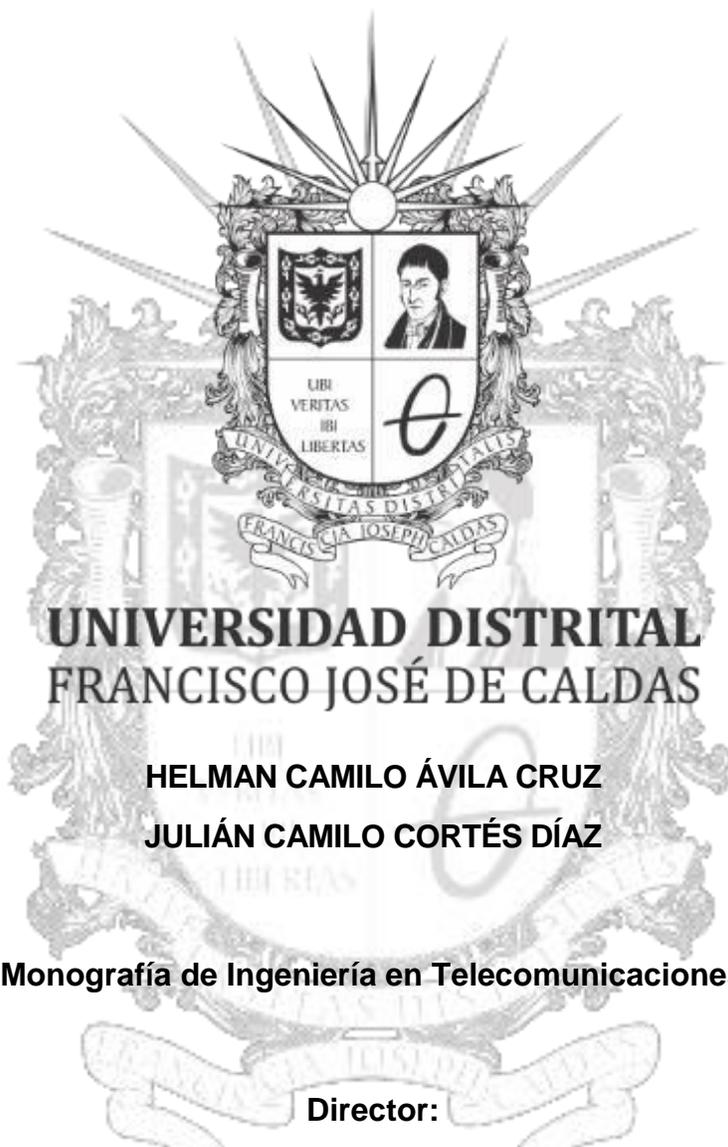


**GUÍA PARA LA REALIZACIÓN DE APLICACIONES MÓVILES EN LOS
SISTEMAS OPERATIVOS ANDROID E iOS**



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**

**HELMAN CAMILO ÁVILA CRUZ
JULIÁN CAMILO CORTÉS DÍAZ**

Monografía de Ingeniería en Telecomunicaciones

Director:

Ing. JOSÉ DAVID CELY CALLEJAS

**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**

**Universidad Distrital Francisco José de Caldas
Facultad Tecnológica
Programa de Ingeniería en Telecomunicaciones
Bogotá, Abril de 2016**

HOJA DE ACEPTACIÓN

GUÍA PARA LA REALIZACIÓN DE APLICACIONES MÓVILES EN LOS SISTEMAS OPERATIVOS ANDROID E iOS

Observaciones:

Vo. Bo. Tutor
Ing. JOSE DAVID CELY CALLEJAS

Vo. Bo. Evaluador del proyecto



Dedicatoria

A mis padres por el apoyo que me han brindado; a mi esposa y familia en general. A cada uno de los amigos y compañeros que conocí en la universidad; a los profesores que me han aportado en este camino de conocimiento, y a la Universidad Distrital Francisco José de Caldas, Facultad Tecnológica

Helman Camilo Ávila Cruz



Dedicatoria

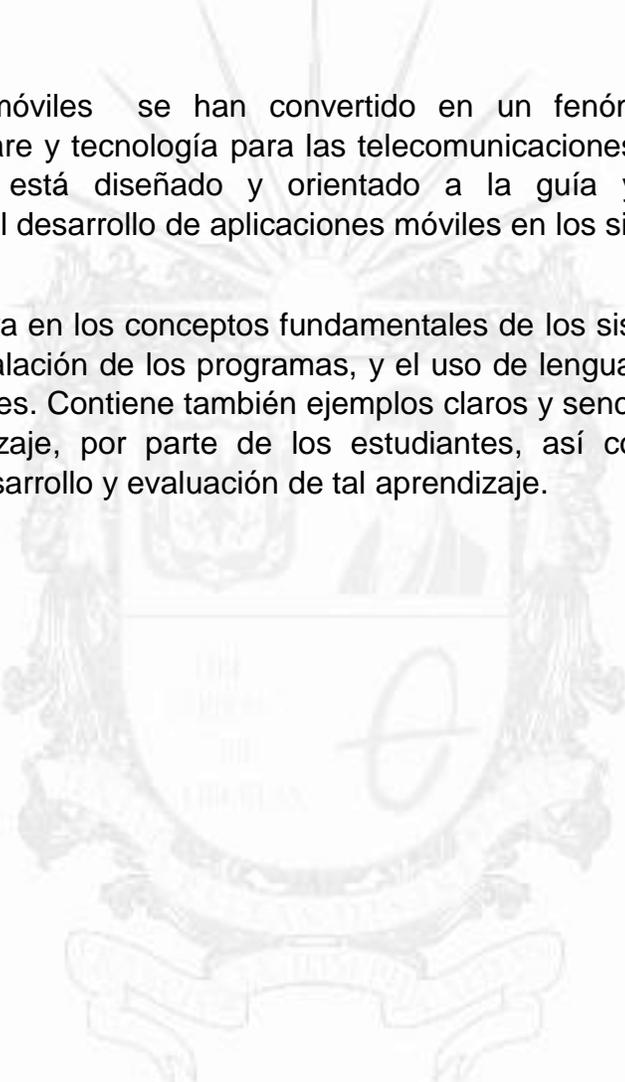
A mis padres por el gran apoyo y amor que me han brindado. A todos aquellos con quienes me he cruzado en la carrera y puedo llamar amigos; a los profesores por sus enseñanzas, apoyo y disponibilidad brindada; y a la Universidad Distrital Francisco José de Caldas, Facultad Tecnológica

Julián Camilo Cortés Díaz

Resumen

Las aplicaciones móviles se han convertido en un fenómeno mundial de desarrollo de software y tecnología para las telecomunicaciones. Por tal razón, el presente proyecto está diseñado y orientado a la guía y aprendizaje de herramientas para el desarrollo de aplicaciones móviles en los sistemas operativos iOS y ANDROID.

La guía se estructura en los conceptos fundamentales de los sistemas operativos, así como en la instalación de los programas, y el uso de lenguaje y componentes de los desarrolladores. Contiene también ejemplos claros y sencillos que ayudan a reforzar el aprendizaje, por parte de los estudiantes, así como de ejercicios propuestos para desarrollo y evaluación de tal aprendizaje.

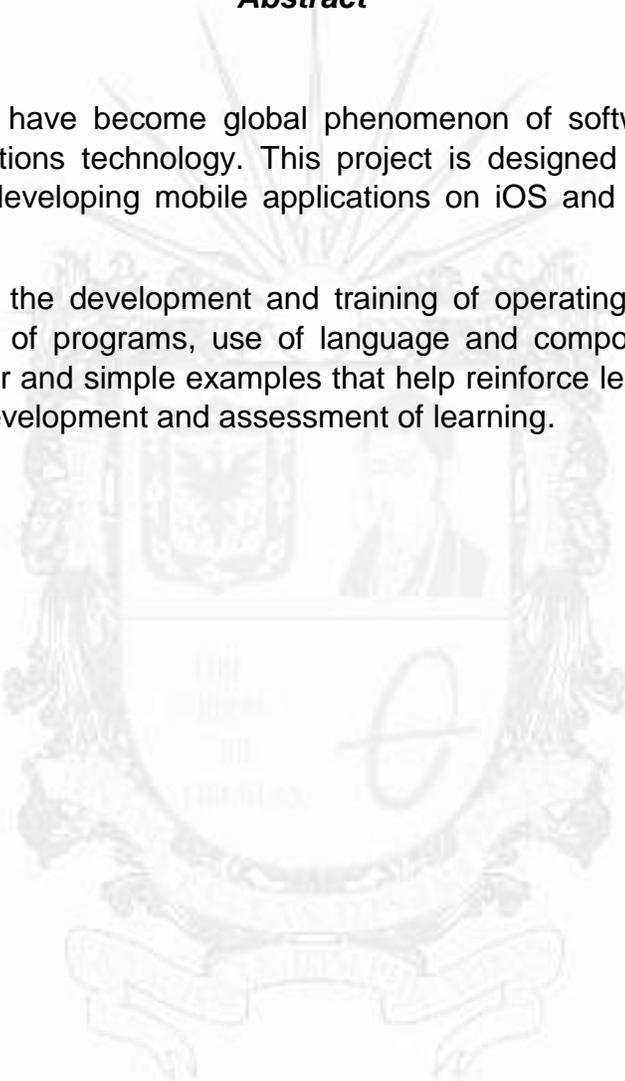


UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Abstract

Mobile applications have become a global phenomenon of software development and telecommunications technology. This project is designed and oriented as a learning guide for developing mobile applications on iOS and Android operating systems.

The guide contains the development and training of operating systems, starting with the installation of programs, use of language and components developers. Also it contains clear and simple examples that help reinforce learning by students and exercises for development and assessment of learning.

The image features a large, faint watermark of the University of Francisco José de Caldas. The watermark includes a central crest with a sunburst at the top, a portrait of a man, and a shield with a caduceus. Below the crest is a banner with the university's name. The text 'UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS' is printed in a large, bold, sans-serif font across the bottom of the page.

