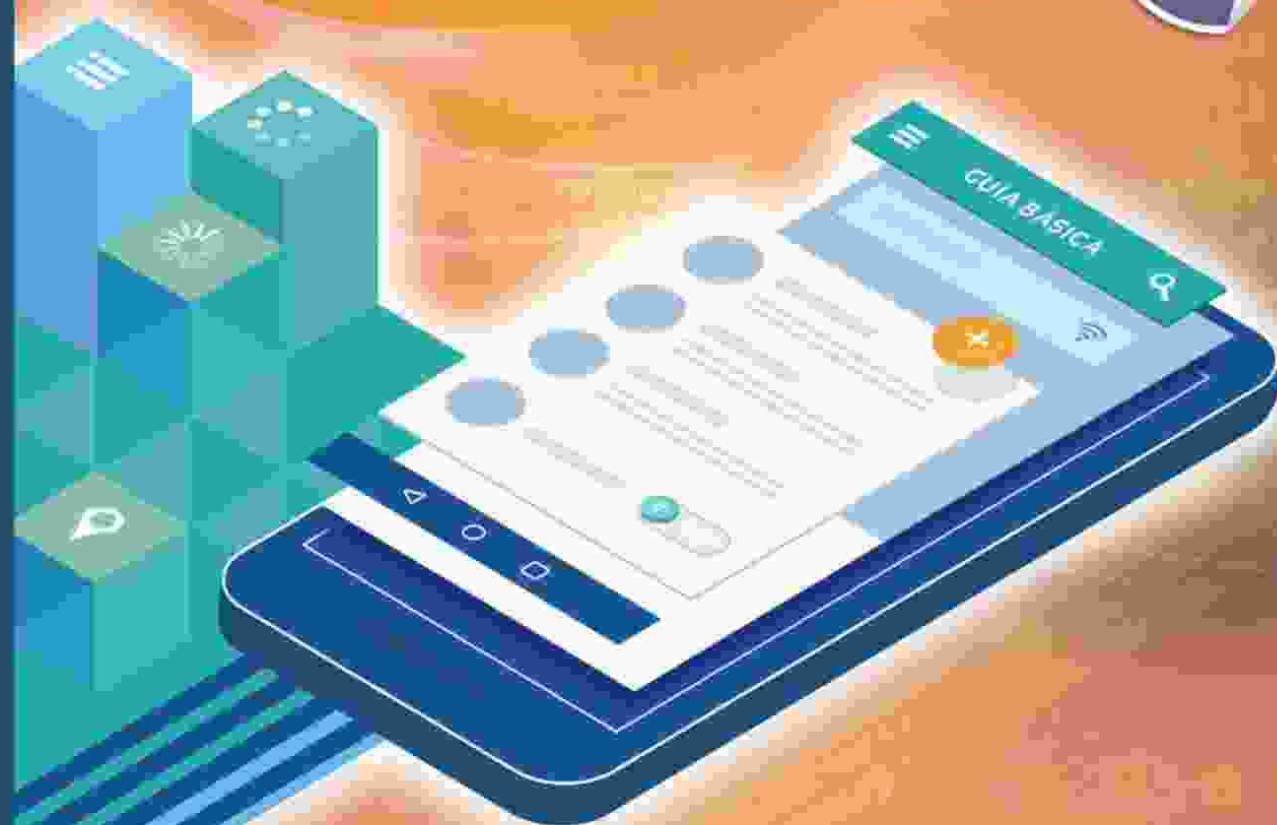


..... DISEÑO DE

INTERFACES

EN APLICACIONES

MÓVILES



SEBASTIAN
SERNA
CÉSAR
PARDO



Desde www.ra-ma.es podrá
descargar material adicional.



Ra-Ma®

..... DISEÑO DE

INTERFACES

EN APLICACIONES

MÓVILES

SEBASTIAN SERNA / CÉSAR PARDO





Diseño de interfaces en aplicaciones móviles
© Sebastian Serna y César Jesús Pardo Calvache
© De la edición: Ra-Ma 2016

MARCAS COMERCIALES. Las designaciones utilizadas por las empresas para distinguir sus productos (hardware, software, sistemas operativos, etc.) suelen ser marcas registradas. RA-MA ha intentado a lo largo de este libro distinguir las marcas comerciales de los términos descriptivos, siguiendo el estilo que utiliza el fabricante, sin intención de infringir la marca y solo en beneficio del propietario de la misma. Los datos de los ejemplos y pantallas son ficticios a no ser que se especifique lo contrario.

RA-MA es marca comercial registrada.

Se ha puesto el máximo empeño en ofrecer al lector una información completa y precisa. Sin embargo, RA-MA Editorial no asume ninguna responsabilidad derivada de su uso ni tampoco de cualquier violación de patentes ni otros derechos de terceras partes que pudieran ocurrir. Esta publicación tiene por objeto proporcionar unos conocimientos precisos y acreditados sobre el tema tratado. Su venta no supone para el editor ninguna forma de asistencia legal, administrativa o de ningún otro tipo. En caso de precisarse asesoría legal u otra forma de ayuda experta, deben buscarse los servicios de un profesional competente.

Reservados todos los derechos de publicación en cualquier idioma.

Según lo dispuesto en el Código Penal vigente, ninguna parte de este libro puede ser reproducida, grabada en sistema de almacenamiento o transmitida en forma alguna ni por cualquier procedimiento, ya sea electrónico, mecánico, reprográfico, magnético o cualquier otro sin autorización previa y por escrito de RA-MA; su contenido está protegido por la ley vigente, que establece penas de prisión y/o multas a quienes, intencionadamente, reprodujeran o plagiaran, en todo o en parte, una obra literaria, artística o científica.

Editado por:
RA-MA Editorial
Calle Jarama, 3A, Polígono Industrial Igarsa
28860 PARACUELLOS DE JARAMA, Madrid
Teléfono: 91 658 42 80
Fax: 91 662 81 39
Correo electrónico: editorial@ra-ma.com
Internet: www.ra-ma.es y www.ra-ma.com
ISBN: 978-84-9964-615-2
Depósito legal: M-1272-2016
Maquetación: Sebastian Serna
Diseño Portada: Sebastian Serna
Filmación e impresión: Copias Centro
Impreso en España en septiembre de 2016

A mis padres Adriana y Pablo, por su motivación e inmenso respeto hacia todos los proyectos que he emprendido.

Sebastian Serna

A Mara, por su comprensión y apoyo incondicional en todo momento, y a Juan por invitarme a ser parte de este maravilloso proyecto.

César Pardo

AGRADECIMIENTOS

Un agradecimiento muy especial a la editorial RA-MA por hacer realidad la publicación de esta obra y a la Facultad de Comunicaciones de la Universidad de Antioquia donde se gestó este proyecto en sus inicios como parte de una temática nueva que requiere mayor estudio en programas académicos. De igual forma a la Universidad del Cauca y sus líneas de investigación que apoyaron con su enfoque general en la segunda etapa de reescritura.

A Mauricio Naranjo por su asesoría y valiosas recomendaciones que permanecen en la esencia de estas páginas. A Estiven Restrepo, Juan David Trujillo y Luz Adriana Jaramillo por los continuos retos y aprendizajes en los proyectos móviles que emprendimos durante varios años que han quedado en la memoria como base para hacernos mejores profesionales y agudizar nuestra capacidad de innovar. Y a todos aquellos que han estado cerca en diferentes etapas influyendo directa o indirectamente en el desarrollo de lo que supone esta obra: Alejandro Carmona, Silvia Lozano, Dider Gonzalez, Hector Pico, Camilo Torres, Alexander Hernández, Jhosman Hoyos, Juan Cadavid, Beatriz Carmona, Gabriel Balzán, Daisy Paniagua, Yonathan Abad, Fernando Hurtado, Elena Correa, Vanessa Reyes, Juan Fernando Acosta, John Lopera, Alejandro Cock, Luis Fernando Ramírez, Clara Posada, Ana Victoria Ochoa, John Jaime Restrepo, Emerson Guerrero, Jorge Mario Restrepo, Victor Arias, Esteban López Uribe, Camilo Arango, Armando Ospina, Tita Ochoa, Catalina Hoyos, Gabriel Gómez, Natalia Virgen, Ana Catalina Rodríguez, Diana Ospina, Laura Caro, Belmar Santanilla, Victor Lozano, Juan Diego Jaramillo, Javier Royo, Carlos César Arbeláez, Mercedes Cardona, Ana Cristina Moya, Willy Aguirre, Javier Dávalos, Rubén Martín, Jusaí Prieto, Carlos Henao, Spiros Stathoulopoulos, Martha Ligia Parra, Camilo Virgen, Adriana Carmona, Pablo Serna, Guillermo Serna, Santiago Muñoz, Marzela Serna y Juanita, ese pequeño ser incondicional presente en todo momento con su apoyo y cariño.

LOS AUTORES

Sebastian Serna

Diseñador de Interfaces y Comunicador Audiovisual y Multimedial de la Universidad de Antioquia, Colombia. Desde el año 2007 trabaja en proyectos de diseño para medios digitales y aplicaciones móviles, en especial para el sistema operativo Android apoyando activamente comunidades que fomentan el desarrollo sobre esta plataforma de código abierto. Ha participado en proyectos de innovación y formación para entidades como Samsung Electronics Colombia, Millicom Cellular a través de su filial colombiana TigoOne, la Universidad EAFIT, Owens-Illinois, The ASLG (The Arab Sustainability Leadership Group) de Jordania y eventos de tecnología como Campus Party Colombia. En la actualidad alterna su trabajo entre proyectos comerciales para empresas de desarrollo de software e iniciativas de código abierto en su propio estudio, Zemiotic Interfaces, especializado en diseño de interfaces gráficas para aplicaciones, iconos y branding de software. Para conocer más sobre sus proyectos pueden visitar <http://www.zemiotic.com>

César Jesús Pardo Calvache

Doctor y magister en Ingeniería Informática por la Universidad de Castilla-La Mancha, España. Ingeniero de Sistemas por la Universidad del Cauca, Colombia, donde actualmente labora como profesor asistente de tiempo completo adscrito a la Facultad de Ingeniería Electrónica y Telecomunicaciones y al Departamento de Sistemas orientando materias relacionadas a la teoría de la Computación, Calidad de Software, Programación, Ingeniería de Software y Metodologías ágiles. El profesor Pardo es Scrum Master por la SCRUM ALLIANCE en Madrid, España. Asimismo, se ha desempeñado como consultor e investigador con más de 11 años de experiencia en temas relacionados a: la calidad de procesos, calidad de producto, mejora de procesos, armonización de múltiples modelos y metodologías ágiles para la gestión de equipos de trabajo y el desarrollo de productos y servicios. Autor de más de 50 publicaciones entre artículos, capítulos de libros y ponencias a nivel nacional e internacional, lo que le permitió en el 2015 y 2016 estar en el listado de los 500 colombianos más citados en Colombia a través de una consulta realizada por webometrix.

CONTENIDO

INTRODUCCIÓN	XV
CAPÍTULO 1: ECOSISTEMAS MÓVILES.....	1
Sistemas operativos móviles.....	3
Android.....	4
iOS.....	6
Windows Phone.....	8
BlackBerry OS.....	10
Sistemas emergentes	12
Tizen	12
Sailfish OS.....	13
Ubuntu.....	14
WatchOS.....	15
Participación de mercado entre sistemas	16
Dispositivos móviles	18
Aplicaciones móviles.....	20
Mercados de aplicaciones.....	24
Tipos de aplicaciones.....	26
Disciplinas relevantes en el diseño de aplicaciones	28
Experiencia de Usuario (UX)	30

Interacción Persona-Ordenador (IPO)	31
Diseño Visual	32
Diseño de Interacción (IxD).....	32
Arquitectura de Información (IA)	33
Programación.....	33
Diseño Industrial	33
Importancia del diseño de interfaces.....	34
Lenguajes de diseño emergentes	35
Modern UI (Microsoft).....	36
Material Design (Google)	38
Post-Aqua UI (Apple).....	40
CAPÍTULO 2: ANÁLISIS Y ARQUITECTURA.....	45
Planeación y definición	47
Identificando a los usuarios.....	50
Entrevistas contextuales.....	50
Encuestas	50
Grupos focales	51
Personas	51
Escenarios	53
Definiendo la Arquitectura de Información	54
Card sorting.....	54
Creación de diagramas.....	56
Tipos comunes de navegación	58
Navegación lineal	58
Navegación centralizada	60
Navegación interconectada	62
Navegación plana	64
Profundidad de navegación	66
Modelos conceptuales y mentales.....	67

CAPÍTULO 3: DISEÑO Y PROTOTIPADO	73
De la estructura a lo tangible	75
Gestos de interacción comunes	76
Creación de prototipos.....	80
Prototipos de baja fidelidad.....	80
Prototipos de alta fidelidad.....	82
Elementos básicos de la interfaz.....	84
Pantallas	85
Contenedores.....	87
Elementos multimedia.....	88
Imágenes.....	89
Iconos	90
Textos.....	92
Audio	94
Video	95
Elementos interactivos	96
Botones	97
Botones de radio	98
Cajas de chequeo.....	99
Listas desplegables y menús	100
Campos de inserción de texto	101
Interruptores.....	102
Deslizadores	103
Barras de desplazamiento.....	104
Indicadores de progreso	105
Transiciones y efectos de animación.....	106
Patrones de diseño comunes	108
Selección de menús.....	110
Barras de opciones	110

Paneles de pestañas	112
Listados de opciones	114
Matrices de opciones	116
Menús laterales	118
Tarjetas de contenido	120
Cuadros de diálogo	121
Formularios.....	122
Campos de búsqueda	122
Campos de mensajes.....	124
Pantallas de acceso.....	125
Manipulación directa.....	126
Lienzo de dibujo	126
Metáforas	127
Visores de mapas.....	128
Otros patrones	130
Pantallas de contenido.....	130
Paneles vacíos	132
Pantallas de bienvenida	133
CAPÍTULO 4: PRUEBAS E IMPLEMENTACIÓN	137
Importancia de las pruebas	139
El concepto de usabilidad	140
Pruebas con usuarios	141
Pruebas de laboratorio	142
Pruebas de guerrilla.....	143
Pruebas remotas	144
Realización de las pruebas.....	145
Planeación	145
Preparación.....	145
Realización	146

Elaboración de reportes.....	147
Revisiones de expertos.....	148
Evaluación heurística.....	148
Del prototipo a la implementación	150
iOS.....	152
Android.....	156
Windows Phone	160
Equivalencias entre componentes de interfaz	164
CAPÍTULO 5: IDENTIDAD Y PROMOCIÓN.....	169
Creando una identidad	171
Imagen de marca en el software.....	172
Iconos de aplicación.....	173
Uso del color	176
Paletas monocromáticas	178
Paletas complementarias	179
Paletas análogas	180
Paletas triádicas	181
Tipografía.....	182
Guías de estilo	186
Canales de promoción	188
Mercados de aplicaciones	188
Sitios web.....	190
Redes sociales	192
Medios tradicionales.....	194
Televisión y video.....	194
Publicidad exterior	196
Publicidad Impresa.....	198
CRÉDITOS DE IMÁGENES	203
ÍNDICE ALFABÉTICO	207

INTRODUCCIÓN

La computación móvil ha tenido un crecimiento acelerado en los últimos años y la proliferación de dispositivos inalámbricos, en especial de teléfonos inteligentes, está transformando la manera como interactuamos y realizamos las tareas de la vida cotidiana, un indicador de peso que presenta al dispositivo móvil como un actor de cambio que está permeando todas las esferas de la sociedad. De igual forma, la creciente popularidad de sistemas operativos móviles como Android y iOS ha provocado la explosión de tiendas de aplicaciones con un crecimiento exponencial, tanto que el software es ahora, más que nunca, el encargado de promover las bondades de los dispositivos móviles. La reciente inclusión de esta tecnología como medio masivo de comunicación obliga a todas las áreas relacionadas a reinventarse y adaptarse a los desafíos del desarrollo de software para este nuevo contexto.

Este libro ofrece una guía introductoria sobre el tema, con un enfoque didáctico para a un público diverso como directores de proyecto, estudiantes, diseñadores, ingenieros y desarrolladores que deseen complementar sus conocimientos básicos sobre el proceso de diseño de interfaces en aplicaciones móviles. Los temas se abordan de manera breve presentando sus componentes esenciales sin profundizar en aspectos técnicos como tutoriales paso a paso sobre la implementación de una interfaz por medio de código de programación, labor que ya cubren diversos libros especializados para cada plataforma. Por su parte, esta obra presenta contenidos generales que ofrecen una visión panorámica de los temas abordados, y para aquellos que deseen ampliar sus conocimientos, al final de cada capítulo se incluye el listado de referencias y recomendaciones de lecturas que permiten profundizar en las temáticas abordadas hacia trabajos de diferentes autores y recursos. Los contenidos del libro se encuentran divididos en 5 capítulos:

CAPÍTULO 1: ECOSISTEMAS MÓVILES

En este capítulo se abordan conceptos generales de actualidad en el contexto móvil, referencias básicas sobre los sistemas operativos más importantes del mercado y las apuestas futuras de sistemas emergentes; así como los diferentes mercados y tipos de aplicaciones. También se mencionan las disciplinas relevantes en el proceso de desarrollo y la adopción de los lenguajes de diseño como estrategias para mejorar y fidelizar el posicionamiento de marca.

CAPÍTULO 2: ANÁLISIS Y ARQUITECTURA

En este apartado se estudian temas de análisis e investigación de usuarios como las entrevistas, encuestas, grupos focales, personas y escenarios como apoyo inicial en ese reconocimiento del público objetivo de un proyecto móvil y se aborda la definición de la Arquitectura de Información presentando algunos métodos básicos como el card sorting, la creación de diagramas y los tipos de navegación comunes en aplicaciones.

CAPÍTULO 3: DISEÑO Y PROTOTIPADO

En el tercer capítulo se estudian los elementos que componen el proceso de diseño de la interfaz gráfica pasando por el reconocimiento de los gestos táctiles comunes y los métodos de prototipado de alta y baja fidelidad. También se describen los componentes multimedia y elementos interactivos que podemos utilizar en la construcción de una interfaz gráfica con el uso de patrones de diseño comunes en aplicaciones móviles como los paneles de pestañas, listados y matrices de opciones, visores de mapas, menús laterales, entre otros.

CAPÍTULO 4: PRUEBAS E IMPLEMENTACIÓN

En este apartado se describe el concepto de usabilidad y las pruebas más conocidas, éstas se pueden realizar en laboratorios especializados o en diferentes lugares de una manera informal, incluso realizar pruebas de forma remota es posible gracias a servicios en línea que ofrecen estas características. En este apartado también se mencionan las tecnologías y lenguajes utilizados para integrar un diseño de interfaz en los sistemas operativos móviles así como los entornos oficiales, pantallas e iconos de las plataformas más conocidas.

CAPÍTULO 5: IDENTIDAD Y PROMOCIÓN

Para finalizar, en este capítulo se abordan algunos temas importantes sobre la identidad de marca en aplicaciones móviles, así como su promoción en diferentes canales. Se hace un recorrido por los iconos de aplicación, el uso del color a través del círculo cromático, tipografías, guías de estilo y canales publicitarios comunes para realizar campañas en medios digitales. También se destina un espacio para hablar de la publicidad tradicional como estrategia de posicionamiento y promoción indirecta de aplicaciones móviles.



RECURSOS ADICIONALES

El material adicional de este libro puede descargarlo en nuestro portal web: <http://www.ra-ma.es>. Debe dirigirse a la ficha correspondiente a esta obra, dentro de la ficha encontrará el enlace para poder realizar la descarga. Dicha descarga consiste en un fichero ZIP con una contraseña de este tipo: XXX-XX-XXXX-XXX-X la cual se corresponde con el ISBN de este libro. Podrá localizar el número de ISBN en la página IV (página de créditos). Para su correcta descompresión deberá introducir los dígitos y los guiones. Cuando descomprima el fichero obtendrá los archivos que complementan al libro para que pueda continuar con su aprendizaje. Desde la web de los autores: <http://interfacesmoviles.com/> podrá acceder a plantillas de interfaz y gestos táctiles para utilizar en proyectos generales así como recomendaciones de herramientas útiles de consulta, diseño y prototipos.

INFORMACIÓN ADICIONAL Y GARANTÍA

- RA-MA EDITORIAL garantiza que estos contenidos han sido sometidos a un riguroso control de calidad.
- Los archivos están libres de virus, para comprobarlo se han utilizado las últimas versiones de los antivirus líderes en el mercado.
- RA-MA EDITORIAL no se hace responsable de cualquier pérdida, daño o costes provocados por el uso incorrecto del contenido descargable.
- Este material es gratuito y se distribuye como contenido complementario al libro que ha adquirido, por lo que queda terminantemente prohibida su venta o distribución.

1

ECOSISTEMAS MÓVILES

En este capítulo se abordan conceptos generales de actualidad en el contexto móvil, referencias básicas sobre los sistemas operativos más importantes del mercado y las apuestas futuras de sistemas emergentes; así como los diferentes mercados y tipos de aplicaciones. También se mencionan las disciplinas relevantes en el proceso de desarrollo y la adopción de los lenguajes de diseño como estrategias para mejorar y fidelizar el posicionamiento de marca en la industria del software.

Sistemas operativos móviles

En la actualidad, los ordenadores personales son equipos indispensables y complementarios en labores cotidianas e industriales y desde su aparición, a finales de los años setenta, estos equipos informáticos han incorporado sistemas operativos para controlar su hardware. Sin embargo, en los últimos años, hemos visto un notable aumento en la proliferación de dispositivos móviles con capacidades muy similares a las de los ordenadores de escritorio y, además, con software especializado para su contexto.

Un sistema operativo móvil es similar a los ya conocidos Mac OS, Windows o Linux, pero con funciones más simplificadas. Estos, vienen instalados en dispositivos como tabletas, teléfonos inteligentes, cámaras fotográficas, relojes, entre otros, y permiten por lo general una interacción enriquecida a través de pantallas táctiles. Sistemas como Android y iOS materializan lo que se intentó lograr con Symbian en los años noventa y comienzos del 2000 con su presencia destacada en la industria y una ventana para desarrolladores, usuarios, fabricantes y operadores de telefonía.

Su adopción masiva ha transformado la dinámica a partir de la última década, permitiendo la creación de ecosistemas de servicios y aplicaciones tanto que ahora las interacciones con los dispositivos móviles han dejado de ser simples como se realizaban en los teléfonos convencionales y han abierto posibilidades infinitas para el consumo de contenido. Este consumo ha sido promovido por la creación de tiendas o mercados de aplicaciones donde se encuentran e instalan miles de paquetes de software disponibles a través de una comunidad de desarrolladores creciente.

Si tenemos en cuenta su impacto actual, estamos asistiendo a un cambio de paradigma donde la ubicuidad de los equipos inalámbricos y los sistemas operativos móviles han comenzado a volverse una necesidad para los usuarios quienes utilizan cada vez más estas plataformas que ofrecen ecosistemas muy atractivos de aplicaciones y servicios de computación en la nube.

A continuación se presentan los sistemas operativos más representativos del panorama actual (Android, iOS, Windows Phone, BlackBerry OS) y los sistemas operativos emergentes que se perfilan como interesantes apuestas futuras (Tizen, Sailfish OS, Ubuntu, WatchOS).

Android

Es un sistema operativo de código abierto y ecosistema de servicios. Su desarrollo inició en el año 2003 en la empresa Android Inc. fundada por Andy Rubin y adquirida por Google en 2005 [1]. Luego de esta adquisición, fue presentado en el año 2008 como apuesta principal de la Open Handset Alliance, agrupación de más de 80 empresas lideradas por Google, entre las que se encuentran fabricantes de dispositivos, operadoras de telefonía y empresas de desarrollo de software.

Esta plataforma, creada a partir del núcleo de Linux, fue la encargada de masificar el concepto de sistema operativo móvil que, como sucedió años atrás con Symbian OS, fue concebida para instalarse en multitud de terminales de marcas diversas. El primer teléfono Android comercializado fue el G1 de la marca taiwanesa HTC, pero en la actualidad es utilizado por los principales fabricantes de dispositivos móviles como Samsung, Xiaomi, Lenovo, Motorola, Sony, LG, Huawei, ZTE, entre otras. Su cuota de participación de mercado se debe a esta diversidad de marcas, lo que también ha generado multitud de versiones personalizadas del sistema. Android ha tenido una evolución marcada en este proceso de posicionamiento y se ha transformado cada año lanzando versiones compatibles con tabletas, televisores y su versión para dispositivos corporales llamada Android Wear presentada en 2014 como sistema operativo para relojes inteligentes.

Las aplicaciones Android se desarrollan en el lenguaje de programación Java y utilizan XML para la presentación de su interfaz gráfica. Ofrece un SDK que se adapta a las necesidades comunes de desarrollo de aplicaciones que se integra con entornos de desarrollo como Netbeans, Eclipse, IntelliJ y en los últimos años, su propio software llamado Android Studio enfocado al desarrollo de aplicaciones. También ofrece, con un propósito más específico, un NDK que permite a desarrolladores acceder a opciones de bajo nivel para programar el hardware, esto ha permitido usos específicos de Android en proyectos de robótica y el desarrollo de aplicaciones que requieren un consumo alto de recursos al permitir la programación en lenguajes como C y C++.

Para la distribución de aplicaciones Android tiene una tienda oficial llamada Google Play Store, pero también se distribuyen en tiendas alternativas de otras empresas como la Amazon App Store y Samsung Galaxy Apps.



Figura 1-1 Nexus 5X de LG con interfaz gráfica nativa de Android. Los dispositivos Android tienen diversas interfaces dependiendo del fabricante. La interfaz presentada en la imagen es la oficial que promueve Google, sin embargo, Samsung, Sony y otros fabricantes de dispositivos utilizan su propia modificación de la interfaz nativa.

iOS

Es el sistema operativo que se instala en el iPod Touch, iPhone y el iPad. iOS ha sido desarrollado a partir del sistema operativo Mac OS disponible en los diferentes ordenadores de escritorio y portátiles de Apple. Fue presentado por primera vez en 2007 con el lanzamiento del iPhone, sin embargo, en los años siguientes, Apple lo promocionó solo como iPhone OS y finalmente en 2010 presentó iOS de forma oficial como su sistema operativo móvil, unificando su estrategia para todos los equipos de la compañía.

A esta plataforma se le atribuye el nacimiento del ecosistema de móviles inteligentes modernos con pantallas táctiles y la ejecución de un modelo de negocio exitoso basado en los mercados de aplicaciones. Su tienda App Store lanzada en 2008, ha sido el punto de referencia para las demás empresas de tecnología que impulsan plataformas móviles. iOS posee un ecosistema muy regulado por Apple y utiliza un proceso de aprobación de aplicaciones donde se tienen lineamientos muy estrictos, pese a esto tiene una participación de mercado estable con una comunidad de desarrolladores creciente.

Además de iOS Apple también ha creado un sistema operativo específico llamado WatchOS (que se describe más adelante como un sistema emergente) para el desarrollo de aplicaciones que se instalan en el Apple Watch ofreciendo algunas variaciones con respecto al desarrollo tradicional para el iPhone y el iPad. Las aplicaciones iOS se desarrollan en los lenguajes de programación Objective C y Swift, este último presentado por Apple en el 2014 para promover un desarrollo más ágil dada su flexibilidad con respecto a Objective C. El desarrollo de aplicaciones se centraliza en el entorno Xcode donde se encuentran las herramientas disponibles del SDK de iOS. Un punto a tener en cuenta es que Xcode solo está disponible en Mac OS por lo que es necesario tener un Mac, o utilizar una máquina virtual para desarrollar aplicaciones nativas.

Para el diseño de la interfaz gráfica, iOS se apoya en un editor visual llamado Interface Builder que permite arrastrar y soltar componentes sin escribir estructuras de código de marcado, además soporta un formato llamado Storyboard que permite relacionar en un solo archivo las pantallas de la aplicación, su navegación y estructura. Antes de incluir esta característica, el Interface Builder solo utilizaba archivos XIB. Ambos formatos (XIB y Storyboard) se basan en código XML y se complementan en el diseño de interfaces para iOS.



Figura 1-2 iPhone 6s y iPad Pro en color blanco con sistema operativo iOS 9. A partir de la versión 7 del sistema operativo, Apple ha cambiado la estética general de la interfaz de usuario siguiendo la tendencia impulsada por Microsoft de utilizar colores planos y un estilo minimalista.

Windows Phone

Es el sistema operativo móvil oficial de Microsoft lanzado a finales de 2010. No debe confundirse con Windows Mobile, el sistema con el que la empresa estadounidense comenzó desde el 2000 su carrera por la movilidad. Windows Phone se creó como un sistema operativo nuevo, con una arquitectura más avanzada y enfocado en pantallas táctiles de terminales de última generación de marcas como Dell, HTC, Samsung y Nokia.

Sin duda, el movimiento más importante de Microsoft con esta nueva plataforma fue su alianza con Nokia para adaptarlo como sistema operativo oficial en sus teléfonos inteligentes, para esto la empresa finlandesa creó una línea especial de dispositivos conocida como Nokia Lumia, con una popularidad moderada que llevó a que Microsoft adquiriera en 2013 la división de equipos a Nokia absorbiendo la línea Lumia como parte de su marca, al renombrar sus equipos como Microsoft Lumia [2].

Con la adquisición de Nokia, Microsoft está cambiando de forma radical su enfoque de solo producir software, para también encargarse de su propio hardware, como lo hace Apple. En 2015, siguiendo el proceso comenzado en Windows 8, se integró con toda la *suite* del sistema Windows 10 para permitir una compatibilidad entre las aplicaciones de escritorio y móviles de Microsoft.

El cambio más notorio de Windows Phone ha sido la interfaz de usuario del sistema, el más radical introducido por Microsoft en toda su historia, ofreciendo una experiencia muy distinta a los demás sistemas operativos. Este diseño fue presentado en sus inicios como Metro Style y luego renombrado como Modern UI como semilla del rediseño de todos los sistemas de Microsoft tanto en móviles como de escritorio, unificando la interfaz de usuario en un nuevo esquema digital de mosaicos llamados *tiles*.

Las aplicaciones para este sistema se desarrollan en el IDE Visual Studio utilizando los lenguajes soportados por Microsoft como VB, C# y C++. En cuanto al manejo de la interfaz gráfica, las aplicaciones se construyen con un lenguaje llamado XAML una variante de XML creada por Microsoft para el diseño de interfaces en aplicaciones. La distribución se realiza a través de la Windows Store con un número creciente de aplicaciones disponibles, similar a las tiendas de la competencia (Google Play y App Store).



Figura 1-3 Teléfono Microsoft Lumia 640 XL con sistema operativo Windows Phone. A raíz de la compra de la división móvil de Nokia la marca Lumia ahora es propiedad de Microsoft que la utiliza como su marca oficial de dispositivos móviles.

BlackBerry OS

Es un sistema operativo móvil de la compañía canadiense BlackBerry (hasta 2012 llamada Research In Motion) que viene instalado en dispositivos BlackBerry como teléfonos móviles y tabletas. Sus primeras versiones se presentaron a finales de los años noventa instaladas en diversos equipos que se han reestructurado y mejorado durante la última década. Entre todos los sistemas operativos móviles actuales, es el más antiguo.

BlackBerry tuvo un éxito notable entre 2002 y 2009 donde se especializó en un mercado empresarial, ejecutivos y hasta funcionarios de gobierno utilizaron de forma masiva este sistema que fue de los primeros en ofrecer prestaciones avanzadas de correo electrónico y mensajería. BlackBerry Messenger fue la aplicación pionera que popularizó los chats a través del teléfono móvil, categoría dominada en la actualidad por servicios como WhatsApp y Line.

En el año 2010 BlackBerry adquirió la empresa QNX y su sistema operativo como estrategia para integrarlo a su nueva apuesta por las tabletas. Es así como las posteriores versiones de su plataforma se han realizado bajo esta fusión [3]. Las aplicaciones para BlackBerry OS se desarrollan por medio de su SDK en el lenguaje de programación Java, pero en los últimos años con las mejoras de BlackBerry 10 ha incluido soporte para desarrollo en lenguajes como C, C++, Qt, Javascript, CSS y HTML e incluso Java para Android.

En la actualidad, la empresa atraviesa por un momento de declive y se encuentra evaluando opciones para recuperar su posicionamiento. En esta búsqueda por recuperarse ha lanzado al mercado el BlackBerry Priv, primer terminal BlackBerry en incluir Android como sistema operativo, con una personalización en temas de seguridad e incluyendo un teclado físico discreto detrás de la pantalla táctil. El futuro de BlackBerry OS aún está en una posición desequilibrada con respecto a las demás plataformas móviles como Android y iOS.

En cuanto a la distribución masiva de software, BlackBerry tiene su propia tienda de aplicaciones llamada BlackBerry World donde se encuentran disponibles las aplicaciones compatibles con terminales de la marca. Similar a Apple, BlackBerry OS solo se instala en los dispositivos BlackBerry centrándose en un control de hardware por parte de la empresa Canadiense.



Figura 1-4 Vista frontal del dispositivo BlackBerry Q10 que conserva el diseño clásico con teclado físico QWERTY de los productos de la empresa canadiense. Sin embargo, BlackBerry también soporta interacciones con pantallas táctiles sin teclado físico en otros terminales.

Sistemas emergentes

Tizen

Es un sistema operativo que mezcla aspectos de código abierto y código propietario. Su desarrollo ha sido adelantado por The Linux Foundation con el apoyo de Samsung e Intel. Su primera versión fue lanzada a comienzos de 2012 y al igual que Android se basa en el núcleo de Linux. Tizen está desarrollado para instalarse en diferentes categorías de dispositivos como teléfonos inteligentes, televisores, relojes y tabletas.

En sus inicios, Tizen perseguía el objetivo de construir un sistema operativo de código abierto sin las restricciones de una empresa privada, pero su estrategia se ha transformado con el aporte de Samsung en 2013 que fusionó su sistema operativo, conocido como Bada con el actual desarrollo de Tizen, desde entonces el mayor movimiento de la empresa coreana ha sido integrarlo en sus relojes, su familia de televisores y teléfonos inteligentes donde el primer terminal comercializado con Tizen fue el Samsung Z [4].

Tizen utiliza aplicaciones móviles basadas en HTML5, CSS y Javascript. No obstante, también soporta aplicaciones Android con la ayuda de software de terceros.



Figura 1-5 Tizen instalado en teléfono móvil Samsung Z, dispositivo comercializado en Rusia y luego en India. Samsung es una de las empresas que impulsa Tizen como sistema operativo móvil luego de fusionarlo con su sistema Bada. Esta nueva plataforma se instala de forma oficial en relojes, televisores inteligentes y algunos teléfonos de la compañía coreana.

Sailfish OS

Se podría decir que Sailfish es la “resurrección” de un sistema operativo anterior llamado MeeGo, promovido desde 2010 por The Linux Foundation, Intel y Nokia para su gama de teléfonos. Esta plataforma fue una evolución del sistema operativo Maemo. Sin embargo, fue descartado en 2011 por Nokia que le apostó de lleno a su alianza con Microsoft y Windows Phone para incluirlo en sus teléfonos inteligentes.

Como respuesta, otra empresa finlandesa llamada Jolla, creada por antiguos trabajadores de Nokia [5], decidió llevar las riendas del proyecto retomado por la comunidad de desarrolladores de MeeGo por medio de una bifurcación llamada Mer. Sailfish se basa en Mer y al igual que Android está desarrollado sobre el núcleo de Linux. Las aplicaciones para Sailfish OS se desarrollan en el entorno Qt con C++ soportando QML para su interfaz gráfica. También permite el desarrollo en el lenguaje Python y aplicaciones Android con sus librerías de compatibilidad.

Jolla fabrica sus propios terminales con Sailfish OS en teléfonos inteligentes y aunque también desarrolló una tableta con recursos del sitio Indiegogo, este proyecto (Jolla Tablet) fue discontinuado en 2016.



Figura 1-6 Sailfish OS en teléfono Jolla. El proyecto está teniendo expectativa en un nicho de mercado como la continuación de lo que pudo ser el sistema oficial de Nokia. Aunque se embarcó en la fabricación de una tableta (Jolla Tablet) con apoyo de *crowdfunding* tuvo problemas para sacar adelante el proyecto y retomó el enfoque original hacia los teléfonos inteligentes.

Ubuntu

Es una de las distribuciones más conocidas del sistema operativo Linux, y desde 2004 ha estado haciéndose un lugar en la industria de los ordenadores de escritorio. En la actualidad espera competir en el mercado móvil con la inclusión de Ubuntu Phone, una versión optimizada del sistema con una interfaz de usuario basada en gestos táctiles. Las primeras pruebas del sistema se presentaron durante el CES 2013 (Consumer Electronics Show) realizado en Las Vegas, Estados Unidos.

Ubuntu es un sistema de código abierto desarrollado por la empresa británica Canonical que se encarga de promoverlo. Aunque esta versión para móviles tuvo un intento frustrado de financiación a través de la plataforma de *crowdfunding* Indegogo [6], esto no detuvo el desarrollo del proyecto y en la actualidad la compañía tiene acuerdos con BQ y Meizu para instalar el sistema sobre estas marcas de dispositivos móviles.

Las aplicaciones para Ubuntu se pueden desarrollar en su propio SDK en el entorno Qt con los lenguajes C/C++ y QML pero también tiene soporte para HTML5, CSS y Javascript en la creación de aplicaciones web que pueden ejecutarse en este sistema operativo.



Figura 1-7 Ubuntu Phone en un dispositivo móvil BQ Aquaris E5. Esta marca española de teléfonos ha lanzado algunas versiones especiales con Ubuntu instaladas, los dispositivos BQ Aquaris también se comercializan con el sistema operativo Android. Otra marca que tiene alianzas con Canonical es el fabricante chino Meizu que instala Ubuntu en algunos de sus teléfonos inteligentes.

WatchOS

WatchOS es un nuevo sistema operativo para relojes inteligentes desarrollado por Apple y derivado de iOS. Disponible en el año 2015 con la introducción del Apple Watch, producto emergente de la compañía estadounidense. En sus inicios se pensaba que el sistema operativo utilizado por este dispositivo corporal era iOS pero luego Apple presentó WatchOS como un sistema operativo específico para relojes.

Aunque Google ya había comenzado a innovar en este sector desde el año 2012, no fue hasta 2014 que la tendencia en relojes inteligentes comenzó a crecer con la presentación de Android Wear, tanto que fabricantes como Samsung, Motorola, LG, Sony y ahora Apple le están apostando fuerte a este nuevo medio de pantallas reducidas que permite una interacción más personal.

Las aplicaciones para WatchOS se desarrollan, al igual que iOS, en el entorno Xcode disponible en los ordenadores con Mac OS. Los lenguajes soportados son Swift y Objective C. Además posee una serie de librerías que ayudan al desarrollo de aplicaciones para este sistema operativo. Aunque su lanzamiento es reciente, está generando expectativa en la comunidad de desarrolladores.



Figura 1-8 Reloj inteligente Apple Watch con sistema operativo WatchOS 2. Este nuevo dispositivo es una apuesta sólida de Apple por imponerse en el sector de rápido crecimiento de los dispositivos corporales.

Participación de mercado entre sistemas

Como pudimos observar, existen diversos sistemas operativos que actúan dentro de su propia estrategia y tecnologías, pero un factor importante para dimensionar el impacto que las plataformas tienen a nivel global es la evolución del posicionamiento que cada una tiene en el mercado. La Figura 1-9 presenta estadísticas generales sobre la cuota de participación de mercado donde Android posee una predominancia marcada con respecto a los demás sistemas [7].

Es notable el posicionamiento que Android ha conseguido desde su lanzamiento en el año 2008 con una participación de mercado dominante gracias a su estrategia de código abierto que resultó muy atractiva para la industria. Y aunque este sistema operativo es propiedad de Google, en la práctica lo utilizan bastantes empresas como plataforma principal, no se reduce a una sola compañía. A diferencia de Microsoft, Apple y BlackBerry que promueven sus propias marcas de dispositivos, Android representa la adopción masiva de gran parte de los fabricantes de terminales móviles y esta supremacía también ha traído problemas en la plataforma como la falta de estandarización entre terminales, rendimiento intermitente y la conocida fragmentación de versiones del sistema, uno de los grandes inconvenientes para Google. Esta fragmentación es notable en el código fuente pero también en su interfaz gráfica ya que existen versiones personalizadas de interfaz de muchos fabricantes quienes se resisten a adoptar la versión nativa del sistema, algunas capas de interfaz conocidas son Touchwiz de Samsung, Sense de HTC y MIUI de Xiaomi.

Por su parte, iOS ha tenido una estabilidad notable a lo largo de los años, conservando su mercado de nicho que supone una base de usuarios fidelizados que están a la expectativa de los nuevos lanzamientos de la empresa asegurando su posicionamiento. La situación de Apple es privilegiada ya que representa una sola marca que posee una participación de mercado considerable.

Windows Phone ha tenido desde su presentación en 2010 una cuota muy limitada, es un fenómeno particular pues una compañía como Microsoft que tiene todos los recursos económicos para impulsar sus proyectos ha tenido grandes dificultades para generar empatía con el mercado de usuarios. Aunque este sistema ha tenido críticas positivas por parte de usuarios y expertos, esto no ha inclinado la balanza y su participación actual es poco significativa con respecto a los demás ecosistemas.

De igual forma BlackBerry OS ha tenido un declive desde el año 2009 y su participación en el mercado ahora es casi inexistente. Algo que ha llevado a la empresa a replantear su estrategia realizando pruebas con el sistema operativo Android en sus propios terminales.

Los sistemas emergentes (Tizen, Sailfish OS, Ubuntu) todavía no aparecen en el panorama general y su participación aún estará por verse en una industria dominada por Android y iOS. Sin embargo, conviene recordar que en un entorno tan volátil como la industria tecnológica ocurren sorpresas, un ejemplo de ello es el caso de Nokia y RIM (ahora BlackBerry) empresas que tenían un posicionamiento dominante en el año 2008 que parecía no tener competencia hasta que iOS y Android transformaron el mercado de forma disruptiva.

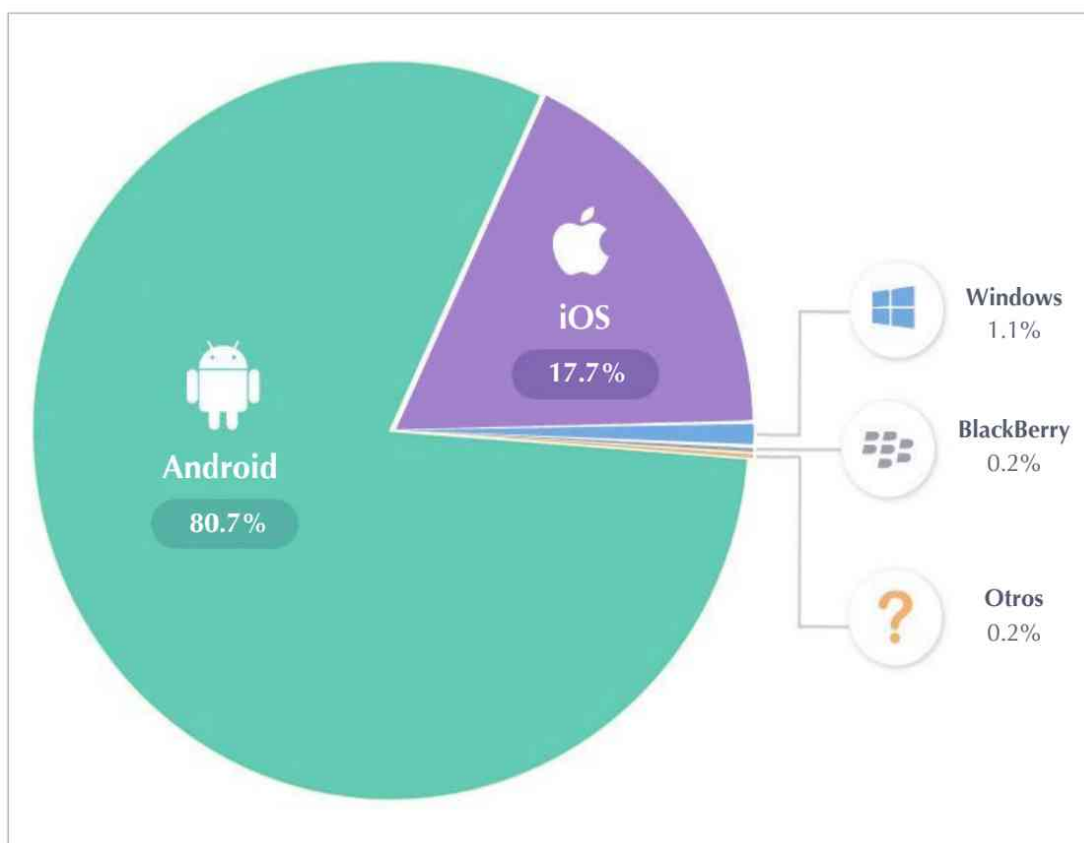


Figura 1-9 Participación de mercado a nivel mundial de sistemas operativos móviles. Estadísticas de la consultora Gartner del cuarto trimestre de 2015 (Q4 2015) donde se tiene en cuenta las ventas de dispositivos de las diferentes marcas y su relación con la cuota de participación de cada sistema operativo en cada trimestre.

Dispositivos móviles

Los dispositivos necesarios para correr un sistema operativo móvil se pueden agrupar en teléfonos inteligentes, tabletas, relojes, cámaras digitales y cualquier tipo de dispositivo corporal. Pero se utilizan con mayor frecuencia los primeros dos tipos. La interacción con el sistema se realiza por medio de pantallas táctiles o teclados físicos incorporados en el equipo que permiten una interacción cercana con el sistema y las aplicaciones instaladas.

Un dispositivo móvil inteligente posee los componentes necesarios en una sola pieza: teclado, pantalla, cámara, procesador, memoria RAM y memoria interna están unidos en un objeto que conforma la interacción global. Además, las tecnologías inalámbricas actuales permiten que pueda ser llevado a cualquier lugar, sin necesidad de cables. Su reducido tamaño ayuda a que se pueda guardar en bolsos, carteras y bolsillos sin ninguna dificultad de tipo ergonómico.

La anatomía básica de un dispositivo móvil moderno comprende un diseño rectangular de mayor o menor tamaño (sea teléfono o tableta) donde una gran superficie corresponde a la pantalla con la que se interactúa. Algunos equipos incluyen un teclado físico (BlackBerry y ciertos móviles Android) sin embargo, existe una clara tendencia hacia la utilización de pantallas táctiles con teclados virtuales que aparecen sólo cuando el usuario necesita escribir. También existen dispositivos como relojes inteligentes que se presentan como nuevos medios emergentes, aunque su adopción es aún limitada. Este modelo estandarizado con pantallas táctiles que cubren gran parte de la superficie del móvil también soporta accesorios: es posible usar dispositivos periféricos como audífonos, teclados, lápices digitales (*stylus*), ratones, parlantes, bases para ubicarlos en el automóvil que se utilizan como una extensión de sus prestaciones, si embargo, no son indispensables para interactuar de forma eficiente con el dispositivo móvil.

Aunque este diseño minimalista es el nuevo estándar de la industria, todavía tiene retos importantes en accesibilidad para brindar soporte a poblaciones en situación de discapacidad visual para quienes la pantalla táctil no es un modelo ideal que soporte su interacción básica, donde el teclado físico ofrece una experiencia más adecuada. No obstante, también ofrece nuevos avances que prometen mitigar estas deficiencias a través de métodos como el ingreso de datos por voz e interacciones con lenguaje natural con asistentes personalizados.



Figura 1-10 Los dispositivos móviles más utilizados son los teléfonos, tabletas y en creciente auge los relojes inteligentes. Dependiendo del tipo de dispositivo la interacción puede tener modificaciones sutiles como ocurre con los teclados físicos muy utilizados en terminales BlackBerry. Por otro lado la interacción con tabletas también se complementa con el uso de lápices digitales (*stylus*) como en el Galaxy Note.

Aplicaciones móviles

Una aplicación móvil es un pequeño paquete de software que sirve para resolver una o varias tareas en específico. Son similares a los conocidos procesadores de texto, las hojas de cálculo, los programas de diseño y edición de video de los ordenadores de escritorio pero con una complejidad menor y optimizadas para el contexto móvil. El concepto de aplicación se ha utilizado desde los años 70 en la industria informática y el origen del término se le atribuye a Jef Raskin mientras trabajaba para Apple [8].

En la actualidad, más de tres décadas después, el ecosistema de aplicaciones ha servido para extender las capacidades de los ordenadores personales y ha migrado a los teléfonos inteligentes gracias a las tiendas de aplicaciones de los sistemas operativos móviles que ofrecen un gran número de opciones para los usuarios, quienes pueden descargar e instalar videojuegos, administradores de finanzas, lectores de noticias, aplicaciones para video, redes sociales, chat, correo electrónico, calculadoras y un largo etcétera.

Estas aplicaciones, algunas de pago y muchas gratuitas, se han convertido en plataformas de negocio para los desarrolladores de software, en algunos casos, llevando a pequeñas *startups* a convertirse en grandes compañías en pocos años. Fenómenos de impacto como las compras recientes de aplicaciones móviles como WhatsApp por \$19.000 millones de dólares e Instagram por \$1.000 millones, las dos aplicaciones adquiridas por Facebook [9]. Estos y otros casos similares han motivado una ola de emprendedores que buscan lanzar sus proyectos para capturar la atención de este público masivo con servicios y negocios escalables que generen valor para los usuarios y oportunidades de compra por parte de las grandes empresas de tecnología.

Las tiendas de aplicaciones se organizan en categorías y permiten que los usuarios encuentren con facilidad las aplicaciones deseadas teniendo en cuenta sus gustos particulares. Algunas aplicaciones ofrecen contenido y mezclan funcionalidades diversas que pueden relacionarse con otros paquetes de software disponibles. En la Tabla 1-1 se agrupan 8 tipos comunes de aplicaciones basadas en su propósito general luego de analizar 24 categorías oficiales y equivalentes en las principales tiendas que distribuyen aplicaciones (Google Play, App Store y Windows Store). A continuación se describen de forma breve estos tipos comunes de aplicaciones disponibles en las tiendas mencionadas.

Tabla 1-1 Categorías de aplicaciones comunes en los sistemas operativos móviles.

Contenido y noticias	Aplicaciones enfocadas para el consumo de noticias y contenido de revistas, comics, catálogos, libros y similares. Ejemplos: Pulse, Filpboard, Feedly.
Mensajería	Aplicaciones de comunicación vía chat como WhatsApp, Line, Chat On. Así como clientes de correo electrónico como Gmail, Zoho, Outlook.
Navegación y mapas	Aplicaciones de geolocalización como Google Maps o servicios extendidos relacionados con transporte y mapas como Uber, EasyTaxi y Waze.
Multimedia	Aplicaciones centradas en consumo de contenido multimedia en especial audio, música y video como Youtube, Spotify y Shazam.
Sociales	Aplicaciones que mezclan el consumo de contenido y la interacción social como eje principal. Ejemplos de aplicaciones sociales por excelencia: Facebook, Twitter, Google+, Pinterest, Instagram.
Herramientas	Utilidades y herramientas de todo tipo, aplicaciones enfocadas a resolver una necesidad específica, a menudo diferente al consumo de contenido. Calculadoras, traductores y editores de imágenes.
Compras	Aplicaciones concentradas a la venta de productos. Amazon y Net-a-Porter son un ejemplo concreto. Algunas aplicaciones como Fancy ofrecen un modelo híbrido de aplicación de compras y red social.
Juegos y ocio	Una categoría particular y extensa que posee su propio ecosistema. Juegos de video de todo tipo como Candy Crush, Monument Valley y aplicaciones de ocio para entretenerse mientras se tiene tiempo libre.

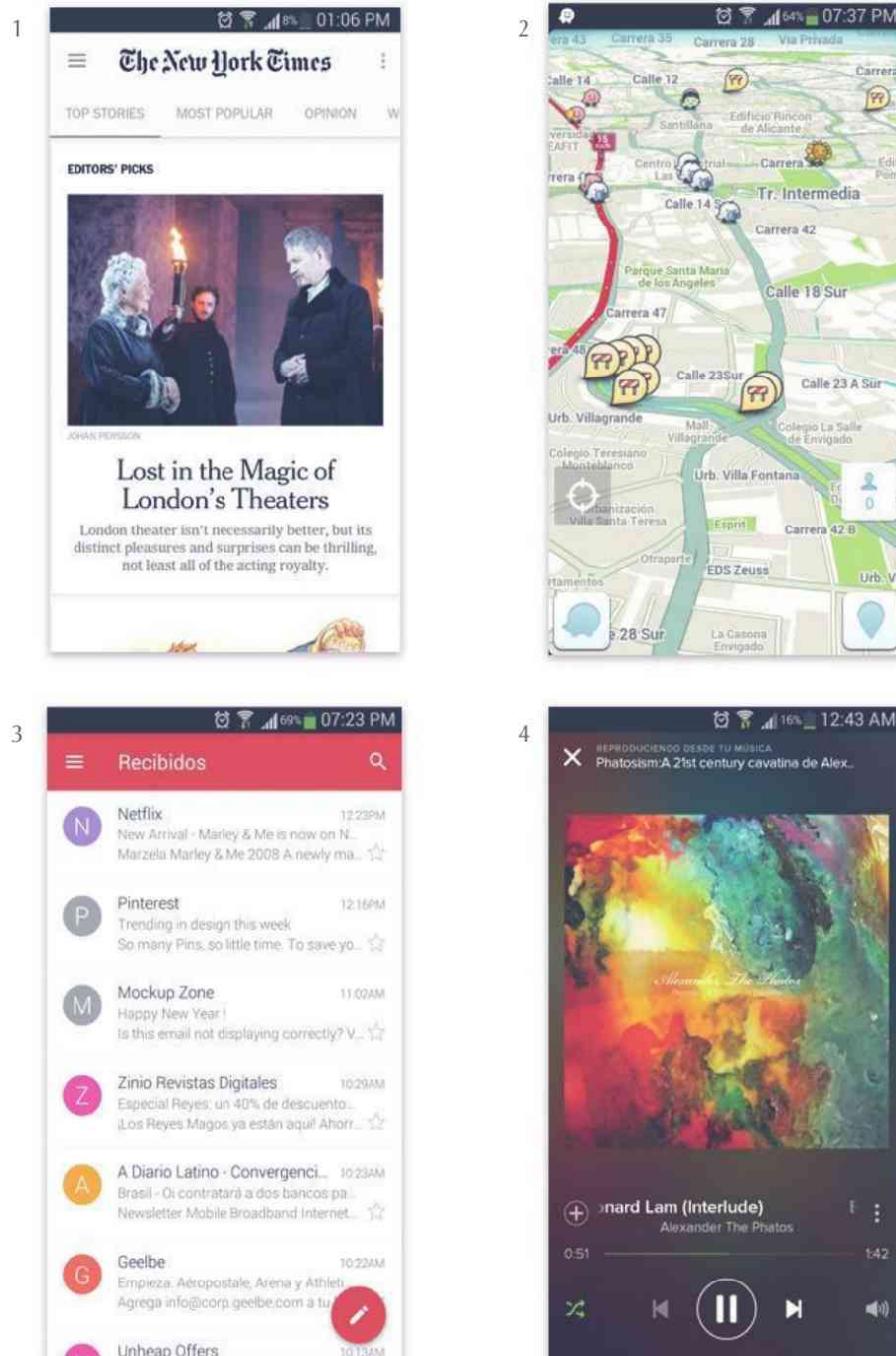


Figura 1-11 Ejemplos de aplicaciones: 1- The New York Times (contenido) 2- Waze (navegación) 3- Gmail (mensajería) 4- Spotify (multimedia).

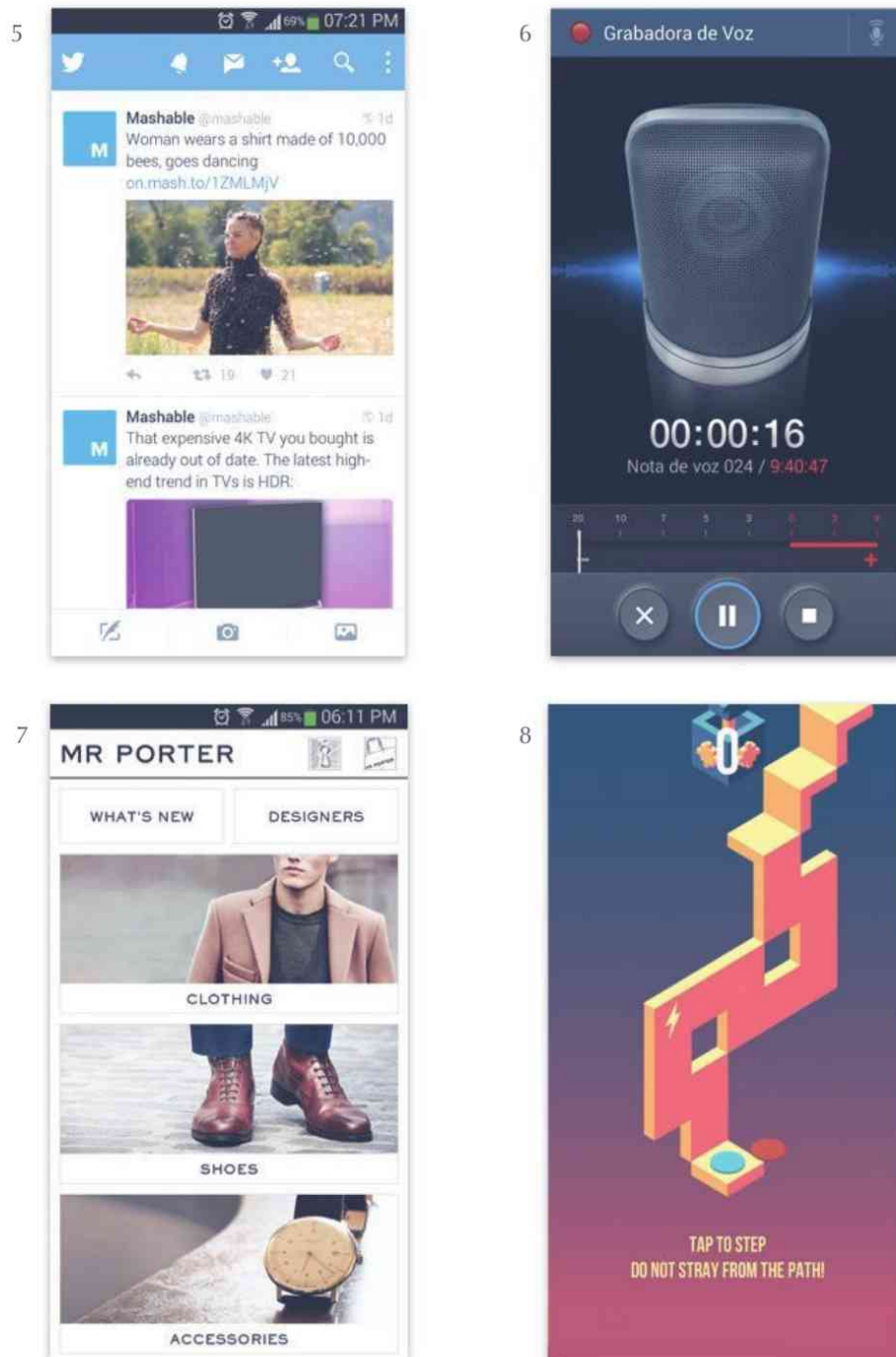


Figura 1-12 Ejemplos de aplicaciones: 5-Twitter (social) 6- Grabadora de Voz Samsung (herramienta) 7- Mr. Porter (compras) 8- Skyward (juego).

Mercados de aplicaciones

Los sistemas operativos móviles han traído consigo plataformas que le permiten a desarrolladores y empresas externas distribuir sus propias aplicaciones gratuitas o de pago a través de sus mercados de aplicaciones. Estas plataformas también se conocen como tiendas de aplicaciones y han tenido un crecimiento considerable desde el año 2008 cuando Apple presentó su App Store y Google su propia plataforma conocida en sus inicios como Android Market y luego renombrada como Google Play a inicios de 2012.

