Diseño de interfaces para dispositivos móviles

Jordi Flamarich Zampalo

PID_00204730
Índice

1. Introducción: del diseño centrado en el usuario al diseño centrado en el contexto......................................................... 7
   1.1. La interfaz móvil .................................................................................................................................................. 9
       1.1.1. Móvil o sobremesa .................................................................................................................................. 9
       1.1.2. Web, web móvil, app ......................................................................................................................... 10

2. El ecosistema móvil......................................................................................................................................................... 12
   2.1. Dispositivos ........................................................................................................................................................ 13
       2.1.1. Teléfonos inteligentes ......................................................................................................................... 13
       2.1.2. Tabletas ............................................................................................................................................... 14
   2.2. Sistemas operativos ............................................................................................................................................. 15
       2.2.1. iOS ........................................................................................................................................................ 15
       2.2.2. Android ............................................................................................................................................... 16
       2.2.3. Windows Phone ..................................................................................................................................... 18
       2.2.4. Otros sistemas operativos ................................................................................................................ 19

3. Principios para el diseño de interfaces móviles.................................................. 22
   3.1. Simplicidad ....................................................................................................................................................... 22
   3.2. Eficiencia .......................................................................................................................................................... 23
   3.3. Consistencia ..................................................................................................................................................... 23
   3.4. Interacción ....................................................................................................................................................... 23
   3.5. Metáforas ........................................................................................................................................................ 24
   3.6. Respuesta ......................................................................................................................................................... 25

4. Diseño para la interacción................................................................................................................................. 27
   4.1. Galería de gestos ............................................................................................................................................. 27
       4.1.1. Toque .................................................................................................................................................... 27
       4.1.2. Toque largo ......................................................................................................................................... 28
       4.1.3. Deslizar ............................................................................................................................................... 28
       4.1.4. Arrastrar ............................................................................................................................................. 29
       4.1.5. Separar y pellizcar con dos dedos .................................................................................................. 30
       4.1.6. Rotar .................................................................................................................................................... 31
       4.1.7. Girar y rotar el dispositivo ................................................................................................................ 31
       4.1.8. Mover .................................................................................................................................................... 33
   4.2. Elementos gráficos ........................................................................................................................................... 34
       4.2.1. Títulos .................................................................................................................................................... 34
       4.2.2. Iconos .................................................................................................................................................. 34

5. Patrones para el diseño de interfaces móviles.......................................................... 36
   5.1. Navegación ..................................................................................................................................................... 36
       5.1.1. Pantalla de inicio .................................................................................................................................. 36
       5.1.2. Notificaciones .................................................................................................................................... 37
5.1.3. Paginación ................................................................. 38
5.1.4. Menús ................................................................. 41
5.1.5. Elementos de control ............................................. 44

5.2. Mostrar información ......................................................... 47
5.2.1. Listados ............................................................... 48
5.2.2. Pestañas ............................................................... 48
5.2.3. Carrusel ............................................................... 49
5.2.4. Cuadrícula ............................................................ 50
5.2.5. Pila ................................................................. 50
5.2.6. Pase de diapositivas ............................................. 51
5.2.7. Anotación ............................................................ 51

5.3. Entrada de datos ............................................................. 52
5.3.1. Teclado ............................................................... 52
5.3.2. Voz ................................................................. 55
5.3.3. Campos de texto y formularios ................................ 55
5.3.4. Botones .............................................................. 58
5.3.5. Interruptores, barras y selectores ............................ 58

5.4. Buscar y filtrar información ............................................. 61
5.4.1. Búsqueda dinámica .................................................. 63
5.4.2. Autocompletado .................................................... 64
5.4.3. Búsqueda con filtros .............................................. 65
5.4.4. Últimas búsquedas .................................................. 66
5.4.5. Formulario de búsqueda ....................................... 67
5.4.6. Búsqueda por geoposición ...................................... 67
5.4.7. Ordenar los resultados ............................................. 67

5.5. Respuesta al usuario .......................................................... 68
5.5.1. Confirmación ........................................................ 68
5.5.2. Mensajes de error ................................................... 69
5.5.3. Indicadores de espera ............................................. 71

5.6. Ayuda ............................................................................ 73
5.6.1. Diálogo ............................................................... 73
5.6.2. Consejo ............................................................... 74
5.6.3. Instrucciones ......................................................... 75
5.6.4. Visita guiada ........................................................ 76
5.6.5. Transparencias ...................................................... 77
5.6.6. Video ayuda ........................................................ 78

6. Desarrollar aplicaciones para dispositivos móviles .................. 80
6.1. Fase de conceptualización ............................................. 80
6.1.1. Definición de las funciones .................................... 80
6.1.2. Definición del usuario y el contexto ......................... 81
6.1.3. Descripción de la aplicación .................................... 81
6.2. Fase de diseño .............................................................. 82
6.2.1. 10 consejos para un buen diseño ............................ 82
6.2.2. Usabilidad y experiencia de usuario .......................... 86
6.2.3. Prototipado .......................................................... 87
6.3. Fase de desarrollo .......................................................... 93
6.3.1. Aplicaciones nativas ...................................................... 93
6.3.2. Aplicaciones web ........................................................... 94
6.3.3. Aplicaciones híbridas ..................................................... 94

Bibliografía......................................................................................... 97
1. Introducción: del diseño centrado en el usuario al diseño centrado en el contexto

En su libro *The design of everyday things*, Donald A. Norman popularizó el término *diseño centrado en el usuario* (en inglés, *user-centered design*), una forma de diseñar productos “basada en los intereses y las necesidades del usuario, que pone el énfasis en fabricar productos usables y fáciles de entender”.

Según Norman, el diseño tendría que:

• dejar muy claro qué se puede hacer en cada momento;
• ser transparente, es decir, dejar a la vista tanto las posibles acciones como sus resultados;
• facilitar la evaluación del estado del sistema en todo momento.

Dicho de otro modo, el usuario tendría que saber en todo momento qué puede hacer y qué sucede. Norman ponía la facilidad de uso por delante incluso de la estética del producto.

Aplicado al desarrollo de interfaces, el diseño centrado en el usuario implica que los requerimientos de los usuarios son tenidos en cuenta desde las primeras etapas de desarrollo, y no solo al final del proceso, en la fase de pruebas con usuarios finales, cuando a menudo puede ser demasiado tarde para corregir errores. Por lo tanto, para tener una mínima garantía de éxito, todo proceso de desarrollo de interfaces tendría que empezar pensando en las personas, sus usuarios potenciales:

• ¿Quién usará esta interfaz?
• ¿Por qué la usará?
• ¿Cuáles son sus intereses?
• ¿Y sus motivaciones?

Obviamente, en esta batería de preguntas iniciales también nos tenemos que preguntar en qué circunstancias interactuarán los usuarios con nuestra interfaz:

• ¿Dónde usarán la aplicación?
• ¿Será un espacio público o privado?
• ¿Será de día o por la noche?
• ¿Estarán haciendo otra cosa al mismo tiempo?
Con un ordenador de sobremesa, este contexto era más o menos predecible – como mínimo, el usuario tenía que estar quieto ante la pantalla manipulando un teclado y un ratón– pero con la llegada de la tecnología móvil, ubicua por definición, los contextos de uso se han multiplicado, y en ocasiones se vuelven casi impredecibles.

Brian Fling (2009) distingue tres tipos de contexto:

1) **Localización** o **contexto físico**: es el lugar en el que nos encontramos. El contexto físico condiciona nuestra actividad, ya sea por la privacidad que nos proporciona o por el nivel de atención que podemos dedicar al dispositivo. Por otro lado, gracias a la capacidad de geolocalización de estos dispositivos, el contexto físico nos proporciona toda una serie de información que podemos aprovechar para ofrecer servicios al usuario en función de su situación sobre el mapa.

   **Ejemplo de condicionamiento por el contexto físico**

   Cuando vamos solos en el coche podemos consultar información confidencial en el móvil sin problemas, pero no le podremos dedicar mucha atención si no queremos tener un accidente. En cambio, sentados en el bus no podremos consultar mucha información privada, pero sí que le podremos dedicar toda nuestra atención al móvil.

2) **Contexto de acceso**: los dispositivos móviles nos permiten estar permanentemente conectados a la red, obteniendo información en tiempo real sobre nuestro entorno. Este contexto de acceso, sin embargo, puede variar tanto en función del dispositivo como de las condiciones en que se encuentre: cobertura, calidad y velocidad de la conexión a la red, características del dispositivo (tamaño de la pantalla, procesador, memoria RAM...), estado de la batería...

   **Ejemplo de condicionamiento por el contexto de acceso**

   No tiene sentido desarrollar una guía para turistas que se encuentran en el extranjero si la aplicación funciona solo conectada a internet, puesto que la mayoría de sus usuarios potenciales no tendrán itinerancia de datos y no la podrán usar.

3) **Contexto modal** o **estado de ánimo**: nuestro estado de ánimo o humor es quizás la mayor influencia sobre qué, cuándo y dónde hagamos las cosas que hacemos. Así, si estamos aburridos, querremos que nuestro dispositivo nos distraiga; si estamos tristes, que nos levante el humor; si estamos nerviosos, que nos calme...

Al desarrollar interfaces móviles, pues, tenemos una oportunidad singular de crear experiencias únicas basadas en los más variados contextos. Solo si somos capaces de entender tanto al usuario como su contexto, su estado de ánimo o sus intereses, podremos desarrollar aplicaciones que le aporten valor.
1.1. La interfaz móvil

En este subapartado compararemos las ventajas y los inconvenientes que presentan las interfaces móviles frente a las de los ordenadores de sobremesa, y las interfaces web móvil y app frente a los esquemas clásicos de interfaces web.

1.1.1. Móvil o sobremesa

A la hora de diseñar una aplicación o estructurar contenidos para dispositivos móviles, tenemos que intentar huir de los clásicos esquemas aprendidos con la web y los ordenadores personales.

Con un ordenador:

- El usuario está sentado y prestando toda la atención.
- El entorno es predecible, no acostumbra a variar.
- El usuario tiene un teclado y un ratón para interactuar con el ordenador.
- La pantalla es grande y permite al usuario tener mucha información delante al mismo tiempo.
- La multitarea es posible.

Con un móvil:

- El usuario puede estar sentado, andando por la calle, a salto de mata, circulando en bicicleta, etc.
- El entorno es variable.
- La atención es limitada, sujeta a todo tipo de interrupciones.
- El usuario interactúa con una pantalla táctil y/o teclas muy pequeñas.
- La pantalla es pequeña.
- La multitarea es complicada y farragosa.

Por otro lado, el espacio dentro de la pantalla de un dispositivo móvil es un bien muy preciado. Se tiene que evitar el uso de cualquier elemento (franjas fijas –o banners, en inglés–, barras, imágenes o gráficos) que ocupe espacio sin tener una función específica. Los usuarios quieren acceder a la información o
completar la tarea requerida de la forma más rápida y más eficiente posible, de manera que los elementos irrelevantes serán ignorados en cuanto descubran qué es importante y qué no para conseguir sus objetivos.

Cuando diseñamos para un móvil no se trata de hacer las cosas más pequeñas, sino de centrarnos en aquello que es esencial.

1.1.2. Web, web móvil, app

Una buena manera de entender cómo el diseño de interfaces móviles tiene poco que ver con un ordenador o con el diseño de páginas web es comparando un lugar como Amazon en sus versiones web, web móvil y aplicación:

Web, web móvil, app

a) Vista directamente en el navegador de un teléfono iPhone, la web de Amazon resulta incómoda: tenemos que ampliar constantemente la pantalla para poder leer el contenido o ampliar los enlaces hasta un tamaño que nos permita no cometer errores al pulsarlos con el dedo.

b) Para evitar las incomodidades de la web tradicional en la pequeña pantalla de un móvil, es habitual encontrarnos con versiones adaptadas, donde el ancho se ajusta perfectamente a la pantalla, y elementos como por ejemplo botones y menús se han adaptado para su interacción con los dedos. Aun así, esta versión no aprovecha al máximo las capacidades táctiles del dispositivo: por ejemplo, el desplazamiento lateral se tiene que hacer pulsando botones, y no deslizando con el dedo.

Lectura para profundizar


Sobre las diferencias entre web y móvil: Shanshan Ma (2011). “10 ways mobile sites are different from desktop sites” (accesible en línea). UX matters (11 de marzo).
c) Finalmente, la aplicación de Amazon para iPhone nos muestra los mismos contenidos pero con un sistema de navegación y una interfaz plenamente adaptada al dispositivo. Aquí, el usuario se encontrará confortable con el menú inferior (un estándar del sistema operativo) y podrá desplazar los elementos tanto vertical como horizontalmente sin necesidad de usar botones.
2. El ecosistema móvil

Cuando hablamos de dispositivos móviles, a la mayoría nos viene a la cabeza un teléfono, seguramente de última generación (también llamados smartphones, o teléfonos inteligentes). Si lo pensáramos mejor, seguramente acabaríamos incluyendo también en esta categoría las tabletas, unos aparatos cada vez más habituales en los hogares.

Pero en el mercado hay multitud de otros dispositivos móviles, como por ejemplo:

- Agendas electrónicas
- Consolas de videojuegos
- Reproductores multimedia
- Libros digitales (e-readers)
- Marcos de fotografía digitales
- Sistemas de navegación vía satélite (GPS)
- Cámara de fotografía
- ...

Algunos ejemplos del ecosistema móvil

Fuente: Wikipedia Commons

Aun así, cuando hablamos de dispositivos móviles en este espacio, estaremos haciendo referencia básicamente a teléfonos y tabletas, los dispositivos que hoy por hoy están marcando el camino a seguir en el diseño de interfaces. Y estaremos hablando de dispositivos con pantallas táctiles, una característica que se ha convertido ya en un estándar en el sector.

En este sentido, buena parte de los principios y patrones expuestos aquí para teléfonos y tabletas táctiles son plenamente válidos para el resto de dispositivos, toda vez que la enorme popularidad de estos aparatos está provocando que el resto haya empezado a imitar su apariencia y funcionamiento, incorporando pantallas táctiles.
2.1. Dispositivos

A continuación pasamos a describir las características de los teléfonos inteligentes y las tabletas.

2.1.1. Teléfonos inteligentes

La telefonía móvil dio un giro radical en enero del 2007, fecha en la que Steve Jobs, cofundador y presidente ejecutivo de Apple, presentó por primera vez el iPhone. Con este dispositivo, Apple transformó el concepto de smartphone (en inglés, teléfono inteligente), y cambió radicalmente la manera como se había interactuado hasta entonces con un teléfono: Apple convirtió el móvil en un aparato intuitivo, multitáctil, y siempre conectado a internet.

La gran revolución del iPhone fue doble: en primer lugar, Apple consiguió llevar la experiencia de navegar por internet al teléfono móvil, prácticamente en igualdad de condiciones respecto a un ordenador de sobremesa.

En segundo lugar, Apple reinventó el concepto de aplicación para los teléfonos móviles: el iPhone era un dispositivo en el que, como si de un ordenador se tratara, se podían instalar pequeños programas desarrollados por terceros (apps), los cuales se podían descargar –de forma gratuita o mediante pago– desde un mercado controlado por el propio fabricante (App Store). De este modo, la aparición del iPhone también marcó el nacimiento de una nueva industria dedicada a la creación de aplicaciones.