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Tabla de contenido

Resumen.....	5
Abstract	6
1. Introducción	9
1.1 Planteamiento del Problema	10
1.2 Objetivos	11
1.2.1 Objetivos General	11
1.2.2 Objetivos Específicos	11
2 Marco de Referencia	12
2.1. Antecedentes	12
3. Metodología	13
3.1 Marco teórico	13
3.1.1 Aplicaciones móviles	13
3.1.2 Los Smartphone	15
3.1.3 Android	16
3.1.3.1 Versiones de Android	17
3.1.3.2 Entornos gráficos	17
3.1.4 iOS	19
3.1.4.1 Arquitectura iOS	19
3.1.4.2 Herramientas de desarrollo y Frameworks	20
3.2 Metodología de desarrollo	20
4. Resultados	21
4.1 Guía Android	21
4.1.1 Instalación	21
4.1.2 Temáticas y aprendizajes	24
4.1.3 Ejemplos de aplicaciones	27
4.1.4 Ejercicios propuestos	32
4.2 Guía iOS	33
4.2.1 Instalación	33
4.2.2 Metodología pedagógica	34

5.	Conclusiones	38
6.	Referencias	39

Lista de Figuras

Fig. 1:	Sistema de capas Android	16
Fig. 2:	Modelo arquitectura iOS	19
Fig. 3:	Interfaz SDK de Android	23
Fig. 4:	Interfaz Inicio Android Studio	24
Fig. 5:	Emulador AVD Android	25
Fig. 6:	Aplicación de ejemplo Android	27
Fig. 7:	Composición de ejemplo actividades e ítems	28
Fig. 8:	Ejemplo calculadora	28
Fig. 9:	Activity principal	29
Fig. 10:	Activity secundaria	29
Fig. 11:	Clases para el manejo de SQLite	30
Fig. 12:	Interfaz SQLite	30
Fig. 13:	Emulador ejemplo SQLite	30
Fig. 14:	Instalación de herramientas google play services	31
Fig. 15:	Conexión de API para acceder a los servicios de google	31
Fig. 16:	Emulador de google maps	32
Fig. 7:	Interfaz de inicio de Xcode.....	33

Lista de Tablas

Tabla 1:	Especificación del sistema operativo iOS	19
----------	--	----

Lista de Diagramas

Diagrama 1:	Diagrama de bloques guía Android	21
Diagrama 2:	Diagrama de contenido guía Android.....	22
Diagrama 3:	Diagrama de bloques guía iOS	34

1. Introducción

Este documento contiene información sobre la realización del proyecto de grado: *guía para realización de aplicaciones móviles en los sistemas operativos Android e iOS*. Se realizó una guía completa para el desarrollo de aplicaciones móviles, comenzando desde su respectiva instalación a los entornos de desarrollo, pasando por la estructura de lenguajes, y finalizando con ejemplos claros tanto en la plataforma de Android como iOS. Todo esto con el fin de tener un soporte para la enseñanza básica de los sistemas operativos con mayor demanda en el mercado, y así contribuir al mejoramiento de la comprensión de los conceptos y procesos implicados en este entorno de programación, ya que personas interesadas en esta guía básica tendrán una posibilidad de incrementar una información que se depurará medida que esta sea requerida.

El uso de aplicaciones móviles se ha centrado actualmente en el desarrollo de plataformas de distribución que se manejan con dispositivos móviles compatibles con diferentes entornos a través de sus sistemas operativos.

1.1 Planteamiento del Problema

Debido al incremento de software y hardware en los dispositivos móviles, y la gran revolución de las aplicaciones móviles, se hace necesario retroalimentarse en cuestión de sistemas operativos para Smartphone, en lo que concierne al uso de aplicaciones móviles y a su vez la compatibilidad en diferentes sistemas de trabajo y comunicación. Es por eso que se justifica realizar una guía completa en los sistemas operativos con mayor demanda en el mercado -como Android e iOS- que abarcan la mayor parte de sistemas operativos de equipos móviles hoy en día. Esta guía será de ayuda como herramienta al momento de impartir una formación inicial de tal temática.

Esta guía contempla la estructura de una herramienta para el uso de aplicaciones móviles, desde los conceptos más básicos hasta los más complejos, que ilustrará los distintos procesos que aparecen, desde la instalación de plataformas y software de trabajo hasta ejemplos con diferentes niveles de compatibilidad de comunicación, basándose en la introducción, tanto de los lenguajes como ítems de trabajo y objetos de desarrollo.

La población objetivo del presente proyecto se enfatiza, particularmente, en los estudiantes o personas que deseen adquirir conocimientos en el desarrollo de aplicaciones; quienes, dado el incremento de software que se ha desplegado permanentemente y los múltiples desarrollos en aplicaciones móviles, se ven afectados por la competencia en el mercado laboral.. De esta manera lo que se busca es actualizar y fortalecer en los usuarios este tipo de temáticas generando una base fundamental de conocimiento necesaria en el momento de abordar estas aplicaciones en distintos sistemas operativos.

1.2 Objetivos

1.2.1 General

Diseñar una guía de clase para el desarrollo de aplicaciones móviles en los sistemas operativos Android e iOS

1.2.2 Específicos

Generar un reporte de la instalación del software y desarrolladores de los sistemas operativos Android e iOS.

Crear un informe de introducción a los lenguajes de programación con contenido temático.

Modelar un informe detallado del uso de objetos e ítems de composición para el desarrollo de aplicaciones móviles.

Implementar diferentes ejemplos de aplicaciones realizadas en los sistemas operativos Android e iOS, utilizando las diferentes estructuras de compatibilidad y comunicación.

Diseñar una herramienta para la evaluación del curso.

2. Marco de Referencia

Diferentes proyectos de grado y/o trabajos realizados en empresas se han orientado por las aplicaciones móviles. Esto ocurre, por ejemplo, en los proyectos de grado de la Universidad Distrital en donde se nota un aumento, por parte de los estudiantes de Ingeniería en Telecomunicaciones y de Sistemas, de proyectos realizados a través de aplicaciones móviles en sistemas operativos de software libre -como Android-, y que han sido de gran ayuda no solo para la comunidad académica universitaria, sino al sector social de influencia general de la Facultad Tecnológica. Algunos proyectos de grado que se tomaron como referencia son:

- Prototipo de aplicación móvil instructiva, para el manejo y disposición de desechos sólidos en la ciudad de Bogotá. Realizado por Melo Murcia, Andrés Rojas Reyes, el 15-oct-2015
- Prototipo de aplicación móvil como herramienta de apoyo para la prevención de riesgos y guía de operación en el acontecimiento de siniestros mediante el uso de realidad aumentada y geo posicionamiento. Realizado por Fabio Andrés Roa y Miguel Angel Montañez.
- Implementación de un aplicativo multiplataforma de red social y mensajería instantánea para la red inalámbrica comunitaria del barrio vista hermosa Universidad Distrital "Francisco José De Caldas" Estado: Tesis concluida Ingeniería en Telecomunicaciones, 2013, Realizado por: Miguel Angel Ganoa Suarez - Guilberto Ortiz Mora.
- Desarrollo de una aplicación móvil en realidad aumentada para mejorar el campo de la comunicación de niños autistas en la clínica neurorehabilitar Universidad Distrital "Francisco José De Caldas" Estado: Tesis concluida Ingeniería Electrónica, 2013, Realizado por: Wilmer Pérez Betancourt.

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

3. Metodología

3.1 Marco Teórico

3.1.1 Aplicaciones móviles

El éxito de las aplicaciones móviles ha supuesto un nuevo revulsivo para el mercado de los Smartphone, cada vez más valorados por las bondades de sus sistemas operativos. La aparición de los Smartphone ha revolucionado el panorama de la telefonía móvil, posibilitando el acceso a Internet desde todos los rincones. Las aplicaciones potencian y dan verdadero sentido a estos terminales inteligentes. [1].

La aparición de los Smartphones o teléfonos inteligentes ha generado una industria que no se limita a la mera fabricación de hardware, como ocurría con los primeros teléfonos móviles. En efecto, hasta hace poco, lo más importante era el terminal y el software no suponía un elemento tan decisivo a la hora de hacerse con un producto u otro. Sin embargo, en estos momentos la marca del dispositivo cada vez se está asociando más con el sistema operativo móvil. Esta identificación está ligada a la importancia de las plataformas móviles en detrimento del hardware. Actualmente la tecnología progresa a pasos de gigante y los fabricantes imitan cualquier avance de la competencia, siendo capaces de acortar distancias en pocos meses. Por ello, gran parte de la innovación se sitúa en el software y una porción importante del mismo lo constituyen las aplicaciones móviles, desarrolladas específicamente para cada plataforma, al estilo de los programas destinados tradicionalmente a ordenadores de mesa, pero con un afán de eficiencia, una mayor ligereza y, generalmente, basadas en Internet. Por lo tanto, se podría decir que las aplicaciones son las verdaderas causantes de la revolución móvil, ya que han adaptado los sitios de Internet para que pueda accederse cómodamente a ellos desde dispositivos que se transportan en el bolsillo y cuya pantalla es de apenas cuatro pulgadas. [1].

En este contexto hay que destacar la aparición de las tabletas, cuya primera generación llegó en 2010 con el lanzamiento del iPad, un dispositivo moderno, diferenciado de los anteriores tablet PC. Este producto de Apple se basa en un concepto más parecido al de un smartphone con pantalla grande - de hecho utiliza el mismo sistema del iPhone - que al de un ordenador portátil. Consecuentemente, estas tabletas usan el ecosistema de aplicaciones destinadas a los teléfonos inteligentes, lo cual amplía las posibilidades de los desarrolladores, que disponen de una pantalla más grande para adaptar sus apps. [1].

En estos momentos existen dos plataformas móviles dominantes: iOS y Android. Aunque por volumen de mercado Symbian es la más extendida, las dos anteriores son las que más pujanza están mostrando en los últimos meses. Aunque tampoco

hay que dejar de lado a BlackBerry, centrado en el entorno empresarial, y a Windows Phone, cuyo acuerdo con Nokia tiene muchas posibilidades de posicionarse en un fragmento muy grande del mercado. Pero no todo son halagos. De hecho, algunas veces se han mostrado críticas con este sistema de aplicaciones móviles implantado. Por ejemplo, Tim Berners Lee, el que está considerado como el padre de Internet, creador del sistema World Wide Web, ha señalado que la decisión de construir este tipo de apps, desarrolladas exclusivamente para cada plataforma, se encuentra en clara oposición con la creación de aplicaciones web y está fragmentando la Red. [2].