El modo de operación común de un mercado se realiza a través de una aplicación instalada por defecto en el dispositivo móvil, esta aplicación es el mercado en sí mismo, que permite a su vez instalar múltiples aplicaciones que el usuario desee. También se puede acceder a estos mercados desde un navegador web permitiendo instalarlas en el dispositivo móvil ingresando con una cuenta de usuario, funcionalidad muy utilizada en la Google Play.

Las aplicaciones permiten mejorar y personalizar un dispositivo, de esta forma dos usuarios pueden tener un iPhone o un equipo Android con las mismas prestaciones pero en cada caso será diferente ya que las aplicaciones entre uno y otro teléfono serán distintas en función de los gustos y necesidades del usuario propietario, quien instalará, a su criterio, las que le interesen o que considere necesarias.

Estas tiendas o mercados de software han influido en la venta masiva de los teléfonos inteligentes y, como se comentó en líneas anteriores, se organizan por medio de categorías más o menos coherentes entre sistemas operativos con una oferta de aplicaciones notable. Android y iOS en conjunto poseen más de dos millones de aplicaciones móviles disponibles en sus mercados de distribución [10], una cifra importante donde podemos observar el posicionamiento de la industria móvil como medio masivo.

El éxito de este modelo es tal que ha influenciado incluso la creación de mercados para sistemas operativos de escritorio como Mac OS y Windows que han incluido tiendas para descarga e instalación de aplicaciones con un funcionamiento muy similar a sus contrapartes móviles. Esto ha permitido expandir su estrategia global de distribución de software en los ordenadores personales de una manera similar a como lo han hecho en sus ecosistemas móviles.

Tabla 1-2 Mercados de aplicaciones móviles más conocidos

App Store	Es la tienda de aplicaciones para iOS, aunque no fue la primera tienda de aplicaciones moderna, marcó la tendencia desde el año 2008 con soporte para aplicaciones táctiles y con un modelo de negocio que permitió a los desarrolladores comercializar estas aplicaciones. Es conocida por tener un proceso de aprobación cerrado y riguroso con las aplicaciones que se distribuyen.
Google Play	Es la tienda oficial de aplicaciones Android, fue lanzada en el año 2008 con el nombre Android Market, pero en 2012 se actualizó por Google Play. En los servicios ofrecidos por esta plataforma se concentra el mayor número de aplicaciones Android y es el centro oficial donde se pueden descargar las demás contenidos de Google (música, libros y películas).
Windows Store	Microsoft presentó su plataforma en 2010 como Windows Marketplace y luego cambió su nombre por Windows Store para unificarla con la tienda de aplicaciones de su sistema operativo. En esta plataforma de comercialización se encuentran aplicaciones para su plataforma móvil y de escritorio.
BlackBerry World	Presentada en 2009 como la tienda de aplicaciones oficial para BlackBerry OS, se concentra en la distribución de aplicaciones para los desarrolladores del ecosistema de la empresa canadiense.
Amazon App Store	Esta tienda creada por Amazon se enfoca en la distribución de aplicaciones Android. Después de Google Play es el mayor mercado de aplicaciones Android existente. Ha tenido mucho éxito en la venta masiva de aplicaciones a través de los dispositivos Kindle Fire, tabletas que incluyen una versión modificada de Android que se enfoca de forma exclusiva en el ecosistema de servicios de Amazon.

Tipos de aplicaciones

En la actualidad, se reconocen tres tipologías de aplicaciones básicas que deben diferenciarse. Esta clasificación se concentra en el aspecto técnico, y se enfoca en los diferentes lenguajes soportados por los sistemas operativos móviles más comunes.

Aplicaciones nativas

Este tipo de aplicaciones utilizan los lenguajes de programación nativos del sistema operativo y utilizan todo el potencial de hardware de los terminales a través de paquetes de desarrollo del sistema. Por ejemplo, en Android las aplicaciones nativas se desarrollan en el lenguaje Java y se instalan como un paquete específico que se distribuye a través de los mercados de aplicaciones, al igual que en iOS con sus lenguajes Objective C y Swift, Windows Phone con C# y BlackBerry con Java. Es el tipo de aplicación más potente, pero suele ser compleja de desarrollar ya que requiere personal especializado en estas tecnologías. También se pueden crear aplicaciones nativas multiplataforma utilizando herramientas como Xamarin que permite desarrollar aplicaciones en código C# y convertirlo a otras plataformas como iOS y Android.

Los sistemas operativos móviles actuales promueven el desarrollo de aplicaciones nativas cuando se trata de software que requiere un uso importante del hardware y sensores. En estas situaciones una aplicación nativa es ideal, ya que el rendimiento observado y respuesta al tacto de los demás tipos de aplicaciones es todavía intermitente. Este tipo de respuesta se reconoce con facilidad en desarrollos híbridos o web, ya que HTML, la tecnología más utilizada para estos modelos, tiene limitaciones evidentes en la interacción con pantallas táctiles en dispositivos móviles.

En este libro se hace especial énfasis en las aplicaciones nativas como fenómeno reciente y los ejemplos del Capítulo 3 se enfocan en definir los elementos principales en el diseño de aplicaciones nativas. Sin embargo, muchos de los conceptos aplican para el diseño de aplicaciones híbridas o web porque en estas últimas se busca emular en gran medida la interfaz gráfica de las aplicaciones nativas, tanto que en muchas aplicaciones básicas centradas en contenido como noticias o artículos que ofrecen una interacción sencilla, es difícil identificar si se trata de una aplicación nativa o híbrida.

Aplicaciones híbridas

Las aplicaciones híbridas combinan diversas tecnologías de los lenguajes del sistema operativo fusionados con elementos web en su interfaz. En este tipo de aplicaciones es común utilizar elementos incrustados que presentan partes del navegador para visualizar la interfaz web. Estas por lo general son empaquetadas y distribuidas a través de los mercados de aplicaciones al igual que el software nativo. Sin embargo, la gran mayoría de estos desarrollos lo que hacen es incrustar un navegador en el espacio disponible en pantalla para presentar la aplicación que está construida en HTML, CSS y algún lenguaje de programación como Javascript, PHP, Ruby o Python, muy utilizados en *frameworks* de desarrollo para aplicaciones web.

Las aplicaciones híbridas son atractivas para muchas empresas porque el costo de desarrollo es más reducido y no requiere mano de obra especializada o capacitaciones adicionales para aprender nuevos lenguajes de programación, más allá de las tecnologías web conocidas. Además, existen muchos *frameworks* para desarrollo de aplicaciones híbridas multiplataforma que ayudan a simplificar el proceso de desarrollo como PhoneGap, Ionic y Titanium, por mencionar solo algunos.

Aplicaciones web

Este último tipo de aplicaciones se presenta como una respuesta enriquecida a los tradicionales sitios o servicios en línea. Su apariencia e interfaz puede ser similar a la de una aplicación nativa pero se diferencia en que utiliza en su totalidad tecnologías web. Una característica común en estas aplicaciones es su adaptación a móviles con técnicas como el *responsive* y *adaptive design*, permitiendo el uso de una sola plantilla de HTML que se adapta por medio de CSS o algunos lenguajes de servidor a la pantalla del navegador, evitando tener diversos sitios web o archivos que impliquen un mantenimiento de código mayor. Esta técnica también es utilizada en las aplicaciones híbridas.

La interacción general con aplicaciones web se realiza desde el navegador del dispositivo móvil a través de una dirección URL como ocurre con los sitios web tradicionales, no requieren una descarga o instalación en el teléfono o tableta desde mercados de aplicaciones como Google Play y App Store.

Disciplinas relevantes en el diseño de aplicaciones

Como fenómeno reciente, las aplicaciones móviles han tenido un auge notable en la industria del software y son ahora crecientes las cifras de desarrolladores que le están apostando a la creación de software móvil, situación que hace algunos años era poco común. Según un estudio realizado en 2014 por la empresa de investigación británica Vision Mobile, se estima que más de 2 millones de desarrolladores se dedican a crear aplicaciones móviles distribuidos a nivel mundial, sobre todo en Asia, Estados Unidos y Europa, un indicador de peso que reafirma la importancia creciente de este sector [11].

Si ya hacemos parte de esta comunidad de desarrolladores, es necesario mencionar que independiente del tipo de aplicación móvil que estemos creando: sea nativa, híbrida o web, existen disciplinas generales que intervienen en la construcción del producto final y conviene reconocerlas aunque sea de forma breve para establecer relaciones y a la vez diferencias en su aporte individual durante el proceso de desarrollo. En la Figura 1-13 se presentan estas disciplinas básicas con fines didácticos, sin el ánimo de ofrecer una clasificación absoluta, ya que en la práctica los límites entre ellas son difusos y es habitual que actividades que involucren una especialidad complementan a las demás.

Disciplinas transversales como la Experiencia de Usuario (UX) y la Interacción Persona-Ordenador (IPO) se comportan como integradoras entre diversas áreas de estudio, ambas muy relacionadas con las personas y los fenómenos que se presentan mientras interactúan con un producto o servicio. Estas disciplinas utilizan prácticas diversas que ayudan a mejorar la interacción del usuario con el software y el hardware a través de procesos de diseño, investigación y pruebas generales de atributos como la usabilidad y accesibilidad.

En el núcleo de la figura se representan en varias capas disciplinas más concretas que intervienen: el Diseño Visual, Diseño de Interacción (IxD), Arquitectura de Información (IA), Programación y Diseño Industrial como oficios con actividades específicas que permiten una definición de roles más clara entre los profesionales que intervienen. Dichas especialidades en conjunto son muy importantes en el diseño de todo tipo de aplicaciones móviles y se utilizan de igual manera en el desarrollo de software de escritorio o sistemas especializados para uso industrial. Estas disciplinas y sus aportes individuales se presentan como síntesis en la capa visible y operable de la interfaz de usuario (UI).



Figura 1-13 Disciplinas relevantes que intervienen en el diseño y desarrollo de aplicaciones. Como se observa en la figura los dos anillos principales corresponden a las disciplinas más amplias y con cierto nivel de abstracción como son la IPO (Interacción Persona-Ordenador) y la Experiencia de Usuario (UX). En el centro se representan las disciplinas más concretas en el proceso de diseño de una aplicación.

Experiencia de Usuario (UX)

Es un área de estudio y práctica reciente que tiene un enfoque amplio e interdisciplinario relacionado en el diseño de productos y su impacto emocional en las personas. Se conoce en inglés como *User Experience* (UX) y como oficio surgió durante los años noventa al interior de Apple Computer cuando un empleado de la compañía se autodenominó *User Experience Architect*. Este primer profesional en UX fue el conocido autor y teórico Donald Norman que acuñó su nuevo cargo para definir un enfoque más amplio que le permitiera crear productos integrales desde la funcionalidad, el diseño industrial, la usabilidad y su documentación [12].

Años después, con la popularidad de la Web, autores como Jesse James Garret y Peter Morville promovieron esta práctica como una visión holística del diseño de productos que va más allá del aspecto funcional, teniendo en cuenta cuestiones de contexto y de estrategia para satisfacer a los usuarios desde múltiples dimensiones [13], [14]. Dichas propuestas influenciaron a numerosos profesionales, empresas y asociaciones que desde entonces han adoptado de forma paulatina esta especialidad como un eje transversal en su estrategia de diseño, tanto que la Usability Professionals Association, conocida como UPA desde su fundación en 1991, decidió en 2012 renovar su nombre por UXPA (User Experience Professionals Association) al considerar que la Experiencia de Usuario acoge un área mayor donde por supuesto está incluida la usabilidad.

Esta amplitud que agrupa bajo una sola capa a otras disciplinas ha servido para relacionar aspectos comunes, pero al mismo tiempo ha generado confusión en la industria ya que en la práctica la etiqueta UX sigue siendo difusa y compleja de encasillar. Esta ambigüedad entre sus métodos y significado evidencian que aún se encuentra en una etapa de adolescencia donde tiene el reto de enfocarse para mantener su credibilidad más allá de las modas [15]. Al respecto, una de las confusiones más recurrentes en el desarrollo de software es la suposición de que diseñar la Interfaz de Usuario, conocida en inglés como *User Interface* (UI) es equivalente a diseñar la Experiencia de Usuario (UX), dos actividades muy relacionadas pero, como se puede ver en la Figura 1-13, la UX involucra todo el espectro de disciplinas en sinergia, mientras que la Interfaz de Usuario (UI) representa solo la superficie digital o espacio visible donde las personas interactúan con una aplicación. Esta superficie y su creación está muy ligada a la práctica del Diseño de Interacción y el Diseño Visual.

Interacción Persona-Ordenador (IPO)

Es otra disciplina amplia e integradora, pero en este caso más específica que la anterior. Mientras que la Experiencia de Usuario puede aplicarse como elemento estratégico en el diseño de cualquier producto o servicio como puede ser un automóvil o un restaurante, el enfoque principal de la Interacción Persona-Ordenador (IPO) se concentra de manera particular en el proceso donde la persona interactúa con un equipo informático.

Se conoce en inglés como *Human-Computer Interaction* (HCI) y su práctica ha estado muy ligada al desarrollo científico y empresarial desde hace más de tres décadas. Sin embargo, requiere mayor adopción en programas académicos, incluso en Estados Unidos donde por tradición ha tenido mucho apoyo de las universidades y organizaciones como la Association for Computing Machinery (ACM) que la promueve como una disciplina indispensable en las carreras relacionadas con las ciencias de la computación [16].

Es necesario aclarar que esta disciplina, aunque tiene un nombre muy específico, no se enfoca solo en el estudio de la interacción de las personas con ordenadores de escritorio o portátiles; el término también aplica para cualquier sistema informático creado para interactuar con el ser humano [17]. Es así como los teléfonos móviles, los equipos electrónicos como reproductores de música, tabletas y relojes inteligentes son también objeto de estudio de esta disciplina. La lista es extensa, ya que en la actualidad se utilizan incontables dispositivos operados a través de software en espacios cotidianos y entornos empresariales como se observa en el sector médico, de transporte, financiero, energético y en otras industrias especializadas.

Su abanico de actividades complementa las disciplinas concretas presentadas en la Figura 1-13 donde un punto fuerte de esta especialidad son las investigaciones de usuario y los métodos de pruebas de usabilidad y accesibilidad relacionados desde sus inicios con la IPO como prácticas que ayudan a mejorar el diseño de software y hardware apoyándose en métodos eficientes que permitan a las personas lograr sus objetivos de manera óptima con el apoyo de un sistema informático. Durante estas actividades de evaluación y diseño, se apoya en otras disciplinas generales como la ergonomía, etnografía, psicología, sociología, programación, diseño y documentación integrando algunos de sus métodos durante el proceso de desarrollo de software [18].

Diseño Visual

Esta disciplina se utiliza en casi todas las industrias para crear piezas gráficas que identifican productos o empresas por medio de logos, empaques y rótulos, así como la promoción de productos o servicios con diseños publicitarios en carteles y volantes. También posee una función informativa e instruccional que se evidencia a través de piezas de señalética y de contenido como letreros, libros y mapas. Es un arte aplicada que se considera en la práctica como sinónimo de Diseño Gráfico o Comunicación Visual, este último término más general fue promovido por el diseñador suizo Max Bill [19]. En el desarrollo de aplicaciones cumple un papel transversal en el diseño de la identidad de marca en el plano estético como actividad que se encarga de la definición de vocabularios visuales para el manejo de la información presentada, el color y la tipografía como elementos esenciales para transmitir con eficacia y singularidad unos valores estéticos muy específicos para cada software.

Diseño de Interacción (IxD)

Esta especialidad y sus prácticas ha estado muy ligada al Diseño Industrial y a la IPO en el proceso de creación de productos electrónicos que operan a través de software. Como disciplina, surgió en los años ochenta en pleno auge de los ordenadores personales y se ha relacionado desde entonces con el diseño de interfaces digitales. El término, que en inglés se conoce como *Interaction Design* (IxD), fue acuñado por el diseñador británico Bill Moggridge quien diseñó el primer ordenador portátil (el GRiD Compass) experiencia que lo motivó para definir una disciplina equivalente al Diseño Industrial pero en el área digital que se encargara de reconocer al usuario y sus necesidades así como diseñar formas, animaciones, sonidos y secuencias entre los diferentes puntos de contacto con un sistema digital [20].

También se puede analizar desde una visión social como una disciplina que va más allá de los productos electrónicos y se relaciona con la interacción de las personas con cualquier tipo de objeto o entorno [21]. Sin embargo, en este libro por motivos prácticos asumiremos la función del Diseño de Interacción desde la perspectiva de Moggridge, como apoyo en el diseño de software, lo que implica reducir su alcance a la definición del comportamiento de un producto digital y la disposición de elementos interactivos en pantalla definiendo flujos, secuencias y sonidos en el diseño de la interfaz de usuario.

Arquitectura de Información (IA)

Richard Saul Wurman definió en los años setenta y décadas posteriores el norte de esta disciplina como oficio, pero no fue hasta el año 1998 cuando fue popularizada por Louis Rosenfeld y Peter Morville en su libro “Information Architecture for the World Wide Web” trabajo que definió las bases y componentes de esta especialidad en el área digital: los sistemas de organización, navegación, etiquetado y búsqueda [22]. Conceptos que se siguen utilizando en la creación de contenidos, diagramas y estructuras para organizar la información en pantalla de manera lógica y consistente para el usuario, sea en un sitio web, una aplicación móvil o cualquier sistema digital. Dichos elementos constituyen una actividad previa indispensable para el diseño posterior de la interfaz de usuario.

Programación

Es una disciplina que se encarga de codificar a través de lenguajes de programación la funcionalidad de las aplicaciones. Está muy ligada a la ingeniería del software y las ciencias de la computación como un área extensa que aborda temas diversos desde la creación de programas de cálculo sencillos, hasta sistemas complejos basados en inteligencia artificial. En el proceso de diseño de aplicaciones se reconocen dos capas conocidas como: programación *front-end* y programación *back-end*. Las actividades de implementación de la interfaz gráfica se asocian a la programación *front-end* que se concentra en lenguajes de etiquetado menos complejos.

Diseño Industrial

Como disciplina abarca el diseño y fabricación de todo tipo de objetos físicos, desde sillas y relojes, hasta vehículos como automóviles y aviones. En el caso particular que nos interesa, el diseño y creación de dispositivos móviles es tarea de esta disciplina. Mencionarla es importante porque muchos de los elementos virtuales que se utilizan para diseñar interfaces gráficas se inspiran en los controles y elementos utilizados en la práctica del Diseño Industrial. En el proceso de interacción con un teléfono móvil se hace tangible esta disciplina donde el hardware y el software se fusionan de una manera complementaria permitiendo interactuar con la interfaz gráfica en pantalla y al mismo tiempo accionar botones físicos en el dispositivo móvil.

Importancia del diseño de interfaces

Como pudimos ver en páginas anteriores, aunque queramos aislar las disciplinas mencionadas, el aporte de cada una se complementa a las demás durante el proceso de diseño de forma inseparable. La síntesis de todas estas disciplinas se presenta en la capa visible que permite al usuario interactuar con la aplicación conocida como la interfaz gráfica, por eso en muchas ocasiones se utiliza la analogía de la interfaz como “la punta del iceberg” de un software ya que representa la capa superior con la que el usuario tiene contacto directo pero, al igual que un iceberg, depende de sus partes invisibles [23]. En nuestro caso, disciplinas como la programación y la Arquitectura de Información son imperceptibles para el usuario final quien concibe la aplicación (en su totalidad) como lo que ve en pantalla, y es difícil para este comprender que la interfaz es solo una parte del sistema [24].

Por ese motivo, las decisiones y cambios en las capas visibles de una aplicación afectarán en forma notable en la percepción del usuario, situación que obliga a los desarrolladores y diseñadores a tener una visión estandarizada de la interfaz como un recurso valioso que debe ser gestionado con rigor y cautela. Al respecto, en los últimos años muchas empresas de tecnología han comenzado a estandarizar sus aplicaciones y software, no solo desde el punto de vista del código fuente, sino también desde el diseño e interacción particular del sistema. En el contexto móvil retoma con gran importancia la adopción de dichas estrategias dada la fuerte competencia de mercado donde, para el usuario final, quien por lo general no conoce nada de lenguajes de programación, el diseño es un elemento determinante que puede generar un impacto en su percepción general con la marca y sus aplicaciones.

Esta relación se establece desde una perspectiva subjetiva donde el usuario se fideliza a través de su experiencia y una comunicación estandarizada con el software. Este enfoque no es nuevo, desde los años setenta las empresas de tecnología comenzaron esta carrera por imponer sus hegemonías desde el diseño, inspirándose a lo largo de las décadas: Xerox como punto de partida en la creación de la primera interfaz gráfica con la metáfora del escritorio, Apple como punto de maduración del ordenador personal y Microsoft como seguidor de la tendencia durante varias décadas. Esta competitividad consolidó el diseño de interfaces como un elemento estratégico que puede mejorar la experiencia de los usuarios y el posicionamiento de las empresas.

Lenguajes de diseño emergentes

Desde los años ochenta se han utilizado lenguajes de diseño en la industria informática, un caso pionero es el de Snow White, proyecto liderado por el diseñador alemán Hartmut Esslinger quien definió este lenguaje para estandarizar el *look & feel* de los equipos Macintosh de Apple, aunque solo se enfocó en el diseño industrial más que en la interfaz gráfica [25]. Estos lenguajes ofrecen un marco de referencia para definir un vocabulario visual compartido entre los productos de una marca, unificando principios que sugieren una dirección estética y conceptual para el diseño de interfaces físicas y digitales. Un ejemplo reciente de estos vocabularios es lo que está ocurriendo con los sistemas operativos móviles de Microsoft y Google, empresas que presentaron sus lenguajes Modern UI en 2010 y Material Design en 2014 influenciando de forma notable la escena actual del diseño de interfaces.

En cierto modo, estas innovaciones en el diseño de interfaces se han realizado desde el concepto de diseño *Mobile First*, acuñado por Luke Wroblewski como un enfoque que propone diseñar la interfaz primero en su versión móvil y luego adaptarla a los demás soportes, como una manera de abordar el proceso de diseño desde las funcionalidades esenciales. Aunque es una práctica más común en sitios web, también aplica en el diseño de aplicaciones y software nativo [26]. La importancia de dicha tendencia es tal que si observamos los eventos anuales como el Google I/O, Apple WWDC y Microsoft PDC donde se presentan las novedades de cada empresa, cada año desde 2008, la expectativa principal ha sido la presentación de sus nuevos dispositivos y software móviles, dejando en segundo plano a los ordenadores de escritorio que aunque siguen siendo importantes, los desarrollos en tecnologías móviles están teniendo un protagonismo absoluto siendo los primeros en incluir novedades en teléfonos, tabletas y relojes que despiertan gran interés en los usuarios.

En este panorama dominado por los sistemas operativos móviles, los lenguajes de diseño emergen y se utilizan para establecer principios y lineamientos en la creación de interfaces atractivas y consistentes, pero más allá de ser unas guías de estilo rigurosas, plantean un enfoque aspiracional del diseño. De esta manera se están convirtiendo en un activo para las empresas de tecnología al fijar sus estándares en la mente de los usuarios a través de su modo de interacción y estética particular. Una estrategia muy ligada a la identidad de marca que aporta valor adicional en el diseño de software.

Modern UI (Microsoft)

En 2010 Microsoft presentó Windows Phone como una respuesta inesperada en el panorama móvil, un cambio que puede tomarse como el más extremo y radical que Microsoft ha presentado en toda su historia. Este nuevo Windows fue creado siguiendo los principios de un lenguaje de diseño conocido en un primer momento como Metro Style pero luego renombrado como Modern UI. Este sistema utiliza una interfaz minimalista, casi tipográfica y un modo de interacción muy específico con un diseño que popularizó las interfaces gráficas de colores planos, tendencia conocida en inglés como *flat design*.

La creación de este lenguaje fue dirigida por Jeff Fong, líder de diseño en Microsoft, quien por sus innovaciones en la nueva interfaz de Windows Phone fue reconocido por la revista Fast Company como una de las personas más creativas de 2012. Las inspiraciones de Modern UI se remontan a lo que supuso la escuela de diseño alemana Bauhaus, en donde se reafirmó el valor del diseño y su funcionalidad por encima de los adornos destacando la importancia de comunicar desde lo esencial [27]. Búsqueda reflejada en la interfaz de usuario de Microsoft que ha incluido nuevos elementos de interacción como los *tiles*: grupos de mosaicos que representan aplicaciones y atajos en la pantalla de inicio del sistema con un estilo que recuerda la señalética de los lugares físicos.

Modern UI integra un sistema de iconos conocidos como Segoe que se utilizan en los espacios y barras de acciones de la interfaz. También, el uso del texto es protagonista, inspirado en el estilo tipográfico internacional o estilo suízo que promueve un diseño minimalista y claro. Asimismo los colores vivos son comunes y se emplean en los bloques de *tiles* del sistema de forma recurrente y apoyándose en efectos sutiles de animación.

Microsoft con Modern UI ha tenido buenas críticas de la prensa especializada y público general, pero tiene retos importantes ya que confronta a los usuarios tradicionales con este cambio radical que implica un nuevo aprendizaje para quienes se acostumbraron a la antigua interacción de Windows. Pese a esto, desde 2012 Microsoft ha utilizado este lenguaje de diseño en todas sus aplicaciones que fueron integradas en 2015 en el sistema operativo Windows 10 e incluso en el software principal de la consola de videojuegos X-BOX como una estrategia de Microsoft para estandarizar su estilo en todas las plataformas y diferenciarse de las demás empresas de tecnología.

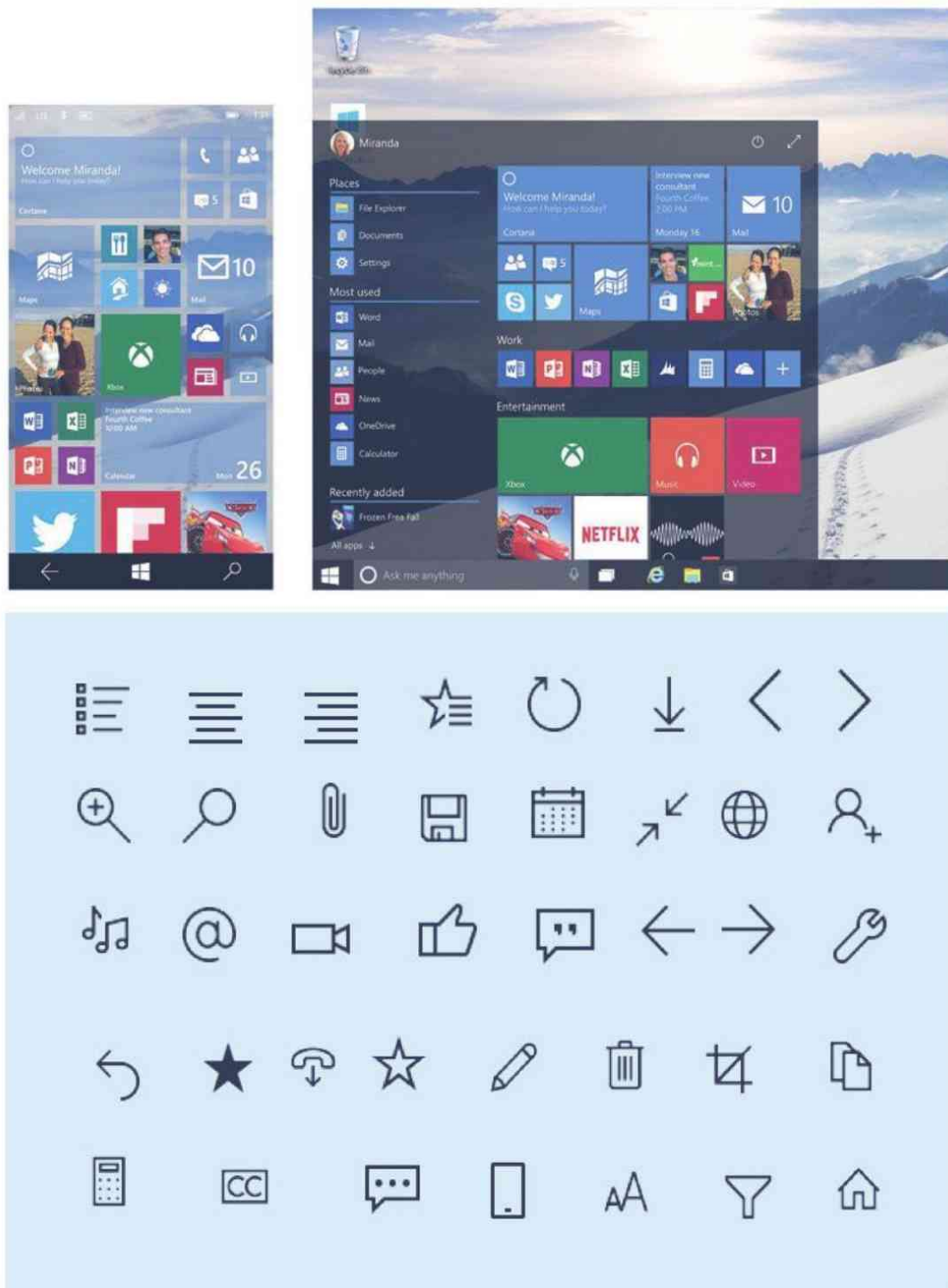


Figura 1-14 Arriba: Pantallas principal de Windows para teléfonos y ordenadores presentando la interfaz con estilo Modern UI. Abajo: Algunos iconos Segoe, este paquete de iconos y su tipografía del mismo nombre es la oficial empleada por el sistema operativo de Microsoft.

Material Design (Google)

Android tuvo en sus inicios un desarrollo acelerado, se lanzaban dos versiones del sistema operativo al año, esto permitió una evolución ágil en términos funcionales y estéticos pero acentuó la conocida fragmentación del sistema tanto en versiones del código y también en su interfaz de usuario.

Para mitigar esta situación, desde el año 2010 Google comenzó un proceso de maduración de la plataforma y su diseño buscando trascender la estética anterior y abordando un enfoque más estratégico de sus marcas. Ese mismo año contrató a Matías Duarte como director de diseño para el equipo de Android quien inició el desarrollo de lo que sería Material Design. Duarte trabajó en el pasado con Andy Rubín y en años recientes fue el responsable de la creación de la interfaz del sistema operativo WebOS de Palm. Su misión en Google fue unificar las interfaces de usuario para Android. La primera versión que incluyó un rediseño dirigido por Duarte fue Android 3.0 (Honeycomb) compatible solo con tabletas, seguido se lanzó Android 4.0 (Ice Cream Sandwich) que unificó la interfaz tanto para tabletas como teléfonos móviles con una consistencia mayor utilizando un diseño plano y una tipografía conocida como Roboto [28]. Esta versión fue la semilla del nuevo lenguaje de diseño de Google.

Material Design fue presentado de forma oficial en 2014 con Android 5.0 (Lollipop) como una conciliación entre el estilo plano que ha adoptado Microsoft y una nueva concepción espacial que se inspira en la metáfora del papel físico empleando sombras para generar profundidad o volumen en los elementos de interfaz, esta característica se conoce como elevación (*elevation*) y se aprecia en el modo básico de sus nuevos botones flotantes, conocidos como FAB (Floating Action Buttons), que se ubican en capas planas superpuestas en varios niveles. También utiliza de una manera notable animaciones sofisticadas entre pantallas con un comportamiento dinámico y minimalista. Algunos se refieren a este nuevo estilo como *flat 2.0* [29].

Google ha comenzado a integrar este lenguaje de diseño en todas sus aplicaciones buscando unificar el diseño y la interacción de los usuarios en todo su ecosistema tanto móvil como web. Sin embargo, un reto para la compañía continúa siendo la dependencia a los fabricantes de teléfonos y tabletas que se resisten a cambiar sus propias interfaces propietarias, pues la personalización del diseño fue uno de los atractivos de Android en sus primeros años.

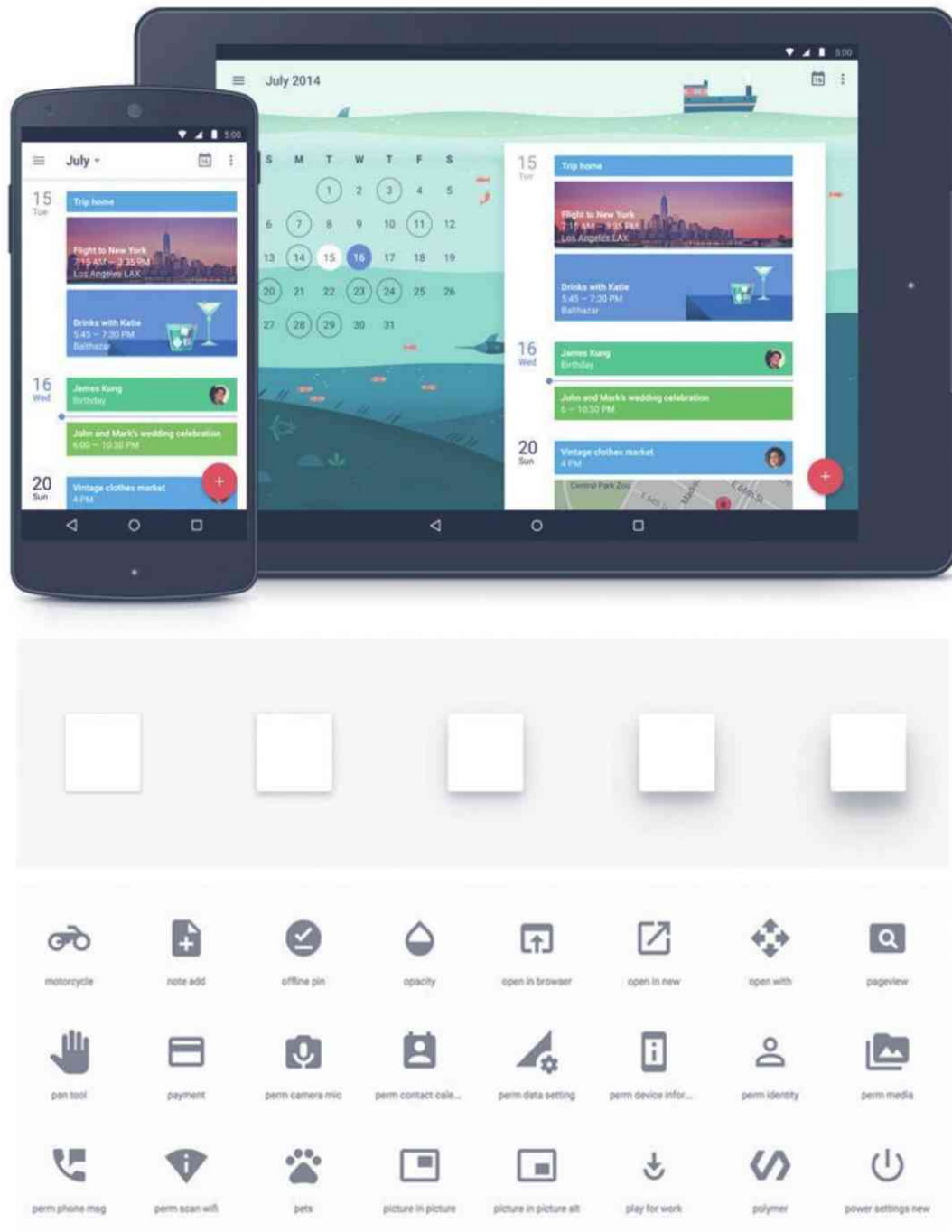


Figura 1-15 Material Design posee un estilo plano y elementos con sombras para generar sensación de profundidad, también posee su propio sistema de iconos que pueden utilizarse en aplicaciones de terceros. Aplicaciones de Google como Calendar, Gmail y Youtube han adoptado las convenciones de este lenguaje.

Post-Aqua UI (Apple)

Es innegable la influencia que ha tenido Apple en la industria de la computación al crear productos que han marcado hitos del diseño industrial y de interfaces durante más de tres décadas. Sin embargo, la ausencia reciente de Steve Jobs y la toma de mando de Tim Cook ha generado fricciones internas en la compañía. Estos cambios se acentuaron en el año 2012 con la salida forzada de Scott Forstall [30] quien dirigió el diseño de la interfaz Aqua presentada en el año 2000 y el desarrollo de iOS desde sus inicios como dos proyectos icónicos de Apple. El enfoque de Forstall estuvo muy coordinado con los intereses de Steve Jobs en el diseño de interfaces, este último partidario del estilo visual realista que utiliza texturas simulando materiales como madera, vidrio y metal en las interfaces digitales, conocido como *skeuomorphism* [31].

En 2013, luego de la salida de Forstall, Apple presentó una nueva versión del sistema operativo (iOS 7) con un marcado rediseño de su interfaz gráfica utilizando un estilo plano. Este cambio de norte fue dirigido por Jonathan Ive, quien ha trabajado desde los años noventa en Apple liderando el área de diseño industrial y ahora también ha tomado el control creativo del diseño de interfaces gráficas de la compañía. Este diseño no fue presentado con un nombre oficial, algo poco común en Apple, por eso en este apartado con una intención didáctica le llamamos “Post-Aqua” haciendo referencia a la transición del nuevo estilo visual plano de Apple con respecto a la antigua interfaz Aqua que se mantuvo durante 14 años en todas sus aplicaciones.

El nuevo diseño deja atrás la apariencia realista, reemplazando las texturas por colores en bloque e iconos llamativos, un cambio que ha generado opiniones divididas entre los usuarios y la prensa. Aunque conserva los paradigmas existentes en iOS y Mac OS, principios documentados en su conocida guía Human Interface Guidelines, la nueva interfaz abandona los adornos favoreciendo el uso de colores y transparencias que siguen en parte el estilo minimalista popularizado por Microsoft en 2010. Estos cambios representan una renovación superficial de la apariencia estética con respecto a su anterior diseño característico inspirado en metáforas del mundo físico. Este apariencia de iOS se ha integrado de forma gradual en el sistema Mac OS unificando el estilo visual en todas las plataformas y aplicaciones tanto móviles como de escritorio de Apple, similar a como han hecho Google y Microsoft con sus lenguajes de diseño integrados en sus ecosistemas de aplicaciones y productos.

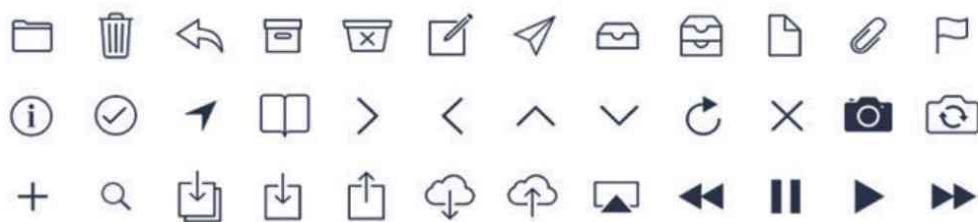


Figura 1-16 La nueva interfaz de iOS conserva los elementos característicos del sistema pero cambia la apariencia estética y texturas por iconos minimalistas y predominancia de colores planos.

REFERENCIAS Y LECTURAS DE INTERÉS

CAPÍTULO 1

-
- [1] A. Hoog, *Android Forensics: Investigation, Analysis and Mobile Security for Android*. Syngress, 2011. pp. 3-7.
- [2] «Microsoft adquiere la división de servicios y dispositivos de Nokia, licencia las patentes de la compañía y el servicio de mapas», Microsoft, 09-mar-2013. [Disponible en línea] <http://interfacesmoviles.com/ref/web1>
- [3] «Research In Motion to Acquire QNX Software Unit from Harman International - Press Releases», QNX, 04-sep-2010. [Disponible en línea] <http://interfacesmoviles.com/ref/web2>
- [4] «Samsung Launches Industry's First Tizen Smartphone – the Samsung Z», 02-jun-2014. [Disponible en línea] <http://interfacesmoviles.com/ref/web3>
- [5] A. Souppouris, «Ex-Nokia staff to build MeeGo-based smartphones», The Verge, 07-jul-2012. [Disponible en línea] <http://interfacesmoviles.com/ref/web4>
- [6] N. Lomas, «Ubuntu Edge Smartphone Fails To Hit Ambitious \$32M Crowdfunding Target», TechCrunch, 22-ago-2013. [Disponible en línea] <http://interfacesmoviles.com/ref/web5>
- [7] «Gartner Says Worldwide Smartphone Sales Grew 9.7 Percent in Fourth Quarter of 2015: Global Smartphone Sales», 18-feb-2016. [Disponible en línea] <http://interfacesmoviles.com/ref/web6>
- [8] D. A. Norman, *El Ordenador Invisible*. Barcelona: Paidós, 2000. p. 67.
- [9] A. Hartung, «Three Smart Lessons From Facebook's Purchase Of WhatsApp», Forbes (Leadership), 24-feb-2014. [Disponible en línea] <http://interfacesmoviles.com/ref/web7>
- [10] D. Smith, «Google Play has more apps than Apple now, but it's still behind in one key area», Business Insider, 02-feb-2015. [Disponible en línea] <http://interfacesmoviles.com/ref/web8>
- [11] «Developer Economics Q1 2014: State of the Developer Nation», Vision Mobile (research series), feb-2014. pp. 9-10. [Disponible en línea] <http://interfacesmoviles.com/ref/web9>
- [12] P. Merholz, «Peter in Conversation with Don Norman About UX & Innovation», Adaptive Path, 13-dic-2007. [Disponible en línea] <http://interfacesmoviles.com/ref/web10>
- [13] J. J. Garret, «The Elements of User Experience» (diagram), 30-mar-2000. [Disponible en línea] <http://interfacesmoviles.com/ref/web11>

- [14] P. Morville, «User Experience Design», Semantic Studios, 21-jun-2004. [Disponible en línea]
<http://interfacesmoviles.com/ref/web12>
- [15] J. Innes, «UX Strategy: Fad or New World Order?», UXPA Magazine, sep-2015. [Disponible en línea]
<http://interfacesmoviles.com/ref/web13>
- [16] A. Dix, J. Finlay, G. D. Abowd, y R. Beale, Human-Computer Interaction, 3.^a ed. United Kingdom: Prentice-Hall Europe, 2004. pp. 7-10.
- [17] Hewett, *et al.*, «ACM SIGCHI Curricula for Human-Computer Interaction», 1992. [Disponible en línea]
<http://interfacesmoviles.com/ref/web14>
- [18] T. Granollers, J. Lorés, J. J. Cañas. Diseño de sistemas interactivos centrados en el usuario. Barcelona: Editorial UOC, 2005. pp. 32-33.
- [19] Q. Newark, ¿Qué es el Diseño Gráfico? Traducido por Joan Escófet. Barcelona: Gustavo Gili, 2002. pp.12-13.
- [20] B. Moggridge, Designing Interactions. Cambridge: MIT Press, 2007. pp. 9-14.
- [21] D. Saffer, Designing for Interaction. California: New Riders, 2010. pp. 5-10.
- [22] P. Morville y L. Rosenfeld, Information Architecture for the World Wide Web: 3.^a ed. Sebastopol, California: O'Reilly Media, 2006. p.43.
- [23] D. Berry, «The user experience: The iceberg analogy of usability», IBM developerWorks (Technical library), 01-oct-2000. [Disponible en línea]
<http://interfacesmoviles.com/ref/web15>
- [24] J. Raskin, The Humane Interface: New Directions for Designing Interactive Systems. New York: Addison-Wesley, 2000. pp. 5-6.
- [25] H. Esslinger, Keep It Simple: The Early Design Years of Apple. Arnoldsche Verlagsanstalt, 2014. pp. 3-5.
- [26] L. Wroblewski, Mobile First. New York: A Book Apart, 2011. pp. 14-17.
- [27] S. Clayton, «Modern Design», Microsoft Corporation. [Disponible en línea]
<http://interfacesmoviles.com/ref/web16>
- [28] J. Topolsky, «Matias Duarte on the philosophy of Android, and an in-depth look at Ice Cream Sandwich», The Verge, 18-oct-2011. [Disponible en línea]
<http://interfacesmoviles.com/ref/web17>
- [29] K. Mayer, «Flat Design: Its Origins, Its Problems, and Why Flat 2.0 Is Better for Users», Nielsen Norman Group, 27-sep-2015. [Disponible en línea]
<http://interfacesmoviles.com/ref/web18>
- [30] C. Bonnington, «Apple Announces Major Executive Shakeup», Wired, 29-oct-2012. [Disponible en línea]
<http://interfacesmoviles.com/ref/web19>
- [31] N. Carlson, «The Untold Story Of How Steve Jobs Reintroduced His Signature Design Style To Apple», Business Insider, 19-oct-2014. [Disponible en línea]
<http://interfacesmoviles.com/ref/web20>

2

ANÁLISIS Y ARQUITECTURA

En este apartado se estudian temas de análisis e investigación de usuarios como las entrevistas, encuestas, grupos focales, personas y escenarios como apoyo inicial en ese reconocimiento del público objetivo de un proyecto móvil y se aborda la definición de la Arquitectura de Información presentando algunos métodos básicos como el *card sorting*, la creación de diagramas y los tipos de navegación comunes en aplicaciones.

Planeación y definición

Luego de observar el panorama de los sistemas operativos móviles y aplicaciones, podemos reconocer que un gran número de empresas y desarrolladores están apostando por este nuevo ecosistema y sus modelos de negocio, ya sea como extensión de sus propias empresas o en el desarrollo de proyectos innovadores que buscan romper con esquemas preestablecidos. Cuando nos enfrentamos a un proyecto de diseño de aplicación móvil debemos reconocer que es un mercado muy competitivo que implica retos importantes si queremos ofrecer soluciones eficaces y que a la vez se diferencien de la competencia.

Para ello es muy recomendable que todo el equipo de desarrollo y personas involucradas puedan participar de este proceso. Como pudimos observar en el Capítulo 1 donde se presentaron las disciplinas básicas relacionadas, el diseño de la interfaz no es una actividad reservada del diseño visual, sino que se apoya en la coordinación del equipo en una labor interdisciplinaria.

Una visión formal en el proceso de diseño supone tener un equipo con múltiples profesionales, programadores, diseñadores visuales, diseñadores de interacción, arquitectos de información, entre otros que aportan conocimientos valiosos. Sin embargo, algunas empresas pueden tener diferentes nombres para referirse a estos roles y en otras los miembros del equipo pueden ejecutar actividades de varias disciplinas sin tener un rol separado por cada una. Cada empresa es única y tiene sus propios métodos para estructurar los proyectos que desarrolla, sin embargo, independiente de la forma de organizar un equipo de diseño con sus roles cada vez más diversos [1] lo importante es asegurar la incorporación de las disciplinas básicas mencionadas en el Capítulo 1 durante el proceso de desarrollo [2].

También puede ocurrir que no se tenga un equipo numeroso para la ejecución del proyecto, en especial si se trata de una *startup* conformada por pocas personas. En este último caso no quiere decir que se realizará un proyecto de baja calidad, esto es relativo, ya que si se desarrolla una solución que tenga en cuenta las necesidades del usuario, en muchos proyectos, incluso un equipo reducido puede desarrollar una aplicación móvil superior a la de las grandes empresas que desperdician tiempo valioso en procesos burocráticos que no les permiten avanzar en la ejecución de sus proyectos [3]. Algo frecuente en la industria móvil donde pequeños equipos pueden desarrollar grandes soluciones.

Un primer paso en el proceso de análisis es reconocer a nuestro cliente y sus objetivos comerciales, incluso si no existe un cliente y se trata de nuestra propia empresa que desarrolla una aplicación móvil, es importante tener presentes los objetivos del negocio, ya que diseñar una interfaz no solo implica satisfacer las necesidades del usuario, también debe garantizar que el diseño soporte un modelo comercial, aquí es donde se debe equilibrar lo que se ofrece como funcionalidad y los objetivos globales de la empresa para tener una visión general del diseño como un elemento estratégico.

Luego de tener claridad en los objetivos comerciales conviene ubicar al usuario en el panorama. Para esto una práctica común es utilizar el enfoque del Diseño Centrado en el Usuario, del inglés *User-Centered Design* (UCD), método en el que los diseños se validan con personas reales durante todo el proceso de desarrollo. Esta filosofía promueve el diseño iterativo de productos teniendo en cuenta las necesidades del usuario, el contexto de uso, la producción de soluciones de diseño y la evaluación con pruebas de usabilidad para validar los diseños (Figura 2-1). Esta metodología, aunque puede ser integrada al flujo de trabajo de manera informal, también tiene un estándar relacionado: la norma ISO 9241-210 antes conocida como ISO 13407 [4], [5].



Figura 2-1 Gráfico que representa el proceso de Diseño Centrado en el Usuario (UCD), actividad cíclica relacionada con la norma ISO 9241-210 (antes conocida como ISO 13407).

Sin embargo, sea que utilicemos un proceso formal como el que se especifica en la norma ISO o utilicemos nuestra propia metodología particular, supone una premisa importante conocer al usuario y diferenciarlo de nosotros mismos. Este enfoque tiene muchos beneficios para el desarrollo de un proyecto, ya que evita riesgos y ayuda a esclarecer las necesidades específicas que los usuarios tienen en la realización de determinadas tareas y con base en esto ofrecer soluciones de diseño.

Por otra parte, también debemos tener en cuenta el contexto de uso al momento de utilizar la aplicación. En este punto, es de vital importancia identificar de manera adecuada el entorno, ya que el uso de una aplicación móvil, como no ocurre con un ordenador de escritorio, puede variar de forma muy dinámica [6]. Por ejemplo, una aplicación de chat puede ser utilizada con mayor o menor dificultad cuando el usuario está en la calle, en la oficina o en el supermercado. Este tipo de aplicación, aunque posee una interfaz que considerada “estándar”, en determinado contexto puede entorpecer algunas actividades por su interacción básica, ya que requiere de una atención del usuario quien debe sostener el teléfono para escribir mensajes. Utilizar esta aplicación de chat mientras se conduce un vehículo puede ser peligroso hasta el punto de provocar accidentes debido a las distracciones que tendrá el usuario al interactuar con una interfaz en un contexto para el que no fue diseñada.

Las aplicaciones basadas en geolocalización donde el usuario se encuentra en un automóvil son un ejemplo crítico en el proceso de diseño, ya que este contexto particular obliga a tener soluciones especiales para que la experiencia de ese usuario sea memorable y a la vez segura [7]. En dichos escenarios, el teléfono se puede ubicar en soportes físicos que lo sostienen y se utilizan sistemas basados en lectura y reconocimiento de voz. También el diseño de la interfaz gráfica puede personalizarse utilizando botones de mayor tamaño para permitir una selección más cómoda por parte del usuario; estas soluciones tienen mucha relevancia en dichos contextos de uso pero en otro tipo de aplicaciones pueden ser innecesarias. Este ejemplo evidencia que las decisiones de diseño no son aleatorias, se realizan luego de analizar al usuario y su contexto para comprender dónde se utilizará la aplicación, sin suponer o crear hipótesis que no han sido validadas con personas reales. Para esto podemos utilizar métodos de investigación que nos ayudan a recolectar datos específicos y construir un perfil de ese público objetivo y sus necesidades.

Identificando a los usuarios

Los métodos de investigación de usuarios se utilizan desde hace décadas y existen multitud de procedimientos, pero aquí nos concentramos en presentar en forma breve algunos métodos básicos en los procesos iniciales del desarrollo. Las pruebas de usabilidad también se incluyen en los métodos de investigación de usuarios pero aquí, como estamos en las etapas tempranas, no profundizaremos en este aspecto, ya que es necesario tener una aplicación desarrollada o un prototipo de interfaz (en papel o interactivo) para realizar las pruebas, este método se aborda en el Capítulo 4.

Entrevistas contextuales

Son actividades que se realizan en el contexto propio del usuario y están muy relacionadas con la observación etnográfica o de campo, método utilizado para acercarse y analizar a las personas en su entorno cotidiano. Por lo general se realizan de manera individual dirigiéndose al lugar de trabajo o vivienda del usuario para observar cómo este realiza las tareas diarias y tener más claridad sobre el flujo de acciones que realiza e identificar patrones de conducta que pueden ser útiles para mejorar o definir el diseño de nuestra aplicación móvil. Al finalizar se realizan sesiones de preguntas al usuario para obtener más información y aclarar dudas sobre lo observado. Los datos obtenidos con este método son cualitativos y representan información valiosa para definir escenarios realistas que se pueden utilizar en pruebas de usabilidad [8].

Encuestas

Es un método convencional que consiste en elaborar una lista de preguntas con las cuales queremos obtener información concreta. Hay encuestas que buscan identificar las horas en las que un determinado tipo de usuario utiliza un dispositivo móvil para leer noticias, identificar la edad, el sexo y la ubicación geográfica del público objetivo o recolectar información que puede ser útil para un análisis posterior. Por lo general se realizan en persona, pero también se realizan a través de Internet donde un gran número de usuarios pueden completarlas sin tener que desplazarse a un lugar concreto. Ofrece resultados cuantitativos muy específicos que pueden arrojar información útil para construir los perfiles de usuarios. Para realizar encuestas remotas existen multitud de herramientas, algunas conocidas son Jotform y SurveyMonkey.

Grupos focales

Se utilizan como método de análisis de las personas y sus opiniones particulares. En el formato habitual se trata de una sesión con varios usuarios que discuten un tema específico o incluso se pueden emplear como método de comparación de la competencia, presentando a los usuarios aplicaciones específicas y preguntando sobre sus percepciones o puntos de vista. Un grupo focal debe tener un moderador que guía la discusión hacia los aspectos más críticos que deben ser evaluados para garantizar que no se desvíe la atención hacia asuntos sin relevancia. Este método arroja información útil, sin embargo, como sucede con las entrevistas y encuestas, no sustituye a una prueba de usabilidad [9].

Personas

En el diseño de software, una persona es un usuario ficticio que se crea para apoyar el proceso de desarrollo de una determinada aplicación. En este ejercicio imaginario se elabora un perfil detallado del usuario como si de una persona real se tratara, nombre, ocupación, nacionalidad, necesidades y hasta se elige una fotografía para representarlo. En los procesos de diseño, se utilizan personas para preguntarse si estos usuarios, que representan el público objetivo, encontrarán valor a determinada funcionalidad, algo como ¿Qué esperaría Andrés de esta funcionalidad? ¿Andrés utilizaría en su día a día esta función? y luego se contrastan para tomar decisiones de diseño basadas en las “posibles respuestas” que obtendríamos de ese personaje.

En este método, más que generar un perfil de consumidor rígido, estamos creando un arquetipo que representa a un grupo de usuarios y sus necesidades, por eso no es habitual que se creen demasiados tipos de personas, solo las justas que puedan representar a los grupos de usuarios con necesidades y contexto similares. Al respecto, Alan Cooper, quien propuso este método en los años noventa, recalca la importancia de crear un modelo verosímil, ya que si bien son personajes lo que estamos creando en un perfil de persona, éstos deben tener las características reales y evitar estereotipos o modelos ideales que serían una caricatura del usuario [10] para esto desde la investigación previa podemos recolectar información y sintetizarla en estos perfiles. Sin embargo, este enfoque ideal que supone el desarrollo de personas en un proceso riguroso previo, en ocasiones no es posible desde un comienzo, para esto también se pueden crear proto-personas que son bocetos de una persona que

se crean sin tener investigación previa. Al respecto Jeff Gothelf conocido por su método Lean UX recomienda en sesiones colaborativas crear estas personas provisionales para unificar la visión del grupo sobre su público objetivo [11]. Estos modelos o personajes son recursos que nos invitan a ponernos en los zapatos del usuario para analizar sus necesidades y tener una visión compartida entre todo el equipo de trabajo. Luego de crearlos se deben compartir con el equipo y utilizarlos como recursos que aportan valor a las empresas ayudando a reconocer al usuario durante toda la ejecución del proyecto [12].

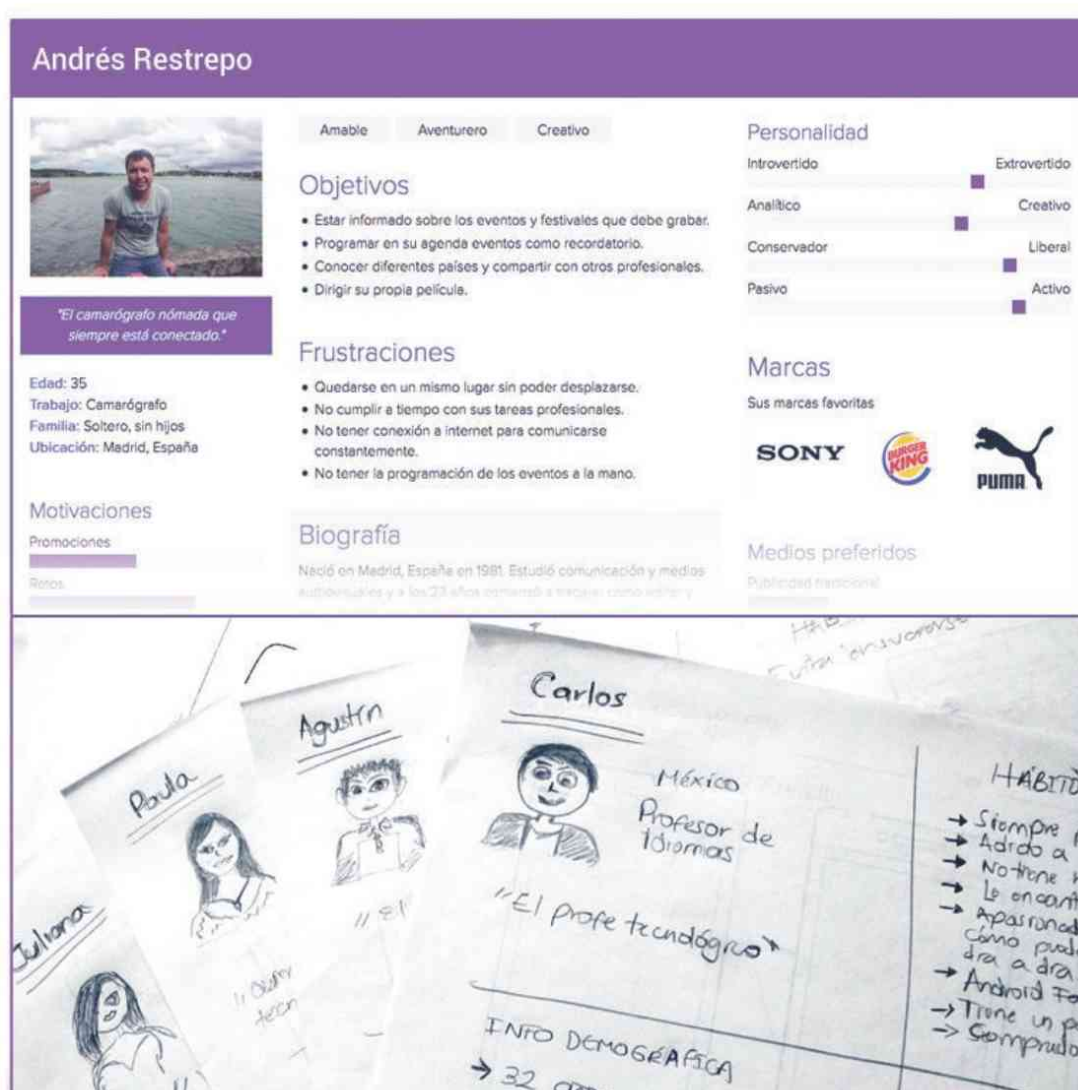


Figura 2-2 Arriba: un ejemplo de persona habitual creada con la herramienta en línea Xtensio. Abajo: Ejercicio de creación de proto-personas en una sesión colaborativa siguiendo el método Lean UX.

