Guía de desarrollo, Anexo 29

Guía de usabilidad y accesibilidad

Autor: Oficinas de Calidad y Pruebas

GERENCIA INFORMÁTICA

JOSEFA VALCÁRCEL, 44
28027-MADRID

Índice General

1 INTRODUCCIÓN 4

1.1 OBJETIVO 4
1.2 AUDIENCIA 4
1.3 ESTRUCTURA DEL DOCUMENTO 4
1.4 GLOSARIO 4
1.4.1 Términos 4
1.4.2 Acrónimos 5

2 REGLAS GENERALES 6

2.1 ACCESIBILIDAD MÓVIL 6
2.1.1 Introducción 6
2.1.2 Normativa de Accesibilidad 7
2.1.3 Accesibilidad a contenidos web desde dispositivos móviles 7
2.2 BUENAS PRÁCTICAS PARA LA WEB MÓVIL 8
2.3 USABILIDAD EN DISPOSITIVOS MÓVILES 15
2.3.1 Definiciones de usabilidad 15
2.3.2 El contexto en un entorno móvil 16
2.3.3 Características propias del entorno móvil 20
2.3.4 Diseño adaptable 22
2.4 DIRECTRICES PARA APLICACIONES MÓVILES 24
2.5 DISEÑO DE INTERFAZ 27
2.5.1 Recursos y herramientas para el diseño de interfaz 34

3 DIRECTRICES DE DISEÑO EN ANDROID 38

3.1 SIMPLICIDAD DEL DISEÑO 45
3.2 PRESENTACIÓN MULTI-PANTALLA 46
3.3 VISTAS COMPUESTAS Y CAMBIO DE ORIENTACIÓN 48
3.4 WIDGETS 50

Índice de Ilustraciones

Ilustración 1: Usabilidad -> ergonomía 18
Ilustración 2: Usabilidad -> tipos de pantallas 18
Ilustración 3: Usabilidad -> gestos 19
Ilustración 4: Elementos principales de interacción 21
Ilustración 5: Usabilidad -> tipos de pantallas 22
Ilustración 6: Ejemplo de diseño adaptable de la web de Antonio Gaudí 23
Ilustración 7: Ejemplo de wireframe 29
Ilustración 8: Ejemplo de jerarquía 30
Ilustración 9: Ejemplo de estructura 31
Ilustración 10: Niveles de Profundidad 32
Ilustración 11: Tamaño de fuente 33
Ilustración 12: Combinación de colores con contraste accesibles 34
Ilustración 13: Elaboración de wireframes -> justinmind 35

Ilustración 14: Elaboración de wireframes -> Evolus pencil 35
Ilustración 15: Banco de iconos -> Icon finder 36
Ilustración 16: Banco de tipografías -> Dafont 36
Ilustración 17: Banco de fotos -> iStock 37
Ilustración 18: Metodología “User Centered Mobile Design” 38
Ilustración 19: Patrón de Navegación básica 42
Ilustración 20: Patrón de Navegación con menú lateral 43
Ilustración 21: Patrón de Navegación basada en "tabs" 44
Ilustración 22: Presentación multipantalla 47
Ilustración 23: Relación entre paneles 47
Ilustración 24: Expansión/comprensión 48
Ilustración 25: Apilar 49
Ilustración 26: Expansión/Colapso 49
Ilustración 27: Mostrar/Ocultar 50
Ilustración 28: Ejemplo de widgets 51

Tabla 1: Pautas de MWBP 15
Tabla 2: Briefing -> Informe preliminar, descripción y contenido de la aplicación 40

1 Introducción

1.1 Objetivo

El presente documento detalla los criterios de accesibilidad y usabilidad que deberán seguir todos los proyectos de desarrollo para dispositivos móviles.

1.2 Audiencia

Este documento está dirigido a todas las personas que colaboren en labores relacionadas con la gestión, desarrollo, auditoría, implantación y explotación de los sistemas de información de la gerencia de informática de la Dirección General de Tráfico.

1.3 Estructura del documento

El documento se divide en tres capítulos. En el segundo capítulo se establecen las reglas generales de accesibilidad y usabilidad a seguir en todos los proyectos, ya sean webs móviles, aplicaciones híbridas o aplicaciones nativas, y un tercer capítulo relacionado con las especificaciones de acuerdo al sistema operativo Android.

1.4 Glosario

1.4.1 Términos

En este apartado se incluyen, organizados por orden alfabético, los términos que se utilizan en el propio documento para facilitar su lectura y comprensión.

- **MediaQueries**: Módulo CSS3 que permite adaptar la representación del contenido a características del dispositivo como la resolución de pantalla

- **Responsive Design:** El diseño web adaptable o adaptativo, conocido por las siglas RWD (del inglés, Responsive Web Design) es una filosofía de diseño y desarrollo cuyo objetivo

es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visualizarla.

- **Widgets:** Pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget Engine. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.

- **Wireframes:** Guía visual que representa el esqueleto o estructura visual de un sitio web. El wireframe esquematiza el diseño de página u ordenamiento del contenido del sitio web, incluyendo elementos de la interfaz y sistemas de navegación, y cómo funcionan en conjunto.

1.4.2 Acrónimos

En este apartado se incluyen, organizados por orden alfabético, los acrónimos que se utilizan en el propio documento para facilitar su lectura y comprensión.

- **CSS:** Hoja de estilo en cascada

- **LBS:** Sistemas Basados en Localización

- **MWBP:** Mobile Web Best Practices

- **NFC**: Near Field Communication

- **RFID:** Radio-frequency identification

- **URI:** Uniform Resource Identifier

- **W3C:** World Wide Web Consortium

- **WCAG:** Web Content Accessibility Guidelines

2 Reglas generales

A continuación se especifican los criterios generales de accesibilidad y usabilidad que se deberán tener en cuenta en el desarrollo de proyectos con tecnologías móviles.

2.1 Accesibilidad móvil

2.1.1 Introducción

La accesibilidad móvil es el grado en el que todas las personas pueden utilizar una aplicación desde un dispositivo móvil, independientemente de sus capacidades técnicas, cognitivas o físicas.

Para mejorar el grado de accesibilidad de una aplicación pueden utilizarse diferentes tipos de ayudas, tales como las tipografías de alto contraste o gran tamaño, magnificadores de pantalla, lectores y revisores de pantalla, programas de reconocimiento de voz, teclados adaptados, y otros dispositivos apuntadores y de entrada de información.

Una condición necesaria para ofrecer una buena accesibilidad es la usabilidad, si bien el concepto de accesibilidad es más amplio. Cuando hablamos de usabilidad nos estamos restringiendo a un usuario de la audiencia objetivo del sitio, y cuando hablamos de accesibilidad nos estamos refiriendo al máximo rango posible de usuarios, incluyendo a personas con discapacidad.

Las buenas prácticas de usabilidad y accesibilidad que se deben tener en cuenta en el desarrollo y mantenimiento de las aplicaciones web, hacen referencia a los diferentes criterios a seguir relativos a la clasificación de la información y las formas de acceder a la misma, los niveles de profundidad de los menús de navegación, la utilización de botones de acción, los tipos de formularios y entradas de datos, y demás aspectos que es necesario tener en cuenta para asegurarse que la experiencia de todos los usuarios a través de los portales sea efectiva.

2.1.2 Normativa de Accesibilidad

El 20 de septiembre de 2018, un día después de su publicación en el Boletín Oficial del Estado, entró en vigor el Real Decreto 1112/2018, de 7 de septiembre, sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público, que traspone al ordenamiento jurídico español la Directiva (UE) 2016/21022.

En el Real Decreto se obliga a que todas las páginas web de la Administración Pública o dependientes de ésta sean accesibles para las personas con discapacidad. Aplica a todos los sitios web del sector público, desde aquellos pertenecientes a las administraciones, tribunales y servicios de policía a los de los hospitales, universidades y bibliotecas públicas, haciendo que sean accesibles a todos los ciudadanos, especialmente a aquellos con dificultades auditivas, visuales o funcionales.

2.1.3 Accesibilidad a contenidos web desde dispositivos móviles

Las aplicaciones seguirán las pautas de accesibilidad recogidas por el W3C (World Wide Web Consortium) en las WCAG 2.1 (Web Content Accessibility Guidelines), en las que se describen una serie de recomendaciones para hacer más accesible el contenido web:

- **Perceptible**
  - Proporcionar alternativas de texto para el contenido no textual
  - Proporcionar subtítulos y otras alternativas para multimedia
  - Crear contenidos que se puedan presentar de diferentes maneras sin perder significado
  - Facilitar a los usuarios el ver y escuchar contenido
- **Operable**
  - Hacer toda la funcionalidad disponible desde un teclado
  - Proporcionar a los usuarios el tiempo suficiente para leer y utilizar el contenido
  - No utilizar contenido de capturas
  - Ayudar a los usuarios a navegar y encontrar el contenido
- **Comprensible**
  - Utilizar texto legible y comprensible
  - Hacer que los contenidos se visualicen y operen de manera predecible
  - Ayudar a los usuarios a evitar y corregir errores

- **Robusto**
    - Maximizar la compatibilidad con herramientas de usuario actuales y futuras

Acceder a las guías completas WCAG 2.1.

2.2 Buenas prácticas para la web móvil

Las aplicaciones se ajustarán a los criterios de usabilidad y accesibilidad recogidos en por el **W3C en las MWBP (Mobile Web Best Practices)**, en las que se describe cómo mejorar la experiencia de usuario en el acceso al contenido web desde los dispositivos móviles.

