



Juan Antonio Karmy, Matías Fernández , Gonzalo Díaz, Maximiliano Guerrero

## INTRODUCCIÓN

Las tecnologías móviles han producido un impacto gigantesco en el mundo actual y en la forma como nos comunicamos. Con el tiempo, estas tecnologías han abierto el paso a extender el desarrollo de software y entregar a las personas distintas experiencias acorde a sus necesidades. Siguiendo esta sinergia, surgió este proyecto de investigación, desarrollado durante el primer semestre del año 2011, para ahondar sobre las ventajas de desarrollar software para estas plataformas y, a su vez, entregar un producto de calidad para toda la comunidad UC. Nuestro proyecto ahondó en el desarrollo de aplicaciones móviles para las dos plataformas más populares en el mercado de hoy; Apple iOS y Google Android. El propósito es justamente comparar el desarrollo en ambas plataformas y analizar las ventajas y desventajas de una y otra.



Pontificia Universidad Católica, Departamento de Ciencias de la Computación de la Facultad de Ingeniería.

**Juan Antonio Karmy**, alumno de cuarto año.  
**Matías Fernández**, alumno de cuarto año.

No aparecen en la foto:  
**Maximiliano Guerrero**, alumno de cuarto año.  
**Gonzalo Díaz**, alumno de cuarto año.

Contacto: Andrés Neyem - andres.neyem@gmail.com

## EXPERIMENTACIÓN

El proyecto se llevó a cabo el primer semestre de 2011, bajo la supervisión del profesor del Departamento de Ciencias de la Computación de la Facultad de Ingeniería, Andrés Neyem. Buscábamos crear una herramienta que permitiera a las personas poder ubicarse en cualquier campus de la PUC. La aplicación contaría con un mapa personalizado de cada campus para mostrar todos los edificios e instalaciones. A su vez, el usuario sería capaz de revisar información importante acerca de esa construcción, como salas de clases, baños, fotocopiadoras, etc. Parte de la metodología de trabajo consistió en reuniones semanales con el profesor para monitorear los avances y definir cómo iríamos implementando las diferentes funciones del programa. Además, contábamos con dos módulos académicos semanales para avanzar en el proyecto, tiempo que aprovechábamos en grupo para poder resolver dudas e inquietudes. El grupo de 4 personas fue dividido en dos. Uno se encargó del desarrollo de la aplicación en la plataforma iOS y el otro se concentró en Android.

Para el desarrollo en iOS, requerimos tanto de hardware como de software de Apple. Utilizamos dos MacBook Pro, dos iPhone 4 y un simulador virtual para el desarrollo inicial de la aplicación. Cada computador estaba equipado con la IDE de programación XCode 4. Posteriormente, en la etapa de testing, se utilizaron 2 iPod Touch de tercera generación para probar la aplicación en otros dispositivos Apple. Además, gracias a MECOLAB UC, pudimos contar con las licencias especiales de desarrollo para poder probar la aplicación en los teléfonos.

El ambiente de desarrollo de XCode usa el lenguaje de programación Objective-C, que hasta comienzos del 2011 era desconocido para nosotros. Es por esto que el equipo que lideró el desarrollo en iOS reforzó sus conocimientos del lenguaje en enero del mismo año [3], antes del comienzo del período académico.

Para el desarrollo en Android, los requisitos eran menos estrictos, ya que se podía contar con cualquier computador para el desarrollo. Utilizamos la IDE Eclipse para escribir el código y no se requirió ningún conocimiento extra en el ambiente de desarrollo, ya que las aplicaciones son

escritas en Java, lenguaje de programación ampliamente conocido por los alumnos del Departamento de Computación. Para el testing de la aplicación, se utilizaron celulares HTC y SonyEricsson, ambos con Android, más el simulador virtual incluido en el SDK de Android.

A medida que el proyecto fue avanzando, en las reuniones sucesivas fue naciendo la idea de implementar la funcionalidad de Realidad Aumentada (Augmented Reality). El enfoque que adoptamos para la aplicación fue de adjuntar información de geolocalización en cada construcción. De esta forma, cuando el usuario activa la modalidad de realidad aumentada, la cámara se activa, permitiéndole al usuario explorar su alrededor mientras unas etiquetas aparecen de forma dinámica mostrándole al usuario qué es lo que está viendo.

Para lograr esto fue necesario incluir API especiales de Realidad Aumentada [1] en cada aplicación para el soporte matemático que hay detrás.