El teléfono iPhone de Apple revolucionó el mercado de los teléfonos inteligentes, tanto por las características de su hardware como por la posibilidad de instalar todo tipo de aplicaciones (apps).

Por otro lado, el enorme impacto mediático del iPhone provocó que el resto de fabricantes de teléfonos empezaran a fabricar terminales con pantallas táctiles, imitando con mejor o peor fortuna el funcionamiento casi mágico del dispositivo de Apple. En este sentido, poco después (octubre del 2008) aparecía el primer teléfono móvil con el sistema operativo Android –impulsado por Google–, entrando en competencia directa con los teléfonos de Apple.

En cualquier caso, cinco años después del primer modelo de iPhone, se calcula que ya hay más de 1.000 millones de usuarios de teléfonos inteligentes en el mundo, cifra que se prevé que se duplique en el 2015 (Strategyanalytics.com). No es Apple, sin embargo, quien lidera el mercado: prácticamente siete de cada diez teléfonos inteligentes son Android (noviembre del 2012, IDC - Press Release).
En cuanto a las características básicas que definen un teléfono inteligente, este dispositivo tiene que ser:

- **Pequeño**: su tamaño tiene que permitir llevarlo siempre encima, si puede ser, en el bolsillo.

- **Portátil**: funciona con baterías, y por lo tanto, no hay que tenerlo enchufado para que funcione.

- **Siempre conectado**: se trata de un aparato con conexión constante a las redes, ya sea para la transmisión de datos o voz.

- **Interactivo**: tiene que permitir que el usuario lleve a cabo todo tipo de acciones: entrada de datos y texto, búsquedas en internet, organizar información, etc.

- **Sensible al entorno**: gracias a sus sensores, puede facilitar el trabajo al usuario detectando dónde se encuentra (GPS), adaptando la brillantez de la pantalla a las condiciones lumínicas, etc.

### 2.1.2. Tabletas

Apple volvió a marcar un punto de inflexión en el 2010 cuando presentó el iPad, un aparato a medio camino entre el teléfono móvil y el ordenador portátil, pero que a diferencia de este último prescindía del teclado físico y el ratón para interactuar con él. Con un iPad se podía navegar por internet, enviar y recibir correo, hacer fotos y vídeos, reproducir música... todo en un aparato multitáctil pensado para que el usuario descargara aplicaciones (juegos, mapas, redes sociales...) a través de la tienda virtual de la compañía, la App Store.

Cuando Steve Jobs subió al escenario para presentarlo, muchos se apresuraron a decir que se trataba de un “iPhone gigante” o de un juguete pretencioso abocado al fracaso. Por concepto, se alejaba demasiado de un ordenador, y al mismo tiempo se asemejaba demasiado a su hermano pequeño, el iPhone.

Pero a pesar de no ser el primer aparato de estas características que salía al mercado, el iPad de Apple sí que acabó siendo el primero en obtener un gran éxito comercial. Después de él empezaron a aparecer tabletas de todos los tamaños, la mayoría –una vez más– con el sistema operativo Android, intentando ganar parte del nuevo mercado abierto por el iPad.

Las características básicas de una tableta son idénticas a las de un teléfono inteligente, a excepción de su tamaño (según el modelo) y de que una tableta no está pensada para hacer llamadas telefónicas (aunque potencialmente sí que tendría la capacidad para hacerlo).
2.2. Sistemas operativos

Los sistemas operativos para dispositivos móviles más representativos hoy en día son iOS, Android, y Windows Phone.

2.2.1. iOS

iOS es el sistema operativo desarrollado por Apple para los dispositivos iPhone, iPad, iPod Touch y Apple TV. Se trata de un sistema cerrado, basado en el sistema operativo Mac OS X (el de los ordenadores Apple), y que solo se puede instalar en los dispositivos de la misma compañía. Apple suele lanzar una gran actualización del sistema operativo una vez al año, y va liberando posteriormente pequeñas correcciones y mejoras.

El usuario se puede descargar aplicaciones nuevas para su dispositivo solo a través del App Store, una tienda virtual que cuenta con más de 700.000 apps disponibles, bien gratuitas o de pago (el precio de venta mínimo es de 0,89 euros).

Su interfaz está pensada para la manipulación directa, mediante gestos sobre la pantalla, de elementos como por ejemplo iconos, botones o interruptores. Se trata de un sistema multitáctil, de modo que el dispositivo es capaz de identificar y responder cuando el usuario toca dos o más puntos de la pantalla a la vez.

La interacción con el sistema operativo incluye gestos como por ejemplo, deslizar con el dedo (swipe), dar un toque (tap) o dos sobre la pantalla (double tap) o pellizcar (pinch), que con el tiempo han sido comportamientos esperados por parte de los usuarios en otros sistemas operativos móviles. Los dispositivos de Apple también incorporan elementos, como el acelerómetro o el giroscopio, que les permite responder en caso de que el usuario mueva el dispositivo o lo coloque en posición horizontal, cambiando automáticamente la disposición de los elementos en la pantalla.

La interfaz gráfica está basada en una pantalla principal (springboard) donde se encuentran los accesos directos a las aplicaciones en forma de iconos cuadrados con las esquinas redondeadas. Para abrir una aplicación solo hay que dar un toque en la pantalla con el dedo sobre el icono correspondiente. El usuario puede volver a la pantalla de inicio en cualquier momento pulsando el único botón físico presente en el frontal del dispositivo, llamado “Inicio” (home).

### Enlaces para profundizar

- Web oficial Android: http://www.android.com/
- Web oficial Windows Phone: http://www.windowsphone.com/es-es
- Web oficial Blackberry: http://es.blackberry.com/
- Web oficial Tizen: https://www.tizen.org/
- Web oficial Ubuntu for phones: http://www.ubuntu.com/devices/phone
La distribución de los iconos en la pantalla principal se hace mediante una malla organizada en páginas, que el usuario puede ir pasando horizontalmente deslizando con el dedo. La parte inferior de la pantalla (denominada puerto –en inglés, dock–) es fija, y permite al usuario situar hasta cuatro iconos con las aplicaciones que use más a menudo. En la parte superior, se encuentra la barra de estado (status bar), con información sobre la hora, el nivel de batería restante y la calidad en la recepción de las redes disponibles (WiFi, 3G, etc.).

<table>
<thead>
<tr>
<th>Características</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sistema operativo</td>
<td>iOS</td>
</tr>
<tr>
<td>Lenguaje de programación</td>
<td>Objective-C</td>
</tr>
<tr>
<td>SDK (software development kit)</td>
<td>Xcode</td>
</tr>
<tr>
<td>Coste de la licencia del desarrollador</td>
<td>99 dólares/año</td>
</tr>
</tbody>
</table>

2.2.2. Android

Es el sistema operativo desarrollado por Google junto con el Open Handset Alliance, un consorcio de fabricantes de teléfonos, desarrolladores de software y operadoras de telefonia para el desarrollo de estándares abiertos para dispositivos móviles.

Android es un sistema operativo de código abierto basado en Linux que se puede instalar en multitud de dispositivos más allá de teléfonos móviles y tabletas: ordenadores portátiles, libros electrónicos, televisores, reproductores multimedia, o incluso gafas y relojes.

La versión a marzo del 2013 es la 4.2, denominada Jelly Bean. A diferencia de iOS, donde los usuarios pueden actualizar el sistema operativo de su dispositivo hasta que Apple considera que este ha quedado obsoleto, es poco habitual
que un dispositivo Android pueda actualizarse a una versión superior del sistema operativo. Es por este motivo por lo que en el mercado conviven gran cantidad de dispositivos con diferentes versiones de Android.

Dispositivos con el sistema operativo Android

Google Play es el principal mercado desde donde los usuarios pueden descargar nuevas aplicaciones (hay cerca de 700.000 disponibles), pero no es el único, puesto que el sistema permite la descarga de aplicaciones de otros mercados gestionados por terceros (por ejemplo, el Amazon Appstore).

La interfaz de Android está pensada para su manipulación directa mediante gestos sobre la pantalla (deslizar con el dedo, tocar la pantalla, pellizcarla, ...), siendo un sistema operativo multitáctil. Los dispositivos Android también suelen incorporar elementos como el acelerómetro o el giroscopio, que permiten detectar la posición del aparato y adaptar en consecuencia la orientación de la pantalla en vertical (portrait) u horizontal (landscape).

La interfaz gráfica está basada en una pantalla de inicio que, al igual que si se tratara del escritorio de un ordenador, contiene iconos con accesos directos a las aplicaciones instaladas en el sistema, carpetas y widgets que muestran información de forma actualizada sin necesidad de abrir la aplicación correspondiente (por ejemplo, predicción del tiempo, correo electrónico, canales de noticias...).

La pantalla de inicio puede estar formada por varias páginas con iconos y widgets, que el usuario puede ir pasando deslizando con el dedo. Al igual que con los dispositivos iOS, en la base de la pantalla hay una zona fija donde los usu-

Fuente: Android Developers
Los dispositivos Android también cuentan con una barra de navegación con tres botones que permiten, de izquierda a derecha, ir un paso atrás (back), volver a la pantalla de inicio (home) y ver las últimas aplicaciones que se han usado, ordenadas de forma cronológica (recents).

<table>
<thead>
<tr>
<th>Características</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sistema operativo</td>
</tr>
<tr>
<td>Lenguaje de programación</td>
</tr>
<tr>
<td>SDK (software development kit)</td>
</tr>
<tr>
<td>Coste de la licencia del desarrollador</td>
</tr>
</tbody>
</table>

2.2.3. Windows Phone

Windows Phone es el sistema operativo desarrollado por Microsoft para entrar en competencia con Apple (iOS) y Google (Android) y no quedar fuera del emergente mercado de teléfonos inteligentes y tabletas. Se trata de un sistema cerrado, basado en el mismo sistema operativo (Windows NT) que llevó a Microsoft a dominar el mercado de los ordenadores de sobremesa. Si bien se puede encontrar instalado en varios fabricantes de hardware, Windows Phone tiene como principal referente al fabricante finlandés Nokia, fruto del acuerdo al que llegaron ambas compañías en el 2011 para crear un nuevo ecosistema móvil.

La versión a mes de marzo del 2013 es la Windows Phone 8, plenamente compatible con la versión del sistema operativo para ordenadores personales de Microsoft, Windows 8, lo que permitirá a los desarrolladores llevar las aplicaciones fácilmente de un sistema al otro.

Los usuarios pueden descargarse aplicaciones a través de Windows Phone Store, la tienda oficial con más de 100.000 apps disponibles.

La interfaz de Windows Phone está diseñada para su manipulación directa mediante gestos sobre la pantalla del dispositivo, siendo un sistema operativo multitáctil. Windows Phone también detecta las dos posiciones de pantalla (portrait y landscape).

Windows Phone presenta una novedosa interfaz gráfica, denominada Metro, basada en una pantalla de inicio (home) con pequeñas gemas (live tiles) de diferentes tamaños que funcionan como accesos directos a aplicaciones, funciones del sistema o incluso elementos concretos, como por ejemplo, un contacto o página web. El usuario puede añadir tantas gemas como quiera a la panta-
lla de inicio, así como modificar su posición y tamaño. Las gemas tienen un comportamiento dinámico y modifican su apariencia en función de la información disponible (por ejemplo, muestran el número de mensajes pendientes de leer, llamadas perdidas, las últimas fotografías hechas, etc.).

A diferencia de otros sistemas, la pantalla principal de Windows Phone no se organiza en páginas, sino que las gemas se acumulan en un único espacio que se puede desplazar verticalmente. El desplazamiento lateral (de derecha a izquierda) sirve aquí para mostrar el listado entero de aplicaciones instaladas en el sistema.

Los dispositivos con Windows Phone cuentan con tres botones que, de izquierda a derecha, sirven para ir un paso atrás (back), volver a la página de Inicio (home) y hacer búsquedas en internet a través de Bing, el buscador de Microsoft.

<table>
<thead>
<tr>
<th>Características</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sistema operativo</td>
<td>Windows Phone</td>
</tr>
<tr>
<td>Lenguaje de programación</td>
<td>C, C++</td>
</tr>
<tr>
<td>SDK (software development kit)</td>
<td>Windows Phone 8 SDK</td>
</tr>
<tr>
<td>Coste de la licencia del desarrollador</td>
<td>99 dólares/año</td>
</tr>
</tbody>
</table>

### 2.2.4. Otros sistemas operativos

Aparte de los que ya hemos visto, también podemos encontrar otros sistemas operativos como por ejemplo Blackberry OS, Symbian o Firefox OS.

**Blackberry OS**

Blackberry OS es el sistema operativo desarrollado por Research in Motion (RIM) para los dispositivos Blackberry, unos teléfonos conocidos por su comodidad a la hora de gestionar el correo electrónico o enviar mensajes de texto, gracias a su teclado físico QWERTY. Este, sin embargo, ha sido el principal problema de estos dispositivos, que no han sabido adaptarse al éxito de las pantallas táctiles, motivo por el cual su cuota de mercado está en claro declive.
Se trata de un sistema cerrado, solo disponible para los dispositivos Blackberry, diseñado para su control mediante elementos físicos: teclado, rueda giratoria, botones... Con el tiempo han ido apareciendo modelos de Blackberry (teléfonos y tabletas) con pantalla táctil o bien que combinaban la pantalla táctil con un teclado físico.

De hecho, la versión del sistema operativo presentada en enero del 2013, Blackberry 10, es una buena muestra de cómo RIM está intentando no perder el tren de los teléfonos inteligentes con pantallas táctiles. Uno de los primeros modelos de teléfono presentado bajo este sistema operativo, el Z10, no tiene botones físicos y funciona completamente a base de gestos sobre la pantalla (básicamente, deslizar con el dedo), hecho que, si bien demuestra un intento de marcar diferencias con la competencia, no fue muy bien recibido por los analistas (“Watch People Try to Use BlackBerry 10 for the First Time”, accesible en línea), al considerar el sistema demasiado complicado y poco intuitivo. En respuesta, la compañía tuvo que publicar una guía de gestos para aprender a controlar el nuevo dispositivo (“How to use BlackBerry 10 gestures”, accesible en línea).

Los usuarios se pueden descargar aplicaciones a través de la Blackberry App World, con más de 70.000 títulos disponibles para la última versión del sistema operativo.

**Symbian**

Hasta hace muy poco era el sistema operativo por excelencia de los teléfonos Nokia, antes de que la compañía finlandesa llegara a un acuerdo con Microsoft para empezar a fabricar dispositivos con el sistema Windows Phone. Se trata de un sistema operativo para teléfonos inteligentes previo a la eclosión de las pantallas táctiles y, por lo tanto, diseñado para ser operado mediante un teclado físico.

En las últimas versiones del sistema operativo se ha hecho un esfuerzo para adaptarlo a dispositivos táctiles, pero sin volver a tener el favor del público. Tanto es así, que la cuota de mercado de Symbian está en caída libre, pasando de un 15% a finales del 2011 a poco más de un 2% en el tercer trimestre del 2012 (datos: IDC).

**Firefox OS**

Todavía en fase de desarrollo, Firefox OS es un buen ejemplo de lo que está por llegar en el sector de los dispositivos móviles. Se trata de un sistema operativo impulsado por Mozilla bajo la misma marca del popular navegador de internet, construido sobre Linux y basado en el lenguaje HTML5 –evolución del tradicional HTML de internet– y Javascript. Incluso las aplicaciones más básicas del teléfono (llamadas, mensajería, etc.) están programadas en HTML5.
En palabras de la propia fundación Mozilla, el objetivo de Firefox OS es usar la propia web como plataforma de desarrollo, liberando los dispositivos móviles de “las normas y restricciones de las plataformas privadas existentes”.