Escenarios

Los escenarios se utilizan en conjunto con las personas para realizar narrativas ficticias en situaciones de uso concretas, de esta manera se puede tener un acercamiento a las realidades que enfrentará el usuario y entender sus limitaciones. En un escenario se trabaja por lo general con situaciones concretas como si de un cuento o historia narrada se tratara, donde se describe todo lo que le pasa al usuario y cómo interactúa con el entorno. Es la reconstrucción de una situación o un problema preciso con una solución satisfactoria [13].

Los escenarios descritos en este ejercicio difieren un poco de los casos de uso y también de las historias de usuario, muy utilizadas por los equipos de desarrollo ágiles. A diferencia de estos, los escenarios al igual que el ejercicio de creación de personas buscan una narración más realista sobre los dilemas a los que se enfrenta el usuario para ofrecer una visión global del problema y su solución. Las distracciones o limitaciones habituales que tienen las personas en el uso de un sistema a menudo son ignoradas por los equipos de desarrollo cuando están creando una aplicación, por esta razón es útil complementar el proceso con estos métodos de simulación.

En un escenario se crea una historia que ayuda a recrear una situación, sin embargo, no busca crear una lista de tareas detallada, mas bien se enfoca en tener presente el contexto natural del usuario y sus necesidades [14]. Como se ha reiterado varias veces, en el diseño de aplicaciones móviles el contexto de uso es muy importante ya que no es lo mismo utilizar una aplicación en un lugar cerrado, que en el exterior donde las condiciones del clima, distracciones y ruido pueden afectar la experiencia de un determinado usuario.

Aunque los escenarios se representan en texto simple narrando los diferentes sucesos, también pueden complementarse con métodos más didácticos como la utilización de *storyboards* donde se representa la situación en forma gráfica. En estas narrativas visuales se incluye a la persona como protagonista de la secuencia y su resolución es por lo general positiva para los intereses del usuario final. Para aquellos que se sientan intimidados por tener que dibujar, es importante aclarar que un *storyboard* puede ser elaborado con trazos simples, no es indispensable que sean dibujos muy complejos, lo importante es que se pueda reconocer la secuencia y comunique la narrativa a los miembros del equipo en forma clara y práctica.

Definiendo la Arquitectura de Información

Esta disciplina se consolidó en los años noventa por las investigaciones de Louis Rosenfeld y Peter Morville [15], desde entonces ha sido adoptada por multitud de profesionales que utilizan sus métodos para clasificar el contenido y definir con rigor las maneras adecuadas de estructurar la información y vías de acceso más intuitivas a funcionalidades o contenido de sitios web, aplicaciones móviles y todo tipo de productos digitales.

En este apartado presentamos dos métodos muy utilizados en la definición inicial de esa arquitectura: el *card sorting* que apoya el proceso de elección de los conceptos y etiquetas globales realizando pruebas con usuarios aún cuando no se tiene un prototipo de interfaz y también se describe el método de creación de diagramas muy utilizado para representar en forma de esquemas las relaciones de navegación entre los procesos y pantallas de una aplicación.

Card sorting

Esta técnica, que se traduciría en español como ordenamiento de tarjetas, se emplea por lo general en etapas tempranas de planeación de software o en procesos de rediseño. Es un método bastante simple pero muy útil que consiste en categorizar las funciones globales de una aplicación en un grupo de tarjetas que tendrán escrito un concepto, tarea o característica del sistema. Luego, este grupo de tarjetas (en promedio entre 30 y 100) [16] se le entrega al usuario para que éste las organice de una manera que considere adecuada según su percepción, en este punto debemos aclararle al usuario que no existe una forma correcta o incorrecta de agrupar dichas tarjetas, por lo que puede ordenarlas con total libertad según su criterio particular.

Este método es de mucha ayuda en arquitectura de información para anticiparse a las concordancias entre una muestra diversa de usuarios y cómo estos asocian diferentes conceptos y tareas del sistema, lo que nos puede ayudar a ordenar las taxonomías y elegir los términos más apropiados para las etiquetas de nuestra aplicación, similar a las expectativas de los usuarios finales.

Existen variaciones de esta técnica, en algunos casos se puede utilizar de manera abierta, esto quiere decir que se le entregan las tarjetas al usuario con los términos que debe ordenar y este los organiza creando sus propias categorías

o es posible crear unas categorías fijas donde el usuario debe ubicar las tarjetas relacionándolas [17]. Para facilitar la sesión con el usuario, es importante definir qué método o variación de card sorting se va a utilizar antes de realizar la actividad. Una sesión de este tipo se puede realizar en persona con tarjetas físicas o de manera remota utilizando algún software que ofrezca esta funcionalidad. Si se realiza en persona lo único que necesitamos es papel, lápiz y por supuesto, usuarios con los cuales realizar la actividad.

Si se realiza de manera remota, herramientas como Optimal Workshop simplifican la logística de organización necesaria y permiten que los usuarios accedan a través de un enlace único que se envía por correo electrónico. Esta modalidad está teniendo mucha acogida porque permite realizar la sesión con más usuarios. En cuanto al número de participantes para realizar una sesión de *card sorting*, algunos autores como Jakob Nielsen recomiendan 15 usuarios para observar los datos del ejercicio y compararlos de una manera más precisa [18]. No obstante, si no tenemos alternativa podemos realizar estas sesiones con un número más reducido de personas. Siempre será mejor validarlo con pocos usuarios que sacar conclusiones arbitrarias y subjetivas basándonos en nuestro propio sentido común.



Figura 2-3 Un ejemplo de tarjetas utilizadas para la realización de una sesión de *card sorting* presencial. Este es el método básico pero también se pueden realizar sesiones remotas.

Creación de diagramas

En arquitectura de información un diagrama es una representación gráfica de un proceso y su estructura, pueden mostrar la realización de determinada tarea por parte del usuario o la estructura general de navegación. Los diagramas son útiles para delimitar el alcance de lo que nuestra aplicación va a realizar y las estructuras básicas que tendrá. También se conocen como *blueprints* y son una herramienta importante en la etapa de planeación de un producto digital pues nos ayudan a clarificar muchos aspectos del alcance del proyecto.

Es importante reconocerlos como punto de partida y documentos comunes que sirven como guía para todo el equipo de desarrollo. Los diagramas son útiles para definir la secuencia de navegación del usuario antes de pensar en la apariencia estética e incluso la disposición de los elementos en pantalla. Crear un diagrama con la estructura de pantallas nos ayuda a definir un punto de vista macro del sistema [19] y aunque en muchos casos no incluyan la representación específica de la interfaz, los diseñadores y desarrolladores pueden analizar los retos técnicos que supone cierto tipo de estructuras para pensar en su posterior implementación. Algunos diagramas pueden llegar a ser muy complejos, sin embargo, como estamos creando una aplicación móvil, es recomendable realizar estructuras sin muchos niveles de profundidad por las limitaciones en el contexto de uso y el espacio disponible en pantalla.

Los diagramas se componen de formas básicas para definir su estructura representando pantallas y flechas para indicar relaciones de navegación entre las diferentes secciones o espacios virtuales por donde el usuario se desplazará en la aplicación. Para crear estos diagramas, un recurso muy utilizado es el vocabulario visual de Jesse James Garret bastante práctico y que puede emplearse para definir la Arquitectura de Información y el Diseño de Interacción de un producto digital sea un sitio web o una aplicación, definiendo las rutas de acceso y convenciones generales del sistema (Figura 2-4).

Este vocabulario se encuentra disponible en los principales programas que se utilizan para crear diagramas como Omnigraffle, Axure y Visio. Si aún no tenemos experiencia en la estructuración de sistemas digitales, los consejos de Garret pueden ser de mucha utilidad. También se pueden descargar estas formas básicas desde el sitio web del autor donde describe con gran detalle las convenciones del vocabulario, sus formas básicas y el modo de uso [20].

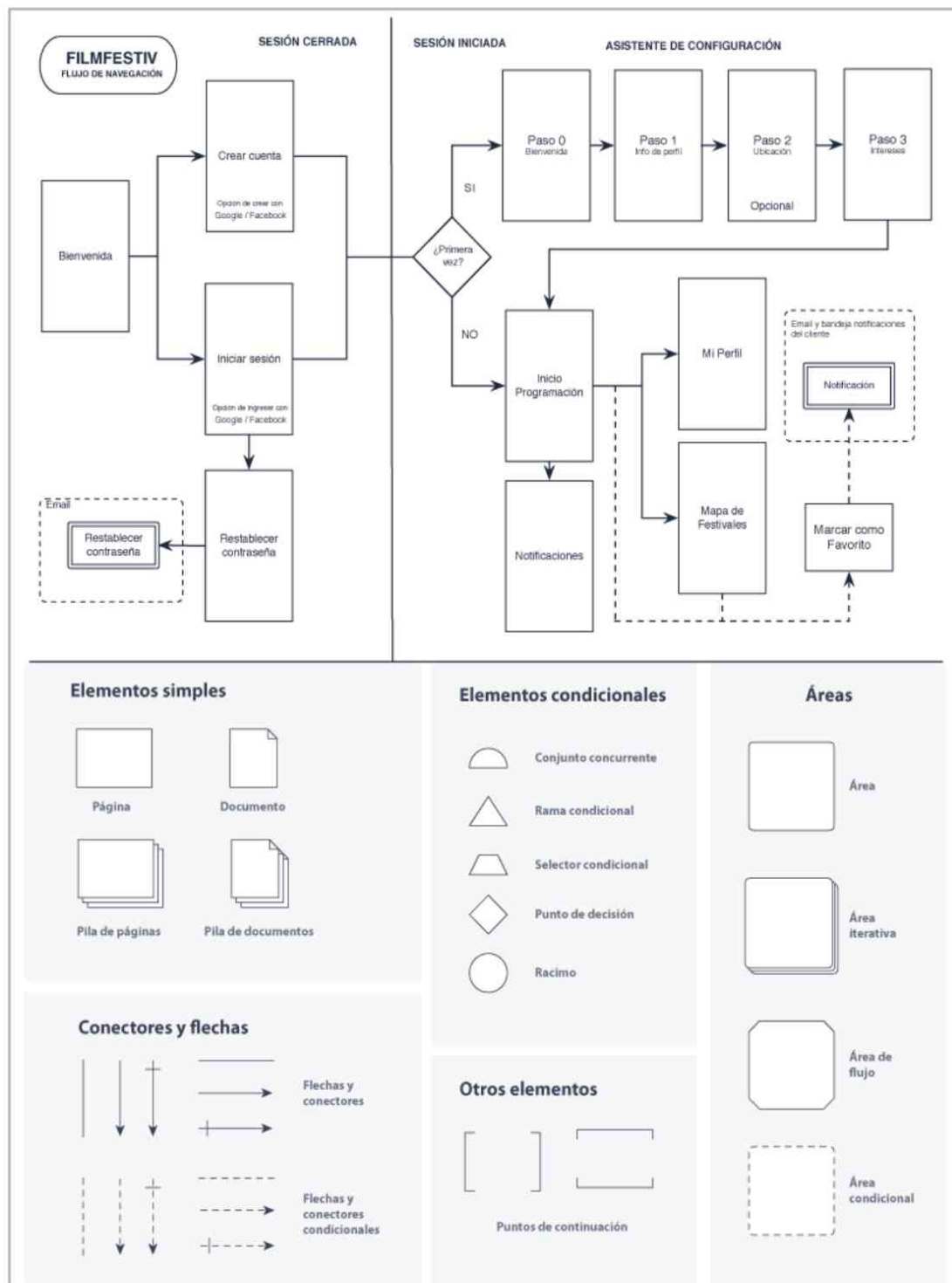


Figura 2-4 Arriba: Fragmento de diagrama de flujo para una aplicación móvil de festivales de cine. Abajo: Resumen de vocabulario visual de Jesse James Garrett. Disponible en <http://www.jjg.net/ia/visvocab/>

Tipos comunes de navegación

Como pudimos ver, los diagramas nos ayudan a representar la estructura de los procesos generales y su conexión a través de pantallas. Esta navegación se realiza por medio de menús, botones específicos, listas o matrices de opciones permitiendo al usuario desplazarse entre pantallas. Para organizar esta navegación es conveniente revisar algunos tipos de estructuras comunes que se utilizan con regularidad en aplicaciones móviles y software en general [21].

Navegación lineal

La navegación lineal podemos encontrarla en procesos paso a paso o en secciones donde navegamos entre elementos como la sucesión de artículos de noticias. Otro ejemplo puede ser el proceso de registro o una compra a través de la aplicación que se realiza paso a paso. Un modelo particular de navegación lineal lo podemos ver en Windows Phone que utiliza un componente nativo llamado *panorama*, donde el usuario se desplaza entre secciones de una aplicación ancha que muestra segmentos de navegación lineal [22].

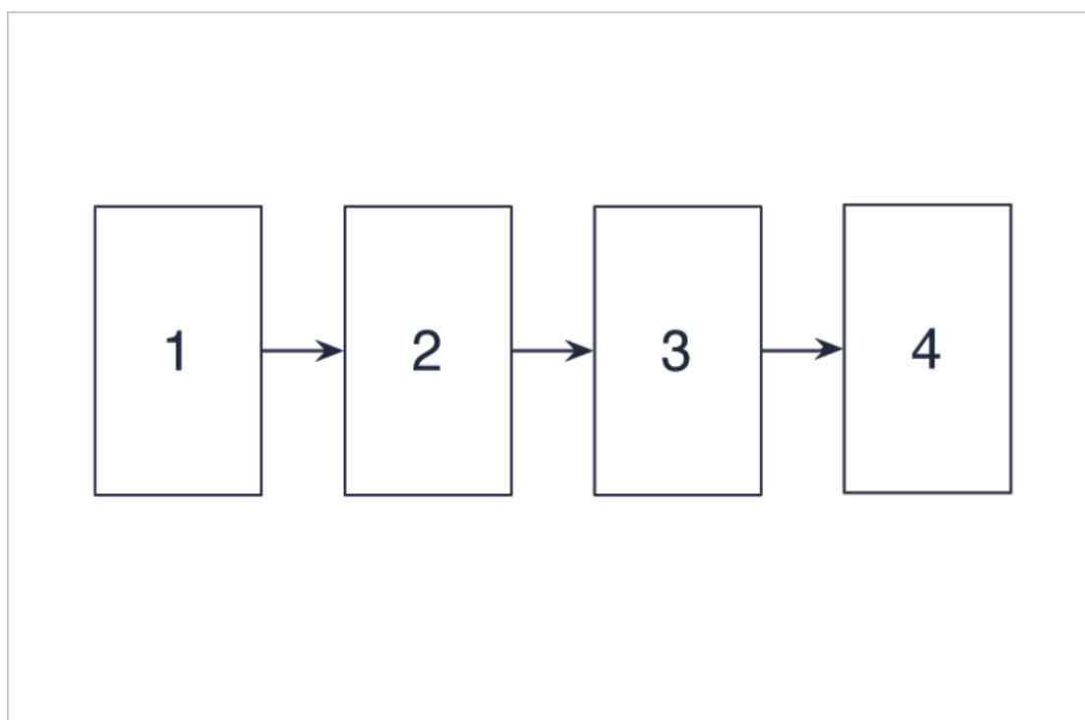


Figura 2-5 La navegación lineal es un recurso que permite navegar hacia adelante y hacia atrás para cumplir un objetivo que requiere de una secuencia específica.

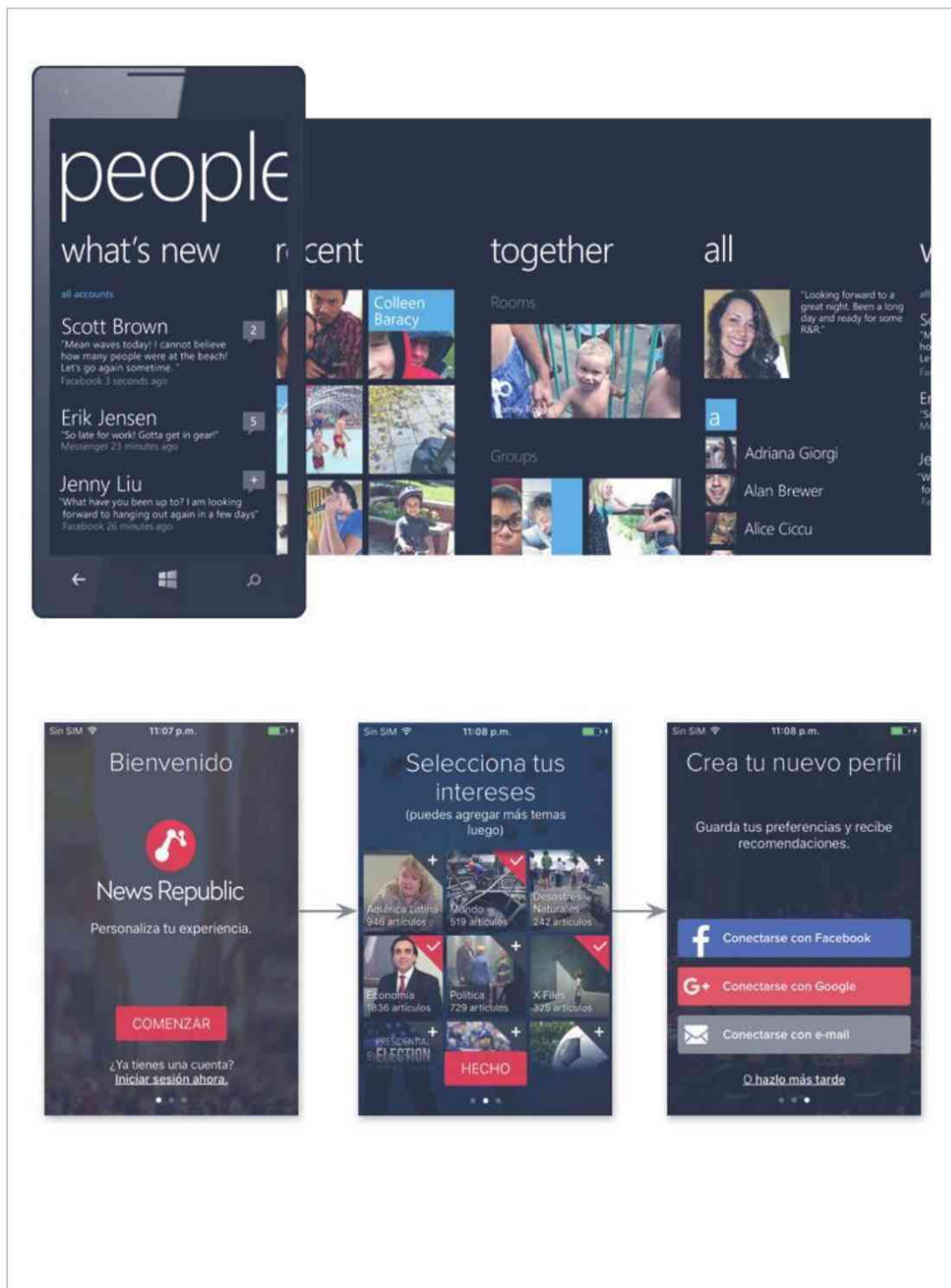


Figura 2-6 Arriba: algunas aplicaciones de Windows Phone utilizan un tipo de navegación lineal conocida como *panorama*. En el ejemplo se puede ver el People hub que utiliza este sistema. Abajo: La aplicación News Republic (iPhone) utiliza una navegación lineal en el proceso de registro y configuración inicial.

Navegación centralizada

En este modelo hay una pantalla inicial que posee un menú global en un listado o matriz de opciones, el usuario elige una opción y navega a una siguiente pantalla. Si el usuario quiere acceder a otra pantalla de segundo nivel debe volver a la pantalla de inicio para seleccionar su nueva ruta. En inglés, este modelo se conoce como *hub and spoke*.

La interacción básica de los sistemas operativos móviles se concentra en este tipo de navegación centralizada. En Android, iOS, Windows Phone y en los demás sistemas emergentes es común este tipo de navegación basada en una pantalla central con iconos que nos permiten abrir las aplicaciones instaladas. Este modelo también puede ser replicado en una aplicación móvil que muestra las diferentes secciones en una matriz principal y permite navegar hacia el interior de la aplicación móvil a pantallas con funcionalidades o contenido específico. La navegación centralizada es un tipo de navegación jerárquica.

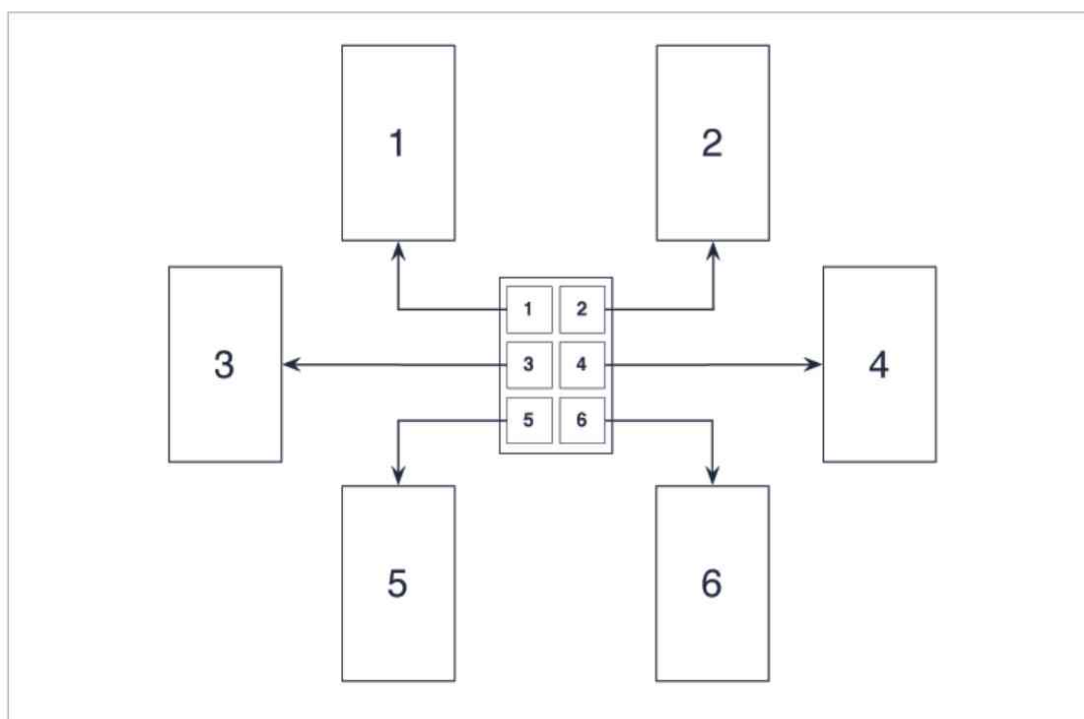


Figura 2-7 La navegación centralizada es un recurso frecuente en la interacción de los sistemas operativos móviles, utilizado en la interfaz general y en el diseño de aplicaciones.

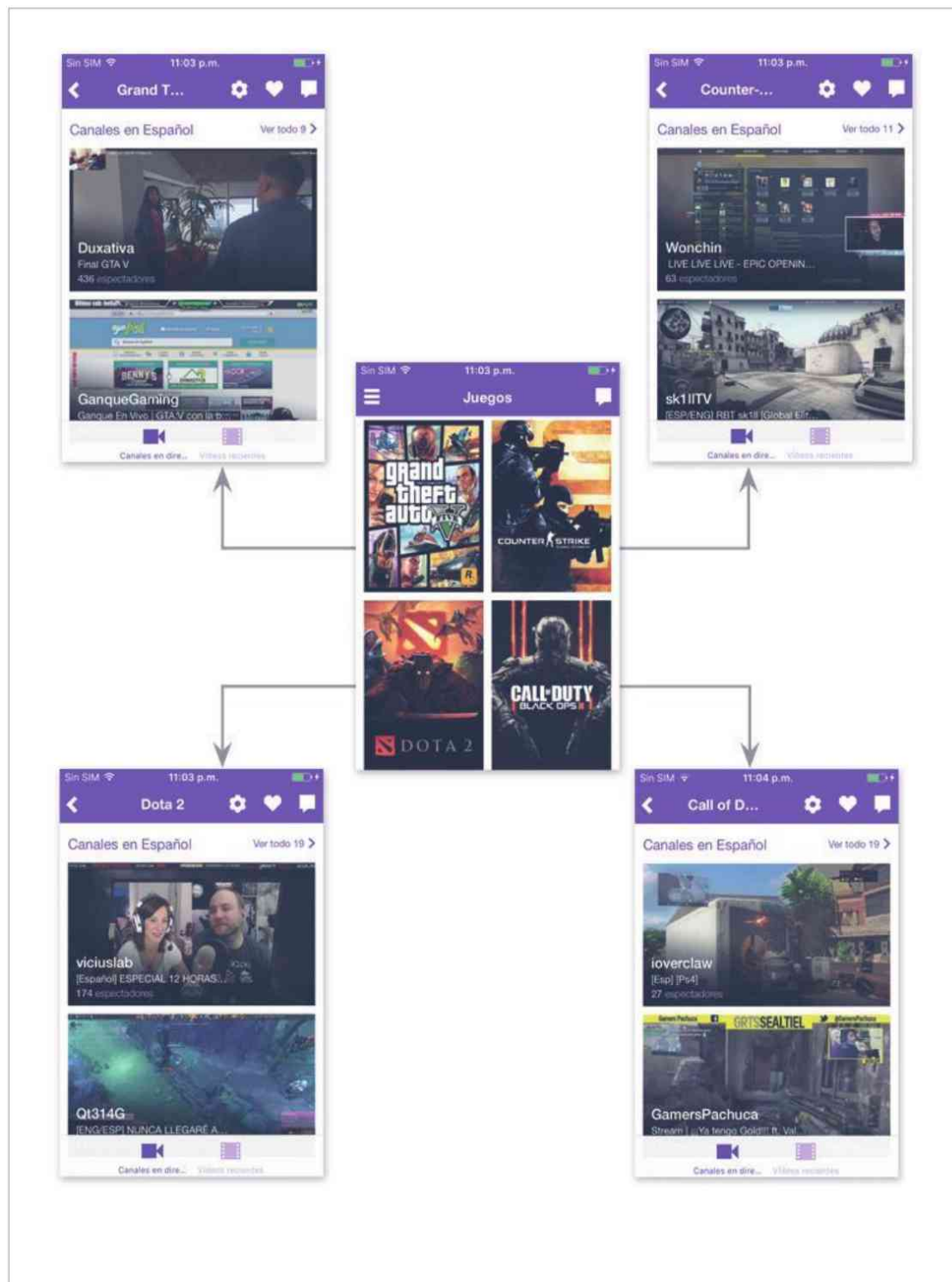


Figura 2-8 La aplicación Twitch (iPhone) utiliza una navegación principal centralizada donde las diferentes secciones se ubican en una matriz de fotos que permiten navegar hacia el interior. Para acceder a otra sección se debe ir atrás al menú principal y elegir la nueva sección deseada.

Navegación interconectada

Este tipo de navegación emplea una pantalla de inicio que lleva a las demás pantallas secundarias y desde estas también se puede acceder a las demás secciones de segundo nivel. La forma más común de aplicar este modelo es a través de los paneles de pestañas muy utilizados en los diferentes sistemas operativos móviles, en especial en Android y iOS.

Sin embargo, las pestañas tienen sus limitaciones en las pantallas de teléfonos que son más reducidas. Una solución alternativa para presentar mayor número de opciones de navegación es a través de los menús laterales que se muestran cuando se pulsa un botón de menú en pantalla, estos menús agrupan las opciones globales de navegación. La ventaja de estos menús es que pueden ubicarse en todas las pantallas facilitando la navegación desde cualquier sección. Este patrón de menú ha sido elegido por Google para integrar su navegación global a través de Material Design.

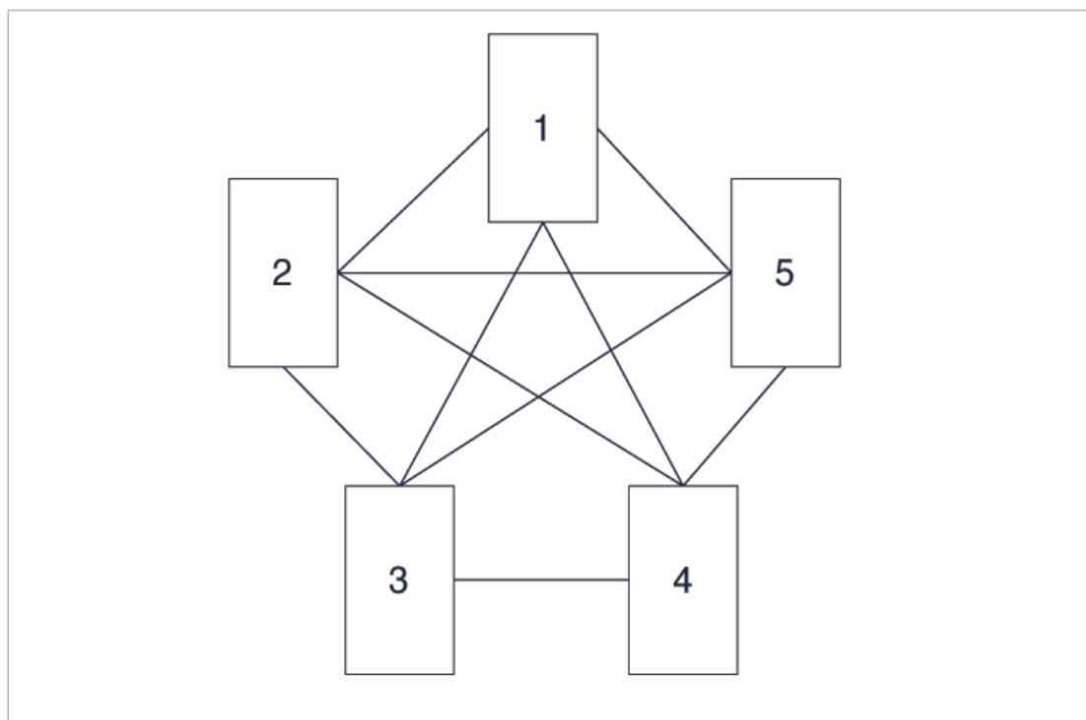


Figura 2-9 La navegación interconectada permite acceder a todas las secciones desde cualquier pantalla de la estructura. Por lo general se utilizan diferentes patrones de navegación, paneles de pestañas o menús en listado para ofrecer una navegación global desde la cual acceder con facilidad a las diferentes secciones.



Figura 2-10 Los paneles de pestañas son componentes muy utilizados en una navegación interconectada. En el ejemplo se puede ver la aplicación Sneaker Crush (iPhone) que posee un panel de pestañas como navegación global que permite acceder a las diferentes secciones principales desde cualquier pantalla en la que nos encontremos.

Navegación plana

En la oferta de aplicaciones móviles también se encuentran muchas enfocadas a una funcionalidad específica que no requiere una división por medio de secciones, este tipo de navegación se concentra en una sola pantalla que utiliza cuadros de diálogo o menús como apoyo para realizar las tareas, entre estas aplicaciones se pueden encontrar utilidades como calculadoras, conversores de divisas, editores de dibujo, entre otras.

Sin embargo, este tipo de navegación plana no es la más utilizada en aplicaciones que poseen secciones definidas donde el usuario busca contenido a través de diferentes niveles de profundidad. En ciertas aplicaciones se puede utilizar una navegación plana combinada con otros tipos de navegación, por ejemplo en un modelo lineal paso a paso en la configuración inicial de una aplicación para luego acceder a la funcionalidad ofrecida. Software de utilidades y herramientas emplean de forma habitual navegaciones planas en su modelo de interacción principal (Figura 2-12).

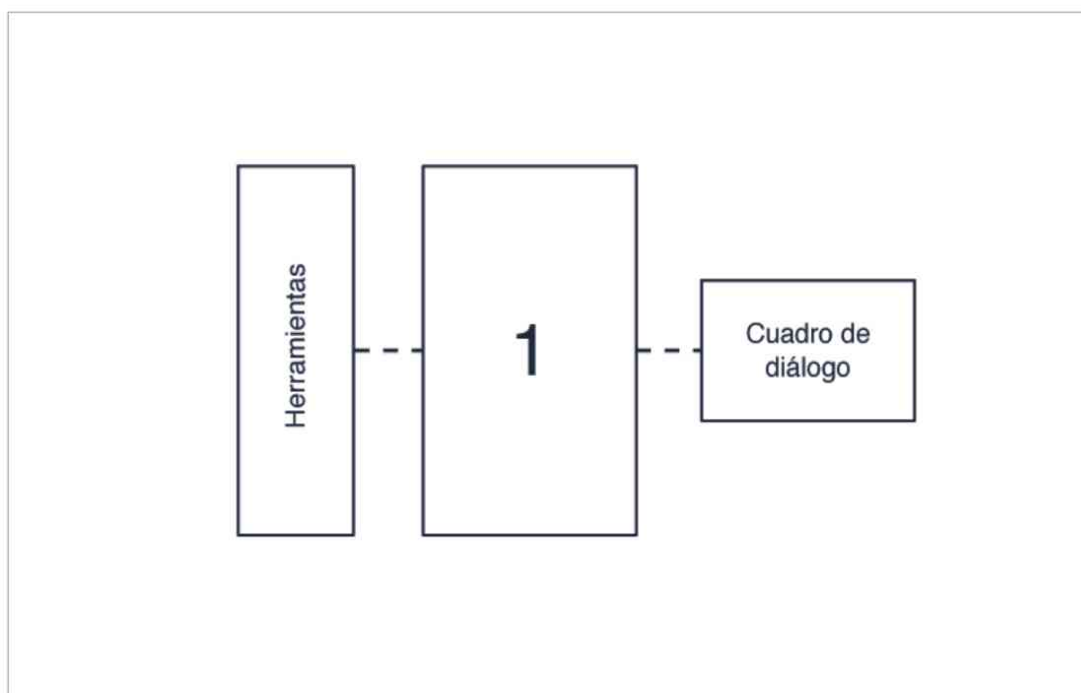


Figura 2-11 La navegación plana mezcla una pantalla principal que no posee una navegación hacia otros niveles de profundidad. Por lo general, se apoya en cuadros de diálogo o menús de herramientas visibles o desplegables que permiten interactuar con la aplicación sin desplazarse hacia otras pantallas.



Figura 2-12 La navegación plana se utiliza en aplicaciones que funcionan como herramientas específicas, ejemplos claros de este tipo de navegación se pueden encontrar en calculadoras, radios y aplicaciones para editar y tomar fotografías. Muchas de estas aplicaciones se apoyan en navegaciones planas.

Profundidad de navegación

Si una aplicación móvil tiene muchas funcionalidades o contenido, puede combinar los diferentes tipos de navegación en modelos híbridos de estructuras que mezclan procesos lineales con otros más libres y jerárquicos donde se pueden segmentar las acciones en pantallas secundarias o incluirlas en los primeros niveles de navegación. Este caso trae consigo un reto y es que si optamos por utilizar pocos niveles de profundidad, podemos reducir los pasos necesarios para llegar al contenido o sección deseados, pero también puede incrementarse la complejidad en el uso de la aplicación, ya que para algunos usuarios supone un dilema y situación molesta el tener que elegir entre demasiadas opciones simultáneas [23].

Por eso, en el diseño de aplicaciones móviles se utilizan estructuras de pocos niveles, pero si nuestro proyecto tiene que utilizar una navegación profunda debemos considerar incluir un sistema de búsqueda que permita llegar al contenido de una manera eficiente, sin obligar al usuario a navegar entre numerosas pantallas en forma innecesaria [24]. Este tipo de estructura que lleva hacia el interior de una aplicación también se conoce como *nested doll* haciendo referencia a las muñecas rusas *matrioska* que se insertan una dentro de la otra en varios niveles de profundidad [25]. Un modelo común en la navegación en dispositivos móviles pero que debe ser utilizado con moderación.

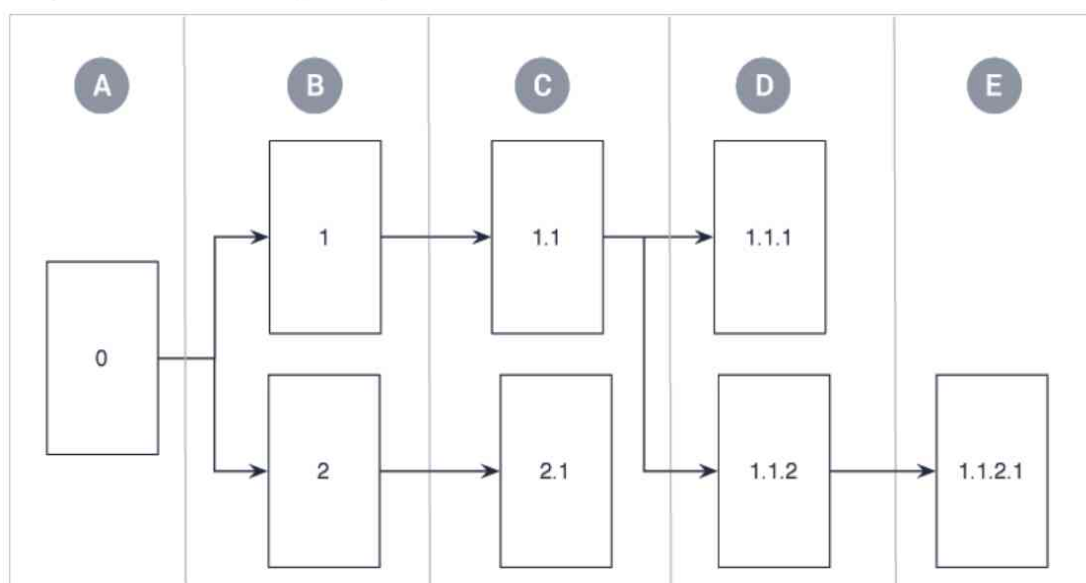


Figura 2-13 La profundidad de navegación es un aspecto importante a controlar, en este diagrama se presenta una navegación de cinco niveles representados por las letras A, B, C, D y E.

Modelos conceptuales y mentales

Una parte esencial que se define en esta primera etapa de análisis y diseño de la Arquitectura de Información se relaciona con identificar la manera más eficiente de presentar y categorizar el contenido y las etiquetas de la aplicación buscando que tenga un sentido lógico en la mente del usuario. De hecho, al definir un vocabulario consistente en las etapas iniciales a través de métodos como el *card sorting* y estructurar la navegación del sistema por medio de diagramas estamos definiendo en una etapa inicial modelos conceptuales, proceso a menudo ignorado por los equipos de desarrollo pero que tiene mucha importancia e impacto en el diseño posterior de la interfaz de usuario [26].

Donald Norman quien popularizó el término de modelo conceptual en el mundo del diseño lo clasifica en un esquema de tres elementos interdependientes: el modelo conceptual del diseño, el modelo mental del usuario y la imagen del sistema [27]. Desde esta perspectiva, el modelo conceptual es elaborado por el diseñador o por el equipo encargado de definir las reglas o lineamientos de uso que va a tener un sistema, explorando conceptos familiares y metáforas generales a emplear. Por su parte, el modelo mental se relaciona con el conocimiento del usuario y sus esquemas de pensamiento (que utiliza para interpretar el modelo conceptual). Por último, la imagen del sistema en el diseño de aplicaciones equivale a la interfaz y su documentación. Es preciso aclarar que el usuario solo tiene contacto con el modelo conceptual a través de la imagen del sistema (Figura 2-14).

Al respecto, un ejemplo clásico de modelo conceptual es el que soporta la metáfora del escritorio en los sistemas operativos, definida en los años setenta por el equipo de desarrollo de Xerox PARC que se inspiró en el uso de conceptos como la representación de archivos, documentos y carpetas, elementos familiares que vienen del mundo físico y que se utilizan como analogías para que el usuario comprenda de forma sencilla estas relaciones y su funcionamiento como un escritorio digital [28].

Desde entonces, es común utilizar metáforas como recurso para representar estos modelos, sin embargo, deben ser empleadas con cautela ya que en algunas situaciones, su interpretación literal puede desorientar al usuario. Como ocurrió durante el rediseño del sistema operativo Windows 95: los usuarios inexpertos durante las sesiones de pruebas no comprendían la nueva funciona-

lidad de subcarpetas, algo ilógico para ellos, quienes se cuestionaban ¿cómo una carpeta podía guardarse dentro de otra carpeta? [29]. Estas confusiones surgen al interpretar de manera literal el modelo propuesto en una interfaz, este ejemplo, aunque antiguo, ilustra una situación habitual en la que nuevos diseños se apoyan en metáforas a las que los usuarios asocian las mismas limitaciones del mundo físico en sus versiones digitales. Para evitar este tipo de ambigüedades es conveniente regular el uso de analogías muy detalladas, solo utilizándolas cuando sean prácticas y faciliten el aprendizaje (Figura 2-14). En ese sentido, aplicaciones sobrecargadas de realismo con una finalidad decorativa o efectista pueden entorpecer las tareas del usuario más allá de facilitarlas.

En un plano más concreto dirigido al diseño de interfaces, los modelos conceptuales se hacen visibles por medio de la imagen del sistema, donde se integran una serie de cualidades que definen su modo de operar a través de cinco atributos que Norman describe como principios fundamentales de interacción en un producto físico o digital (Tabla 2-1) [30].

Tabla 2-1 Principios de interacción de Donald A. Norman (2013)

<i>Affordances</i>	Son las prestaciones o acciones posibles que un diseño permite realizar al usuario ya sea en una interfaz física o digital.
<i>Signifiers</i>	Son los signos o indicios que utiliza el sistema para comunicar su funcionamiento. Se trata de las pistas visuales, físicas o sonoras que ayudan a indicarle al usuario sobre las acciones posibles.
<i>Constraints</i>	Son las limitaciones que se establecen para restringir acciones indeseadas, por lo general para dirigir un proceso o para evitar el error humano.
<i>Mapping</i>	Se refiere a la forma lógica de ubicar los controles de un objeto, la ubicación de los mandos y sus operaciones posibles para activar funciones del sistema de una manera intuitiva.
<i>Feedback</i>	Es la retroalimentación o respuesta que la interfaz ofrece al usuario cuando éste interactúa con el sistema como elemento básico de comunicación.

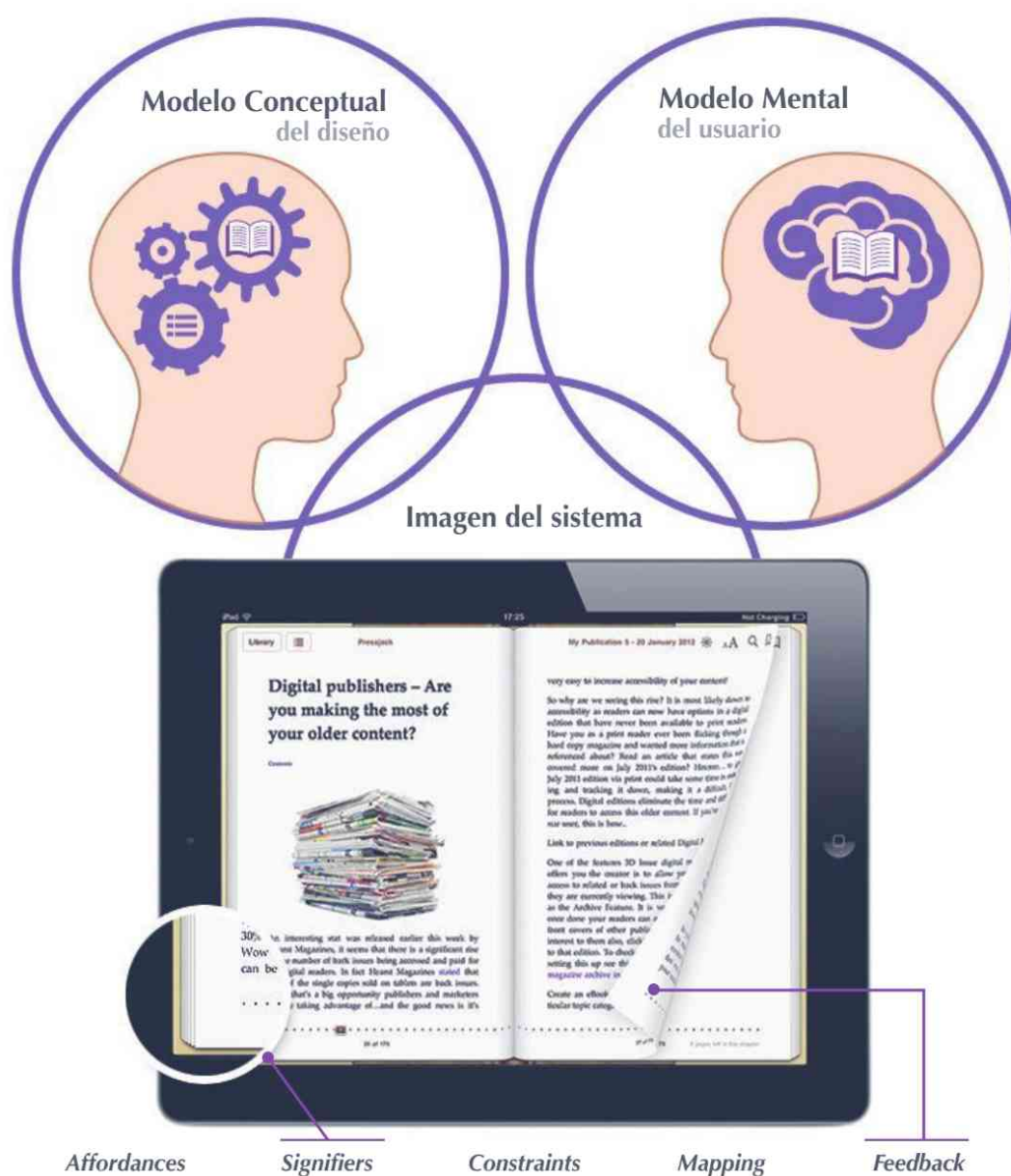


Figura 2-14 El modelo conceptual del diseño y el modelo mental del usuario se comunican a través de la imagen del sistema. Un ejemplo de esta relación se puede ver en la aplicación iBooks que utiliza la metáfora del libro físico para proponer un modelo familiar que facilita el aprendizaje sobre su funcionamiento. En la imagen se pueden observar los principios de interacción de Norman. **Affordances**: en las prestaciones del libro y sus funcionalidades básicas. **Feedback**: en el efecto de pasar la hoja como retroalimentación del sistema. **Signifiers**: en las pistas visuales como las páginas apiladas. **Mapping**: en su interacción natural con las hojas y en la ubicación de los controles en la parte superior. **Constraints**: en la metáfora misma del libro que implica limitaciones en la visualización del contenido en forma secuencial bajo el concepto de página.

REFERENCIAS Y LECTURAS DE INTERÉS

CAPÍTULO 2

-
- [1] D. Travis, «The UX Job Title Generator», User Focus, United Kingdom, 25-ago-2011. [Disponible en línea] <http://interfacesmoviles.com/ref/web21>
 - [2] J. J. Garret, The Elements of User Experience: User-Centered Design for the Web and Beyond, 2.^a ed. California: New Riders, 2011. p. 155.
 - [3] Z. Johnson, «What The Largest Organizations Can Learn From Startups», Forbes - Young Entrepreneur Council, 24-jun-2015. [Disponible en línea] <http://interfacesmoviles.com/ref/web22>
 - [4] «ISO 9241-210:2010 - Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems». [Disponible en línea] <http://interfacesmoviles.com/ref/web23>
 - [5] «About UX», UXPA (User Experience Professionals Association). [Disponible en línea] <http://interfacesmoviles.com/ref/web24>
 - [6] J. Kjeldskov, Mobile Interactions in Context: A Designerly Way Toward Digital Ecology. Morgan & Claypool Publishers, 2014. p. 28.
 - [7] S. Lynden, «6 Mobile Applications to Prevent Distracted Driving Accidents», Automotive Fleet (Online Magazine), ago-2011. [Disponible en línea] <http://interfacesmoviles.com/ref/web25>
 - [8] «Contextual Interview», Usability.gov, 30-jun-2013. [Disponible en línea] <http://interfacesmoviles.com/ref/web26>
 - [9] S. Krug, Don't Make Me Think: A Common Sense Approach To Web Usability, 2.^a ed. Berkeley, California: New Riders, 2006. pp. 132-133.
 - [10] A. Cooper, About Face 3: The Essentials of Interaction Design. Indianapolis: Wiley Publishing, 2007. p. 83.
 - [11] J. Gothelf y J. Seiden, Lean UX. California: O'Reilly Media, 2013. pp. 26-29.
 - [12] S. Mulder y Z. Yaar, The User is Always Right: A Practical Guide to Creating and Using Personas for the Web. New Riders, 2006. p. 28.
 - [13] A. Cooper, About Face 3: The Essentials of Interaction Design. Indianapolis: Wiley Publishing, 2007. pp. 111-113.
 - [14] J. Preece y Y. Rogers, Interaction Design - Beyond Human-Computer Interaction, 3.^a ed. United Kingdom: John Wiley & Sons, 2015. pp. 374-376.
 - [15] P. Morville y L. Rosenfeld, Information Architecture for the World Wide Web, 1.^a ed. Sebastopol, California: O'Reilly Media, 1998.

- [16] W. Hudson, «Card Sorting», The Encyclopedia of Human-Computer Interaction, 2nd Ed. [Disponible en línea] <http://interfacesmoviles.com/ref/web27>
- [17] D. Spencer, Card Sorting: Designing Usable Categories. New York: Rosenfeld Media, 2009. pp. 52-53.
- [18] J. Nielsen, «Card Sorting: How Many Users to Test», Nielsen Norman Group, jul-19-2004. [Disponible en línea] <http://interfacesmoviles.com/ref/web28>
- [19] P. Morville y L. Rosenfeld, Information Architecture for the World Wide Web, 3.^a ed. Sebastopol, California: O'Reilly Media, 2006. p. 296.
- [20] J. J. Garret, «A Visual Vocabulary for Information Architecture», jgg.net, 06-mar-2002. [Disponible en línea] <http://interfacesmoviles.com/ref/web29>
- [21] J. Tidwell, Designing Interfaces: Patterns for Effective Interaction Design, 2.^a ed. Sebastopol, California: O'Reilly Media, 2011. pp. 80-83.
- [22] «Panorama control for Windows Phone 8», Microsoft. [Disponible en línea] <http://interfacesmoviles.com/ref/web30>
- [23] L. Mathis, Designed for Use: Create Usable Interfaces for Applications and the Web, P2.0 ed. United States: The Pragmatic Bookshelf, 2011. pp. 73-74.
- [24] J. Nielsen y R. Budiu, Mobile Usability. [Versión Kindle]. Berkeley: New Riders, 2012. pp. 26-27.
- [25] R. Hinman, «Good Mobile Experiences Unfold & Progressively Reveal their Nature», Rosenfeld Media, 27-jul-2011. [Disponible en línea] <http://interfacesmoviles.com/ref/web31>
- [26] J. Johnson y A. Henderson, Conceptual Models: Core to Good Design. Morgan & Claypool, 2011. pp. 17-20.
- [27] D. A. Norman, The Design of the Everyday Things: Revised and Expanded Edition. New York: Basic Books, 2013. pp. 25-32.
- [28] J. H. Murray, Inventing the Medium: Principles of Interaction Design as a Cultural Practice. Cambridge: MIT Press, 2011. pp. 71-72.
- [29] K. Sullivan, «The Windows 95 User Interface: A Case Study in Usability Engineering», Microsoft Corporation, 1996. [Disponible en línea] <http://interfacesmoviles.com/ref/web32>
- [30] D. A. Norman, The Design of the Everyday Things: Revised and Expanded Edition. New York: Basic Books, 2013. pp. 10-25.

3

DISEÑO Y PROTOTIPADO

En el tercer capítulo se estudian los elementos que componen el proceso de diseño de la interfaz gráfica pasando por el reconocimiento de los gestos táctiles comunes y los métodos de prototipado de alta y baja fidelidad. También se describen los componentes multimedia y elementos interactivos que podemos utilizar en la construcción de una interfaz gráfica con el uso de patrones de diseño comunes en aplicaciones como los paneles de pestañas, listados y matrices de opciones, mapas, menús laterales, entre otros.

De la estructura a lo tangible

El proceso de diseño comienza en el momento mismo en que definimos la Arquitectura de Información, un tema abordado de forma introductoria en el Capítulo 2. Esta etapa previa de diseño estructural, sus conceptos y la conexión entre secciones por medio de la navegación es fundamental para todo tipo de proyectos, pero es invisible para el usuario hasta que no se convierte en la interfaz gráfica como diseño tangible [1].

La estructura de navegación representa una fotografía global de todo el proyecto y desde esta visión del sistema podemos abordar de lo general a lo específico definiendo la interfaz y su relación funcional entre pantallas. Es necesario aclarar que en este capítulo cuando hablamos de diseño, no nos referimos en específico al Diseño Visual como recurso estético o de identidad, aquí el énfasis recae más en el Diseño de Interacción para definir el comportamiento y disposición de elementos en la interfaz [2].

Para estas actividades podemos elaborar diferentes prototipos en papel o de apariencia primitiva como los *wireframes* y prototipos interactivos con apariencia gráfica más elaborada como soluciones a las necesidades del usuario. Estos prototipos son recursos importantes en un proceso de diseño iterativo ya que nos permiten validar conceptos rápidos sin tener que implementar la interfaz gráfica por medio de código de programación.

No obstante, para crear un diseño de interfaz concreto, debemos reconocer los elementos básicos que se pueden incluir para definir su interacción específica a través de gestos táctiles en botones, menús y controles en pantalla para construir la superficie que servirá como medio de comunicación entre el usuario y la aplicación móvil.

Las posibilidades de crear diseños de interfaz son muchas si tenemos en cuenta la diversidad de aplicaciones, sean de contenido, música, navegación, mensajería, multimedia, compras y más. Cada categoría tiene sus particularidades, sin embargo podemos reconocer elementos de diseño comunes que se replican en todas ellas, estos elementos son paradigmas que se utilizan desde hace varias décadas en el diseño de software y aunque han evolucionado hacia nuevos modos de interacción, representan la materia prima de una interfaz que sigue siendo la misma sin importar el proyecto en el que nos encontremos.

Gestos de interacción comunes

La pantalla táctil es el medio de interacción más utilizado en los sistemas operativos móviles que han adoptado esta tecnología basada en gestos. Un paradigma que tiene sus ventajas porque elimina los dispositivos intermedios como el ratón para interactuar con una aplicación, pero también tiene sus limitaciones a la hora de seleccionar elementos de forma precisa. Esta dificultad se vuelve crítica si la interfaz incluye botones muy pequeños o que no tienen límites apreciables. Aunque no hay reglas absolutas, se recomienda utilizar un tamaño básico que puede ir entre 6 y 10mm de la superficie táctil de un objeto en la interfaz para que pueda ser pulsado sin dificultades [3], [4], [5].

Entre los gestos existentes podemos encontrar el conocido toque del inglés *tap*, gesto principal para elegir un elemento, es equivalente al clic en las interfaces WIMP (acrónimo de Windows, Icons, Menus and Pointing Devices) paradigma de interacción tradicional con ventanas, menús y ratones. En las interfaces móviles este modelo no aplica, ya que las pantallas táctiles permiten interactuar de manera más directa (Figura 3-1) añadiendo la posibilidad de combinar diferentes gestos táctiles en una misma acción del sistema, una técnica popularizada por Apple desde 2007 con su sistema *Multi-Touch* en el iPhone.

Otros gestos, aunque no táctiles, son los que utilizan los sensores de los dispositivos: agitar para activar funciones específicas, o como sucede a menudo con los videojuegos, los sensores ayudan a dirigir hacia una dirección concreta sin necesidad de tocar la pantalla, solo con inclinar el dispositivo le indicamos a un objeto virtual que debe moverse hacia la izquierda, derecha, arriba o abajo.

Los gestos que se presentan en la Figura 3-2 y Figura 3-3 son los más comunes en aplicaciones convencionales, sin embargo, se debe tener en cuenta que algunas aplicaciones y juegos pueden utilizar gestos más específicos y personalizados. Identificar estos gestos nos ayudará a definir el comportamiento básico de la interfaz y cómo reacciona hacia los usuarios en forma específica, pero recordemos que pese a la diversidad [6], no conviene agregar maneras extrañas o combinaciones innecesarias para la interacción con el sistema. Al respecto, un gesto básico como el toque puede ser intuitivo, pero su combinación con otros gestos debe ser aprendida por parte del usuario por lo que, si no es vital para la aplicación, debemos simplificar las tareas y utilizar los gestos más comunes que impliquen poco esfuerzo de aprendizaje para al usuario.

En este proceso, podemos identificar diversas maneras como el usuario interactúa con una aplicación móvil en tres modos principales: 1) Interacción directa de la mano en pantallas táctiles con teclados virtuales. 2) Interacción con un lápiz digital. 3) Interacción tradicional con teclados físicos (Figura 3-1). Este modelo estándar durante décadas en los dispositivos móviles, se encuentra en declive, solo Android y BlackBerry ofrecen algunos modelos que soportan este tipo de interacción no disponible en los sistemas iOS y Windows Phone.

Otro método especial para ingreso de datos e interacción con dispositivos móviles consiste en el lenguaje natural como un tipo de “gesto” utilizado para indicar con la voz, acciones concretas al teléfono y sus aplicaciones.

Esta técnica pese a su prematura incorporación actual, ha demostrado su potencial en el uso diario de asistentes como Google Now en Android, Siri en iOS o Cortana en Windows, en especial para situaciones específicas como ocurre cuando se maneja un automóvil o para personas con limitaciones físicas que tienen dificultades para interactuar con pantallas táctiles ya sea porque no pueden utilizar sus manos o por discapacidades visuales diversas, de esta forma este método en creciente auge ayuda a mejorar la accesibilidad en el uso de teléfonos inteligentes.



Figura 3-1 La interacción con pantallas táctiles se puede realizar en tres modos comunes: tocando la pantalla, utilizando un lápiz óptico (*stylus*) o por medio de teclados físicos. Este último método, es menos común en la actualidad.