Las **MWBP 1.0** son 60 pautas organizadas en 10 principios clave:

- Diseña para una Web única. Si diseñas el contenido teniendo en cuenta los diferentes dispositivos, reducirás costes, tu página será más flexible y satisfarás las necesidades de más personas

- Confía en los estándares Web. En un mercado tan fragmentado como el de dispositivos y navegadores, los estándares son la mejor garantía de Interoperabilidad

- Evita los riesgos conocidos. Un diseño bien planificado ayuda a reducir los problemas de usabilidad causados por pantallas y teclados pequeños u otras funciones de los dispositivos móviles

- Sé prudente con las limitaciones de los dispositivos. Cuando elijas una tecnología Web concreta, ten en cuenta que los dispositivos móviles tienen funciones muy diversas

- Optimiza la navegación. La simplificación de la navegación y del uso del teclado son factores esenciales cuando se utilizan pantallas y teclados pequeños y se tiene un ancho de banda limitado

- Comprueba gráficos y colores. Las imágenes, los colores y el estilo destacan el contenido, pero hay dispositivos con pantallas de bajo contraste o problemas de compatibilidad con algunos formatos

- Hazlo pequeño. Un sitio Web de tamaño reducido supondrá un ahorro de tiempo y dinero para los usuarios
- Economiza el uso de la red. Las funciones de los protocolos Web pueden mejorar la experiencia del usuario al reducir los retrasos y los tiempos de espera en la red
- Facilita la entrada de datos. En los dispositivos móviles, los teclados y demás métodos de introducción de datos pueden ser tediosos para el usuario. Un diseño eficaz minimiza su uso
- Piensa en los usuarios de la Web móvil. Los usuarios de la Web móvil necesitan información sintetizada al disponer de poco tiempo y existir distracciones externas

Acceder al detalle de las MWBP 1.0

A continuación se enumeran aquellas pautas del MWBP 1.0 que se considerarán de obligatorio cumplimiento, independiente del ámbito tecnológico aplicado en el proyecto (Nativo, Híbrido o Web Móvil)

Código	Descripción
5.1.1 Thematic Consistency of Resource Identified by a URI	El contenido debe ser accesible en una gama de dispositivos, independientemente de las diferencias en la capacidad de presentación y mecanismo de acceso. Los sitios web pueden paginar su contenido de varias formas correspondientes a las diferencias en las características del dispositivo; Por lo tanto, la estructura de navegación del sitio, y posiblemente su realización técnica, pueden variar de acuerdo con la clase de dispositivo que se sirve Un marcador capturado en un dispositivo debe ser utilizable en otro tipo, incluso si no produce exactamente la misma experiencia

5.1.4 Testing	Cualquier sitio Web debe ser probado en una amplia gama de navegadores. Los navegadores móviles a menudo muestran marcadas diferencias en las características a los navegadores de escritorio. Además de evaluar la idoneidad de un sitio para la exhibición en formato reducido, se anima a los proveedores a comprobar que las características incluidas en el proyecto funcionan correctamente en dispositivos reales Los proveedores también deben probar el correcto comportamiento del sitio en aquellos estados en los que el navegador presenta ciertas características deshabilitadas, como el uso de los modos de sólo texto y/o con scripting deshabilitado En el caso de desarrollo Nativos y/o Híbridos las pruebas se deberán realizar en diferentes dispositivos (diferentes vendors, diferentes características técnicas, , diferentes tamaños de pantalla, diferentes versiones de Android,…) para permitir homologar el correcto funcionamiento del proyecto en un amplio espectro de dispositivos
5.2.2 Navigation Bar	Siempre que sea necesario, se deberá proporcionar una navegación básica, que se deberá situar en la zona superior de la vista. En este caso se deberá optar por aquellos elementos de navegación que permitan maximizar en área dedicada al contenido (p.ej: menú lateral retractil) Si fuera imprescindible otro elemento de navegación

	secundaria, este se colocará en la parte inferior de la página. Es importante que los usuarios sean capaces de ver el contenido de la página una vez que la página se ha cargado sin necesidad de desplazarse, por lo que se optará, siempre que sea posible, por scrolls verticales infinitos siempre sobre otro tipo de desplazamientos (p.ej: scrolls laterales, navegaciones laterales (swipe),…)
5.2.3 Balanced Structure	El diseño debe tratar de proporcionar un equilibrio entre tener un único nivel de información (acceso inmediato a la información pero con poca o ninguna estructura) o tener múltiples niveles de información (necesidad de realizar varias acciones de navegación para llegar al contenido deseado). Hay que tener en cuenta dos premisas principales a la hora de realizar el diseño / wireframe de la aplicación: o Los usuarios se sienten frustrados si se requiere realizar más de cuatro (4) acciones de navegación para acceder al contenido deseado. Deberá estructurarse la información de la aplicación de manera que aquella información crítica para el usuario esté contenida en el primer nivel de navegación. Se deberá también limitar el número de niveles de información en la aplicación o Cada recuperación de un nivel de información, que no se encuentre previamente cacheado en el dispositivo, tiene un coste importante en tiempo y es muy sensible a la disponibilidad de conectividad en ese momento (la conectividad es extremadamente

	fluctuante en los dispositivos móviles). En el diseño de los servicios responsables de recuperar la información de la aplicación, siempre que sea posible, se maximizará la información recuperada por petición (2 niveles). Se utilizarán estrategias de cacheo local para evitar realizar peticiones de información que ya se encuentra recuperada y que no ha variado.
5.2.6 Link Target Identification	Los usuarios de dispositivos móviles pueden sufrir demoras como resultado de las acciones de navegación seleccionadas, por eso se deberán tener en cuenta las siguientes consideraciones: o En cualquier tipo de desarrollo móvil (Nativo, Híbrido o Web móvil) es importante identificar correctamente al usuario donde conduce un elemento de navegación o que información es accedida a través de él, para que de esa manera los usuarios puedan realizar una evaluación del interés que tiene para ellos respecto a la posibilidad de sufrir una demora en su obtención o Todos aquellos elementos de navegación que referencien una descarga de contenido deberán proporcionar información al usuario sobre el tamaño de dicho contenido (en bytes o de manera abstracta, por ejemplo, tiempo aproximado con la conectividad actual). El objetivo es evitar la frustración del usuario

	por tiempos excesivos de espera para acceder al contenido. o Todos aquellos elementos de navegación que referencien una descarga de contenido deberán proporcionar información al usuario sobre si dicho contenido podrá ser (o no) reproducido en el dispositivo del usuario. El objetivo es evitar la frustración del usuario por no poder reproducir el contenido una vez se ha realizado la descarga. o Todos los procesos de descarga deberán presentar un elemento que permita la cancelación del mismo por el usuario
5.3.3 Scrolling (Desplazamiento)	La vista se debe presentar de tal manera que el scroll vaya en dirección vertical permitiendo que el usuario pueda tener una buena experiencia en todo el contenido. Si un elemento de la página necesita el scroll secundario (puede suceder con imágenes o mapas) no se debe hacer que el resto de la página lo requiera. En este caso se debe se debe considerar poner este contenido en una vista diferente con un enlace que lleve al contenido principal.
5.3.4 Navigation Bars (Extraneous material)	Se debe poner la mínima información posible – como por ejemplo elementos de navegación, imágenes decorativas, anuncios… – delante del contenido principal para que el usuario tenga una idea del dicho contenido la primera vez que la ve y evitar que tenga que mover la barra de scroll para encontrarlo.

5.3.7 Background Images	El uso de imágenes para el fondo de la pantalla puede llevar a hacer que sea difícil ver el contenido principal de la página, especialmente debido al poco contraste que suelen tener los dispositivos móviles y a las malas condiciones en las que se suelen utilizar estos dispositivos. Si se utilizan imágenes de fondo hay que asegurarse de que el contenido sigue siendo legible en el dispositivo.
5.4.13 Error Messages	Es inevitable que, en ocasiones, un usuario móvil no tenga éxito en el acceso al contenido o información deseada. Por ello se deberán tener en cuenta las siguientes consideraciones: o En desarrollos de Web móvil se deberá proporcionar una navegación sencilla integrada dentro del elemento que informa del error (de la ausencia del contenido o la información). Esto es particularmente importante si tenemos en cuenta que en el ámbito móvil los navegadores pueden no tener un botón fácilmente localizable para realizar una acción de "Atrás", donde la contextualización con frecuencia es difícil y donde la reescritura de URIs como medio de recuperación de errores suele resultar complicada o Todos aquellos errores que sean gestionados por la aplicación (quedan excluidos aquellos proporcionados por el SO del dispositivo) deberán presentar de manera clara y concisa la causa del

	error, si es un error permanente o temporal (que pueda permitir el reintento por parte del usuario) y como se escala dicho problema para su resolución o Debe ser posible para el usuario, volver a la vista en la que se encontraba antes del error o a otra vista desde donde se pueda reintentar o alterar la operación que se pretendía realizar

Tabla 1: Pautas de MWBP

2.3 Usabilidad en dispositivos móviles

2.3.1 Definiciones de *usabilidad*

“La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso. “ (ISO/IEC 9126)

"Usabilidad es la eficacia, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico. “ (ISO/IEC 9241)

“La usabilidad es un atributo de calidad que mide qué tan fáciles de utilizar son las interfaces de usuario.” (Jakob Nielsen)

Para Jakob Nielsen la usabilidad se define por 5 componentes que influyen en la calidad:

- **Facilidad de aprendizaje.** ¿Qué tan fácil es para los usuarios llevar a cabo tareas básicas la primera vez que se encuentran con el diseño?

- **Eficiencia.** Una vez que los usuarios han aprendido el diseño, ¿con qué rapidez pueden realizar las tareas?

- **Recursividad.** Cuando los usuarios vuelven al diseño después de un periodo de no utilizarlo, ¿con qué facilidad pueden restablecer su capacidad previa de uso?

- **Errores.** ¿Cuántos errores cometen los usuarios, qué tan grave son estos errores y con qué facilidad se pueden recuperar de los mismos?

- **Satisfacción.** ¿Qué tan agradable es de utilizar el diseño?

2.3.2 El contexto en un entorno móvil

Todos los principios de usabilidad que ya se habían de tener en cuenta en las aplicaciones web seguirán siendo válidos para las aplicaciones móviles. Estos principios son:

- **Funcionalidad**: Un sitio web debe estar orientando a la simplicidad y facilidad con la que el usuario podrá realizar las tareas previstas, minimizando al máximo el número de clics necesarios para realizar cada acción y el número de ventanas por las que tendrá que pasar para hacerlo

- **Navegabilidad**: La arquitectura de la información ofrecida por un sitio web debe ser clara, lógica, consistente, coherente a lo largo de todo el sitio, y debe responder a las necesidades de los usuarios según sus diferentes perfiles. Las opciones disponibles en cada momento deben estar siempre bien visibles y deben evitarse niveles profundos de navegación

- **Formularios**: La entrada de datos a un sitio web a través de un formulario debe simplificarse el máximo posible, evitando al usuario introducir datos innecesarios, utilizando nombres de campo suficientemente descriptivos, ofreciendo ayuda cuando es necesario, manteniendo coherencia en las etiquetas utilizadas, indicando los campos obligatorios,…

- **Confianza**: Los contenidos ofrecidos por un sitio web deben ser fidedignos, indicándose siempre que sea posible los autores, responsables, o las fuentes de los mismos. A su vez deben mantenerse siempre actualizados, haciendo constar cuando corresponda la fecha en que fueron modificados por última vez

- **Contenido**: El lenguaje utilizado en un sitio web debe ser claro, conciso y adaptado al perfil de los diferentes grupos de usuarios que puedan acceder al mismo, cuidándose

especialmente los textos y formatos utilizados en la definición de títulos, subtítulos, links, botones de acción y etiquetas

- **Diseño**: La diagramación y el diseño gráfico de un sitio web deben ser atractivos y contribuir a la facilidad de uso y navegación a través del mismo por parte de los usuarios. El sitio debe ser consistente y coherente a nivel de tipografía, cromatismo y estándares, y en línea con el estilo gráfico corporativo

- **Búsquedas**: La caja de búsqueda y sus controles deben estar claramente etiquetados y ubicados donde el usuario espera encontrarlos. Los términos de búsqueda deben ser intuitivos, sin necesidad de utilizar operadores booleanos sino ofreciendo la posibilidad de redefinir la búsqueda a través de un formulario. La lista de resultados debe ser clara y ordenada por relevancia.

