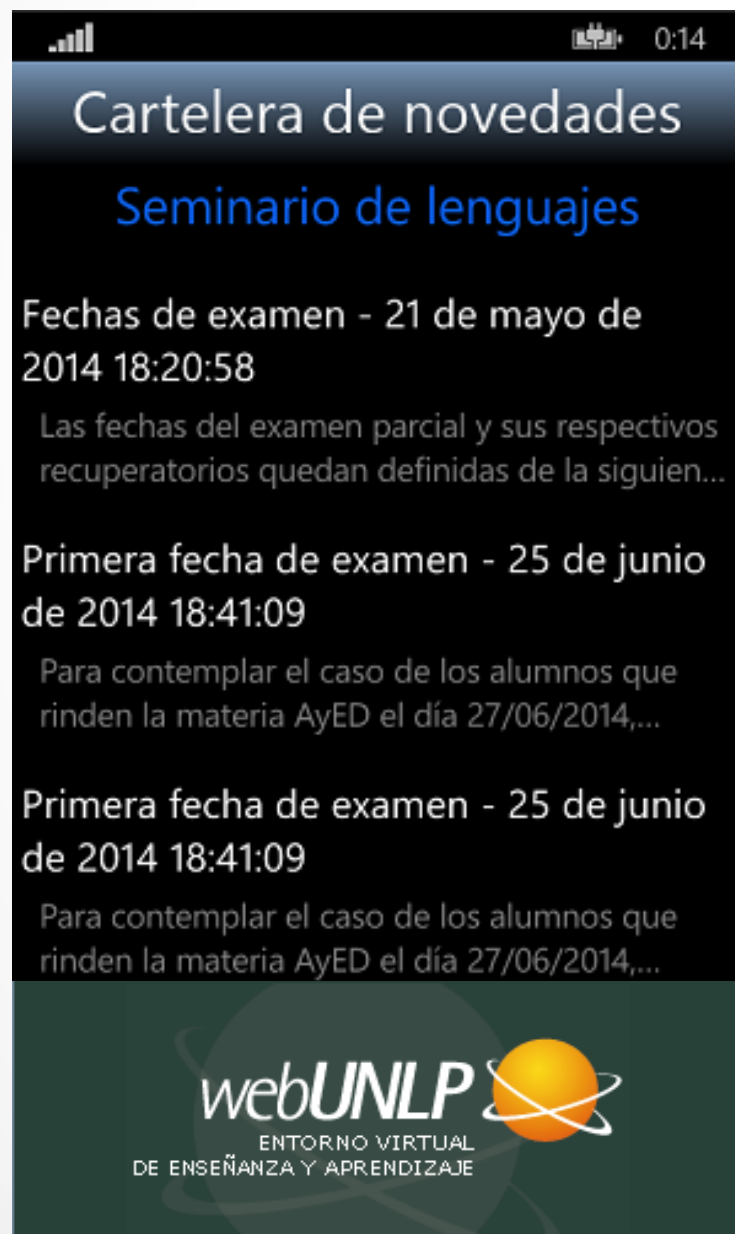
Case Study

- **Interpreted Application (Appcelerator Titanium)**



Case Study

- Applications Generated by cross-compilation
(Xamarin/Visual Studio) (Delphi XE6)



Case Study



- In summary, case studies were performed using:
 - native development approach
 - Android
 - iOS
 - multi-platform development approach
 - Móvil Web (HTML5 + CSS + JS)
 - Hybrid (Phonegap + JQueryMobile)
 - Hybrid (Sencha Touch)
 - Interpreted (Appcelerator Titanium)
 - Cross-compilation (Xamarin / Visual Studio)
 - Cross-compilation (Delphi XE6)

Agenda



1. Introduction
2. Multi-Platform Mobile Applications
3. Case Study
4. **Conclusions**
5. Future Work

Conclusions



- All the development alternatives analyzed have both advantages and disadvantages.
- If priority is given to good performance and user experience towards native applications, interpreted and cross-compilation approaches are the most appropriate.
- If the goal consists of minimizing the development effort, then hybrid and mobile web approach are the best options (the hybrid approach should be adopted if access to all device features is required).

Agenda



1. Introduction
2. Multi-Platform Mobile Applications
3. Case Study
4. Conclusions
5. **Future Work**

Future Work



- Analyzing other tools for multi-platform mobile development, as AppMethod and Applause among others.
- Obtaining quantitative and qualitative aspects metrics for each approach, such as levels of code reuse, performance, battery consumption, user acceptance.
- Studying different Hybrid UI frameworks including Ionic, Framework 7, Onsen UI, etc.

Thanks for your time !

Suggestions and questions

Lisandro Delía

ldelia@lidi.info.unlp.edu.ar

Institute for Research in Computer Science LIDI
Faculty of Computer Science
National University of La Plata
La Plata - Buenos Aires - Argentina