En este sentido, Mozilla no es la única que está trabajando en un nuevo sistema operativo basado en Linux. Además de Firefox OS, en el 2013 se introducen en el mercado terminales con Tizen, sistema operativo impulsado por la Linux Foundation con el apoyo de Samsung e Intel, y Ubuntu, impulsado por Canonical.
3. **Principios para el diseño de interfaces móviles**

Los principios básicos que hay que seguir para el diseño de interfaces móviles son simplicidad, eficiencia, consistencia, interacción y metáforas, que pasamos a describir a continuación.

3.1. **Simplicidad**

Andando por la calle, bajando unas escaleras, en la sala de espera del médico, en el metro camino del trabajo... los usuarios usan sus dispositivos móviles en las más variadas situaciones, sometidos a constantes interrupciones, y a menudo sin prestarles el 100% de su atención. Es por este motivo por lo que las interfaces móviles tienen que ser fáciles de usar desde el primer momento: en general, el usuario percibirá como un contratiempo tener que pararse en medio de la calle para descubrir cómo funciona la aplicación que se acaba de descargar.

**Versiones Photoshop para ordenador (a) y para interfaz móvil (b)**

![Imagen de comparación: versión de ordenador y versión móvil de Photoshop]

La versión móvil del popular programa de retoque fotográfico Photoshop ha reducido drásticamente el número de opciones disponibles respecto a la versión de ordenador, limitándose a las funciones más básicas.

La mayoría de usuarios de dispositivos móviles no valoran la complejidad, sino que sus aplicaciones hagan bien –y rápido– una sola cosa.

Hay que presentar al usuario en primer término solo aquellas opciones imprescindibles para conseguir su objetivo o completar la tarea central de la aplicación. Una interfaz llena de botones y opciones desconcertará al usuario. En cambio, si le damos pocas opciones, el usuario aprenderá rápidamente los gestos necesarios para manipular con éxito la interfaz, haciendo que le resulte
cada vez más agradable de usar. El resto de opciones pueden quedar escondidas detrás de un menú, y dejar que sea el usuario quien las descubra poco a poco, a medida que vaya usando la aplicación.

3.2. Eficiencia

La eficiencia de una interfaz viene definida por el número de pasos que tiene que dar el usuario para conseguir un determinado objetivo. Las tareas más importantes, por lo tanto, tienen que estar claramente accesibles y conseguirlas con el menor número de toques o movimientos sobre la pantalla.

En este sentido, podemos aprovechar los sensores del dispositivo para evitarle al usuario tener que introducir datos manualmente (por ejemplo, ofreciéndole automáticamente la previsión del tiempo de la ciudad donde se encuentre), o bien recordar sus preferencias de una sesión a otra, aprender del uso que haga de la aplicación o incluso prever su próxima acción, todo con el objetivo de facilitarle el trabajo.

3.3. Consistencia

La interfaz de nuestra aplicación tiene que ser consistente con el dispositivo, el sistema operativo donde se usará y el resto de aplicaciones con las que convivirá. Esto no quiere decir que tenga que ser una copia poco innovadora, sino que el usuario tendrá que encontrar la aplicación fácil de usar porque respete los estándares y paradigmas del sistema operativo del dispositivo. Además, la interfaz tiene que ser consistente consigo misma (por ejemplo, los botones se encuentran siempre en el mismo lugar y tienen el mismo comportamiento en todas las pantallas).

3.4. Interacción

En los humanos, el sentido del tacto es una herramienta indispensable para interactuar con nuestro entorno, quizás la más intuitiva. Hasta hace muy poco, esta forma de interactuar no era posible en entornos virtuales. Con el auge de las pantallas táctiles en los dispositivos móviles, la interacción háptica (del griego haptikos, ‘relativo al sentido del tacto’) se ha popularizado, puesto que permite a los usuarios manipular directamente los objetos que hay en la pantalla. Este hecho aumenta su sensación de control sobre la interfaz y les permite una rápida y mejor comprensión sobre las consecuencias de sus acciones.

Para reforzar esta sensación hace falta que el dispositivo responda fielmente a los movimientos del usuario. En este sentido, el usuario también puede recibir una respuesta háptica a sus acciones, normalmente a través de la vibración del dispositivo cuando usa el teclado virtual o pulsa un botón en la pantalla.
Finalmente, la interacción con los dispositivos móviles puede darse tanto tocando la pantalla como moviendo el propio aparato (por ejemplo, rotando la pantalla), gracias a los sensores de movimiento que incorporan.

Siempre que sea posible, facilitaremos que el usuario interactúe directamente con el contenido sin la intermediación de botones, menús, barras, etc.

A la hora de diseñar la interfaz, es aconsejable tener siempre en cuenta todas las maneras en que el usuario intentará interactuar con ella, no solo aquellas que nosotros hayamos pensado y diseñado.

3.5. Metáforas

Si presentamos los objetos y las acciones en la interfaz como una metáfora de objetos y acciones en el mundo real, los usuarios aprenderán rápidamente a interactuar con la interfaz y la encontrarán más atractiva.

El ejemplo clásico en este sentido es la carpeta: la gente usa carpetas en el mundo real para guardar documentos; en la interfaz, por lo tanto, una carpeta será rápidamente identificada como un elemento contenedor. Otras metáforas habituales en las interfaces son la caja de herramientas o los engranajes para representar las opciones de configuración, un sobre de cartas para el correo electrónico, un auricular para el teléfono, etc.

Estas metáforas no hace falta que se limiten al diseño de iconos u otros elementos gráficos: por ejemplo, también se pueden usar para diseñar los gestos que después tendrá que hacer el usuario, como por ejemplo pasar las páginas de un libro con el dedo, encender o apagar interruptores o cambiar la fecha en una rueda giratoria.
3.6. Respuesta

La sensación de control derivada de la manipulación directa de la interfaz por parte de los usuarios se tiene que ver reforzada en todo momento por una respuesta rápida, inmediata, a sus acciones. Esta respuesta confirma al usuario la acción realizada, quien de otra manera podría sentirse frustrado o intentar la misma acción sucesivas veces, produciendo distorsiones en la entrada de datos.

Imaginemos por ejemplo alguien que quiere comprar billetes de avión a través de su teléfono móvil. Mediante un formulario parecido al que encontraría en una página web, introduce origen, destino, número de pasajeros y las fechas de ida y vuelta deseadas. A continuación, pulsa el botón que activa la búsqueda, pero no parece que ocurra nada. Espera unos cuantos segundos, y lo vuelve a pulsar, fijándose esta vez en si el botón responde de alguna manera al toque de su dedo, cosa que no sucede. Entonces empieza a dudar de si la aplicación funciona, pulsa el botón de buscar un par de veces más y a continuación intenta entrar, también sin éxito, en los campos del formulario. Parece que la aplicación se ha quedado efectivamente “colgada”. Pero justo en aquel momento, la pantalla cambia mostrándole los resultados de su búsqueda.

En este ejemplo, podemos distinguir dos tipos de respuesta a la acción del usuario que han fallado. En primer lugar, el usuario no ha advertido ninguna reacción cuando ha pulsado el botón de búsqueda, cosa que le ha hecho desconfiar. En segundo lugar, cuando ha pulsado el botón de nuevo, ha visto que este no reaccionaba a su gesto, y esto le ha hecho desconfiar aún más.

La velocidad de respuesta ante una acción del usuario no significa que el sistema tenga que ser igualmente rápido procesando su petición. Dicho de otro modo: es posible que el usuario no obtenga al instante aquello que pide del sistema, pero sí tiene que saber al momento que el sistema está trabajando en ello.

En el ejemplo que acabamos de ver, el hecho de pulsar repetidamente el botón de búsqueda no tiene consecuencias negativas para el usuario, pero si esto mismo sucediera en otro lugar de la aplicación, por ejemplo a la hora de pagar los billetes, podría ser que el usuario acabara pagando sus billetes dos veces o más.

Por otro lado, la sensación de respuesta va más allá de usar indicadores de respuestas. En pantallas táctiles, la sensación de respuesta se puede conseguir mediante cambios en la posición, orientación, color e iluminación del elemento accionado, animándolo de forma sutil con sonidos o incluso activando brevemente el mecanismo de vibración del dispositivo. Buena parte de estas res-
puestas vienen predeterminadas por el sistema operativo mismo, de forma que con su uso continuado devienen comportamientos esperados por parte de los usuarios.

En este sentido, hay que tener en cuenta que el usuario valorará la velocidad en la respuesta de su dispositivo por encima de la estética, de modo que, antes de diseñar animaciones complejas, tendremos que tener bien claro que el dispositivo será capaz de llevarlas a cabo con fluidez.

Ejemplo de respuesta del sistema operativo Android
En este ejemplo del sistema operativo Android, el usuario detecta que ya no puede seguir pasando páginas porque el contenido de la última página topa con una barrera invisible, sobre la cual se apoya brevemente.
Fuente: developer.android.com (accesible en línea)
4. Diseño para la interacción

Con el auge de los teléfonos móviles inteligentes y las tabletas hemos entrado en una nueva era del diseño para la interacción. Durante los últimos 40 años hemos usado los mismos paradigmas para la interacción entre personas y ordenadores (en inglés, *human-computer interaction*), la misma metáfora del escritorio. Estos paradigmas son todavía plenamente vigentes, pero irán siendo progresivamente sustituidos por otros, basados en nuevas formas, más directas, de relacionarnos con la tecnología. Como dice Dan Saffer, estamos ante una oportunidad que se presenta solo una vez por generación: es ahora cuando apenas estamos empezando a diseñar nuevas formas de interactuar con nuestros dispositivos, nuestro entorno e incluso entre nosotros.

4.1. Galería de gestos

A continuación presentamos el conjunto de gestos más habituales que se emplean para la interacción con las interfaces de dispositivos móviles.

4.1.1. Toque

El toque (*tap*) es el gesto más natural que se puede hacer sobre una pantalla, el más primario, y por lo tanto, el más importante, puesto que con él se llevan a cabo la mayor parte de acciones sobre un dispositivo táctil: abrir aplicaciones, accionar botones, seleccionar elementos... Para quien no ha interactuado nunca con una pantalla táctil el toque es el gesto más fácil de aprender, puesto que le recordará al clic que se hace con un ratón de ordenador.
Variantes de este gesto son el doble toque (*double tap*) y el triple toque (*triple tap*), aunque este último es muy poco habitual y hay que justificar muy bien su uso, puesto que puede resultar difícil de ejecutar por parte de algunos usuarios.

### 4.1.2. Toque largo

En este gesto, el toque largo (*tap and hold*), el usuario tiene que tocar un elemento en pantalla y no levantar el dedo durante un par de segundos, hecho que desencadena acciones alternativas o el despliegue de menús sobre este elemento. En algunos casos, el toque largo se puede comparar al clic con el botón secundario del ratón, aunque sus usos son varios en función del sistema operativo o la aplicación en la que se use. Así, por ejemplo, sobre un teclado, este gesto permite tener acceso a los caracteres acentuados, mientras que sobre un icono de aplicación, nos permite arrastrarlo y cambiarla de lugar.

### 4.1.3. Deslizar

Después del toque, deslizar (*swipe*) con el dedo sobre la pantalla es el otro gesto más natural que podemos hacer sobre un dispositivo táctil, y con el cual se pueden llevar a cabo las más variadas acciones. La más básica es desplazar el contenido de la pantalla en vertical, como si lo hiciéramos con la rueda giratoria de un ratón, para continuar leyendo una página web, por ejemplo. En horizontal, este gesto es habitual para pasar las imágenes de una galería, o cambiar de página en la pantalla de inicio.

Las variantes de este gesto tienen que ver con:

a) La velocidad con que se haga. De hecho, es el propio usuario quien, una vez acostumbrado al movimiento de deslizar el dedo sobre la pantalla, empezará a hacerlo cada vez más rápido y más corto para minimizar esfuerzos. Este movi-
miento rápido del dedo sobre la pantalla en una dirección recibe el nombre de *flick*, y se puede usar para pasar páginas de un libro, fotografías de una galería, desplazarnos rápidamente por una página web, etc.

b) El número de dedos que hacemos deslizar a la vez sobre la pantalla. Podemos diseñar acciones para dos, tres, e incluso cuatro dedos deslizándose al mismo tiempo y en la misma dirección sobre la pantalla, aunque para teléfonos –dado el tamaño de sus pantallas– resulta recomendable no pasar de dos. Por otro lado, en tabletas no hay que abusar de gestos con tres o cuatro dedos, puesto que resultan incómodos y estorban la visión de buena parte de la pantalla.

### 4.1.4. Arrastrar

Con el gesto de arrastrar (*drag*) se puede tanto mover un elemento dentro de la pantalla (por ejemplo, trasladar un ícono a una nueva ubicación o reordenar los elementos de una lista) como desplazar el foco de la propia pantalla cuando tenemos una imagen ampliada al máximo.

![Arrastrar](image)

A pesar de disponer de un botón específico para esta función en la parte superior izquierda de la pantalla, la aplicación de Facebook (iOS) permite mostrar el menú arrastrando el dedo de izquierda a derecha sobre la pantalla. Haciendo el gesto en sentido contrario, volvemos a esconder el menú.

Hasta cierto punto, este gesto se puede confundir con el de hacer deslizar el dedo (*swipe*) sobre la pantalla: cuando arrastramos, sin embargo, normalmente estamos interactuando con un elemento concreto (un ícono, una imagen ampliada), mientras que el gesto de deslizar se hace normalmente para navegar o desplazarnos.
En función de dónde se inicia el gesto de arrastrar también se usa para mostrar las notificaciones (arrastrando en vertical, de arriba abajo, y empezando fuera de la pantalla), o el menú dentro de una aplicación (arrastrando en horizontal, empezando desde fuera de la pantalla).

4.1.5. Separar y pellizcar con dos dedos

El gesto de separar con dos dedos (spread) se ha acabado convirtiendo en sinónimo de ampliar (zoom in) aquello que tenemos en pantalla, y como tal es un comportamiento esperado por los usuarios, sobre todo cuando visualizan fotografías, consultan un servicio de mapas o usan el navegador web. Su contrario, es decir, hacer el gesto de pellizcar la pantalla (pinch), hace la acción a la inversa, alejando (zoom out) aquello que tengamos en pantalla.

Hay aplicaciones que están empezando a usar los gestos de separar y pellizcar con otras finalidades más allá de ampliar o reducir lo que tenemos en pantalla. Así, por ejemplo, la aplicación Clear para crear listas los usa para crear nuevos elementos dentro de una tabla (separando) o navegar hacia un nivel superior del menú (pellizcando la pantalla).

Fuente: “Clear for iPhone - Available Now!”, Vimeo (accesible en línea)
4.1.6. Rotar

Con este gesto hacemos rotar (rotate) sobre su eje el contenido mostrado en pantalla. Es un gesto habitual cuando se visualizan mapas y que, combinado con el de separar y pellizcar, permite controlar la interfaz con agilidad y precisión.

4.1.7. Girar y rotar el dispositivo

Gracias al acelerómetro y giroscopio que incorporan la mayoría de dispositivos móviles, son capaces de detectar cambios en su orientación, de vertical (portrait) a horizontal (landscape) o viceversa.