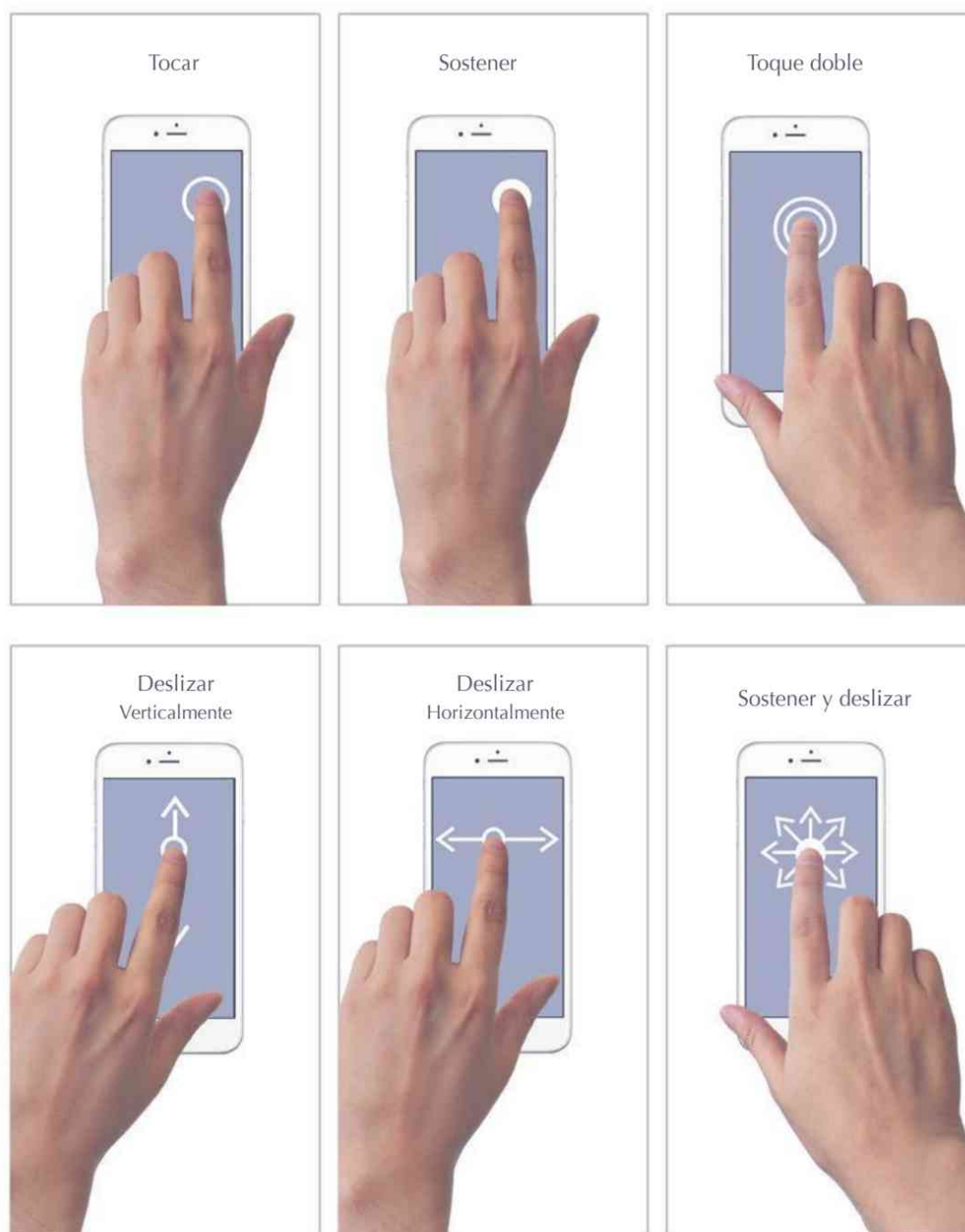


Figura 3-2 Arriba: gestos comunes en la interacción con pantallas táctiles. Encontramos el toque básico, el toque sostenido que equivale al doble clic en las interfaces WIMP y el toque doble que se utiliza como atajo u opción secundaria para activar una acción de forma concreta. **Abajo:** se encuentran los gestos generales de desplazamiento o *scroll* para revelar contenido o arrastrar objetos virtuales en pantalla.

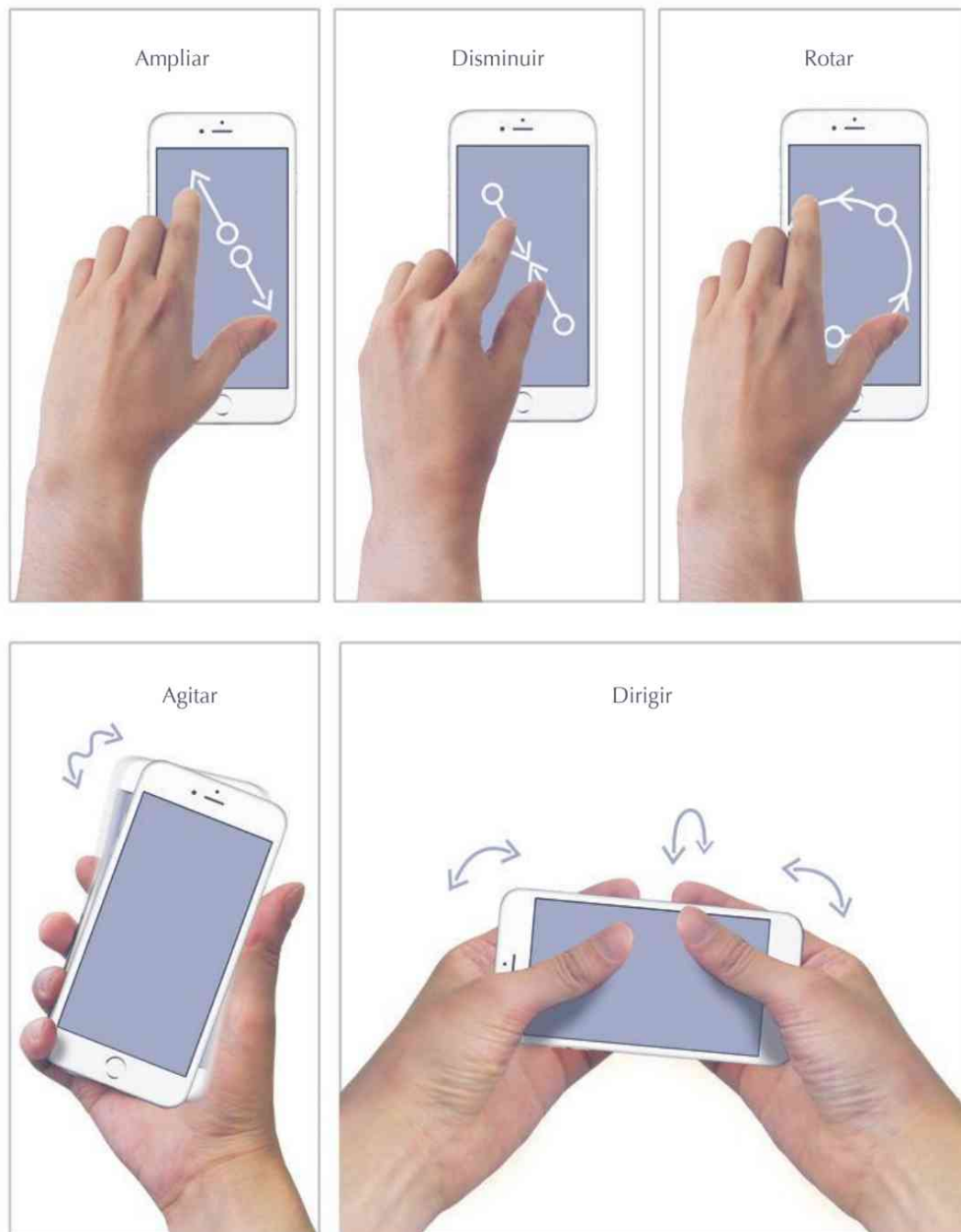


Figura 3-3 Arriba: gestos *Multi-Touch* más utilizados, estos gestos tan conocidos en la actualidad se utilizan para rotar, ampliar, disminuir el tamaño de textos, imágenes o visores de mapas en efectos de *zoom in* y *zoom out*. Abajo: gestos que se utilizan sin tocar la pantalla táctil. En estos se utilizan los sensores internos del móvil para interactuar con aplicaciones, gestos muy empleados en videojuegos.

Creación de prototipos

El prototipo es la unidad básica de prueba de un sistema o producto. Es una elaboración inacabada que posee cualidades cercanas al producto final que permiten validar su funcionamiento en etapas iniciales y de esta manera tomar decisiones pertinentes antes de implementar un diseño. Estos se clasifican en dos tipos básicos: prototipos de baja fidelidad y prototipos de alta fidelidad.

Prototipos de baja fidelidad

Son los primeros bocetos en el diseño de aplicaciones, no presentan decorados o estética particular, se centran en la funcionalidad y distribución adecuada de los elementos que componen la interfaz. Se pueden realizar en papel o con software de diagramación como Omnigraffle o Axure.

Tabla 3-1 Clasificación básica de prototipos de baja fidelidad

Sketches	Son bocetos y dibujos iniciales que presentan la versión más simple y básica de la interfaz. Por lo general se elaboran en papel y lápiz. Su apariencia es primitiva, pero su propósito es claro cuando se trata de validar algunas soluciones iniciales. Son borradores y se utilizan en sesiones colaborativas donde todo el equipo puede aportar.
Wireframes	Son esqueletos, estructuras de líneas como representaciones de la estructura de los elementos en pantalla, son más elaborados y se pueden realizar con software como Axure, Omnigraffle o incluso Keynote o PowerPoint, pero al igual que un boceto carecen de estética. Se podría decir que son los planos del diseño ya que incluyen la base de lo que será la interfaz gráfica.
Storyboards	Representan secuencias entre diferentes pantallas, animaciones y uso combinado de sonidos, texto, imagen, video. Se asemejan a la narrativa de los <i>storyboards</i> del mundo audiovisual pero adaptada al diseño de interacción. Son importantes en sistemas que combinan medios diversos y transiciones muy específicas entre pantallas y sus diferentes estados.

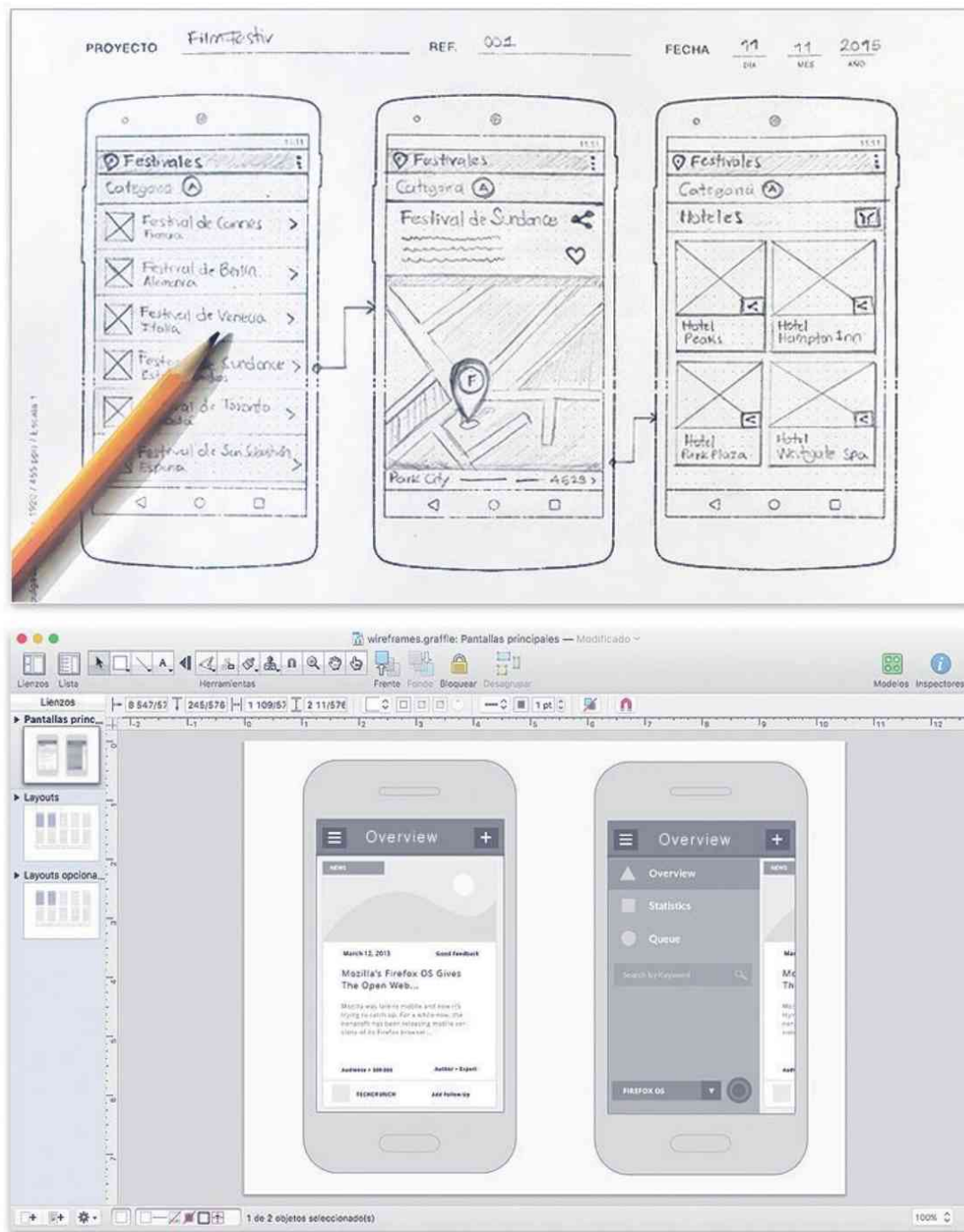


Figura 3-4 Arriba: Bocetos que muestran la estructura de interfaz de una aplicación móvil en sus diferentes pantallas. Abajo: Wireframes de referencia para construir una aplicación móvil, como se puede ver en el ejemplo no poseen decorados ni colores, su objetivo principal es comunicar lo esencial.

Prototipos de alta fidelidad

Son diseños maduros que se asemejan bastante al producto final terminado. Su apariencia es muy trabajada y se incluye el valor estético (color, texturas, decorados, efectos). Este tipo de prototipos se elaboran por lo general con software de edición de imágenes como Photoshop, Illustrator o Sketch.

Tabla 3-2 Clasificación básica de prototipos de alta fidelidad

Mockups	En español se traduciría como maquetas y son prototipos de representación gráfica de media-alta fidelidad incluyendo colores y texturas. En un prototipo como estos se puede visualizar el diseño de una manera clara y muy cercana al producto terminado. Es habitual realizarlos como diseños estáticos de pantallas individuales.
Mockups interactivos	Las primeras versiones de un prototipo se realizan en papel y se hacen algunas pruebas que pueden ser utilizadas para medir su efectividad, pero algunas veces no es suficiente un prototipo estático en papel. Por esto, cuando se requiere un detalle más real de la navegación se pueden realizar versiones que incluyen una interacción básica entre pantallas con hipervínculos. Con estos prototipos, el usuario puede interactuar con nuestro sistema y hacerse una idea del funcionamiento general de la aplicación.

Es necesario aclarar que aunque los prototipos interactivos se suelen crear a partir de un diseño de alta fidelidad, también se pueden realizar con prototipos en papel. Para crearlos, está creciendo el uso de aplicaciones que permiten tomar fotografías de los bocetos y agregar interacciones básicas entre pantallas, entre estas herramientas podemos encontrar Marvel o POP. También se pueden realizar a través de software de diagramación como Omnigraffle, Axure o servicios como InVision o UXpin que permiten organizar una secuencia de pantallas para simular la interacción básica con la aplicación. Este método es eficiente porque permite evaluar una aplicación móvil simulando un producto terminado cuando, en realidad, el equipo de desarrollo no ha tenido que escribir una sola línea de código de programación para validar la efectividad de su interfaz con usuarios reales.

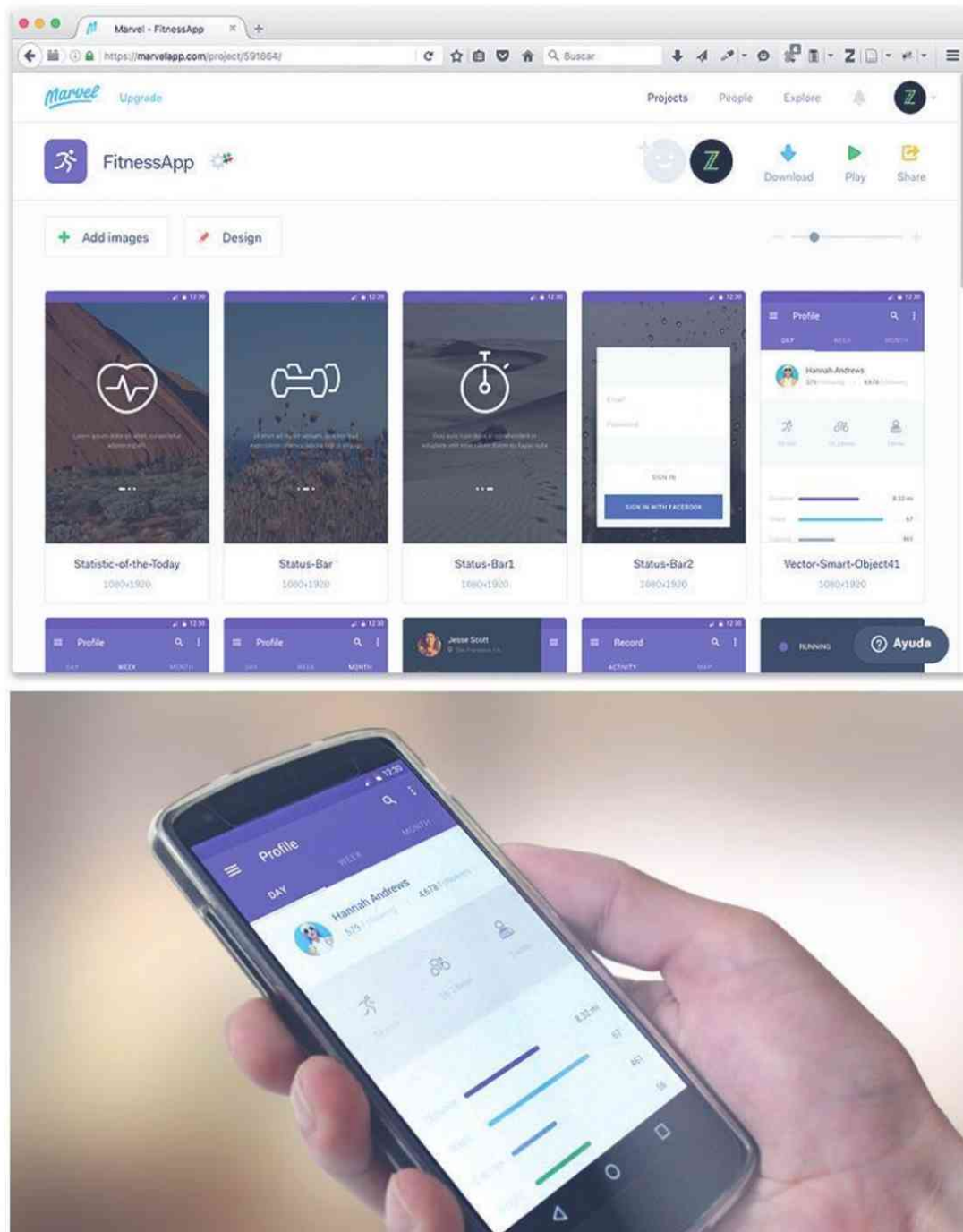


Figura 3-5 Ejemplo de *mockups* de una aplicación sin implementar en la herramienta de prototipado Marvel, este tipo de imágenes presentan un alto nivel de detalle como si se tratara de una aplicación implementada pero en realidad se trata de un concepto. En una etapa inicial es innecesario realizar prototipos tan detallados, pero a medida que avanza el proyecto y se validan con pruebas es necesario tenerlos.

Elementos básicos de la interfaz

En líneas anteriores describimos la importancia de los prototipos y su clasificación en determinadas tipologías según su nivel de detalle y finalidad. Sin embargo, para diseñar una aplicación no solo hacen falta ideas y trazos al azar, no se trata de un ejercicio de dibujo, sino de construir un artefacto que va a resolver una necesidad del usuario y por eso deben representarse de una manera eficaz; así como ocurre en el diseño industrial y otras disciplinas, en el diseño de interfaces también debemos identificar la “materia prima” que tenemos disponible reconociendo sus elementos básicos para luego crear diseños consistentes, similar a como aprendemos la gramática específica de un lenguaje para poder escribir frases o textos con sentido.

A continuación se describen los elementos básicos para construir una interfaz gráfica como paradigmas que se utilizan de forma recurrente en el diseño de aplicaciones. Se presentan con ejemplos en baja fidelidad y alta fidelidad.

Tabla 3-3 Elementos comunes en el diseño de la interfaz

Pantallas	Son la unidad básica de interacción, algunos autores se pueden referir a estas como páginas o ventanas pero su función básica y elemental es la misma. Interactuamos con una interfaz por medio secuencias entre pantallas.
Contenedores	Son elementos que permiten modularizar la interfaz a través de paneles específicos. Definen las estructuras básicas y ayudan a segmentar de manera lógica las pantallas.
Elementos multimedia	En este grupo podemos encontrar los principales insumos para el contenido presentado en pantalla: imágenes, iconos, textos, audio y video.
Elementos interactivos	También conocidos como <i>widgets</i> de interfaz, son los controles que le aportan interactividad a la aplicación y permiten crear componentes como formularios y controles específicos.

Pantallas

La pantalla es la base principal de la interfaz y se comporta como un lienzo interactivo. Es la superficie que permite diseñar la interfaz y que representa los elementos visuales necesarios. En las estructuras de navegación nos acercamos a las secuencias entre determinadas pantallas para proponer al usuario un tipo de interacción. En aplicaciones móviles también se debe tener en cuenta la orientación del dispositivo que se puede utilizar en dos modos: horizontal (*landscape*) y vertical (*portrait*). Otro punto crítico en el diseño para pantallas móviles es su tamaño variable, realidad que implica retos importantes, sobre todo en Android que tiene una diversidad de fabricantes que hacen complejo definir una pantalla estándar, situación más controlada en iOS y Windows que poseen menor diversidad de terminales.

En un primer momento se pueden elaborar prototipos en papel para realizar pruebas con usuarios; en estos primeros bocetos también conviene aproximarnos a la escala real de la pantalla para representar la aplicación de una forma más realista. En el Capítulo 4 se especifican los tamaños de pantallas más comunes, medidas que debemos considerar por las limitaciones físicas impuestas por dichos tamaños y la manera en la que los usuarios pueden acceder a ciertos elementos sosteniendo el teléfono. En este contexto retorna con importancia la conocida Ley de Fitts que establece que el tiempo necesario para alcanzar un objetivo depende de la distancia y el tamaño de dicho objetivo [7], por eso, como se mencionó en líneas anteriores sobre los gestos comunes, es recomendable utilizar elementos y botones de mayor tamaño en una aplicación móvil donde el “dispositivo” de entrada principal es el dedo (por naturaleza más impreciso que el ratón de los ordenadores).

Estas restricciones condicionan el diseño de una interfaz móvil, y aunque en general se pensaba que el usuario sostenía el teléfono con una sola mano interactuando en la pantalla con el dedo pulgar [8], recientes estudios de Steven Hoober demuestran que los usuarios combinan varias formas de sostener el teléfono cambiando de una u otra según sus necesidades, lo que indica que el usuario interactúa con el teléfono de formas diversas (Figura 3-6), algo que conviene tener en cuenta en el diseño de la interfaz. Estas variaciones determinan qué aspectos debemos priorizar, por ejemplo si nuestra aplicación se presenta en modo horizontal o vertical debemos tener presentes las limitaciones físicas impuestas por cada orientación de pantalla [9].

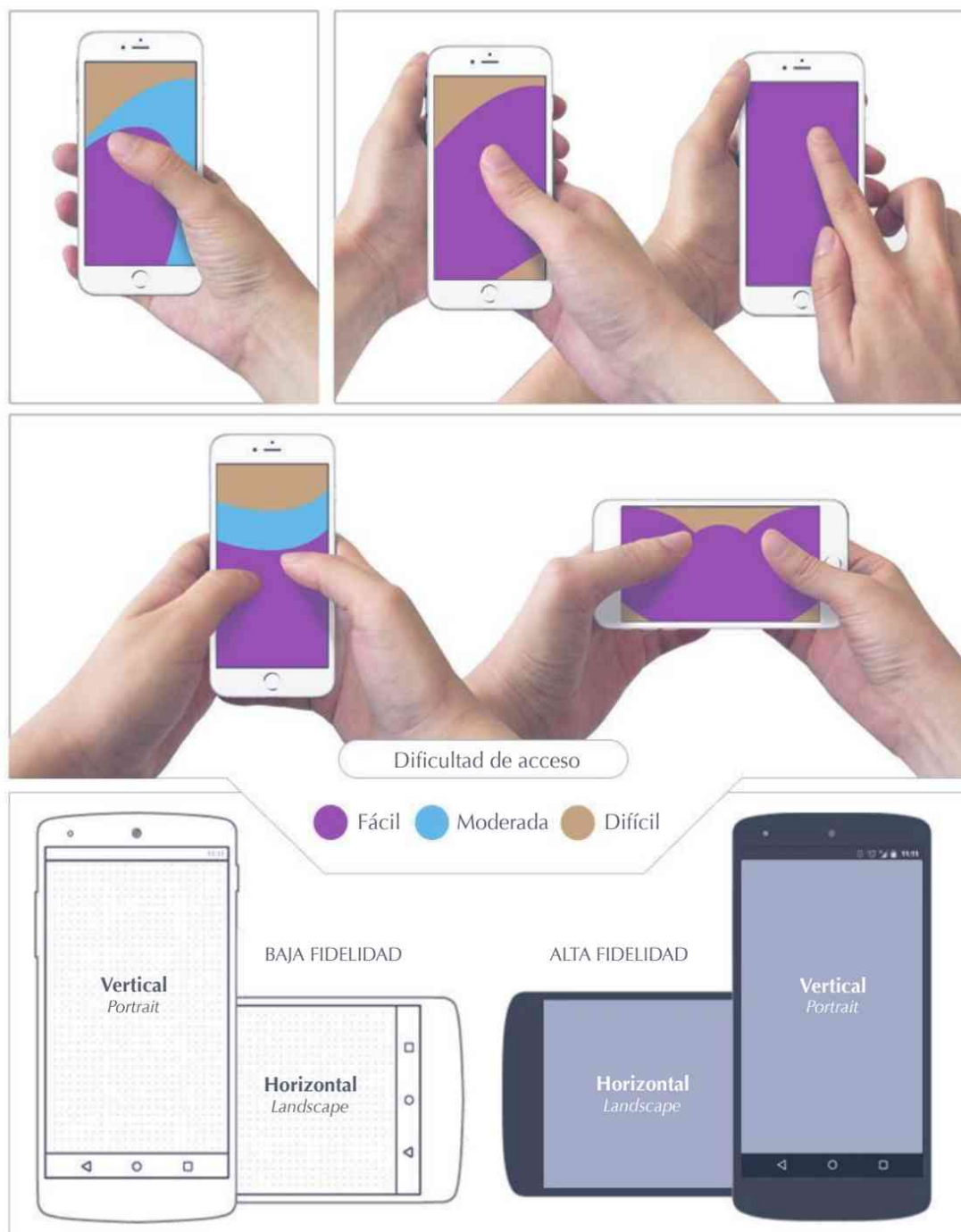


Figura 3-6 Arriba: Formas comunes de sostener el teléfono, según investigaciones de Steven Hoober en 2013. Dependiendo de la manera como se sostiene el móvil, el acceso a determinadas áreas de la pantalla será fácil, moderado o difícil. **Abajo:** Las pantallas poseen dos modos de orientación básicos: vertical (*portrait*) que es el más común en teléfonos y horizontal (*landscape*) la orientación estándar en tabletas

Contenedores

En inglés, la disposición de un diseño se conoce como *layout*, término muy utilizado en arquitectura y en la industria editorial donde se utilizan retículas de estructuras para ubicar en forma organizada el contenido de libros, revistas y carteles. De manera similar una pantalla digital también utiliza un *layout* que puede estar organizado y estructurada por medio de un contenedor principal y otros secundarios que representan paneles con textos, imágenes, video y controles interactivos.

El contenedor es el elemento primario que permite segmentar los paneles de la interfaz. Puede tener límites apreciables con fondos y líneas, o puede ser invisible (esto último no invalida su función organizativa). Dependiendo de las plataformas con las que trabajemos estos contenedores serán llamados de diferentes maneras, por ejemplo en Android se tienen cinco tipos de *layout* que en esencia son contenedores, en Windows los StackPanel cumplen esta misma función. Aunque el término es utilizado en la etapa de implementación de aplicaciones, también es un concepto que conviene tener en cuenta en el momento de realizar prototipos ya que nos ayudan a estructurar la interfaz en secciones o componentes diferenciados.

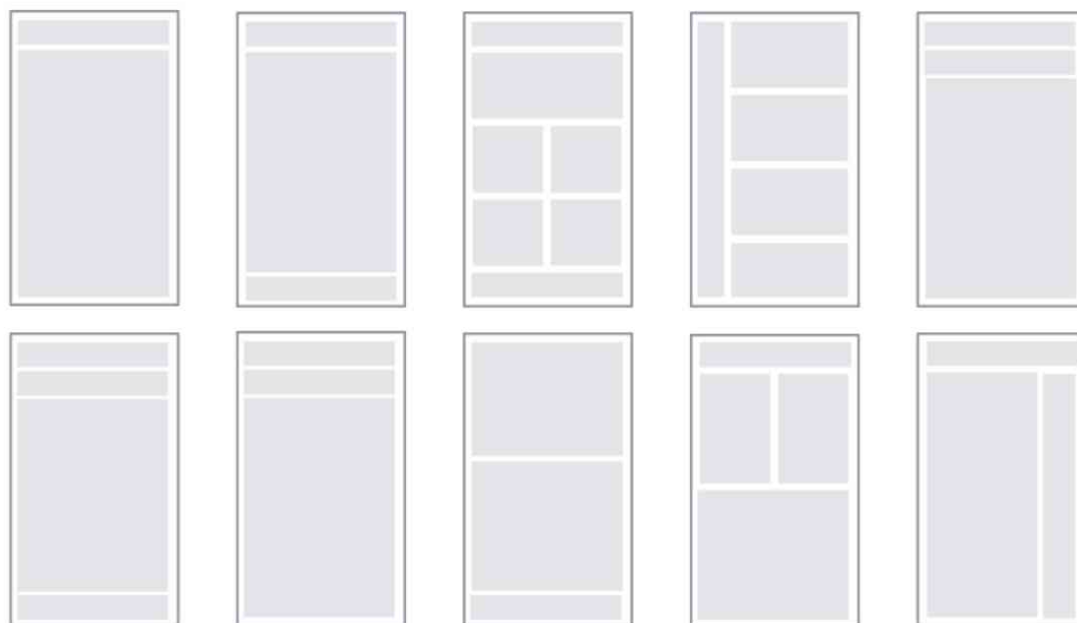


Figura 3-7 Ejemplos de contenedores en diferentes distribuciones comunes en aplicaciones móviles. Estos elementos son la base modular de las interfaces y sirven para definir paneles para organizar el contenido.

Elementos multimedia

Los elementos multimedia se han utilizado en el diseño de software desde los años setenta con la creación de las primeras interfaces gráficas, antecedentes directos del uso de estos medios para una comunicación efectiva con el usuario, ya sea con fines informativos, didácticos o de entretenimiento.

En esencia, los elementos multimedia son aquellos que le otorgan valor semántico a los contenidos que se presentan en pantalla, con esto nos referimos al significado de los mensajes transmitidos por medio de imágenes, texto, audio o video. Estos elementos son el insumo más básico y transversal de la interfaz y se utilizan de forma habitual en el diseño de todo tipo de aplicaciones móviles y de escritorio. Por lo general, no poseen capacidades interactivas pero es frecuente combinarlos en otros elementos como botones, campos de texto y controles de interfaz que si poseen interactividad.

En la actualidad, dadas las características avanzadas de plataformas móviles como Android, iOS y Windows Phone se pueden utilizar elementos multimedia como si se tratara de una aplicación de escritorio, sin las limitaciones de los teléfonos tradicionales que tenían prestaciones muy pobres en el manejo de elementos como video e imágenes.

Tabla 3-4 Elementos multimedia en aplicaciones

Imágenes	Fotografía e ilustraciones como parte del contenido o elementos decorativos.
Iconos	Como sistemas de señalización de la interfaz en botones y otros espacios.
Texto	Elemento sustancial para organizar el contenido escrito y la señalización de la interfaz.
Audio	Puede presentarse como efectos de sonido, música o narraciones.
Video	Recurso muy utilizado en aplicaciones multimedia para presentar videos, películas y material audiovisual.

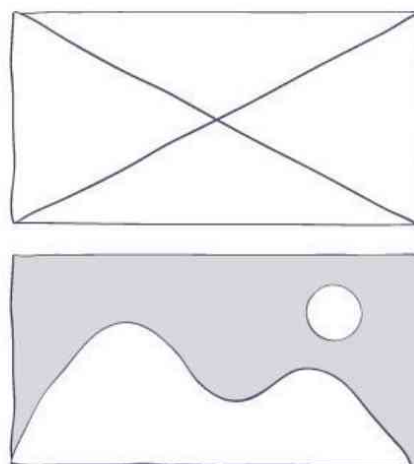
Imágenes

Las imágenes hacen parte de la representación virtual de lo que ve el usuario y junto con el texto son el principal insumo comunicativo del sistema. En esta categorización podemos encontrar fotografías, ilustraciones e imágenes particulares que pueden utilizarse como fondos de los contenedores y diferentes controles interactivos.

Como su nombre indica, la interfaz gráfica de usuario está compuesta de imágenes que representan en pantalla diversos elementos gráficos. Pero aquí el énfasis recae en la fotografía y la ilustración como principales componentes que acompañan a los demás medios. Las aplicaciones de contenido, noticias y multimedia en general suelen utilizar imágenes junto con el texto, ya que su riqueza visual ayuda a captar la atención y comunicar de manera eficaz ciertos mensajes al usuario.

En aplicaciones móviles las imágenes se utilizan con frecuencia para acompañar artículos de contenido o galerías de fotos donde se presentan miniaturas en modo de matriz de opciones. Otro ejemplo se da en aplicaciones de música que suelen emplear imágenes para mostrar las carátulas de CD en reproducción.

BAJA FIDELIDAD



ALTA FIDELIDAD

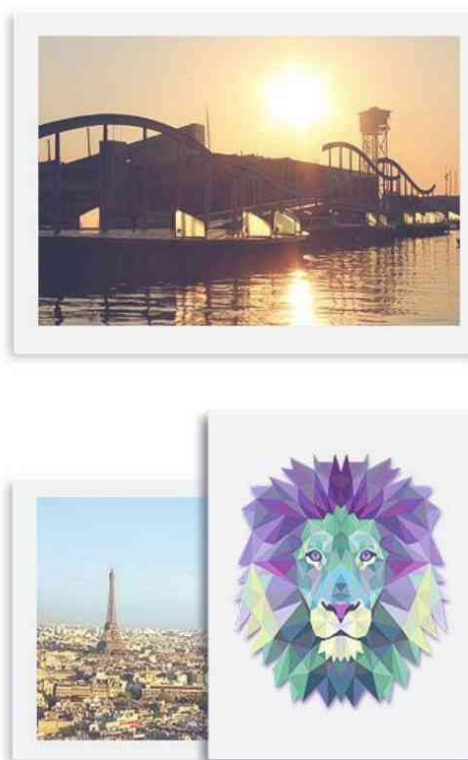


Figura 3-8 Las fotografías e ilustraciones se utilizan como apoyo o recurso principal en galerías en aplicaciones móviles de todo tipo

Iconos

Aunque en esencia también se trata de imágenes, los iconos tienen unas características propias como elementos de comunicación. Son gráficos útiles para utilizar en pestañas, menús de navegación y botones. Cada sistema operativo posee sus tamaños oficiales para representar iconos en pantalla, lo que permite definirlos de forma coherente en proporción a otros iconos de aplicaciones.

Tabla 3-5 Tipos comunes de iconos en aplicaciones móviles

Iconos de aplicación	Conocidos en inglés como <i>launchers</i> , o iconos de lanzamiento, son el identificador más importante y el punto de entrada a nuestra aplicación. Su diseño es muy trabajado y de estilo singular.
Iconos de interfaz	Son iconos estándar que se utilizan en la interfaz de la aplicación, son muy comunes en pestañas, botones y menús. Otro uso habitual es como emoticonos y <i>badges</i> , un tipo de elemento o insignia que incluye información adicional.
Iconos de notificación	Son iconos que acompañan los mensajes de notificación del sistema, fuera de la aplicación. En iOS y Windows se presenta el icono de aplicación en un tamaño reducido mientras que en Android es habitual utilizar iconos personalizados.

Un icono eficaz se visualiza de manera correcta en diferentes escalas de tamaño, tanto en pantallas grandes como pequeñas, esta cualidad es importante en iconos de interfaz y notificación donde utilizar diseños muy realistas y complejos puede ser problemático ya que en tamaños pequeños son más difíciles de reconocer que un icono de trazos simples [10], [11].

Otro aspecto a menudo ignorado es reconocer el público y ubicación geográfica donde estará disponible nuestra aplicación para identificar si a nivel cultural los iconos que utilizamos son familiares, y de esta forma evitar connotaciones erróneas o negativas. Situación muy ligada a que el significado de un icono debe ser aprendido, sobreexponer una interfaz de iconos no convencionales sin ofrecer textos descriptivos puede generar confusión en el usuario [12].

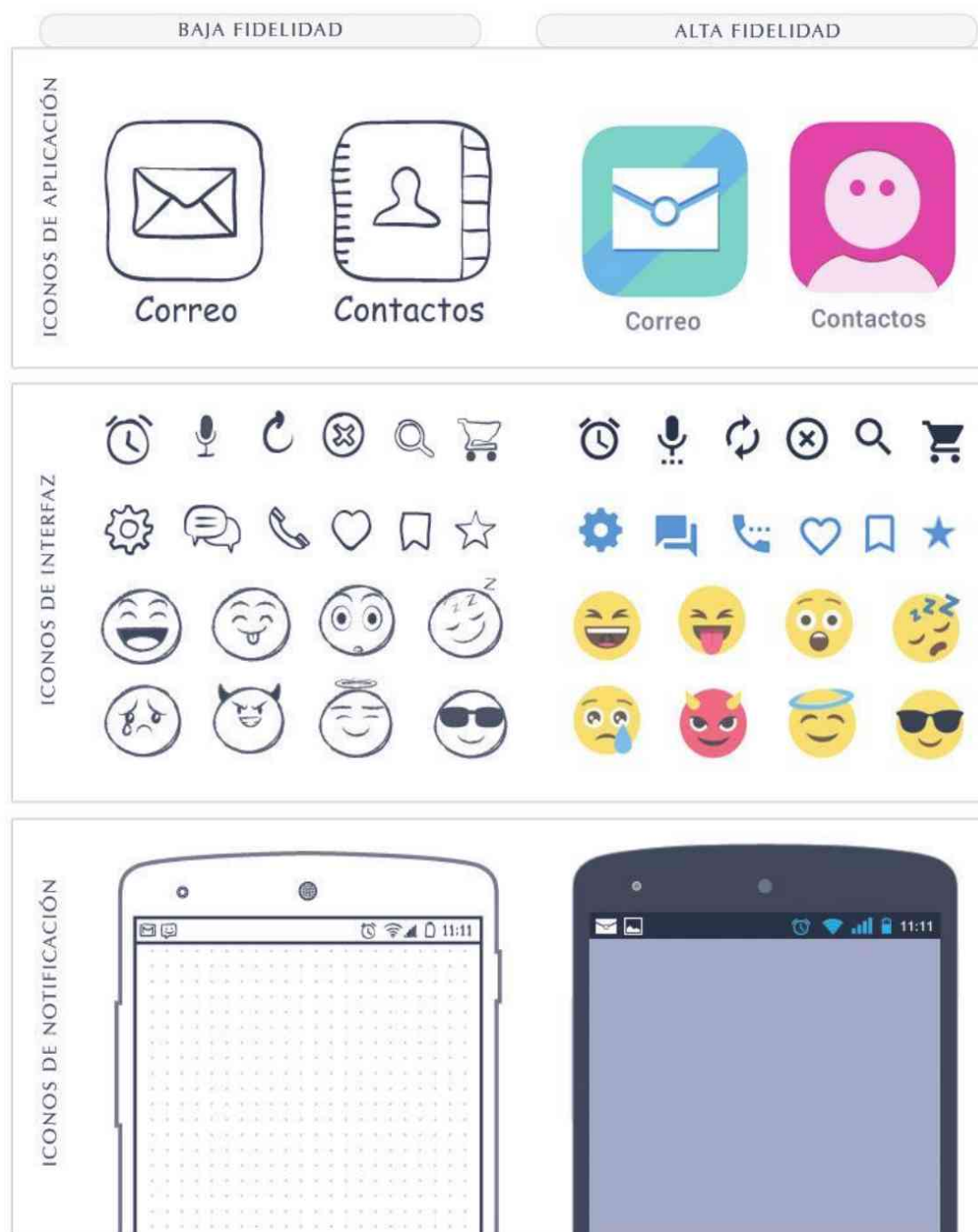


Figura 3-9 Ejemplos de iconos de aplicación, iconos de interfaz e iconos de notificación muy utilizados en los sistemas operativos móviles. Es importante conservar la unidad visual en nuestros iconos, ya que cada diseño creado pertenecerá a una familia o sistema gráfico con características comunes en colores, trazos y estilos que se utilizarán en piezas de identidad y en nuevas pantallas de la aplicación.

Textos

Los textos informan y le otorgan sentido a la interfaz. No cabe duda que el lenguaje escrito es una de nuestras principales herramientas comunicativas siendo una representación del lenguaje oral. Por lo general se utilizan en artículos donde el material informativo es presentado por medio de contenido escrito al igual que en los demás elementos de la interfaz gráfica como botones, cuadros de diálogo y componentes del sistema.

La arquitectura de información se hace visible a través del texto y su contenido particular, por ejemplo si hacemos el ejercicio de remover el contenido de texto presente en una interfaz quedaría ininteligible, por esto su importancia en el diseño de aplicaciones y cualquier producto digital es fundamental.

En el diseño de interfaces podemos utilizar textos para etiquetar y resaltar elementos, como sucede en Windows, sistema que se apoya en el uso del texto como elemento protagonista. De esta forma es común utilizar bloques de texto escrito en encabezados de contenido específicos o títulos de pantallas. Este uso destacado es útil para definir una jerarquía del contenido general con respecto a bloques de párrafos que representan mensajes extensos como artículos de noticias, pero también descripciones cortas como las que se emplean en las notificaciones del sistema y en los *tooltips* emergentes.

Otro uso común del texto en aplicaciones se puede ver en mensajes descriptivos o informativos con indicaciones necesarias para rellenar formularios en campos de inserción de texto e indicadores de progreso. También se emplean como un recurso valioso en componentes extendidos para presentar subtítulos y letras de canciones, utilizados en aplicaciones generales para reproducir videos, películas, música y audio-narraciones.

También es importante emplear textos legibles en la elaboración de prototipos (sea de baja o alta fidelidad) asegurando que diferentes términos del sistema asociados a una misma característica se escriban de forma consistente en toda la interfaz, evitando el uso de generadores automáticos de texto, conocidos como texto *dummy* o "*lorem ipsum*" [13]. Para regular el uso de términos arbitrarios es recomendable estandarizar los conceptos con métodos como el *card sorting*, descrita en el Capítulo 2, técnica muy utilizada para mantener la consistencia en los textos de la interfaz de usuario y su terminología.



Figura 3-10 Los textos son un recurso fundamental en el diseño de la interfaz gráfica. Se utilizan en botones, barras de encabezados y en bloques de texto para crear jerarquías a través de títulos, etiquetas y párrafos. También se emplean como mensajes emergentes que informan al usuario.

Audio

El audio puede tener presencia en una aplicación bajo respuestas sonoras, por ejemplo: al pulsar determinado elemento, en notificaciones al usuario o bien puede utilizarse en archivos de sonido más complejos, su elección dependerá de la naturaleza del proyecto. En la actualidad incluso existen aplicaciones que se encargan de leer textos completos al usuario para que este solo tenga que preocuparse por escuchar el contenido por medio de una narración.

Usos comunes del audio se pueden encontrar en alarmas, tonos de teléfono, efectos sonoros que complementan gestos y acciones muy utilizados en los videojuegos y aplicaciones que requieren una retroalimentación por parte del usuario utilizando el sonido como pista que indica un cambio de estado o toque en la aplicación. También se utiliza como recurso en las aplicaciones de música que permiten reproducir pistas de canciones así como narraciones como programas de radio, *podcasts* y audio-libros. Por último, en creciente auge se puede observar el uso de inteligencia artificial con interacciones en lenguaje natural a través de mensajes sonoros, por ejemplo en asistentes como Siri (iOS), Google Now (Android) y Cortana (Windows).



Figura 3-11 El audio en muchos casos es invisible en la interfaz, se escucha como consecuencia de una acción, sin embargo cuando se trata de narraciones o música se puede presentar en una interfaz a través de reproductores de sonido con líneas de tiempo.

Video

También conocido como imagen en movimiento, termina de conformar el grupo de elementos multimedia. El cine, la televisión y otros productos audiovisuales comunes en pantallas de mayor tamaño, ahora tienen su respectiva presencia móvil. En la actualidad, es común el empleo de video en aplicaciones, en especial, con la incorporación de tecnologías de comunicación 4G que permiten transferencias de datos a una velocidad muy alta.

El video es uno de los contenidos multimedia más utilizados en aplicaciones móviles de contenido y se puede visualizar a través de software del sistema o aplicaciones como Youtube, Vimeo, Netflix y Twicht, servicios muy populares que han masificado el consumo de videos enfocándose en la utilización de este recurso multimedia como canal de comunicación masivo.

También se utilizan videos en tutoriales al interior de las aplicaciones indicando al usuario cómo puede utilizar las diferentes opciones del sistema y aunque no es tan común, se pueden encontrar videos como elemento publicitario y decorativo en fondos de contenedores, un recurso en creciente auge.

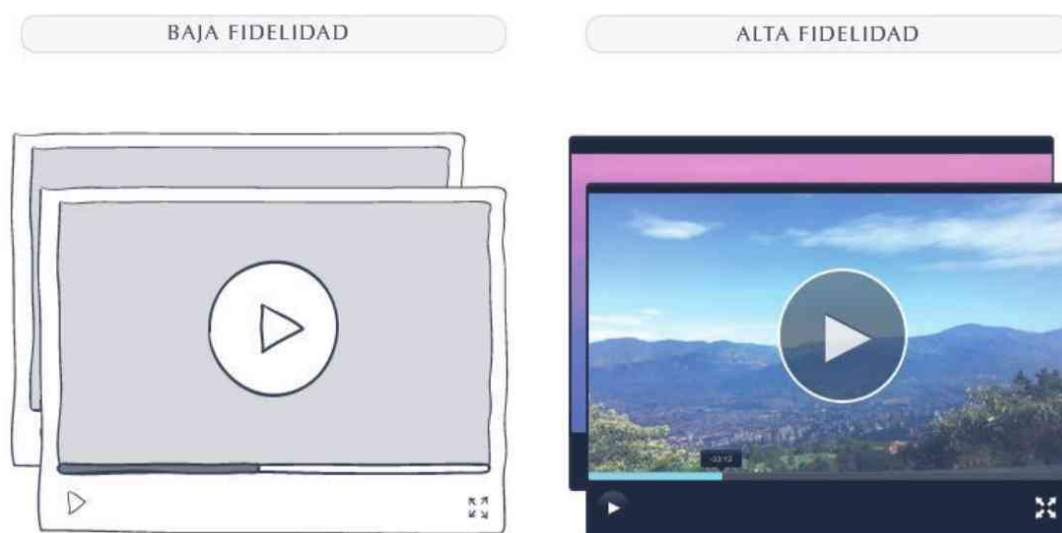


Figura 3-12 El video tiene amplia presencia en los dispositivos móviles con servicios como Youtube, así como en aplicaciones para captura y edición de video. Es común encontrar estos elementos en reproductores que permiten visualizar el video en pantalla.

Elementos interactivos

También llamados controles interactivos o *widgets* de interfaz. Son metáforas de modelos conceptuales preexistentes. Los controles permiten la transformación de la interfaz y son los que le otorgan las capacidades interactivas a la aplicación. Se utilizan de forma recurrente desde los años setenta en el diseño de interfaces gráficas y en la actualidad su empleo es tan generalizado que se han convertido en elementos comunes para los usuarios de cualquier tipo de producto informático o sistema operativo.

Muchas veces no somos conscientes de que estamos haciendo uso de ellos, pues se ha vuelto una labor cotidiana interactuar con botones, formularios, escribir una dirección de internet, disminuir el volumen de una canción, desplazarnos por la pantalla cuando el contenido es mayor, esperar cuando el sistema nos indica que se encuentra cargando contenido. Todas estas acciones se realizan a través de elementos interactivos.

Los controles interactivos son el exponente más claro de la influencia del diseño industrial en las interfaces gráficas. Muchos de ellos parten de representaciones del mundo físico, y junto con los elementos multimedia se pueden considerar como paradigmas en el diseño de interfaces. Su combinación en determinados patrones de diseño, permite crear componentes específicos resultado de la suma de sus características funcionales. Estos elementos varían entre diferentes sistemas operativos a nivel estético, pero su funcionalidad primaria no se ve modificada.

Tabla 3-6 Elementos interactivos básicos

Botones	Deslizadores
Botones de radio	Interruptores
Cajas de chequeo	Barras de desplazamiento
Listas desplegables y menús	Indicadores de progreso
Campos de inserción de texto	Transiciones y efectos de animación

Botones

Un botón es un elemento gráfico rectangular, cuadrado, ovalado o de formas varias que al ser pulsado realiza determinada acción del sistema. Esta representación gráfica obedece a la metáfora de los botones del mundo físico utilizados en todo tipo de equipos electrónicos.

Los botones son elementos fundamentales para las interacciones con una aplicación. Dada la libertad para personalizar estos elementos se pueden crear diseños propios acordes con la estética de la aplicación. Sea cual sea la naturaleza o apariencia de un botón siempre existirán como constante los estados del mismo para comunicar al usuario su actividad, por lo general son: reposo, presionado, seleccionado e inactivo. Dichos estados también se presentan en la gran mayoría de controles interactivos y su importancia es clave para tener una retroalimentación coherente por parte del sistema.

En Android, iOS, Windows Phone y en los demás sistemas se utilizan botones de manera regular, muchos de ellos en las barras de acciones acompañados por iconos específicos. Difieren en su estética particular, dependiendo del estilo gráfico por defecto de cada sistema, pero su función básica no se modifica.



Figura 3-13 Los botones son elementos universales en cualquier interfaz. Es común utilizarlos en diversos tamaños y colores. Se pueden ubicar en barras o de manera fija como sucede en Material Design con sus botones flotantes (FAB).

Botones de radio

Los botones de radio se utilizan por lo regular en formularios, para seleccionar una opción específica. El uso principal de este control es el de elegir entre varias, una única opción. Su comportamiento es excluyente, por lo que sólo es posible la selección de una opción de las disponibles. En el momento de seleccionar la alternativa deseada la anterior se desactiva para darle paso a la nueva.

Se conocen en inglés como *option buttons* o *radio buttons* y su utilización también se inspiran en una metáfora del mundo real. En los radios de los autos se usaban grupos de botones para seleccionar estaciones predefinidas [14]. En este esquema solo una de las estaciones podía estar seleccionada a la vez, cuando se elegía otra, se expulsaba el botón actual y se sintonizaba la nueva estación presionada. Esta misma solución se mantiene con los botones de radio en las interfaces gráficas.

En aplicaciones móviles se utilizan en formularios, por ejemplo cuando se va a elegir una opción de género o un rango de edad. Android y Windows utilizan este tipo de botón de forma nativa y en iOS no existe este elemento; para utilizarlo se debe recurrir a implementaciones desarrolladas por terceros.



Figura 3-14 Los botones de radio también se presentan en diversos tamaños, colores y estilos. En los ejemplos, se puede ver que sólo puede estar seleccionada una opción a la vez.

Cajas de chequeo

Las cajas de chequeo son controles que se utilizan en formularios para seleccionar una o varias opciones en el sistema. Su funcionamiento obedece a dos estados, activo o reposo. La metáfora de base en las cajas de chequeo es la de las listas impresas de cuestionarios donde se pueden seleccionar varias respuestas.

A diferencia de los botones de radio, que son excluyentes, las cajas de chequeo permiten la selección de opciones simultáneas o presentes en listados, también permiten la selección de opciones individuales.

En este último ejemplo tiene el mismo comportamiento de un interruptor, incluyendo una caja de chequeo que activa o desactiva una característica del sistema, muy común en aplicaciones móviles Android que por lo regular utilizan este control en las pantallas de configuración o en formularios específicos.

Android y Windows soportan este elemento de forma nativa, sin embargo, iOS no soporta las cajas de chequeo como control interactivo disponible en su SDK. Para incluir cajas de chequeo en iOS, como ocurre con los botones de radio, se debe recurrir a implementaciones desarrolladas por terceros.

BAJA FIDELIDAD

Elige tus intereses

☒ Cocina	☐ Deportes
☐ Autos	☒ Diseño
☒ Tecnología	☒ Fotografía

☒ Acepto los términos y condiciones

Brillo de pantalla automático ☒

ALTA FIDELIDAD

Elige tus intereses

☒ Cocina	☐ Deportes
☐ Autos	☒ Diseño
☒ Tecnología	☒ Fotografía

Medios preferidos

☒ Internet	☒ Jamón
☐ Radio	☐ Tomate
☒ Televisión	☒ Queso
☐ Correo	

☒ Acepto los términos y condiciones

Brillo de pantalla automático ☒

Figura 3-15 Cajas de chequeo diferentes con sus respectivos estados. El uso más común de este control interactivo se da a través de la selección de múltiples opciones.

Listas desplegables y menús

Las listas desplegables son elementos que permiten elegir una opción entre varias disponibles. Se diferencian con los botones de radio en la manera como se presentan: como un campo de texto que incluye un icono para indicar la presencia de más opciones. Al ser pulsado se despliega un listado y al elegir la opción deseada, la lista vuelve a su estado de reposo almacenando la información en un formulario o ejecutando alguna acción en pantalla.

Estas se conocen en inglés como *drop-down lists* y son muy empleadas para elegir una opción entre gran cantidad de datos como listados de países o idiomas donde utilizar botones de radio es poco recomendado por el espacio que ocuparían en pantalla. Windows Phone se refiere a ellas como *combo boxes* y Android como *spinners*. Por su parte, iOS no soporta este control pero tiene un componente nativo llamado *picker* que permite elegir una única opción en un listado (equivalente a las listas desplegables).

De forma similar, los menús desplegables ofrecen un comportamiento parecido a las listas desplegables pero se enfocan por lo general en la ejecución de acciones del sistema y en ofrecer opciones de navegación.



Figura 3-16 Las listas desplegables y menús se utilizan de diferentes formas, colores y estilos, su apariencia diversa no altera su función básica.

Campos de inserción de texto

Los campos de texto sirven para insertar información necesaria en un formulario, por lo general se utilizan dos tipos de campos de texto. Campo de texto de una línea y campo de texto a múltiples líneas. La diferencia entre los dos solo radica en la cantidad de texto que puede incluirse en determinadas circunstancias. Sea cual sea el objetivo, este tipo de elemento es útil para introducir texto o información que sólo el usuario sabe, datos desconocidos para el sistema y que se necesitan como parámetro para arrojar un resultado específico.

Los campos de inserción de texto se utilizan muy a menudo como cajones de búsqueda generales, para ingresar la URL de un sitio web en un navegador sea móvil o de escritorio o para registrarse en algún servicio. Como se describió, su uso mayoritario se da en los formularios (en específico de registro y de contacto).

Este control es el elemento de interacción principal en aplicaciones de chat como WhatsApp y Line donde el usuario envía mensajes a través de campos de inserción de texto, en estos mensajes también se pueden incluir otros gráficos acompañando al texto como emoticonos y símbolos específicos de uso común en las aplicaciones mencionadas.



Figura 3-17 Los campos de texto se presentan en una línea o multilínea. Son comunes en formas rectangulares, pero pueden tener estilos diversos con decoraciones originales.

Interruptores

El interruptor tiene un funcionamiento similar a los botones de radio y a las listas de chequeo. Como en los interruptores del mundo físico tiene dos estados, activo e inactivo. Por lo general se trata de un botón que sirve para activar o desactivar alguna propiedad, funcionalidad o acción de la aplicación. Estos controles se utilizan mucho en formularios de configuración del sistema y se les conoce en inglés como *switches*.

Su apariencia visual es a menudo realista, tratando de parecer interruptores físicos. El uso de estos elementos interactivos se puede ver en muchas interfaces basadas en metáforas, donde se representan objetos digitales con alto grado de detalle. Este tipo de diseño recibe el nombre en inglés de *skeuomorphism*.

Otro tipo de botón conocido como *Toggle Button* tiene un funcionamiento similar, aunque en apariencia no trata de emular un interruptor, se comporta de la misma manera, un ejemplo de uso es en las opciones de funciones de los sistemas operativos cuando se activa la rotación de pantalla, uso de datos, wi-fi en la barra de notificaciones de Android o en paneles como el Control Center de iOS introducido en la versión 7 de este sistema operativo móvil.



Figura 3-18 Los interruptores se caracterizan por remitirnos de manera realista. Incluso con el empleo de materiales, sombras y degradados.

Deslizadores

Los deslizadores se utilizan para elegir un valor en proporción a un rango. Un ejemplo de su uso lo podemos ver en los deslizadores de audio para aumentar o disminuir el volumen del sonido de una aplicación o del sistema. De igual forma también se utilizan en los reproductores de audio y video para representar la línea de tiempo de reproducción.

Como en el caso de los interruptores, siguen el estilo *skeuomorphism* y con frecuencia se representan de forma realista pareciendo elementos físicos que pueden ser pulsados. Los deslizadores se inspiran en la metáfora de los mandos de las consolas de sonido donde se pueden regular las intensidades de audio en diferentes posiciones con respecto a una fuente dada [15]. Se conocen en inglés como *sliders*.

Este tipo de control se utiliza en los diferentes sistemas operativos móviles de manera común, en iOS y Windows Phone se conoce como *slider*, en Android como *seekbar*. Aunque varía su nombre, su función principal es similar sin importar el sistema operativo. Este control es impreciso y complejo de operar cuando se trata de definir un valor exacto por lo que es adecuado utilizarlo teniendo en cuenta esta limitación [16].



Figura 3-19 Los deslizadores se suelen utilizar como controles de audio y video. Se pueden presentar en forma horizontal o vertical.

Barras de desplazamiento

Las barras de desplazamiento son conocidas en inglés como *scrollbars* y se utilizan para ayudarnos a descubrir partes de contenido cuando éste es mayor que el tamaño disponible en pantalla. En este caso, cuando se desborda el contenido, se hace necesaria una barra de desplazamiento para navegar por la pantalla en sentido horizontal o vertical.

Desde la incorporación de los teléfonos inteligentes han tenido una tendencia hacia el minimalismo. Incluso se presentan algunas innovaciones como en el caso de la barra de desplazamiento alfabética, muy utilizada en dispositivos móviles para desplazarse de forma rápida sobre un listado de contactos y encontrar con facilidad la opción deseada.

Las barras de desplazamientos han existido desde los inicios de la computación gráfica y la mayoría de usuarios están familiarizados con su uso en los navegadores Web.

En los sistemas operativos se representa con una barra en los extremos de la pantalla que sólo aparece cuando hacemos *scroll* con un gesto táctil. Aunque no es tan común, también suelen personalizarse para añadir una estética particular con colores afines a la aplicación móvil.



Figura 3-20 Algunos ejemplos de barras de desplazamiento de diferentes tamaños y orientaciones. También se puede ver un ejemplo de la barra de desplazamiento alfabética muy utilizada en aplicaciones móviles.

Indicadores de progreso

Por lo general se conocen como barras de progreso y son muy útiles para indicar en pantalla el estado actual del sistema. Son conocidas dos formas de representación de un proceso, la primera, es a través de una barra que se llena indicando el porcentaje de carga. La segunda es la versión indeterminada donde el sistema no le indica al usuario en qué etapa del proceso se encuentra, solo muestra que está procesando la petición.

Los indicadores de progreso son elementos informativos que no permiten interacciones; pese a esto, se incluyeron dentro del grupo de controles pues se relacionan de lleno con las diferentes acciones del sistema permitiendo una retroalimentación efectiva con el usuario. La manera más común de encontrarlos es en forma de barra horizontal pero también se presentan circulares o secuencias de puntos. Se conocen en inglés como *progress bars*.

Estos se ubican en el centro de la pantalla pero también se utilizan en los extremos superior o inferior. Incluso podemos verlo en diferentes controles de interfaz en barras de búsqueda y en botones como indicador de progreso, este último muy utilizado en iOS.



Figura 3-21 Ejemplos de indicadores de progreso circulares en diferentes estilos y también se incluyen algunos indicadores clásicos en forma de barra horizontal.

Transiciones y efectos de animación

El diseño de una aplicación móvil difiere al de productos estáticos como los carteles o volantes publicitarios, por el contrario, presenta diversos comportamientos y secuencias visuales al usuario, muy ligadas a la interacción con pantallas como respuesta a los gestos presentados al comienzo del capítulo.