- **Ayuda**: Un sitio web debe incluir una sección de FAQ o ayuda en línea que proporcione a los usuarios información paso a paso que les permita llevar a cabo las acciones más importantes. En caso de error, se debe informar claramente al usuario lo que ha pasado y cuáles son los pasos que debe seguir.

El contexto en el que se utilizan las aplicaciones móviles hace que sea más importante aún cuidar todas estas pautas de usabilidad, teniendo en cuenta además otros factores que intervienen cuando se utiliza una aplicación desde un dispositivo móvil:

- **Ergonomía**
    - Características y condiciones variadas del lugar desde el cual se puede estar accediendo a la aplicación.
    - Condiciones de iluminación en las cuales se puede utilizar la aplicación (luz u oscuridad excesivas, reflejos del sol, etc.)
    - Tamaño de botones adecuados a diferentes personas, con separación suficiente entre ellos para que se puedan pulsar sin problemas en condiciones particulares (con guantes, en movimiento, personas que puedan padecer temblores en las manos, etc.)

- Movimiento y rotación de la pantalla móvil (la aplicación debe funcionar y visualizarse correctamente tanto en vertical como en horizontal)

Ilustración 1: Usabilidad -> ergonomía

- **Pantallas**

  - Qué pantallas se van a utilizar (tipos, tamaños, etc.)
  - Para qué tamaño y resolución de pantalla se desea optimizar la aplicación

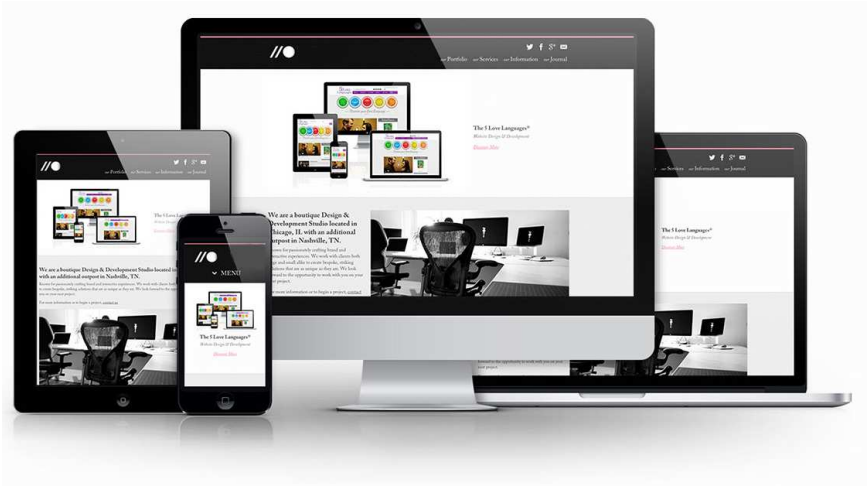

Ilustración 2: Usabilidad -> tipos de pantallas

- **Gestos**

  - Sirven como *short cuts* (formas rápidas de activar una funcionalidad)

- El usuario interactúa directamente sin necesidad de utilizar un botón, con lo cual cambia la relación del usuario con el contenido. Se actúa directamente sobre el contenido

Pulsar: Ejecutar la funcionalidad asociada al elemento pulsado.

Mantener la pulsación: Entrar en el modo de selección.

Deslizar: Realizar scroll o cambiar a distintas vistas en la misma jerarquía.

Mantener la pulsación y arrastrar: Mover datos a un contenedor o recolocar un elemento dentro de una vista.

Doble pulsación: Escalar una cantidad predeterminada del total en cada doble pulsación o seleccionar texto.

Doble pulsación y arrastrar: Escalar centrándose en el lugar pulsado.

Pellizcar hacia afuera: Zoom para acercar.

Pellizcar hacia adentro: Zoom para alejar.

Ilustración 3: Usabilidad -> gestos

- **Factor emocional**
  - Cercanía del tacto

- Dispositivos personales (se puede utilizar el conocimiento que tiene el móvil sobre el usuario)
- Diálogo con el producto

- **Limitaciones**
  - Tamaños de las pantallas, que pueden no ser apropiados para determinadas funcionalidades o situaciones
  - Conexiones intermitentes o falta de conexión
  - Desempeño del dispositivo
  - Tiempos de uso limitados y con interrupciones

2.3.3 Características propias del entorno móvil

Al diseñar aplicaciones para dispositivos móviles cambian tanto las formas en las que interactúa el usuario con el dispositivo como los medios disponibles para obtener información del entorno a través del móvil. Estos elementos variarán en función del dispositivo y del sistema operativo a utilizar.

Principales elementos de interacción con el usuario:

- Barras de navegación
- Menús
- Botones de acción
- Ventanas modales
- Gestos
- Voz

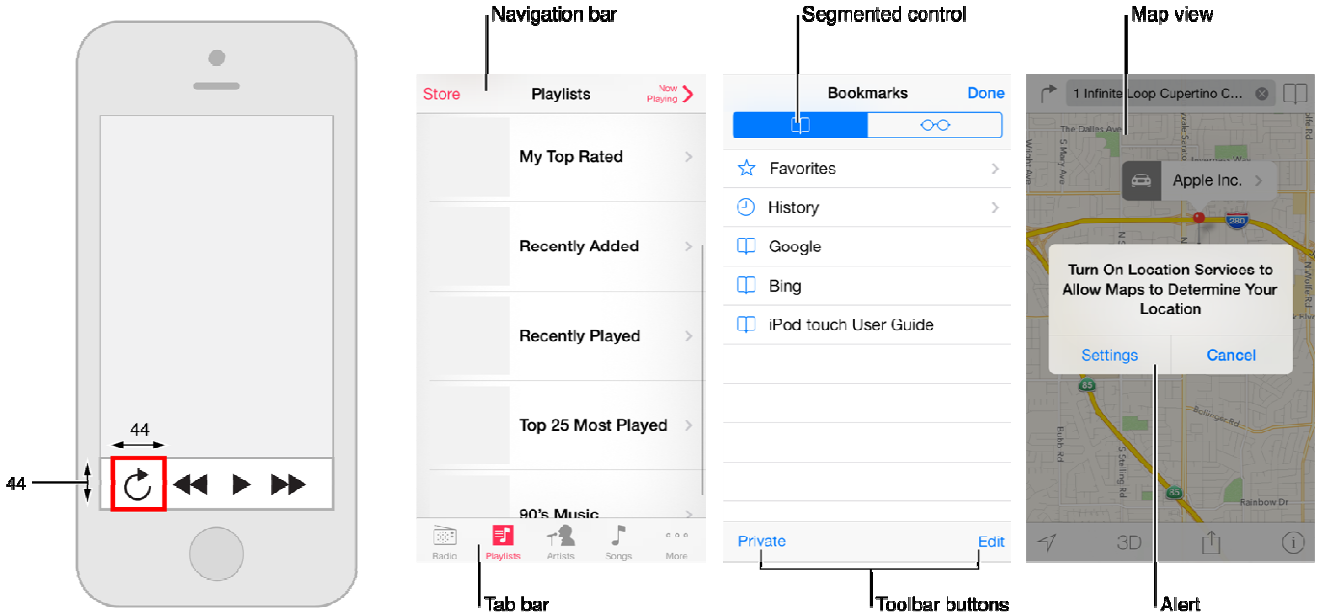

Ilustración 4: Elementos principales de interacción

Funcionalidades propias de los dispositivos móviles:

- Notificaciones Push
- Sistemas Basados en Localización (LBS) y contexto
- Bluetooth 4.0
- Sensores
- Realidad Aumentada
- NFC (Near Field Communication)
- Códigos QR
- RFID
- Pago con el móvil
- Machine to Machine (M2M) and Smart Products

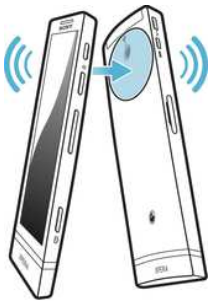

Ilustración 5: Usabilidad -> tipos de pantallas

2.3.4 Diseño adaptable

El diseño adaptable o *responsive design* permite a una página web adaptarse al tamaño de pantalla del dispositivo a través del cual el usuario esté accediendo. Su uso es de obligado cumplimiento en el diseño de aplicaciones.

El principal concepto en el que se apoya es el de abandonar los anchos fijos de la web, sustituyéndolos por anchos fluidos, pensados en términos de proporciones.

Ilustración 6: Ejemplo de diseño adaptable de la web de Antonio Gaudí

En el caso de webs móviles, que deban funcionar correctamente -tanto desde el punto de vista de la funcionalidad como a nivel de experiencia de usuario- en diferentes tipos de dispositivos, es obligatorio optar por este tipo de diseño.

Dentro de las técnicas ***responsive design***, la manera de diseño más extendida es ***mobile first***. Los *wireframes* se empiezan a diseñar pensando desde los dispositivos de baja resolución a los de mayor resolución. Esta forma de diseño proporciona diseños más simples y sin elementos redundantes. Simplemente se implementa lo que se necesita y se amplía para versiones con mayor resolución.

Asimismo, existen elementos en una interfaz que dependiendo de la resolución de una pantalla se deberán mostrar de una u otra manera para garantizar su buena usabilidad. Por ello es importante decidir en qué puntos el diseño se romperá y se mostrará de manera diferente dependiendo del dispositivo. Cada uno de estos puntos clave es conocido como ***breakpoint***.

Gracias al análisis y diseño de los *breakpoints* se pueden definir *MediaQueries* en CSS3 de manera más rápida y fluida.

2.4 Directrices para aplicaciones móviles

Al diseñar y desarrollar una aplicación para un dispositivo móvil, ya sea nativa o híbrida, se deberán seguir las pautas que se detallan a continuación.

- **Rapidez**:

Los usuarios de móviles suelen estar ocupados y aprovechan cualquier momento libre para realizar tareas online. Además, en el caso de los *smartphones*, la mayor parte de su tiempo de uso se realiza en tránsito y con una atención parcial.

Por ello se deberá utilizar un diseño que permita cargar el contenido rápidamente y que facilite la lectura selectiva.

  - Resalta el contenido y las funciones más necesarias para los usuarios.
  - Evita los párrafos extensos y utiliza viñetas para facilitar la lectura.
  - Minimiza la latencia percibida:
  - En caso de necesitar scripts, éstos se situarán al final de la página de forma que la información importante esté visible mientras se carga el resto de contenido.
  - Usar indicadores de progreso.
  - Actualizar las páginas vía DOM en vez de recargarlas, manteniendo el foco en el elemento que lo tuviera.
  - Comprime las imágenes para reducir su tamaño y agrupa todas las imágenes estáticas en un único recurso compuesto (sprite), para que el sitio se cargue más rápidamente.
  - Elimina espacios en blanco, comentarios y cualquier elemento superfluo que contribuya a hacer las descargas más pequeñas.
  - No se debe interrumpir al usuario si no es necesario.
  - No abuses de los diálogos.

- **Facilidad para la entrada de datos**:

Facilita la entrada de datos a los usuarios y permite que se pongan en contacto cuando quieran.