Dentro del desarrollo de las aplicaciones encontramos el término de aplicación nativa: o la que se desarrolla de forma específica para un determinado sistema operativo, llamado Software Development Kit o SDK. Cada una de las plataformas, Android, iOS o Windows Phone, tienen un sistema diferente, por lo que si se quiere que la app esté disponible en todas las plataformas se deberán crear varias apps con el lenguaje del sistema operativo (S.O.) seleccionado:

- Las apps para iOS se desarrollan con lenguaje Objective-C o Swift.
- Las apps para Android se desarrollan con lenguaje Java.
- Las apps en Windows Phone se desarrollan en .Net.

Los sistemas operativos existentes actualmente son variados, pero algunos tienen más cuota de mercado que otros, por lo que unos serán imprescindibles si se quiere crear una app móvil de éxito y otros serán prescindibles, en efecto:

Android: el sistema operativo de Google es el líder absoluto del mercado en todo el mundo, excepto quizás en Estados Unidos, donde Apple todavía tiene una gran presencia. Más del 80% de usuarios de dispositivos móviles usa Android.

iOS: subcampeón en el top de sistemas operativos más utilizados, Apple sigue a Android, y lidera con el iPad el mercado de las tablets.

Windows Phone: es el S.O. de Microsoft, que también ha querido lanzarse al mundo mobile con sus dispositivos móviles. Es el tercer sistema más utilizado pero con una cuota muy baja.

BlackBerry: con BlackBerry ocurre lo mismo, aunque el sistema operativo de esta marca sigue dando sus últimos coletazos en ciertos países.

Symbian, Ubuntu Touch y Firefox OS: estos sistemas operativos menores son objeto de unos pocos desarrolladores de apps especializados. Pero siempre tendrá que haber algún desarrollador Symbian, Ubuntu o Firefox para desarrollar esas grandes apps como la de Facebook o la de Whatsapp.

Cuando se habla de desarrollo móvil casi siempre se hace referencia a aplicaciones nativas. La principal ventaja con respecto a otros tipos, es la

posibilidad de acceder a todas las características del hardware del móvil: cámara, GPS, agenda, dispositivos de almacenamiento y otras muchas. Esto hace que la experiencia del usuario sea mucho más positiva que con otro tipo de apps.

Existen otro tipo de aplicaciones como las web o webapp, desarrolladas con lenguajes muy conocidos por los programadores, como es el HTML, Javascript y CSS. La principal ventaja con respecto a la nativa es la posibilidad de programar independiente del sistema operativo en el que se usará la aplicación. De esta forma se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones. Las aplicaciones web se ejecutan dentro del propio navegador web del dispositivo a través de una URL. Por ejemplo en Safari, si se trata de la plataforma iOS. El contenido se adapta a la pantalla adquiriendo un aspecto de navegación APP. [2].

3.1.2 Los Smartphones

Los dispositivos que están haciendo posible la transformación en el consumo de Internet y cuya influencia se está notando en diversos ámbitos de la sociedad son los Smartphones, terminales que suponen la puesta al día de las antiguas PDAs o agendas electrónicas. El salto diferencial que distingue a los antiguos dispositivos de los terminales modernos tiene lugar en 2007, cuando Apple lanza su iPhone, un teléfono inteligente que no sólo llegó acompañado de una puesta en escena impecable y un marketing agresivo, tal y como acostumbra la firma de Steve Jobs, sino que además popularizó las aplicaciones móviles. Con el iPhone ya fue posible descargar aplicaciones e instalarlas sin necesidad de conectar el terminal al ordenador de mesa. A partir de este momento el mercado se fue orientando hacia este tipo de terminales, de tal modo que Apple marcó una tendencia seguida por el resto de fabricantes de móviles. No en vano, la App Store de la compañía de la manzana no tardó mucho en ser imitada por nuevas tiendas de aplicaciones para los sistemas Symbian, Nokia, BlackBerry, RIM, y Palm OS, Palm. Pero la Apple App Store no fue la única novedad del primer iPhone, ya que la tecnología multitáctil de la pantalla del terminal representa otra de las más destacadas características de los smartphones de hoy en día. Esta capacidad facilitó la utilización del dispositivo, de tal modo que el reconocimiento de los gestos era mayor y el usuario navegaba de forma más cómoda por la interfaz del móvil, un avance que se convirtió en uno de los rasgos distintivos de estos dispositivos modernos. Desde principios de 2007, cuando el iPhone salió a la luz, el mercado se ha ido conformando con el paso del tiempo y ahora existe un buen número de fabricantes consagrados a construir hardware para este sector, al igual que una amplia comunidad de desarrolladores dedicados a programar aplicaciones para cada plataforma. [3].

3.1.3 Android

En los últimos años los teléfonos móviles han experimentado una gran evolución, desde los primeros terminales, grandes y pesados, pensados sólo para hablar por teléfono en cualquier parte, a los últimos modelos, con los que el término “medio de comunicación” se queda realmente corto. Es así como nace Android. Android es un sistema operativo -y una plataforma software- basado en Linux para teléfonos móviles. Además, también usan este sistema operativo (aunque no es muy habitual), tablets, notebooks, reproductores de música e incluso PC's. Android permite programar en un entorno de trabajo (framework) de Java, aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución). Además, lo que le diferencia de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas aplicaciones, widgets, o incluso, modificar el propio sistema operativo, dado que Android es de código libre, por lo que sabiendo programar en lenguaje Java, va a ser sencillo comenzar a programar en esta plataforma. Dado que Android está basado en el núcleo de Linux, tiene acceso a sus recursos, pudiendo gestionarlo, gracias a que se encuentra en una capa por encima del Kernel, accediendo así a recursos como los controladores de pantalla, cámara, memoria flash, entre otros, [4].

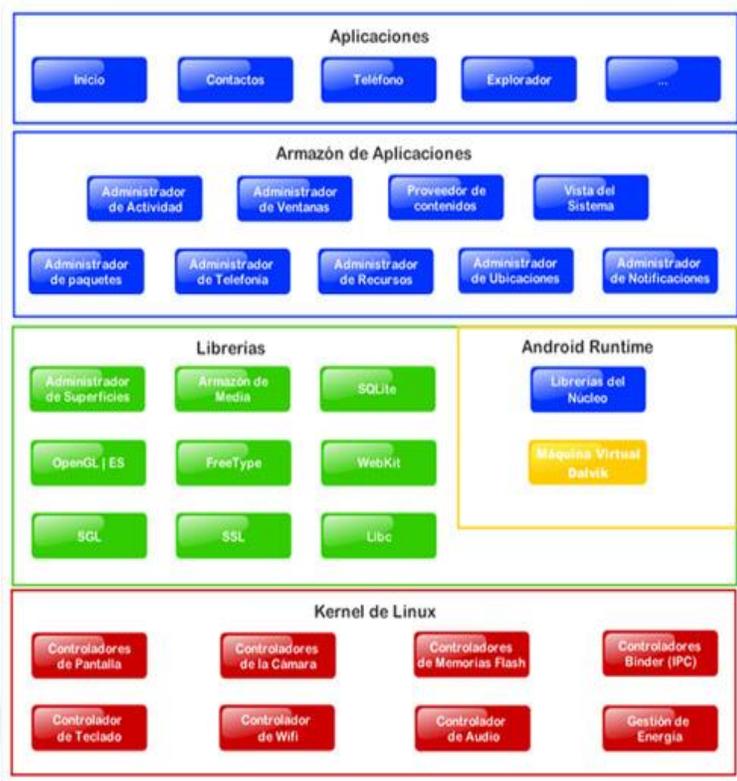


Figura 1. Sistemas de capas Android. Fuente: [5].

3.1.3.1 Versiones de Android

El sistema operativo Android, al igual que los propios teléfonos móviles, ha evolucionado rápidamente, acumulando una gran cantidad de versiones, desde la 1.0 para el QWERTY HTC G1, hasta la 5.2 [6]:

- Android Versión 1.5 Características: widgets, teclado QWERTY virtual, copy & paste, captura de vídeos y poder subirlos a YouTube directamente.
- Android Versión 1.6 Características: añade a la anterior la mejoría de la interfaz de la cámara, búsqueda por voz, y navegación en Google Maps.
- Android Versión 2.0/2.1 3 Características: mejoras en Google Maps, salvapantallas animado, incluye zoom digital para la cámara, y un nuevo navegador de internet.
- Android Versión 2.2 Características: incluye hotspot Wifi, mejora de la memoria, más veloz, Microsoft Exchange y video-llamada.
- Android Versión 2.3 Características: mejoras del consumo de batería, el soporte de vídeo online y el teclado virtual, e incluye soporte para pagos mediante NFC2
- Android Versión 3.0/3.4 Características: mejoras para tablets, soporte Flash y Divx, integra Dolphin, multitarea pudiendo cambiar de aplicación dejando las demás en espera en una columna, widgets y homepage personalizable. Ice Cream Sándwich:
- Android Versión 4.0 Características: multiplataforma (tablets, teléfonos móviles y netbooks), barras de estado, pantalla principal con soporte para 3D, widgets redimensionables, soporte usb para teclados, reconocimiento facial y controles para PS3.
- Android Versión 5.0 Android 5.0 Lollipop es una versión del sistema operativo para dispositivos móviles Android. Los cambios más prominentes en Lollipop incluyen una interfaz de usuario rediseñada construida sobre un diseño de lenguaje responsivo denominado como "Material design".
- Google presentó Android 6.0 Marshmallow, la próxima versión de su sistema operativo para dispositivos móviles, sucesor de Android Lollipop.