Figura 1: Maqueta de la aplicación en iPhone 3G



Figura 2: Simulación de Realidad Aumentada en teléfono Android

## RESULTADOS Y DISCUSIÓN

Al finalizar el semestre, ambos equipos llegaron a versiones muy similares de la aplicación para ambas plataformas. Todas las funcionalidades mencionadas anteriormente fueron implementadas. A continuación, mencionamos las ventajas y desventajas de desarrollar en cada una de las aplicaciones:

### iOS

**Ventajas:** La API (Librería de Funciones) de Apple nos permitió implementar diversas funciones de una manera muy sencilla. La cantidad de

funciones en el lenguaje es muy amplia y variada, lo que permite simplificar el trabajo de los programadores y reducir el tiempo de desarrollo considerablemente. A su vez, por el hecho de estar trabajando con equipos de Apple, sabemos exactamente con qué dispositivos podríamos enfrentarnos. De esta forma, preparar la App para cada dispositivo no fue una tarea difícil.

Otro punto a destacar es la gran cantidad de documentación existente [2], tanto de Apple, como de terceros, lo que nos permitió resolver dudas de forma rápida y



Buscábamos crear una herramienta que permitiera a las personas poder ubicarse en cualquier campus de la UC.

precisa.

El concepto y paradigma en el que está fundado el desarrollo en iOS (el mismo Cocoa Framework) es muy sólido y consistente con el contexto móvil, lo que permite seguir fácilmente una lógica de cómo se deberían ir implementando ciertas funciones.

Finalmente, rescatamos la facilidad que provee XCode (IDE de programación) de armar bastantes partes de la aplicación de forma gráfica, contando con una maqueta del teléfono, permitiendo agregar la Interfaz Gráfica directamente como "dibujo", y no necesariamente a través de código puro.

**Desventajas:** La IDE (Herramienta de programación) XCode es una herramienta un poco incómoda de usar, ya que aún presenta muchas fallas importantes que se hacen molestas para el desarrollador. El mismo ambiente de depuración deja bastante que desear, especialmente en la forma como entrega resultados sobre los errores de la aplicación. En este sentido, el desarrollo tuvo sus momentos de retraso, ya que al momento de detectar y corregir errores, el proceso se volvía largo y tedioso. Para ejemplificar, un error común que nos aparecía era el de excepción de memoria. En vez de que la IDE nos indicara dónde estaba el problema, solo nos decía la clase de problema que existía.

Lo otro que es bastante tedioso es

el manejo de licencias de desarrollo. Para poder probar la aplicación en teléfonos se debe pasar por un proceso de autorización de licencias para los desarrolladores encargados, proceso el cual puede tomar hasta un día completo.

#### Android

**Ventajas:** Una de las gracias de desarrollar en Android, es que uno puede recurrir a cualquier sistema operativo y a cualquier IDE que soporte Java. El mismo hecho de trabajar en Java permite desarrollar con un lenguaje rico en funciones. La misma IDE Eclipse (IDE de programación) es extraordinaria para la depuración y el costo de aprendizaje es bastante bajo.

Además, una vez finalizada la aplicación, estará disponible para todos los dispositivos con Android, lo que considera un mercado muy amplio de los celulares a la venta hoy en día.

**Desventajas:** El hecho de desarrollar para Android implica estar preocupado del comportamiento de la aplicación en muchos dispositivos, lo que hace que la preparación final de la *App* sea más lenta y tediosa que en otras plataformas.

Además, siendo el SDK de Android una herramienta genérica (no necesariamente se trabaja en Eclipse) hace que la integración con la IDE no sea la deseada, por lo que no se le saca el máximo

provecho a todas las facilidades de la IDE que se esté usando.

El simulador virtual es realmente malo, y el desarrollo fue testeado en los teléfonos casi todo el tiempo. A diferencia de iOS, no existe una herramienta gráfica para diseñar la GUI (Graphic User Interface), lo que hace que sea aun más tedioso escribirla completamente en código.

Por último, nos encontramos con la sorpresa de que la documentación de la plataforma no contiene documentación completa, tanto de Google, como de terceros, lo que nos hizo perder mucho tiempo resolviendo problemas por nuestra cuenta.

## CONCLUSIONES

Finalmente, pudimos ser capaces de terminar un producto que entrega valor a la comunidad UC, y que representa un servicio para todas las personas. Pudimos reforzar nuestros conocimientos de programación en tecnologías relativamente nuevas y que están produciendo un gran impacto a nivel mundial

Ambas aplicaciones están siendo mejoradas por el momento, y ya se puede encontrar la aplicación para iOS en la Apple App Store, por el nombre de UC-AR. La aplicación para Android estará disponible muy pronto en el Android Market.