La forma más común de utilizar animaciones viene por defecto en los sistemas operativos como una función transparente en la interacción del usuario indicando cambios de estado y revelando componentes ocultos como menús y paneles deslizables. También los indicadores de progreso utilizan animaciones que ayudan a comunicar de forma clara el estado del sistema con una función informativa. Los gestos táctiles le otorgan al usuario un control directo sobre la interfaz y permiten ampliar o disminuir el tamaño de mapas, imágenes y textos así como arrastrando objetos virtuales de un lugar a otro [17].

A menudo las animaciones soportan pequeños detalles que se presentan de manera breve para sorprender al usuario con acciones simples y sofisticadas conocidas como *microinteracciones*, un concepto reciente acuñado por Dan Saffer [18]. Un ejemplo de este tipo de detalles se puede ver en el efecto de *scroll* elástico que muestra un indicador de progreso como retroalimentación. Cada aplicación puede ofrecer múltiples efectos de este tipo y aunque la creatividad en el uso de la animación puede ayudar a establecer un vínculo emocional entre el usuario y nuestro proyecto, conviene utilizar estos recursos de forma dosificada para evitar saturarlo o distraerlo.

También se utilizan animaciones complejas y orgánicas en cierto tipo de aplicaciones como los videojuegos que se apoyan en este recurso para transmitir dinamismo y representar secuencias y efectos elaborados que mezclan sonido e imagen. Los videojuegos 2D y 3D son el exponente más claro del uso de la animación en aplicaciones. Aunque, como hemos visto, en la interacción con aplicaciones básicas se utilizan gestos apoyados en animaciones generales que se pueden representar por medio de *storyboards* y bocetos en papel entre pantallas [19]. Pero dada la naturaleza dinámica de las transiciones y animaciones, conviene utilizar servicios que permiten simular efectos en los prototipos creados para realizar pruebas rápidas de concepto. Herramientas como Framer y Proto.io permiten realizar simulaciones de elementos animados para probarlos antes de implementar.

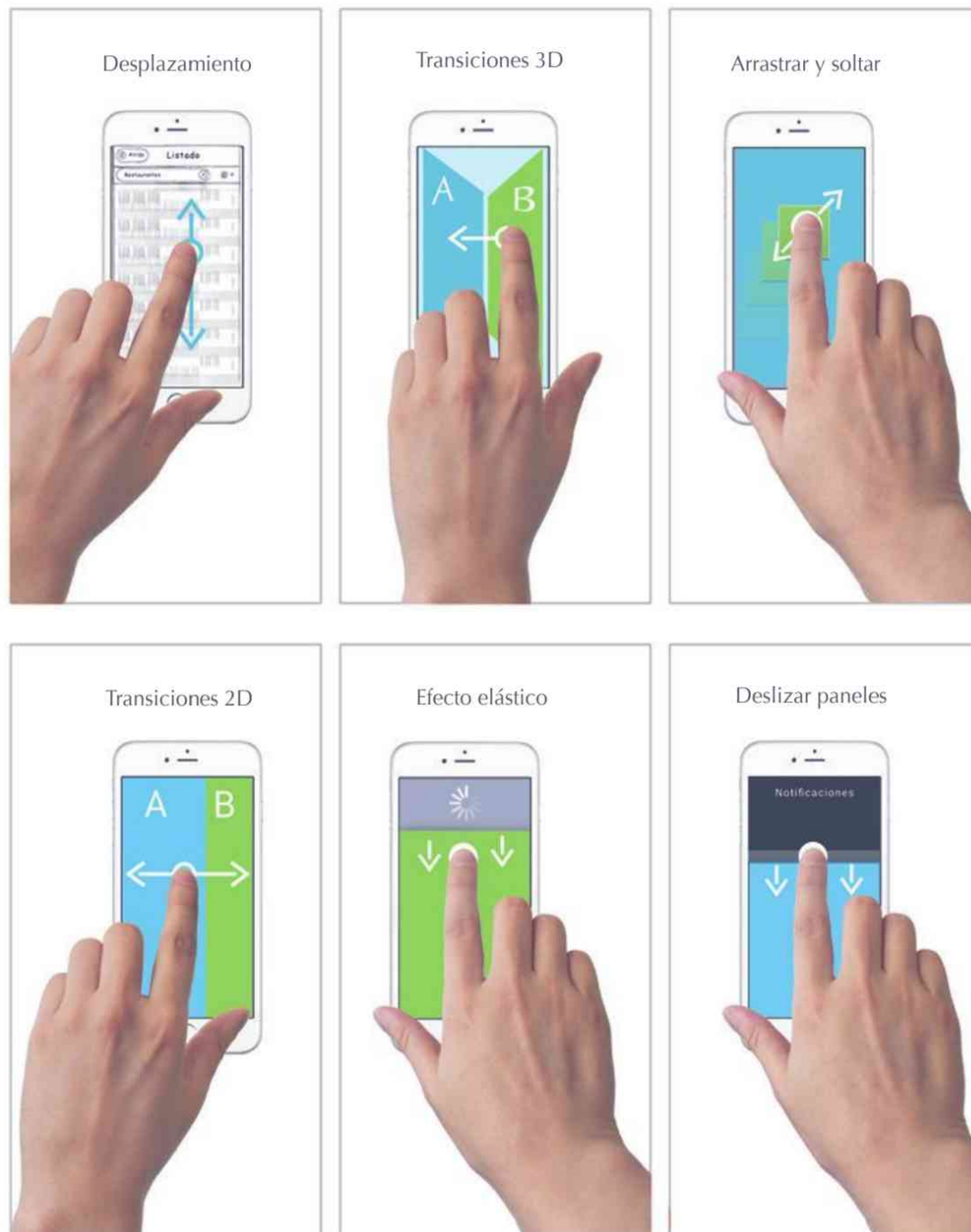


Figura 3-22 Las animaciones se utilizan para indicar comportamientos y secuencias visuales al usuario que interactúa como transiciones básicas entre pantallas o paneles ocultos, un recurso en creciente auge, pero es conveniente advertir que una interfaz no debería abusar de animaciones muy complejas para tener sentido para los usuarios, a menos que sea un videojuego o una aplicación basada en metáforas donde las animaciones cumplan una función primordial.

Patrones de diseño comunes

Los patrones son soluciones a necesidades comunes que se utilizan para resolver un determinado problema de diseño [20]. Estos problemas son repetitivos y factibles de encontrarse varias veces en el proceso de creación de un producto de software. El concepto surge por primera vez en la arquitectura de espacios físicos, fue acuñado por Christopher Alexander en los años setenta y popularizado en el mundo del software por los ingenieros Kent Beck y Ward Cunningham en el de diseño de interfaces gráficas durante los años ochenta [21]. Desde entonces, se han utilizado de manera recurrente diversas convenciones que pueden tomarse como patrones o elementos estándar en el desarrollo de aplicaciones.

El concepto de patrón también se suele utilizar desde la perspectiva de la programación, donde el código reutilizable puede ser implementado con facilidad y de esta forma se agiliza el desarrollo de software. Sin embargo, en nuestro caso nos concentraremos en la definición conceptual de los patrones más comunes, sin profundizar en la forma en cómo estos son implementados por medio de código de programación. Estos recursos son útiles para identificar soluciones efectivas en diseño y ahorrarse horas de trabajo buscando crear elementos innovadores donde no es necesario. En esencia, todo componente o patrón de diseño se construye con base en los elementos de interfaz descritos en líneas anteriores, lo que sucede aquí es que cobran significado para el usuario: una imagen por sí sola, un icono sin contexto, un botón anónimo en pantalla no significan mayor cosa si no hacen parte de un todo que ofrezca significado a través de la combinación.

Estos elementos combinados forman patrones de diseño utilizados en todo tipo de aplicaciones móviles y se pueden emplear de manera individual o conjunta dependiendo de cada interfaz. Para organizarlos, hemos categorizado los patrones en los estilos básicos de interacción de Ben Shneiderman: selección de menús, formularios, manipulación directa, lenguaje natural y lenguaje de comandos [22]. No obstante, en este capítulo nos enfocamos en los tres primeros ya que existen componentes más generalizados y estándar en el contexto móvil. Aunque es necesario aclarar que en el caso del lenguaje natural existen avances notables que lo están consolidando de un modelo experimental e inestable a una tecnología madura que promete imponerse de forma masiva en la interacción con aplicaciones móviles en próximos años.

Tabla 3-7 Estilos de interacción que se utilizaron para definir la galería de patrones

Selección de menús	Permiten una interacción por medio de menús de opciones para navegar entre secciones o ejecutar acciones del sistema. Utilizan este estilo las barras de opciones, listados, matrices, paneles de pestañas, menús laterales, tarjetas de contenido y cuadros de diálogo.
Formularios	Permiten el ingreso de datos por parte del usuario y emplean este esquema como interacción principal. Utilizan este estilo las pantallas de búsqueda, registro, ingreso de mensajes (correos, chat, comentarios, descripciones) así como los filtros de contenido.
Manipulación directa	Estilo acuñado por Ben Shneiderman que se refiere a una interacción más libre con la aplicación utilizando gestos para arrastrar y soltar objetos virtuales en pantalla, algunos patrones que utilizan este estilo son los visores de mapas, edición de imágenes, lienzos de dibujo, metáforas realistas y videojuegos.

Los patrones presentados a continuación no son los únicos existentes y en otros escritos pueden referirse a ellos con nombres diferentes o con sus propias categorizaciones. Al respecto, vale destacar las investigaciones de Theresa Neil quien ha escrito un libro entero con casos de estudio sobre patrones de diseño en aplicaciones móviles, una referencia útil en el diseño de interfaces [23]. De manera similar, hemos creado en este capítulo una galería breve de patrones comunes con ejemplos en aplicaciones reales para reconocer su empleo en diferentes tipos de aplicaciones Android, iOS y Windows. Estos patrones se pueden reconocer de una manera sencilla en el diseño de las aplicaciones más básicas ya que los sistemas operativos móviles incorporan la mayoría de estos patrones en componentes nativos soportados en sus paquetes de desarrollo. La descripción de cada patrón está acompañada de ejemplos reales de aplicaciones móviles que utilizan la solución de diseño presentada.

Selección de menús

Barras de opciones

Una aplicación móvil tiene poco espacio en pantalla para fondos gráficos muy elaborados que se utilizan a menudo en los sitios web. Esta limitación se reduce a una necesidad de optimizar la comunicación visual. En ese aspecto, el uso de barras de opciones es la solución estándar. Las barras que por lo general son encabezados, incluyen, el icono de aplicación y las opciones globales del sistema.

Para mostrar dichas opciones iOS utiliza dos tipos comunes de barras de aplicación conocidas como *Navigation Bar* y *Toolbar* la primera se destina en exclusiva para incluir el nombre de la pantalla actual y botones de navegación. La segunda se ubica en la parte inferior de la pantalla ofreciendo acciones comunes en la aplicación. Android por su parte utiliza un componente oficial llamado *Action Bar*, en esta barra de encabezado se ubican acciones del sistema y opciones generales. También cumple una función de navegación para volver a las pantallas anteriores. Windows Phone cambia este enfoque y utiliza encabezados de título con gran espacio reservado en pantalla y utiliza una barra inferior para incluir las acciones principales en un componente nativo llamado *App Bar*.

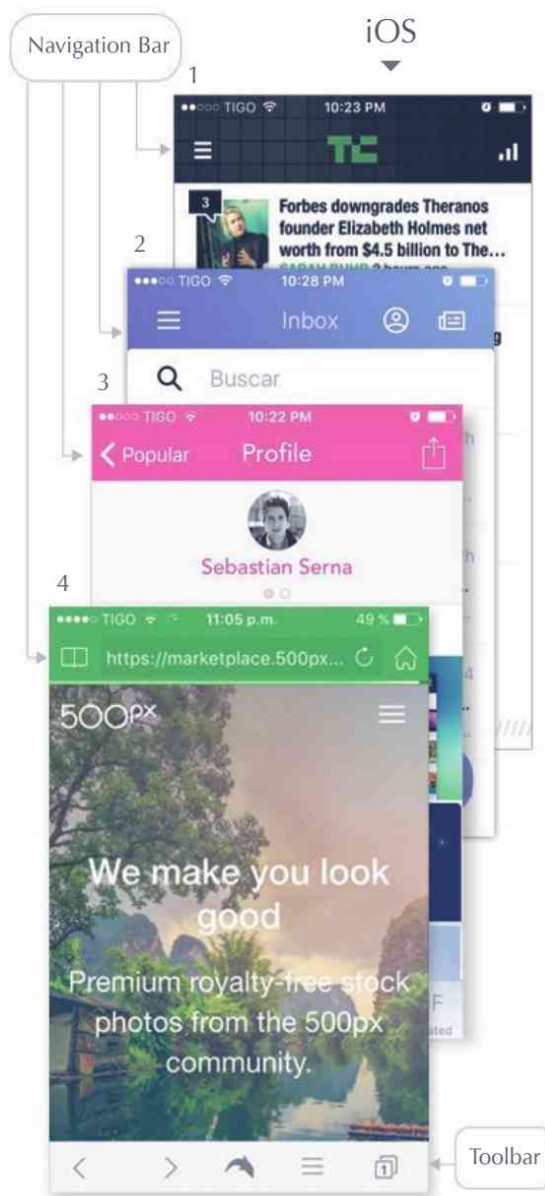


Figura 3-23 iOS utiliza dos componentes principales para ubicar las acciones en pantalla: *Navigation Bar* (parte superior) y *Toolbar* (parte inferior). Ejemplos: 1- TechCrunch (iPhone) 2- Yahoo! Mail (iPhone) 3- Posterized (iPhone) 4- Dolphin Browser (iPhone).

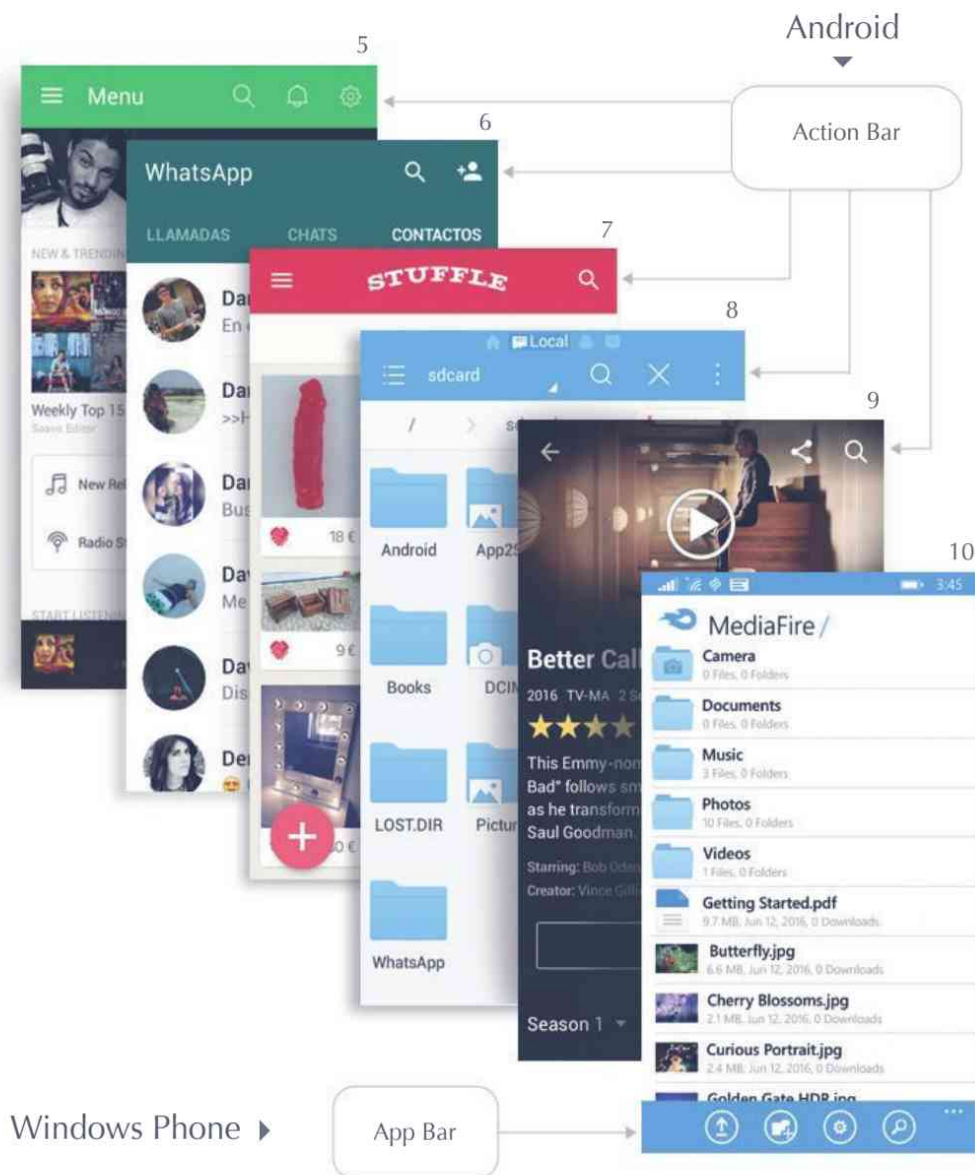


Figura 3-24 Android utiliza un componente nativo llamado Action Bar donde se ubican en la parte superior de la pantalla las acciones más comunes de la aplicación representadas por iconos, este componente puede tener un color de fondo o también puede ser transparente. Por su parte en Windows Phone se ubican dichas acciones en la parte inferior de la pantalla en un componente llamado App Bar. Ejemplos: 5- Saavn (Android) 6- WhatsApp (Android) 7- Stuffle (Android) 8- ES File Explorer (Android) 9- Netflix (Android) 10- MediaFire (Windows Phone).

Paneles de pestañas

Es uno de los patrones más utilizados en el contexto móvil. Es común encontrar interfaces de aplicaciones compuestas de un panel de pestañas con tres a cinco opciones señalizadas con iconos distintivos para cada sección. Estos paneles son similares a los que se utilizan en las aplicaciones de escritorio y aplicaciones web pero más simplificados.

Este patrón de diseño puede ubicarse en la parte inferior de la pantalla, como ocurre en las aplicaciones iOS, o puede ubicarse en la parte superior, como es común encontrar en aplicaciones Android y Windows Phone. Los paneles de pestañas permiten una navegación clara y efectiva que ayuda a simplificar la interacción con el contenido de una aplicación móvil.

Sin embargo, este patrón tiene limitaciones de espacio en pantalla. Cada sistema tiene sus métodos para presentar este grupo de pestañas, aunque no se recomienda esconder muchas opciones en un panel de pestañas específico ya que si no se indica que existe más contenido por medio de pistas visuales como flechas, puede que el usuario ignore su presencia y piense que no existen más secciones específicas de las que puede ver en pantalla ya que éstas aparecen fuera del área visible.



Figura 3-25 BBC Mundo (Android)



Figura 3-26 MBC Radio (Android)



Figura 3-27 Investing.com (Android)

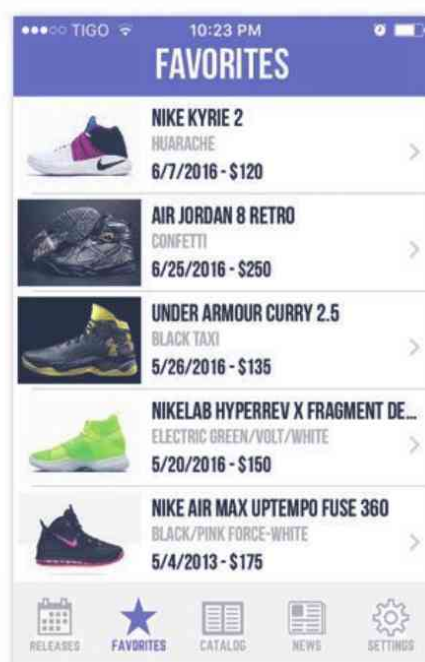


Figura 3-28 Sneaker Crush (iPhone)



Figura 3-29 Street Dance (iPhone)



Figura 3-30 Magic Hour (Android)

Listados de opciones

Se utilizan como patrones de navegación. Se presentan como un listado de opciones susceptibles de ser pulsadas. Cuando una de ellas es seleccionada, se carga una nueva pantalla con el contenido. Este patrón es muy utilizado cuando se muestran listados de noticias. Cada ítem de una lista representa un resumen o resumen del contenido que será mostrado al pulsar la opción deseada.

También se pueden utilizar como tablas de resultados musicales, entre otros. Lo interesante de este patrón es que permite incluir suficiente cantidad de texto para etiquetar las opciones de una manera ordenada. Además, cuando se acompaña de imágenes que apoyen la navegación logra ser bastante eficiente.

Los listados de opciones son patrones útiles a emplear como navegación secundaria en aplicaciones móviles, aunque en muchos casos se utilizan en conjunto con los paneles de pestañas como navegación global. Pero no es una regla absoluta, ya que algunos proyectos de aplicación también pueden utilizar listados de opciones como navegación global de la aplicación. Este patrón posee una variación que permite cargar contenido gradual, modelo que se conoce en inglés como *infinite list* [24].

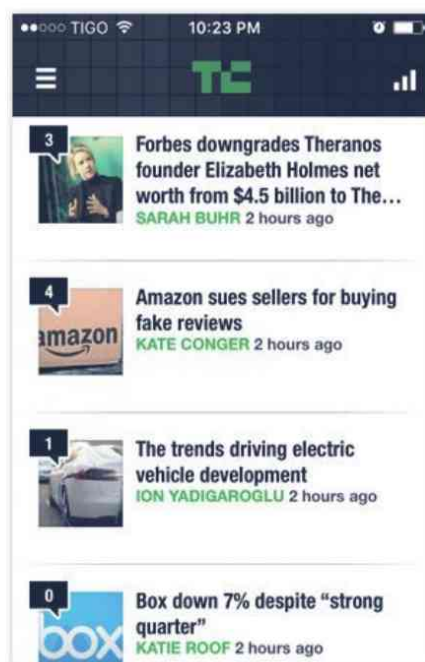


Figura 3-31 TechCrunch (iPhone)



Figura 3-32 Yahoo! Mail (iPhone)



Figura 3-33 Google Play (Android)

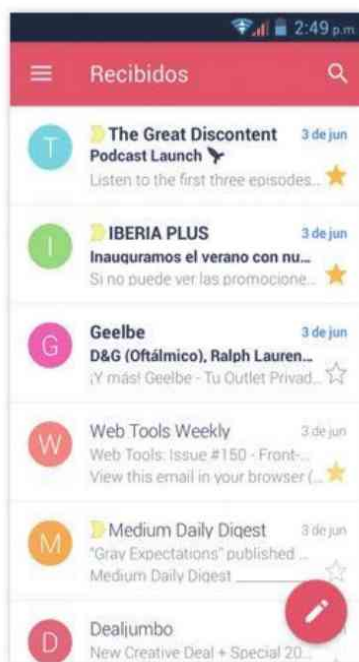


Figura 3-34 Gmail (Android)



Figura 3-35 Reuters (Android)



Figura 3-36 Zedge (Android)

Matrices de opciones

Se utilizan como patrones de navegación. En su forma básica se presentan como una matriz de iconos o imágenes disponibles. Cuando una de ellas se selecciona carga una nueva pantalla con el contenido. Este patrón es el más utilizado en las aplicaciones de navegación centralizada que se conoce como *hub and spoke*.

Aplicaciones muy populares han utilizado este esquema por su eficiencia al combinarse con filas en tipo carrusel, como hace la aplicación del grupo japonés AKB48 (Figura 3-40) en donde el contenido presentado puede ser descubierto con un gesto de paneo horizontal para darle paso a nuevos artículos. Los menús organizados con este patrón son eficientes para presentar las opciones globales de una manera clara y ordenada.

Windows Phone ha adoptado este tipo de patrón en todo el sistema operativo para presentar las diferentes opciones de navegación y aplicaciones al usuario. En este sistema los elementos de la matriz de opciones se conocen como *tiles*. Pero quizá el uso más extendido de este patrón se puede observar en galerías fotográficas. En iOS, Android y Windows Phone no se aprecian mayores diferencias en su modo de operar cuando se trata de galerías de imágenes.



Figura 3-37 iBooks (iPhone)

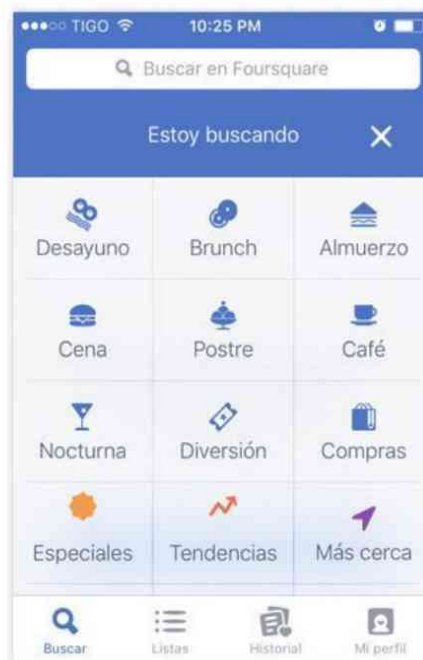


Figura 3-38 Foursquare (iPhone)

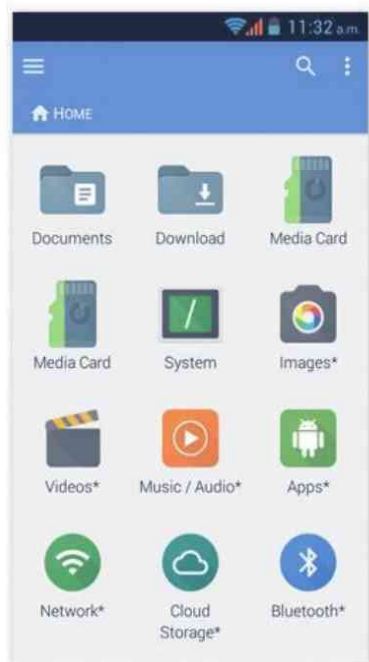


Figura 3-39 File Explorer (Android)



Figura 3-40 AKB48 (iPhone)



Figura 3-41 Netflix (Android)



Figura 3-42 Zedge (Android)

Menús laterales

Se trata de un menú escondido que se muestra llenando gran parte de la pantalla y cuando se elige la opción requerida, el menú vuelve a ocultarse. Este diseño es eficiente, en especial para pantallas de teléfono que no poseen mucho espacio. Aunque en pantallas grandes como las de las tabletas no conviene ocultarlo ya que hay suficiente espacio en pantalla para presentar las opciones junto al contenido de la aplicación.

Estos menús pueden incluir mayor cantidad de opciones comparados con los paneles de pestañas que tienen menos espacio disponible.

Por lo general se revela para el usuario al pulsar un botón de menú representado por el conocido *hamburger icon* [25]. Este patrón se utiliza de manera habitual en aplicaciones móviles iOS y Android, este último sistema operativo adoptó este tipo de menú como patrón de navegación global en aplicaciones y se encuentra disponible como un componente del sistema y todas las aplicaciones de Google han comenzado a unificar su navegación a través de menús laterales.

El uso de este patrón es útil todo tipo de aplicaciones. Sin embargo, algunos autores consideran que no es tan intuitivo incluso en teléfonos [26].

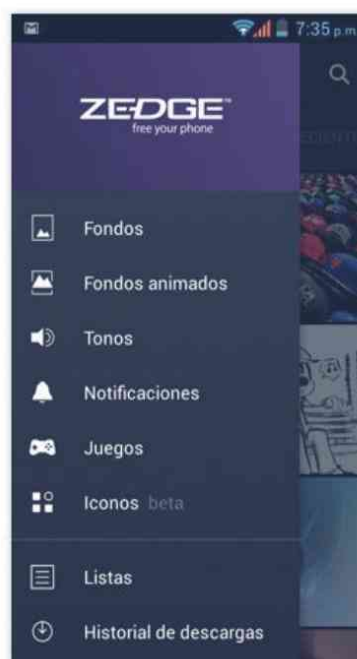


Figura 3-43 Zedge (Android)



Figura 3-44 Shoes Shopping (iPhone)

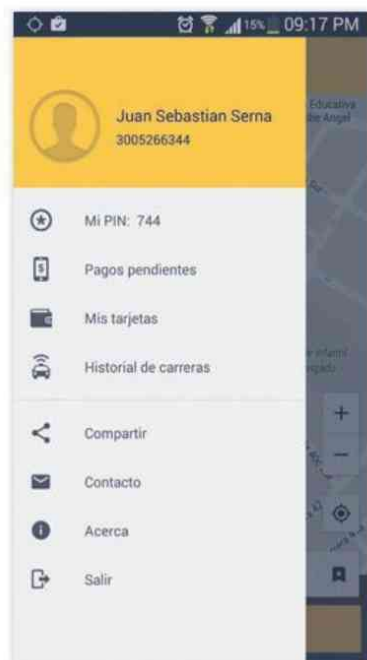


Figura 3-45 EasyTaxi (Android)

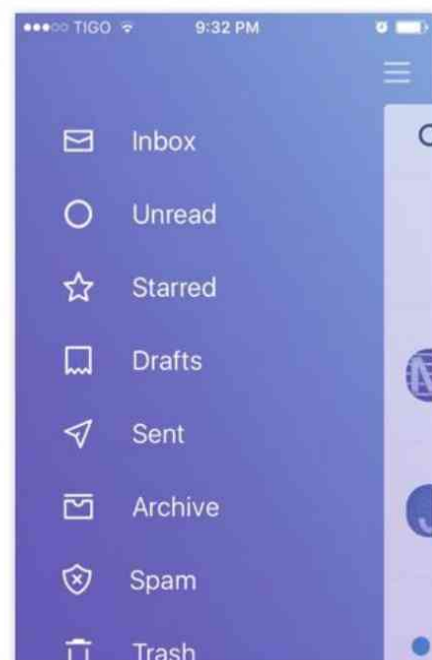


Figura 3-46 Yahoo! Mail (iPhone)

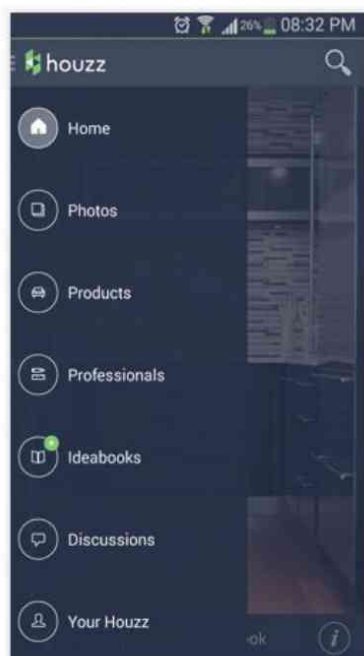


Figura 3-47 Houzz (Android)



Figura 3-48 Gmail (Android)

Tarjetas de contenido

Las tarjetas han cobrado popularidad en aplicaciones móviles. En específico Google Now hizo que muchos desarrolladores se inspiraran en esta solución específica para integrarla en su proyectos utilizando tarjetas con diferentes propósitos, para presentar el estado del clima, fragmentos de mapas y recorridos, entre otros. Las nuevas versiones de Android han incluido este patrón de diseño como un componente nativo del sistema muy utilizado en aplicaciones que siguen el estilo de Material Design.

Este patrón se utiliza como solución alternativa a los listados de opciones, ya que presentan mayor cantidad de contenido centralizado en una ficha que puede incluir imágenes y texto. La característica que la identifica es su clara delimitación en pantalla como tarjeta con respecto a las demás.

En las tarjetas se incluyen acciones que permiten navegar hacia un contenido específico o ejecutar una acción como compartir y marcar como favorito. El uso creciente de este patrón y su flexibilidad para presentar contenido le ha servido para ser el patrón habitual en las líneas de tiempo de noticias en aplicaciones como Facebook e Instagram donde el uso de tarjetas es ideal porque soportan mayor contenido por cada ítem en pantalla.

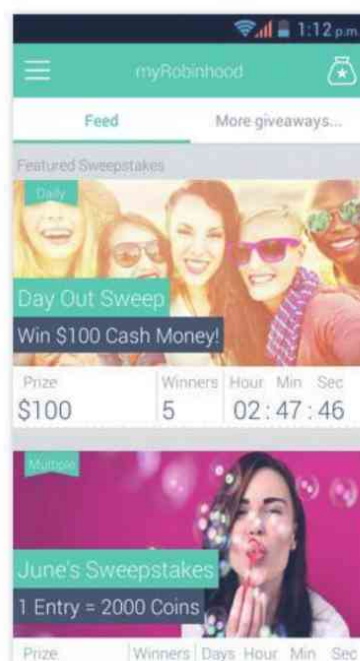


Figura 3-49 myRobinHood (Android)



Figura 3-50 Google Now (Android)

Cuadros de diálogo

El cuadro de diálogo es un patrón conocido en inglés como *dialog box*. Este se muestra al usuario para presentar un mensaje o incluir opciones a determinada acción del sistema. Son comunes en las notificaciones *push* o para tomar una decisión de un proceso que necesita una confirmación.

Por lo general, aparece como un elemento flotante que le pide una respuesta al usuario. Este patrón se incluye en la categoría de selección de menús porque en su forma básica presenta alternativas de decisión por medio de botones en pantalla para aceptar o cancelar una acción del sistema. Sin embargo, también puede contener campos de inserción de texto, en este último caso emplea un estilo de interacción híbrido que utiliza opciones de selección en conjunto con campos de formulario.

Es importante utilizarlos con moderación porque en algunas ocasiones pueden llegar a ser muy molestos para el usuario cuando se emplean en situaciones innecesarias, como sucede con los cuadros de diálogo que no ofrecen mayores opciones que confirmar algo que ya es obvio. Para este último tipo de mensajes bastaría con mostrar un *tooltip* informativo sin necesidad de solicitar una confirmación al usuario [27].

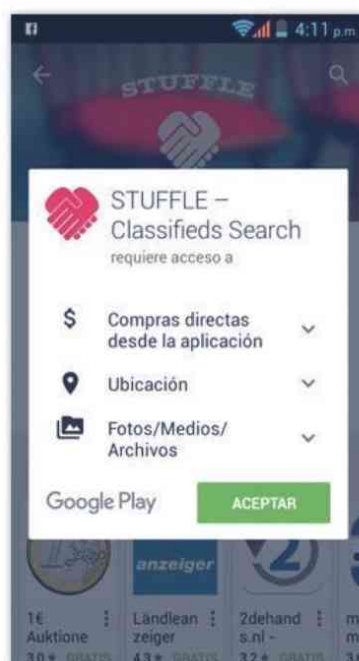


Figura 3-51 Google Play (Android)

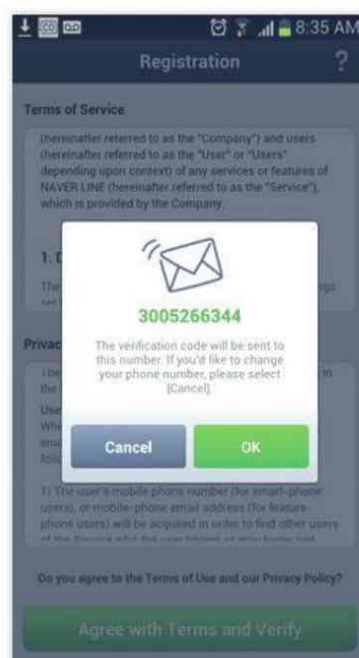


Figura 3-52 Line (Android)

Formularios

Campos de búsqueda

Los campos de búsqueda componen por sí mismos un patrón muy utilizado en productos digitales para llegar al contenido de manera directa sin navegar entre pantallas hasta llegar a la sección o contenido deseado. En una aplicación, al igual que ocurre en sitios web, son importantes los campos de búsqueda cuando se tiene mucho contenido que rota con rapidez [28], por ejemplo, en las aplicaciones de diarios, noticias o aplicaciones de gestión de correo electrónico donde encontrar un dato específico se vuelve crítico.

Se pueden ubicar en pantallas exclusivas con propiedades avanzadas para filtrar la búsqueda o pueden incluirse de forma directa en las barras de opciones de la aplicación con elementos de autocompletado instantáneo, método muy utilizado por la rapidez de respuesta para el usuario. Entre los componentes comunes que se pueden encontrar en los sistemas operativos, iOS tiene su conocida *Search Bar* que ofrece una solución de búsqueda visible en la parte superior de la aplicación. Por su parte en Android y Windows Phone también se utilizan campos de búsqueda combinando elementos nativos para inserción de texto y botones del sistema.

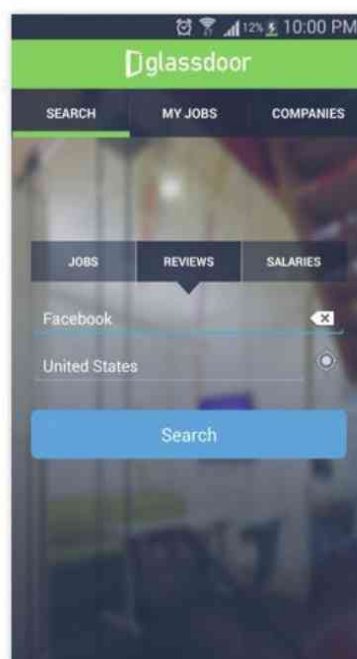


Figura 3-53 Glassdoor (Android)



Figura 3-54 IMDb (Android)



Figura 3-55 Flipboard (Android)



Figura 3-56 Google Now (Android)

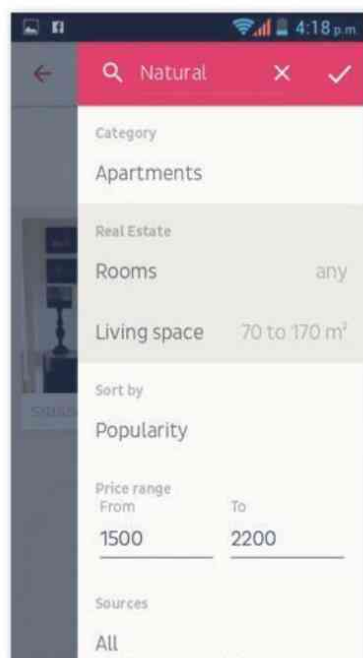


Figura 3-57 Stuffle (Android)

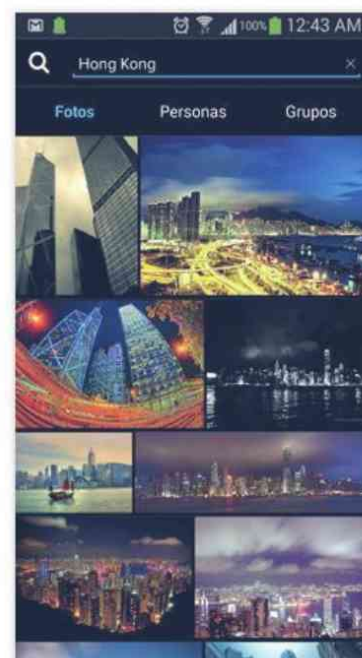


Figura 3-58 Flickr (Android)

Campos de mensajes

La interacción básica en diversos tipos de aplicaciones se realiza por medio de formularios para redactar mensajes. En algunos casos para una comunicación en tiempo real por medio de chats, en otras ocasiones (como en los gestores de correo electrónico) permite ingresar contenido en campos de texto y enviarlo a otras personas o para compartirlos en espacios de interacción como las redes sociales.

En los campos de mensajes se utiliza el texto de forma predominante pero también se pueden incluir otro tipo de contenidos por medio de archivos adjuntos como imágenes o incluso emoticonos. En aplicaciones móviles enfocadas a la comunicación vía chat como WhatsApp se utilizan campos de inserción de texto de una sola línea para aprovechar el espacio en pantalla, sin embargo, estas aplicaciones permiten ingresar tanto contenido como se desee.

En otro tipo de aplicaciones como Gmail, se utiliza el modelo clásico de composición de mensaje con los campos de correo, asunto y mensaje. Por su parte, en aplicaciones de redes sociales se utilizan a menudo como campos esenciales para actualizar el estado del usuario por medio de tweets o mensajes diversos según el tipo de contenido a compartir.



Figura 3-59 Gmail (Android)



Figura 3-60 WhatsApp (Android)

Pantallas de acceso

Es un tipo de patrón muy utilizado en aplicaciones móviles de servicios y contenido que tienen su respectiva pantalla de ingreso y registro. Se agrupan estas dos acciones porque su utilización se relaciona de manera recurrente en una misma pantalla.

Una variación que ha tenido mucha popularidad es el ingreso a través de medios sociales como Facebook y Google, este tipo de servicios facilita el proceso de ingreso en la aplicación móvil, evitando pasos de registro lentos que desaniman a los usuarios, quienes por lo general son impacientes y no están dispuestos a invertir tiempo en el registro si no están seguros de que obtendrán un beneficio real al registrarse en el servicio.

En lo posible, el registro en una aplicación debe ser opcional, permitiendo acceder a las secciones básicas sin exigirlo, sobretodo en servicios de compras [29]. Pero si se trata de una aplicación que requiera de una autenticación, utilizar las opciones de registro social puede agilizar de una manera notable este proceso.

En caso de no tener registro por medios sociales, es recomendable evitar solicitar datos innecesarios para hacer lo más rápido posible el registro con pocos campos de texto.



Figura 3-61 Airbnb (Android)

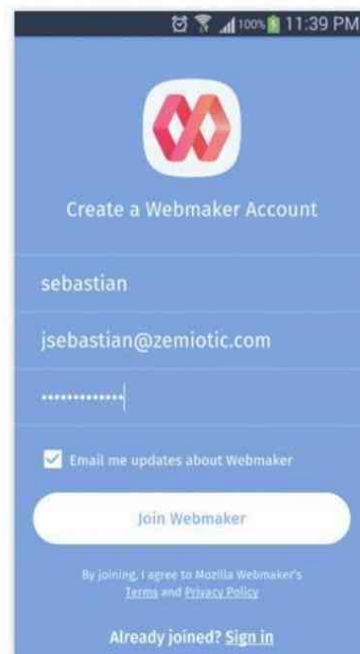


Figura 3-62 Webmaker (Android)

Manipulación directa

Lienzo de dibujo

En un creciente número de aplicaciones se utiliza el modelo de lienzo de dibujo para interactuar de una manera creativa con las pantallas de dispositivos y sus capacidades táctiles. En las aplicaciones que utilizan este patrón podemos encontrar editores de imágenes con filtros y tipografías decorativas así como programas de edición básica que permiten dibujar en un lienzo digital por medio de gestos táctiles y seleccionando herramientas en paneles de opciones respectivos.

Estos lienzos ofrecen mayor libertad de interacción y el usuario puede incluir objetos gráficos para crear collages en pantalla utilizando agrupaciones de imágenes o trazos de pincel específicos. Muchas aplicaciones que utilizan el lienzo de dibujo son utilizadas para preparar imágenes que se comparten en otras aplicaciones como Instagram o redes sociales masivas como Facebook, pero también se pueden utilizar para explorar de forma libre con gráficos y dibujo generando creaciones propias.

Para interactuar en el lienzo de dibujo con mayor comodidad se pueden utilizar accesorios externos como lápices digitales o *stylus* para lograr mayor precisión en los trazos.



Figura 3-63 MediBang Paint (Android)



Figura 3-64 Paper Artist (Android)

Metáforas

La representación en pantalla por medio de metáforas es muy utilizada en aplicaciones de entretenimiento. Este patrón es muy diverso ya que puede utilizar cualquier elemento conocido por lo general del mundo físico y trasladarlo al mundo digital.

Por ejemplo, las aplicaciones de enseñanza musical que presentan pianos o instrumentos como guitarras en pantalla y permiten interactuar con el instrumento como si se tratara de un objeto real son ejemplos de metáforas realistas como patrón.

También el uso de movimientos libres que se observa en los videojuegos son ejemplos básicos de manipulación directa en aplicaciones móviles permitiendo que el usuario seleccione personajes, vehículos y objetos en pantalla en universos creados de una manera específica para entretener y avanzar entre niveles.

Las metáforas también se pueden utilizar de una forma más básica como apoyo en la interacción con algunas aplicaciones para lectura de contenido como Flipboard que utiliza una navegación que simula el paso de hoja para avanzar entre diferentes artículos como si se tratara de una revista, este tipo de interacción natural se apoya en metáforas del mundo físico.

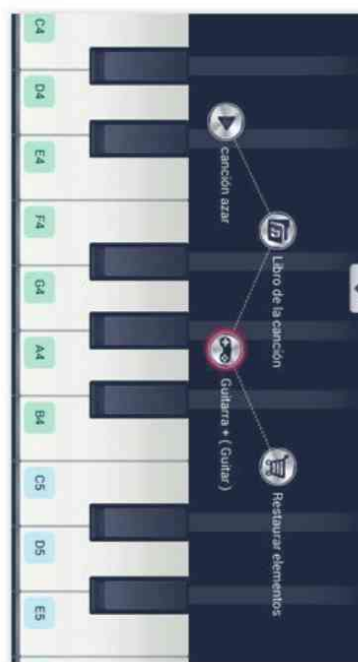


Figura 3-65 Learn Piano (Android)



Figura 3-66 Spring Ninja (Android)

Visores de mapas

La visualización de mapas es una característica muy utilizada en aplicaciones de geolocalización como Google Maps, Uber y otras. No obstante, un visor de mapa también se utiliza de forma recurrente en muchas aplicaciones para presentar la ubicación geográfica de un servicio, negocio o producto. El tipo de interacción en estas pantallas suele ser de manipulación directa donde podemos desplazarnos a través de gestos por el mapa presentado en pantalla.

Por lo general, los visores de mapas ocupan toda la pantalla del teléfono, pero también pueden ubicarse en espacios o segmentos más discretos. También es un patrón muy utilizado por aplicaciones para reserva de servicios hoteleros y de turismo en general, ya que permite la ubicación en contexto de los lugares específicos o aplicaciones sociales como Foursquare que permiten compartir la ubicación geográfica con otros usuarios.

La navegación por medio de mapas se puede realizar con gestos táctiles que permiten en algunos casos definir la perspectiva de los mismos mostrando un modelo 3D y poseen una interacción muy dinámica realizando gestos directos de *zoom in* y *zoom out* en la pantalla táctil del dispositivo.



Figura 3-67 Google Maps (Android)



Figura 3-68 Airbnb (Android)



Figura 3-69 Waze (Android)

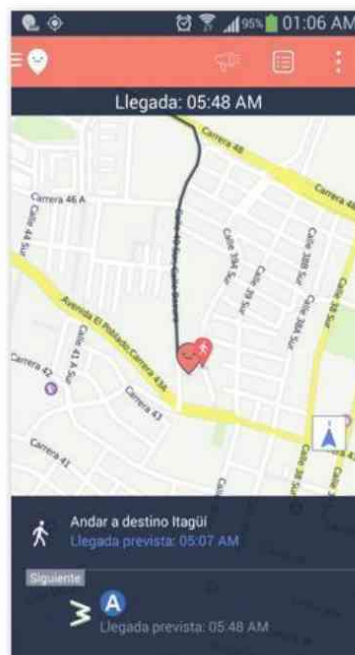


Figura 3-70 Moovit (Android)

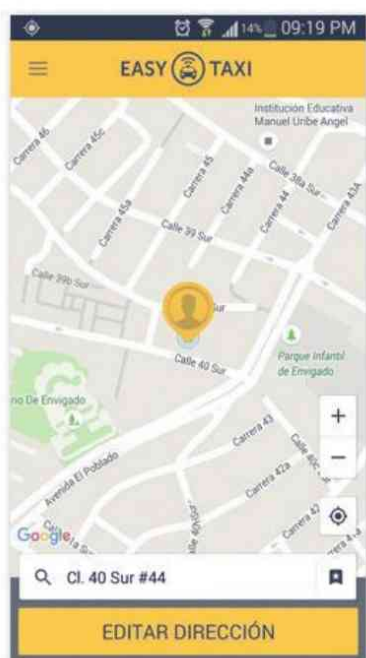


Figura 3-71 EasyTaxi (Android)

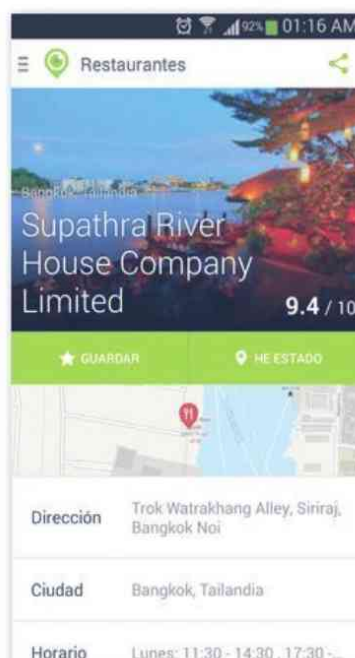


Figura 3-72 Tourist Eye (Android)

Otros patrones

Pantallas de contenido

Este patrón se puede reconocer como una pantalla exclusiva donde se muestra en detalle el contenido para lectura o visualización de contenido multimedia. Este tipo de patrón permite revelar fragmentos de contenido adicional por medio de barras de desplazamiento y en algunos casos representa el nivel de profundidad máximo en la estructura de navegación.

Se puede llegar a estas pantallas a través de componentes como listados y matrices de opciones que permiten agrupar contenidos sean noticias, fotografías, música, video o productos para comercializar que se presentan en diferentes formas al usuario.

En muchos casos, estas pantallas incluyen acciones específicas como calificar, comprar, comentar y compartir dicho contenido sea de audio, video o texto con otras aplicaciones, emulando en cierto modo los artículos de sitios web tradicionales. En muchos proyectos existe una conexión real entre un sitio web y su versión móvil, ya sea como una aplicación híbrida que presente el sitio en pantalla o una aplicación nativa sincronizada con la fuente de contenido o elementos directos del sitio web utilizando este patrón para presentar información.



Figura 3-73 News Republic (Android)



Figura 3-74 App Store (iPhone)

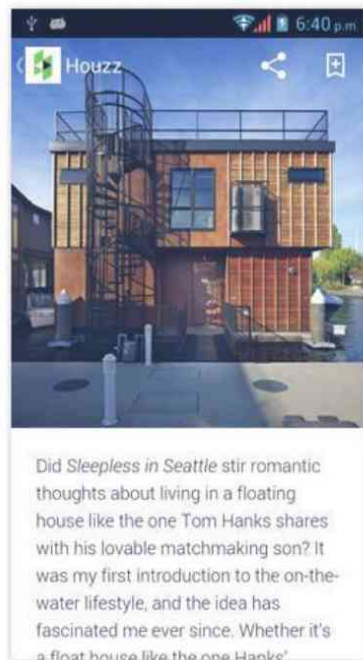


Figura 3-75 Google Kiosk (Android)

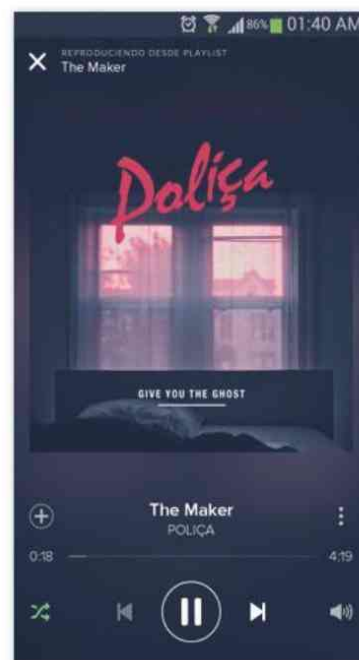


Figura 3-76 Spotify (Android)



Figura 3-77 Youtube (Android)

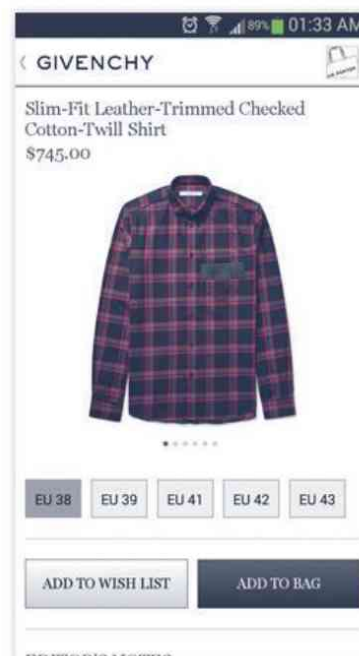


Figura 3-78 Mr. Porter (Android)

Paneles vacíos

Quizá el patrón que menos se tiene en cuenta cuando se diseña una interfaz, pero tiene mucha importancia para indicarle al usuario que no hay contenido o datos en la aplicación. Aunque en el caso más habitual se muestra un texto informativo que no permite interactuar, conviene tomar estas pantallas como una oportunidad invitando al usuario a crear contenido o utilizar las opciones básicas de la aplicación [30].

Por lo general, se muestran la primera vez que se abre la aplicación presentando las funcionalidades disponibles. Este patrón es muy necesario y recurrente en los servicios que requieren de un contenido que debe ser organizado o creado por el usuario, como sucede con los favoritos de noticias, música, videos, mensajes de correo o cualquier aplicación que permita personalizar y organizar el contenido para verlo, editarlo o escucharlo en otro momento.

También se utiliza como patrón informativo para comunicar de manera clara el estado del sistema. Por ejemplo, es común utilizar paneles vacíos con mensajes cuando la conexión a internet es inestable, situación que impide presentar contenido o datos específicos al usuario, un caso muy recurrente en aplicaciones móviles.

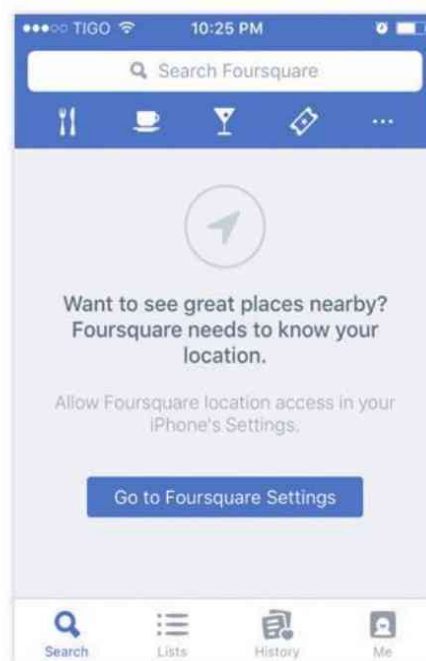


Figura 3-79 News Republic (Android)



Figura 3-80 App Store (iPhone)

Pantallas de bienvenida

También conocidas en inglés como *splash screens*. Por lo general se utilizan, como su nombre indica, para dar la bienvenida al usuario de la aplicación. Algunas poseen contenido informativo y otras permiten la realización de acciones del sistema como formularios de ingreso social a través de Facebook, Twitter cumpliendo la misma función del patrón de pantalla de ingreso o registro.

Aunque, en la mayoría de los casos se utilizan como pantallas decorativas de progreso inicial: cuando la aplicación debe cargar contenido desde Internet. Es atractivo porque ayuda al posicionamiento de marca de nuestra aplicación, ya que se presentan el logo e icono distintivo del proyecto en un tamaño considerable con respecto a las demás secciones de la aplicación móvil.

Con las pantallas de bienvenida sucede algo similar como lo fueron los intros en Flash hace algunos años en sitios web, sus detractores consideran innecesaria su implementación cuando es con fines estéticos [31]. En el contexto móvil se puede utilizar este patrón con acciones específicas, por ejemplo, solo cuando se abre por primera vez la aplicación ya que utilizarlo siempre que se abre la aplicación puede ser molesto para el usuario.



Figura 3-81 Vogue (Windows Phone)



Figura 3-82 Vyclone (iPhone)

REFERENCIAS Y LECTURAS DE INTERÉS

CAPÍTULO 3

-
- [1] P. Morville y L. Rosenfeld, *Information Architecture for the World Wide Web*, 3.^a ed. Sebastopol, California: O'Reilly Media, 2006. p. 390.
 - [2] B. Moggridge, *Designing Interactions*. Cambridge: MIT Press, 2007. p. 13-14.
 - [3] S. Hooper, «Common Misconceptions About Touch», UXmatters, 18-mar-2013. [Disponible en línea] <http://interfacesmoviles.com/ref/web33>
 - [4] «Android Metrics & keylines - Layout Touch target size», Google design guidelines. [Disponible en línea] <http://interfacesmoviles.com/ref/web34>
 - [5] «Microsoft Guidelines for targeting», 26-abr-2016. [Disponible en línea] <http://interfacesmoviles.com/ref/web35>
 - [6] L. Wroblewski, «Touch Gesture Reference Cards». [Disponible en línea] <http://interfacesmoviles.com/ref/web36>
 - [7] W. Lidwell, K. Holden, y J. Butler, *Universal Principles of Design: 125 Ways to Enhance Usability, Influence Perception, Increase Appeal, Make Better Design Decisions*, 2.^a ed. Rockport Publishers, 2010. p. 98.
 - [8] J. Clark, *Tapworthy: Designing great iPhone apps*. California: O'Reilly Media, 2010. pp. 58-61.
 - [9] S. Hooper, «How Do Users Really Hold Mobile Devices?», UXmatters, 18-feb-2013. [Disponible en línea] <http://interfacesmoviles.com/ref/web37>
 - [10] S. Weiss, *Handheld Usability*. England: Wiley Publishing, 2002. p. 70.
 - [11] L. Mathis, *Designed for Use: Create Usable Interfaces for Applications and the Web*, P2.0. United States: The Pragmatic Bookshelf, 2011. pp. 121-123.
 - [12] A. Bedford, «Icon Usability», Norman Nielsen Group, 27-jun-2014. [Disponible en línea] <http://interfacesmoviles.com/ref/web38>
 - [13] J. Ross, «Tips on Prototyping for Usability Testing», UXmatters, 08-oct-2012. [Disponible en línea] <http://interfacesmoviles.com/ref/web39>
 - [14] J. Johnson, *Gui bloopers 2.0: Common User Interface Design Don'ts and Dos* 2.^a ed. United States: Morgan Kaufmann, 2008. p. 53.
 - [15] A. Cooper, *About Face 3: The Essentials of Interaction Design*. Indianapolis: Wiley Publishing, 2007. p. 461.
 - [16] A. Bedford, «Slider Design: Rules of Thumb», Norman Nielsen Group, 13-sep-2015. [Disponible en línea] <http://interfacesmoviles.com/ref/web40>

- [17] J. Tidwell, *Designing Interfaces: Patterns for Effective Interaction Design*, 2.^a ed. Sebastopol, California: O'Reilly Media, 2011. pp. 127-129.
- [18] D. Saffer, *Microinteractions: Designing with Details*. Sebastopol, California: O'Reilly Media, 2014. pp. 98-99.
- [19] R. Hinman, «A New Mobile UX Design Material», *Smashing Magazine*, 30-oct-2012. [Disponible en línea] <http://interfacesmoviles.com/ref/web41>
- [20] E. Folmer, «Interaction Design Patterns», *The Glossary of Human Computer Interaction*. [Disponible en línea] <http://interfacesmoviles.com/ref/web42>
- [21] T. Taibi, *Design Patterns Formalization Techniques*. United States: IGI Publishing, 2007. p. VIII.
- [22] B. Shneiderman y C. Plaisant. *Designing the User Interface: Strategies for Effective HCI*, 4^a ed. United States: Addison-Wesley, 2009. pp. 71-74.
- [23] T. Neil, *Mobile Design Pattern Gallery*. California: O'Reilly Media, 2012.
- [24] J. Tidwell, *Designing Interfaces: Patterns for Effective Interaction Design*, 2.^a ed. Sebastopol, California: O'Reilly Media, 2011. p. 462.
- [25] «Hamburger icon: How these three lines mystify most people», *BBC News*. [Disponible en línea] <http://interfacesmoviles.com/ref/web43>
- [26] L. Wroblewski, «Obvious Always Wins», *lukew.com* 27-abr-2015. [Disponible en línea] <http://interfacesmoviles.com/ref/web44>
- [27] T. Neil, *Mobile Design Pattern Gallery*. California: O'Reilly, 2012. p. 229.
- [28] P. Morville y L. Rosenfeld, *Information Architecture for the World Wide Web*, 3.^a ed. Sebastopol, California: O'Reilly Media, 2006. p. 147.
- [29] A. Schade, «Don't Force Users to Register Before They Can Buy», *Norman Nielsen Group*. [Disponible en línea] <http://interfacesmoviles.com/ref/web45>
- [30] «Empty states - Patterns», *Google design guidelines*. [Disponible en línea] <http://interfacesmoviles.com/ref/web46>
- [31] S. Lennartz, «Splash Pages: Do We Really Need Them?», *Smashing Magazine*, 11-oct-2007. [Disponible en línea] <http://interfacesmoviles.com/ref/web47>

4

PRUEBAS E IMPLEMENTACIÓN

En este apartado se aborda el concepto de usabilidad y las pruebas más conocidas, éstas se pueden realizar en laboratorios especializados o en lugares de una manera informal, incluso realizar pruebas de forma remota es posible gracias a servicios en línea que ofrecen estas características. En este apartado también se mencionan las tecnologías y lenguajes utilizados para integrar un diseño de interfaz en los sistemas operativos móviles así como los entornos oficiales, pantallas e iconos de las plataformas más conocidas.