  - Céntrate en la información importante.

- Reduce los pasos necesarios para llevar a cabo una transacción.
- Ofrece formularios breves y utiliza el menor número de campos posible.
- Utiliza la función de clic de llamada en todos los números de teléfono.

- **Gestión correcta de los datos de aplicación**:

Utiliza una tecnología apropiada para almacenar información en el dispositivo en función de la naturaleza del dato y teniendo en cuenta las consideraciones indicadas en la especificación DGT acerca de su uso:

- *"Shared Preferences"* para almacenar las preferencias
- *"Internal Storage"* para datos privados de aplicación
- *"External Storage"* para datos públicos en la memoria externa del dispositivo. Se debe comprobar previamente si está disponible.

El acceso a la ubicación física de almacenamiento no se realizará usando direcciones absolutas sino utilizando el API disponible para tal efecto.

- **Navegación simple**

Si ofreces una navegación intuitiva y una función de búsqueda los usuarios encontrarán fácilmente lo que buscan.

- Evita que los usuarios tengan que desplazarse por la página.
- Utiliza jerarquías de menú claras.
- Facilita la navegación entre secciones con botones bien visibles para volver atrás y acceder a la página principal.
- Incluye un cuadro de búsqueda siempre a la vista en sitios complejos.

- **Carácter local**:

Los usuarios buscan información local en sus teléfonos constantemente. Incluye funciones que ayuden a los usuarios a localizar los lugares que buscan.

- Facilita las direcciones o un localizador de dependencias en la página de destino.
- Incluye mapas e indicaciones. Utiliza la función de GPS para personalizar la aplicación en la medida de lo posible.
- Permite a los usuarios comprobar información de las dependencias más cercanas.

- **Diseño para múltiples métodos de interacción:**

Diseña la aplicación para que esté preparada para atender diferentes formas de interacción del usuario.

- Basada en foco
  - Marcar el elemento que tiene el foco
  - Saltar automáticamente al siguiente foco sin afectar a la visibilidad
- Basada en puntero
  - Los elementos relacionados han de estar cerca
  - Los elementos seleccionables han de ser suficientemente grandes para ser seleccionados fácilmente
  - Los elementos seleccionables deben resaltarse al entrar el puntero en su área activa
- Basada en tacto
  - Los elementos seleccionables han de ser suficientemente grandes para que incluso una persona con las manos grandes pueda interactuar con ella (Facilidad para clicar con el pulgar).
    - Utiliza botones grandes y centrados, y deja distancia entre ellos para evitar clics accidentales.
    - Amplía el área de respuesta de los botones de menor tamaño.
    - Amplía el área de respuesta de las casillas de verificación permitiendo los clics en el texto.
  - Los elementos no tienen el foco hasta que han sido seleccionados

- **Experiencia homogénea**:

Los usuarios se mueven de una pantalla a otra continuamente. Utiliza las mismas funciones de la versión del sitio para ordenadores en la medida de lo posible para garantizar una experiencia homogénea.

- Permite que los usuarios guarden sus búsquedas más frecuentes y el contenido de sus favoritos.

- Incluye las funciones más importantes del sitio en todos los canales siempre que sea posible.
- Muestra la misma información de productos o servicios.

- **Optimización de la visibilidad:**

Las aplicaciones móviles comunican el mensaje de la forma más cómoda posible para la vista a fin de facilitar su lectura.

- El texto debe resaltar sobre el fondo.
- Asegúrate de que el contenido se ajuste a la pantalla y se pueda leer sin necesidad de agrandarlo.
- Recurre todo lo posible al espacio negativo.
- Utiliza el tamaño y el color para indicar la prioridad de los enlaces o de los botones.
- Utiliza efectos 3D y el sombreado para los botones.
- Usa imágenes mejor que palabras.

- **Accesible:**

Lo ideal es que una aplicación para móviles funcione en todos los dispositivos y en todas las orientaciones de pantalla.

- Utiliza HTML5 para la interactividad y la animación.
- Adapta la aplicación para orientación de pantalla horizontal y vertical.
- Utiliza la etiqueta "meta viewport" para establecer el tamaño de la pantalla deseado a través de las constantes "device-width" y "device-height".
- Mantén a los usuarios en la misma sección de la página al cambiar la orientación.

2.5 Diseño de interfaz

La interfaz gráfica de cualquier aplicación es un elemento clave en el proceso de comunicación e interacción con los usuarios.

La estrategia a seguir para conseguir una interfaz visual eficaz se basa, principalmente, en conseguir una sencillez formal en la apariencia y una simplicidad en la usabilidad de la aplicación, que hagan suficientemente ágil e intuitiva la consulta de los contenidos y el uso de las funcionalidades.

El diseño de la interfaz gráfica deberá ajustarse a la tipografía, el cromatismo y las demás normas de diseño definidas en la guía de estilo corporativo de la DGT, así como mantener una homogeneidad visual con el resto de aplicaciones de la DGT.

Los elementos que se deberán tener en cuenta a la hora de diseñar la interfaz gráfica de una aplicación serán los siguientes:

- *Wireframes* de pantallas
- Jerarquía de los contenidos
- Estructura
- Navegabilidad
- Texto y colores
- Imágenes y animación
- Creatividad y diseño
- Accesibilidad

- ***Wireframes***

Una vez definidos los requerimientos funcionales de una aplicación y previo al diseño gráfico de la misma, es muy aconsejable elaborar los *wireframes* de las diferentes pantallas de la aplicación, en los cuales se definirá la estructura de contenidos y funcionalidades, la navegación entre las páginas, y todos los elementos de interacción con los usuarios. Se trata de una representación esquemática de una pantalla sin elementos gráficos que distraigan la atención (colores, tipografías, imágenes, textos…). Qué hace la pantalla y no cómo se ve.

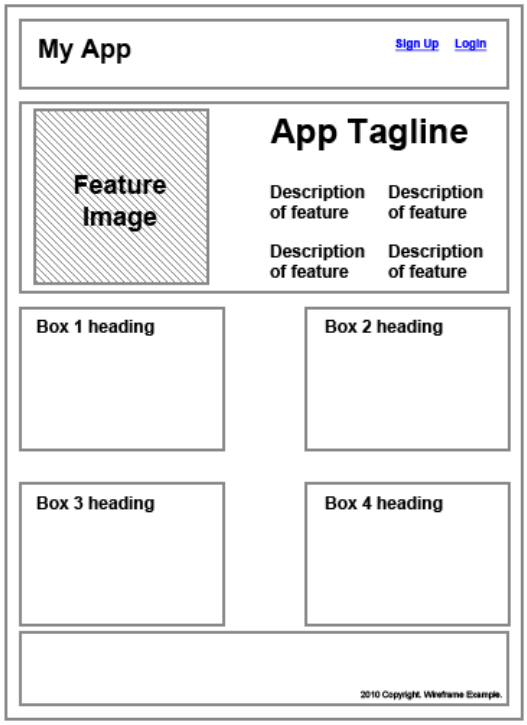

Ilustración 7: Ejemplo de wireframe

- **Jerarquía**

Mediante un adecuado diseño se puede establecer un camino visual que conduzca el ojo del espectador, evitando las distracciones y destacando los elementos más relevantes. Para ello es importante establecer una jerarquía visual adecuada que se mantenga en todas las páginas de la aplicación.

Los recursos de los que se dispone para jerarquizar los diferentes elementos son:

- Posición de los elementos
- Tamaño
- Color, forma, contraste

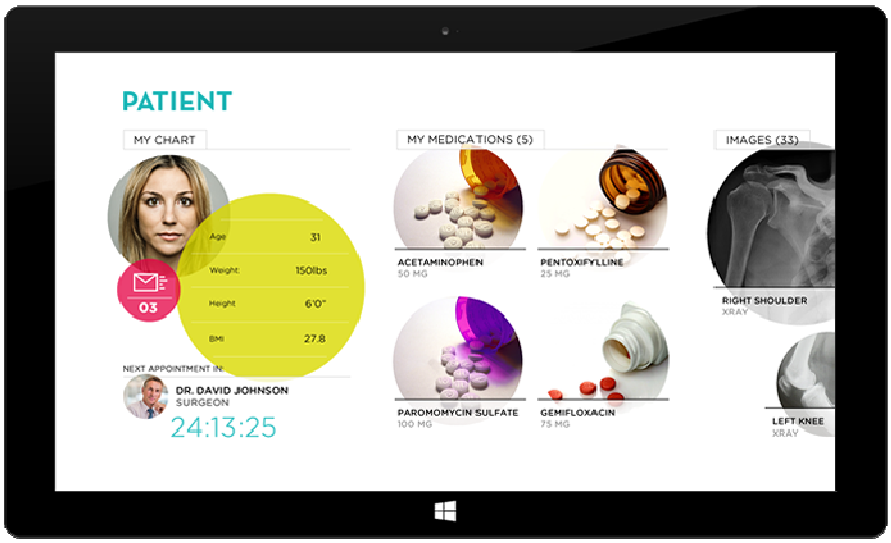

Ilustración 8: Ejemplo de jerarquía

- **Estructura**

La estructura es la cuadrícula no visible de una pantalla, que ayuda a ordenar de forma proporcionada y estable los contenidos. Se debe definir antes de empezar a diseñar.

Elementos que han de definir la estructura:

- Bloques de información básicos: cabecera, menús, contenidos,…
- Resolución de pantalla
- Alineación del diseño
- Tamaño fijo o adaptable
- Espacios y márgenes de reserva

Ilustración 9: Ejemplo de estructura

- **Navegabilidad**

Es importante que una aplicación sea fácilmente navegable, es decir, que el usuario visualice y acceda a la información deseada desplazándose fácilmente por todas las páginas.

Para ello se deberán tener en cuenta las siguientes consideraciones:

- Mapa de contenidos bien jerarquizado.
- Menús claros, concretos y bien equilibrados en cuanto a la cantidad de opciones disponibles en cada sección.
- Ofrecer siempre la opción de volver atrás y a la página de inicio.
- Identificar claramente la sección en la que se encuentra el usuario, de ser posible incluyendo un “camino de migas” que indique al usuario el recorrido que ha hecho para llegar allí (por ejemplo: Inicio > Clientes > Recomendaciones)
- Minimizar la cantidad de niveles de profundidad de la información (en lo posible no más de tres niveles). Muestra en el primer nivel los conceptos candidatos a ser

seleccionados con la mínima información necesaria y utiliza los siguientes niveles para mostrar información más detallada de la selección realizada.