Aunque no es un gesto que se haga directamente sobre la pantalla, este cambio de orientación de la pantalla se está convirtiendo rápidamente en un comportamiento esperado por parte de los usuarios, sobre todo si están visualizando fotografías o vídeos.

Este gesto, sin embargo, se puede usar para más cosas que para permitir una visión más cómoda de la pantalla:

a) Se pueden diseñar vistas alternativas de la interfaz en función de la orientación del dispositivo. Este recurso puede ir más allá de la simple reordenación de los elementos mostrados en pantalla, ofreciendo nuevas funciones al usuario. Un buen ejemplo lo tenemos en la calculadora, que en vertical ofrece los controles básicos mientras que en horizontal –al disponer de más espacio para situar botones– ofrece los controles de una calculadora científica:

![Dos vistas de la calculadora según la orientación del dispositivo](image)

En este ejemplo de iOS, la calculadora cambia de modo (básico a científico) en función de la orientación del dispositivo.

Otro ejemplo es el calendario del iPhone, que con el dispositivo en horizontal muestra la distribución horaria de la agenda:
Dos vistas de la agenda según la orientación del dispositivo

El calendario del iPhone con el dispositivo en horizontal muestra la distribución horaria de la agenda.

Hay que dejar muy claro al usuario la disponibilidad de estas vistas alternativas, puesto que de otro modo podrían pasar del todo inadvertidas. Si bien es cierto que puede haber usuarios que disfruten descubriendo estas opciones por sí solos, la mayoría agradecerá siempre recibir información clara sobre qué puede hacer y qué no puede hacer con su dispositivo.

Algunos dispositivos Samsung (Android) permiten controlar el zoom en pantalla inclinando el dispositivo.

b) También se pueden usar el acelerómetro y el giroscopio para convertir el propio dispositivo en un mando, un recurso habitual en juegos que permite al usuario, por ejemplo, controlar la dirección de un vehículo sobre una carretera simplemente girando el dispositivo.
c) En aplicaciones de realidad aumentada (AR, de la expresión inglesa *augmented reality*), este gesto se convierte en la principal manera de interactuar con la interfaz, puesto que es girando el dispositivo cómo este le muestra, como si fuera una ventana, la capa de información sobre las imágenes que proporciona la cámara.

**Nokia City Lens**

Nokia City Lens es una aplicación que usa la realidad aumentada para mostrar servicios y otros lugares de interés alrededor de donde se encuentra el usuario.

Fuente: Nokia.com

**Vídeo**

Sobre la realidad aumentada, podéis ver el vídeo siguiente en YouTube:

“Nokia City Lens for Nokia Lumia: Augmented Reality Browser (Beta)” (accesible en línea)

---

4.1.8. Mover

El acelerómetro también nos permite detectar cuándo el usuario mueve lateralmente y de forma reiterada (*shake*) su dispositivo, un recurso que podemos aprovechar para activar una acción determinada, crear atajos a funciones habituales (por ejemplo, reproducir o parar el reproductor de música, descartar una llamada) o facilitar la interacción o el intercambio de información con otros dispositivos.
Mover con Line

Line es una aplicación que permite enviar mensajes y hacer llamadas vía internet, donde los usuarios pueden añadir nuevos contactos moviendo a la vez sus dispositivos.

Así, por ejemplo, en el sistema operativo iOS, el gesto de mover de este modo el dispositivo permite deshacer (undo) la última acción, mientras que en aplicaciones como Line (mensajería instantánea) permite añadir nuevos contactos sin necesidad de tenerlos que introducir manualmente.

4.2. Elementos gráficos

A continuación pasamos a describir algunos de los elementos gráficos que podemos encontrar habitualmente en las interfaces de dispositivos móviles.

4.2.1. Títulos

Los títulos son una parte esencial de cualquier sistema operativo, aplicación o página web. También en los dispositivos móviles: todas las páginas, secciones y elementos de la interfaz tienen que ir convenientemente identificados para no desorientar al usuario. Las guías de diseño de cada sistema operativo acostumbran a incluir indicaciones sobre cómo tienen que ser las barras de título.

4.2.2. Iconos

Los iconos se acostumbran a usar como atajos tanto para acceder a aplicaciones desde la pantalla de inicio como para acceder a elementos o funciones específicas una vez dentro de estas. Es aconsejable que los iconos estén acompañados de texto para su fácil identificación. Esta etiqueta suele estar situada
bajo el icono y centrada respecto a la imagen. La etiqueta no debería ocupar más de una línea; en caso de ser demasiado larga, se puede truncar el texto o hacerlo pasar como si fuera una marquesina.

Podemos encontrar varios tipos de icono:

**a) Iconos fijos**: representan con una imagen el elemento o función a la que dan acceso. La imagen tiene que ser muy clara y entenderse al primer vistazo.

*Juego de iconos del sistema operativo Android*

Fuente: developer.android.com (accesible en línea)

**b) Iconos de estado**: cambian en función de elementos externos, como por ejemplo la llegada de nuevos mensajes, llamadas perdidas, etc. Es importante que la base del icono se mantenga inalterable para que el usuario la pueda reconocer en todo momento. El elemento variable se puede mostrar, por ejemplo, mediante un contador en una de las esquinas.

*Icono de estado*

*Widget con iconos interactivos del sistema operativo Android*

Fuente: developer.android.com (accesible en línea)

**c) Iconos interactivos**: no sirven como atajo, sino para llevar a cabo acciones directas, como por ejemplo conectar el WiFi o el GPS, ajustar la brillantez de la pantalla, etcétera. Su diseño tiene que permitir visualizar rápidamente el estado del elemento.
5. Patrones para el diseño de interfaces móviles

A la hora de diseñar interfaces, los patrones nos pueden resultar de gran utilidad, puesto que:

- Nos ayudan a ahorrar trabajo porque proponen soluciones a problemas ya resueltos. Y si los aplicamos bien, nos ayudan a resolver otros problemas parecidos.

- Hacen que nuestra interfaz sea fácil de usar, puesto que una vez aplicados de forma generalizada por la comunidad de diseñadores y desarrolladores, acaban siendo modelos mentales para los usuarios.

5.1. Navegación

A continuación comentamos los principales elementos de navegación que podemos hallar en dispositivos móviles.

5.1.1. Pantalla de inicio

Si en los ordenadores nos hemos acostumbrado a hablar de escritorio para hacer referencia a la pantalla principal del sistema, en los dispositivos móviles hablamos de pantalla de inicio. Es allí donde se encuentran los accesos directos a las aplicaciones y funciones del sistema. En el sistema operativo móvil de Apple, iOS, esta pantalla recibe el nombre de Springboard, y con el tiempo ha resultado un patrón para pantallas de inicio de otros sistemas operativos.

La pantalla de inicio puede constar de varias páginas, donde el usuario ordena los iconos de acceso a las aplicaciones y funciones disponibles en el sistema. Para pasar de página en la pantalla de inicio, el usuario tiene que deslizar con el dedo lateralmente.

En el sistema Android, además de iconos, el usuario puede situar sobre la pantalla de inicio widgets, pequeños programas con información que se actualiza constantemente (previsión del tiempo, noticias, redes sociales, etc.).

Finalmente, en la parte inferior de la pantalla hay una parte fija (llamada dock) que no cambia cuando el usuario pasa de página, y es donde se pueden fijar los accesos a las aplicaciones más utilizadas.
5.1.2. Notificaciones

Las notificaciones sirven para advertir al usuario de alertas o cambios en el estado del sistema y aplicaciones. El ejemplo más claro es el aviso de una llamada perdida o la llegada de un nuevo mensaje, pero las notificaciones también se pueden usar para mostrar al usuario el siguiente acontecimiento marcado en su agenda o alertarlo sobre nuevas actualizaciones disponibles para el sistema.

Normalmente, es el propio sistema operativo quien marca cómo y dónde se muestran las notificaciones, con las características generales siguientes:

- Además de advertir al usuario de forma clara, las notificaciones en ningún caso pueden interrumpir la tarea que en aquel momento se esté llevando a cabo. En el caso de teléfonos móviles, obviamente, la única excepción a esta norma es la entrada de una llamada de teléfono.

- Ante una nueva notificación, el usuario tiene que poder decidir si quiere atenderla o dejarla para más tarde. Lo más habitual es mostrar las notificaciones solo durante unos segundos, para dar tiempo al usuario a reaccionar. En caso de mostrar la notificación por medio de una ventana emergente (pop-up), también hay que ofrecer al usuario la opción de atender el mensaje más tarde.

- Las notificaciones se suelen mostrar mediante una tira, animada o no, que aparece durante unos segundos en la parte superior de la pantalla. Tocándola en aquel momento, la notificación tiene que llevar al usuario a la aplicación o función que ha generado el aviso. También es habitual que...
las notificaciones vengan acompañadas de un sonido para llamar la atención del usuario.

- Algunos sistemas operativos contemplan un espacio donde se van acumulando todos los avisos que se han ido recibiendo. En este espacio, el usuario tiene que poder decidir si quiere consultar el mensaje (normalmente, dando un toque con el dedo encima) o descartarlo definitivamente. Este espacio de notificaciones tiene que ser en todo momento accesible. En este sentido, aun no siendo todavía un estándar, parece que se impone la opción de mostrar el centro de notificaciones haciendo el gesto de arrastrar con el dedo de arriba abajo desde la parte superior de la pantalla, como si bajáramos una persiana.

- Las notificaciones se acostumbran a agrupar de acuerdo con la aplicación o función que las ha generado, para facilitar una lectura rápida.

![Centro de notificaciones en un iPad](Fuente: Apple.com)

### 5.1.3. Paginación

El usuario tiene que poder navegar y saber en todo momento su posición ante un contenido organizado por páginas. En ocasiones, la paginación también puede servir para navegar dentro del contenido y permitir que el usuario salte fácilmente de una página a otra. Disponemos de varias maneras para hacerlo: a partir de puntos, botones, barra de desplazamiento, miniaturas, menús, menú desplegable o menú fijo.

#### Puntos

El sistema más básico, convertido ya en un estándar en los dispositivos móviles, es mostrar la posición a través de unos pequeños círculos o puntos al pie o en la cabecera de la página. Es el sistema habitual en pantallas de inicio o slideshows, donde la navegación se hace deslizando lateralmente con el dedo.
Este sistema no permite al usuario saltar páginas, simplemente le muestra el número de páginas existente (una por cada punto) y su situación en estas (punto destacado). Por lo tanto, resulta un sistema bastante limitado, que conviene no usar cuando tenemos muchas páginas.

**Botones**

Con este sistema el usuario puede pasar páginas hacia delante y hacia atrás mediante botones. Adicionalmente, se muestra el número de página en el que se encuentra. Este sistema es heredero de la web y está en decadencia, en cuanto que el usuario preferirá siempre realizar un gesto sobre cualquier parte de la pantalla antes que ir pulsando un botón.

**Barra de desplazamiento**

Mediante este sistema el usuario puede ver rápidamente su posición respecto del total de páginas del contenido y trasladarse fácilmente adelante y atrás. Para un mejor control, resulta conveniente mostrarle el número de página durante el desplazamiento.
El lector de documentos CloudReaders (iOS) permite navegar rápidamente a través de las páginas con una barra de desplazamiento.

Miniaturas

Este sistema permite navegar por las páginas a la vez que se visualizan las miniaturas, lo que permite identificar rápidamente el contenido.
5.1.4. Menús

Al igual que en cualquier programa informático o sistema operativo de ordenador, los menús son un elemento imprescindible para mostrar las diferentes opciones o funciones disponibles en un dispositivo móvil. La falta de espacio disponible en pantalla, sin embargo, puede condicionar mucho la manera como se accede o se visualizan. Encontramos dos soluciones básicas, que se pueden presentar alternativamente o a la vez en una misma aplicación.

**Menú desplegable**

Para no sacrificar espacio en pantalla, el menú aparece cuando el usuario activa un botón convenientemente identificado o hace un gesto determinado sobre la pantalla (normalmente, arrastrando con un dedo desde un lado):
Menú desplegable a la aplicación de mapas de iOS

En la aplicación de mapas de iOS, el usuario puede acceder al menú “estirando” con el dedo desde la esquina inferior derecha de la pantalla. En este caso no hace falta un botón, puesto que el diseño invita a descubrir qué hay detrás.

A menudo, sin embargo, en las aplicaciones donde se puede mostrar el menú arrastrando desde un lado de la pantalla se suele mantener un botón con la misma función, para garantizar que todos los usuarios, con independencia de sus habilidades haciendo gestos sobre la pantalla, puedan acceder:

Combinación de menú y botón en la aplicación Evernote

La aplicación Evernote (versión para tabletas Android) permite mostrar el menú tanto arrastrando con el dedo desde el margen izquierdo de la pantalla como mediante el botón situado en la parte superior de la pantalla.

También se tiene que prever la manera de volver a “esconder” el menú, ya sea tocando la pantalla fuera de su superficie (para casos de menús que aparecen en ventanas emergentes), activando de nuevo el mismo botón que lo ha hecho aparecer –o uno nuevo fácilmente identificable– o, en caso de usar gestos, haciendo el movimiento contrario al que abrió el menú.
Menú fijo

En este caso sacrificamos parte del espacio disponible para mostrar el menú al usuario en todo momento. Normalmente, este elemento se sitúa arriba o abajo de todo de la pantalla, en forma de barra de navegación (para el usuario de teléfono, el lugar más cómodo será siempre la parte inferior de la pantalla, fácilmente accesible con el pulgar cuando sujeta el dispositivo con una mano). De nuevo, las limitaciones de espacio pueden hacer que el menú fijo se use en combinación con el escondido, de forma que una de las opciones del primero permita el acceso al segundo, como en este ejemplo:

Tal y como indica su nombre, este menú es fijo, es decir, no se puede mover cuando el usuario desplace el resto de la pantalla, ya sea vertical u horizontalmente.

En tabletas, donde el espacio disponible en pantalla es superior, se puede recurrir a uno u otro menú en función de la orientación del dispositivo. Así, en posición horizontal, el menú se puede encontrar siempre visible a un lado de la pantalla, mientras que en posición vertical puede esconderse y mostrarse de nuevo con un gesto para no estorbar la vista principal.
5.1.5. Elementos de control

Dado el reducido espacio de que disponemos en el entorno móvil, el usuario necesitará elementos de control que le permitan lograr rápidamente sus objetivos. En general, estos elementos tendrán que presentar las características siguientes:

- Tener cierto parecido con la función que cumplen: siempre será mejor usar unos botones con “+” o “−” para ampliar o reducir la información mostrada en pantalla que otros etiquetados aleatoriamente. El uso de metáforas, como por ejemplo una lupa, también resulta conveniente para que el usuario identifique rápidamente para qué sirve el elemento de control.

- Ser visibles: Si queremos que el usuario interactúe de forma natural con la interfaz, tiene que ser plenamente consciente de la existencia de los elementos de control a su disposición. Su ubicación tiene que ser la misma en todo momento para no despistar al usuario. En caso de necesidad, se pueden esconder los controles para que no estorben (por ejemplo, escondemos los controles de reproducción cuando el usuario ve un vídeo), pero estos tienen que volver a mostrarse rápidamente en caso de que el usuario los vuelva a necesitar (por ejemplo, cuando vuelve a tocar la pantalla).