Importancia de las pruebas

Aunque este libro se concentra con énfasis en los aspectos de diseño de la interfaz gráfica, no estaría completo si evitara reconocer la importancia que tienen las pruebas de usabilidad en el desarrollo de una aplicación móvil o cualquier sistema interactivo. Aunque sea de manera breve conviene mencionar algunos conceptos básicos.

Para realizar una sesión de pruebas no es obligatorio tener una versión terminada de la aplicación, se pueden realizar con todo tipo de avances en el proyecto como prototipos en papel, o prototipos interactivos que presentan una simulación de la interfaz.

En una prueba de usabilidad se observa a las personas interactuando con la aplicación o prototipo. Este método confronta nuestras suposiciones, ya que nos lleva a “la hora de la verdad” con los usuarios, una actividad muy útil que nos permite obtener de primera mano datos relevantes sobre su satisfacción o frustración, así como sus dificultades al realizar determinadas tareas y establecer correctivos con base en la información obtenida. Estos ejercicios son muy importantes en el enfoque del Diseño Centrado en el Usuario (UCD) ya que ayudan a priorizar la implementación y a tomar decisiones pertinentes sobre la efectividad de la interfaz que estamos diseñando [1].

Invertir en usabilidad en etapas tempranas del desarrollo es un buen negocio para las empresas, ya que incrementa la productividad de los usuarios, reduce los costos de entrenamiento, pues el software resulta más sencillo de utilizar, reduce los costos de soporte y mantenimiento del mismo. De igual forma, invertir en este tipo de actividades ayuda a mitigar los riesgos del rediseño de software [2], similar a como sucede en la arquitectura de espacios físicos, donde es evidente que un problema de diseño, en la construcción de un edificio, tiene menos impacto si se resuelve en la etapa de diseño (cuando aún se encuentra en planos y modelos 3D) que cuando ya se ha construido, donde los cambios y reestructuraciones son muy costosos. Esto mismo ocurre con el software, que también tiene una etapa de diseño donde podemos elaborar prototipos y realizar pruebas con usuarios para evitar reprocesos futuros por haber desarrollado la aplicación en su totalidad sin realizar validaciones previas, situación que implicaría pérdidas de tiempo y dinero indeseadas para el equipo de desarrollo.

El concepto de usabilidad

Antes de describir en qué consisten las pruebas, conviene aclarar el significado de este concepto tan mencionado pero que genera confusión entre muchos profesionales. El término *usabilidad*, que viene del inglés *usability*, no aparece aún en el diccionario de la Real Academia Española como parte de nuestro idioma, sin embargo, se utiliza en el mundo del software de manera frecuente para referirse a la facilidad de uso de un producto. Pero esta cualidad en el contexto de una aplicación no se analiza solo desde el punto de vista funcional sino desde diferentes dimensiones para los usuarios.

La ISO (International Organization for Standardization) tiene su propia definición de la usabilidad que se apoya en tres cualidades primordiales que describe como *“La eficacia, eficiencia y satisfacción con la que usuarios específicos cumplen objetivos en un contexto de uso específico”* [3]. Las pruebas con usuarios buscan validar estas cualidades en un sistema informático. Al respecto, otra definición más desglosada la ofrece Jakob Nielsen [4], uno de los teóricos más destacados en el estudio de la usabilidad, quien define cinco características que un sistema debe tener para facilitar las tareas disponibles y ser considerado usable (Tabla 4-1).

Tabla 4-1 Atributos de un sistema usable, Jakob Nielsen (1993)

Fácil de aprender	El usuario puede comenzar a trabajar con el sistema de forma rápida.
Eficiente en el uso	Cuando el usuario aprende a utilizar el sistema puede conseguir el mayor grado de productividad posible.
Fácil de recordar	El usuario casual podrá volver al sistema (tiempo después) sin tener que aprender de nuevo las funciones principales.
Baja tasa de errores	Evita que se cometan fallas. Si el usuario comete un error permite que pueda volver al estado previo con facilidad.
Genera satisfacción	Es placentero. Los usuarios se sienten satisfechos al utilizarlo. Es importante que les guste como valor subjetivo.

Pruebas con usuarios

Observar a los usuarios y cómo utilizan nuestra aplicación puede dar más luces a un proyecto que las eternas reuniones del equipo de desarrollo con debates innecesarios en donde no se llega al acuerdo [5]. En etapas tempranas de creación, utilizar las técnicas de personas y escenarios mencionadas en el Capítulo 2 son útiles para realizar una primera versión de prototipo. Pero en cuanto tengamos un diseño tangible, lo más recomendable es validar con usuarios reales antes de tomar la decisión de implementar. Puede que el uso de patrones de diseño y modelos conceptuales nos ayude a definir la interacción utilizando elementos comunes que son conocidos por el usuario, pero esto no garantiza que van a entenderlo por completo, es aquí donde las pruebas de usabilidad nos ayudan a comprobar su efectividad. Estas pruebas se suelen realizar de forma iterativa durante todo el proceso de desarrollo.

En una clasificación formal se conocen como pruebas formativas las que se realizan durante el proceso de desarrollo de la aplicación y ayudan a identificar problemas para corregirlos antes de lanzar el producto. Por su parte, se conocen como pruebas sumativas las que se realizan luego del lanzamiento al mercado [6]. No obstante, en el alcance de lo abordado en este capítulo nos concentraremos en describir de manera breve tres modalidades comunes de pruebas que se pueden utilizar en evaluaciones, sea en etapas tempranas o luego de publicada la aplicación (Tabla 4-2).

Tabla 4-2 Modalidades comunes de pruebas

Pruebas de laboratorio	Son pruebas formales realizadas en lugares especiales o adaptados por profesionales con experiencia en pruebas de usabilidad.
Pruebas de guerrilla	Son pruebas comunes en equipos pequeños. Se realizan en cualquier lugar de manera informal y puede realizarla incluso el equipo de desarrollo.
Pruebas remotas	Un modelo utilizado tanto por profesionales en usabilidad como en equipos de desarrollo pequeños que permite realizar pruebas a distancia con herramientas en línea.

Pruebas de laboratorio

Por lo general las pruebas formales de usabilidad se realizan en un laboratorio controlado donde se observa a los usuarios interactuando con la aplicación. Este método suele ser costoso, pues se realiza en lugares reservados con equipos, software especial y cámaras de video o herramientas que sirven para capturar los movimientos de la vista del usuario en técnicas como el *eye tracking*. Sin embargo, cuando las personas encargadas de las pruebas deben viajar a diferentes sitios para realizar evaluaciones, se utilizan laboratorios portátiles que se llevan en maletas con los equipos necesarios [7].

Estas pruebas se utilizan para realizar evaluaciones de campo en el entorno propio del usuario o para transformarlos en “laboratorios temporales” reservando auditorios o salones en hoteles para realizar las sesiones. Las pruebas en laboratorios portátiles se pueden considerar formales en la medida en que las realicen personas expertas con equipos y métodos rigurosos para este fin.

Para registrar la sesión en aplicaciones móviles, se pueden instalar cámaras en bases que graban los movimientos de la mano sobre los diferentes puntos de la pantalla o también se puede utilizar software para grabar la pantalla del teléfono, para esto, herramientas como Reflector o AirServer ayudan a compartir la pantalla del móvil con el ordenador sin emplear cámaras externas.

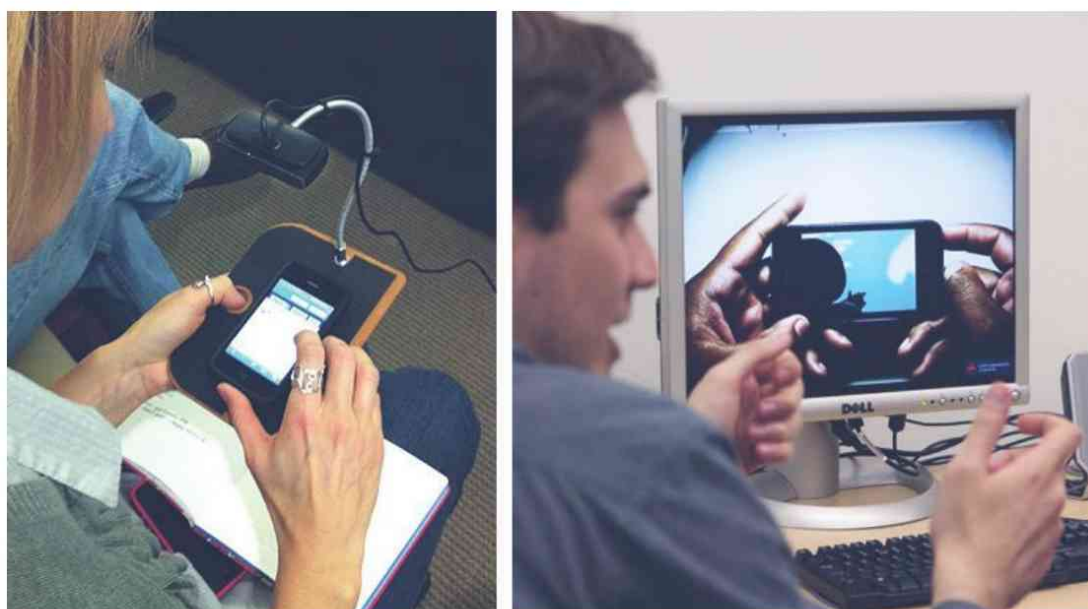


Figura 4-1 Para registrar la sesión en pruebas formales en laboratorios se instalan cámaras en el móvil.

Pruebas de guerrilla

Las pruebas informales se conocen también con el nombre de pruebas de guerrilla dada su naturaleza “subversiva”. Aquí hay que aclarar que buscan lo mismo que las pruebas formales, evaluar la usabilidad de la aplicación, pero difieren en que, las de guerrilla, se realizan con pocos recursos y no hay dispositivos especializados, ni software para capturar los movimientos del usuario, pero son efectivas en el flujo de trabajo de muchos equipos que no poseen un presupuesto elevado para realizar pruebas formales.

Se puede realizar una prueba de guerrilla en cualquier lugar, en oficinas o cafés, donde se le pide a los participantes que utilicen la aplicación y la persona que modera la prueba observa las diferentes reacciones del usuario [8]. En algunas ocasiones se graba al usuario con una cámara, pero no es esencial, ya que podemos tomar nota de los diferentes hallazgos en una libreta. Este método también es utilizado por expertos en usabilidad como una manera ágil de probar un diseño sin tener que invertir mucho dinero en las pruebas [9].

Las pruebas de guerrilla se realizan de una manera menos rígida y los usuarios elegidos por lo general son personas conocidas o también desconocidas que se abordan de forma espontánea en el sitio elegido. Para realizarlas basta con tener un prototipo, un plan definido y una libreta para tomar apuntes.



Figura 4-2 Una prueba de guerrilla se puede realizar en cualquier lugar de manera informal.

Pruebas remotas

Las pruebas remotas son una modalidad de evaluación que se realiza por internet permitiendo realizar evaluaciones con usuarios reales sin tener que desplazarse a lugares específicos, lo que las hace muy atractivas, pues reduce costos de transporte y logística, en especial cuando se realizan con participantes de otras ciudades o países. Por lo general se utilizan dos tipos: moderadas y sin moderar. En la primera el protocolo es el habitual, con la presencia remota del facilitador y observadores. En la segunda, el usuario realiza la sesión sin el acompañamiento del moderador, siguiendo unas tareas definidas que el sistema le presenta de forma automática. Para proyectos móviles, existen servicios como UserTesting o Appsee que permiten instalar aplicaciones y realizar pruebas remotas, permitiendo grabar videos de la sesión.

Otra alternativa informal pero muy práctica de compartir la pantalla del móvil se conoce como *“hug your laptop”* popularizada por Jenn Downs del equipo UX de Mailchimp. Se utiliza en sesiones de teleconferencia donde le pedimos al usuario que gire su computador portátil y “lo abrace” para que la cámara grabe la pantalla del móvil [10]. Estas pruebas son cómodas para los usuarios porque las realiza en su propia casa o entorno, pero también tienen limitaciones cuando se trata de observar el lenguaje corporal de los usuarios y sus comportamientos específicos, algo que si ofrece una sesión presencial que puede darnos más información valiosa para un análisis posterior.

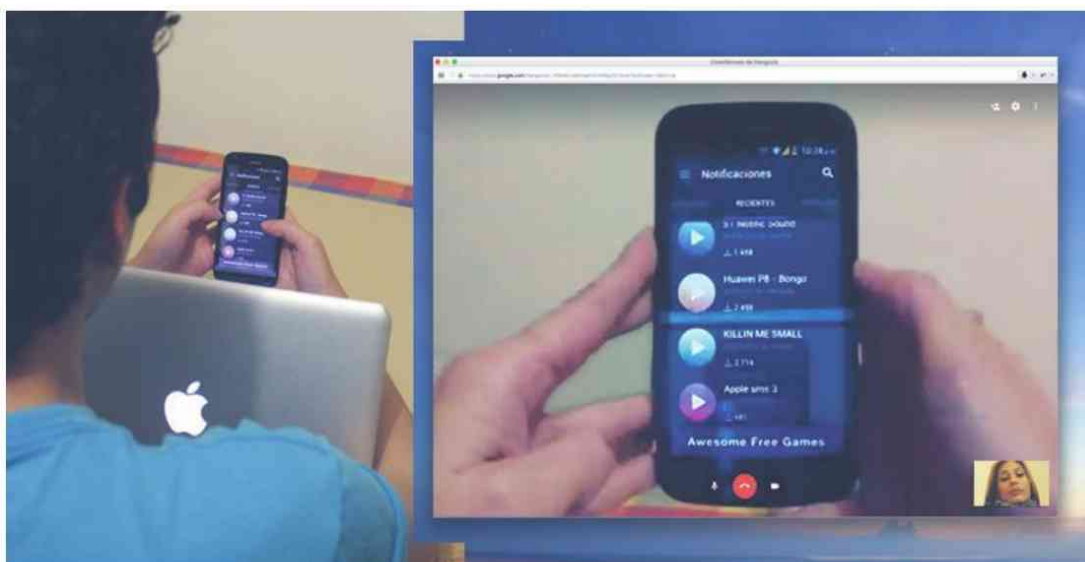


Figura 4-3 Usuario utilizando el método *“hug your laptop”* para compartir la pantalla del móvil.

Realización de las pruebas

Para realizar una sesión de pruebas, podemos organizar un plan que permita gestionar la logística necesaria y llevarla a cabo sin contratiempos. Un flujo habitual y simplificado puede incluir las siguientes etapas:

Planeación

Lo primero que debemos tener es un diseño de interfaz a evaluar, puede ser un prototipo (de alta o baja fidelidad) o una aplicación funcional, pero en cualquier caso la interfaz debe incluir contenido legible para los usuarios evitando el abuso de texto *dummy* en títulos, párrafos y nombres de personas [11]. Con este prototipo, definimos qué aspectos del sistema queremos probar, pero con el fin de optimizar el tiempo de la sesión es recomendable elegir las funcionalidades más críticas y que generen más incertidumbre para el proyecto, de esta forma se definen una serie de tareas y escenarios resultantes en un guion secuencial que indica el orden en el que los usuarios realizarán la prueba.

Preparación

Se define el lugar donde se realizará la sesión, si es un laboratorio, un salón privado o incluso si se trata de una prueba remota elegir una herramienta en línea idónea. En esta preparación, si se trata de un prototipo funcional, por ejemplo una aplicación Android o iOS, se deben tener teléfonos con la aplicación instalada y realizar revisiones técnicas antes de la sesión.

También se eligen los usuarios de acuerdo al perfil definido y si tienen características especiales, como algún tipo de discapacidad visual o auditiva, esto último tiene una exigencia mayor de planeación pues requiere recursos y material adecuado para facilitar su correcta ejecución [12], pero este no es el escenario más habitual. En cuanto al número, no hay una respuesta absoluta, dependerá de cada proyecto, sin embargo, para una prueba sencilla que busca analizar datos cualitativos, con 5 usuarios puede ser suficiente [13]. También se suele ofrecer un incentivo para los participantes: se les puede pagar una suma de dinero o definir algún tipo de retribución por su ayuda. Esta decisión y el costo del incentivo dependerá del presupuesto disponible, por ejemplo, en las pruebas de guerrilla se le da algún detalle simbólico a los usuarios (golosinas, dulces, bonos de descuento, etc) algo que los motive a participar en la sesión.

Realización

Se comienza con el protocolo de rigor de presentación, agradecimiento al usuario e indicaciones sobre la sesión, así como, en caso de tratarse de una prueba formal que requiera registro en video, la firma de consentimiento para aceptar la grabación de la sesión. Existen multitud de técnicas que se pueden utilizar, algo que se escapa del alcance de este libro, sin embargo, para la realización de una prueba de usabilidad, la actividad más elemental consiste en observar al usuario interactuando con la aplicación o prototipo.

En este método, se le pide al participante que complete las tareas definidas en la etapa de planeación con escenarios que ayuden a imaginar la situación. Por ejemplo: *Estás invitada a una fiesta de gala y no tienes aretes, ingresa a la aplicación y compra un accesorio apropiado para la ocasión*. Este tipo de indicaciones permiten contextualizar al usuario y son descripciones abiertas que mencionan el objetivo a lograr sin referirse a los elementos del sistema (como los textos descriptivos que aparecen en los botones de la interfaz).

Durante el proceso, Los usuarios pueden encontrar inconvenientes en la realización de algunas tareas, pero debemos evitar indicarles la manera de completarlas, ya que una prueba de usabilidad no es una sesión de entrenamiento para el usuario, por el contrario el objetivo principal de este tipo de sesiones es validar si el diseño es intuitivo por si mismo, sin necesidad de ayuda. Por eso, es importante explicarle al usuario que en la sesión no están siendo evaluadas sus habilidades técnicas, lo que se busca con la prueba es evaluar el diseño de la aplicación. Por este motivo, si el usuario tiene dificultades para completar una tarea, no debe sentirse culpable ya que en realidad esto solo revela una falla en la interfaz [14].

Como apoyo adicional en el proceso de observación se suele utilizar un método conocido en inglés como *Thinking Aloud*. En este, se le pide al usuario que “piense en voz alta” narrando lo que ve, lo que espera que suceda y que describa en sus propias palabras lo que está haciendo en la aplicación. Este método permite analizar las impresiones directas del usuario y qué les causa frustración o confusión mientras utilizan la interfaz más allá de sacar conclusiones o interpretar gestos subjetivos. Estas narraciones en primera persona son muy útiles para identificar problemas ignorados en un primer momento.

Elaboración de reportes

Luego de finalizar la sesión se analizan los hallazgos. La forma más común de hacerlo es con métricas, por ejemplo: el número de tareas completadas y los errores encontrados por cada usuario (como indicador de la eficacia), el tiempo que les tomó completar dichas tareas (como indicador de la eficiencia). También se pueden utilizar encuestas donde los usuarios califican su experiencia durante la sesión (como un indicador de la satisfacción). Otra métrica muy común consiste en categorizar los problemas encontrados en una escala de prioridad. Un ejemplo de este tipo de clasificación lo ofrecen Jeff Rubin y Dana Chisnell quienes ubican los problemas de usabilidad según su gravedad en una escala de 1 a 4 (Tabla 4-3) [15]. Por su parte, otros profesionales pueden utilizar una escala más sencilla donde clasifican los problemas encontrados con prioridad alta, media o baja. Aunque no es una regla absoluta, lo importante es comunicar los hallazgos en forma clara al equipo para poder priorizar y hacer correctivos en el diseño, sea la clasificación que se utilice.

Para presentar los hallazgos, diversos autores coinciden en que más allá de realizar un informe extenso y difícil de leer, es recomendable elaborar un documento breve o presentación con los resultados más importantes del estudio [16]. Aunque esto no quiere decir que se eviten los reportes detallados, ya que son útiles como material de archivo para análisis en profundidad [17].

Tabla 4-3 Clasificación de problemas de usabilidad según su gravedad. Rubin y Chisnell (2008)

4	No es usable	Es un problema donde el usuario no puede utilizar una función y completar la tarea por la forma en la que está diseñada o ha sido implementada.
3	Severo	Es un problema en donde el usuario tiene grandes dificultades y debe hacer un esfuerzo considerable para completar la tarea.
2	Moderado	Es un problema que dificulta el uso del sistema, el usuario debe esforzarse para completar la tarea.
1	Irritante	Es un problema que ocurre de forma intermitente y molesta a los usuarios. Sin embargo no impide la realización de la tarea. Es un problema cosmético.

Revisiones de expertos

Para finalizar esta breve descripción de las pruebas conviene mencionar otro tipo de métodos muy comunes para evaluar la usabilidad de una aplicación o un sistema digital. En estos métodos se elige a uno o más expertos en usabilidad que tratan de ponerse en el lugar del usuario verificando principios de diseño aceptados por el mundo profesional o realizando tareas comunes en el flujo de interacción del sistema. Estas técnicas son también conocidas como métodos de inspección de la usabilidad y se realizan como actividades de apoyo a las pruebas con usuarios. A continuación se describe uno de los métodos más utilizados y del que podemos sacar provecho para evaluar la usabilidad de nuestra aplicación móvil.

Evaluación heurística

Este método lo realizan varios expertos en usabilidad, quienes evalúan la interfaz y elaboran un informe con una calificación de los problemas encontrados. Fue propuesto en 1990 por Jakob Nielsen y Rolf Molich, donde incluye una serie de principios que se pueden tener en cuenta para diseñar y verificar la usabilidad de un sistema específico [18]. Es una de las técnicas más utilizadas en la industria como un complemento a las pruebas con usuarios dada su facilidad y bajo costo. Entre los principios aceptados, los más conocidos son las 10 heurísticas de Nielsen (Tabla 4-4) [19] pero también son relevantes las 8 reglas de oro para el diseño de interfaces de Ben Shneiderman [20] y los principios de diseño de interacción de Bruce Tognazzini [21]. Estos principios se toman como base para organizar una lista de chequeo en forma de preguntas que permiten inspeccionar, por ejemplo: el tiempo de respuesta, consistencia y características generales de la aplicación, de esta forma permite verificar los problemas más evidentes.

Si la realizan personas sin experiencia puede arrojar falsas alarmas. Al respecto, es curioso que uno de sus creadores, Rolf Molich, mencionara en una entrevista para UIE en 2003 que no utiliza con regularidad el método y no lo recomienda a sus clientes por serias inconsistencias en los resultados obtenidos en sus estudios [22]. Como él, otros expertos encuentran más valor en utilizar pruebas con usuarios [23]. A pesar de ello, es importante aclarar que este método no sustituye a una prueba de usabilidad con personas reales, se debe tomar como una actividad complementaria en el proceso general de diseño.

Tabla 4-4 Heurísticas de Jakob Nielsen (1994)

1	Visibilidad y estado del sistema	El sistema debe mantener informado al usuario sobre lo qué está pasando enviando respuestas apropiadas en un tiempo razonable.
2	Concordancia entre el sistema y el mundo real	El sistema debe hablar en el lenguaje del usuario, con palabras, frases y conceptos familiares para este; no se recomienda utilizar términos muy técnicos.
3	Control y libertad del usuario	Los usuarios a menudo eligen funciones del sistema por error y necesitan encontrar una “salida de emergencia”. Es importante soportar las funciones de deshacer y rehacer.
4	Consistencia y estándares	El usuario no tiene porqué preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. El sistema debe manejar las convenciones y estándares de la plataforma que lo soporta.
5	Prevención de errores	Mejor que buenos mensajes de error, se debe tener un diseño cuidadoso que prevenga la aparición de problemas. Cualquier condición que sea propensa al error debe presentar al usuario una opción de confirmación antes de que éste realice la acción.
6	Reconocer antes que recordar	Minimizar la carga de memoria a los usuarios haciendo que los objetos, acciones y opciones sean visibles. El usuario no tendrá que recordar información de un diálogo a otro. Las instrucciones de uso del sistema deben ser visibles o encontrables con facilidad cuando se requieran.
7	Flexibilidad y eficiencia	Los atajos ocultos para el usuario novato ayudan a mejorar la rapidez de interacción del usuario experto. El sistema puede ofrecer opciones para ambos tipos de usuario y permitir a éstos personalizar las acciones más frecuentes.
8	Estética y diseño minimalista	Los diálogos no deben tener información que sea poco necesaria o irrelevante. Cada unidad extra de información en un diálogo compite con las unidades relevantes disminuyendo su visibilidad.
9	Ayudar a los usuarios a recuperarse de los errores	Los mensajes de error deben expresarse en un lenguaje sencillo (sin códigos), indicar el problema preciso y sugerir una solución constructiva.
10	Documentación y ayuda	Sería ideal que el sistema pueda utilizarse sin documentación, sin embargo, en caso de ser necesario, toda información debe ser fácil de buscar, enfocándose en las tareas del usuario y enumerar los pasos concretos a realizar (estos no deben ser demasiado largos).

Del prototipo a la implementación

El prototipado es un proceso necesario para definir la interfaz en una etapa inicial, y aunque en el Capítulo 3 se ha hecho énfasis en realizar prototipos en papel o simulando interacciones con software, resulta necesario aclarar que algunas funcionalidades solo pueden ser probadas en un prototipo más elaborado. Cuando hablamos de implementación en este apartado, nos referimos a la integración de la interfaz gráfica utilizando las tecnologías móviles elegidas. Es la etapa en la que los gráficos necesarios (iconos, fondos, botones y demás controles) son integrados y activados por medio de código de programación, en últimas, se podría definir como el proceso de ensamblado del producto.

Como se comentó en el Capítulo 1, los sistemas operativos móviles se basan en ciertas tecnologías y lenguajes que se utilizan para programar una aplicación en sus respectivas plataformas, pero también se complementan con otros lenguajes de marcado menos complejos para definir el diseño de la interfaz por lo general en archivos XML o sus equivalentes. Para aquellos que utilicen HTML la manera de implementar la interfaz, en algunas plataformas nativas, es similar pero utilizando sus propias etiquetas y convenciones particulares como sucede en Android (que emplea XML) y Windows (que utiliza XAML).

Es común encontrar marcadas diferencias entre las plataformas móviles actuales donde cada sistema operativo fomenta el uso de sus tecnologías. Esta competitividad por el liderazgo del mercado hace que los sistemas dominantes sean elegidos por las empresas y desarrolladores. Un caso concreto es el posicionamiento de Android y iOS, situación que obliga a nuevos desarrolladores a elegir los lenguajes promovidos por estas plataformas [24]. Quizá sea este el punto menos atrayente en el desarrollo nativo para móviles, ya que al no existir estándares oficiales que regulen estas hegemonías, se tiende a controlar la implementación a través de paquetes de desarrollo SDK (Software Development Kits) que exigen un aprendizaje adicional para el desarrollador.

Por otro lado, algunos entornos como Smartface y Xamarin (adquirida por Microsoft) permiten el desarrollo de aplicaciones nativas multiplataforma programando en un solo lenguaje (como Javascript o C#) y convirtiendo el código a las demás plataformas. Aunque, no todos los desarrolladores optan por este tipo de herramientas que en su mayoría son de pago, por lo que sigue primando el desarrollo en los paquetes oficiales de cada sistema.

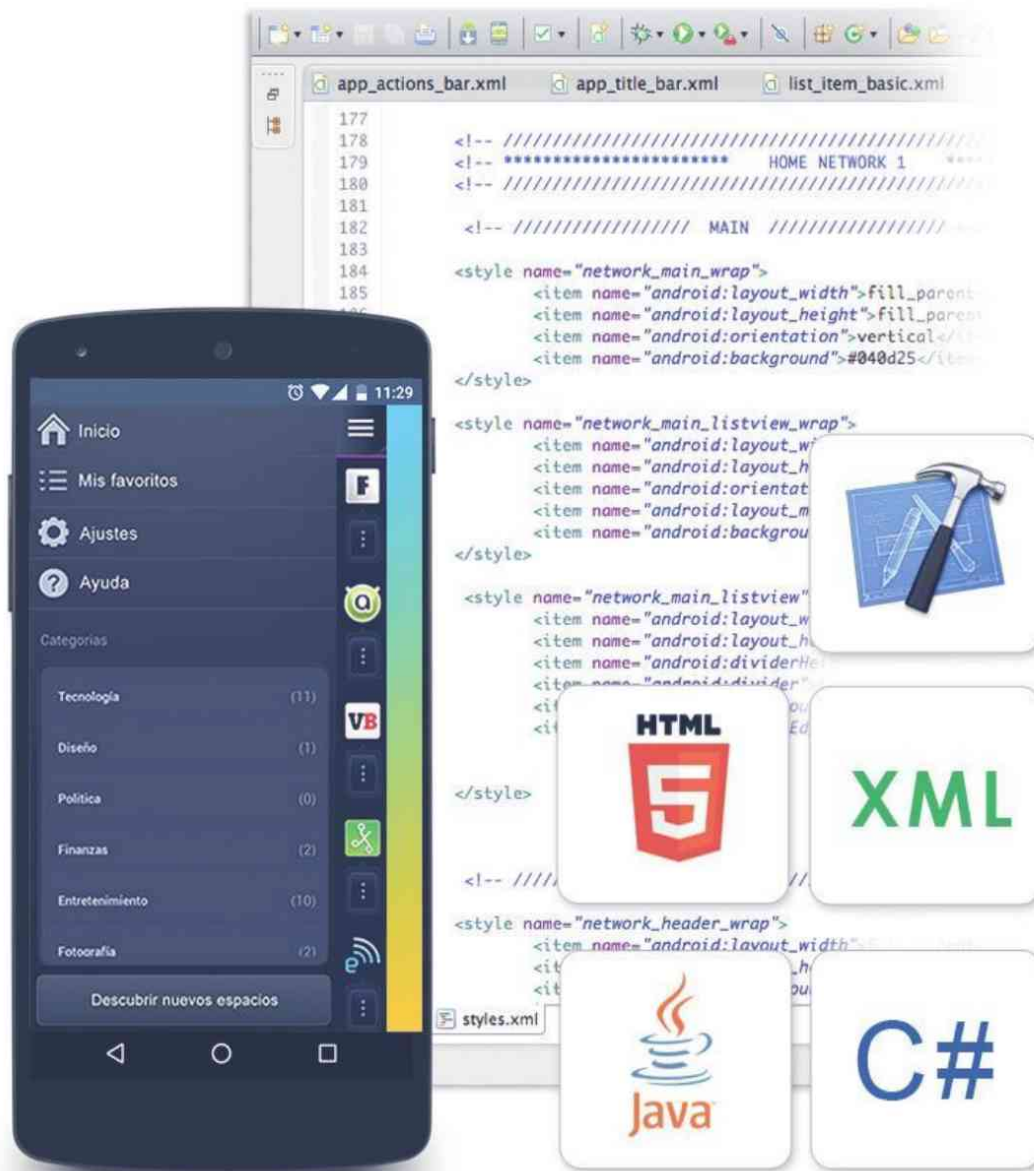


Figura 4-4 Representación de hoja de estilos en una aplicación Android en el entorno de desarrollo Eclipse. Para la implementación de la interfaz en aplicaciones se utilizan lenguajes de marcado como XML en Android y XAML en Windows Phone. Sin embargo, también se utiliza HTML en aplicaciones híbridas y web como apoyo a los diferentes lenguajes de programación oficiales en el desarrollo de software móvil en cada sistema operativo (Java, Objective C y C#).

iOS

El entorno de desarrollo para aplicaciones iOS se llama Xcode. Este software solo está disponible en Mac OS, por lo que si se tiene un ordenador con Windows o Linux la única manera de utilizar Xcode será por medio de una máquina virtual. Las aplicaciones se desarrollan en los lenguajes Objective C y Swift, este último promovido por Apple desde su lanzamiento en 2014. No obstante, para proyectos específicos iOS también soporta el desarrollo de aplicaciones programadas con C++.

Xcode utiliza un editor de diseño llamado Interface Builder que le permite a los desarrolladores construir la interfaz de la aplicación. El archivo resultante que contiene el diseño de la interfaz puede manejar el formato XIB o Storyboard. Estos dos utilizan XML para definir los elementos de interfaz, sin embargo, la creación de los mismos se realiza a través del Interface Builder, que permite arrastrar y soltar los elementos de forma visual, Xcode no soporta la edición manual de los archivos XML en aplicaciones iOS, algo que si permiten Android y Windows Phone.

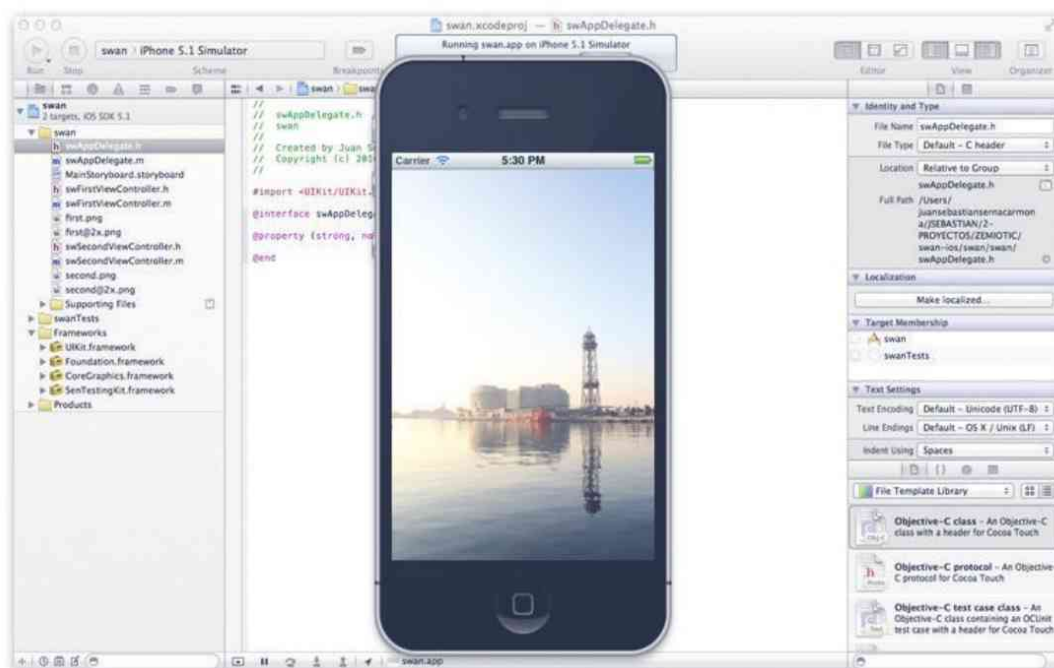


Figura 4-5 En el entorno de desarrollo Xcode se desarrollan las aplicaciones iOS. La interfaz gráfica se gestiona por medio de archivos XIB o Storyboard. Este software oficial se encuentra disponible en el sitio web para desarrolladores de Apple en <https://developer.apple.com/xcode/>

Desde la presentación de iOS, Apple ha estandarizado de manera precisa las resoluciones de pantalla de sus dispositivos en las diferentes generaciones, esta regulación ha facilitado el desarrollo de las aplicaciones móviles para su sistema operativo. El iPhone y iPad de las primeras generaciones han sido descontinuados, sin embargo, se incluyen en la Tabla 4-5 porque son la referencia original para definir los tamaños de pantalla posteriores en donde Apple utiliza los prefijos @1x, @2x y @3x para identificar cada resolución.

Tabla 4-5 Tamaños de pantallas en iOS (iPhone y iPad)

iPhone - original (@1x)		Pantalla	Densidad	Resolución px	Resolución pt
	iPhone / iPhone 3G / 3GS	3.5"	163dpi	320 x 480	320 x 480
iPhone - Retina Display (@2x)					
	iPhone 4 / 4s	3.5"	326dpi	640 x 960	320 x 480
	iPhone 5 / 5c / 5s / SE	4"	326dpi	640 x 1136	320 x 568
	iPhone 6 / 6s	4.7"	326dpi	750 x 1334	375 x 667
iPhone - Retina Display (@3x)					
	iPhone 6 Plus / 6s Plus	5.5"	401dpi	1242 x 2208 (física 1080 x 1920)	414 x 736
iPad (@1x)					
	iPad / iPad 2	9.7"	132dpi	768 x 1024	768 x 1024
	iPad mini	7.9"	163dpi	768 x 1024	768 x 1024
iPad - Retina Display (@2x)					
	iPad mini 2 / mini 3 / mini 4	7.9"	326dpi	1536 x 2048	768 x 1024
	iPad (3ª y 4ª generación) / iPad Air / Air 2 / iPad Pro	9.7"	264dpi	1536 x 2048	768 x 1024
	iPad Pro	12.9"	264dpi	2048 x 2732	1024 x 1366

Los formatos de imagen JPG y PNG son los más comunes en el diseño de aplicaciones iOS y se utilizan para gestionar los gráficos de interfaz (iconos y demás elementos visuales). Dependiendo de la pantalla estos gráficos deben ser escalados porque la plataforma en sus diferentes generaciones ha incluido pantallas con mayor resolución conocidas como Retina Display. Para garantizar su correcta visualización sin pérdida de calidad en equipos con pantallas Retina se deben crear gráficos escalados al doble (@2x) o al triple (@3x) esto se debe a que la escala estándar (@1x) que corresponde a la primera generación de pantallas ha sido descontinuada en la mayoría de dispositivos, esta resolución correspondía a la pantalla del iPhone original (presentado en 2007) donde 1pt equivalía a 1px en pantalla como se muestra en la Tabla 4-5.

Para crear las imágenes y en especial los iconos, Apple no ofrece herramientas específicas de diseño, solo se limita a definir unas recomendaciones de tamaños para las diferentes versiones que serán utilizadas en la interfaz y en su tienda de aplicaciones App Store. Para utilizar estas imágenes, la plataforma solo exige que el icono de aplicación se llame *iTunesArtwork*, los demás iconos pueden llamarse como se desee, sin embargo, para que el sistema pueda reconocer estas imágenes y presentarlas de forma correcta en cada densidad de pantalla se deben crear como *[nombre_imagen]@2x.png*, *[nombre_imagen]@3x.png*. En la Tabla 4-6 se presentan los tamaños comunes en iconos de aplicación y de interfaz para iPhone y iPad.

iOS también permite utilizar gráficos elásticos o *stretchables* que se adaptan al tamaño de botones, por ejemplo permitiendo utilizar esquinas redondeadas y otra serie de acabados; recurso que se utiliza también en aplicaciones Android y Windows Phone con técnicas equivalentes. iOS además soporta gráficos vectoriales en PDF, formato que ofrece beneficios con respecto a las imágenes JPG y PNG ya que pueden ser escalados sin inconvenientes a todas las resoluciones de pantalla, pero la utilización de dichos gráficos vectoriales requiere de conocimientos más avanzados de la plataforma.

Para el desarrollo de aplicaciones web también se generan iconos asociados a las aplicaciones, estos se conocen en iOS como *Web Clip* y se definen en los encabezados de los archivos HTML para que el navegador pueda utilizarlos como identificador de la aplicación que se presenta en pantalla. Este tipo de gráfico es un recurso importante para todo tipo de sitios web ya que en la actualidad las personas acceden a Internet desde el móvil con mayor frecuencia.

Tabla 4-6 Tamaños de iconos estándar en iOS (dimensiones en píxeles)				
Tipo de icono	iPhone 4 / 4s / 5 / 5c / 5s / SE / 6 / 6s (@2x)	iPhone 6 Plus / 6s Plus (@3x)	iPad / iPad mini (@1x)	iPad mini 2,3,4 / iPad Air / iPad Pro (@2x)
Icono de aplicación	120 x 120	180 x 180	76 x 76	152 x 152 (167 x 167 en iPad Pro)
Icono de búsqueda (Spotlight)	80 x 80	120 x 120	40 x 40	80 x 80
Icono de ajustes (Settings)	58 x 58	87 x 87	29 x 29	58 x 58
Iconos de Toolbar y Navigation bar	44 x 44	66 x 66	22 x 22	44 x 44
Iconos de Tab bar	50 x 50 (max 96 x 64)	75 x 75 (max 144 x 96)	25 x 25 (max 48 x 32)	50 x 50 (max 96 x 64)
Icono Web clip	120 x 120	180 x 180	76 x 76	152 x 152 (167 x 167 en iPad Pro)
Icono de Aplicación (App Store)	1024 x 1024	1024 x 1024	1024 x 1024	1024 x 1024

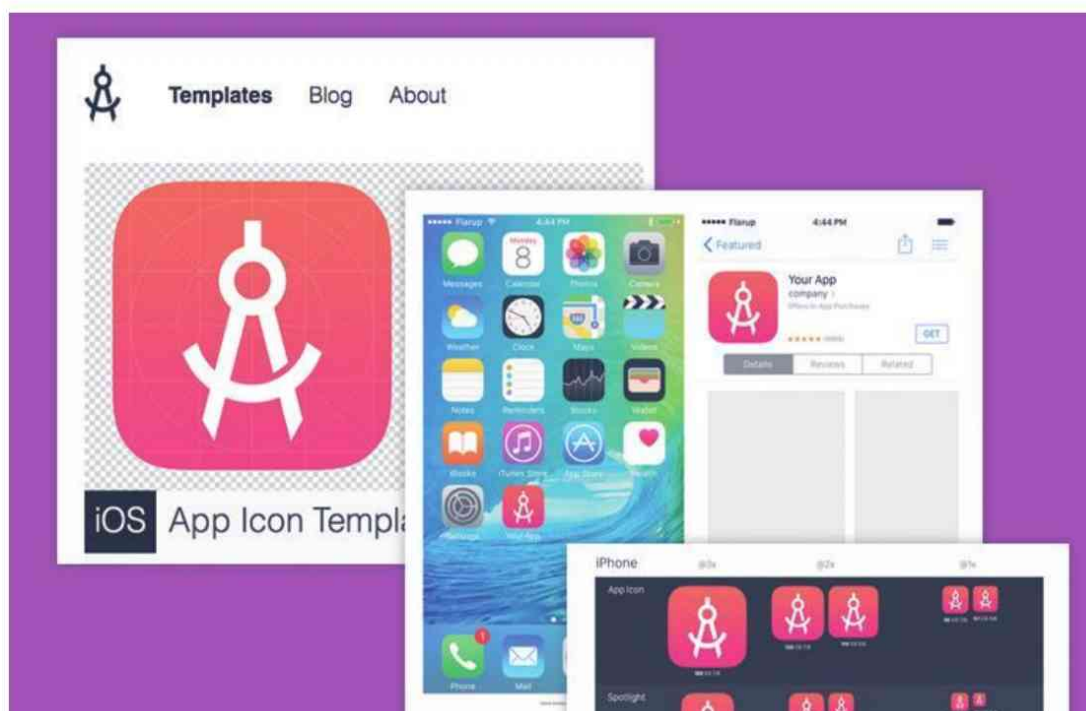


Figura 4-6 Existen plantillas gratuitas para automatizar la creación de iconos de aplicación iOS, una de las más conocidas es App Icon Template creada por el diseñador danés Michael Flarup. Se puede acceder a este recurso en la dirección <https://appicontemplate.com/>

Android

Las aplicaciones Android se desarrollan por lo general en el lenguaje de programación Java (aunque también soportan C y C++). La plataforma utiliza XML para definir las plantillas de interfaz, estos archivos XML se conocen como *layouts* y se definen con etiquetas propias del sistema operativo similar a como ocurre con HTML, pero con sus propias convenciones. También soporta archivos de estilos similar a las CSS pero utilizando de igual manera XML.

El entorno de desarrollo de la plataforma se llama Android Studio y se puede descargar de forma gratuita desde el sitio web para desarrolladores, se encuentra disponible para Windows, Linux y Mac, pero también se puede utilizar el SDK de Android en otros IDE como Eclipse, IntelliJ y Netbeans. Como se indicó en el Capítulo 3, Android representa muchas marcas de dispositivos, realidad que la convierte en la plataforma más compleja de abordar en el diseño de la interfaz. Sin embargo, como ayuda y referencia en este proceso tiene 6 densidades de pantallas para agrupar los terminales (sean teléfonos o tabletas) en la Tabla 4-7 se presentan ejemplos de estas densidades.

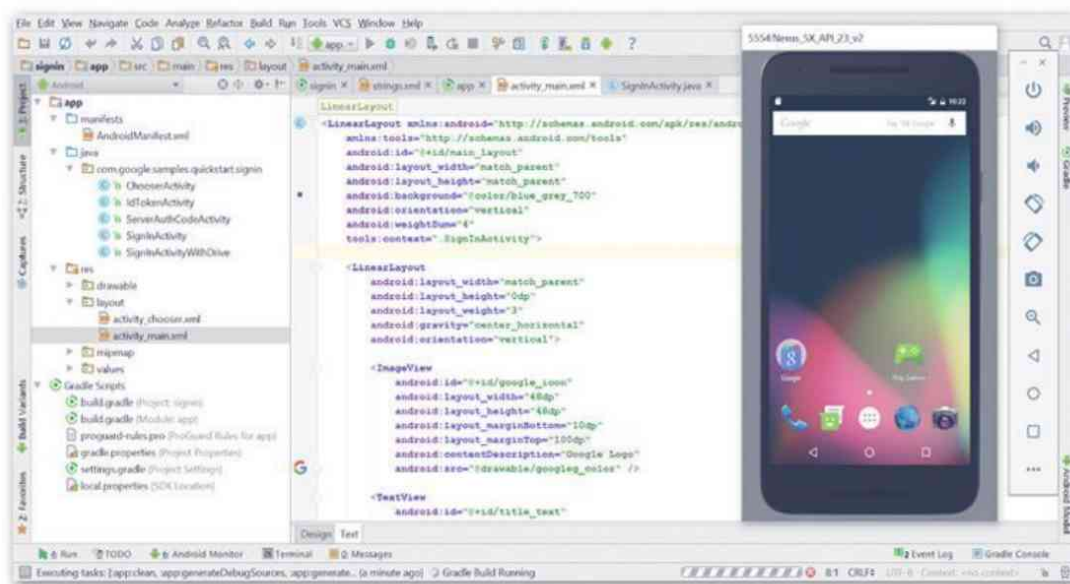


Figura 4-7 El IDE oficial de la plataforma se llama Android Studio, mismo que recomienda Google desde su presentación en 2012. Sin embargo, también se pueden utilizar otros entornos de desarrollo como Eclipse y Netbeans instalando el SDK respectivo. Android Studio se puede descargar desde el sitio de desarrolladores oficial disponible en <https://developer.android.com/studio/>

Los terminales HTC G1 y Galaxy Mini incluidos en la Tabla 4-7 están discontinuados, no obstante, se incluyeron como ejemplo para las densidades ldpi y mdpi, esta última densidad es la referencia oficial para la medida relativa dp (*density-independent pixel*) que utiliza Android en el diseño de interfaces: 1dp equivale a 1px en una pantalla con densidad de 160dpi (como la del HTC G1). En la actualidad, los móviles Android utilizan en su mayoría densidades hdpi y xhdpi pero conviene tener en cuenta que la cantidad de terminales dificulta el control sobre la visualización final. Estas densidades son orientativas y Google no las impone a los fabricantes; estos pueden manejar sus propios estándares con independencia.

Tabla 4-7 Ejemplos de pantallas Android en sus densidades oficiales				
ldpi ~120dpi (low / baja)		Pantalla	Densidad	Resolución px
	Samsung Galaxy Mini	3.2"	126dpi	240 x 320
mdpi ~160dpi (medium / media)*				
	HTC G1	3.2"	160dpi	320 x 480
hdpi ~240dpi (high / alta)				
	Nexus S	4"	233dpi	480 x 800
xhdpi ~320dpi (extra-high / extra-alta)				
	Nexus 4	4.7"	318dpi	768 x 1280
xxhdpi ~480dpi (extra-extra-high / extra-extra-alta)				
	Nexus 5	5"	445dpi	1080 x 1920
xxxhdpi ~640dpi (extra-extra-extra-high / extra-extra-extra-alta)				
	Samsung Galaxy S6	5.1"	576dpi	1440 x 2560

Por lo general, los gráficos para el diseño de interfaces se gestionan en los formatos JPG, PNG y GIF. Por otra parte, Android utiliza un formato especial de imagen PNG conocido como Nine Patch, estas imágenes con extensión .9.png utilizan un borde de 1px transparente que sirve para incluir señas de color negro y definir áreas en donde la imagen se ampliará y conservará otras regiones sin alterar, estas imágenes son el equivalente a los *stretchables* de iOS y se utilizan para que el sistema amplíe los gráficos en un tamaño definido sin distorsionar algunas partes de la imagen permitiendo generar gráficos elásticos.

Para generar este tipo de imágenes, Android tiene el editor Draw 9-patch incluido en el SDK, aunque también se pueden generar con cualquier software de edición de imágenes como Photoshop o Sketch. Los gráficos Nine Patch se utilizan en fondos de botones y otros elementos de interfaz como globos de mensaje que algunas veces tienen bordes redondeados y un tamaño final variable, por esto es una herramienta muy potente y sencilla que ayuda a facilitar este proceso de creación, utilizando una sola imagen liviana que se redimensiona al tamaño necesario. El sistema operativo utiliza muchos de estos gráficos en sus temas por defecto.

En cuanto a los iconos, elementos muy utilizados en la interfaz, debemos prepararlos para las distintas resoluciones, para esto Android soporta gráficos separados en carpetas para cada densidad. En estas carpetas se guardan las imágenes como recurso de la interfaz utilizando el mismo nombre de archivo y de esta manera el sistema (dependiendo de las pantallas) elige la imagen con la resolución respectiva. La Tabla 4-8 muestra los tamaños de iconos comunes que se utilizan en una aplicación Android en sus diferentes densidades.

También las aplicaciones Android pueden incluir gráficos generados en su totalidad con archivos XML que soportan formas básicas, colores y degradados, estos archivos también se utilizan para definir estados en los botones o diferentes gráficos para la interfaz de usuario. En caso de utilizar archivos XML para este fin, estos no se rigen por las diversas densidades de pantalla ya que se escalan a cualquier tamaño sin problemas, recurso útil para generar los gráficos XML en interfaces de colores planos ya que su mantenimiento es más sencillo para los desarrolladores que no dominan programas de diseño como Photoshop o Illustrator. Pero también es necesario advertir que los archivos XML tienen sus limitaciones en cuanto a la generación de formas muy complejas y decoraciones elaboradas como las que se utilizan en algunos iconos de interfaz.

Tabla 4-8 Tamaños comunes de iconos Android (dimensiones en píxeles)

Tipo de icono	mdpi (x1)	hdpi (x1.5)	xhdpi (x2)	xxhdpi (x3)	xxxhdpi (x4)
Icono de Aplicación (Launcher)	48 x 48	72 x 72	96 x 96	144 x 144	192 x 192
Icono de Aplicación (Google Play)	512 x 512	512 x 512	512 x 512	512 x 512	512 x 512
Iconos de Action Bar y pestañas	32 x 32	48 x 48	64 x 64	96 x 96	128 x 128
Iconos generales (en la interfaz)	40 x 40	60 x 60	80 x 80	120 x 120	160 x 160
Iconos de notificación (API Level 11)	24 x 24	36 x 36	48 x 48	72 x 72	96 x 96

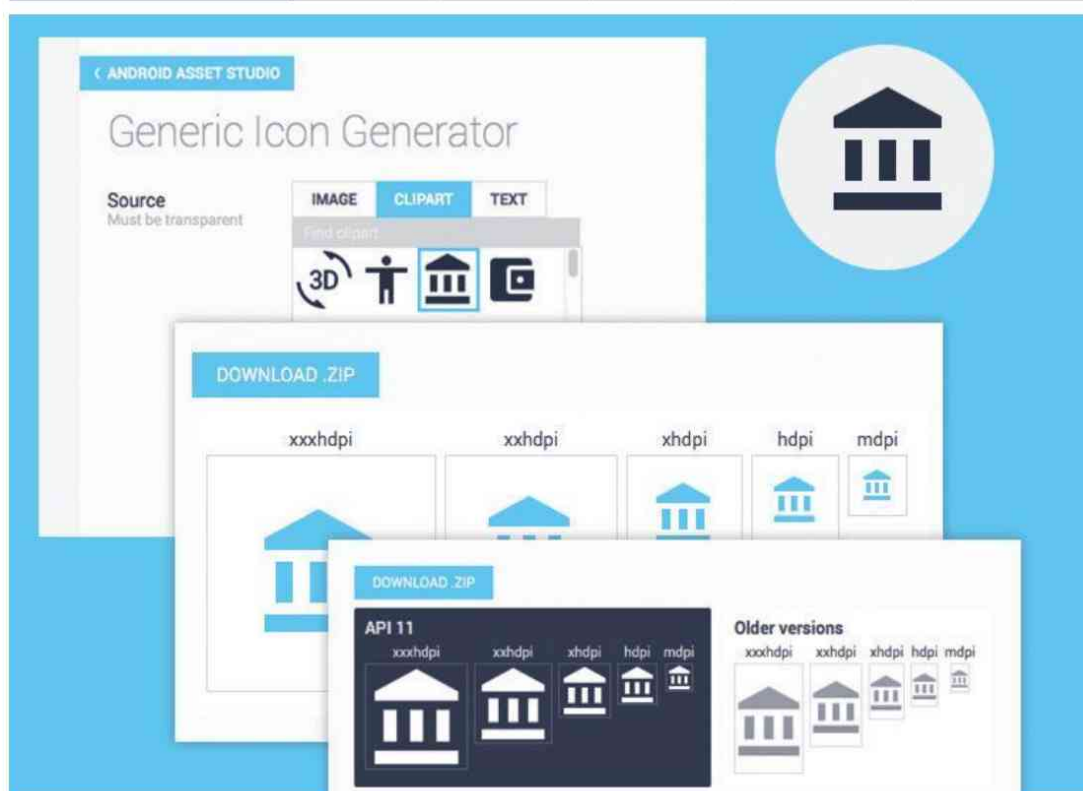


Figura 4-8 Android Asset Studio es una herramienta web muy útil para crear iconos y otro tipo de gráficos de aplicaciones Android. Permite generar iconos de interfaz, notificaciones y launchers en las diferentes densidades del sistema operativo, así como imágenes *Nine Patch* y el código XML para crear Action Bars personalizadas. También incluye el paquete de iconos de Material Design para utilizarlos en nuestra aplicación. Se puede acceder a la herramienta desde <https://romannurik.github.io/AndroidAssetStudio/>

Windows Phone

Microsoft se encuentra en un proceso de transición donde ha unificado sus sistemas desde el año 2015 con Windows 10. Con esta nueva versión, ya no diferencia entre diferentes plataformas, sino que promueve un ecosistema que utiliza aplicaciones universales para diversas familias de dispositivos. Este nuevo modelo será el empleado en próximos años, sin embargo, en este apartado con fines prácticos se presentan solo referencias comunes para Windows Phone.

El sistema operativo utiliza una variante de XML llamada XAML (eXtensible Application Markup Language) para definir la interfaz de usuario de las aplicaciones Windows en el entorno de desarrollo Visual Studio donde se puede construir la interfaz arrastrando y soltando controles básicos disponibles o escribiendo de forma directa el código XAML. Por lo general, las aplicaciones para este sistema se desarrollan en los lenguajes de programación C# y VB, aunque también se puede utilizar C++ como alternativa. Además soporta el desarrollo de aplicaciones con Javascript y HTML, lo que permite utilizar estos lenguajes para crear aplicaciones compatibles.

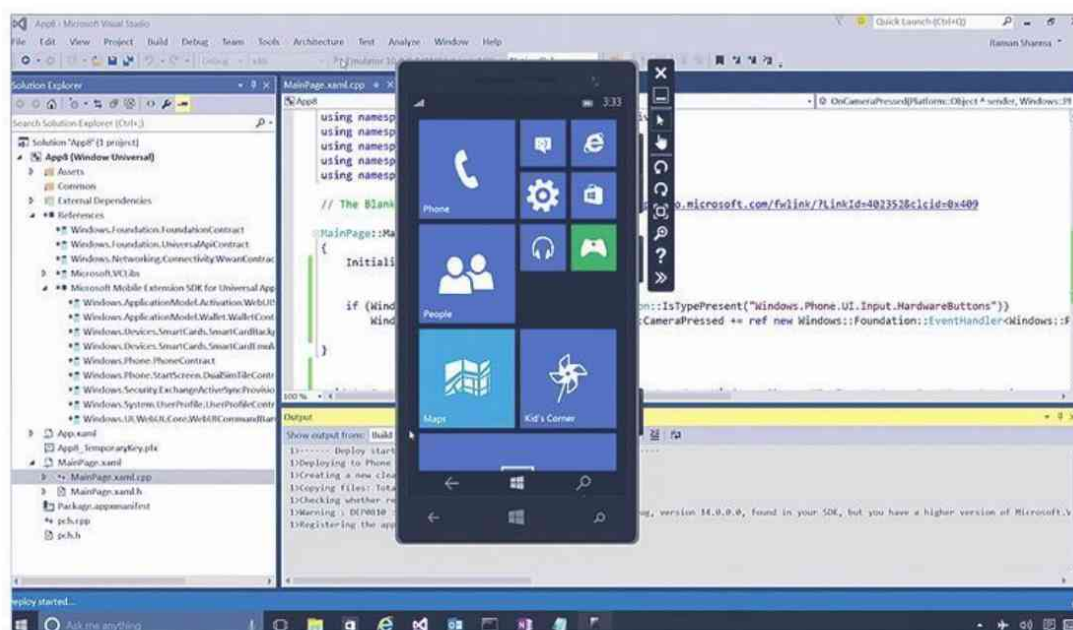


Figura 4-9 Las aplicaciones para Windows Phone se desarrollan en el entorno Microsoft Visual Studio, este software solo está disponible en Windows, sin embargo, puede utilizarse con máquinas virtuales en Mac y Linux. El sitio oficial donde se pueden encontrar las diferentes versiones de este software y sus novedades se puede encontrar en <https://www.visualstudio.com/>

Windows Phone clasifica las pantallas en cuatro tipos: WVGA, WXGA, 720p y 1080p estas dos últimas corresponden a los modelos HD de los terminales. Aunque la línea de dispositivos que utilizan este sistema operativo es reducida, comienza a presentar diversas variaciones de tamaño que hace necesario el uso de las diferentes referencias (Tabla 4-9). Se incluyen como ejemplo algunos teléfonos Lumia que corresponden a las diferentes resoluciones oficiales definidas por Microsoft. Windows 8 soporta todas las resoluciones incluidas en la tabla mientras que Windows Phone 7 solo soporta la resolución WVGA.

Los archivos XAML poseen estructuras de etiquetas similares a las que se conocen en HTML, y sirven para separar el código de programación del código de marcado para definir la interfaz. No obstante, como también sucede con Android, tendremos que aprender sus convenciones: el uso de las etiquetas y la forma en que pueden organizarse para implementar la interfaz de usuario en las aplicaciones Windows Phone. La ventaja de aprender XAML es que se utiliza en todas las aplicaciones Windows, lo que permite crear una interfaz que puede ser utilizada en diferentes versiones del sistema operativo.