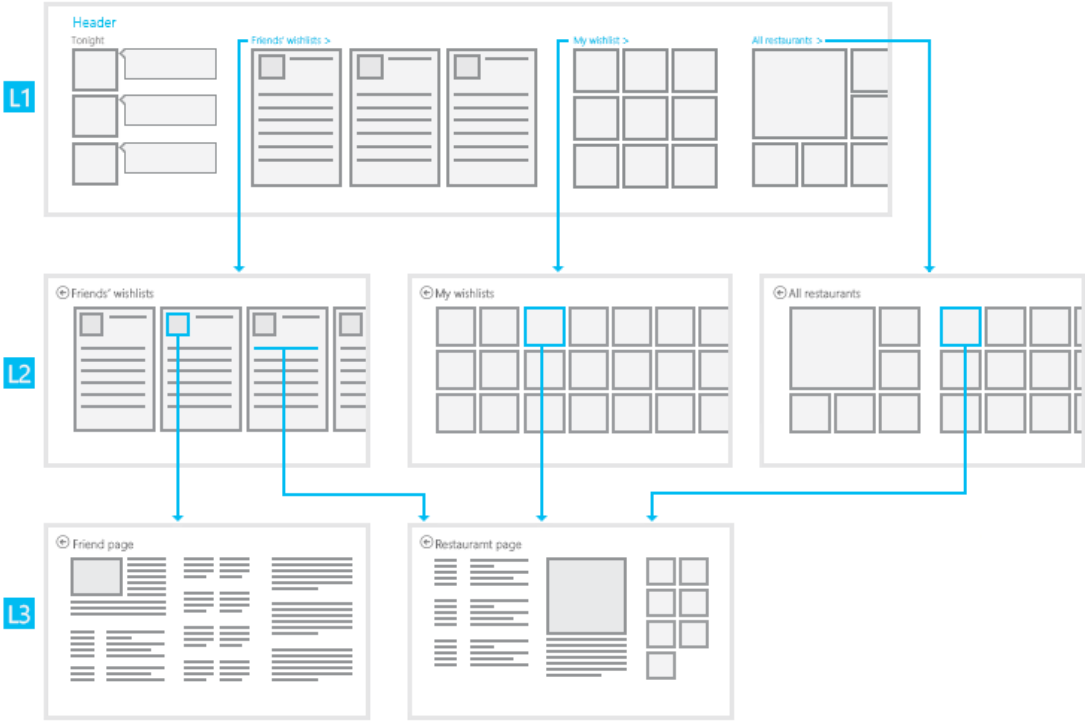

Ilustración 10: Niveles de Profundidad

- **Texto y colores**

Los textos son una parte fundamental del contenido de una aplicación.

Las consideraciones que deben tenerse respecto al uso de tipografías son las siguientes:

- Tipografías corporativas.
- Respetar la guía de colores.
- Minimizar la cantidad de tipos de letra diversos. Máximo 3 tipos con sus variantes.
- Limitar la cantidad de texto mostrado para evitar que el usuario pierda interés.
- Limitar el uso de cursivas. Utilizarlo exclusivamente para citas y anglicismos.
- Evitar el uso de subrayados, excepto para links.
- Tamaño de texto legible y con un contraste adecuado:

- Es aconsejable utilizar tipografía oscura sobre fondos claros.
- El tamaño de letra irá en función de la resolución de pantalla y del público al cual se dirige la aplicación (para mayores de 40 años es recomendable utilizar tamaños de letra más grandes).

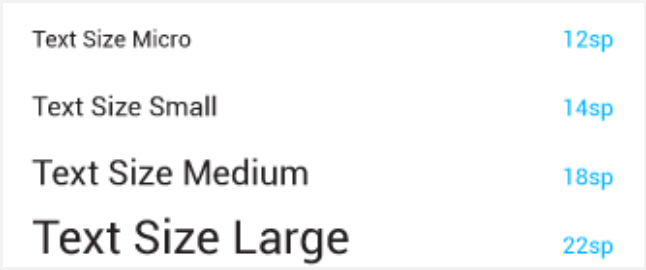

Ilustración 11: Tamaño de fuente

- **Imágenes y animación**

Las imágenes y animaciones proporcionan gran cantidad de información y dan estilo a una aplicación, ya que es lo primero que llama la atención. Conviene por tanto ser selectivos en las imágenes a mostrar.

Las consideraciones que deben tenerse respecto al uso de imágenes y animaciones son:

- Respeto de imágenes corporativas y fondos, no se debe distorsionar proporciones ni aumentar de forma forzosa imágenes o logos.
- Evitar lo máximo posible el uso de: animaciones, introducciones en video o Flash y Gifs animados que distraigan.
- Conocer los derechos de imagen de cada foto

- **Creatividad y diseño**

Una vez que se tienen los wireframes de la aplicación validados por parte de los responsables del proyecto, antes de empezar a diseñar hay que tener en cuenta:

- Referentes gráficos del cliente.
- Guía de estilo, Logo en formato adecuado, web, banco propio de imágenes…
- Estudio de la competencia.

- Resolución y formato para el cual diseñaremos.

- **Accesibilidad**

A fin de mejorar la accesibilidad de la aplicación deberán tenerse en cuenta los siguientes factores de diseño:

- Contraste de color: utilizar tipografías contrastadas con el fondo. Mejor texto negro sobre fondo blanco que texto blanco sobre negro.
- Colores: dependiendo del color o foto al que superpongamos un texto, éste puede no leerse adecuadamente. Vigilar los ruidos y contrastes entre colores complementarios (por ejemplo rojo sobre verde).
- Tamaños de fuente legible: utilizar un tamaño de fuente adecuado al público objetivo de la aplicación (si bien los sistemas operativos ofrecen la posibilidad de ajustarlo desde la configuración de opciones de accesibilidad del propio sistema).

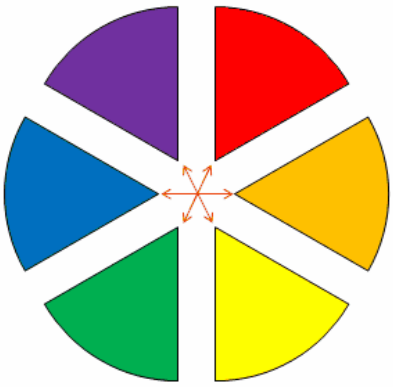

Ilustración 12: Combinación de colores con contraste accesibles

2.5.1 Recursos y herramientas para el diseño de interfaz

- **Elaboración de wireframes**:

  - Justinmind Prototyper (http://www.justinmind.com)

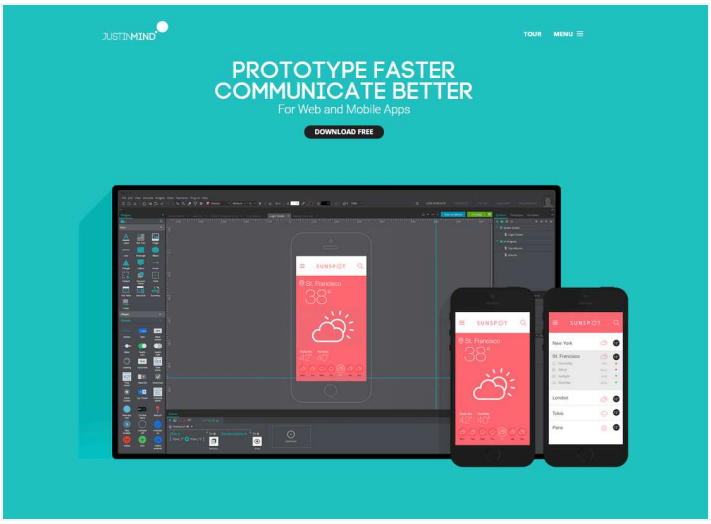

Ilustración 13: Elaboración de wireframes -> justinmind

- Evolus Pencil (http://pencil.evolus.vn)

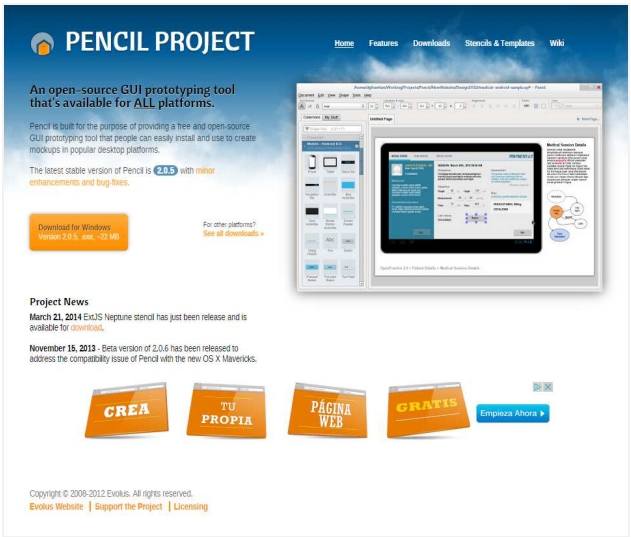

Ilustración 14: Elaboración de wireframes -> Evolus pencil

- **Cuadrícula para la alineación de elementos**

  - 960 Gridder (http://960.gs/)

- **Banco de iconos:**

  - Icon Finder (https://www.iconfinder.com)

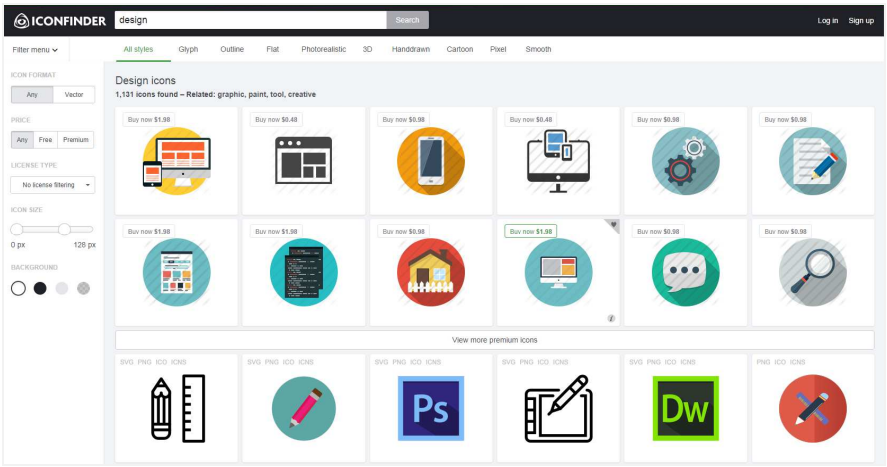

Ilustración 15: Banco de iconos -> Icon finder

- **Banco de tipografías:**

  - Dafont (http://www.dafont.com/es)

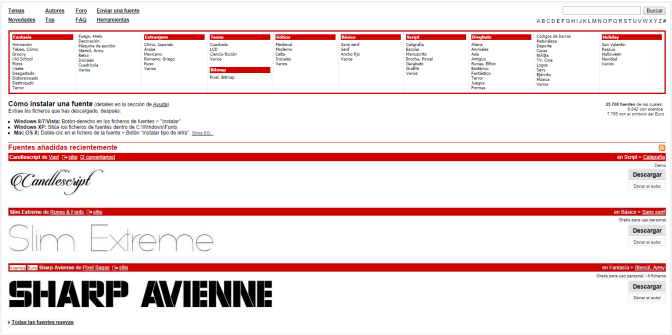

Ilustración 16: Banco de tipografías -> Dafont

- **Banco de fotos:**

  - iStock (http://www.istockphoto.com/)

