
CAPÍTULO 1

Hardware y Software

1.1. Introducción

En el presente capítulo se introducen algunos conceptos básicos para el desarrollo sobre plataformas móviles. En particular se describe el *hardware* que fue utilizado, el lenguaje de programación y el entorno de desarrollo que permitieron incorporar procesamiento sobre los dispositivos. En lo que respecta al *hardware*, esto es todo lo que se comenta sobre el mismo y a nivel de *software* simplemente se introducen algunos conceptos básicos. En el Apéndice ?? se hace un desarrollo más extensivo de las características del *hardware* de Apple Inc., de los sistemas operativos que emplean, del lenguaje de programación y entorno de desarrollo así como del conjunto de librerías existentes escritas en Objective-C y su aplicación.

1.2. Hardware

1.2.1. Elección de plataforma

La elección de la plataforma para desarrollar fue una de las primeras decisiones que se tuvo que tomar más allá del proceso natural inicial de investigación del proyecto. Esto es debido a que existen muchos factores que se ven afectados en función de la plataforma que van desde el aprendizaje del lenguaje de programación y del entorno de desarrollo hasta la adquisición de plataformas y/o máquinas para desarrollar. Así entonces, las dos grandes posibilidades que se tenían al comienzo y que determinaban estos factores eran plataformas que tuvieran los sistemas operativos *Android* o *iOS* (en ningún momento se consideró desarrollar sobre plataformas con *Symbian* dado que está planificado que deje de tener soporte para 2016 y tampoco sobre plataformas con *Windows Phone 8* dado que no había sido lanzado en el momento de la decisión). Uno de los aspectos desfavorables que se veía de trabajar sobre *Android* era la multiplicidad de plataformas existentes, de distintas características de procesamiento, cámara y sensores entre otras cosas con respecto a las plataformas que utilizan *iOS*. Por otra parte, luego de un proceso de investigación se vio que el conjunto de herramientas existentes y el estado de maduración del desarrollo de aplicaciones para *iOS* hacían de este sistema operativo un camino más seguro para comenzar en un área que era desconocida. *Android* es un sistema operativo que viene en pleno crecimiento y que a nivel masivo es una buena alternativa desarrollar en él, pero para los fines de investigación y de desarrollo de *software* que tenía el presente proyecto se vio que era mejor desarrollar sobre *iOS*, presente en plataformas de Apple.

Al trabajar con Apple entonces se cuenta con la ventaja de contar con pocas variantes en cuan-

to al *hardware* utilizado. Básicamente, existen tres familias de dispositivos en los que se puede desarrollar: *iPhone*, *iPad* y *iPod Touch*. Para cada familia existen distintos modelos que hacen que algunas características importantes como la capacidad de procesamiento, la resolución de cámara o el tamaño de la pantalla entre otras, puedan verse afectadas. A continuación se presenta brevemente cómo fue el surgimiento de cada uno de los dispositivos al mercado y se describen resumidamente las principales características.

1.2.2. Comparación de plataformas con iOS: iPhone, iPad, iPod Touch

Sin dudas el *iPhone* fue uno de los saltos más grandes en el mundo tecnológico en los últimos años. Logró llenar el hueco que los *PDA*s (*Personal Digital Assistant*) de la década de los 90 no habían sabido completar y comenzó a desplazar al invento que revolucionó el mercado de los contenidos de música, el *iPod*. Gracias a su pantalla táctil capacitiva de alta sensibilidad logró reunir todas las funcionalidades agregando solamente un gran botón y algunos extra para controlar volumen o desbloquear el dispositivo.

La primera generación del *iPhone* fue lanzada por Apple en Junio de 2007 en Estados Unidos, luego de una gran inversión de la operadora AT&T que exigía exclusividad de venta dentro de dicho país durante los siguientes cuatro años. La misma soportaba tecnología GSM cuatribanda y se lanzó en dos variantes de 4GB y 8GB de ROM. El segundo modelo lanzó como novedad el soporte de tecnología 3G cuatribanda y GPS asistido. Luego le siguieron el *iPhone 3GS*, *4*, *4S* y el *5*, siendo este último, la sexta y última generación disponible al momento de la redacción de este trabajo.

Por su parte, tanto el *iPad* como el *iPod Touch* también representaron un gran salto en el mundo de las plataformas y *Tablets*, agrandando las posibilidades de desarrollo y procesamiento. Como se dijo, de cada una de estas tres familias de dispositivos existen distintas versiones y modelos. Por eso, a continuación se muestra una tabla comparativa de determinadas características que son de interés a los efectos del presente proyecto.