Tabla 4-9 Tamaños de pantallas oficiales en Windows Phone

WVGA (480 x 800) (Referencia)		Pantalla	Densidad	Resolución px
	Lumia 520	4"	235dpi	480 x 800
WXGA (768 x 1280)				
	Lumia 920	4.5"	332dpi	768 x 1280
HD 720p				
	Lumia 735	4.7"	316dpi	720 x 1280
HD 1080p				
	Lumia 1520	6"	368dpi	1080 x 1920

A diferencia de iOS y Android, Windows Phone emplea iconos de aplicación en un modelo propio, estos iconos se utilizan como un elemento de interfaz llamado *tile*, que representa a una aplicación en la pantalla de inicio del teléfono. Es un elemento original de este sistema y funciona como una especie de *widget* que puede tener diferentes modos para presentar al usuario información específica sobre notificaciones, estados o contenido de una aplicación. La plataforma utiliza tres tipos de *tiles*: *Flip* que opera como una tarjeta con dos caras: delante y atrás, *Iconic* que presenta el icono de aplicación sobre un fondo que puede ser de color plano o incluir una imagen específica, y por último, utiliza los *Cycle*, estos se utilizan para mostrar diferentes imágenes que pasan como una secuencia de diapositivas (permite incluir hasta 9 imágenes).

Para los *tiles* mencionados, Microsoft recomienda evitar el uso como fondo de los colores blanco y negro, ya que estos se utilizan de forma regular en los diferentes estilos del sistema operativo, por lo que si se aplican dichos colores en el *tile* puede perder su límite visual apreciable en la pantalla de inicio.

En cuanto a los iconos de interfaz, Microsoft ha creado su propia familia Segoe UI Symbol que se puede integrar en las interfaces utilizando un método en creciente auge en el diseño de interfaces: el uso de fuentes tipográficas de iconos, que permite emplear estos gráficos y escalarlos sin problema, ya que se comportan con las mismas propiedades que el texto, de esta forma pueden ser escalados y cambiar el color como se hace con los textos. Las fuentes de iconos también se pueden utilizar como recurso de interfaz en aplicaciones nativas para Android y iOS [25].

Windows Phone tiene una herramienta de diseño conocida como Blend, software que se complementa con Visual Studio, integrada en la *suite* a partir de la versión 2013, y se enfoca en la construcción de la interfaz gráfica permitiendo definir el diseño de una forma más sencilla e incluso creando animaciones para las aplicaciones desarrolladas. Con Blend se pueden construir de una manera sencilla interfaces XAML, lo que garantiza la compatibilidad entre aplicaciones diversas en la plataforma. Este software también permite crear gráficos como los *Nine-patch* de Android o los *Stretchables* de iOS dividiendo imágenes en varias secciones que a través de indicaciones específicas se amplían a un tamaño definido, conservando sus esquinas en un tamaño fijo y también se puede gestionar el color y realizar animaciones básicas en el desarrollo de aplicaciones para Windows Phone.

Tabla 4-10 Tamaños comunes de *tiles* en Windows Phone

Tipo de tile	Pequeño	Mediano	Ancho
Flip	159 × 159	336 × 336	691 × 336
Cycle	159 × 159	336 × 336	691 × 336
Iconic	110 × 110	202 × 202	No es necesario

Tabla 4-11 Tamaños de iconos de aplicación

Tipo de icono	Windows Phone 7	Windows Phone 8
Icono de aplicación	62 × 62	99 × 99
Icono de aplicación (Windows Store)	300 × 300	300 × 300
Iconos de App bar	48 × 48	76 × 76



Figura 4-10 Herramientas web como Modern UI Icons permiten adaptar los iconos de aplicación en los diferentes *tiles* y tamaños oficiales del sistema operativo utilizando un gráfico SVG. Se puede acceder a este recurso en <http://modernuiicons.com/tiles>

Equivalencias entre componentes de interfaz

Para la implementación, además de utilizar los gráficos específicos que se describieron en líneas anteriores, en los SDK de cada plataforma podemos utilizar elementos multimedia y componentes interactivos básicos, conocidos como *widgets* de interfaz. Como se describió en detalle en el Capítulo 3, los patrones de diseño más básicos también se han estandarizado en Android, iOS y Windows Phone hasta el punto de tener sus respectivos componentes nativos para utilizar en el desarrollo de aplicaciones.

En la Tabla 4-12 se presentan dichos elementos y sus equivalentes en cada sistema, estos pueden tener nombres o terminología distinta entre plataformas pero su funcionamiento básico no se ve modificado. También ocurre que diversos elementos no se encuentran disponibles en algunos sistemas, por ejemplo, el caso más conocido es la ausencia de cajas de chequeo (*checkboxes*) y botones de radio (*radio buttons*) en iOS, estos elementos o controles no soportados en forma nativa se pueden activar por medio de librerías de terceros.

Es necesario aclarar que el listado presenta solo los componentes más básicos, en los sitios oficiales para desarrolladores de cada plataforma se puede encontrar el listado completo, así como la documentación de referencia para implementar dichos elementos en el desarrollo de aplicaciones Android [26], iOS [27] y Windows Phone [28].

Tabla 4-12 Equivalencias en elementos nativos de interfaz (Android, iOS y Windows Phone)

Elementos de interfaz	Android	iOS	Windows Phone
Pantallas	- Activity	- ViewController	- Page
Contenedores	- LinearLayout - RelativeLayout - FrameLayout - TableLayout - GridLayout	- View - Container View - Stack View	- StackPanel - RelativePanel - WrapPanel - Viewbox - Hub
Textos	- TextView	- Text View - Label	- TextBlock - RichTextBlock
Imágenes / Iconos	- ImageView	- Image View - Image	- Image - ImageBrush
Audio / Video	- VideoView - AudioManager	- AVPlayer - AVAudioPlayer	- MediaElement

Tabla 4-12 Equivalencias en elementos nativos de interfaz (Android, iOS y Windows Phone)

Elementos de interfaz	Android	iOS	Windows Phone
Botones	- Button - ImageButton	- Button - Bar Button Item	- Button - RepeatButton - HyperlinkButton
Botones de radio	- RadioButton	- No disponible	- RadioButton
Cajas de chequeo	- CheckBox	- No disponible	- CheckBox
Listas desplegables	- Spinner	- Picker View	- ComboBox
Campos de inserción de texto	- EditText	- Text Field	- TextBox - PasswordBox
Interruptores	- Switch - ToggleButton	- Switch	- ToggleSwitch - ToggleButton
Deslizadores	- SeekBar	- Slider	- Slider
Barras de desplazamiento	- ScrollView	- Scroll View	- ScrollBar - ScrollViewer
Indicadores de progreso	- ProgressBar	- Progress View - Activity Indicator	- ProgressBar - ProgressRing
Barras de opciones	- Action Bar - Navigation Bar	- Navigation Bar - Toolbar	- App Bar - App Bar Menu
Paneles de pestañas	- TabView	- Tab Bar	- Pivot
Listados de opciones	- ListView	- Table View	- ListView - ListBox
Matrices de opciones	- GridView	- Collection View	- GridView - Grid
Menús laterales	- NavigationView - Navigation Drawer	- No disponible	- No disponible
Tarjetas de contenido	- CardView	- No disponible	- No disponible
Cuadros de diálogo	- AlertDialog	- Alert View	- MessageBox - Flyout
Campos de mensajes	- No disponible	- No disponible	- RichEditBox
Campos de búsqueda	- SearchView	- Search Bar	- SearchBox
Lienzo	- Canvas	- Canvas	- Canvas
Visores de mapas	- MapView	- Map View	- MapControl
Visores web	- WebView	- Web View	- WebView

REFERENCIAS Y LECTURAS DE INTERÉS

CAPÍTULO 4

-
- [1] «Benefits of User-Centered Design», Usability.gov. [Disponible en línea] <http://interfacesmoviles.com/ref/web48>
 - [2] A. Marcus, Cost-Justifying Usability, 2 Ed. Cap. 2: User Interface Design's Return on Investment: Examples and Statistics. San Francisco: Morgan Kaufmann, 2005. pp. 19-26.
 - [3] «ISO/DIS 9241-11 - Ergonomics of human-system interaction -- Part 11: Usability: Definitions and concepts», ISO. [Disponible en línea] <http://interfacesmoviles.com/ref/web49>
 - [4] J. Nielsen, Usability Engineering. Mountain View, California: Morgan Kaufmann, 1993. p. 26.
 - [5] S. Krug, Don't Make Me Think: A Common Sense Approach To Web Usability, 2ª ed. Berkeley, California: New Riders, 2006. pp. 123-127.
 - [6] C. Barnum, Usability Testing Essentials: Ready, Set...Test. United States: Morgan Kaufmann, 2010. pp. 13-17.
 - [7] J. Nielsen, «Traveling Usability Lab», Nielsen Norman Group, 10-sept-2012. [Disponible en línea] <http://interfacesmoviles.com/ref/web50>
 - [8] «Guerrilla testing: Getting input into products and services», UK Government. [Disponible en línea] <http://interfacesmoviles.com/ref/web51>
 - [9] M. Belam, «Changing the Guardian through guerilla usability testing», 23-jun-2011. [Disponible en línea] <http://interfacesmoviles.com/ref/web52>
 - [10] J. Downs, «Remote Usability Testing on Mobile Devices», MailChimp, 28-oct-2011. [Disponible en línea] <http://interfacesmoviles.com/ref/web53>
 - [11] J. Ross, «Tips on Prototyping for Usability Testing», UXmatters, 08-oct-2012. [Disponible en línea] <http://interfacesmoviles.com/ref/web39>
 - [12] S. L. Henry, Simplemente pregunta: Integración de la accesibilidad en el diseño. Lulu.com, 2008. pp. 145-149.
 - [13] J. Nielsen, «Why You Only Need to Test with 5 Users», Nielsen Norman Group, 19-mar-2000. [Disponible en línea] <http://interfacesmoviles.com/ref/web54>
 - [14] D. A. Norman, The Design of the Everyday Things: Revised and Expanded Edition. New York: Basic Books, 2013. pp. 65-67.
 - [15] J. Rubin y D. Chisnell, Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests 2ª ed. Wiley Publishing, 2008. p. 262.

- [16] C. M. Nielsen, M. Overgaard, M. B. Pedersen, y J. Stage, Feedback from Usability Evaluation to User Interface Design: Are Usability Reports Any Good?, HCI - INTERACT: International Conference. Rome: Springer Science, 2005. pp. 393-394.
- [17] J. M. Spool, «Seven Common Usability Testing Mistakes» UX Articles - User Interface Engineering (UIE). 15-feb-2005. [Disponible en línea] <http://interfacesmoviles.com/ref/web55>
- [18] J. Nielsen, Usability inspection methods. New York: John Wiley & Sons, 1994. pp. 25-32.
- [19] J. Nielsen, «10 Heuristics for User Interface Design», Nielsen Norman Group, 1995. [Disponible en línea] <http://interfacesmoviles.com/ref/web56>
- [20] B. Shneiderman, «The Eight Golden Rules of Interface Design», University of Maryland. [Disponible en línea] <http://interfacesmoviles.com/ref/web57>
- [21] B. Tognazzini, «First Principles of Interaction Design», Ask TOG, 03-may-2014. [Disponible en línea] <http://interfacesmoviles.com/ref/web58>
- [22] C. Perfetti, «Usability Testing Best Practices: An Interview with Rolf Molich», User Interface Engineering (UIE), 24-jul-2003. [Disponible en línea] <http://interfacesmoviles.com/ref/web59>
- [23] D. Chisnell, «What You Really Get From a Heuristic Evaluation», UX Mag, 19-feb-2010. [Disponible en línea] <http://interfacesmoviles.com/ref/web60>
- [24] «Developer Economics: State of the Developer Nation Q3 2015». Vision Mobile (research series). pp. 16-19.
- [25] R. Brauer, «How to use icon fonts in your mobile apps, Creative Bloq, 19-feb-2014. [Disponible en línea] <http://interfacesmoviles.com/ref/web61>
- [26] «User Interface», Android Developers. [Disponible en línea] <http://interfacesmoviles.com/ref/web62>
- [27] «UIKit User Interface Catalog: About Views». Apple. [Disponible en línea] <http://interfacesmoviles.com/ref/web63>
- [28] «Lista de controles (XAML) (Windows)», Microsoft Developer Network. [Disponible en línea] <http://interfacesmoviles.com/ref/web64>

5

IDENTIDAD Y PROMOCIÓN

En este capítulo se abordan algunos temas importantes sobre la identidad de marca en aplicaciones móviles, así como su promoción en diferentes canales. Se hace un recorrido por los iconos de aplicación, el uso del color a través del círculo cromático, tipografías, guías de estilo y canales publicitarios comunes para realizar campañas en medios digitales. También se destina un espacio para hablar de la publicidad tradicional como estrategia de posicionamiento y promoción indirecta de aplicaciones móviles.

Creando una identidad

La creación de una marca va más allá del propio producto o servicio, acercándonos a su identificación a través de unos valores y una personalidad propia, por eso, resuelto el aspecto funcional, primera preocupación de una aplicación móvil, podemos comenzar la creación de una identidad sólida que ayude a reforzar la imagen del producto con respecto a las demás aplicaciones. Esta concepción está muy ligada a los aspectos de mercadeo de un producto y también a los significados que trata de generar, pero el concepto de marca es difuso y a menudo tiende a confundirse con la creación de logotipos, sin embargo, su estudio abarca mucho más en el proceso de construcción de una experiencia global, reconocible en la mente del usuario a través de todos los canales de comunicación de una empresa, servicio o producto [1].

Un ejemplo claro podemos encontrarlo en la industria del lujo, donde una marca como Ferrari no representa solo una empresa de autos, a través de sus productos, su diseño y vanguardia le imprime a sus automóviles una personalidad de élite y sofisticación. Por su parte, una empresa como Lamborghini refuerza otros valores en sus vehículos con un espíritu más agresivo y poderoso, solo basta con observar los dos símbolos de sus marcas (el *cavallino* rampante y el toro indomable) que representan una influencia clara en sus diseños y tradición, a través de una identidad de marca con un enfoque casi mitológico que va más allá de la funcionalidad de sus productos, pues está claro que las dos marcas fabrican automóviles que cumplen una misma función, pero cada una le imprime su personalidad, estilo de vida y valores subjetivos que transmiten a sus clientes un sentido de pertenencia y exclusividad.

Lo mismo sucede con la tecnología y el software, por ejemplo Apple ha construido una marca de lujo cercano, abriendo nuevos mercados con sus equipos y aplicaciones desde hace décadas [2]. Sea real o imaginaria la historia detrás de la manzana mordida como un símbolo de la tentación de Eva [3], es claro que su influencia va más allá de la venta de teléfonos móviles y ordenadores, ofreciendo una experiencia global en sus productos a través del diseño como elemento de diferenciación. Un tema extenso que abarca mucho más de lo que puede ser la interfaz de usuario, sin embargo, afecta todos los aspectos de una aplicación a través de unos valores estandarizados de diseño y estilo característico a través del uso del color, la tipografía, iconos y sus lineamientos generales de comunicación para diferenciarse de la competencia.

Imagen de marca en el software

Como se ha comentado a lo largo del libro, la interfaz gráfica es el punto de comunicación entre el usuario y la aplicación, pero como unidad aislada no podría llegar a ser exitosa si no se incluye dentro de una estrategia global de marca. Como categoría propia, el software es un producto que requiere del mercadeo para poder distribuirse, sin importar que sea de carácter privativo, gratuito o libre, requiere de campañas que lo den a conocer y, antes de la ejecución de estas, debe tener una imagen de marca coherente y poderosa que transmita confianza a los actuales y futuros usuarios. El software desde hace mucho que dejó de ser tosco y poco atractivo, ahora importa hasta el más mínimo detalle de lo que se muestra al usuario.

Una aplicación puede ser efectiva, intuitiva y clara cumpliendo su promesa básica, algo que se evalúa con las pruebas funcionales y de usabilidad, pero si carece de una identidad de marca fuerte y definida, puede ser reemplazada por los usuarios sin ningún remordimiento. Esto suele ocurrir con otros productos físicos que se conocen en inglés con la expresión *commodities*, un producto intercambiable que no se diferencia de los demás, fenómeno mucho más común en el hardware donde la similitud entre diversos teléfonos los convierte en bienes genéricos, sin embargo, en el software también se ve este comportamiento [4], que cuando se trata de resaltar sobre las miles de aplicaciones disponibles, tener una identidad propia es un reto para el proyecto y su diferenciación con respecto a la competencia.

Cuando se construye una imagen de marca para una aplicación móvil es necesario definir si se va a crear una identidad desde cero o generar una extensión de una marca posicionada. Esta última opción, en creciente auge para las grandes empresas [5], combina publicidad y a la vez herramientas útiles para el usuario. Marcas como Nike que con su aplicación Nike+ ha tenido más de 28 millones de descargas son un ejemplo claro sobre las aplicaciones que pueden tener objetivos publicitarios y a la vez acumular valor ofreciendo soluciones que fidelizan a los usuarios [6]. Este valor percibido solo se construye con el tiempo, pero si algo es claro, es que antes de existir una marca debe existir un producto que la soporte y que permita construir a su alrededor el imaginario de ideas y valores que representa para las personas [7], esto es muy claro en algunas industrias pero también ocurre en el diseño de aplicaciones como herramientas cada vez más valoradas por los usuarios.

Iconos de aplicación

En un contexto de uso rápido donde no tenemos tiempo de leer instrucciones, un icono distintivo es un elemento poderoso que tiene una aplicación para identificarla y diferenciarla sobre otras opciones en el teléfono o en las tiendas de aplicaciones [8]. Cuando vemos la pantalla de nuestro móvil no leemos, solo escaneamos e identificamos iconos para elegir la aplicación deseada. Por esta razón y a diferencia de los demás iconos de interfaz, en estos se rompen las convenciones tradicionales y se impone la originalidad del diseño.

Los iconos se han utilizado en los sistemas operativos desde los años setenta para representar acciones del sistema y paquetes de software como elementos recurrentes en las primeras interfaces gráficas [9]. Sin embargo, en la actualidad por la cantidad de aplicaciones, el diseño de iconos ha cobrado una mayor importancia como recurso equivalente al logo en el diseño de productos.

En ese sentido, un logo tradicional se diseña de manera libre con dos componentes básicos: el símbolo y el logotipo, el símbolo es la figura característica, abstracta o concreta de una identidad, mientras que el logotipo, en sentido literal, es el nombre tipográfico de la marca. En el uso cotidiano se le llama logo a la unión de estos dos elementos combinados, no obstante, algunos logos carecen de símbolo, como la identidad de Sony e IBM. Otros por el contrario son iconográficos, un ejemplo claro es la identidad de Apple, que representa la conocida manzana mordida sin necesidad de utilizar la palabra Apple [10]. Los iconos de aplicación se acercan más a esta última clase de logo sin tipografía pero tienen unas limitaciones específicas en su proporción: se diseñan sobre una base cuadrada con tamaños definidos por cada sistema operativo.

El modo más común de utilizarlos se da en las plataformas que emplean una versión de la metáfora del escritorio donde las aplicaciones se representan agrupadas en pantalla sobre una matriz de opciones. En iOS y Ubuntu el sistema establece para los iconos de aplicación una máscara cuadrada con bordes curvos. Por su parte, Windows Phone genera una pantalla principal geométrica que representa los iconos en forma de mosaicos de base cuadrada. Tizen y WatchOS emplean por lo general iconos circulares y Android permite el uso de siluetas libres para cada icono, pueden ser redondos, cuadrados o presentar límites irregulares. Por último, Sailfish OS utiliza iconos de formas semi-circulares con algunos bordes afilados.

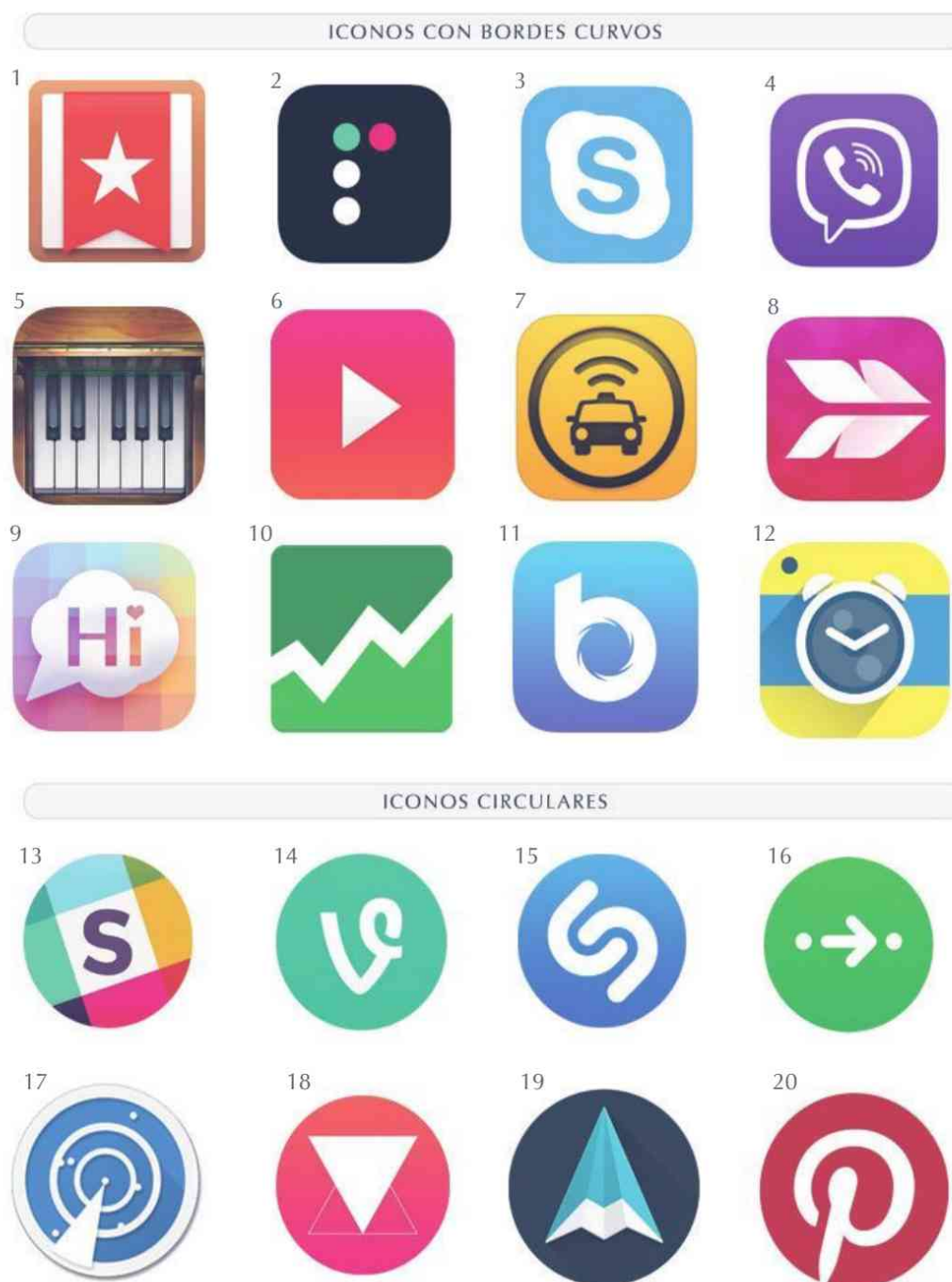


Figura 5-1 Ejemplos de iconos: 1- Wunderlist (iOS) 2- Pacemaker (iOS) 3- Skype (iOS) 4- Viber (iOS) 5- Piano Real (Android) 6- Skin for Poweramp (Android) 7- Easy Taxi (iOS) 8- Skitch (iOS) 9- Say-Hi! (Android) 10- Stock Watcher (Android) 11- App Booster (Android) 12- Alarmy (Android) 13- Slack (WatchOS) 14- Vine (WatchOS) 15- Shazam (WatchOS) 16- Citymapper (WatchOS) 17- Flight Radar Pro (Android) 18- Lidow Layout (Android) 19- AutoMate Car D. (Android) 20- Pinterest (Android).

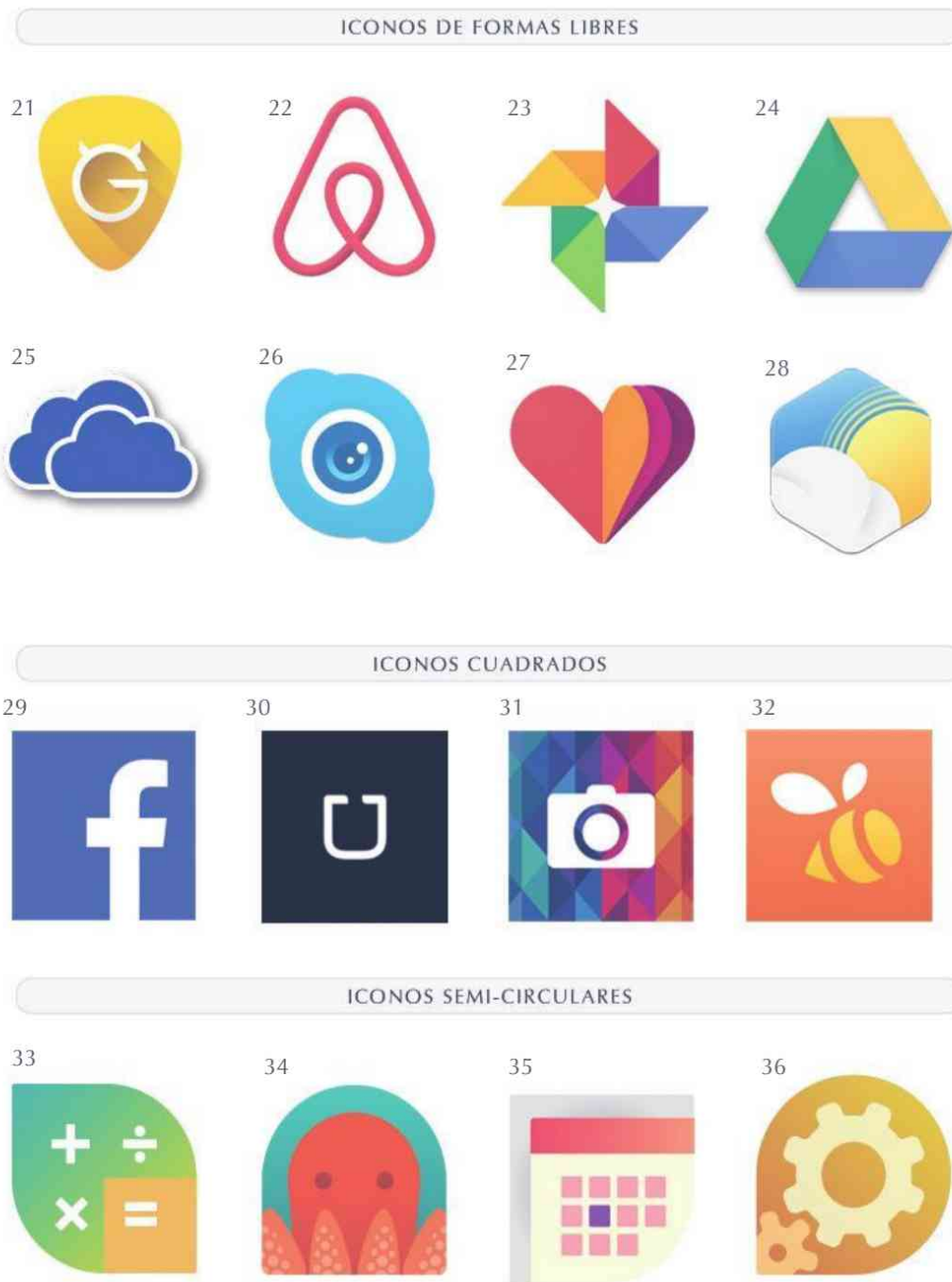


Figura 5-2 Ejemplos de iconos: 21- Ultimate Guitar Tabs Chords (Android) 22- Airbnb (Android) 23- Google Photos (Android) 24- Google Drive (Android) 25- One Drive (Android) 26- Skype Camera (Android), 27- Google Fit (Android) 28- Amber Weather (Android) 29- Facebook (Windows Phone) 30- Uber (Windows Phone) 31- Phototastic (Windows Phone) 32- Swarm (Windows Phone) 33- Calculator (Sailfish OS) 34- Octoslush (Sailfish OS) 35- Calendar (Sailfish OS) 36- Settings (Sailfish OS).

Uso del color

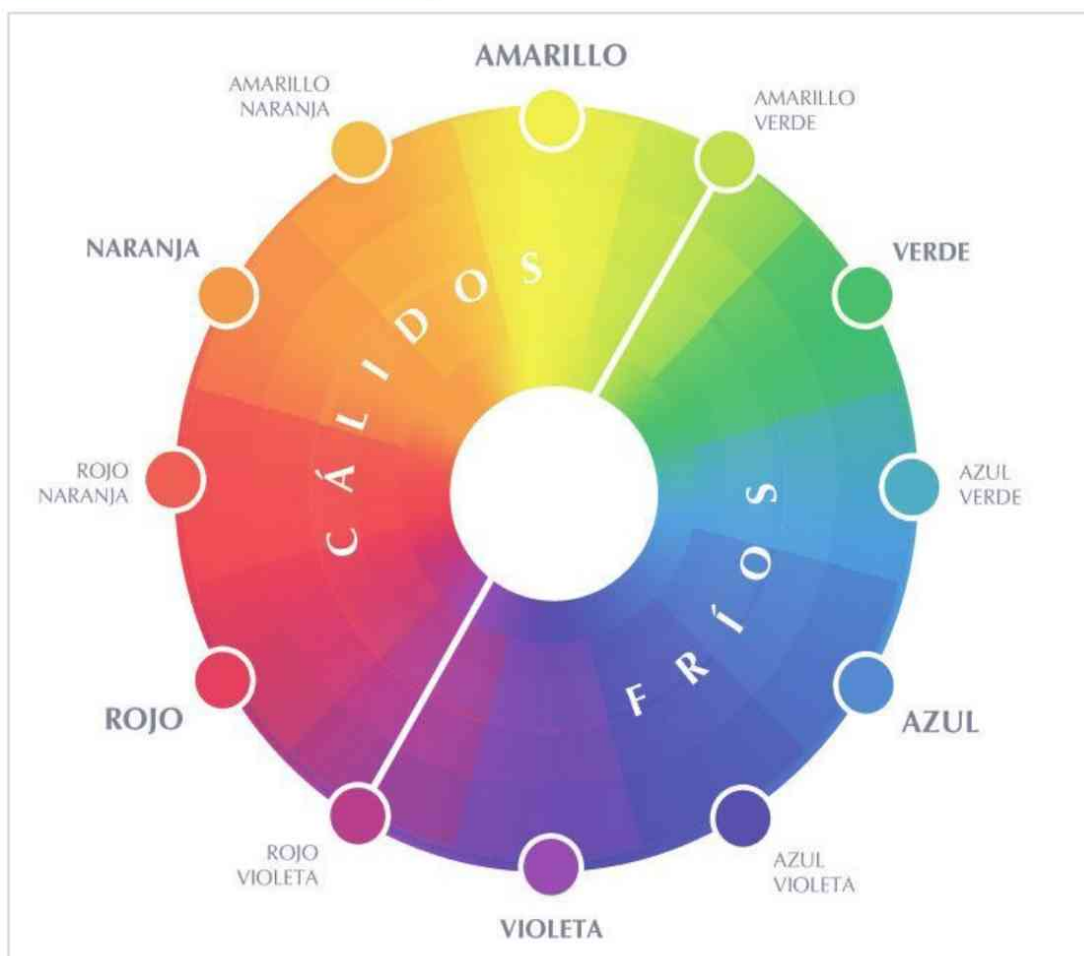
Elegir el color no siempre es sencillo, así como los iconos, las connotaciones que puede tener un cierto color varían entre culturas y personas. En las interfaces el ejemplo más evidente es el uso del color rojo como símbolo de alerta, error o peligro, sin embargo, existen aplicaciones que utilizan el color rojo sin remitirse o transmitir sensación de peligro. Por ejemplo Gmail y Path utilizan el rojo como color base de una manera clara. Esto nos indica que el significado del color es un asunto de interpretación que puede simbolizar algo dependiendo de la cultura o el contexto donde aparece dicho color [11]. Por otro lado, las limitaciones físicas en la percepción del color también generan problemas de accesibilidad ya que se estima que el 8% de los hombres sufren daltonismo, condición que les impide reconocer algunos colores primarios [12]. Por eso, utilizar el color como único elemento de instrucción en iconos o mensajes en la interfaz puede dificultar su comprensión en las personas con daltonismo [13]. Existen software gratuitos como Color Oracle o Coblis dedicados a exponer ejemplos de las diferentes variaciones de esta condición que pueden ser útiles como recursos de prueba y consulta [14].

Esta percepción subjetiva del color puede ser física o social, pero en muchas ocasiones si la aplicación que estamos diseñando hace parte de una empresa o marca existente, es probable que los colores no se puedan elegir a libertad pues ya están condicionados por los colores corporativos. Pero en otros casos, cuando tenemos libertad de elegir, el método de combinación del círculo cromático es un recurso útil para la elección de colores. El esquema más conocido es el de Isaac Newton [15] que ha tenido evoluciones a lo largo del tiempo y donde prevalece el círculo cromático de 12 tonos RYB (Red Yellow Blue) entre fríos y cálidos (Figura 5-3). También existen los modelos RGB (Red Green Blue) y CMYK (Cyan Magenta Yellow Black) con base más científica, sin embargo, como punto de partida en la combinación del color, el esquema RYB sigue siendo un recurso práctico para diseñadores y artistas (Tabla 5-1).

En internet existen numerosas herramientas que ayudan a combinar los colores como Paletton, Color Calculator y Adobe Color que tienen definidas cientos de paletas como inspiración para nuestros diseños y también permiten crear paletas personalizadas empleando armonías de color: monocromáticas, análogas, complementarias, triádicas. Estas armonías son útiles para comprender el uso del color en el círculo cromático realizando ejercicios de combinación [16].

Tabla 5-1 Tipos de colores del círculo cromático natural RYB (Red Yellow Blue)

Colores primarios	Son los colores base desde los cuales se crean las combinaciones de los demás tonos, estos son: el azul, el rojo y el amarillo.
Colores secundarios	Son las primeras combinaciones de color de los tres colores primarios, en estos encontramos el naranja (rojo + amarillo), el verde (azul + amarillo) y el violeta (rojo + azul).
Colores terciarios	Se conocen como terciarios al resultado de la mezcla de un color secundario, por ejemplo el violeta, y un color primario como el azul.

**Figura 5-3** El círculo cromático natural RYB es un recurso útil y didáctico para definir paletas de colores.

Paletas monocromáticas

En este tipo de armonía se utiliza un único color que representa la base principal. Sus combinaciones se realizan por medio de variaciones de ese mismo tono elegido. Las paletas monocromáticas suelen tener cierta monotonía porque el uso del color es bastante limitado. Sin embargo, con ingenio pueden utilizarse de manera creativa.

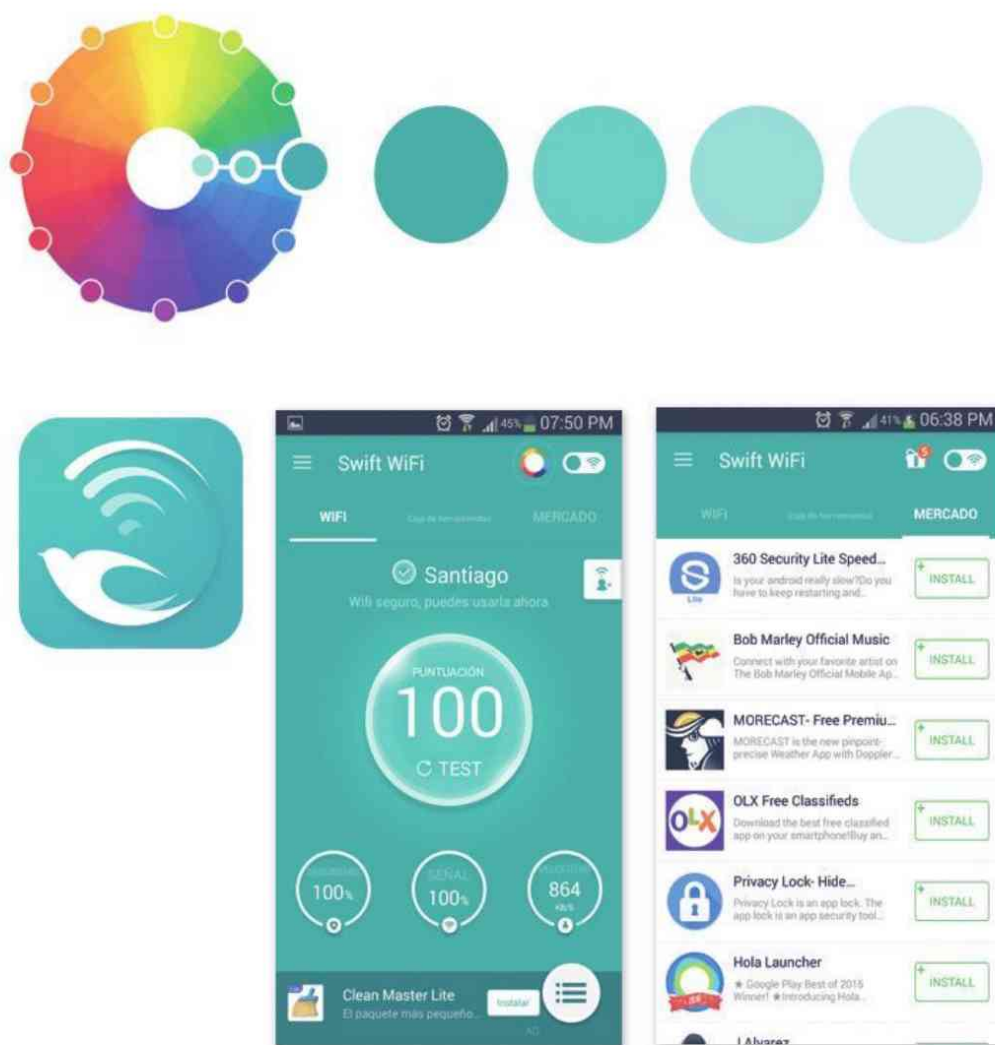


Figura 5-4 La aplicación Swift WI-FI (Android) utiliza una paleta monocromática de varios tonos aguamarina como base para los colores de su interfaz y su icono de aplicación.

Paletas complementarias

Se trata de combinaciones de los colores opuestos en el círculo cromático. Estas paletas tienen una base interesante entre juego de colores cálidos y fríos, ya que un color como el verde puede tener su contraparte roja que la complementa en armonía. Estas paletas ofrecen una combinación interesante para elegir los colores de la aplicación.

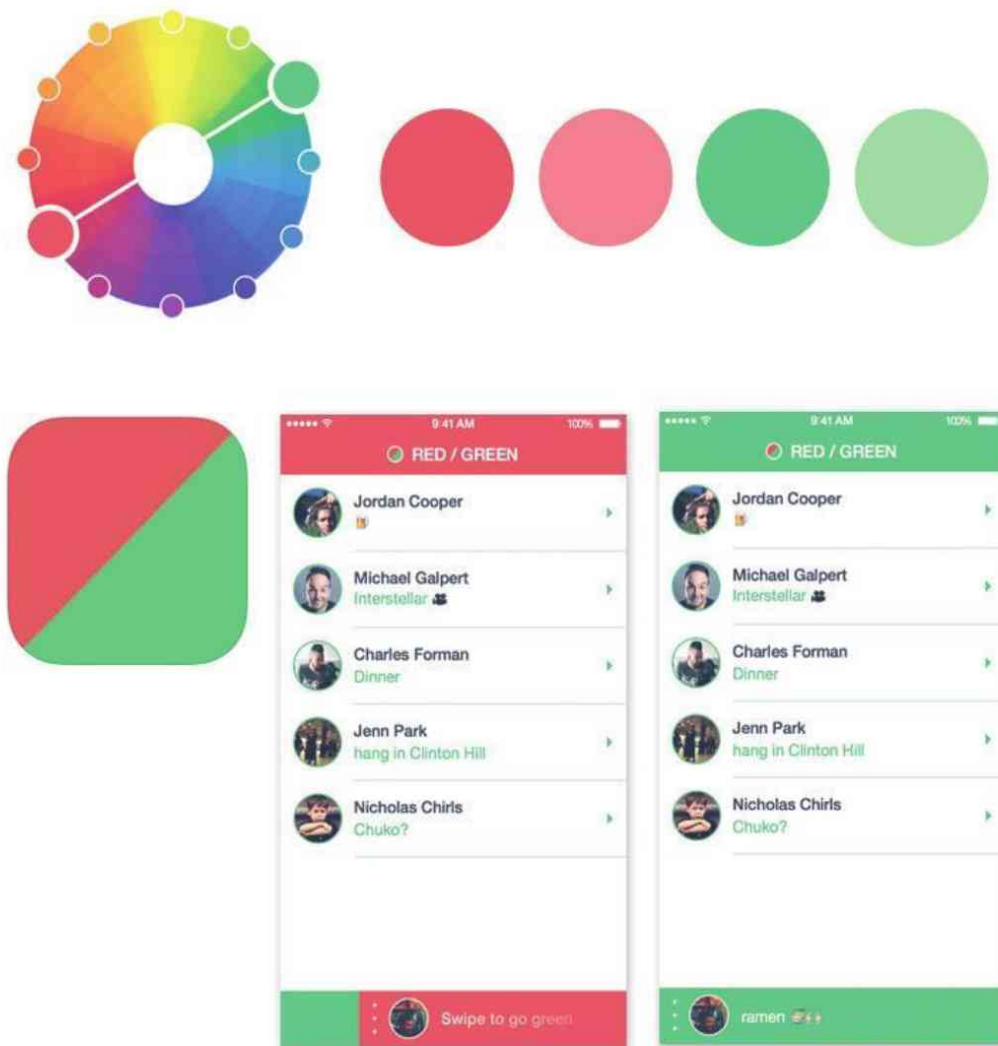


Figura 5-5 La aplicación Red / Green (iPhone) utiliza una paleta complementaria donde los colores protagonistas son el verde y el rojo utilizados en el icono de aplicación y en las barras superiores e inferiores.

Paletas análogas

En estas paletas se eligen los colores que están ubicados uno al lado del otro en el círculo cromático, pueden ser en variaciones de tono pero su característica principal es que se encuentran juntos en la escala de color. Tonos cálidos que se combinan como los verdes, amarillos y naranjas son muy utilizadas en diseño de interfaz, así como los tonos fríos como el violeta y azul.



Figura 5-6 La aplicación Yahoo! Mail (iPhone) en su nuevo diseño combina colores en una paleta análoga donde se mezclan el azul y el violeta, combinación visible en su menú lateral, botones y barras de opciones.

Paletas triádicas

Utilizan un esquema armónico que se compone de una combinación de tres tonos de colores del círculo cromático que se encuentran en una distancia equivalente, como los tres colores primarios o los tres colores secundarios. Estas paletas se ubican en este esquema de combinaciones específico.

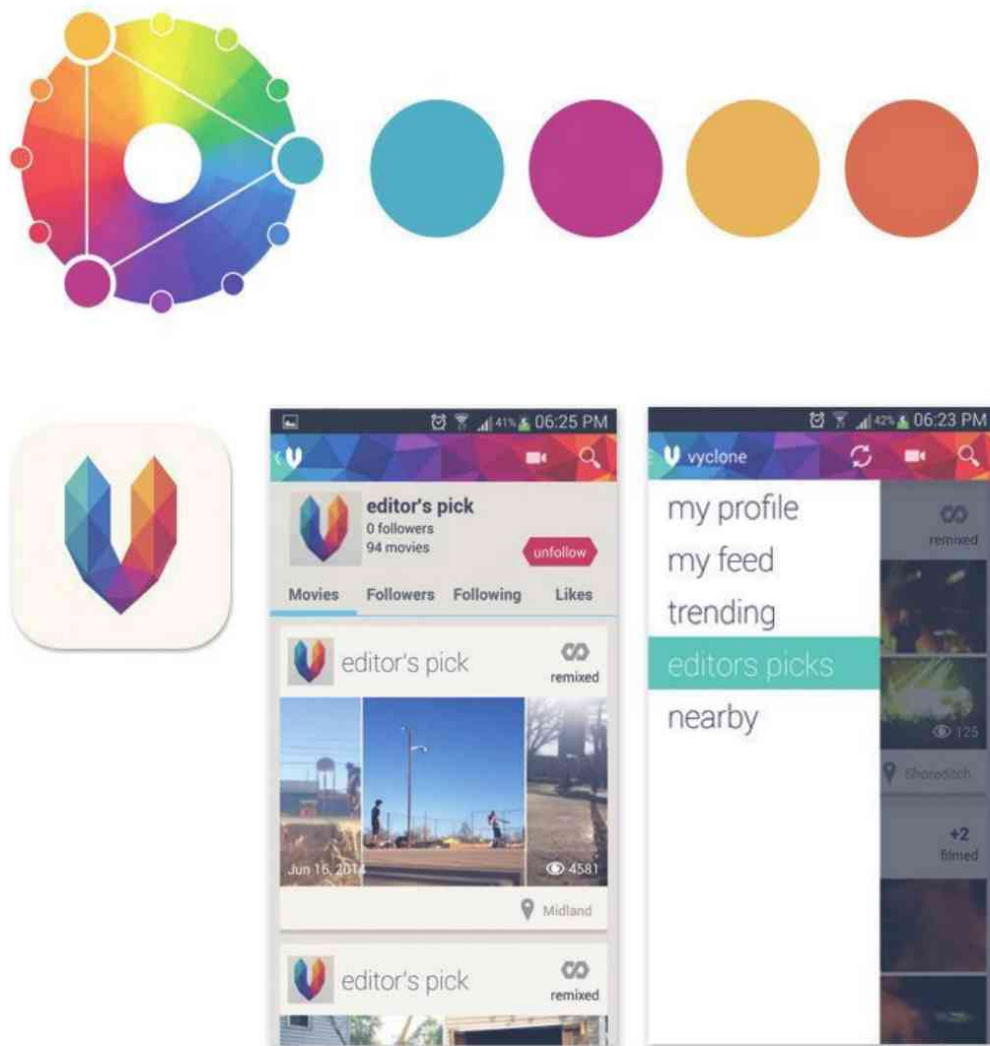


Figura 5-7 La aplicación Vyclone (Android) tiene en su imagen e identidad una paleta que mezcla diferentes tonos de azul, amarillo y violeta. Su interfaz utiliza una paleta triádica.

Tipografía

Al igual que el color, la tipografía puede reforzar la identidad como elemento característico que genera recordación e imprime una personalidad. El estudio de la tipografía es extenso y se han escrito innumerables libros dedicados en exclusiva a este tema. Sin embargo, un primer acercamiento básico lo ofrece la clasificación tipográfica de Francis Thibaudeau propuesta a comienzos del siglo XX y que las agrupa según su estilo general en tipos *serif*, *sans-serif* y decorativas [17]. En la Tabla 5-2 describimos esta clasificación, sin desconocer que existen otros modelos posteriores más detallados como el de Maximilien Vox que posee 11 grupos de familias tipográficas en diferentes estilos asociados a su origen histórico [18].

Tabla 5-2 Clasificación básica de tipografías basada en el modelo de Francis Thibaudeau

<i>Serif</i>	Conocidas como romanas y egipcias, poseen formas adornadas en los extremos de cada caracter, estas terminaciones se llaman remates y se utilizan con más regularidad en medios impresos.
<i>Sans-serif</i>	Tienen un estilo lineal y sin remates. En la clasificación de Thibaudeau se conocen como antiguas. Los textos en pantallas digitales utilizan en forma habitual tipos <i>sans-serif</i> .
Decorativas	Exploran con libertad, formas de caracteres y estilos a mano alzada, artísticos o de la cultura popular como textos deteriorados y estéticas ornamentales, modernas tipo graffiti o futuristas.

Los sistemas operativos móviles, al igual que ocurre con los sistemas de escritorio utilizan por defecto tipografías *sans-serif*: Windows Phone maneja Segoe UI que da protagonismo al texto como elemento de la interfaz [19]. Android incluye Roboto desde la versión 4.0 (Ice Cream Sandwich) y Noto, esta última muy utilizada en lenguas asiáticas (coreano, chino, japonés, tailandés, entre otras) [20]. Por su parte Apple, que utilizó durante mucho tiempo la Neue Helvetica, presentó en 2015 su nueva tipografía llamada San Francisco [21]. Estas tipografías oficiales son las más comunes, sin embargo, las diversas plataformas no limitan a los diseñadores para utilizar otros estilos: Windows y iOS tienen disponibles variadas tipografías *serif* y *sans-serif*. En Android solo están disponibles sus estilos por defecto, incluir otros es posible pero requiere un trabajo adicional instalando las fuentes específicas en cada aplicación.

Perpetua Times Georgia Baskerville Rockwell

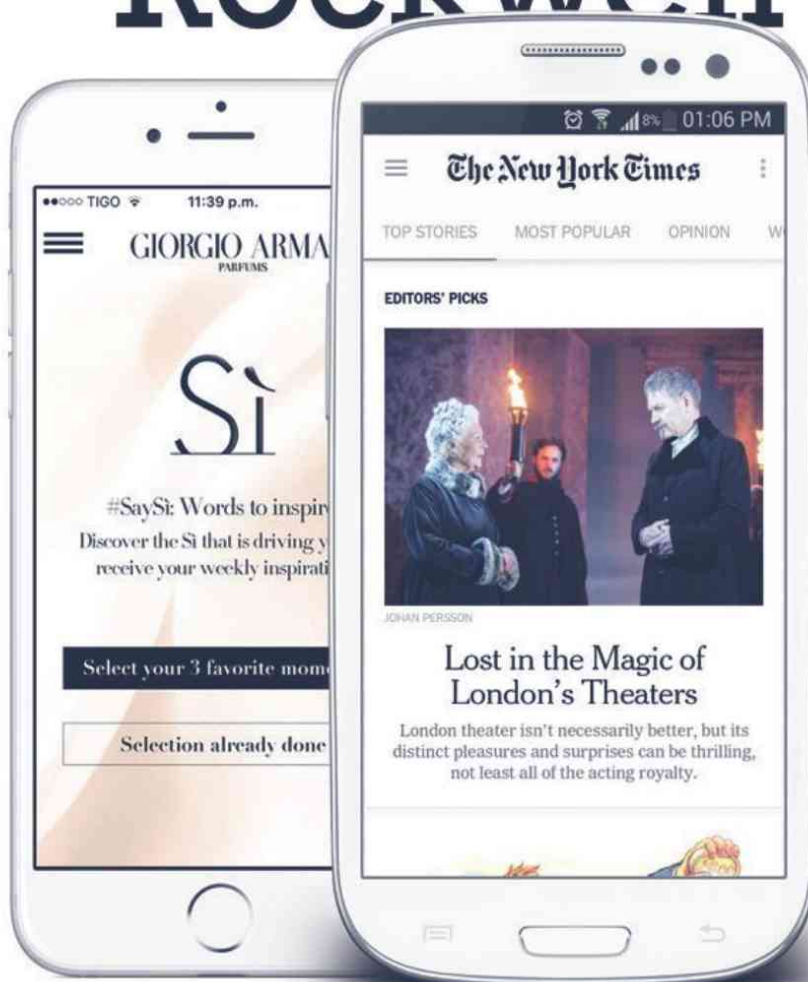


Figura 5-8 Las tipografías *serif* tienen adornos en los bordes conocidos como remates y son menos utilizadas que las *sans-serif* en pantallas digitales. Un ejemplo de tipografía *serif* se puede observar en las aplicaciones Giorgio Armani Beauty (iPhone) y The New York Times (Android), esta última utiliza las tipografías Cheltenham y Georgia en sus artículos conservando su relación con el medio impreso tradicional.

Century Gothic
San Francisco
Roboto
Segoe Armata
Gill Sans

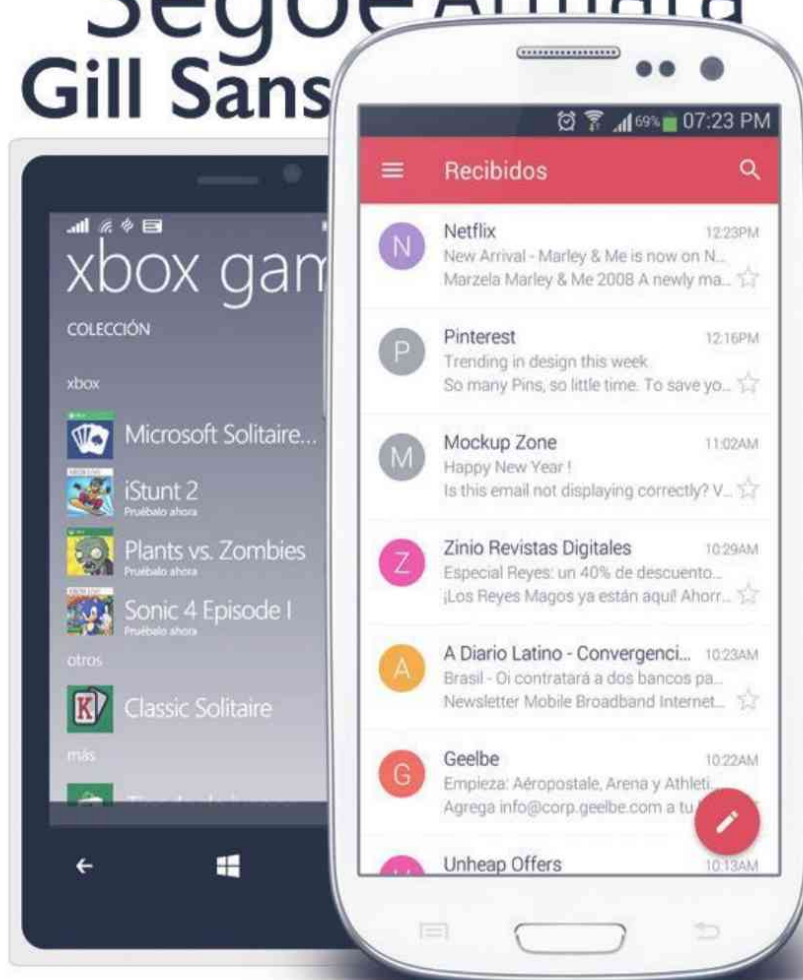


Figura 5-9 Las tipografías sans-serif son las más utilizadas en pantallas digitales, son limpias y no poseen remates en sus terminaciones. Un ejemplo se puede observar en las aplicaciones móviles Xbox games (Windows Phone) que utiliza la tipografía Segoe UI de Microsoft y Gmail (Android) que utiliza la tipografía Roboto creada por Google para todos los textos de sus aplicaciones y software en general.



Figura 5-10 Las tipografías decorativas tienen estilos radicales y dada su poca legibilidad en bloques de texto, no se utilizan mucho en el diseño de interfaces. Sin embargo, son muy empleadas en aplicaciones móviles como Typographer (iPhone) y PicLab (Android) que permiten a los usuarios editar fotografías utilizando todo tipo de estilos de texto, entre ellos gran cantidad de tipografías decorativas.

Guías de estilo

En las guías de estilo o manuales de identidad se recopilan una serie de normas para el uso correcto de una marca. En ellas se especifican los logos e iconos en sus diferentes versiones, tipografías y paletas de colores. También se suelen incluir recomendaciones sobre el uso del lenguaje escrito y cómo se dirige la marca a los usuarios. En compañías de gran tamaño, que poseen oficinas en diferentes ciudades, la estandarización de su marca es algo obligatorio para asegurar una consistencia global en su estrategia de comunicación [22].

En una guía también se pueden incluir los elementos de interfaz como un activo estratégico para reutilizar el diseño en diferentes pantallas y soportes, ya que si estos elementos no se estandarizan, es complicado escalar un diseño y coordinar el trabajo con otros diseñadores y desarrolladores [23]. Como vimos en el Capítulo 1, empresas de tecnología como Google y Microsoft están haciendo un gran esfuerzo por unificar su imagen de marca, no solo desde las piezas corporativas sino también desde el uso de las interfaces. Una respuesta concreta han sido los lenguajes de diseño utilizados en los diferentes sistemas operativos como Material Design y Modern UI que han servido de referencia y principios para crear sus propias guías de estilo oficiales.

Aunque una práctica habitual consiste en crear la guía de estilo en PDF, dichos documentos resultan difíciles de mantener y en pocos meses se vuelven obsoletos. Además, requieren de mucho tiempo invertido para un manual de difícil acceso porque suele enviarse por correo electrónico o estar disponible en forma estática [24]. Por esta razón, en proyectos de software, una alternativa en auge son las llamadas *living styleguides*, o guías de estilo dinámicas que se publican en internet en formato de sitio web y se actualizan cada vez que hay cambios o se agregan nuevos recursos. Este tipo de guías suelen ser compactas y de lectura rápida para tener un acceso inmediato a sus contenidos y recomendaciones (Figura 5-11).

En un entorno de desarrollo volátil este tipo de guías resultan muy efectivas y se convierten en el mayor centro de gestión del conocimiento donde el equipo de diseño puede documentar el uso de patrones de interfaz, piezas de mercadeo, recursos y demás elementos de marca. Los contenidos a incluir en una guía de estilo dependerán de la aplicación o el alcance del proyecto pero en todo caso su empleo regular es una buena práctica en el proceso de diseño.

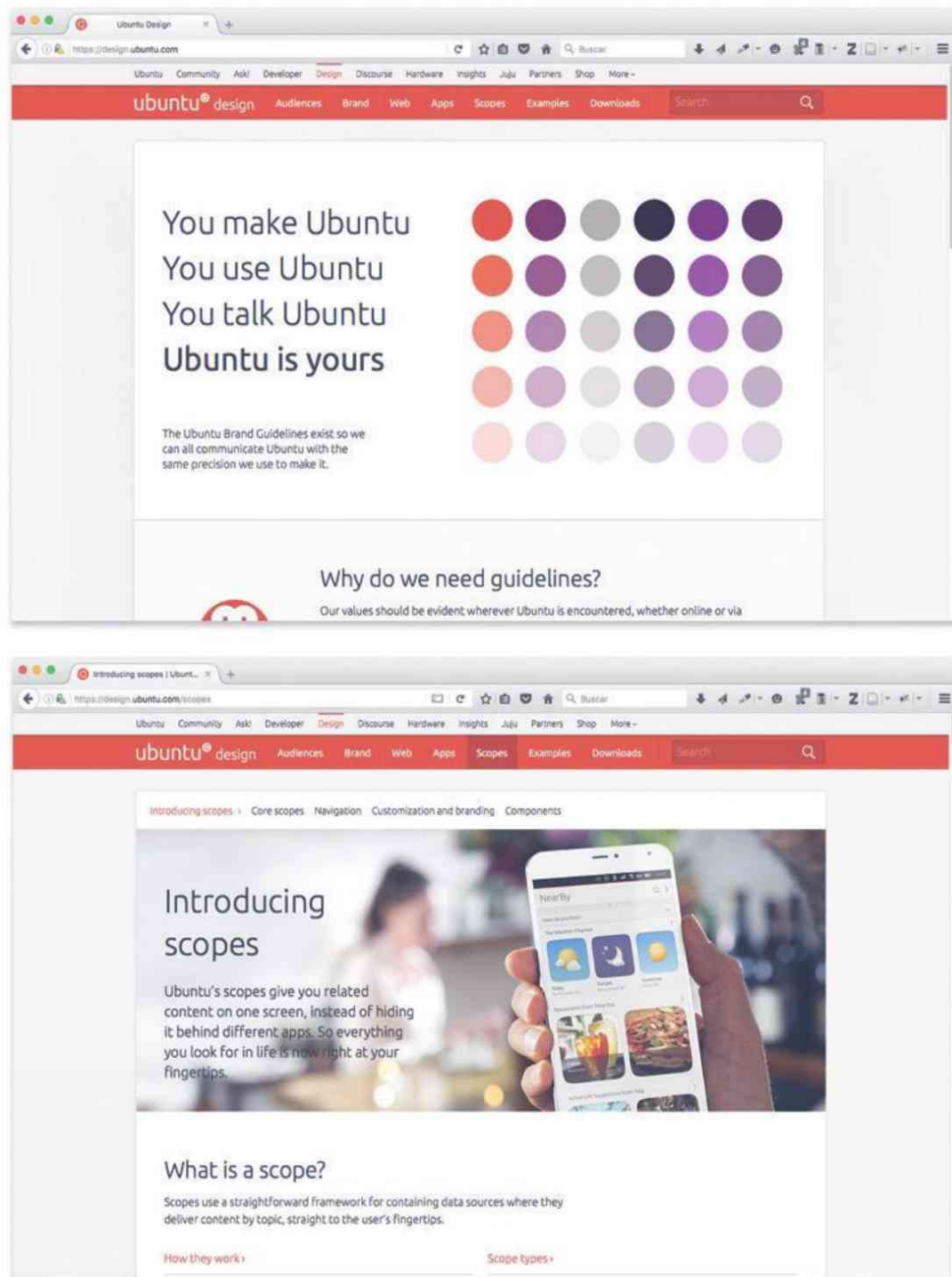


Figura 5-11 En las guías de estilo se pueden incluir librerías de patrones y definición de elementos básicos de interfaz con sus respectivos colores y lineamientos. En el ejemplo, la guía de estilo que ofrece el sistema operativo Ubuntu Linux donde se estandariza el aspecto visual y su estilo general de comunicaciones.