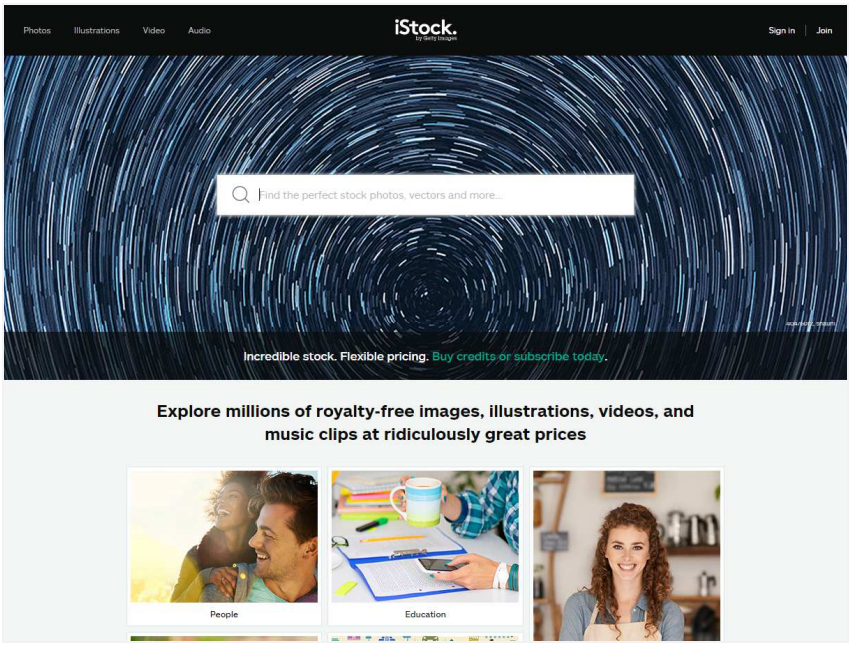

Ilustración 17: Banco de fotos -> iStock

3 Directrices de diseño en Android

El proceso de diseño en Android deberá seguir una metodología del tipo **"User Centered Mobile Design"**

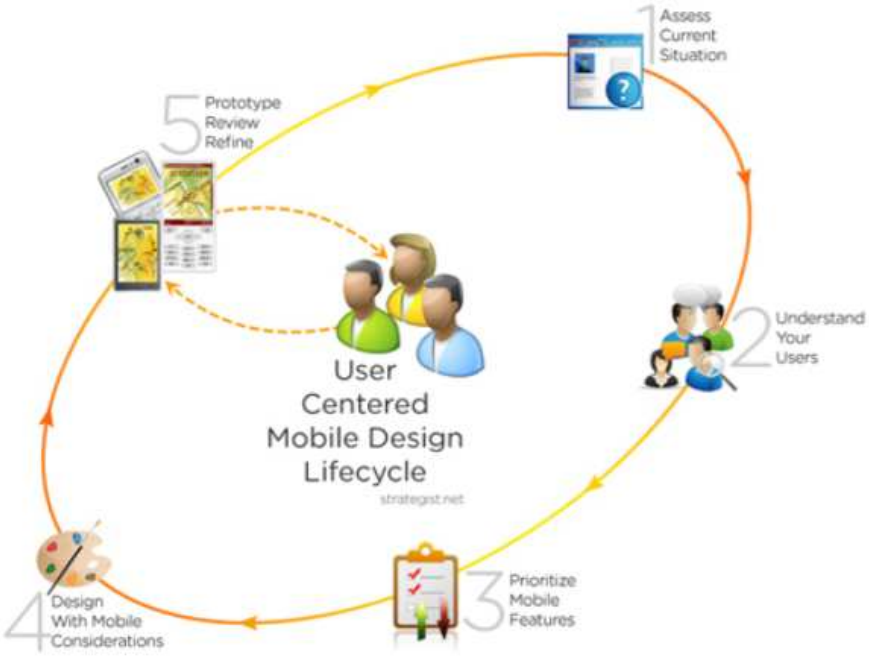

Ilustración 18: Metodología "User Centered Mobile Design"

Como **primer paso se deberá desarrollar un *briefing*** (o informe) de la aplicación que contenga la siguiente información:

Descripción	Breve resumen del producto que se quiere realizar y de sus objetivos.
Antecedentes	Es importante conocer la empresa para saber qué diseño se adapta más a ella. Hay que saber si se han hecho otros productos similares

	con la misma marca y cómo se han tratado.
Existencia de productos similares	Es importante conocer el mercado y saber si existen productos similares al que se quiere desarrollar, ¿Puede ayudar a mejorar el producto?, ¿Habrá que diferenciarse de la competencia en cuanto a diseño y UX?,…
Manual de Marca	Es importante conocer si la organización dispone de un manual de marca: colores, identificación, logo, tipografías... Como han usado ese manual en otros productos. También puede ayudar preguntar sobre la misión y visión de la organización, que quieren transmitir a sus usuarios, si tienen web o si se deberá basarse en otro producto de la marca para desarrollar un diseño coherente,… Si no hay manual de marca será necesario que el cliente genere uno, pudiendo llegar incluso a proporcionarle unas bases para generar uno
Rasgos del producto	Este apartado debe describir los atributos del producto. Qué se quiere transmitir, qué atmósfera quiere el cliente que tenga la App, qué objetivos hay que cumplir con el diseño. También se puede preguntar sobre las ventajas y diferencias respecto de la competencia que hay que resaltar. Si existen pantallas de ejemplo, referentes de otras Apps que les gusten (en cuanto a navegabilidad, a diseño...) sean del mismo sector o no. A veces los referentes gráficos pueden venir de Apps de otras categorías.
Canales de distribución	En el caso de las App, se deberá especificar en qué dispositivos se implementará el producto: iOS, Android, Webapp... ¿Existe una web? ¿queremos que la App tenga los mismos comportamientos y funcionalidades que la web?
Target	Es importante identificar el usuario tipo. El futuro producto debería conservar el carácter del público objetivo que va a ser el usuario de la App. Hay que tener en cuenta el punto de vista demográfico, si es un usuario acostumbrado a los Smartphones, sus hábitos, sus

	conductas, sus debilidades, así como los datos relativos referentes a la percepción que los consumidores tienen del producto o marca. Así se pueden incluir en el producto, por ejemplo, palabras o comportamientos propios del target.
Publico Potencial	Es toda persona (sin diferenciar sexo, edad, o raza) que se sienta atraída y llegue a disponer del producto sin ser parte del público objetivo. Es importante tenerlo en cuenta en el caso de desear una ampliación del target del producto

Tabla 2: Briefing -> Informe preliminar, descripción y contenido de la aplicación

El segundo paso, una vez finalizado el *briefing*, es la realización del Desarrollo de UX.

A veces puede que el cliente proporcione unos *Wireframes*, si no es así el experto UX tendrá que generarlos. Será positivo que el experto UX se reúna con el equipo creativo para hacer una sesión de *Brainstorming* sobre como plasmar los objetivos del proyecto en *Mockups* (previos a los *Wireframes*). Siempre actuando en concordancia con un técnico para asegurarse de que todo lo que se está pensado en los *Mockups* es factible a nivel de desarrollo.

Será la arquitectura de la información la que ordenara los flujos de navegación. Es importante que el usuario sepa siempre dónde está, cuál es la función de cada pantalla y cómo volver atrás. En Móvil, la arquitectura de la información se parece a la de una web, pero hay que tener muy presente que cada funcionalidad de la App ocurra en pantallas individuales, es decir, cada pantalla tendrá un objetivo claro.

En esta fase es obligado hablar de *Paper prototyping*. Será clave plantear los casos de uso y reflejarlos en papel para luego ordenarlos por prioridad. Para ordenar los casos de uso por prioridad y pensar cuáles son *Core* y cuáles son secundarios, siempre habrá que tener en mente el objetivo de la App que nos da el cliente en el *Briefing* y el del usuario objetivo en cada caso.

Una vez finalizados los *Mockups* o prototipos de pantallas, un ejercicio útil es hacer que algún miembro del equipo, externo al equipo de diseño y UX, vaya pulsando los botones imaginarios sobre el papel y otro (el que ha pensado los flujos) vaya cambiando de una pantalla a otra con los prototipos de papel. Es útil para ver si se ha pensado en todos los casos de uso y para comprobar si la navegación se entiende por alguien que no conoce la App.

Los *Mockups* deben estar muy bien definidos antes de empezar con los *wireframes* en el ordenador. A veces en este ejercicio surge, por ejemplo, que será necesario poner una notificación en la que no se había pensado, etc.

Normalmente, los casos de uso *Core* son accesibles desde el menú de navegación (Navigation Bar) y los casos de uso secundarios, se encontraran o bien dentro de las pantallas accesibles desde la Navigation Bar o bien dentro de un menú inferior o secundario.

En función de la arquitectura de información que se haya pensado, se optará por una combinación de estos tres tipos de patrones de navegación:

- **Patrón de navegación Básica:** basada en dos niveles de información. El primer nivel de información se muestra como una lista de opciones, mientras que el segundo nivel se

muestra como un detalle de información respecto a la selección de una de las opciones del listado

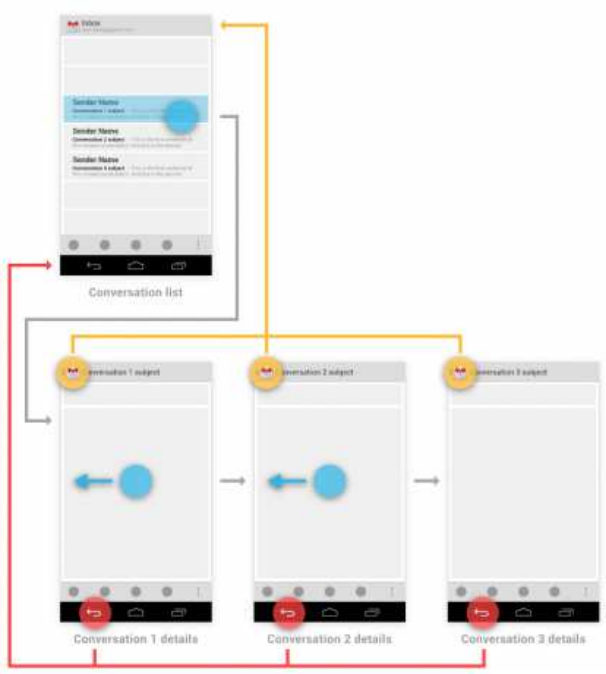

Ilustración 19: Patrón de Navegación básica

Es obligatorio que este tipo de navegación deberá adaptar su presentación al dispositivo donde se esté ejecutando la aplicación. De esta manera si el dispositivo es un Smartphone, la lista de opciones y el detalle de las mismas se representarán en vistas diferentes (primer la lista de opciones y luego, una vez seleccionada una de estas, el detalle). Mientras que si el dispositivo es una Tablet, tanto la lista de opciones como el detalle de las mismas se representarán en la misma vista, siendo la zona izquierda la ocupada por el listado de opciones y la zona derecha la ocupada por el detalle de la opción seleccionada