3.1.3.2 Entornos gráficos de desarrollo

- **Eclipse:** es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

- **Android estudio:** el núcleo de Android Studio es un editor de código inteligente capaz de finalización avanzada código, refactorización, y el análisis de código. El potente editor de código ayuda a ser un desarrollador de aplicaciones para Android más productivo. [7].
- **Basic 4 Android:** la plataforma Basic4Android es una de las grandes competidoras de Android Studio, que programa con el lenguaje Visual BASIC, también rival de Java desde sus inicios. En este sentido, Basic4Android está especialmente indicada para aquellos desarrolladores Android que empezaron a desarrollar apps en Android. La explicación se encuentra en que es un entorno más gráfico y menos abstracto, de manera que se pueden ver los avances conforme se va programando.
- **Mono para Android:** aunque Java sea el lenguaje nativo de Android, la gran fama de este sistema operativo ha conseguido que llegaran herramientas para programar apps en Android conociendo otros lenguajes de programación. Los lenguajes que Microsoft para desarrollar apps C# y .NET han llegado también a Android. Si se desea utilizar este lenguaje seguramente ya se habrá utilizado Visual Basic, por lo tanto solo resta descargar la SDK de Mono para Android y directamente se podrá desarrollar apps en Android desde el entorno Microsoft.
- **App Inventor:** ¿y si no se tiene un background técnico? ¿no se sabes ni C#, ni .NET, ni Java, ni siquiera Visual Basic? Pues entonces aparece App Inventor. Esta es una plataforma desarrollada por Google Labs para que más gente se uniera a programar apps en Android. Es un entorno totalmente visual en el que no hace falta ni escribir una sola línea de código para desarrollar apps en Android. Uno de los grandes inconvenientes es también su punto fuerte, la simplicidad y la rapidez a la hora de crear apps para Android. Es completamente gratuito, solamente hay que descargar App Inventor en la web y empezar a crear apps. [8].

3.1.4 iOS

El sistema operativo iOS (iPhone Operative System), ver tabla 1, es el sistema operativo actual de todos los dispositivos móviles de Apple, (iPhone, iPod Touch, iPad, Apple Tv), iOS está diseñado para satisfacer las necesidades de un entorno móvil, donde las necesidades de los usuarios son ligeramente diferentes.

La interfaz de usuario del iOS se basa en el concepto de manipulación directa mediante la utilización de gestos multicontacto. Los elementos de control de la interfaz consisten en deslizadores, interruptores y botones. La respuesta a las peticiones del usuario es inmediata y proporciona una interfaz fluida. La interacción con iOS incluye gestos como "tocar fuerte", "tocar de forma más débil", "sujetar" y "soltar", que tienen definiciones específicas en el contexto del sistema operativo iOS y su interfaz multicontacto [9].

Características	iOS
Modelo de desarrollo	Software propietario
Última versión	iOS 9.3.1
Escrito en	C, C++, Objective-C, Swift
Interfaz gráfica	Cocoa Touch
Plataformas soportadas	iPad, iPhone, iPod Touch
Lenguajes soportados	Adobe Flash o Java
Método de actualización	Mediante iTunes

Tabla 1: Especificaciones del sistema operativo iOS. Fuente: [9].

3.1.4.1 Arquitectura iOS

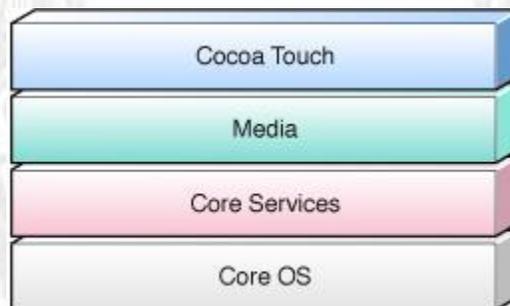


Figura 2. Modelo de arquitectura de iOS. Fuente: [10].

Cocoa Touch

Como se muestra en la figura 2, esta capa contiene los Frameworks necesarios para la creación de aplicaciones iOS, define la infraestructura de las aplicaciones y brinda soporte para tecnologías clave como la multitarea, entrada de datos táctil, notificaciones y otros servicios de alto nivel del sistema. En definitiva, esta capa se encarga de gestionar la interacción visual con el usuario [10][11].

Capa de Medios (Media Layer)

La capa de medios permite la ejecución de tareas multimedia, como la visualización de gráficos, reproducción de audio y video.

Capa de Servicios (Core Services)

Esta capa permite el acceso a los servicios fundamentales del sistema, tales como la agenda, el acceso a la base de datos, preferencias, conexión a servidores, procesamiento de URLs, etc.

Nucleo (Core Services)

Esta capa es la base del sistema operativo iOS. Se encarga de realizar la gestión de memoria, el sistema de ficheros, conexión a la red y procesos que interactúan con el hardware. Las interfaces de esta capa están en su mayoría basadas en el lenguaje C. Las características de esta capa son raramente accedidas directamente por los desarrolladores, pero son comúnmente utilizadas por los Frameworks de capas superiores.

3.1.5 Herramientas de desarrollo y Frameworks

Apple ofrece la mayoría de sus interfaces de sistema en paquetes especiales llamados frameworks. Éstos son directorios que contienen una biblioteca dinámica compartida y recursos como: archivos de encabezado, imágenes, etc. Para que el desarrollador pueda utilizar las características de los frameworks, únicamente necesita vincularlos al proyecto y obtendrá acceso completo a sus recursos [12].

iOS SDK

El SDK de iOS contiene todas las herramientas necesarias para diseñar y crear el software para iOS [12]. Está disponible gratuitamente para los desarrolladores. Sin embargo, para acceder a la descarga del SDK es necesario registrarse como desarrollador Apple a través de un equipo con sistema operativo OS.

IDE XCode

Es el entorno de desarrollo integrado (IDE) que gestiona los proyectos de las aplicaciones, permitiendo editar, compilar, correr y depurar el código. XCode es la aplicación principal que se utiliza para el desarrollo.

3.2 Metodología de desarrollo

El proceso metodológico para alcanzar cada uno de los objetivos, y las diferentes actividades que definen el procedimiento para llegar a la solución, se realizará con una descripción individual de estas actividades donde se plantee a manera de plan el desarrollo de cada uno de los objetivos.

El primer objetivo específico consiste en generar un reporte de instalación de los sistemas para desarrollo de aplicaciones en iOS y Android. Es importante realizar un reporte general donde se ilustre cada uno de los requerimientos del sistema y

procedimientos para la instalación de las herramientas necesarias. Igualmente, es importante identificar los lenguajes y los Framework principales en los que se desarrolla aplicaciones para las diferentes plataformas, realizar una descripción breve de cada uno de ellos, y especificar cuál es la función de cada uno dentro del proceso de construcción de aplicaciones.

El segundo objetivo específico es documentar el contenido básico a manera de introducción o resumen sobre los lenguajes de programación necesarios para la creación de aplicaciones móviles. Para resolver la forma en cómo se abarca este objetivo es necesario un contenido temático en donde se muestre la manera en que se maneja la sintaxis (variables, métodos o funciones, clases, etc.) usadas por los lenguajes de programación.

En el tercer objetivo, que consiste en generar un informe sobre las herramientas y componentes usados en la interfaz gráfica de usuario, es importante identificar y describir los elementos principales del Framework, y la manera en que estos elementos se vinculan con eventos activados por el usuario.

Para el cuarto objetivo, que consiste en elaborar aplicaciones básicas que muestren las principales características del ambiente general de desarrollo de aplicaciones para las diferentes plataformas, se realizará de manera secuencial con la guía, una serie ejemplos sencillos dejando en evidencia el cumplimiento de las etapas y la comprensión del contenido necesario para el desarrollo de la aplicación ejemplo.

4. Resultados

4.1 Guía Android

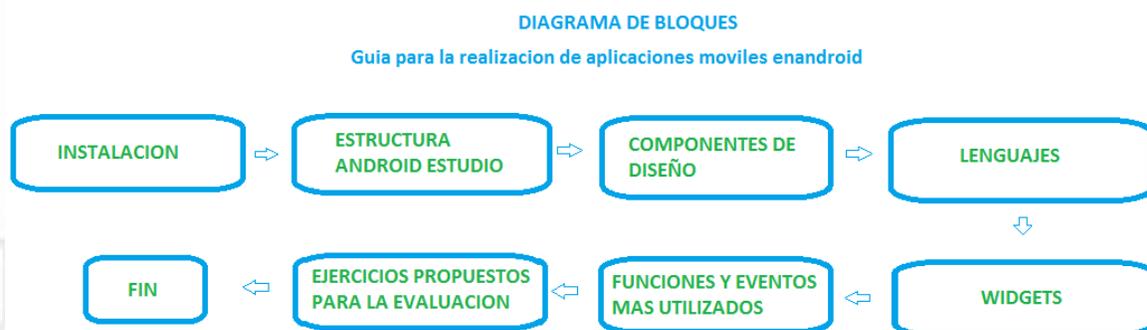


Diagrama 1. Diagrama de bloques guía Android.

GUIA ANDROID



Diagrama 2. Diagrama de contenido guía Android.

4.1.1 Instalación

Requerimientos para el desarrollo de Aplicaciones en Android

Para desarrollar aplicaciones en Android, se debe contar con una serie de programas y configuraciones que permitan manejar las funcionalidades del sistema operativo. El equipo donde instalar el Android SDK necesitará lo siguiente [12]:

- 2GB RAM como mínimo, 4 GB RAM recomendado
- 400 MB de espacio en disco duro
- Al menos 1 GB para el Android SDK
- Resolución de pantalla mínima 1280 x 800 pixeles

Windows:

- Microsoft Windows 8/7/Vista (32- o 64-bit).
- Java Development Kit (JDK) 7

Linux:

- GNOME o KDE desktop
- GNU C Library (glibc) 2.15 o mayor.
- Oracle Java Development Kit (JDK) 7
- 64-bit Capa de distribución para aplicaciones de 32-bits

Mac OS X:

- Mac OS X 10.8.5 o mayor que 10.9 (Mavericks)
- Java Runtime Environment (JRE) 6

El emulador de Android y algunas herramientas de depuración están basados en JAVA y para ejecutarse requieren el JDK. Este software es gratuito y se puede descargar en el sitio de Oracle [13]. A continuación se presentan aquellos programas exigidos por el API nativo de Android, y que generalmente son necesarios para poder trabajar con los diferentes frameworks, y el proceso de instalación de éstos se encuentra de forma detallada en la guía:

Android SDK

El SDK de Android, figura 3, es el encargado de proporcionar las bibliotecas API y las herramientas de desarrollo necesarias para crear, probar y depurar aplicaciones para Android [13]. Contiene una lista con todas las versiones del sistema operativo que han sido desarrolladas hasta el presente, así como también utilidades y funcionalidades especiales de los mismos.