- Dar respuesta rápidamente: Los elementos de control tienen que dar señales inequívocas de que han sido activados (cambiando de tamaño o de color, mediante un sonido o vibración, etc.), puesto que de otra manera el usuario puede pensar que no lo ha hecho bien e intentarlo de nuevo, y provocar entradas duplicadas. Si la acción requiere cierto tiempo de procesamiento por parte del sistema, además de cambiar el estado del elemento de control, resulta conveniente mostrarle un indicador de espera.

A continuación presentamos los elementos de control más habituales: desplazamiento, aumento y acceso rápido.

Desplazamiento

Aunque normalmente la función de desplazamiento (scroll) es una función contemplada por el propio sistema operativo para los casos en los que el contenido es mayor que el espacio disponible en pantalla, sí que deben tenerse en cuenta algunos principios a la hora de usar este elemento:

- Siempre que sea posible, mostraremos la barra de desplazamiento para orientar al usuario.
- El tamaño de la barra tiene que ser inversamente proporcional a la cantidad de contenido por el cual estamos navegando (es decir, más corta cuanto más largo sea el desplazamiento, y prácticamente igual de larga que la pantalla cuando el contenido supere por unos cuantos píxeles el tamaño de esta).
• Para no estorbar y ocupar espacio innecesario en la pantalla, la barra puede aparecer solo cuando el usuario está desplazando el contenido, desapareciendo pasados unos segundos de inactividad.
• Por norma general, la barra de desplazamiento no tiene que servir para navegar por el contenido, puesto que el usuario lo hará tocando la pantalla en cualquier punto.
• Es aconsejable limitar el desplazamiento a un solo eje, preferentemente el vertical, puesto que es al que están acostumbrados los usuarios.
• En determinados casos, como por ejemplo cuando se amplía una imagen, los dos ejes (vertical y horizontal) serán igualmente importantes. En estos casos resulta aconsejable mostrar la barra de desplazamiento en ambos sentidos.

• A veces puede resultar útil combinar los dos desplazamientos para mostrar el contenido, de forma que dentro de un desplazamiento vertical se puedan consultar partes de contenido en desplazamiento horizontal.

La inercia en el desplazamiento se ha convertido en un comportamiento esperado por el usuario: con un simple gesto del dedo en una dirección, el contenido tiene que seguir desplazándose hasta que el usuario lo detenga tocando de nuevo la pantalla. Si la pantalla es muy larga, se puede frenar poco a poco la velocidad de desplazamiento, requiriendo un nuevo movimiento por parte del usuario.
Aumento

El aumento (zoom) permite al usuario cambiar el nivel de detalle sobre la información mostrada en pantalla. Se trata de un elemento de control habitual en la visualización de mapas, imágenes, etc. La manera más habitual de resolver esta función, hasta el punto de poderlo considerar ya un comportamiento esperado por parte del usuario, es mediante el gesto de separar dos dedos sobre la pantalla.

Esta solución precisa que el dispositivo sea multitáctil, por lo tanto, para los casos en los que no estemos seguros de si esta tecnología estará disponible, se pueden combinar los gestos con los elementos de control, como en este ejemplo del servicio de mapas de Google visto a través del navegador web:

Alternativamente, se puede usar una barra que, además de permitir cambiar el aumento mediante los botones –o incluso deslizando el dedo por encima de la barra–, informe sobre los niveles de aumento disponibles y zoom actual. Para no estorbar, esta barra se puede mostrar solo cuando el usuario toque la pantalla, haciéndola desaparecer pasados unos segundos.

Acceso rápido

Para ahorrarle al usuario tenerse que desplazar por listados muy largos, se pueden establecer métodos de acceso rápido que lo trasladen de un salto a la parte deseada de la lista. Se trata de un elemento típico en listas ordenadas alfabéticamente, como por ejemplo, la libreta de contactos.
En este ejemplo, el usuario puede desplazarse verticalmente a lo largo de la lista hasta encontrar el contacto deseado, o bien tocar la letra correspondiente al apellido en la barra lateral para encontrarlo más rápidamente.

El propio sistema operativo acostumbra a ofrecer una solución para el acceso rápido en listas, como en este ejemplo de Windows Phone:

a) El usuario puede navegar por el listado de aplicaciones instaladas en su dispositivo desplazándose verticalmente. El listado está ordenado alfabéticamente, y un cuadro con la inicial correspondiente sirve a la vez de separador y de botón de acceso rápido.

b) Tocando una letra, el sistema muestra al usuario todo el alfabeto, lo que le permite acceder rápidamente a las aplicaciones por su inicial, sin necesidad de desplazarse por toda la lista. Las letras sin contenido (es decir, no hay aplicaciones instaladas que empiecen por aquella letra) aparecen oscurecidas, indicando al usuario que no están activas.

El acceso rápido se puede usar tanto en desplazamientos verticales como horizontales, permitiendo al usuario, por ejemplo, saltar rápidamente de páginas en una pantalla de inicio.

5.2. Mostrar información

Pasamos a describir los elementos que sirven para mostrar información.
5.2.1. Listados

Los listados (lists) –también denominados tablas– son la forma más eficiente de mostrar información en formato texto, opciones de un menú o los resultados de una búsqueda, puesto que permiten escanear rápidamente su contenido y hacer selecciones. Pueden incluir más de una línea de texto, aunque no es recomendable pasar de tres para no dificultar la lectura.

El desplazamiento a lo largo de listados es siempre en vertical, y su movimiento tiene que ser suave, pixel a pixel, para permitir un buen control por parte del usuario.

**Listado infinito**

Los listados infinitos son listados donde la información no está almacenada en local y se tiene que ir cargando progresivamente, “fuera de la vista” del usuario. En condiciones ideales, el usuario la percibirá como una lista vertical. Pero se tiene que prever el caso de que el usuario llegue al final del listado y todavía se tengan que cargar más elementos (hará falta un indicador de espera), o se haya producido un error en la carga. Una alternativa es poner un botón al final del listado para permitir la carga de más datos.

**Listado con miniaturas**

Los listados con miniaturas incorporan un elemento gráfico (normalmente en la parte izquierda de la pantalla), que permite al usuario identificar más rápidamente los elementos del listado. Es un recurso habitual para listar contactos, siendo la miniatura en este caso la fotografía o avatar de la persona.

**Listado con foco**

El listado con foco permite ver más información del elemento que tenemos seleccionado dentro de una lista sin necesidad de abrirlo.

5.2.2. Pestañas

Las pestañas (tabs) se suelen usar para mostrar contenidos que están en el mismo nivel jerárquico.
Conviene tener en cuenta los aspectos siguientes:

- Tenemos que mostrar claramente cuál es la pestaña que está activa, cuántas pestañas hay disponibles, y cuál es el contenido detrás de cada pestaña.

- Es aconsejable que cada pestaña tenga un título que permita identificar el contenido que hay detrás de esta. Podemos truncar el texto, pero no es aconsejable (en estos casos podemos hacer, por ejemplo, que el texto vaya circulando por la pestaña como si fuera una marquesina cuando esta está seleccionada).

- Si tenemos más pestañas de las que la pantalla puede mostrar, tenemos que indicar claramente que el usuario puede desplazarlas lateralmente, usando por ejemplo flechas.

- Las pestañas funcionan bien en horizontal, puesto que el usuario no accostumbrá a entenderlas si las presentamos en vertical.

- Hay que seguir las guías de diseño de cada sistema operativo en cuanto al uso y ubicación de las pestañas.

5.2.3. Carrusel

El carrusel (carroussel) sirve para representar imágenes, normalmente en forma de tira (vertical u horizontal), o de círculo, en caso de que el carrusel quiera dar sensación de profundidad. Normalmente, el elemento situado en el centro de la pantalla será el elemento con foco y sobre el cual el usuario podrá interactuar. En este recurso, el movimiento es importante: al desplazarnos entre las imágenes del carrusel, estas se tienen que mover suavemente, acompañando el gesto que hacemos con el dedo. Se trata de un recurso que popularizó el programa iTunes de Apple para mostrar las carátulas de discos.

Vista Cover Flow del programa iTunes de Apple
5.2.4. Cuadrícula

La cuadrícula (grid) sirve para mostrar de forma ordenada (sobre un eje vertical u horizontal) un conjunto de elementos (normalmente, imágenes), sin mostrar ninguna otra información, que en cambio sí se puede revelar cuando el usuario selecciona uno de ellos. Es un patrón habitual en galerías de imágenes.

![Disposición en cuadrícula en la aplicación Instagram (versión iOS) y en una galería de imágenes de Firefox OS](image1)

5.2.5. Pila

La pila de elementos (stack of items) se acostumbra a usar cuando tenemos una serie de elementos, como por ejemplo fotografías o videos que se pueden mostrar en miniatura. En lugar de mostrar las carpetas contenedoras, los mostramos como si estuvieran literalmente amontonados los unos sobre los otros. Es importante, para ser identificada como tal, que la pila esté mal ordenada, es decir, que por debajo de la primera miniatura se puedan ver las esquinas y márgenes del resto de miniaturas. Visualmente, sin embargo, la pila no tiene que ocupar mucho más espacio que el de una sola miniatura.

![Disposición en red en la aplicación Instagram (versión iOS), Disposición en red en una galería de imágenes de FirefoxOS](image2)

Al igual que con los iconos, es aconsejable que las pilas estén etiquetadas para su correcta identificación. Al abrir una pila –normalmente, dando un solo toque sobre el primer elemento de la pila–, todos los elementos pasan a mostrarse sobre una cuadrícula.

Este patrón requiere de buenas animaciones tanto cuando la pila se despliega en una cuadrícula como cuando vuelve al estado original.
5.2.6. **Pase de diapositivas**

El pase de diapositivas (*slideshow*) sirve para mostrar en pantalla completa una serie de imágenes. El usuario pasa de una imagen a la otra haciendo el gesto de deslizar lateralmente con el dedo. Se puede animar ligeramente el cambio de imagen, ya sea desplazando las imágenes hacia los lados o encadenándolas.

5.2.7. **Anotación**

El elemento anotación (*annotation*) nos ayuda a mostrar más información u opciones de un elemento sin abandonar la pantalla en la que nos encontramos. Es un recurso habitual en mapas y gráficas, o en situaciones en las que tenemos varias capas de información que no se pueden mostrar a la vez. Para hacerlo, tenemos dos recursos básicos, la etiqueta expansiva y la franja fija:

a) **Etiqueta expansiva**: en este patrón, cuando el usuario interactúa con el elemento (normalmente, con un toque), la anotación se despliega para mostrar la nueva capa de información.

Ejemplo de etiqueta expansiva
b) Franja fija (banner): aquí la anotación se muestra siempre en el mismo lugar (normalmente, al pie o en la cabecera de la pantalla), y la información varía en función del elemento seleccionado. Se acostumbra a usar cuando hay muchos elementos en pantalla o mucha información para mostrar.

Ejemplo de franja fija

Las anotaciones pueden presentarse en múltiples formatos: solo texto; título, texto e imagen; solo una imagen... Todo depende de la información que se tenga que mostrar. Incluso se pueden incluir enlaces hacia otra pantalla (en el ejemplo anterior, esta posibilidad de navegación entre pantallas se muestra mediante una flecha apuntando a la derecha).

5.3. Entrada de datos

Los diversos elementos que pueden proporcionar entrada de datos en las interfaces para dispositivos móviles pueden ser el teclado, la voz, los campos de texto y formularios, los botones, y los interruptores, barras y selectores.

5.3.1. Teclado

Ya sea para enviar correos electrónicos, mensajes, hacer búsquedas en internet o llenar un formulario, los usuarios necesitan un sistema que les permita escribir con la mayor facilidad y fiabilidad posible, ya sea a través de un teclado físico –cada vez menos habitual en los dispositivos móviles–, o a partir de uno virtual.
a) La disposición del teclado suele venir definida por el sistema operativo, de forma que cualquier cambio puede desconcertar al usuario o dificultarle la entrada de texto al tener que aprender un nuevo sistema. En este sentido, la tradicional disposición impuesta por el teclado numérico de los teléfonos móviles ha dado paso al sistema QWERTY, aunque muchos desarrolladores trabajan en disposiciones alternativas, plenamente adaptadas a las características de las pantallas táctiles.

b) Para facilitar la entrada de texto –sobre todo en teclados táctiles– se pueden ofrecer ayudas, como por ejemplo la autocorrección, sugerencia o autocompletado de palabras de acuerdo con el diccionario definido por el usuario. Se tiene que evitar, sin embargo, que estas ayudas funcionen de forma automática: tiene que ser el usuario quien, en última instancia, acepte la sugerencia o corrección que le ofrece el sistema. En la misma línea, estas ayudas se tienen que poder desactivar totalmente en el caso de que el usuario no las quiera usar.

c) La entrada de texto se hace normalmente pulsando las teclas una detrás de la otra, aunque hay sistemas operativos, como por ejemplo Android, que permiten la entrada de palabras deslizando el dedo por encima del teclado.
d) Aunque puede desconcertar en un primer momento a quienes no están acostumbrados a los teclados virtuales táctiles, ya se ha convertido en un estándar que la presión sostenida (toque largo) sobre una letra dé acceso a sus formas acentuadas o caracteres alternativos:

![Teclado de emoticonos de Windows Phone](image)

e) Debido a las limitaciones de espacio, el teclado se tiene que dividir en diferentes partes o modos: letras, números, símbolos, emoticonos... El acceso a los diferentes modos de teclado se acostumbra a hacer a través de unas teclas convenientemente identificadas en la parte inferior del teclado. Muchas veces el sistema operativo permite predeterminar o limitar los modos de teclado disponibles en función de las características del campo de texto donde se usa, tanto para facilitarle el trabajo al usuario como para evitar errores en la introducción de datos (por ejemplo, en un campo para introducir direcciones de correo se desactiva el teclado de emoticonos).

![Teclado de emoticonos de Windows Phone](image)
5.3.2. Voz

Hasta hace relativamente poco, la idea de poder hablar con las máquinas era cosa de ciencia ficción, como en el clásico 2001: A Space Odyssey, de Stanley Kubrick. Con la mejora de las técnicas de reconocimiento de voz, sin embargo, cada vez es más habitual encontrarse este sistema como una opción más, a menudo junto al teclado, para escribir mensajes, tomar notas o hacer búsquedas en internet.

Si tenemos en cuenta que una de las máximas a la hora de diseñar interfaces móviles es facilitar al máximo el trabajo a los usuarios agilizando la entrada de datos, parece lógico pensar que pronunciar en voz alta una frase siempre será más cómodo que escribirla letra a letra en un teclado virtual. Ahora bien, hay que estar seguro de que la tecnología de reconocimiento de voz funciona muy bien antes de eliminar la opción del teclado. De igual manera, es aconsejable que el usuario pueda editar con el teclado aquello que el sistema haya “entendido” que decía.

Por otro lado, sistemas como Siri de la compañía Apple (o aplicaciones como Sherpa en Android), están llevando el reconocimiento de voz un paso más allá, pasando de la simple transcripción de lo que el usuario ha dicho a interpretar órdenes o dar respuesta a preguntas sencillas. El potencial de esta tecnología es muy elevado, y con el tiempo podría implicar una nueva forma de relacionarnos con las máquinas, y por lo tanto, de diseñar sus interfaces.