- **Patrón de navegación basada en menú lateral:** basada en un menú lateral *(drawer)* que contendrá el primer nivel de información y que estará situado en la zona superior izquierda de la vista, mientras que el detalle de información se presentará en toda la vista (mediante el repliegue automático del menú una vez se haya seleccionado una de las opciones disponibles)

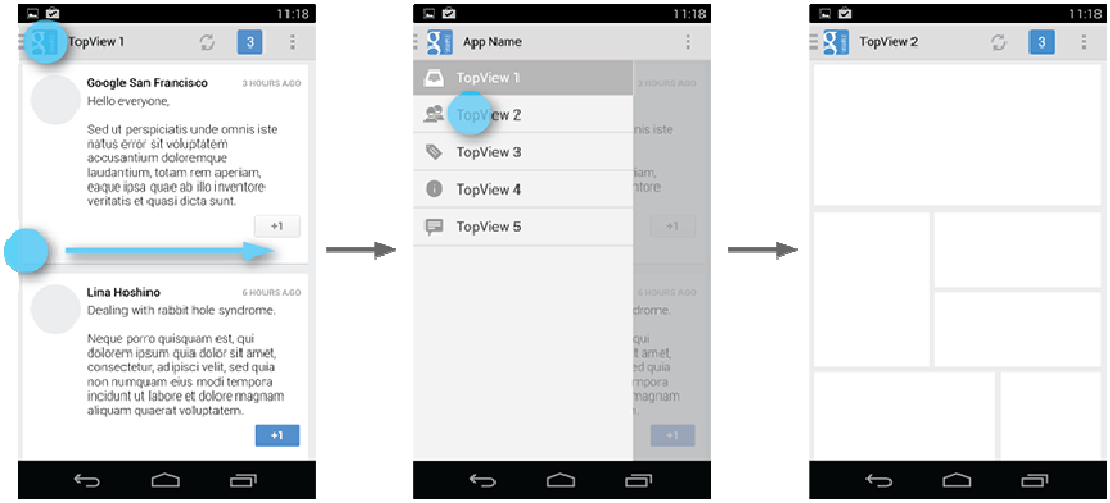

Ilustración 20: Patrón de Navegación con menú lateral

Aunque es técnicamente posible añadir un segundo nivel de información dentro del menú lateral, esta posibilidad queda completamente desaconsejada por la complejidad extra que añade a la interacción que debe realizar el usuario y a la rotura de la premisa de simplicidad que debe tener la estructura de información

No es obligatorio que este tipo de navegación adapte su presentación al dispositivo donde se esté ejecutando la aplicación

- **Patrón de navegación basada en *tabs (ViewPager)*:** basada en el uso de tabs, contendrá el primer nivel de información en la zona superior, mientras que el detalle de información se presentará en la zona inferior

Ilustración 21: Patrón de Navegación basada en "tabs"

La navegación entre las diferentes opciones presentadas en el primer nivel de navegación se podrá realizar de dos maneras: mediante la pulsación de los *tabs* asociados a las diferentes opciones o mediante un gesto de tipo *swipe* lateral en la zona de presentación (correspondiente al segundo nivel de información).

Existe una alternativa a este tipo de navegación basado en el uso de un componente gráfico combobox conteniendo el primer nivel de navegación, mientras que el detalle de información se presentará en la zona inferior. Esta alternativa aunque válida debe usarse solamente en aquellos casos en los que la aplicación de la alternativa basada en *tabs* no sea posible o se encuentre desaconsejada (p.ej: Muchos elementos de navegación en primer nivel)

No es obligatorio que este tipo de navegación adapte su presentación al dispositivo donde se esté ejecutando la aplicación.

Como se puede comprobar, el uso de un tipo de navegación en una vista de la aplicación dependerá de la arquitectura de información que se haya pensado para dicha vista. Estos flujos deben quedar bien reflejados en los *Wireframes*.

Cualquier aplicación de un patrón de navegación diferente a los enumerados en el presente documento (Navegación básica, Navegación basada en menú lateral, Navegación basada en *tabs*) deberá ser previamente valorado y aceptado por el Departamento de Calidad previa justificación.

Aunque los *Wireframes* pueden ser más o menos sintéticos según el experto de usabilidad los quiera representar, lo normal es que no tengan diseño. Existen una serie de elementos que son estándar, por ejemplo, el recuadro con la cruz dentro simboliza que habrá una imagen.

Lo más importante en la usabilidad de una App es que el usuario sepa siempre dónde se encuentra, y qué puede hacer o hacia dónde tiene que ir. El usuario tiene que poder siempre llegar a dónde quiere con el mínimo de clics posible, como en una web. Cuanto más fácil sea la interacción, antes aprenderá a usar la interfaz. Hay que tener siempre presente que ese aprendizaje se deberá realizar normalmente sin ningún tipo de explicación.

A veces se usa un breve tutorial, que sólo se muestra la primera vez que un usuario entre a la App. Aunque el uso de este recurso la mayoría de las veces es un indicativo de que no se ha presentado una interfaz lo suficientemente auto explicativa.

Para los *Wireframes* será necesario ordenar muy bien las ideas reflejadas en los *Mockups* y empezar a hacer las pantallas teniendo en cuenta los flujos y las normas de usabilidad. Es difícil innovar en usabilidad, puesto que ya existen unas *guidelines* bien definidas para las diferentes plataformas (iOS, Android, Windows Phone,…) que deben cumplirse como requerimiento para la publicación en sus respectivos markets.

El **tercer paso**, una vez finalizado el Desarrollo de UX, **es la realización del Diseño**. Para ello se deberán tener en cuenta los siguientes puntos:

3.1 Simplicidad del diseño

Al diseñar una aplicación para Android debe tenerse muy presente la fragmentación en cuanto a tamaños de pantalla y resolución existente en este tipo de dispositivos. Por ello, la aplicación debe proporcionar una disposición equilibrada y estéticamente agradable en su conjunto, ajustando su contenido a pantallas de diferentes tamaños y orientaciones. Por ello se deberán tener en cuenta las siguientes consideraciones:

- Se deberá priorizar el uso de los elementos gráficos nativos frente al de elementos *custom*, que tendrán un mayor impacto en el rendimiento y en el consumo de memoria, y por lo tanto una peor experiencia de usuario
- Un elemento pulsable debe ser mínimo de 48x48 pixels
- Los desarrolladores de Android disponen de una herramienta llamada *9Patch*. El *9Patch* es para poder adaptar los elementos recortados a los distintos tamaños de pantalla. Se aplica a todos esos elementos que tengan que adaptarse a la medida de la pantalla, como pueden ser botones, *backgrounds*, *textfields*, *combobox*...

Es imprescindible que para el *9Patch*, los elementos sean de corte recto (hay que evitar círculos o acabados circulares), puesto que la herramienta *9Patch* permite especificar qué pixels interesa multiplicar de un elemento para que se adapte al dispositivo.

Existe un programa llamado "Draw9Patch"; en este programa, cuando le introduces una imagen, se genera un pixel de más alrededor del elemento a multiplicar. En esta zona de un pixel que rodea el elemento el desarrollador pintará un pixel negro arriba y abajo, y un pixel negro en cada lado, estos puntos negros serán los que especificarán al programa Draw9Patch por donde debe multiplicarse en horizontal y vertical el elemento en cuestión. El *9Patch* multiplicará los pixels del elemento en vertical y horizontal, de forma que este elemento se adaptará siempre fielmente al diseño en las distintas medidas que sea necesario. Es por eso que en Android hay priorizar el diseño de elementos con acabados rectilíneos, puesto que los círculos y las elipses pueden dar problemas con el *9Patch* (puede inducir deformaciones en el proceso de redimensionamiento que realiza Android).

3.2 Presentación multi-pantalla

En los dispositivos más pequeños, el contenido generalmente se divide en una cuadrícula principal o vista de lista y la vista de detalle. Al tocar un elemento en la vista principal se abre otra pantalla que muestra la información detallada de ese elemento

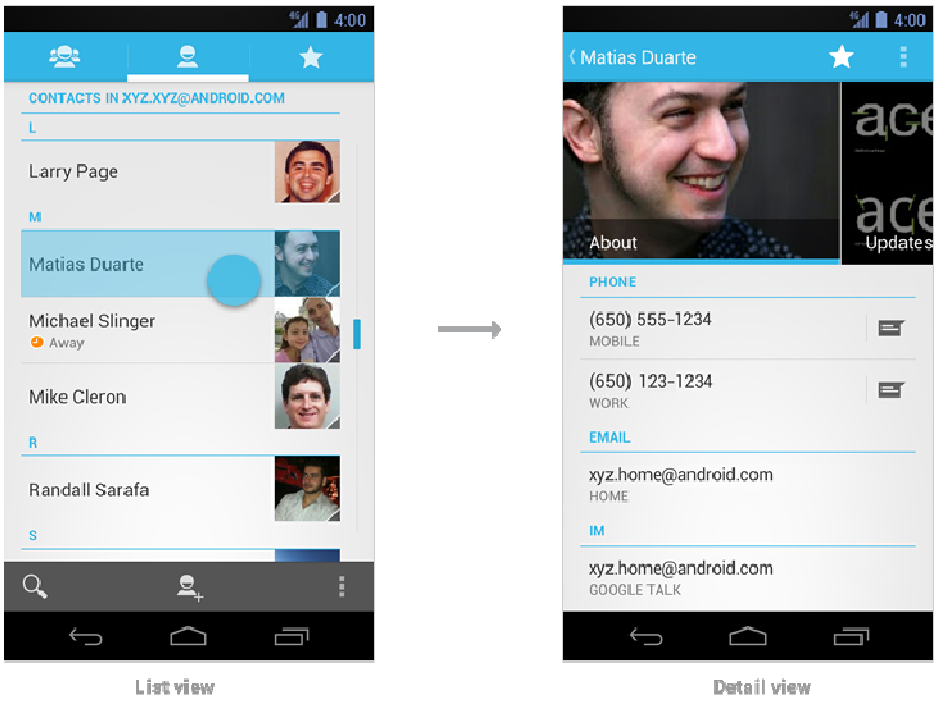

Ilustración 22: Presentación multipantalla

Las tabletas disponen de más espacio en pantalla, con lo cual se puede utilizar el panel para combinar la lista relacionada y las vistas de detalle en una única vista compuesta.

De esta forma se utiliza el espacio adicional de manera más eficiente y hace que la navegación sea más fácil.