El SDK es el encargado de compilar la aplicación con todos los recursos y ficheros de la misma en un único archivo ejecutable con extensión .Apk. Una vez generado dicho archivo, se podrá instalar la aplicación en cualquier terminal Android compatible.

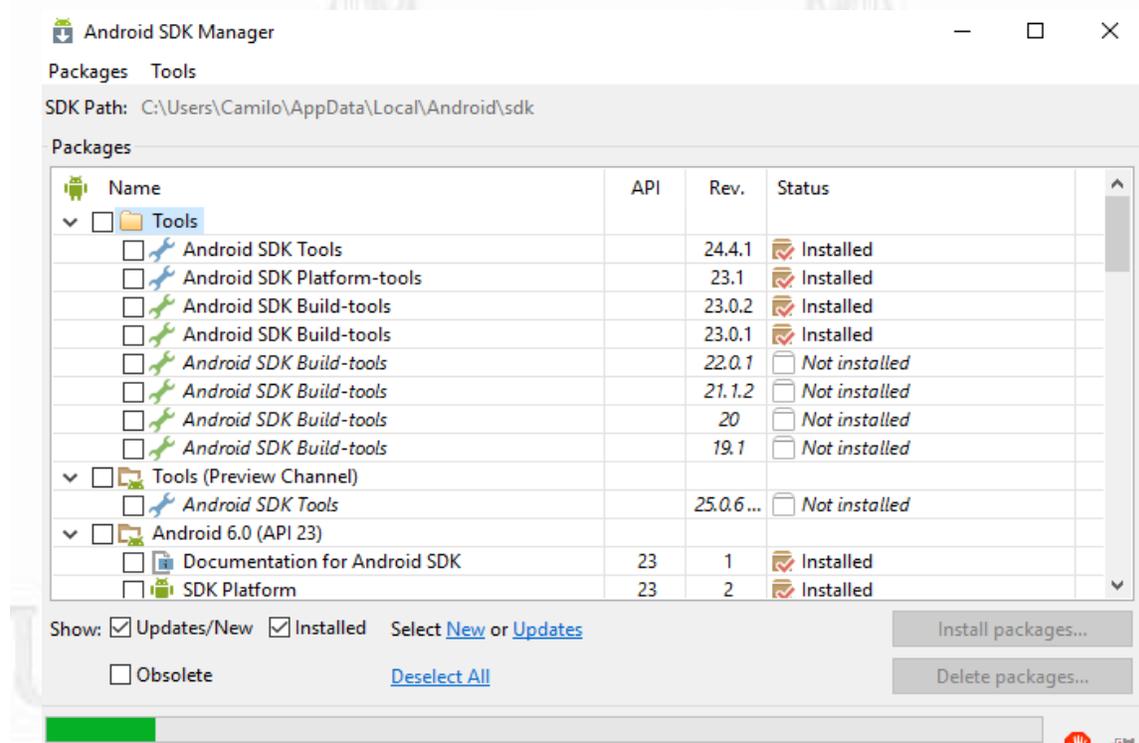


Figura 3. Interfaz SDK de Android.

IDE Android Studio

Un entorno de desarrollo integrado (IDE por sus siglas en inglés), es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI) que proporciona servicios integrales a los programadores para el desarrollo de software [14].

El Android estudio, figura 4, se basa en un potente editor de código y herramientas para desarrolladores, teniendo mayores características para el rendimiento y construcción de aplicaciones como lo es el sistema de edición gráfica de arrastre de componentes, compatibilidad de versiones, la generación de apks y la fácil instalación del programa.

El tamaño promedio de la descarga del Android Studio es aproximadamente 450MB y contiene todas las librerías necesarias para comenzar a crear aplicaciones Android y la instalación es bastante intuitiva.



Figura 4. Interfaz Inicio Android Studio.

Para el desarrollo de la guía se usó el IDE Android Studio bajo un sistema operativo Windows.

4.1.2 Temáticas de Aprendizaje.

El enfoque que abarca las guías de programación se realizan sobre conceptos básicos necesarios para empezar en el desarrollo de aplicaciones. En el caso de programación en Android se realizó una división de ítems para abarcar los temas principales en el desarrollo de aplicaciones bajo Android Studio. Estos ítems

principales son:

- **Introducción a Android :**
Dentro de la guía se encuentra una breve introducción al sistema operativo Android, donde se presenta un corto resumen de las características, parte de su historia y la arquitectura de Android.
- **Uso de Emuladores y dispositivos:**
En esta sección se muestra la forma de presentar una aplicación con el emulador AVD de Android, figura 5. Éste proporciona una vista especial para comprobar si la aplicación hace lo que se desea. Luego de esto de enseña como subir un APK a un dispositivo Android.

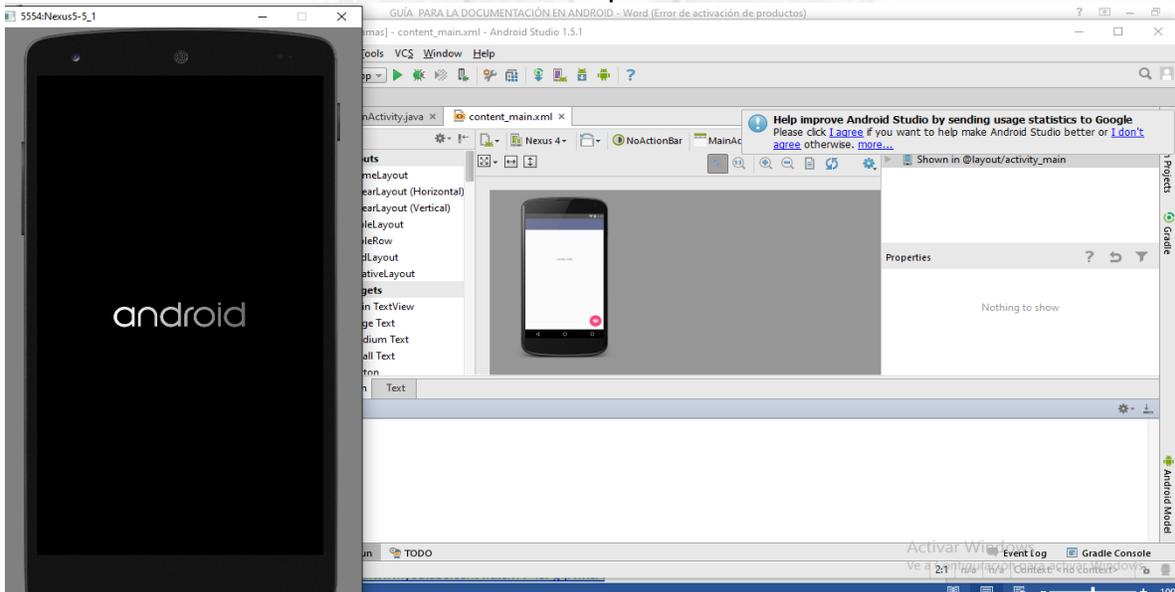


Figura 5. Emulador AVD Android.

- **Actividades e intents**
Este ítem hace referencia a una de las características principales del diseño en Android, la reutilización de componentes entre las aplicaciones, es decir, dos aplicaciones diferentes pueden utilizar una misma componente, aunque esté en otra aplicación para así evitar la repetición innecesaria de código, y la consiguiente ocupación de espacio. Los componentes son los elementos básicos con los que se construye el proyecto. Las aplicaciones se componen principalmente de actividades. Habrá tantas actividades como ventanas distintas tenga la aplicación. Sin embargo, por si solos, los componentes no pueden hacer funcionar una aplicación. Para ello están los intents. Todos ellos deben declararse en el AndroidManifest.xml con el mismo nombre que lleve la clase asociada.

Esta sección está enfocada en el modo de usar y declarar las actividades, servicios e intents dentro de una aplicación en Android.

- **Widgets y contenedores**

La interfaz de usuario es la interacción principal entre el usuario y el dispositivo, por esto se realizó una sección dedicada a los componentes principales en una interfaz gráfica del usuario GUI y los eventos que estos componentes activan en la aplicación.

- **Base de datos SQLite**

En esta sección se explica cómo manejar BBDD en SQLite. SQLite es un motor de bases de datos que se ha ido popularizando en los últimos años dado que maneja archivos de poco tamaño, no necesita ejecutarse en un servidor, cumple el estándar SQL-92 y, además, es de código libre. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un único fichero en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción. Esta es, al mismo tiempo, su gran virtud y su mayor inconveniente, ya que gracias a ello dispone de unas latencias muy bajas, pero también impide el acceso múltiple a la base de datos.

La manera más fácil de acceder a la información de una base de datos local es creando una clase que herede de SQLiteOpenHelper sobre la que se debe adaptar/sobreescribir los métodos proporcionados para obtener la funcionalidad con la base de datos. Básicamente en esta clase se definen los atributos de la base de datos y el comportamiento ante creación y actualización de la misma. Los métodos que se sobrescriben son onCreate() y onUpgrade().

Con la clase heredada SQLiteOpenHelper, se pueden utilizar los métodos getReadableDatabase() o getWritableDatabase() para acceder a la base de datos en modo lectura o lectura/escritura.

- **Multimedia**

En la guía se explica cómo usar las clases necesarias para reproducir audio, vídeo o imágenes en los formatos más usados, pudiendo interactuar con archivos contenidos en el propio terminal Android, con archivos externos al dispositivo, o con aquellos dispuestos a través de Internet. Para la reproducción utiliza MediaPlayer o JetPlayer, siendo MediaRecorder el usado para grabar audio o vídeo, o hacer fotografías a través de la cámara del terminal. Android soporta varios formatos de audio, video e imágenes. Algunas de ellas como mp3, midi y flac para audio; 3gp y mpeg para videos.