	iPhone 4	iPhone 4s	iPod Touch 4G	iPad 2
ROM	8, 16 o 32 GB	16, 32 o 64 GB	8, 32 o 64 GB	16, 32 o 64 GB
RAM	512 MB	512 MB	256 MB	512 MB
SoC	Apple A4	Apple A5	Apple A4	Apple A5
CPU	1 GHz, ARM Cortex-A8	1 GHz, dual-core ARM Cortex-A9	800 MHz, ARM Cortex-A8	1 GHz dual-core ARM Cortex-A9
GPU	PowerVR SGX535 GPU	PowerVR SGX543MP2 (2-core) GPU	PowerVR SGX535 GPU	PowerVR SGX543MP2 (2-core) GPU
CÁMARA	Foto: 5.0 MP Video: 720p HD (30 fps)	Foto: 8.0 MP Video: 1080p HD (30 fps)	Foto: 0.7 MP Video: 720p HD (30 fps)	Foto: 0.7 MP Video: 720p HD (30 fps)
PANTALLA	Diagonal: 3.5" Pixels: 960x640 Densidad de Pixels: 326 ppi Multitáctil	Diagonal: 3.5" Pixels: 960x640 Densidad de Pixels: 326 ppi Multitáctil	Diagonal: 3.5" Pixels: 960x640 Densidad de Pixels: 326 ppi Multitáctil	Diagonal: 9.7" Pixels: 1024x768 Densidad de Pixels: 123 ppi Multitáctil
SENSORES	Girsóscopo de 3 ejes Acelerómetro Sensor de luz ambiente Sensor de proximidad	Girsóscopo de 3 ejes Acelerómetro Sensor de luz ambiente Sensor de proximidad	Girsóscopo de 3 ejes Acelerómetro Sensor de luz ambiente	Girsóscopo de 3 ejes Acelerómetro Sensor de luz

Tabla 1.1: Comparativa de algunas plataformas Apple. [?] [?] [?] [?]

Hay algunos comentarios respecto de la Tabla 1.1 que es bueno destacar. Es importante decir que se eligieron esos cuatro dispositivos, pues pareció de interés conocer al menos una plataforma de cada familia; y dentro de las mismas se eligieron las que fueron utilizadas para desarrollar.

Uno de los puntos a evaluar es el **SoC**, que refiere a *System on Chip* por sus siglas en inglés. “*System on Chip*” es un concepto de los sistemas embebidos que refiere a la integración de todo lo necesario para poder correr un sistema operativo, en un solo circuito integrado. En contraposición a un microcontrolador que es capaz de realizar procesamiento más básico y menos potente, con poca interacción de usuario y menor flexibilidad, un *SoC* refiere a la idea de tener todo lo necesario para desarrollar sobre la plataforma y poder hacer procesamiento sin tantas limitaciones. Básicamente cumplen funciones similares pero el *SoC* forma parte de una evolución de los microcontroladores, siendo de una complejidad mayor e integrado en un tamaño muy reducido buscando poco consumo y eficiencia de costos. Así entonces, un *SoC* puede estar conformado por un microcontrolador y hardware adicional como procesadores de señal y bloques de memoria [?].

En la Figura 1.1 se ilustran los dos tipos de *SoC* de los dispositivos de la Tabla 1.1: Apple A4 y Apple A5. Algunos dispositivos que no figuran en la tabla como el *iPhone 5* o el *iPad 4* usan *SoCs* más recientes como el Apple A6 o Apple A6x respectivamente. La familia de *SoCs* Apple Ax, es la que la mencionada firma utiliza en todas sus plataformas, inclusive en el *Apple TV* y es manufacturada por Samsung. Estos *SoCs* se caracterizan por utilizar CPUs de arquitectura ARM, en su mayoría ARMv7 y GPUs de PowerVR de la línea SGX.

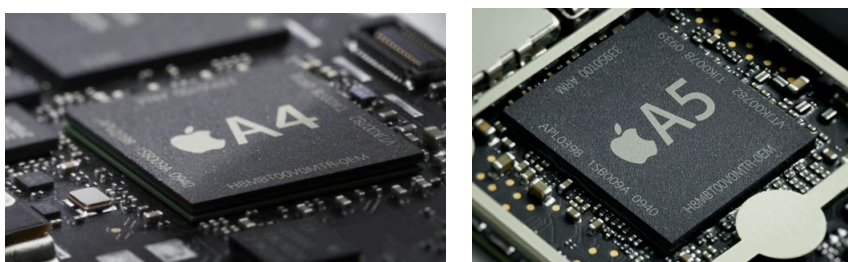


Figura 1.1: *System on Chip*: SoC.

1.3. Desarrollo de Software

Para poder desarrollar aplicaciones sobre dispositivos móviles de la firma Apple Inc. es necesario contar con computadoras que corran el sistema operativo **Mac OS X**. Esto puede ser llevado a cabo, ya sea adquiriendo plataformas de desarrollo de la mencionada firma o creando máquinas virtuales que corran dicho sistema operativo. Para la segunda opción (la más económica pero con ciertas dificultades de *performance*), es necesario que la computadora cuente con virtualización de *hardware*. Se comenzó trabajando de esta manera hasta el momento de adquirir plataformas de desarrollo que contaran con Mac OS X en forma nativa.

Una de las grandes innovaciones de estas plataformas móviles es el hecho de poder desarrollar aplicaciones y correrlas en el propio dispositivo (por supuesto también sucede lo mismo en el mundo Android). Para poder lograr esto, además de contar con una máquina que corra Mac OS X, es necesario contar con el “*Integrated Development Environment*” (IDE) apropiado llamado **Xcode**. Este entorno de desarrollo y su lenguaje, Objective-C, se explican en la Sección 1.3.1.