A día de hoy, sin embargo, esta tecnología todavía está muy verde, así como los usuarios todavía no estamos muy acostumbrados ni nos gusta demasiado hablar con una máquina: basta ver cómo reaccionamos cuando en lugar de un teleoperador/a nos atiende una máquina en un teléfono de atención al cliente.

5.3.3. Campos de texto y formularios

Al igual que en los ordenadores de sobremesa, los campos de texto son un elemento imprescindible en el entorno móvil para permitir la entrada de texto al usuario, como por ejemplo en un formulario de registro.
Dando un toque con el dedo sobre el campo de texto, este se activa, haciendo aparecer el teclado para que el usuario pueda escribir. Si el sistema operativo lo permite, podemos ahorrar trabajo al usuario mostrándole el modo de teclado más apropiado a los datos que tiene que introducir; por ejemplo, teclado numérico cuando lo que tiene que escribir es un número de teléfono o una cantidad.

Recomendaciones:

- Hay que identificar bien en todo momento cuál es el campo activo, ya sea destacando el borde del marco, colocando un cursor, resaltando el color de fondo o con otro efecto que lo diferencie del resto de campos.

- En caso de formularios, se tiene que tener cuidado y resolver bien el paso de un campo de texto a otro, puesto que es posible que el propio teclado impida activar el siguiente campo con el dedo al superponerse a este. Una manera de solucionar este problema es permitiendo al usuario avanzar y retroceder por los campos de texto directamente desde el teclado, como en este ejemplo de iOS:

![Formulario de registro de la aplicación Runkeeper (iOS)](image)
En presencia de un formulario, el teclado de iOS muestra un juego de botones extra que permite avanzar y retroceder por los campos de texto.

- Para ahorrar espacio en formularios, la etiqueta del campo de texto se puede mover en el interior de la caja o, en caso de dejar la etiqueta fuera, para añadir unas breves instrucciones sobre cómo llenar el campo. En estos casos hace falta que este título o instrucciones se diferencien bien del texto que después introducirá el usuario (normalmente, esto se consigue haciendo aparecer el texto en cursiva y de color gris).

- Siempre que sea posible, es recomendable comprobar si el usuario ha introducido bien los datos o bien en el mismo momento en que lo hace (por ejemplo, indicándole cuántos caracteres le faltan para establecer una contraseña segura) o cuándo salta al siguiente campo, de forma que pueda corregir el error lo más rápido posible.

- Para reducir el tiempo necesario para rellenar un formulario, es conveniente usar listas desplegables, casillas de verificación (checkboxes) y otros selectores que faciliten el trabajo al usuario. Así, por ejemplo, siempre será más rápido elegir un país de una lista que tenerlo que escribir letra por letra.

Lectura para profundizar

Más sobre la validación de formularios en los dispositivos móviles en el artículo siguiente:
Los botones son quizás el elemento más habitual de una interfaz, y nos sirven para iniciar acciones, modificar estados o validar la entrada de datos. Por lo tanto, los botones tienen que ser fáciles de identificar y activar, sobre todo cuando hablamos de dispositivos táctiles. Un recurso habitual es diseñar los botones como si tuvieran relieve o añadiendo sombras. Otras maneras de mostrar su cambio de estado son modificando su color, brillantez, etc.

5.3.5. Interruptores, barras y selectores

Los dispositivos táctiles ofrecen múltiples formas de introducir datos y hacer selecciones, más allá del teclado virtual, más interactivas, basadas en gestos sobre la pantalla. Bien diseñados e implementados, estos controles pueden facilitar el trabajo al usuario a la vez que pueden hacerle más intuitivo y agradable el uso del dispositivo. Cada sistema operativo suele definir sus propios sistemas por defecto, pero nada impide que se puedan diseñar otros nuevos para una aplicación en concreto, puesto que es en este tipo de detalles donde se cuida la experiencia de usuario. En este sentido, es importante cuidar la sensibilidad y velocidad de respuesta del sistema en las acciones que haga el usuario, para evitarle frustraciones.

**Interruptores**

Los interruptores (*switches*) se usan sobre todo en menús de configuración, donde hay opciones que solo permiten dos posiciones (activada y desactivada). En lugar de mostrarle las dos opciones en una lista desplegable, lo que hacemos es presentarle un interruptor, un elemento que reconocerá perfectamente, y que cambiará instintivamente de posición deslizando con el dedo:
Ejemplos de interruptores

a. Interruptor de Android.
b. Interruptor de iOS.

**Barras**

Las barras (*sliders*) se suelen usar cuando tenemos que seleccionar un valor dentro de una escala determinada. Los ejemplos más claros son el control de volumen del dispositivo o de la brillantez de una pantalla. Si antes se usaban dos botones (“+” y “–” para desplazar la barra), ahora es el propio usuario quien puede mover el selector con el dedo por la barra hasta el punto deseado.

Ejemplos de barras

a. Barra de Android.
b. Barra de iOS.

Podemos encontrar barras cumpliendo las más variadas funciones, como por ejemplo, controlar el punto de reproducción de una película o vídeo.
Selector

En lugar de hacer que el usuario introduzca, por ejemplo, una cita escribiendo una fecha y una hora con el teclado numérico, podemos dejar que lo haga girando una rueda o cambiando los números sobre la pantalla del dispositivo.

Ejemplos de selectores

a. Selector de iOS.
   b. Selector de Android.

El selector (*picker*) permite múltiples variantes y diseños, desde la ya clásica rueda giratoria de iOS hasta otros que prescinden incluso de los botones para ajustar los valores, como en este ejemplo de Windows Phone, donde el usuario hace deslizar verticalmente, una a una, las columnas hasta dejarlas en la posición deseada.

Prácticamente no hay límite a la hora de diseñar selectores para la entrada de datos. Las únicas limitaciones, como siempre, son la facilidad de uso y la velocidad de respuesta cuando el usuario los manipula. En este otro ejemplo, de la aplicación para el control de peso True Weight (iOS), la entrada del peso del usuario se hace deslizando lateralmente dentro de lo que realmente parece la ventana de una báscula, una solución clarificadora y fácil de usar:
Otros ejemplos de selectores

5.4. Buscar y filtrar información

Los usuarios quieren poder acceder a la información guardada en sus dispositivos móviles rápida y fácilmente. La función de buscar y filtrar, por lo tanto, es crítica, sobre todo en aparatos con gran capacidad para almacenar información en todo tipo de formatos (documentos de texto, fotografías, canciones, contactos, correos electrónicos, aplicaciones, etc.). En este sentido, los patrones más habituales a la hora de implementar esta función persiguen agilizar el proceso de búsqueda para permitir al usuario lograr su objetivo con el mínimo número de toques sobre la pantalla.

Buscar en el móvil implica introducir texto mediante el teclado táctil, un trabajo no siempre fácil ni agradable para el usuario.

Algunas consideraciones generales sobre la búsqueda en los dispositivos móviles:

a) El elemento más habitual para indicar la posibilidad de hacer una búsqueda es una caja de texto, acompañada de un botón convenientemente identificado para iniciar la acción con la etiqueta “buscar” o un ícono (normalmente, una lupa). Para ahorrar espacio, podemos incluir tanto la etiqueta como el ícono dentro mismo de la caja:
b) Si la búsqueda se hace sobre varios tipos de documentos a la vez, resulta conveniente presentar los resultados agrupados por tipología.

c) Si el usuario ha introducido texto para hacer la búsqueda, hay que considerar resaltar este texto (aunque sean tres letras) en los resultados.

d) En búsquedas relacionadas con la ubicación del usuario sobre el mapa, podemos obtener su posición automáticamente con el GPS para ahorrarle teclear donde se encuentra en el momento, por ejemplo, de buscar rutas.
e) Los resultados de la búsqueda se pueden mostrar en formato lista, con miniaturas, sobre un mapa... En función de los contenidos, podemos ofrecer diferentes formas de visualizar los resultados de la búsqueda.

f) Finalmente, en función de la rapidez del sistema (o de la conexión a la red), habrá que usar patrones de respuesta como el de rueda de espera para advertir al usuario que el sistema está trabajando buscando respuesta en su petición.

5.4.1. Búsqueda dinámica

Con este patrón ahorramos trabajo al usuario evitándole tener que desplazarse por un listado, permitiéndole filtrar los elementos de la lista a medida que va escribiendo en la caja de búsqueda, de modo que tenga suficiente con las tres o cuatro primeras letras para encontrar el elemento que busca. Es un patrón habitual en libretas de contactos o bibliotecas de contenido multimedia.
Ejemplos de búsqueda dinámica

a. Con la búsqueda dinámica basta introducir las primeras tres o cuatro letras para encontrar lo que buscamos, y ahorrarnos tener que desplazarnos por todo el listado de contactos.

b. Búsqueda dinámica con Android: con tres letras, el sistema ofrece resultados para artistas, álbumes y canciones.

5.4.2. Autocompletado

En este patrón, heredado directamente de la web (por ejemplo, Google), en cuanto el usuario empieza a escribir en la caja de búsqueda se le empiezan a sugerir posibles resultados. Ante esto, el usuario puede optar por seleccionar una de las sugerencias que le hace el sistema o continuar escribiendo su búsqueda. El comportamiento de este patrón puede parecer idéntico al de la búsqueda dinámica, pero hay que tener en cuenta que mientras que en la búsqueda dinámica discernimos elementos de una lista –es decir, hacemos un tipo de filtrado–, en el patrón de autocompletado no hay ninguna lista previa, sino que se sugieren términos de búsqueda a medida que el usuario escribe.
5.4.3. Búsqueda con filtros

A veces, el usuario encontrará más rápido aquello que busca si puede filtrar los resultados por tipos de documento, categorías, o bien estableciendo ciertas condiciones que tienen que cumplir los resultados obtenidos. Es un patrón habitual cuando nos encontramos ante gran cantidad de información, donde la búsqueda da demasiados resultados como para encontrar rápidamente lo que buscamos. Este filtrado se puede hacer de forma previa a la búsqueda, o como sistema para refinar los resultados ya obtenidos con una búsqueda más general.
Ejemplos de búsqueda con filtros

La aplicación de elTenedor ofrece cuatro tipos de filtro diferentes para buscar un restaurante.

Las variantes a la hora de presentar este patrón son múltiples, ya sea porque los filtros se ofrecen en la misma pantalla donde el usuario ve los resultados, o bien a través de un formulario, una ventana emergente, un menú desplegable...

La selección de uno u otro sistema obedecerá a menudo tanto a razones de diseño como la finalidad misma de la aplicación (por ejemplo, si la aplicación tiene por objetivo encontrar establecimientos, hace falta que los filtros sean fácilmente accesibles para permitir una búsqueda rápida y efectiva).

A la izquierda, ejemplo de formulario de búsqueda con filtros de la aplicación Appetitool (iOS). A la derecha, filtros sobre la misma pantalla de resultados en la aplicación de Time Out Barcelona (iOS).

5.4.4. Últimas búsquedas

Otra manera de ahorrarle tiempo al usuario es recordarle las últimas búsquedas que ha hecho, por si quisiera repetir alguna. Este patrón se puede combinar con otros, como por ejemplo el de autocompletado, de forma que al situarse
sobre el campo de búsqueda, antes de empezar a escribir, el usuario ve las últimas búsquedas que ha hecho, y es al empezar a escribir cuando se le presentan las opciones de autocompletado.

5.4.5. **Formulario de búsqueda**

En lugar de introducir una palabra o cadena de texto, a veces puede ser más sencillo para el usuario establecer unos criterios o parámetros a través de un formulario. Es un patrón habitual a la hora de buscar vuelos, alojamiento, ... pero que también se puede usar para refinar la búsqueda en grandes bases de datos.

5.4.6. **Búsqueda por geoposición**

La mayoría de dispositivos móviles son capaces de detectar su posición sobre el mapa, hecho que podemos aprovechar para ofrecer búsquedas basadas en la geoposición del usuario. Este es un patrón que se ofrece normalmente como alternativa o complemento a otros métodos de búsqueda más “clásicos”, como por ejemplo la búsqueda por texto o mediante categorías, pero que bien implementado, se convierte en un sistema muy eficaz para ofrecer resultados relevantes. Para acelerar la búsqueda reduciendo el número de resultados sobre el mapa, en este patrón también se pueden usar filtros por categorías, distancia respecto al usuario, etc.

5.4.7. **Ordenar los resultados**

Una vez hecha la búsqueda, es importante decidir cuál será el criterio por defecto con que ordenaremos los resultados. En caso de disponer de varias opciones, podemos dejar que sea el propio usuario quien decida cómo quiere...
ordenar los resultados de su búsqueda. Los patrones en este caso son los mismos que podemos usar a la hora de filtrar (selector en pantalla, formulario, ventana emergente, etc.).

Ejemplo de ordenación de los resultados

Una vez obtenidos los resultados, el usuario de la aplicación eTenedor (iOS) puede ordenar los restaurantes de acuerdo a diferentes criterios a través de un menú desplegable.

5.5. Respuesta al usuario

En este subapartado estudiaremos varios tipos de respuesta al usuario: confirmaciones, mensajes de error e indicadores de espera.

5.5.1. Confirmación

Ante un dispositivo móvil es muy fácil despistarse o tocar algún botón de forma accidental, de forma que hay que prever sistemas para confirmar determinadas acciones por parte del usuario, sobre todo cuando es probable que esté cometiendo un error o descuido. En estos casos, hay que interrumpir la tarea e interrogar directamente al usuario para que confirme su acción.

Por otro lado, los mensajes de confirmación también nos pueden servir para preguntar al usuario qué quiere hacer en aquellos casos en los que la acción iniciada nos lleva a diferentes alternativas, o para informarle de que una tarea ha finalizado correctamente.
Ejemplos de acciones de confirmación

a. Se presentan tres opciones cuando el usuario pulsa el botón “Cerrar” antes de finalizar el envío de un correo (aplicación Gmail).

b. El sistema confirma el deseo del usuario de borrar una aplicación (iOS).

c. Con esta simpática ilustración, la aplicación Dropbox (versión iOS) confirma al usuario la finalización con éxito de la tarea requerida.

Lo más habitual es mostrar al usuario mensajes de confirmación con dos o tres opciones (representadas con botones), aunque a veces se le pueden mostrar mensajes al usuario con una o incluso ninguna opción.

No hay que abusar de los mensajes de confirmación, puesto que pueden crear inseguridad en el usuario y lo frenan en su objetivo de cumplir una tarea lo más rápido posible.

Es importante describir bien el motivo del mensaje, así como etiquetar correctamente los botones que expresen las opciones disponibles, que nunca tendrán que ser “Sí” o “No”, sino describir la acción que se llevará a cabo si el usuario lo elige.

5.5.2. Mensajes de error

Aunque indeseables, los mensajes de error son imprescindibles en cualquier interfaz, ya sea para dar salida ante comportamientos inesperados del sistema como para alertar al usuario cuando ha hecho algo mal.

Como norma general, los mensajes de error tienen que cumplir las características siguientes:

- Ser visibles. Lo peor que puede pasar es que algo vaya mal y el usuario no se entere.
• Estar escritos en un lenguaje conciso y claro.

• Ser educados, y no culpar al usuario por el error.

• Describir exactamente qué ha ido mal.

• Dar indicaciones claras sobre cómo resolver el error, y si es posible, ayudar a corregirlo automáticamente.