Figura 3: Código QR (Quick Response Code) para descargar la Aplicación de la App Store



Figura 4: Ícono oficial de la Aplicación

## GLOSARIO

**API (Application Programming Interface):** Corresponde a un set de instrucciones o librería de funciones que permite a los desarrolladores incorporar funcionalidades entre Softwares. Por ejemplo, un software A que permite realizar varias funciones podría tener una API para que un software B implementara una de estas funciones. UC-AR utiliza una API abierta al público para implementar la función de Realidad Aumentada.

**IDE (Integrated Development Environment):** Es un set de herramientas que son usadas por los desarrolladores para programar y crear los distintos softwares. Una IDE está comúnmente formada por un editor de texto para el código fuente, un compilador, que permite interpretar el código y correr el programa, y algunas herramientas de depuración que permiten identificar errores en el código y/o ejecución de la aplicación.

Dado que la IDE tiene un compilador como parte del set de herramientas, va a depender justamente de éste último para poder crear/correr ciertos programas. Cada IDE

## AGRADECIMIENTOS

Agradecemos al profesor Andrés Neyem, por haber apoyado el proyecto desde el comienzo. A Daniela Cid, periodista del Departamento de Ciencias de la Computación, por sus labores de difusión de la App; Andrea Vásquez, alumna de postgrado del Departamento de Ciencias de la Computación, por su ayuda con el diseño de panfletos de difusión; Claudia Gutiérrez, Francisca Hornig y Constanza Carmi, periodistas encargadas de la Web UC, por su ayuda en la adquisición de información sobre los distintos campus de la UC; Sergio Campamá, alumno del Departamento de Ingeniería Eléctrica, por su ayuda en conceptos sobre iOS y a todos los alumnos y compañeros que nos dieron feedback durante el proceso de desarrollo.

## REFERENCIAS

1. STACKOVERFLOW y GOOGLE, iPhone Development [Búsqueda Web / Electrónico] Stack Exchange, Inc., Google Inc. Software API.
2. APPLE, iOS Developer Guide [Documentación Electrónica] Apple Inc. iOS API User Guide.
3. STANFORD UNIVERSITY y APPLE, Developing Apps for iOS [Multimedia Digital] Versión 2010. Stanford University, Apple Inc. iTunes U Service. Multimedia Educacional.



Figura 5: MECOLAB es un laboratorio que tiene la misión de incentivar el desarrollo de aplicaciones móviles y en dispositivos embebidos. Provee dispositivos a las personas que estén interesadas en realizar algún proyecto, y constantemente realiza actividades para los alumnos, como charlas con expertos en el tema. Para mayor información, visita: <http://mecolab.ing.puc.cl>

soporta lenguajes de programación específicos.

Dentro de las IDEs más comunes se pueden encontrar:

**Eclipse:** Permite crear aplicaciones basadas en el lenguaje Java y en otros lenguajes por medio del uso de plug-ins.

**Xcode:** Permite crear aplicaciones basadas en el lenguaje Objective-C. Además tiene soporte para los lenguajes C y C++.

**Visual Studio:** Una de las IDE de programación más populares y completas en el mercado. Permite crear aplicaciones basadas en los lenguajes C, C++, Visual Basic, C#, F#, Python y Ruby, entre otros.

**Realidad Aumentada (Augmented Reality):** El concepto de Realidad Aumentada consiste en el mejoramiento de la percepción de la realidad que tiene el usuario mediante el uso de un programa computacional. Dentro de los ejemplos más comunes, podemos encontrar teléfonos que despliegan información en tiempo

real al usuario a medida que este “observa” su entorno por medio de la cámara del dispositivo.

**Código QR (Quick Response Code):** El código QR es un código de barras bidimensional que fue diseñado en Japón cerca del 1994, y que se ha hecho cada vez más popular con el tiempo, especialmente con la revolución de tecnologías móviles de hoy en día. La gracia de este código, es que permite identificar de forma única en el mundo cualquier tipo de información, como binaria, alfanumérica y acceder a esta de forma muy rápida.

**GUI (Graphic User Interface):** La interfaz gráfica es un concepto muy usado en el área de la computación y que nace en los años '70. Representa una manera de interactuar con distintos dispositivos electrónicos en base a íconos, imágenes o elementos gráficos que entregan un cierto contexto visual a las acciones que el usuario desea realizar. Ej.: Cerrar una ventana se conoce normalmente por un botón rojo que tiene una cruz (X) al centro.