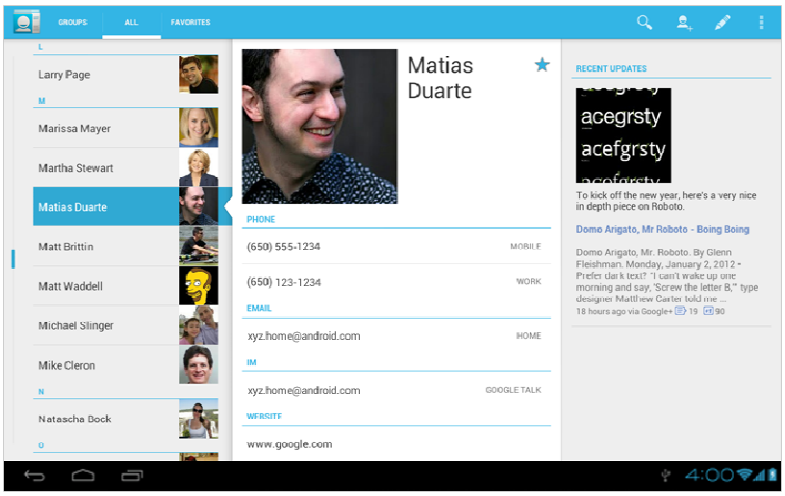

Ilustración 23: Relación entre paneles

El tema en el panel de la izquierda debe mantenerse seleccionado con el fin de establecer la relación entre los paneles.

3.3 Vistas compuestas y cambio de orientación

Las pantallas deben mantener la misma funcionalidad sin importar su orientación. Si se utiliza una vista compuesta en una orientación, en lo posible debe mantenerse cuando el usuario gire la pantalla.

Existen varias técnicas que se pueden utilizar para ajustar el diseño después de un cambio de orientación, manteniendo la paridad funcional intacta.

- **Expansión / compresión:**

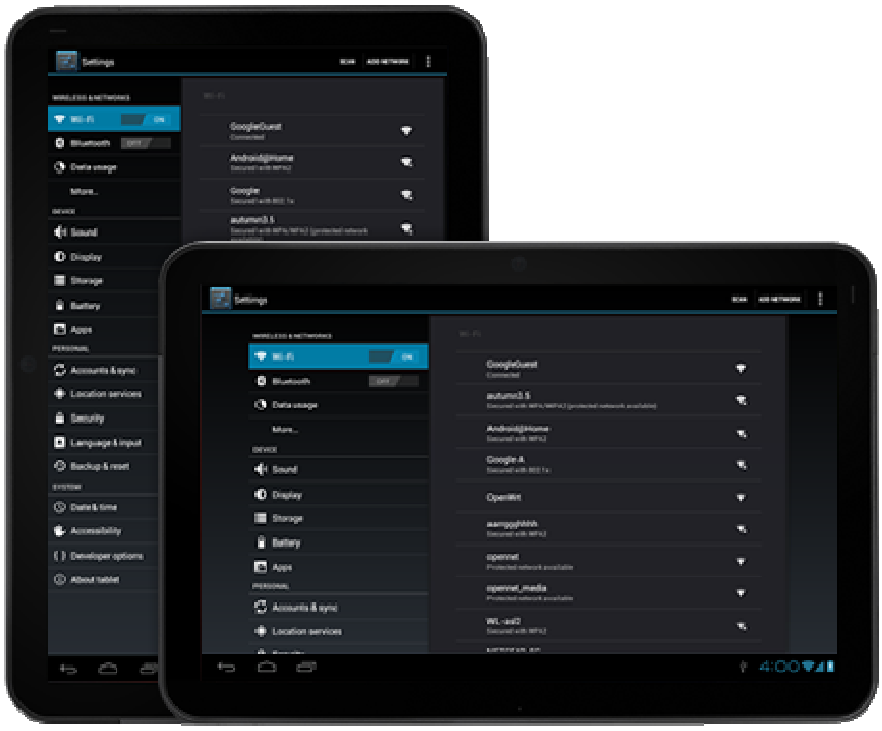

Ilustración 24: Expansión/comprensión

- **Apilar:**

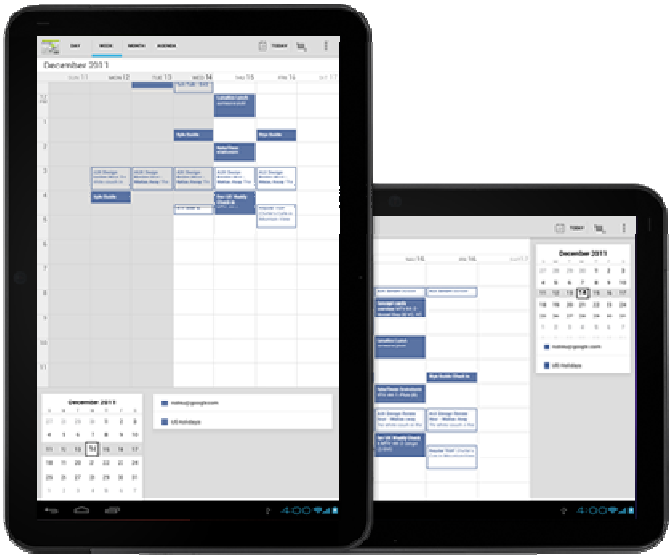

Ilustración 25: Apilar

- **Expansión / colapso:**

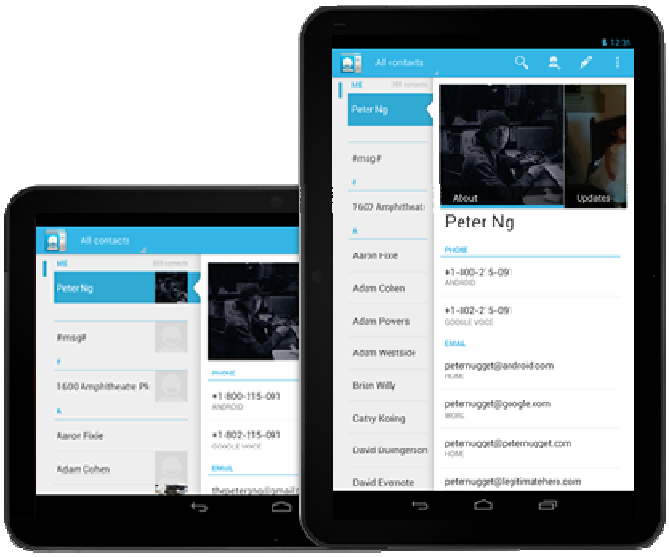

Ilustración 26: Expansión/Colapso

Mostrar / ocultar:

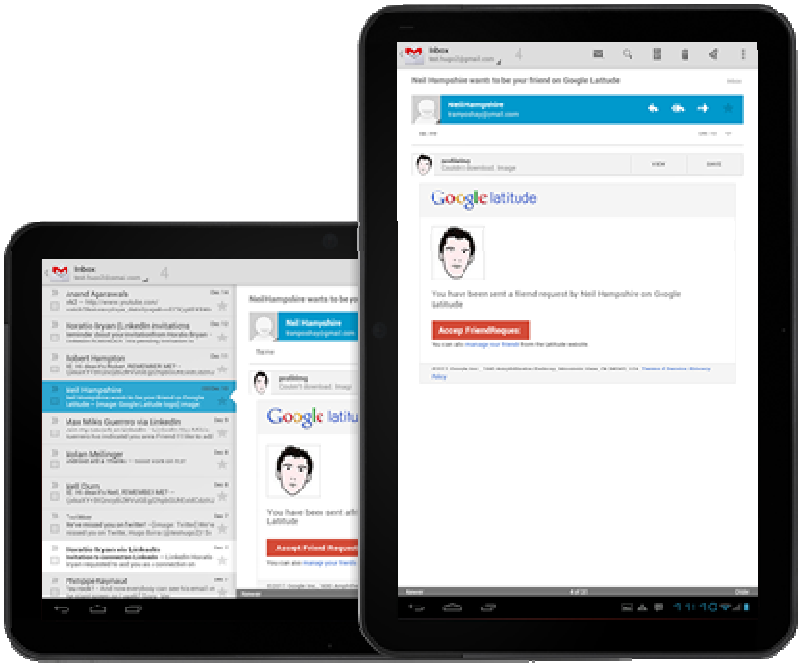

Ilustración 27: Mostrar/Ocultar

3.4 Widgets

Los widgets son un aspecto esencial de la personalización de la pantalla de inicio. Se pueden pensar como vistas rápidas de los datos que contienen. Los usuarios pueden mover los widgets a través de sus paneles de pantalla de inicio y, si es compatible, cambiar el tamaño para ajustar la cantidad de información a en el widget.

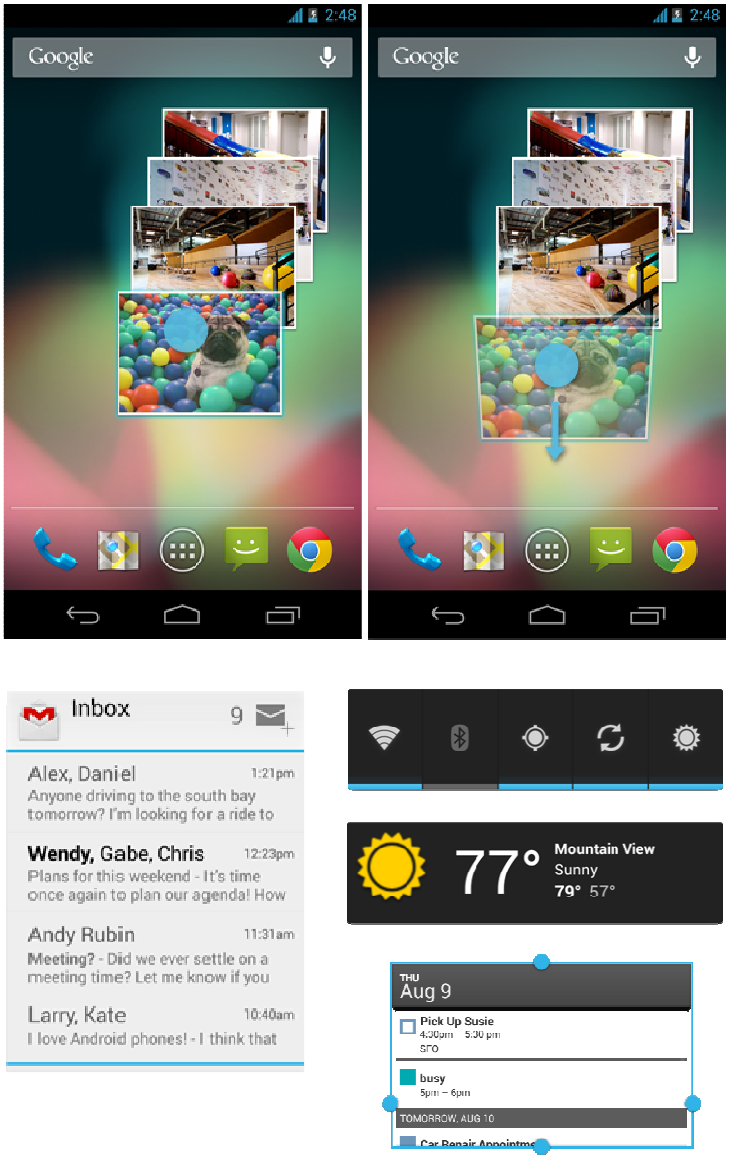

Ilustración 28: Ejemplo de widgets