- **Mapas y ubicaciones**

Para poder usar los mapas, Android provee una librería externa que se encuentra en el paquete com.google.maps. Para poder acceder a los datos desde el MapView es necesario registrarse en el servicio de Google Maps y aceptar los términos de uso, con esto se obtendrá una clave alfanumérica que nos dará acceso.

El registro para obtener la clave consta de dos partes:

1. Registrar la huella digital MD5 de la aplicación para que pueda acceder a los datos de Google Maps.
2. Agregar una referencia a la clave en cada MapView (en el XML o en código).

En esta sección se explicó el funcionamiento del GPS, la forma de obtener los datos de él, y la manera en que se configura una aplicación para usar Api de maps de Google.

4.1.3 Ejemplos de aplicaciones

A modo de práctica, figura 6, para el seguimiento de la guía, se realizaron una serie de ejemplos enfocados en las distintas temáticas abordadas. En primer lugar se mostrará y explicará con detalle la manera de crear una aplicación estática, es decir, sin ninguna funcionalidad. De esta manera se pretende que se consiga entender de manera sencilla y clara la estructura de este tipo de componentes.

Una vez entendida la estructura de una aplicación, se añadirán funcionalidades más avanzadas, al seguir el contenido de la guía.



Figura 6. Aplicación de ejemplo Android.

Dentro del desarrollo de la guía se realizaron diferentes ejemplos en cada uno de los temas, así como se dejaron ejercicios propuestos para su respectiva evaluación. Los ejemplos de los temas propuestos fueron actividades e intents, widget y contenedores, base de datos SQLite, servicio web en Android, sincronización de datos, multimedia, mapas y ubicaciones.

Actividades e intents

En este ejemplo se realizó una calculadora donde se utilizaron los ítems y componentes explicados como lo son los layout, botones, cajas de texto, label, y creaciones de actividades realizando una calculadora que correspondió con los anteriores componentes.

En la siguiente imagen, figura 7, se muestra la estructura de los layout con sus respectivos componentes

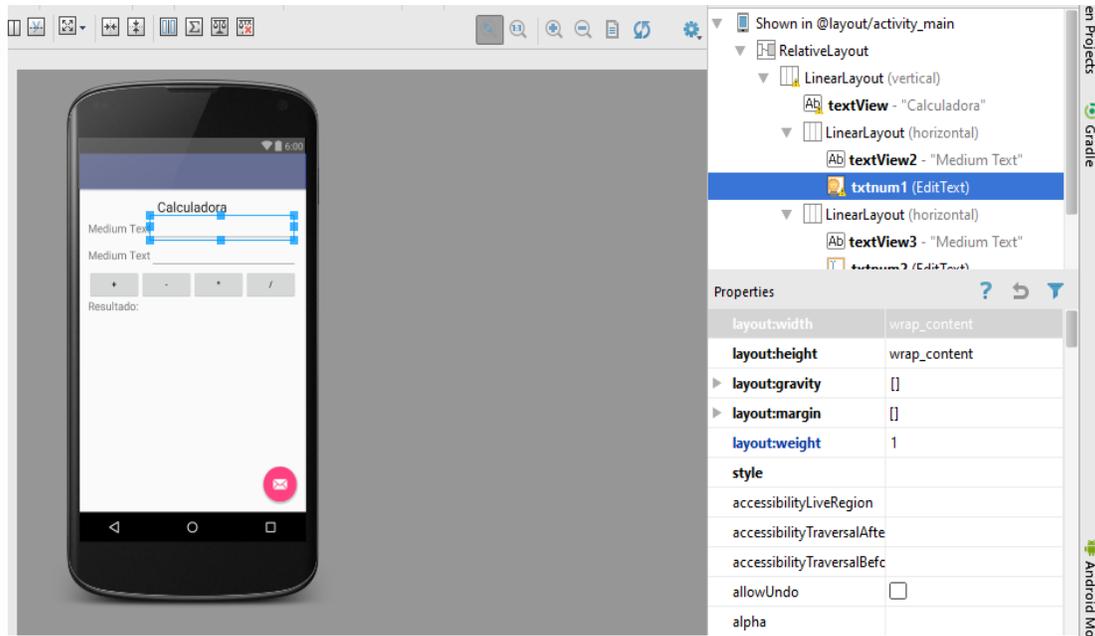


Figura 7. Composición de ejemplo actividades e intents.

Se obtuvo el siguiente resultado, figura 8, al compilar el ejemplo anterior.

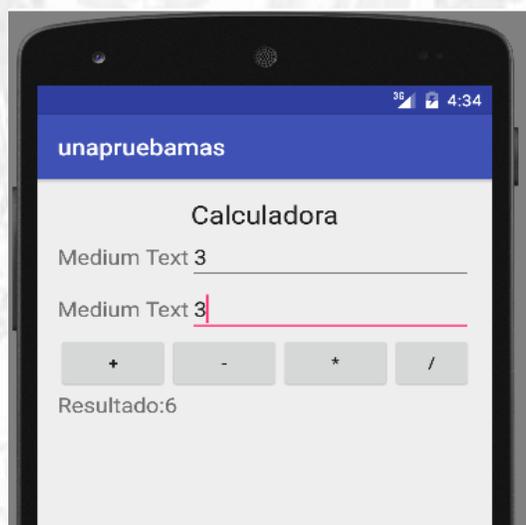


Figura 8. Ejemplo calculadora.

Otro ejemplo que se visualiza es el de creación de una nueva actividad, al dar click sobre un botón nos lleva automáticamente a otra actividad como se muestra a continuación, figura 9.

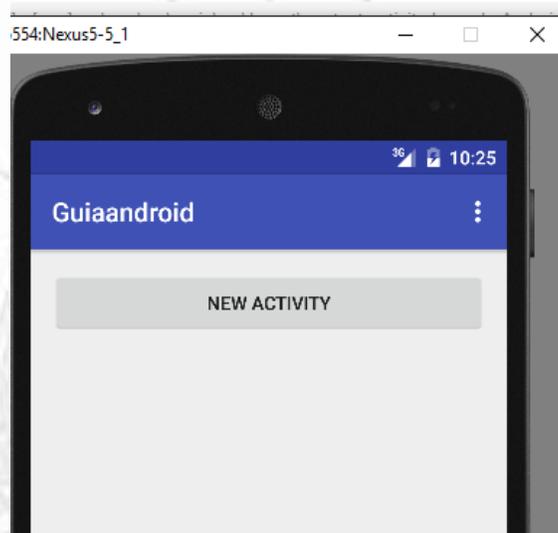


Figura 9. Activity principal.

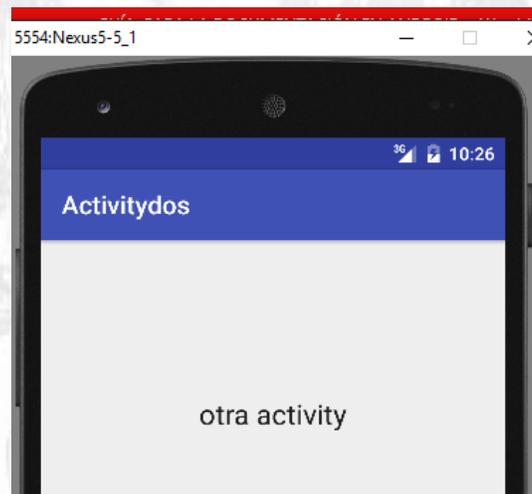


Figura 10. Activity secundaria.

- **base de datos SQLite.**

Se usa especialmente cuando se necesita almacenar datos en las aplicaciones, como por ejemplo datos de contacto. SQLite es la base de datos más popular en dispositivos móviles.

El ejemplo que se realizó para la utilización de base de datos fue crear un usuario y contraseña con una interfaz con dos cajas de texto y tres botones en

los cuales se puede dar new para agregar, editar y eliminar haciendo el manejo de la base de datos a través del SQLiteOpenHelper y el uso de las clases, como se muestra a continuación, figura 11-13.

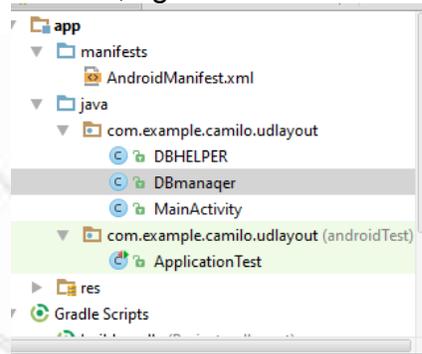


Figura 11. Clases para el manejo SQLite.

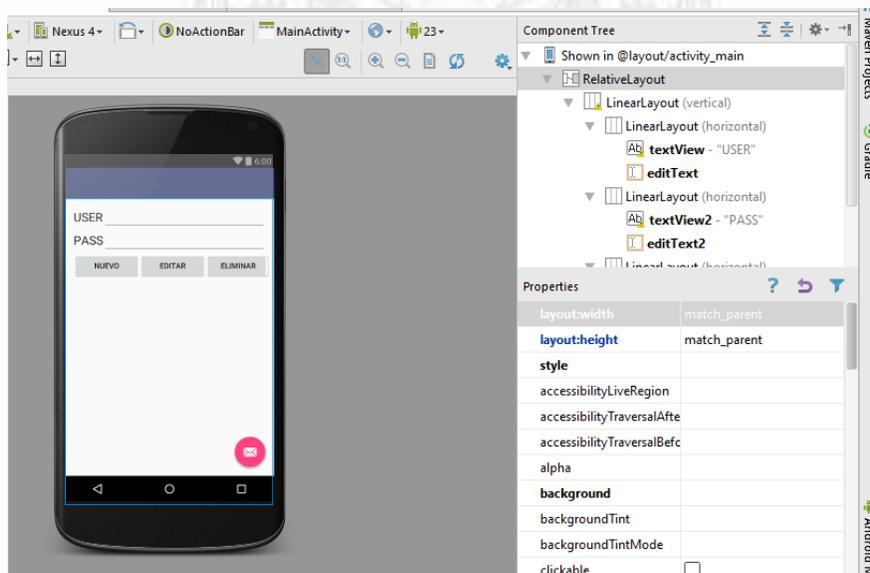


Figura 12. Interfaz SQLite.