1.3.1. Objective-C

El lenguaje que fue elegido por Apple Inc. para desarrollar sobre plataformas móviles es Objective-C. Este lenguaje fue desarrollado en la década de 1980 como un superconjunto de C orientado a objetos. Se trata entonces de una extensión del standard ANSI C que incorpora un modelo orientado a objetos basado en **Smalltalk** [?].

Este lenguaje cuenta con algunas particularidades en cuanto a la invocación de métodos (procedimientos) y algunas otras características que no van más allá de la sintaxis; como los métodos de clase y los métodos de instancia, como los métodos *get* y *set* para acceder y configurar atributos (propiedades) de los objetos, como la notación de *import* en lugar de *include* para quienes están acostumbrados a C, y así varios detalles más. Sin embargo, más allá de estas y otras diferencias y particularidades resulta un lenguaje relativamente ágil y dentro de todo sencillo de aprender para quien tiene ya un conocimiento de otros lenguajes orientados a objetos.

Sin embargo, los algoritmos desarrollados y utilizados en el proyecto para los módulos de detección de características, filtrado y agrupamiento de segmentos, determinación de correspondencias y estimación de pose; se encuentran escritos en lenguaje C y están autocontenidos en el mismo código fuente sin la utilización de librerías externas para su funcionamiento. Se hace énfasis en que estos módulos constituyen el núcleo de la solución al problema de realidad aumentada planteado inicialmente.

Es directo incorporar código escrito en lenguaje C a Objective-C ya que como se dijo el primero está contenido en el segundo. Esto refuerza tanto la elección de la plataforma como del lenguaje ya que es posible sacar provecho de todas las funcionalidades y librerías de alto nivel proporcionadas por Objective-C y a la vez poder implementar algoritmos de bajo nivel en C.

La elección del lenguaje y la filosofía de desarrollo comentada anteriormente permite **portabilidad** en el uso de los mismos para aplicaciones más allá del proyecto y significó también un ejercicio de aprendizaje muy interesante.

1.3.2. Lenguaje C

Al comienzo del proyecto se realizó en curso breve de introducción al lenguaje C alojado en: <http://iie.fing.edu.uy/rosaluna/wiki/Ipol:main>. De este curso se abarcaron principalmente los siguientes temas:

- Módulo 1: Introducción al lenguaje C.
- Módulo 2: Punteros, arreglos y estructuras. Memoria dinámica.
- Módulo 3: Diseño de un filtro FIR.
- Módulo 4: Herramientas de programación.
 - Entornos de desarrollo: e.g. Netbeans.
 - Makefile: make.
 - Debugger: gdb/ddd.
 - Profiler: valgrind/kcachegrind.
 - Documentación: doxygen.

- Generación de makefiles multiplataforma: cmake.
- Módulo 5: Lenguaje C aplicado al procesamiento de imágenes.

La utilización del lenguaje C no es casual en este proyecto. Dicho lenguaje tiene varias ventajas sobre otros lenguajes posibles, en particular para procesamiento de imágenes. En primer lugar, al ser un lenguaje de bajo nivel el código y algoritmos desarrollados pueden ser optimizados hasta el detalle. Por otro lado, la disponibilidad de algoritmos y librerías existentes en C/C++ orientadas a procesamiento de imágenes es superior a la de otros lenguajes. Algunas librerías importantes son OpenCV, ITK e ImageMagik ya que son ampliamente utilizadas y cuentan con una gran comunidad de desarrolladores y/o de empresas que las respaldan. También existen repositorios públicos y comunidades como IPOL que se rigen bajo el modelo de Acceso Abierto e Investigación Reproducible y brindan documentación y desarrollos de algoritmos en C/C++ para procesamiento de imágenes.

Sin ir mas lejos, la implementación del algoritmo LSD utilizada en el presente proyecto, fue tomada de su publicación en IPOL, mientras que el autor de los artículos utilizados para la implementación del algoritmo de estimación de pose POSIT provee en su página el código fuente en C y en Matlab de algunas de las versiones del mismo.

1.4. Resumen

En resumen frente a los dos sistemas operativos más importantes para desarrollar se optó por elegir iOS y desarrollar en plataformas de Apple. Se conoce el poder de procesamiento de los *SoCs* que componen estas plataformas que integran CPU ARMv7 en varias versiones y GPU SGX con distintas capacidades. Se vio un análisis comparativo entre las tres familias de dispositivos, y dentro de las mismas, en detalle los modelos de plataformas que fueron utilizados. Se explicaron algunas características generales de desarrollar sobre plataformas con iOS, así como también del lenguaje de programación Objective-C, superconjunto de C y que es el utilizado para desarrollar en los dispositivos mencionados. En caso de querer profundizar sobre los distintos *frameworks* del SDK de iOS, herramientas de *debugging* y herramientas generales de desarrollo, se sugiere leer el Apéndice ??.