Canales de promoción

Mercados de aplicaciones

Las tiendas de aplicaciones se han convertido en las principales vitrinas de exposición del software para dispositivos móviles. La tendencia a unificar en ecosistemas más o menos cerrados ha hecho que se promocionen en mercados oficiales de los sistemas operativos una gran multitud de aplicaciones.

Desde la perspectiva de la promoción de una aplicación móvil, deben aprovecharse estas tiendas, no solo como canales de distribución sino también como canales de promoción donde cada plataforma tiene sus lineamientos y prácticas documentadas en sus sitios oficiales con recomendaciones de uso. De forma general Google Play [25], App Store [26] y Windows Store [27] permiten a los desarrolladores incluir con el paquete de aplicación (el que descargan e instalan los usuarios) imágenes o *banners* promocionales, así como textos para describir las características principales de la misma en diferentes idiomas, e incluso algún video publicitario. Estas posibilidades, aunque básicas, requieren cuidado y coordinación entre los equipos de desarrollo porque necesitan de claridad y concreción, recordemos que la mayoría de las veces las personas acceden desde sus dispositivos móviles y la decisión entre instalarla o no se define en el poco tiempo que el usuario le pueda dedicar a verificar la procedencia de la misma y a sentir una necesidad y conexión con el producto que va a comprar o descargar de manera gratuita.

Las tiendas de aplicaciones son herramientas que ayudan a posicionar la imagen de marca en los resultados de búsqueda que se presentan a los usuarios, donde estos pueden ubicar con facilidad una aplicación por medio de su icono distintivo. Los listados donde aparecen estos resultados presentan gran multitud de aplicaciones donde el usuario se guía por promedios de calificaciones, nombres comunes e imágenes que despierten su interés o recomendaciones de personas conocidas sobre la existencia de una determinada aplicación. También existen otras herramientas en línea como AppStorm y Mevvy que se dedican a reseñar y organizar aplicaciones disponibles en los mercados oficiales como Google Play y App Store, estas herramientas replican la imagen de marca presentada en los canales oficiales de distribución y promocionan aplicaciones en sus sitios web. Estos espacios de revisiones han aumentado en los últimos años presentando una curaduría de mucha utilidad para los usuarios.

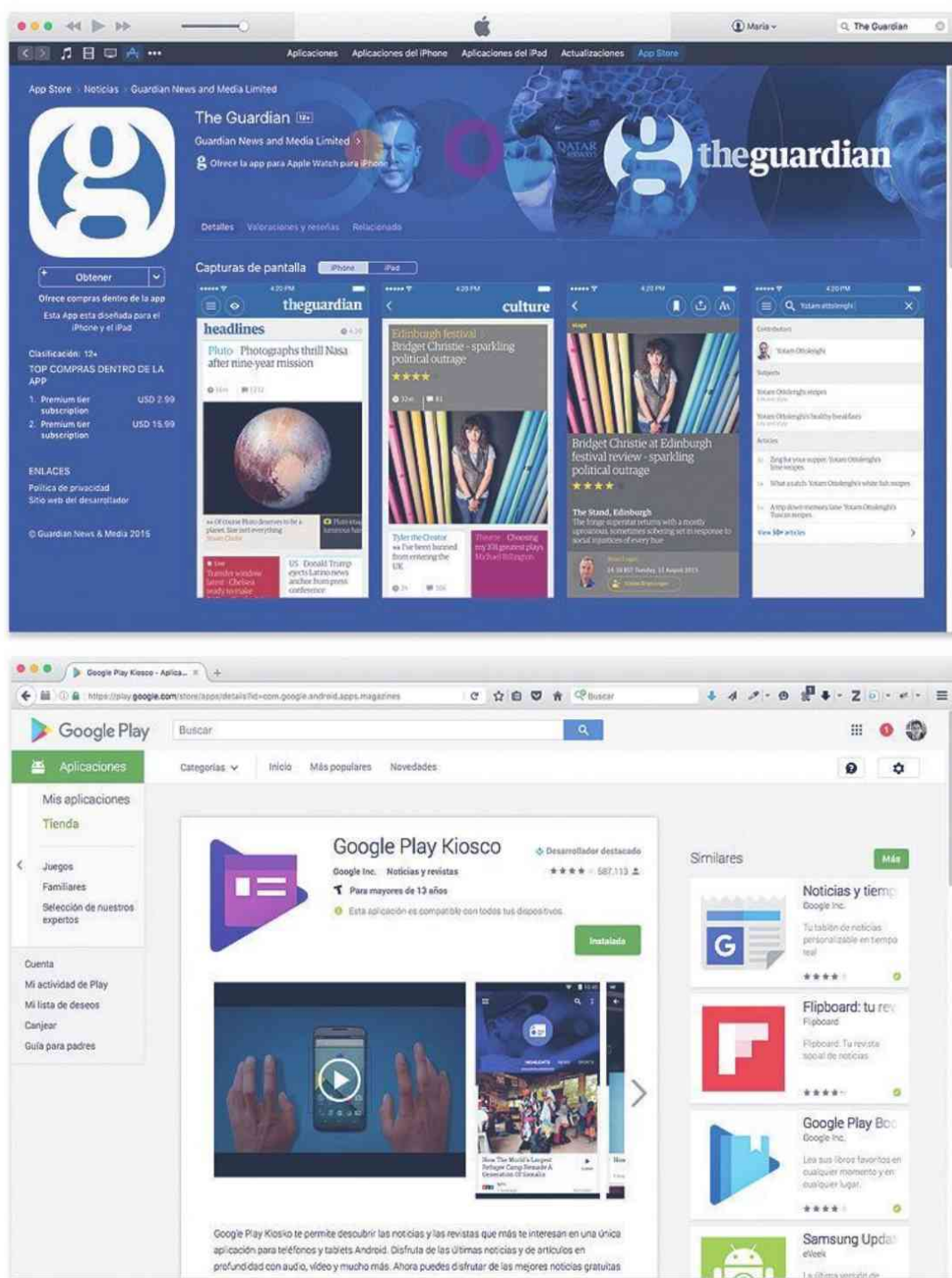


Figura 5-12 Arriba: App Store, tienda de aplicaciones de iOS accedida desde iTunes, se presenta la aplicación The Guardian con imágenes destacadas que refuerzan su identidad de marca. Abajo: la aplicación Google Play Kiosko desde Google Play, tienda de aplicaciones Android desde un navegador web.

Sitios web

Sea que la aplicación móvil haga parte de un paquete de software más extenso o que sea un producto único y enfocado sólo al contexto de la movilidad, debería tener, si es posible, un sitio web oficial donde los usuarios puedan ingresar y ampliar la información disponible en las tiendas de aplicaciones.

Muchas veces los usuarios tratarán de buscar mayor información sobre la empresa o el respaldo de una cierta aplicación, en especial cuando se trata de una versión de pago. En este tipo de situación es importante tener un sitio web donde se pueda suministrar la información pertinente del servicio, además de incluir espacios para contactar con los desarrolladores para solicitar algún tipo de soporte o reportar *bugs*. Así como realizar una donación (en el caso de las aplicaciones gratuitas que se financian con este tipo de estrategias).

El sitio web representa las instalaciones de una empresa digital y como es de esperarse, si no está organizado y carece de canales claros de comunicación, puede transmitir una imagen confusa al usuario y generar desconfianza. De igual forma la imagen corporativa desde el aspecto visual debe ser consistente con los demás espacios o piezas publicitarias presentes en otros canales, de esta manera el usuario sentirá que está ingresando sin duda al lugar correcto, lo que ayuda a posicionar la imagen de la marca y su percepción como un producto de software diferenciado y con un respaldo oficial.

Estos espacios se pueden presentar de diferentes formas, destacando en todo momento la aplicación móvil publicitada. A diferencia de las tiendas de aplicaciones que tienen capacidades reducidas para incluir elementos gráficos, los sitios web son ventanas mucho más amplias cuando se trata de fidelizar a los usuarios. Pueden ofrecer contenido de interés, material complementario o incluso motivar la descarga de otras aplicaciones móviles que hayamos desarrollado. Estos métodos pueden ayudar a promover los servicios de la compañía amparados en la calidad de los productos que desarrolla. Por lo general, un sitio de aplicación se realiza en páginas de aterrizaje, conocidas en inglés como *landing pages* donde se presenta la aplicación, los elementos que la identifican y la descripción básica de sus funcionalidades. En estos sitios se utiliza de forma clara un llamado a la acción en botones destacados que invitan a descargar la aplicación móvil desde los mercados en donde se encuentra disponible (Google Play, App Store y otros).

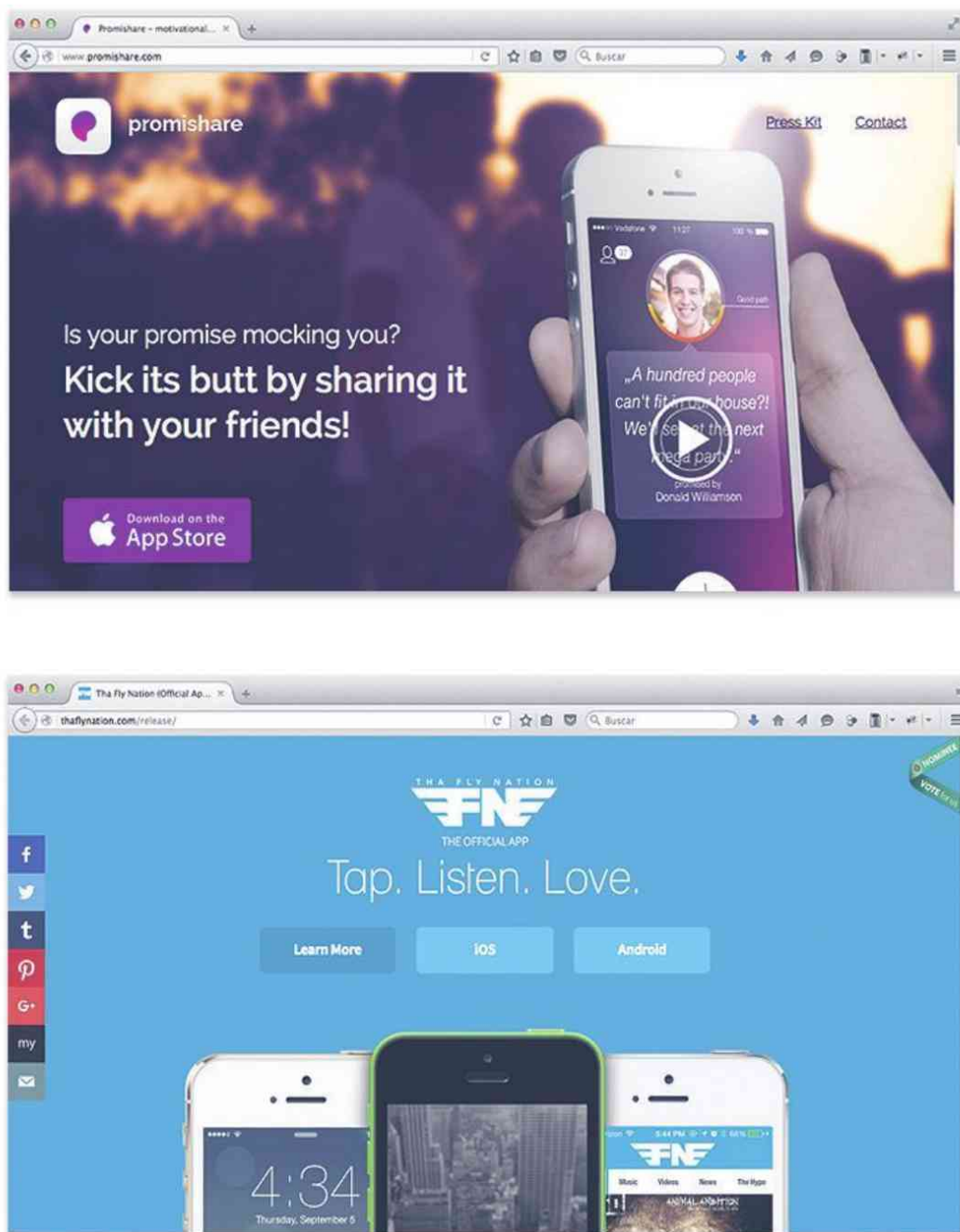


Figura 5-13 Los sitios web son canales de promoción importantes y se utilizan como una estrategia complementaria en el proceso de comunicación y soporte con los usuarios. **Arriba:** Sitio web oficial de la aplicación Promishare. **Abajo:** Sitio web oficial de la aplicación del medio The Fly Nation.

Redes sociales

Las redes sociales han tenido un impacto significativo en Internet y se han convertido en una categoría independiente de sitios web donde la interacción social a través de escenarios virtuales es extensa y poderosa.

Los casos más contundentes son sin duda Facebook y Twitter, dos servicios que han transformado la forma en la que nos comunicamos día a día. Las empresas lo saben y se están acercando a estos espacios con mayor frecuencia para promocionar sus productos. Sin embargo, en estas campañas dirigidas a medios sociales, no basta solo con reproducir un comercial de televisión y esperar a que el usuario descargue nuestra aplicación, sino generar lazos de comunidad entre los usuarios y el proyecto (o producto) ya que las redes sociales son canales muy eficientes para establecer contacto con los potenciales clientes de un servicio, incluso más que los canales tradicionales de comunicación.

Incluir una página de Facebook y un perfil oficial en Twitter de nuestra aplicación que se gestione de forma oportuna y adecuada puede ayudar a fidelizar aún más a los usuarios. Es importante aclarar que la administración de estos perfiles requiere constancia y tacto, de nada sirve crear un espacio que va a estar desactualizado con respecto a otros canales de promoción. Se trata, en todo caso, de promover, escuchar e interactuar con nuestros seguidores aprovechando que este escenario permite comunicarse de manera efectiva.

En la gestión y estrategia de una página o perfil en redes sociales, vuelve a ser muy importante el cuidado de la comunicación escrita y visual. La persona encargada de administrar estos espacios deberá comunicar de igual forma como la empresa que promueve la aplicación. El lenguaje debe ser profesional y lógico de acuerdo a las características del servicio. Si nuestra aplicación se dirige a un público juvenil, infantil o a público más adulto la comunicación en todos los canales, incluidas las redes sociales, deberá ser acorde a lo que espera el público específico. Las guías de estilo son un recurso útil para definir el tono adecuado y estandarizar estas comunicaciones evitando el lenguaje publicitario que busca vender un producto o servicio de un modo poco natural y forzado. En este sentido, para comunicarse en redes sociales, como se mencionó en líneas anteriores, una estrategia más adecuada consiste en la creación de comunidad a través de actividades relacionadas con los beneficios que ofrecemos a nuestros usuarios en estos canales de promoción masiva [28].

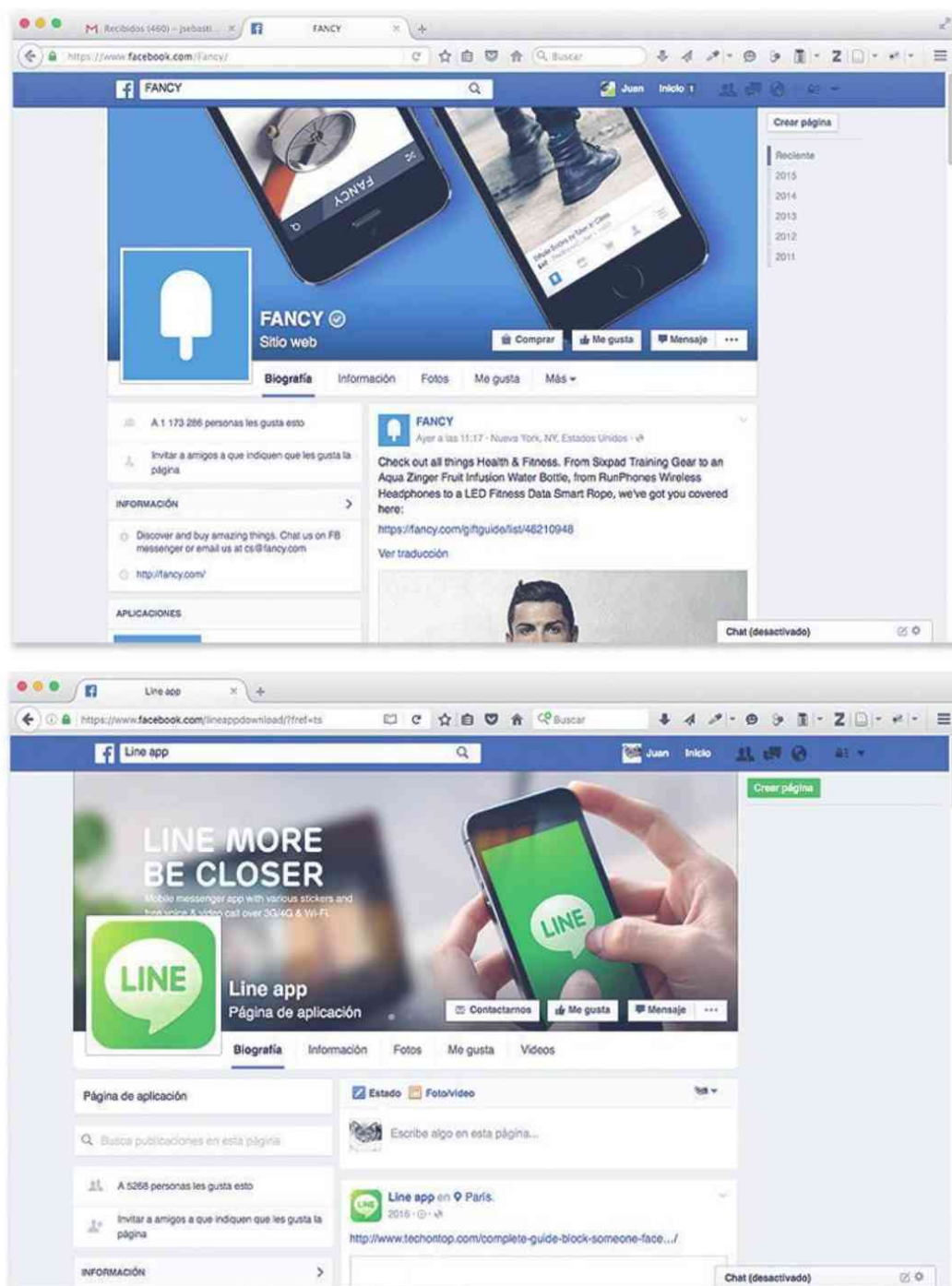


Figura 5-14 Las redes sociales son plataformas que permiten crear lazos de comunidad alrededor de aplicaciones y servicios móviles. **Arriba:** Página oficial de Facebook de la aplicación de compras social Fancy. **Abajo:** Página oficial de Facebook de la aplicación móvil de chat Line.

Medios tradicionales

Si bien los escenarios más utilizados para promocionar los productos de software son los medios digitales, en los últimos años las aplicaciones móviles se han vuelto tan populares que es común encontrar empresas que invierten grandes presupuestos para campañas en medios tradicionales como la televisión y la publicidad exterior.

Televisión y video

La televisión como se conoce en la actualidad sigue siendo un medio de comunicación unidireccional que no permite la interacción con los contenidos presentados, esto no ha cambiado en muchos países del mundo pese a la adopción de la televisión digital y los beneficios que ésta trae. Sin embargo, se espera que dentro de poco esta realidad cambie con la adopción masiva de televisores inteligentes que se conectan a internet.

Por lo pronto, la herramienta más utilizada en este escenario es sin duda la pauta publicitaria por medio de material audiovisual que durante años ha sido el estándar de promoción en este medio. El costo de una pauta en un canal reconocido con gran número de televidentes puede variar dependiendo de la hora en la que se transmita, pero en cualquiera de los casos será elevado. Esta situación impide que las empresas de desarrollo, en especial las *startups*, inviertan su presupuesto en este medio. Sin embargo, otra estrategia en aumento que pueden utilizar empresas pequeñas es la creación de videos explicativos como demostración de la promesa básica de la aplicación móvil [29]. Por lo general, este tipo de videos (al igual que en los comerciales televisivos) se utilizan como un llamado a la acción para dirigirse a otro medio como lo es el sitio web oficial, los mercados de aplicaciones, las páginas, perfiles de redes sociales o como herramienta complementaria a una aplicación donde se podrá acceder a contenido adicional para los usuarios.

Otra modalidad indirecta de promocionar aplicaciones en televisión se da a través de campañas que destacan servicios más generales como sucede con las cadenas y franquicias televisivas que promueven sus aplicaciones oficiales. De un modo similar, las operadoras de telefonía realizan campañas periódicas promocionando sus nuevos teléfonos inteligentes y servicios en donde presentan aplicaciones móviles conocidas por el usuario (Figura 5-15).

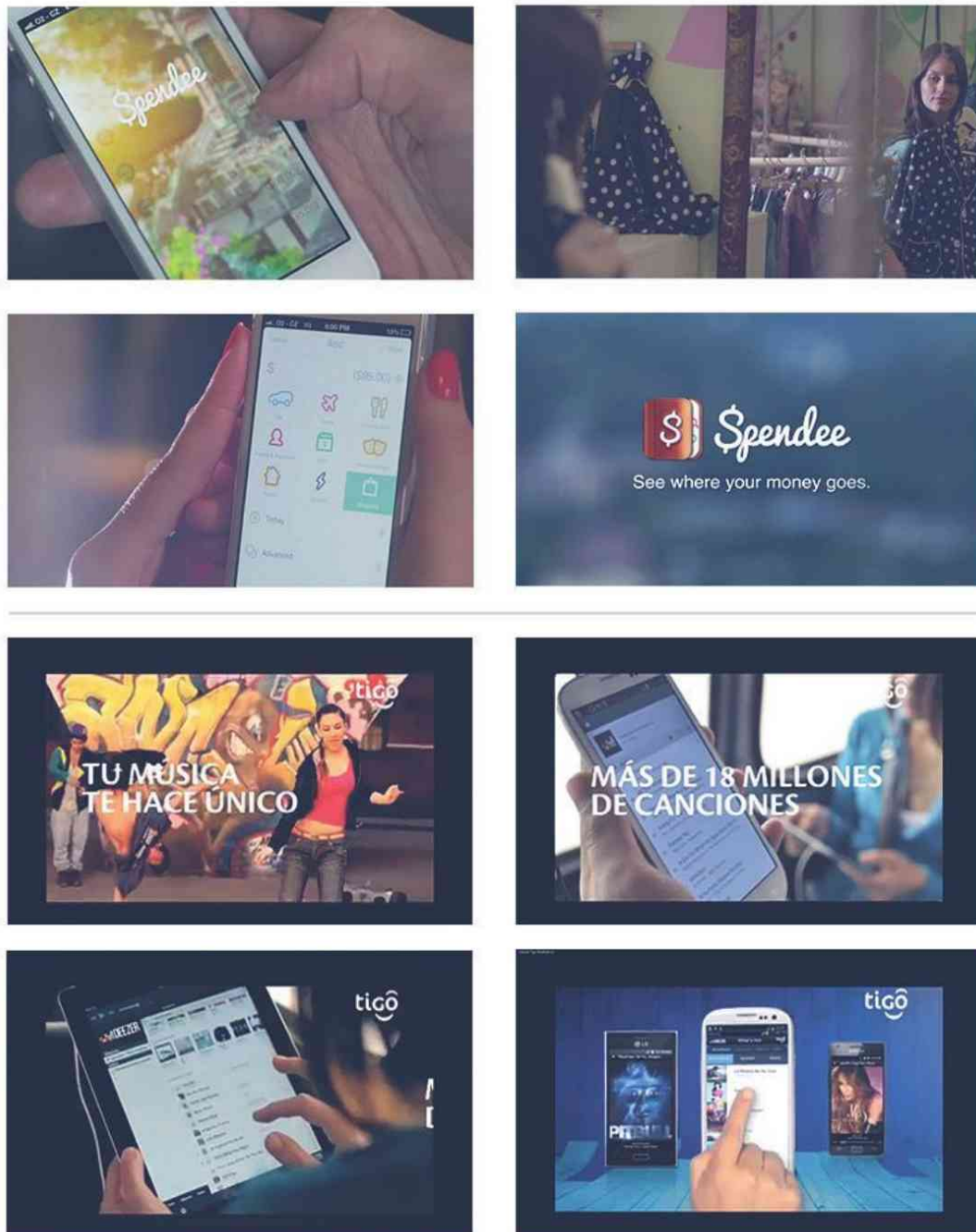


Figura 5-15 Arriba: Secuencia de video comercial de la aplicación Spendee (iPhone) para gestión de finanzas personales. Abajo: Secuencia de video comercial televisivo de la operadora de telefonía Tigo promocionando la aplicación móvil Deezer (Android) como parte de sus servicios y valor agregado.

Publicidad exterior

La publicidad exterior es otro de los medios que ofrece una tasa de exposición elevada y de igual forma tiene un precio alto. Las vallas publicitarias, afiches, carteles en paraderos de buses o publicidad ubicada en aeropuertos o centros comerciales se pueden utilizar para promover una aplicación. Esta modalidad ha sido adoptada poco a poco como un medio estratégico para promocionar servicios y productos móviles. Por lo general, este tipo de publicidad la utilizan grandes empresas como parte de su inversión publicitaria en diferentes lugares estratégicos en las ciudades.

Aunque no aplica para todas las situaciones, la publicidad exterior sigue siendo un medio masivo tradicional que puede ser explotado en espacios muy diversos y es una alternativa interesante, que en caso de tener el presupuesto y los recursos necesarios puede tenerse en cuenta.

Un escenario en el que por lo general se promociona una aplicación en publicidad exterior puede ser en conjunto con un producto como los teléfonos inteligentes y las tabletas. Los fabricantes invierten mucho dinero en promocionar terminales, y si a estos se les añade información sobre aplicaciones que vienen incluidas en estos terminales puede ser una estrategia acertada tanto para los desarrolladores como para los fabricantes. Sin embargo, acceder a estas alianzas se reserva a pocas empresas de desarrollo que gozan el reconocimiento unánime de los usuarios y por ende los fabricantes desean promocionarse junto a ellas para vender sus terminales. Un ejemplo de estas aplicaciones populares que logran motivar la compra serían juegos como Angry Birds o aplicaciones como Netflix, WhatsApp, Instagram y Facebook.

Al igual que la televisión, las campañas de publicidad exterior suelen presentarse como complemento a una campaña en otros medios y se enfocan en un llamado a la acción, como la compra de un producto, o en nuestro caso, la promoción de una aplicación móvil y su descarga a través de mercados de aplicaciones como Google Play y App Store.

Otro uso común se da en lugares públicos donde se ubica publicidad visible para motivar la descarga y dada la popularidad que están teniendo los teléfonos inteligentes, en casi cualquier sitio podemos encontrar una pieza de publicidad exterior promocionando una aplicación móvil (Figura 5-16).

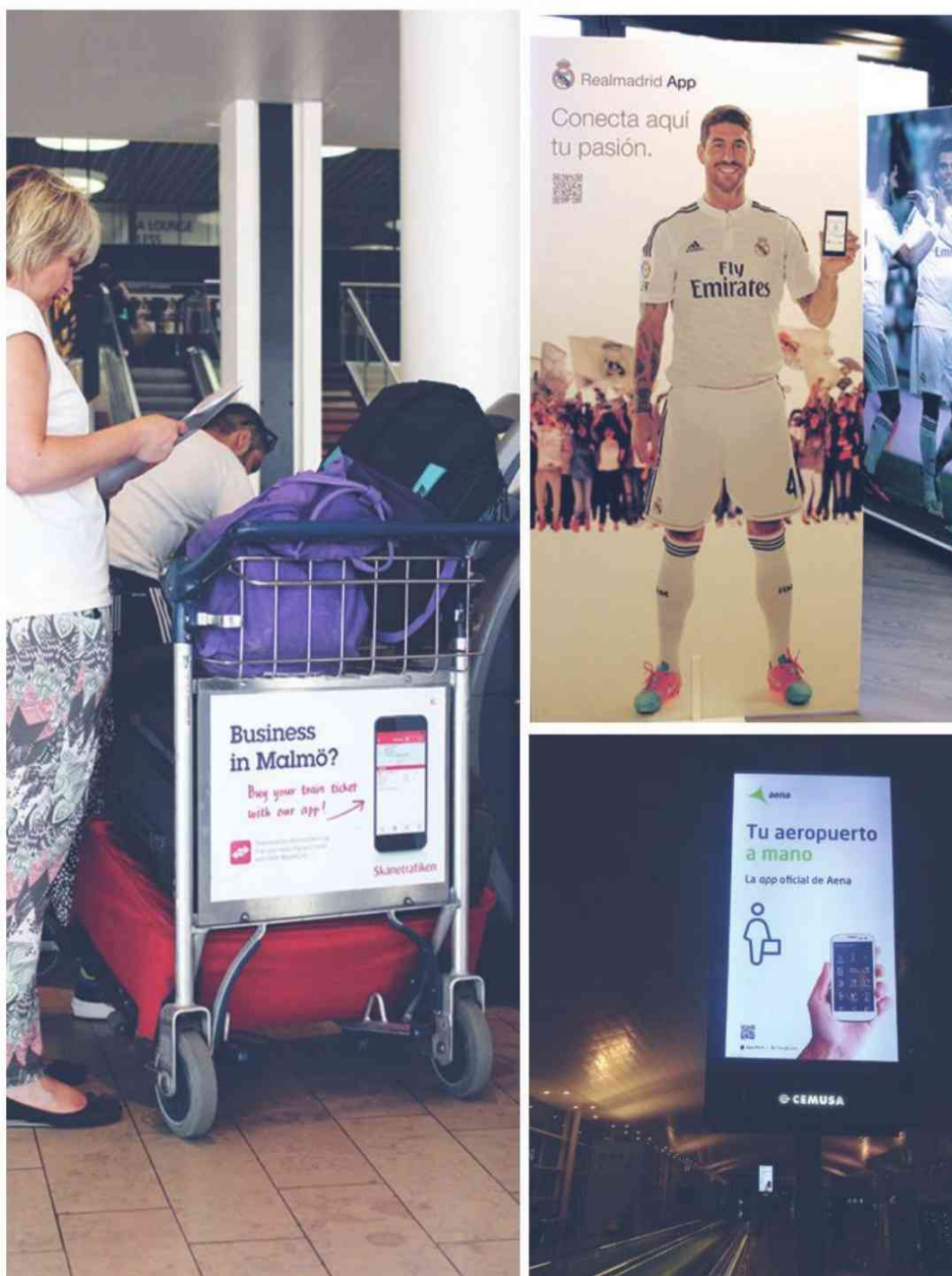


Figura 5-16 Izquierda: publicidad de aplicación móvil en aeropuerto de Copenhague. Derecha-arriba: promoción de aplicación móvil del Real Madrid en el estadio Santiago Bernabéu. Derecha-abajo: Publicidad en aeropuerto de Barcelona promocionando la aplicación móvil de Aena.

Publicidad Impresa

La publicidad impresa de piezas como afiches, volantes, tarjetas, pauta en revistas o lugares estratégicos para capturar la atención del usuario puede ser una vía menos costosa para promover una aplicación en medios tradicionales, en especial cuando se eligen los espacios adecuados para promocionar, un ejemplo pueden ser las revistas especializadas que lee de manera habitual el público objetivo de la aplicación, pero no se reduce solo a esto, la creciente adopción de las aplicaciones ha hecho que empresas de todo tipo, desde casas editoriales hasta restaurantes promocionen sus aplicaciones oficiales en puntos de venta o en las páginas de sus revistas o diarios.

Es común que este tipo de pautas impresas inviten a descargar una aplicación móvil específica con beneficios para el usuario o presenten una edición digital de dicho medio. Por lo general, estas aplicaciones se destacan en concursos o promociones especiales de temporada y también se utilizan en carteles ubicados en diferentes lugares públicos y privados como publicidad de campaña para determinados productos que presentan, como en la publicidad televisiva o exterior, aplicaciones reconocidas por el público masivo. Aunque este tipo de publicidad impresa sigue siendo una práctica poco explotada, se pueden encontrar campañas muy creativas que aprovechan espacios inusuales para promocionar aplicaciones móviles (Figura 5-17).

En otros casos, como método de promoción alternativo a la publicidad explícita y patrocinada, se suele utilizar el *free press*: este método asociado con la aparición en medios, es eficaz porque tiende a ser una legitimación del proyecto por parte de un medio de comunicación. Cuando un periodista de prestigio hace una revisión de una aplicación móvil y arroja comentarios positivos sobre el producto, de inmediato se valida la calidad del mismo ante los lectores que siguen y respetan a ese periodista o medio de comunicación.

Dicha estrategia que por lo general se realiza desde el terreno de las relaciones públicas se convierte en una vía de promoción indirecta que puede ser confiable y muchas veces poco costosa comparada con los precios de publicidad formal y directa. Sin embargo, para acceder a este tipo de plataformas de publicidad debe existir un mérito apreciable en la aplicación que cumpla su función o que aporte algo nuevo al panorama de aplicaciones existente para que un periodista pueda reconocerla y reseñarla.



Figura 5-17 Las piezas impresas donde se promociona la descarga de aplicaciones está en creciente auge. Arriba-derecha: publicidad de la aplicación Live Madrid. Arriba-izquierda y Abajo: piezas en restaurante de McDonalds, incluso una bandeja puede ser un espacio creativo para promocionar una aplicación móvil.

REFERENCIAS Y LECTURAS DE INTERÉS

CAPÍTULO 5

-
- [1] D. Pallotta, «A Logo Is Not a Brand», Harvard Business Review, 15-jun-2011. [Disponible en línea] <http://interfacesmoviles.com/ref/web65>
- [2] J. Hernández Aguirán, El Marketing del Nuevo Lujo. Zargoza, España: Primera Avenida, 2011. pp. 60-61.
- [3] L. R. Oswald, Marketing Semiotics: Signs, Strategies, and Brand Value. OUP (Oxford University Press), 2012. pp. 110-111.
- [4] «Understanding software as a commodity», ZDNet (Tech Industry), 20-jul-2004. [Disponible en línea] <http://interfacesmoviles.com/ref/web66>
- [5] Z. Zhao y C. Balagué, «Designing branded mobile apps: Fundamentals and recommendations», Business Horizons, vol. 58, n.º 3, pp. 305-315, may 2015.
- [6] E. Natanson, «How Major Brands Combine App Innovation With Old-Fashioned Marketing Principles To Win In Mobile», Forbes, 15-jun-2015. [Disponible en línea] <http://interfacesmoviles.com/ref/web67>
- [7] J. Costa, La imagen de marca: un fenómeno social. Barcelona: Paidós, 2004. pp. 147-149.
- [8] J. Clark, Tapworthy: Designing great iPhone apps. Estados Unidos: O'Reilly, 2010. pp. 213-219.
- [9] J. Hicks, «40 years of icons: the evolution of the modern computer interface», The Verge, 21-mar-2013. [Disponible en línea] <http://interfacesmoviles.com/ref/web68>
- [10] A. Hyland y S. Bateman, Símbolos. Barcelona: Gustavo Gili, 2011. p. 9.
- [11] J. M. Klinkenberg, Manual de Semiótica General. Bogotá: Universidad Jorge Tadeo Lozano, 2006. p. 188.
- [12] P. Gabriel-Petit, «Ensuring Accessibility for People With Color-Deficient Vision», UXmatters, 06-feb-2007. [Disponible en línea] <http://interfacesmoviles.com/ref/web69>
- [13] «Use of Color - Understanding Success Criterion 1.4.1», W3C WCAG 2.0. [Disponible en línea] <http://interfacesmoviles.com/ref/web70>
- [14] «Coblis - Color Blindness Simulator», Colblindor. [Disponible en línea] <http://interfacesmoviles.com/ref/web71>
- [15] T. L. Stone, S. Adams, y N. Morioka, Color Design Workbook: A Real World Guide to Using Color. Rockport Publishers, 2008. pp. 20-21.

- [16] D. Kadavy, *Design for Hackers*. Chichester, United Kingdom: John Wiley & Sons, 2011. pp. 264-276.
- [17] F. Thibaudeau, «Manuel français de typographie moderne, faisant suite à "La Lettre d'imprimerie"». Gallica - Bibliothèque Nationale de France, 1924. pp. 108-109 [Disponible en línea] <http://interfacesmoviles.com/ref/web72>
- [18] R. McLean, *Manual de tipografía*. Madrid: Ediciones AKAL, 1993. pp. 58-64.
- [19] «Font and language configuration support for Windows Phone 8». Windows Dev Center. [Disponible en línea] <http://interfacesmoviles.com/ref/web73>
- [20] «Typography - Style», Google design guidelines. [Disponible en línea] <http://interfacesmoviles.com/ref/web74>
- [21] «Fonts - San Francisco», Apple Developer. [Disponible en línea] <http://interfacesmoviles.com/ref/web75>
- [22] J. J. Garret, *The Elements of User Experience: User-Centered Design for the Web and Beyond*, 2nd ed. New Riders, 2011. p.151.
- [23] B. Fling, *Mobile Design and Development: Practical concepts and techniques for creating mobile sites and web apps*. Sebastopol, California: O'Reilly Media, 2009. p.119.
- [24] N. Hagenburger, «Style Guide Driven Development», Speaker Deck. [Disponible en línea] <http://interfacesmoviles.com/ref/web76>
- [25] «Google Play - Graphic assets, screenshots, & video - Developer Console Help». [Disponible en línea] <http://interfacesmoviles.com/ref/web77>
- [26] «Guías de marketing del App Store». Apple Developer. [Disponible en línea] <http://interfacesmoviles.com/ref/web78>
- [27] «Promote your app - Windows app development». Microsoft Developer resources. [Disponible en línea] <http://interfacesmoviles.com/ref/web79>
- [28] H. J. Wilson, P. Guinian y B. D. Weinberg, «What's Your Social Media Strategy?», *Harvard Business Review*, jul-2011. [Disponible en línea] <http://interfacesmoviles.com/ref/web80>
- [29] R. Andrade, «The Rise Of The Explainer Video», *TechCrunch*, 24-mar-2012. [Disponible en línea] <http://interfacesmoviles.com/ref/web81>

CRÉDITOS DE IMÁGENES

Figura 1-1 Dispositivo Nexus 5X, fotografía de prensa. Crédito © Google Inc. y LG Electronics.

Figura 1-2 iPhone 6s y Ipad Pro con sistema operativo iOS, fotografía de prensa. Crédito © Apple Computer.

Figura 1-3 Teléfono Lumia 640 XL, fotografía de prensa. Crédito © Microsoft Corporation.

Figura 1-4 BlackBerry Q10, fotografía de prensa. Crédito © BlackBerry Ltd.

Figura 1-5 Samsung Z, fotografía de prensa. Crédito © Samsung Electronics.

Figura 1-6 Teléfono con Sailfish OS, fotografía de prensa. Crédito © Jolla Ltd.

Figura 1-7 Teléfono BQ Aquaris. Crédito © Mundo Reader S.L.

Figura 1-8 Apple Watch con WatchOS, fotografía de prensa. Crédito © Apple Computer.

Figura 1-10 Imagen creada por los autores utilizando fotografías de prensa de dispositivos móviles Nexus 7, Nexus 4, Samsung Gear S2 y BlackBerry Bold 9790. Créditos © Samsung Electronics, BlackBerry Ltd y Google Inc.

Figura 1-13 Diagrama de disciplinas creado por Sebastian Serna. Crédito de fotografía © Gabriel Valdivia.

Figura 1-14 Iconos Segoe UI Symbol y Windows 10 con interfaz Modern UI, gráficos de prensa. Créditos © Microsoft Corporation.

Figura 1-15 Aplicación Google Calendar con Material Design e iconos, fotografía de prensa. Crédito © Google Inc.

Figura 1-16 Sistema operativo iOS 9, servicio iCloud e iconos de interfaz. Crédito © Apple Computer.

Figura 2-6 Aplicación People hub de Windows Phone que utiliza el tipo de navegación lineal con el control panorama. Crédito © Microsoft Corporation.

Figura 2-14 Diagrama creado por los autores basado en el esquema de Donald A. Norman en el libro "The Design of the Everyday Things: Revised and Expanded" p. 32. El gráfico creado utiliza ilustraciones de Freepik y un dispositivo Ipad con aplicación iBooks. Crédito © Apple Computer.

Figura 3-6 Imagen y simulación creada por Sebastian Serna sobre cómo las personas sostienen el móvil basada en las investigaciones e ilustraciones previas de Steven Hooper.

Figura 3-8 a Figura 3-21 Imágenes creadas a partir de recursos y gráficos de interfaz libres de los siguientes autores: Adrian Pelletier, Alex Boamfa, Alex Patrascu, Alvin Thong, Chris Arvin, Davey Heuser, Dillen Verschoor, EmojiOne, Freepik, GUIToolkits, James Gill, Joachim Löfstedt, Material Icons, Matthew Daniels, Max Fellmuth, MediaLoot, Rami McMin, Riki Tanone, Sebastian Gabriel, Sebastian Serna, Stephen Hutchings, Teehan-Lax, Thomas Bossée, Tomas Gajar y Vladimir Kudinov

Figura 4-1 Pruebas de usabilidad en laboratorio portátil y controlado fotografías propiedad de Mike Lee (cámara instalada en dispositivo móvil) y sesión de pruebas del Interaction Lab, City University London (Reino Unido).

Figura 4-6 Imagen creada por los autores a partir de capturas de pantalla del sitio web y recurso App Icon Template. Crédito © Michael Flarup.

Figura 4-8 Imagen creada por los autores utilizando capturas de pantalla del generador de iconos Android Asset Studio proyecto desarrollado por Roman Nurik.

Figura 4-9 Software Microsoft visual Studio, imagen derivada de video tutorial de Raman Sharma, Senior PM Manager en Microsoft. Crédito © Microsoft Corporation.

Figura 4-10 Imagen creada por los autores utilizando capturas de pantalla del generador web Modern UI Icons para Windows Phone.

Figura 5-3 Círculo cromático natural creado por los autores inspirado en ilustración de S. Davis titulada "Círculo de los colores fundamentales" en el libro La creación en publicidad (1989), Barcelona, Ed. L.E.D.A. p. 33.

Figura 5-5 Imagen creada por los autores utilizando gráficos de la aplicación Red / Green. Crédito © Ryan Dawidjan.

Figura 5-6 Imagen creada por los autores utilizando gráficos de prensa de Yahoo! Mail. Crédito © Yahoo! Inc.

Figura 5-9 Imagen creada por los autores utilizando mockup de Nokia Lumia. Crédito © Corey Ginnivan.

Las demás figuras incluidas en los capítulos del libro corresponden a capturas de pantalla directas de aplicaciones móviles, iconos, videos y sitios web que se utilizan como ejemplo en los diferentes temas abordados. Por otro lado, las imágenes que se excluyeron del anterior listado y no corresponden a capturas de pantalla (gráficos y fotografías) fueron creadas por Sebastian Serna como apoyo a los contenidos del presente libro.

MARCAS COMERCIALES. Las designaciones utilizadas por las empresas para distinguir sus productos (hardware, software, sistemas operativos, etc.) suelen ser marcas registradas. RA-MA ha intentado a lo largo de este libro distinguir las marcas comerciales de los términos descriptivos, siguiendo el estilo que utiliza el fabricante, sin intención de infringir la marca y solo en beneficio del propietario de la misma.

Los autores no poseen afiliación alguna con las marcas comerciales mencionadas en los diferentes capítulos. Estas pertenecen a sus respectivos propietarios y solo se han utilizado en el presente libro con fines didácticos.

ÍNDICE ALFABÉTICO

A

accesibilidad 18, 31, 77, 176
Action Bar (Android) 110, 111
Adobe Color 176
affordances 68, 69
afiches 196
AirServer (App) 142
Alexander, Christopher 108
Amazon App Store 4, 25
Android 3, 4, 10, 12, 16, 17, 18, 24,
38, 109, 150, 156, 189
Android Market 24, 25
Android Studio 4, 156
Android Wear 4, 15
animaciones 32, 96, 106
aplicaciones híbridas 27, 130
aplicaciones móviles 20, 21
aplicaciones nativas 4, 6, 8, 13, 15, 26
aplicaciones web 27
App Bar (Windows) 110, 111
Apple 6, 8, 10, 15, 16, 20, 40, 171,
173
Apple Watch 6, 15
Appsee 144
App Store 6, 8, 20, 24, 25, 27, 154,
188, 189, 196

Aqua (interfaz) 40
armonías de color 176
Arquitectura de Información 28, 29,
33, 54, 92
asistentes de voz 94
audio 88, 94
Axure 56, 82

B

Bada OS 12
banners 188
barras de desplazamiento 96, 104
barras de opciones 110
barras de progreso 105
Bauhaus 36
Beck, Kent 108
Bill, Max 32
BlackBerry 10, 11, 16, 17, 18, 19
BlackBerry Priv 10
BlackBerry World 10, 25
Blend (Visual Studio) 162
blueprints 56
bocetos 80
botones 96, 97
botones de radio 96, 98, 164
BQ Aquaris 14
búsqueda 33, 66, 122

C

C 4, 10, 14, 156
C# 8, 26, 150, 151, 160
C++ 4, 8, 10, 13, 14, 152, 156, 160
cajas de chequeo 96, 99, 164
campañas de promoción 172, 194, 196
campos de búsqueda 122
campos de inserción de texto 92, 96, 101
campos de mensajes 124
canales de promoción 188
Canonical 14
card sorting 54, 67, 92
chat 20, 49, 124
checkboxes 99, 164
Chisnell, Dana 147
círculo cromático 176
colores primarios 177
colores secundarios 177
colores terciarios 177
combo boxes 100, 165
commodities 172
compras 21, 75, 130
Comunicación Visual 32
constraints 68, 69
contenedores 84, 87
contenido 92
contexto de uso 49
controles interactivos 96
Cook, Tim 40
Cooper, Alan 51
correo electrónico 20, 124
Cortana (Windows) 77, 94
CSS 10, 12, 14, 27, 156
Cuadros de diálogo 121
Cunningham, Ward 108

D

daltonismo 176
densidades de pantalla (Android) 156
deslizadores 96, 103
diagramas 56

dialog box 121
Diseño centrado en el usuario 48, 139
Diseño de Interacción (IxD) 28, 29, 30, 32
Diseño Gráfico 32
Diseño Industrial 28, 29, 33
diseño iterativo 48, 141
Diseño Visual 28, 29, 30
dispositivos corporales 15, 18
dispositivos móviles 18, 19
Downs, Jenn 144
dp (density-independent pixel) 157
Draw 9-patch 158
drop-down lists 100, 165
Duarte, Matías 38

E

EasyTaxi 21, 119, 129
Eclipse (IDE) 4, 151, 156
efectos sonoros 94, 106
elementos interactivos 96
elementos multimedia 84, 88
Encuestas 50
Entrevistas contextuales 50
escenarios 53, 146
Esslinger, Hartmut 35
estilos de interacción 108, 109
estilo suizo 36
estructura 56, 75
etiquetas 54
etnografía 31
Evaluación heurística 148, 149
Experiencia de Usuario (UX) 28, 30

F

FAB (Floating Action Button) 38
Facebook 20, 21, 125, 192, 196
Fancy 21, 193
feedback 68, 69
Flarup, Michael 155
flat 2.0 38
flat design 36

Flipboard 127
Fong, Jeff 36
formularios 109, 122
Forstall, Scott 40
front-end (programación) 33

G

galerías 116
Garret, Jesse James 30, 56
Gartner (consultora) 17
geolocalización 49, 128
gestos de interacción 76
gestos táctiles 78, 79, 106
Gmail 22, 124, 176
Google 16, 24, 35, 38, 125
Google Maps 21, 128
Google Now 77, 94, 120
Google Play 4, 8, 20, 24, 27, 188, 196
Gothelf, Jeff 52
gráficos elásticos (stretchables) 154
grupos focales 51
guías de estilo 35, 186

H

hamburger icon 118
hardware 31
hdpi (pantallas) 157
Honeycomb (Android) 38
Hoover, Steven 85, 86
HTC 4, 8, 16, 157
HTML 10, 12, 14, 26, 150, 154, 160
hub and spoke 60, 116
hug your laptop 144
Human-Computer Interaction (HCI) 31
Human Interface Guidelines 40, 164

I

Ice Cream Sandwich (Android) 38, 182
iconos 88, 90, 154
iconos (Android) 159
iconos de aplicación 173, 174, 175

iconos (iOS) 154
iconos (Windows Phone) 162
identidad de marca 35, 171, 172
Illustrator 82
imagen del sistema 69
imagen del sistema (Donald Norman)
67
Imagen en movimiento 95
Imágenes 88
Implementación 150
indicadores de progreso 96, 105, 106
infinite list 114
Instagram 20, 21, 196
Intel 12, 13
inteligencia artificial 33, 94
IntelliJ (IDE) 4, 156
Interacción Persona-Ordenador (IPO)
28, 31
Interface Builder 6, 152
Interfaz de Usuario 30, 32, 34, 88, 172
interruptores 96, 102
investigación de usuarios 50, 139
InVision 82
Ionic 27
iOS 3, 6, 7, 10, 15, 16, 17, 24, 40,
109, 150, 152, 154, 164, 189
iPad 6, 7, 153
iPhone 6, 7, 24, 153, 154
ISO 9241-210 48
ISO 13407 48
iTunes 189
Ive, Jonathan 40

J

Java 4, 10, 26, 151, 156
Javascript 10, 12, 14, 27, 150
jerarquía 92
Jobs, Steve 40
Jolla 13
Jotform 50
JPG 154, 158

K

Kindle Fire 25

L

landing pages 190

landscape (orientación) 85

lápiz digital 77

layout 87

ldpi (pantalla) 157

Lean UX 52

lenguaje de programación 27

lenguaje natural 77, 94, 108

lenguajes de diseño 35

Ley de Fitts 85

lienzo de dibujo 126

limitaciones 68

Linux 3, 12, 13, 14, 156, 187

listados de opciones 114

listas desplegables 96, 100

living styleguides 186

logotipos 133, 171, 173, 174

Lollipop (Android) 38

lorem ipsum 92

Lumia (Microsoft) 8, 161

M

Mac OS 3, 6, 24, 40, 152, 156

Maemo 13

Mailchimp 144

manipulación directa 109, 126

mapas 21, 128, 165

mapping 68, 69

Marvel (App) 82

Material Design 35, 38, 39, 62, 97, 120

matrices de opciones 116

mdpi (pantallas) 157

MeeGo 13

mensajería 21

menús 100

menús laterales 118

mercadeo 171

metáforas 67, 127

métodos de inspección (usabilidad) 148

Metro Style 8, 36

microinteracciones 106

Microsoft 8, 16, 36, 40, 162

Mobile First 35

mockups 82

mockups interactivos 82, 83

modelo conceptual 67, 69

modelo mental 67, 69

Modern UI 8, 35, 36, 37

Moggridge, Bill 32

Molich, Rolf 148

Morville, Peter 30, 33, 54

mosaicos 36

Motorola 4, 15

multimedia 21, 88

multi-touch 76, 79

N

navegación 33, 56, 58, 75, 118, 130

Navegación centralizada 60

Navegación interconectada 62

Navegación lineal 58

Navegación plana 64

Navigation Bar (iOS) 110

NDK (Android) 4

Neil, Theresa 109

nested doll 66

Netbeans (IDE) 4, 156

Netflix 95, 196

Neue Helvética 182

Newton, Isaac 176

Nexus (dispositivo) 5

Nielsen, Jakob 55, 140, 148

Nine-patch 158, 159, 162

Nokia 8, 13, 17

Norman, Donald 67, 68

notificaciones push 121

Noto (tipografía) 182

O

Objective C 6, 15, 26, 152
Omnigraffle 56, 82
Open Handset Alliance 4
Optimal Workshop 55
option buttons 98
ordenamiento de tarjetas 54

P

página de aterrizaje 190
paletas análogas 180
paletas complementarias 179
paletas monocromáticas 178
paletas triádicas 181
Palm 38
paneles de pestañas 112
paneles vacíos 132
panorama (Windows) 58
pantallas 56, 58, 75, 84, 85
pantallas (Android) 157
pantallas de bienvenida 133
pantallas de contenido o detalle 130
pantallas de ingreso 125
pantallas de registro 125
pantallas (iOS) 153
pantallas táctiles 18, 76
pantallas (Windows Phone) 161
párrafos 92
patrones de diseño 108, 164, 187
pauta publicitaria 194
PDF 154, 186
personas 51
PhoneGap 27
Photoshop 82, 158
PHP 27
picker (iOS) 100
piezas publicitarias 190
Pinterest 21
planeación 47
PNG 154, 158
podcasts 94
POP (App) 82

portrait (orientación) 85
Priv (BlackBerry) 10
profundidad de navegación 66
programación 28, 29, 33, 150
progress bars 105
proto-personas 51
prototipos 50, 75, 80, 92, 139
prototipos de alta fidelidad 82
prototipos de baja fidelidad 80
prototipos interactivos 82, 139
pruebas de guerrilla 141, 143
pruebas de laboratorio 141, 142
pruebas de usabilidad 48, 50, 139, 141
pruebas remotas 141
publicidad exterior 194, 196
publicidad Impresa 198
Python 13, 27

Q

QML 13, 14
QNX 10
Qt 10, 13, 14

R

radio buttons 98, 164
Raskin, Jef 20
redes sociales 20, 192
Reflector (App) 142
relojes inteligentes 3, 4, 15, 18, 31
reportes (usabilidad) 147
responsive design 27
retículas 87
retina display 154
retroalimentación 68, 94
revisiones de expertos 148
RIM 17
Roboto (tipografía) 38, 182
Rosenfeld, Louis 33, 54
Rubin, Andy 4
Rubin, Jeff 147
Ruby (lenguaje) 27

S

Saffer, Dan 106
Sailfish OS 17, 173
Samsung 8, 12, 15, 16
San Francisco (tipografía) 182
scrollbars 78, 104
SDK (Android) 4
SDK (iOS) 6
Search Bar (iOS) 122
seekbar 103
Segoe UI Symbol 36, 162
Segoe UI (tipografía) 36, 182
selección de menús 109, 110
Sense (HTC) 16
sensores 76, 79
Shneiderman, Ben 108, 148
signifiers 68, 69
Siri (iOS) 77, 94
sitios web 186, 190, 191, 194
Sketch 82, 158
sketches 80
skeuomorphism 40, 102
sliders 103
Smartface 150
Snow White 35
sociología 31
sonido 94
Sony 4, 15
spinners 100
splash screens 133
Spotify 21, 22
StackPanel 87
startups 20, 47, 194
storyboards 53, 80
stretchables (gráficos iOS) 154, 158
styleguides 186
stylus 18, 19, 77, 126
SurveyMonkey 50
Swift (lenguaje) 15, 26, 152
switches 102
Symbian OS 3, 4

T

tabletas 18, 19, 31, 156
tap (gesto) 76
tarjetas de contenido 120
teclados físicos 18, 77
teléfonos inteligentes 3, 18, 20, 24, 33, 194
televisión 194
textos 88
The New York Times 22, 183
Thibaut, Francis 182
Thinking Aloud 146
tiendas de aplicaciones 20, 188
tile cycle (Windows) 162
tile flip (Windows) 162
tile iconic (Windows) 162
tiles (Windows) 8, 36, 162, 163
tipografía 182, 186
tipografía decorativa 182, 185
tipografía sans-serif 182, 184
tipografía serif 182, 183
Titanium 27
Tizen 12, 17, 173
toggle button 102
Tognazzini, Bruce 148
Toolbar (iOS) 110
tooltip 121
toque (gesto) 76, 78
Touchwiz (Samsung) 16
Transiciones y animación 96, 106
Twitter 21, 23, 192

U

Uber 21, 128, 175
Ubuntu 14, 17, 173, 187
usabilidad 31, 48, 139, 140, 172
usability 140
User-Centered Design 48
User Interface 30
UserTesting 144
UXPA 30
UXpin 82

V

vallas publicitarias 196
VB (lenguaje) 8, 160
video 20, 88, 95, 188, 194
videojuegos 20, 21, 76, 106
videos explicativos 194
Visio 56
Vision Mobile 28
visores de mapas 128
Visual Studio 8, 160, 162
vocabulario visual 35
vocabulario visual (Garret) 56
Vox, Maximilien 182

W

WatchOS 6, 15, 173
Waze 22, 129
Web Clip (iOS) 154
WebOS 38
WhatsApp 10, 20, 101, 124, 196
Widget 162
widgets (controles) 96, 164
WIMP (interfaces) 76, 78
Windows 3, 8, 24, 36, 67, 92, 109, 156, 160
Windows Mobile 8
Windows Phone 8, 13, 16, 58, 160, 182
Windows Store 20, 25, 188
wireframes 75, 80
Wroblewski, Luke 35
WVGA (pantallas) 161
WXGA (pantallas) 161

X

Xamarin 150
XAML 8, 150, 160, 161, 162
X-BOX 36
Xcode 6, 15, 152
Xerox PARC 34, 67
Xiaomi 4, 16

XIB 6

XML 4, 6, 8, 150, 152, 156, 158, 160

Xtensio (App) 52

xxhdpi (pantallas) 157

xxxhdpi (pantallas) 157

Y

Yahoo! 114, 119, 180

Youtube 21, 95

Z

zoom in 79, 128

zoom out 79, 128

DISEÑO DE INTERFACES EN APLICACIONES MÓVILES

En este libro encontrará el contenido y los recursos prácticos para la creación y diseño de aplicaciones móviles.

Encontrará referencia a los sistemas operativos, las disciplinas relevantes en el diseño de interfaces gráficas y su definición a través de diagramas, gestos táctiles, elementos multimedia, controles interactivos, patrones de diseño comunes así como los métodos de evaluación y promoción más representativos. Los temas abordados se desarrollan de manera sencilla y didáctica con más de 80 ejemplos de aplicaciones móviles reales y materiales de apoyo adicionales en la web del libro y en la de los autores con constantes actualizaciones.

Si es programador, director de proyectos, diseñador digital o estudiante de carreras afines al diseño, ingeniería o comunicación, este libro representa una guía esencial que le ayudará a complementar sus conocimientos sobre el proceso de creación de software y su identidad de marca en el contexto de la telefonía móvil.



ra-ma.es



Ra-Ma®