Ejemplo de mensaje de error poco adecuado

Este mensaje de la aplicación Wunderlist (versión por Android) es un buen ejemplo de cómo no se deben dar los mensajes de error, puesto que no describe ni qué ha pasado ni cómo solucionar el problema detectado.

Los errores a la hora de introducir usuario y contraseña son muy habituales cuando el usuario usa un teclado táctil.

En los dispositivos móviles, lo más habitual es mostrar los mensajes de error abriendo un diálogo, aunque no es la forma más idónea, puesto que a menudo la propia ventana emergente del mensaje de error puede estorbar la vista de dónde está el error.
5.5.3. Indicadores de espera

A medida que mejoran el hardware y las conexiones, el usuario le pide cada vez más a su dispositivo. Pero por mucho que avance la tecnología, hay algo que no varía: a veces hay que esperar para obtener aquello que queremos. Para evitar frustraciones, tenemos que ser capaces de mostrar claramente que el dispositivo está procesando información, cargándola o simplemente buscando lo que se le ha pedido.

Los indicadores de espera son un elemento habitual en cualquier sistema operativo (también en los móviles), pero muchas veces habrá que diseñar uno nuevo, adaptado a las necesidades de la aplicación.

Los tres patrones básicos que presentamos aquí se pueden combinar y usar a la vez. Por ejemplo, podemos cargar progresivamente las imágenes a la vez que superponemos una rueda de espera o mostramos una barra de progreso para toda la página.

En todo caso, los indicadores de espera tienen que estar siempre animados: el usuario interpretará a menudo un indicador inmóvil como que ha habido un problema. Así, si usamos una barra de progreso, además de hacerla avanzar a medida que se completa la tarea, es aconsejable que la propia barra tenga una animación interna que muestre actividad, sobre todo en casos de cargas muy largas donde la barra avanza muy lentamente.

Finalmente, y sobre todo si la espera ha sido muy larga, es probable que el usuario ya no esté mirando la pantalla cuando la tarea se haya completado. Es conveniente, pues, alertar al usuario con un sonido, una vibración, volviendo
a iluminar la pantalla, etc., o una combinación de estos junto con un mensaje (por ejemplo, un diálogo) que deje muy claro al usuario que el proceso ha finalizado.

**Rueda de espera**

La rueda de espera se usa para esperas muy cortas o bien cuando no sabemos seguro cuánto durará la espera. Se suele presentar como un gráfico sin final.

**Barra de progreso**

La barra de progreso (*progress bar*) nos muestra el total que se tiene que alcanzar como una barra vacía que va llenando poco a poco a medida que avanza la carga. Se puede acompañar de un contador que muestre el porcentaje de carga alcanzado o el tiempo restante para completar la tarea. Este patrón es habitual en esperas que pueden ser largas, como por ejemplo la descarga de datos. En acciones que impliquen esperas potencialmente largas, es conveniente ofrecer un botón para detener o cancelar la operación.

Ejemplos de barra de progreso

a. Barra de progreso combinada con rueda de espera; el App Store (iOS) combina tanto la barra de progreso como una rueda de espera para indicar que las aplicaciones se están instalando.

b. Barra de progreso con indicador de porcentaje de la aplicación Travoza (Android).

**Carga progresiva**

En el patrón de carga progresiva es el propio contenido el que hace de indicador de espera. Es un recurso habitual con imágenes, que van pasando de borrosas a muy definidas a medida que se cargan.
5.6. Ayuda

Las interfaces móviles tienen que ser fáciles de usar, intuitivas, pero aun así, siempre se tiene que ofrecer algún tipo de ayuda al usuario, sobre todo cuando es la primera vez que abre una aplicación. Los patrones para mostrar la ayuda son variados, y cada día encontramos formas más creativas de ofrecer esta información al usuario.

5.6.1. Diálogo

Los diálogos (*dialog boxes*) se presentan en formato texto, a través de una ventana que se cierra pulsando un botón (normalmente, “Aceptar”, “Continuar”, “De acuerdo”...). Es recomendable que el mensaje con las instrucciones sea lo más breve y claro posible, puesto que el usuario muchas veces lo leerá por encima.

Ejemplo de diálogo

A través de este diálogo, la aplicación para iOS de Privalia indica al usuario dónde se encuentra la opción que permite cambiar de tienda.
No hay que abusar de los diálogos, puesto que se trata de un sistema muy agresivo que interrumpe la tarea que estuviera llevando a cabo el usuario y no lo deja seguir hasta que cierra la ventana.

Por otro lado, y ante el riesgo de que el usuario cierre por error la ventana sin haberle dado tiempo a leer el contenido, es recomendable que este mensaje se pueda recuperar en otro punto de la aplicación (por ejemplo, a través del menú, en un apartado general de ayuda).

5.6.2. Consejo

El consejo (tip) se asemeja mucho al diálogo, pero se diferencia de aquel en el hecho de que el consejo está más integrado a la interfaz, normalmente en forma de globo, que apunta hacia el elemento sobre el que da información. Suele presentarse solo en formato texto, y lo más habitual es que los consejos desaparezcan en cuanto el usuario empieza a interactuar con la pantalla o toca el elemento que explican.

Ayuda en formato ‘tip’ en la aplicación TripAdvisor (Android)

Los consejos se muestran normalmente la primera vez que el usuario usa la aplicación, para ayudarlo a identificar las funciones principales.
5.6.3. Instrucciones

Las instrucciones (*How To*) son la forma que más se acerca a los manuales de instrucciones de toda la vida. En función de su extensión se pueden presentar en una sola página o en forma de menú o listado navegable. Siempre que se pueda, resulta conveniente usar una combinación de texto, capturas de pantalla, imágenes, gráficos...

La aplicación para compartir archivos Dropbox (en este ejemplo, iOS) usa un menú de ayuda para explicar las diferentes funciones disponibles (izquierda). En el detalle de las explicaciones (derecha) se combina texto con capturas de pantalla.
5.6.4. Visita guiada

Con este sistema el usuario puede hacer un recorrido rápido sobre las principales funciones y características de la aplicación. Es habitual mostrar la visita guiada automáticamente la primera vez que el usuario abre la aplicación, pero también es recomendable que esté disponible para futuras consultas.

La visita guiada (tour) puede combinar texto, capturas de pantalla, imágenes, gráficos... La forma más habitual de presentarla es a través de un pase de diapositivas, que hay que mantener lo más breve posible.
Ejemplos de visita guiada

a. Las aplicaciones de Gmail (izquierda) y Nike Training de iOS (derecha) usan una combinación de texto y capturas de pantalla en sus visitas guiadas.

b. Ayuda en formato visita guiada de la aplicación Tweetbot (versión iPad), donde se explican los diferentes gestos que puede hacer el usuario.

5.6.5. Transparencias

Este sistema consiste en colocar una capa semitransparente con ilustraciones explicativas encima de la interfaz de la aplicación, de forma que señalen y expliquen su funcionamiento. Es una manera rápida y visual de ofrecer ayuda (en este sistema suelen primar las ilustraciones por encima del texto). Como en el caso de los consejos, esta capa de ayuda tiene que desaparecer después de que el usuario toque la pantalla.
Ejemplos de transparencias

Aplicaciones como Snapseed (izquierda) o Pulse (derecha) usan las transparencias para mostrar la ayuda al usuario (ejemplos Android).

5.6.6. Video ayuda

Este sistema aprovecha las capacidades multimedia de los dispositivos móviles, y, bien realizado, es quizás la mejor manera de explicar el funcionamiento de una aplicación puesto que permite mostrarla en acción tal y como después la usará el usuario.

Ejemplo de video ayuda

La primera vez que el usuario abre Layar se encuentra con un mensaje de bienvenida, un breve consejo y un vídeo que le explica cómo usar la aplicación (iOS).
Se tiene que intentar que los videos sean breves (si no es posible, se puede optar por hacer un menú de instrucciones donde cada función esté explicada con un breve clip de video) y expliquen las funciones desde el punto de vista del usuario. También es conveniente ofrecer los controles de reproducción básicos para el video (reproducción, pausa, volumen, etc.).
6. Desarrollar aplicaciones para dispositivos móviles

Con la popularización de teléfonos inteligentes y tabletas ha aparecido un nuevo sector dedicado al desarrollo de aplicaciones (*apps*), estas pequeñas –o no tan pequeñas– piezas de software que amplían las capacidades del dispositivo y ayudan a los usuarios en las más variadas tareas cotidianas. Grandes y pequeñas empresas tecnológicas (estas últimas también denominadas *startups*), emprendedores, desarrolladores independientes, todo el mundo quiere entrar en este negocio que a primera vista parece la nueva tierra prometida desde el estallido de la burbuja punto com.

Pero al igual que con cualquier producto o servicio tecnológico, antes de animarnos a diseñar y programar una aplicación hay que tener muy claro cuáles son nuestros objetivos y cómo queremos alcanzarlos. Cuanto más claras tengamos las ideas de entrada, más rápido será el proceso de diseño y desarrollo de la aplicación.

A continuación se describen las fases básicas que tenemos que seguir a la hora de desarrollar una aplicación para dispositivos móviles.

6.1. Fase de conceptualización

Antes de empezar a diseñar una aplicación –y sobre todo, antes de empezar a escribir la primera línea de código– lo primero que hace falta es tener muy claro qué es lo que queremos que haga y a quién está dirigida. Puede parecer obvio, pero a menudo el enorme potencial de los dispositivos móviles hace que empecemos a añadir funciones a la aplicación simplemente “porque se puede hacer”, y no porque realmente hagan falta. Esta trampa, muy difícil de evitar en algunos momentos del desarrollo, puede acabar provocando que el usuario no sepa muy bien para qué sirve nuestra aplicación o la encuentre demasiado complicada de usar.

Una manera de evitar esta trampa es empezar respondiendo unas cuantas preguntas, que nos ayudarán a recordar nuestros objetivos y a guardar registro de los inicios de nuestra idea en caso de que perdamos el norte durante la fase de desarrollo.

6.1.1. Definición de las funciones

¿Qué hará nuestra aplicación? ¿Para qué servirá? ¿Qué necesidad cubre? Estas son solo algunas de las preguntas que nos tenemos que plantear cuando empezamos a definir qué queremos hacer. Es importante conseguir responder a estas preguntas solo con una frase, de la manera más breve posible. Así nos
aseguraremos de que nuestra aplicación esté bien enfocada y será fácil de entender cuando la presentemos al público. No hay que ser muy genérico, simplemente apuntar la idea general que mejor defina nuestra aplicación.

Para ayudarnos en este propósito, es conveniente hacer una lista con todas las funciones que nos gustaría que tuviera. Alternativamente, podemos pensar en los objetivos que tendrán los usuarios cuando la usen (por ejemplo, capturar y compartir imágenes, enviar mensajes, etc.). Resulta aconsejable ir ordenando estas listas según la importancia que demos a las funciones, lo que nos ayudará a discriminar las que constituyen el corazón de nuestra aplicación.

6.1.2. Definición del usuario y el contexto

Una vez tengamos apuntadas todas las funciones que podría tener nuestra aplicación, llega el momento de plantearnos cuál será su perfil de usuarios. ¿Quién usará nuestra aplicación? ¿Cuáles son sus intereses? ¿Y sus motivaciones? ¿Qué esperarán los usuarios de nuestra aplicación cuando la usen?

Para afinar todavía más con nuestra aplicación, también resulta necesario preguntarnos en qué contexto se usará: ¿En el sofá de casa? ¿En la calle? ¿En el transporte público de camino al trabajo? Si bien es cierto que los dispositivos móviles se caracterizan por su ubicuidad, en muchos casos podremos prever las situaciones de uso más habituales para nuestra aplicación.

Así, por ejemplo, es poco probable que una aplicación con recetas de pastelería paso a paso se use en mitad de la calle, pero sí en situaciones donde el usuario quizás no tendrá las manos en condiciones de tocar el dispositivo. En este caso, se puede prever la posibilidad de pasar al siguiente paso de la receta sin tocar el aparato, dando una orden con la voz. En cambio, si tenemos pensado que nuestra aplicación se use mientras el usuario hace deporte (corriendo, en bicicleta...), quizás será conveniente reducir el número de botones y hacerlos mucho más grandes para evitar errores cuando manipule la interfaz.

Si el perfil de usuario nos queda demasiado amplio, es conveniente decidir cuáles son las tres o cuatro características que mejor lo definen, y descartar el resto.

6.1.3. Descripción de la aplicación

Una vez tenemos el perfil de usuario bien definido, habrá que filtrar todas las funciones descritas en el primer paso según este perfil. En este punto, hay que tener presente que siempre será preferible que nuestra aplicación haga una sola cosa, pero bien, a que haga cinco pero regular. Por lo tanto, no nos tenemos que preocupar si una vez pasado el filtro ¿es realmente importante esta función? ¿es lo que necesita el usuario?), reducimos la lista inicial a dos o tres funciones.
Llegados a este punto ya estaremos en condiciones de describir nuestra aplicación con una sola frase que resuma tanto qué hace como a quién va dirigida. Esta descripción nos servirá para no desviarnos del propósito inicial una vez empiece el desarrollo de la aplicación: siempre que aparezcan dudas sobre su diseño o la incorporación de nuevas funciones tendremos que volver a los dos primeros puntos y preguntarnos si nos ayuda a alcanzar nuestro objetivo y si será realmente útil para nuestro perfil de usuarios.

6.2. Fase de diseño

6.2.1. 10 consejos para un buen diseño

A continuación damos una receta de diez consejos que hay que seguir para obtener un buen diseño.

I. Una pantalla para cada cosa

Una pantalla llena de botones y mensajes despistará al usuario y le hará descentrarse sobre lo que quiere hacer. Tenemos que analizar bien qué pondremos en cada pantalla, y al mismo tiempo, preguntarnos si es lo que el usuario necesitará.

II. Piensa de arriba abajo

La parte superior de la pantalla es la más visible y, por lo tanto, la más importante del dispositivo. Sobre todo en los teléfonos móviles, por donde el usuario suele sujetar el dispositivo:

- con una sola mano y usando el pulgar de la misma mano para tocar la pantalla, o bien;
- con una sola mano y usando el índice de la otra mano para tocar la pantalla, o bien;
- con las dos manos y usando los pulgares de ambas para tocar la pantalla.

Las formas de sujetar una tableta son mucho más variadas (con dos manos, sobre las piernas, en un soporte...), pero tenemos que tener claro que, sea un teléfono o una tableta, el usuario siempre escaneará la pantalla de arriba abajo, de modo que habrá que poner lo más importante en la parte superior.

III. Establece relaciones y jerarquías entre pantallas

Cuando empezamos a diseñar, hay que trabajar también el árbol de navegación de la aplicación, es decir, plantear los posibles caminos de pantallas que podrá recorrer el usuario. Es importante tener muy definido este árbol antes
de empezar con la fase de programación, puesto que nos ayuda a establecer jerarquías entre las diferentes pantallas y funciones de la aplicación, y evita que el usuario pueda acabar en algún atolladero.

IV. Haz que el uso sea lo más obvio posible

Tenemos que conseguir que el usuario no tenga que pensar mucho para entender qué hace ni cómo tiene que interactuar con nuestra aplicación. Esto lo podemos conseguir de varias maneras:

- Situando en primer plano la función principal de la aplicación.
- Minimizando el número de botones y controles disponibles.
- Etiquetando bien los botones.
- Siguiendo los estándares que marca el sistema operativo.