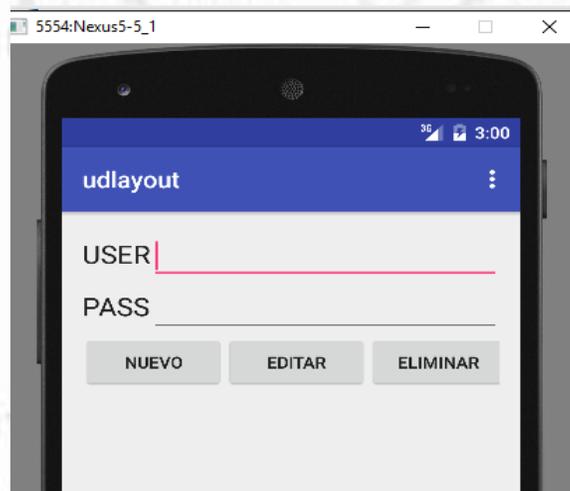


Figura 13. Emulador ejemplo SQLite.

Mapas y ubicaciones

Otro ejemplo representativo para el manejo de aplicaciones fue el uso de ubicación mediante el mapeo y posicionamiento de google. Para el uso de mapas es necesario haber instalado google play services en el SDK, como se muestra a continuación, figura 14.

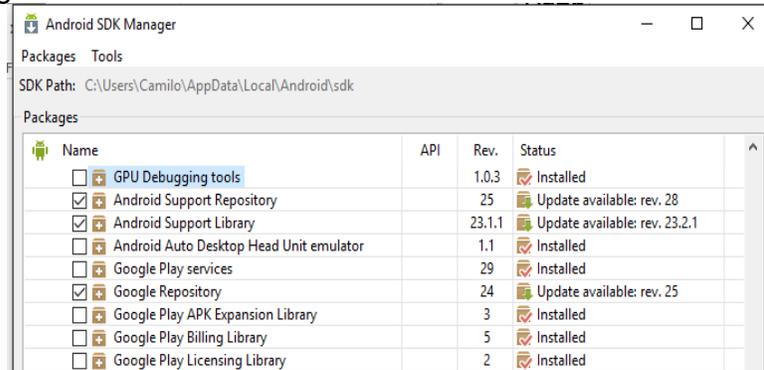


Figura 14. Instalación de herramienta google play services.

Se debe habilitar el api de google maps, para ello debe establecer el login con una cuenta de google, e ingresar a <https://console.developers.google.com/apis/library>, y determinar ciertos procesos para acceder al API mediante una clave; estos procedimientos se explican claramente en la guía, los cuales son necesarios antes de realizar la programación de google maps. La siguiente imagen, figura 15, representa el enlace de la configuración.



Figura 15. Conexión de api para acceder a los servicios de google.

La guía establece los parámetros de configuración en el manifest y para la creación y llamado del `maps.View.`, y cómo se puede manejar el posicionamiento por medio de ubicación o simplemente utilizar el maps como recurso, figura 16.



Figura 16. Emulador de google maps.

4.1.4 Ejercicios propuestos

Algunos de los ejercicios propuestos se muestran a continuación. Estos ejercicios ayudan a reforzar las bases adquiridas por el estudiante en el transcurso de los temas dados en la guía. Cada tema de Android expuesto en la guía, lleva un ejemplo y un ejercicio propuesto para la evaluación de la misma.

- Realizar la carga de tres números en controles de tipo EditText. Realizar las cuatro operaciones básicas mediante botones (suma, resta, multiplicación y división) y publicar en un textView el resultado.
- Disponer 9 botones en forma de un tablero. Utilizar un TableLayout, tres TableRow y nueve botones.
- Crear un proyecto donde se pueda disponer de tres objetos de la clase Button con las etiquetas "Grabar", "Detener Grabación" y "Reproducir Grabación". Disponer además un TextView para informar del estado actual. Cuando se presione el botón "Grabar" permitir registrar todos los sonidos hasta que se presione el botón "Detener Grabación". Cuando se presione el botón "Reproducir Grabación" emitir el archivo de audio previamente generado. [15].
- Crear una aplicación que en su actividad principal deje dibujar círculos con la función de evento touch
- Incorporar una aplicación que tenga como logo la imagen de la Universidad Distrital y de contenido el horario de clase
- Mostrar la ubicación de la universidad distrital Francisco José De Caldas Facultad de tecnología utilizando los ejemplos de mapas y ubicaciones.

- Cargar el contenido de una base de datos al oprimir un botón en una tabla GridView donde se visualizan todos los campos de columnas creados con la información correspondiente
- Eliminar y modificar datos de la base de datos con los botones de eliminar y editar utilizando los métodos explicados en el capítulo SQLite, verificando la base de datos con el complemento de mozilla SQLite manager.

4.2 Guía iOS

4.2.1 Instalación

Requerimientos para el desarrollo de Aplicaciones en iOS

A continuación se describen los componentes necesarios para la creación de aplicaciones iOS [15]:

- **Equipo Mac con procesador Intel:** Cualquier modelo de los disponibles actualmente (Mac Mini, MacBook Air, MacBook Pro o iMac), con al menos 4 GB de RAM, a mayor RAM más fluidez en el Mac.
- **XCode 6 o superior:** Las versiones anteriores del IDE de Apple para programar aplicaciones iOS o OS X no soportan Swift, por lo que solo se pueden compilar aplicaciones programadas en Objective-C [16].
- **OS X 10.9 Mavericks como mínimo:** Aunque es posible programar aplicaciones con Mavericks, es recomendable usar OS X 10.10 Yosemite

La primera opción para realizar programación para iOS sin contar con un ordenador Mac, es instalar y ejecutar una máquina virtual para instalar un sistema operativo OS X 10.10 Yosemite, en el que se instalará el Xcode; y la segunda es alquilar un servicio conocido como cloudcomputing, que nos permite administrar un ordenador remoto a través de Internet, en este caso ordenadores Mac [17].

El XCode se obtiene de forma gratuita desde la App Store; para descargarlo se debe crear un ID de Apple, éste se puede adquirir al registrarse en la página de Mac, aunque para realizar esto se debe hacer la inscripción desde un sistema operativo OS o iOS. La instalación del Xcode es muy sencilla de realizar y no presenta ningún problema si se cumple con los requisitos del sistema y se aceptan los términos de condiciones, figura 17.

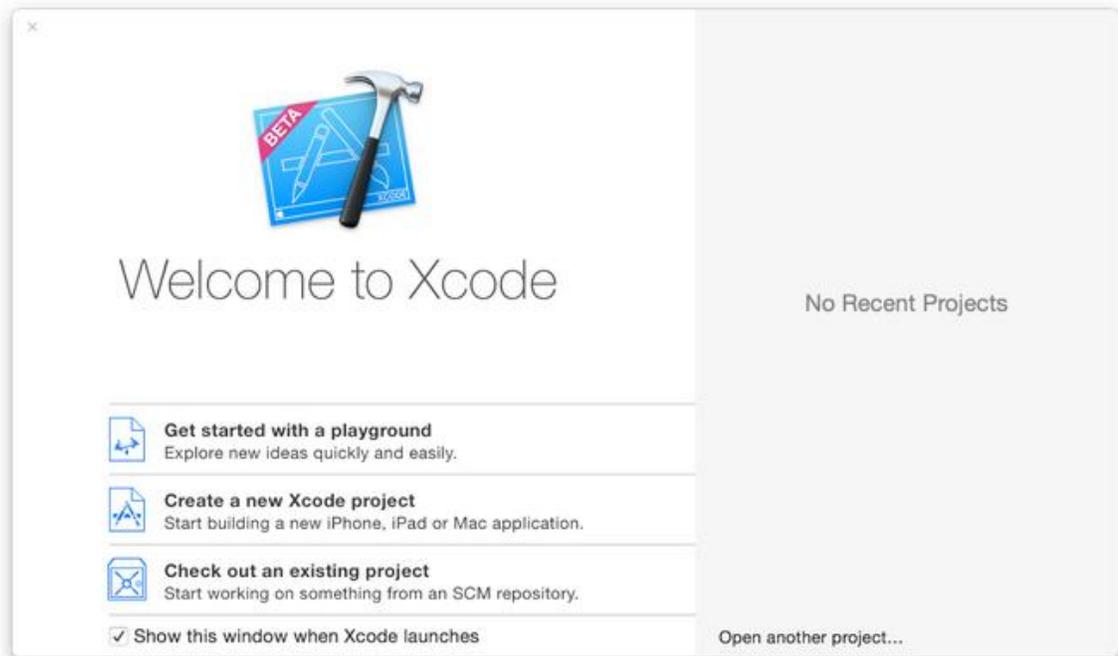


Figura 17. Interfaz de inicio de Xcode.

4.2.2 Metodología pedagógica

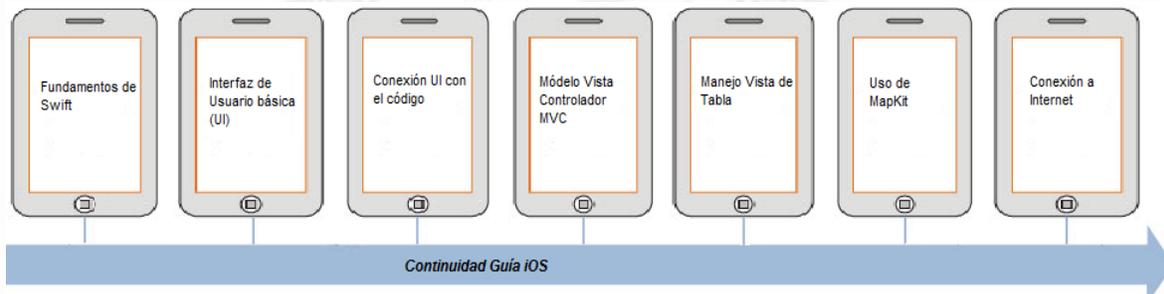


Diagrama 3. Diagrama de bloques guía iOS.