V. Minimiza el esfuerzo para introducir datos

Las mejores aplicaciones son las que facilitan el trabajo a los usuarios, de forma que si conseguimos que introduzcan datos tocando un solo botón lo estaremos haciendo mejor que si tienen que tocar dos. Selectores, listas o accesos rápidos son maneras sencillas de facilitar la entrada de datos. Otras maneras son los códigos QR, sistemas de geolocalización o los sistemas de reconocimiento de voz.

a. La aplicación Google Translator (versión iOS) permite hacer traducciones usando un sistema de reconocimiento de voz.
b. Captura de un código QR con un iPhone.
VI. Diseña los controles pensando en dedos grandes

En una interfaz táctil somos nosotros quienes diseñamos los botones y controles que después usarán los usuarios en las más variadas situaciones (sentados en un taxi, andando por la calle, en el gimnasio, etc.). Tenemos que tener en cuenta que, al manipularla, el usuario tapa la pantalla con su propia mano, hecho que, unido a la falta de relieve de los elementos con los que interactúa, hace que a veces resulte bastante difícil activar botones y controles.

Apple recomienda un tamaño mínimo de 44 × 44 puntos para cualquier elemento seleccionable con el dedo.

Hay que diseñar, pues, pensando en dedos grandes, lo que implica tener en cuenta no solo el tamaño sino también la separación entre los diferentes elementos seleccionables. La receta para el desastre en este punto es diseñar una aplicación llena de botones pequeños y muy juntos.

VII. Sé breve y clarificador

Cuando se trata de comunicarnos con el usuario, tenemos que intentar ser lo más breves y concisos que podamos. Los usuarios quieren hacer cosas, no pasar el rato leyendo mensajes o instrucciones. Algunos consejos básicos:

- **Sé breve**: si puedes decirlo en 30 caracteres, no lo hagas con 50.
- **Hazlo fácil**: el usuario no entiende el lenguaje de los informáticos. Usa palabras cortas y verbos clarificadores.
- **Sé amable**: procura no usar expresiones como *Error* o *Cuidado*, y sí en cambio expresiones que describan de forma sencilla qué ha pasado.
- **Lo más importante, al principio**: las dos o tres primeras palabras del mensaje tienen que incluir ya el motivo del mensaje.
VIII. El contenido, siempre en primer plano

Si trabajamos con contenidos, tenemos que hacer que estos sean los auténticos protagonistas de nuestra aplicación. Todo el resto (interfaz, botones, animaciones...) tiene que estar a su servicio. Podemos conseguirlo de varias maneras:

- Minimizando el número de controles o disimulándolos para que no estorben la visión del contenido.
- Haciendo aparecer los controles solo cuando el usuario toca la pantalla.
- Haciendo que los contenidos sean la misma interfaz.

IX. Dedica atención a los pequeños detalles

Como hemos visto anteriormente, el usuario cada vez valora más aquello que siente y experimenta cuando usa una aplicación. Un diseño atractivo, rapidez en la respuesta, unas animaciones fluidas y muy logradas, formas diferentes y creativas de navegar por los contenidos... Conseguir una buena experiencia de usuario es cuestión de resolver bien los objetivos básicos de usabilidad, pero también de dedicar atención a los pequeños detalles de nuestra aplicación, desde el ícono hasta el sombreado de los botones.

X. Las normas están para saltártelas

El diseño de aplicaciones para teléfonos inteligentes y tabletas todavía está en los prolegómenos, y cada día aparecen nuevas formas de presentar y navegar por contenidos, acceder a la información o compartir conocimiento.

Los patrones que hemos visto ofrecen soluciones estándar a los problemas más habituales cuando diseñamos una interfaz móvil, pero no son ni mucho menos la única solución posible. A veces, hay que saltarse las normas establecidas para poder avanzar. Solo así han podido aparecer aplicaciones como Clear (iOS), que exploran nuevos caminos y sistemas para interactuar con una pantalla táctil.
6.2.2. Usabilidad y experiencia de usuario

Como hemos visto, cuando diseñamos cualquier tipo de sistema interactivo tenemos que poner las necesidades del usuario por delante de todo, incluso de su estética. Cada vez más, sin embargo, los usuarios piden que, además de ser fáciles de usar, las interfaces con las que interactúan sean visualmente atractivas, de tal manera que “apetezca” usarlas.

Por lo tanto, a la hora de diseñar una aplicación distinguiremos entre dos conceptos que están íntimamente ligados. Por un lado, la usabilidad, definida como la propiedad que hace que un producto sea fácil de usar; y por otro, la experiencia de usuario, que es la manera como el usuario percibe y experimenta sensaciones cuando ve y manipula el producto.

Los objetivos básicos de la usabilidad son:

a) **Eficacia**: Se trata del objetivo más básico, puesto que persigue que el sistema cumpla la tarea para la cual ha sido diseñado.

b) **Eficiencia**: Lograremos este objetivo cuando el sistema cumpla la tarea requerida con rapidez, dando el menor número de pasos.

c) **Seguridad**: Una interfaz se considera segura cuando minimiza las posibilidades de que el usuario cometa un error y dispone de mecanismos de recuperación de la información para los casos en los que el error se produce. Los diálogos de confirmación o la opción de deshacer la última acción son mecanismos de seguridad.

d) **Utilidad**: Hace referencia a las funciones disponibles en la aplicación, y que permiten que el usuario pueda hacer más o menos cosas con ella. Como hemos visto, en el entorno móvil hay que limitar al máximo el número de funciones disponibles sin que la aplicación pierda su utilidad.

e) **Facilidad de aprendizaje**: Por norma general, los usuarios de dispositivos móviles se rigen por la ley del mínimo esfuerzo, de manera que conviene no diseñar aplicaciones que requieran muchas horas de aprendizaje para empezar a usarlas. Esto se puede conseguir minimizando el número de funciones disponibles, automatizando procesos y situando en primer término solo aquellas opciones imprescindibles para completar la tarea principal. Hay que señalar que la facilidad de aprendizaje no es sinónimo de simplicidad: podemos diseñar aplicaciones complejas pero con una curva de aprendizaje muy suave, de forma que el usuario se encuentre cómodo y ya obtenga resultados pasados unos cuantos minutos.
f) Memorabilidad: Hace referencia a la capacidad de recordar cómo funciona un sistema una vez hemos aprendido a usarlo. Resulta importante en aplicaciones o funciones que se usan poco.

En cuanto a la experiencia de usuario, resulta evidente que estamos hablando de un hecho totalmente subjetivo, difícilmente medible. Puede variar mucho en función de la persona, su estado de ánimo, experiencias previas o expectativas que pueda tener. No es de extrañar, por lo tanto, que la experiencia de usuario sea un campo multidisciplinar donde tienen cabida psicólogos, antropólogos, sociólogos, ingenieros, diseñadores gráficos, diseñadores industriales, comunicadores, creativos, etc.

La experiencia de usuario alcanza tanto el momento concreto de uso de la aplicación como la imagen previa que se había hecho el usuario y el recuerdo posterior que le queda.

Aplicada a los dispositivos móviles, una buena experiencia de usuario pide cada vez más el desarrollo de aplicaciones que sean divertidas, entretenidas, sorprendentes, motivadoras, satisfactorias, agradecidas o estéticamente atractivas, a la vez que logran con mayor o menor medida los objetivos de usabilidad antes descritos.

6.2.3. Prototipado

Antes de empezar con la fase de programación, es muy recomendable crear prototipos de nuestra aplicación. En primer lugar, los prototipos nos ayudan a explorar diferentes soluciones a los problemas que irán apareciendo durante el desarrollo de la aplicación, y nos ayudan a encontrar la mejor solución posible.

En segundo lugar, los prototipos también nos ayudan a detectar carencias en nuestra aplicación, y corregirlas antes de haber escrito una sola línea de código. En este sentido, un buen prototipo facilitará el trabajo al desarrollador, y ahorrará tiempo y costes de programación.

Finalmente, los prototipos también nos servirán para poner a prueba la aplicación y hacer tests de usabilidad con usuarios potenciales. De hecho, aunque enseñemos el prototipo solo a unos cuantos amigos, a menudo una mirada fresca, sin contaminar, sobre nuestro trabajo nos ayuda a mejorarla sustancialmente y a afinar el resultado final sobre las expectativas reales del usuario.

Podemos distinguir entre dos tipos de prototipos: baja fidelidad y alta fidelidad, en función de su correspondencia con el resultado final.
**Baja fidelidad**

Los prototipos en baja fidelidad (*wireframes*) se suelen hacer en las primeras etapas de desarrollo, justo cuando empezamos a pensar en cuáles serán las funciones de la aplicación. Se pueden empezar a hacer con lápiz y papel, sin preocuparnos mucho por el diseño, puesto que en esta fase lo más importante es visualizar qué es lo que realmente queremos conseguir y cómo queremos hacerlo.

Para ayudarnos a visualizar todavía mejor lo que hacemos, podemos elaborar estos primeros dibujos dentro del marco de un dispositivo móvil para hacernos una idea de los tamaños y proporciones de los elementos que colocamos en pantalla. En la red se pueden encontrar multitud de plantillas imprimibles – a tamaño real– tanto de teléfonos como de tabletas, que facilitarán nuestro trabajo.
Plantillas imprimibles a tamaño real

En baja fidelidad también se puede empezar a trabajar el árbol de navegación de la aplicación, donde planteadremos todos los posibles caminos de pantallas que podrá recorrer el usuario.
“SimplePocket WireFrames”

Alta fidelidad

Los prototipos en alta fidelidad (mockups) suelen ser más o menos interactivos, lo que nos permite hacernos una idea bastante acertada de cómo funcionará la aplicación. Pueden incluir la capa de diseño definitiva, de manera que a simple vista puede costar diferenciar un mockup de la aplicación real misma.

Durante el desarrollo, los prototipos en alta fidelidad nos servirán por testear nuestras ideas más allá del papel. Llegado el momento, también nos servirán para hacer pruebas con usuarios reales, detectando problemas de concepto, diseño o usabilidad antes de entrar en la fase de desarrollo.

Un prototipo en alta fidelidad requiere tiempo. Cuanto más realista sea, más tiempo le habremos dedicado, de forma que hay que encontrar el equilibrio entre realismo y funcionalidad.

Para hacer un prototipo en alta fidelidad no es necesario tener conocimientos de programación. Cada vez hay más programas y aplicaciones que permiten hacerlos de forma intuitiva, como quien va encajando piezas en un puzzle. Los
más sofisticados permiten incorporar animaciones y otros efectos que proporcionan más realismo al prototipo. Los hay que funcionan sobre un navegador web y también otros que se pueden instalar o probar directamente sobre el dispositivo.

Algunos ejemplos de programas para hacer prototipos:

a) **Balsamiq** (www.balsamiq.com): es uno de los clásicos, que empezó como herramienta de prototipado de páginas web y ahora también para aplicaciones móviles. Permite crear y compartir prototipos de forma fácil y rápida, simplemente arrastrando y colocando los elementos desde el menú de herramientas superior. El resultado final parece como si se hubiera realizado a mano, lo que nos permite centrarnos más en la estructura de navegación que en el diseño.

![Captura de pantalla de la versión web de Balsamiq](image)

Se puede instalar en el ordenador (el coste de la licencia para un usuario es de 79 dólares) o usarlo directamente en su web (el coste de suscripción básico es de 12 dólares al mes).

![Captura de pantalla de Balsamiq](image)

Fuente: "Intro to Balsamiq Mockups", YouTube (accesible en línea)

b) **Axure** (www.axure.com): uno de los más potentes y versátiles, usado por muchos profesionales. Permite crear prototipos con gran nivel de detalle e interacción, hasta el punto de parecer réplicas del original. Contiene bibliotecas
especificas para hacer prototipos de aplicaciones, desde donde se van arrastrando y colocando los elementos. La web de Axure contiene multitud de tutoriales que enseñan a sacar partido rápidamente de las funciones del programa.

En el caso de iOS, los prototipos hechos con Axure se pueden visualizar directamente sobre un iPhone o iPad, como si fueran una aplicación de verdad. La licencia básica tiene un precio de 289 dólares.

c) Justinmind (www.justinmind.com): esta empresa catalana trasladada a Silicon Valley ofrece un programa que permite hacer y compartir prototipos de webs y aplicaciones totalmente interactivos. Ofrece bibliotecas de elementos específicas para dispositivos iPhone, iPad, Android, Windows Phone, BlackBerry... El precio de la suscripción anual es de 228 dólares, mientras que la licencia cuesta 495 dólares por usuario.
6.3. Fase de desarrollo

Hay varios tipos de aplicaciones que nos pueden ser útiles en la fase de desarrollo de aplicaciones para dispositivos móviles.

6.3.1. Aplicaciones nativas

Las aplicaciones nativas son aquellas que se han desarrollado en el lenguaje propio del sistema operativo usando el SDK (de la expresión inglesa software development kit) correspondiente.

Ventajas:

- Proporcionan pleno acceso a todas las funcionalidades del dispositivo: agenda, cámara, brújula, acelerómetro, giroscopio, etc.
- Ofrecen un mejor rendimiento.

Inconvenientes:

- Hacen falta programadores expertos para explotar al máximo las posibilidades del dispositivo.
- Presentan un coste de desarrollo más elevado si queremos la misma aplicación en dos o más sistemas operativos, puesto que podremos reaprovechar muy poco del trabajo hecho.
6.3.2. Aplicaciones web

Las aplicaciones web (web apps) son aplicaciones programadas para funcionar sobre el navegador de internet del dispositivo. En el fondo, son como una página web que “imita” el comportamiento de una aplicación nativa. Se suelen programar usando HTML5, la evolución del lenguaje de programación clásico de la web, mucho más versátil y potente. También usan Javascript para las interacciones con el usuario.

Ventajas:

- Las aplicaciones web pueden ser fácilmente multiplataforma, puesto que al haber sido desarrolladas para funcionar en un navegador de internet, en principio se pueden ejecutar en cualquier dispositivo.
- Tienen un coste de desarrollo más bajo, puesto que el mismo código puede servir con pequeñas modificaciones para todos los sistemas operativos.
- Basta con un buen programador web para desarrollarlas.
- Las actualizaciones se pueden hacer de forma automática, sin necesidad de que el usuario se vuelva a descargar la aplicación.
- Se pueden distribuir fuera de los mercados oficiales, y evitar así las comisiones que cobran los gestores de los mercados (por ejemplo, Apple se lleva un 30% de cada venta que se hace a través de la App Store).

Inconvenientes:

- Las interfaces suelen ser más pobres, con un diseño que tiene que ser válido para cualquier dispositivo.
- Su funcionamiento es más lento.
- Permiten un acceso muy limitado a las prestaciones del teléfono.
- No permiten notificaciones push.

6.3.3. Aplicaciones híbridas

Las aplicaciones híbridas tienen parte de su código escrito en HTML 5 y Javascript, pero dentro de un envoltorio de código nativo que les permite tener acceso a todas las funcionalidades del dispositivo, como si se tratara de una aplicación nativa.
En principio, las aplicaciones híbridas se quedan con lo mejor de ambos mundos: la “facilidad” de programación de la web y las prestaciones que da desarrollar bajo el paraguas del código nativo.

Fuente: Adaptado de “Editing Native, HTML5, or hybrid: Understanding your mobile application development options”, Developer force
Bibliografía

Bibliografía básica


Bibliografía complementaria