Para abordar los temas principales en desarrollo de aplicaciones bajo el entorno iOS se decidió implementar la guía con una segmentación con ejemplos prácticos que abarcan las siguientes categorías:

- **Fundamentos de Swift**

Es una sección dedicada a Comprender la sintaxis del lenguaje de programación Swift, su sintaxis, los métodos y clases usados, también se encuentra una breve descripción de XCode.

Herramientas: Computador con Xcode 6.1 o superior, Guía para la realización de aplicaciones en iOS

Desempeños:

- I. Realizar una investigación documental y elaborar una tabla en la que describa las diferencias entre Swift y Objective C.
- II. Realizar la programación en pseudocódigo de 3 problemas propuestos, empleando los conceptos relacionados con las estructuras de datos y estructuras de control del lenguaje Swift, mostrando el resultando en la consola de Xcode por medio de la sentencia print.
 1. Secuencia: Intercambiar los valores de dos variables tipo carácter.
 2. Iteración: Escribir un algoritmo para imprimir los números del 1 al 100.
 3. Funciones: Crear un programa que contenga una función que reciba dos números enteros como parámetros y retorne la suma de éstos.

- **Interfaz gráfica y Storyboard**

En este tema se abarca la integración de Swift con la interfaz de usuario y el uso de la librería UIKit, que contiene clases útiles para trabajar con la capa de Cocoa Touch. Para la familiarización con IDE Xcode se crea un nuevo proyecto y se implementan componentes gráficos apropiados para la creación de una interfaz de usuario básica.

En esta sección se estudia Storyboard de Xcode con el que se puede diseñar visualmente todos los controladores de vista y describir las conexiones y transiciones entre las distintas escenas. Estas transiciones se denominan "segues" y se crean mediante la conexión de los controladores de vista justo en el guión gráfico. Gracias a los segues se necesita menos código para cuidar de la interfaz de usuario, con un guion se tiene una mejor visión general conceptual de todas las escenas en la aplicación.

Herramientas: Computador con Xcode 6.1 o superior, Guía para la realización de aplicaciones en iOS.

Desempeños:

- I. Crear un proyecto Swift que permita habituarse a los componentes de Xcode en la creación de una interfaz de usuario, siguiendo los pasos de la guía para la realización de aplicaciones en iOS.
- II. Realizar un ejemplo de aplicación, donde se tenga un TabBarController inicial con dos pestañas. En la primera de ellas se tendrá un botón que abre una ventana en la cual hay un botón que permite volver a cerrarla. En la segunda pestaña se tendrá un TableViewController con su correspondiente

pantalla de detalle.

- III. Crear una aplicación que permita pasar 2 variables del tipo String de un ViewController a otro utilizando los StoryBoards, en la primera escena se debe pedir los datos Nombre y Apellido y pasarlos a la segunda escena.

- **Modelo Vista Controlador**

La base sobre la que se sustenta el desarrollo en iOS para las apps y uno de los modelos de desarrollo más trabajados es el modelo vista controlador (MVC). Por esto, se implementó en esta sección el contenido para entender y manejar el concepto MVC en los diseños de las aplicaciones que permiten estructurar una aplicación en 3 componentes principales:

1. La vista: Es la interfaz gráfica y todos los elementos que la componen: etiquetas, botones, campos, tablas, imágenes, etc.
2. El modelo: Donde se almacena aquello que procede o se muestra en la vista. Contiene el código que da la funcionalidad a esta.
3. El controlador: es lo que conecta la interfaz con el correspondiente código que da la funcionalidad. Si se pone un dato en la vista, el controlador lo recoge y lo envía al modelo.

Herramientas: Computador con Xcode 6.1 o superior, Guía para la realización de aplicaciones en iOS.

Desempeños:

- I. Incluir un controlador de vista a la aplicación creada en la lección anterior, para comprender el manejo y conexión entre Vistas del Storyboard y los controladores de eventos.
- II. Realizar un ejemplo de aplicación en donde se pida al usuario peso y estatura, que permita calcular el índice de masa corporal al pulsar un botón, implementando el patrón de diseño MVC.

- **Tablas**

Sección Enfocada en organizar el contenido sobre celdas y tablas, en donde se explica cómo diseñar celdas para la correcta distribución de la información e Identificación de los diferentes tipos de celdas y componentes de visualización.

El componente TableView es utilizado en la mayoría de las aplicaciones ya sea para mostrar registros, posts, opciones, o simplemente listas de información.

Herramientas: Computador con Xcode 6.1 o superior, Guía para la realización de aplicaciones en iOS.

Desempeños:

- I. Agregar un componente UITableViewController que permita visualizar información obtenida por el usuario en una escena diferente, para comprender el papel de los delegados de vista tabla y fuentes de datos.
- II. Realizar un ejemplo de aplicación usando TableView que muestre al usuario una lista de canciones.

- **Mapas**

El objetivo de esta sección de la guía es desarrollar apps utilizando ubicación y mapas, aprender a usar correctamente los conceptos de geolocalización y el uso de la información basada en la ubicación para mantener al usuario conectado con el mundo circundante.

Herramientas: Computador con Xcode 6.1 o superior, Guía para la realización de aplicaciones en iOS.

Desempeños:

- I. Agregar una escena que usando Mapkit, para la visualización de mapas en la aplicación que se ha construido, siguiendo la guía para la realización de aplicaciones en iOS.
- II. Realizar un ejemplo de aplicación usando MapKit que muestre un mapa centrado en la ubicación del usuario.

- **Conexión**

Tema realizado para identificar implementar servicios para acceder a la información que se encuentra en un servidor mediante conexiones de internet.

Herramientas: Computador con Xcode 6.1 o superior, Guía para la realización de aplicaciones en iOS.

Desempeños:

- I. Crear una vista en la aplicación que se ha creado con la guía con capacidad de conectarse a un servidor de datos, procesarlos, almacenarlos y visualizarlos, usando servicios web del tipo RESTful. Procesar los datos con el formato JSON y visualizar los dato en una tabla.
- II. Crear una aplicación ejemplo de aplicación que se conecte a los servicios de clima de Yahoo, y los muestre de forma tabular.

5. Conclusiones

1. Durante el desarrollo de las guías, se obtuvieron fácilmente las herramientas de desarrollo; pero se hace evidente la flexibilidad ofrecida por Android al permitir que su plataforma esté disponible para diferentes plataformas (Windows, Linux, Mac); Mac, por otro lado, sólo permite el desarrollo de aplicaciones en sistemas de Apple. Actualmente se combate el diseño multiplataforma con el diseño de aplicaciones híbridas que aprovechan al máximo la versatilidad de un desarrollo web y tienen la capacidad de adaptación al dispositivo como una app nativa. Otra forma de abarcar aplicaciones multiplataforma es usar herramientas diseñadas para esto como Adobe Flex/Builder que usa ActionScript como lenguaje de programación y librerías que permiten acceder a componentes de los dispositivos como cámaras y sensores, con reutilización de código [18].
Por otro lado, existe una manera de implementar apps nativas de iOS en Android por medio de Cider, creado por estudiantes de la Universidad de Columbia. Cider funciona como una capa de compatibilidad entre los kernel de los dispositivos al copiar librerías y frameworks de iOS, así el código en ejecución cree que se está ejecutando en un kernel de iOS, cuando en verdad está bajo el kernel Android [19] [20].
2. El Sistema Operativo Android es el que mayor impulsa el desarrollo de software, y está perfilado como el dominar del mercado en dispositivos móviles, debido a su libre desarrollo.
3. Las características de los dispositivos se aprovechan mejor en la plataforma de Android debido a que no maneja restricciones y es de uso libre.
4. Se pueden adquirir conocimientos básicos con la ayuda de esta guía para el desarrollo de app tanto en Android como IOS, y ser complementada con cursos avanzados para mejorar potencial en la competitividad para el desarrollo de las aplicaciones móviles
5. Se vuelve evidente que carreras afines al uso de software, como son las ingenierías en telecomunicaciones e ingenierías en Sistemas, abarcan un alto porcentaje del mercado laboral, por lo que el uso y desarrollo de aplicativos móviles en los diferentes sistemas operativos que se manejan en el mercado deben ser aprendidos.

6. Referencias

1. INNOVA, Informes y tendencias
http://www.innovanube.com/docs/ticbeat%20%20desarrollo_de_apliaciones_moviles.pdf
2. BÁEZ, M. BORREGO, A. CORDERO, J. (Grupo Tecnología UCM). Introducción a Android. <http://www.it-docs.net/ddata/18.pdf>
3. Campo, Celeste - García Rubio, Carlos. <http://ocw.uc3m.es/ingenieria-telematica/aplicaciones-moviles/material-de-clase-2/sistemas-operativos>
4. Revelo, James. 7 de Agosto de 2014.
<http://www.hermosaprogramacion.com/2014/08/aprendiendo-la-arquitectura-de-android/>
5. Bastera - Berteza - Borello - Castillo – Venturi
<http://androidos.readthedocs.io/en/latest/data/caracteristicas/>
6. <http://www.androidcurso.com/index.php/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/146-las-versiones-de-android-y-niveles-de-api>
7. <http://developer.android.com/intl/es/sdk/index.html>
8. Mocholi, A.(desarrollador de apps).20 de mayo de 2015.Yeeply
<https://www.yeeply.com/blog/entornos-programacion-desarrollar-apps-android/>
9. <https://es.wikipedia.org/wiki/IOS>
10. <http://www.programadorphp.org/blog/cursos/introduccion-al-ide-de-xcode-objective-c-ios/>
11. <http://www.epinom.com/blog/ios-development-primeros-pasos/>
12. <http://developer.android.com/intl/es/tools/studio/index.html>
13. <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
14. <https://fergarciaac.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>
15. <http://www.javaya.com.ar/androidya/androidstudioya/index.php?inicio=0>
16. <http://www.javaya.com.ar/androidya/>
17. <http://www.aprendeswift.com/2014/06/04/que-necesitamos-para-empezar-a-programar-en-swift/>
18. <http://www.adobe.com/es/products/flex.html>
19. <http://www.androidcrush.com/run-ios-apps-on-android/>

20. <http://www.xatakamovil.com/sistemas-operativos/cider-la-ejecucion-de-aplicaciones-ios-en-android-es-posible>



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS