

Diseño de la interfaz gráfica de la aplicación móvil

**Breve descripción:**

El estudio consciente de este componente formativo posibilitará al aprendiz identificar los principales lenguajes de programación para el desarrollo de aplicaciones móviles híbridas y generar análisis de los componentes gráficos más importantes y populares que componen una interfaz gráfica basada en el usuario. Podrá ampliar conocimientos y nociones básicas para realizar una apropiada interfaz.

**Septiembre 2023**

Tabla de contenido

[Introducción 1](#_Toc145600266)

[1. Interfaz 3](#_Toc145600267)

[1.1. Interfaz de usuario 3](#_Toc145600268)

[1.2. Historia de las interfaces 5](#_Toc145600269)

[2. Diseño de interfaces para aplicaciones móviles 7](#_Toc145600270)

[2.1. Legibilidad y contraste 9](#_Toc145600271)

[2.2. Retícula 10](#_Toc145600272)

[2.3. Color 11](#_Toc145600273)

[2.4. Ícono 11](#_Toc145600274)

[2.5. Ícono de lanzamiento 13](#_Toc145600275)

[3. Usabilidad y funcionalidad de las aplicaciones 14](#_Toc145600276)

[4. Diseño centrado en el usuario 17](#_Toc145600277)

[4.1. Fases del diseño de las aplicaciones 17](#_Toc145600278)

[4.2. Principios del diseño centrado en usuarios 18](#_Toc145600279)

[4.3. Técnicas para el diseño centrado en usuarios 18](#_Toc145600280)

[4.4. Ventajas del DCU 21](#_Toc145600281)

[5. Análisis jerárquico de tareas 23](#_Toc145600282)

[5.1. Secuencia 23](#_Toc145600283)

[5.2. Perceptiva 24](#_Toc145600284)

[5.3. La hipermedia 26](#_Toc145600285)

[6. Estructuras de navegación 27](#_Toc145600286)

[7. Dispositivos y visualización 30](#_Toc145600287)

[8. Prototipo de aplicaciones móviles 32](#_Toc145600288)

[9. Lenguaje de programación para desarrollo de “apps” 35](#_Toc145600289)

[9.1. Lenguajes utilizados para desarrollo Android 35](#_Toc145600290)

[9.2. Lenguajes iOS 36](#_Toc145600291)

[9.3. Lenguajes multiplataforma (“Cross-Platform”) 37](#_Toc145600292)

[9.4. Otros lenguajes de programación 40](#_Toc145600293)

[10. Navegación 41](#_Toc145600294)

[11. Componentes gráficos 44](#_Toc145600295)

[12. Gestos 46](#_Toc145600296)

[13. Estructura de navegación 47](#_Toc145600297)

[14. Características de una apropiada interfaz de usuario 49](#_Toc145600298)

[Síntesis 51](#_Toc145600299)

[Material complementario 52](#_Toc145600300)

[Glosario 53](#_Toc145600301)

[Referencias bibliográficas 56](#_Toc145600302)

[Créditos 59](#_Toc145600303)

Introducción

Para dar inicio al desarrollo de una aplicación, lo primero que se debe hacer es definir a quiénes irá dirigida. Este proceso proporcionará bases sólidas para las labores de diseño porque será un trabajo centrado en el usuario. Se deberán considerar las necesidades que pretenden cubrir, las edades, las características del público; definir la interfaz de usuario es una de las actividades más complejas y para ello es fundamental una excelente planeación, correcto y profundo conocimiento de sus componentes para lograr el éxito buscado de la aplicación, también es necesario tener claras todas las variables que se van a utilizar comprendiendo que un solo error, falla, falencia o recurso mal utilizado, puede generar inconvenientes para el correcto funcionamiento.

Lo invitamos a ver el siguiente video:

1. Diseño de la interfaz gráfica de la aplicación móvil



[**Enlace de reproducción del video**](https://www.youtube.com/watch?v=bFlCGeIHY_4)

|  |
| --- |
| **Síntesis del video: Diseño de la interfaz gráfica de la aplicación móvil** |
| En la dinámica agitada del mercado actual, los bienes y servicios que se ofertan y compran son medidos, fundamentalmente, por las percepciones del comprador, cliente o usuario.  Cuando la experiencia de uso de un producto, como las apps o sus entornos gráficos, es satisfactoria, las personas acusan esa satisfacción, la identifican, pero difícilmente ubican el origen de su satisfacción.  De hecho, no es una pregunta o reflexión que las personas consumidoras de apps se planteen con regularidad. Básicamente, se acepta tal experiencia de satisfacción y el producto que la provoca, así, sin más.  En esto radica, entonces, que las interfaces sean un elemento clave y que su diseño responda a las exigencias que, sin mucha consciencia a veces, son solicitadas por los consumidores de dispositivos electrónicos, “smartphones”, apps y otros servicios web.  Los lenguajes de programación y los componentes gráficos de la interfaz toman valor y sentido en la misma medida en que sus destinatarios hallan entornos familiares y, sobre todo, afines a sus intereses de actividad, tanto “online” como “offline”. |

# Interfaz

Una interfaz permite la intercomunicación entre un sistema informático implementado en algún dispositivo electrónico y su usuario. Este término proviene del inglés “interface”, que traduce “superficie de contacto”.

La finalidad es entregar una herramienta que permita, de manera sencilla, visualizar, introducir y editar información generando un mecanismo de interacción de uso.

* **Interfaces de usuario.** Comunicación entre el usuario y un sistema informático implementado en un aparato electrónico: Por ejemplo la pantalla en los “Smartphone”.
* **Interfaces físicas.** Permite la conexión física de dos o más dispositivos. Por ejemplo, los puertos USB de las computadoras.
* **Interfaces lógicas.** Permite realizar comunicaciones entre dos o más programas por medio de “software” que maneja protocolos de interacciones lógicas. Por ejemplo, una API.

## Interfaz de usuario

La interfaz de usuario o el “Front-End” de las aplicaciones móviles, programas y páginas web, sirve de mediador con el usuario de los computadores, “Smartphones” o todo aquel dispositivo electrónico que cuente con un “software”. Su función es ofrecer un entorno visual compuesto por gráficos, imágenes, íconos, y animaciones que contribuyen en la integración, visualización y comprensión de la información que brinda la aplicación y generando las herramientas necesarias para realizar las acciones de interacción.

La interfaz de usuario (UI) también se define como el conjunto de elementos gráficos que permite la interacción funcionando como controles y canales, por medio de los cuales los usuarios se comunican con el dispositivo electrónico. En el diseño de aplicaciones móviles es importante establecer una interfaz de usuario que permita la transferencia de información entre el usuario y la aplicación de manera fluida y sencilla, logrando una buena navegabilidad e interacción.

**Categorías**

Es posible encontrar varias categorías de interfaces de usuarios; estas se clasifican según la interacción que permite con el usuario y lo que se busca es facilitar la interacción entre la máquina y el receptor. Las personas suelen estar familiarizadas con las interfaces gráficas, pero existen otras formas de interacción que serán mostradas a continuación.

* **Interfaz gráfica.** Permite visualizar la información en una pantalla. La mayoría de programas poseen una interfaz visual, que genera la interacción entre el usuario y el “software”, permitiendo así la visualización, introducción y manipulación de datos.
* **Interfaz de voz (VUI).** Son programas desarrollados para identificar e interpretar el habla humana. Los ejemplos más claros que se encuentran en el mercado, actualmente, son Siri y Cortana que, utilizando inteligencia artificial, logran comprender órdenes y dar respuesta clara a las preguntas y consultas de los usuarios.
* **Interfaz de texto.** Es una interfaz simple que, por medio de una pantalla o línea de comando, permite visualizar y entregar la información. Su uso, en la mayoría de los casos, se lleva a cabo en entornos de programación de bajo nivel o configuración y operación de sistemas operativos.
* **Interfaz natural.** Se utiliza para capturar e interpretar los movimientos del cuerpo humano convirtiéndolos en acciones que se interpretan como órdenes. Es muy popular en los videojuegos que permiten este tipo de interacción por medio de la interfaz kinética. También se está implantando en entornos de realidad virtual y realidad aumentada.
* **Interfaz Cerebro-Ordenador.** Es una tecnología relativamente nueva. Se están realizando muchas investigaciones al respecto especialmente en el ámbito médico, donde se utiliza con pacientes con prótesis biónicas y por medio de la captura de ondas cerebrales pueden dar instrucciones sencillas a un “software” conectado al “hardware”.

## Historia de las interfaces

El inicio de las interfaces gráficas se da en el mismo momento que los computadores personales se ofrecieron de manera comercial, en la década de los años 70; antes de esto solo se utilizaba interfaz de texto que hacía el proceso de comunicación más tedioso, por su complejidad. Gracias al trabajo de IBM, Microsoft y Apple, en poco tiempo se lograron concebir unas herramientas gráficas que facilitaron la interacción con sus sistemas operativos y el “software” que se utilizaba en ese momento, utilizando la metáfora del escritorio de oficina como símil del entorno gráfico.

Los primeros productos de Apple y Microsoft e IBM contribuyeron en gran medida con las bases de diseño e interacción que aún son utilizadas, proponiendo al usuario un lenguaje claro y un entorno interactivo, basado en señales, íconos, hipervínculos y también en otras dinámicas que son propias y que representan las acciones del sistema informático.

Se lograron, entonces, entornos de trabajo mucho más familiares y comprensibles para la gran mayoría de usuarios, pasando de tener una pantalla en negro y con una línea de texto de comando, a un entorno totalmente gráfico e interactivo agradable que invita a la interacción.

**Historia de las interfaces**

Para tener más información sobre la historia y evolución de las interfaces de usuario, se le recomienda visitar la página [Historia de las interfaces](https://www.sutori.com/es/historia/historia-de-las-interfaces--NTbnEdSjkFagkT7KH42RZYF2).

# Diseño de interfaces para aplicaciones móviles

Con base en la teoría que fundamenta la creación de interfaces gráficas para páginas web y aplicaciones móviles (Vilchis, 1999) el diseño y la comunicación gráfica buscan proyectar los mensajes que el ser humano requiere para generar y establecer un orden jerárquico, estructural y significativo. Por esa razón se hace uso de los denominados códigos visuales, los cuales permiten estructurar, de manera lógica y significativa, grupos de elementos con los que se representan los mensajes visuales. Se pueden, según Vilchis, establecer seis códigos visuales:

* **El código morfológico.** Son los elementos gráficos que el diseñador usa para generar las estructuras y el lenguaje gráfico que utilizará en el entorno, algunos en su forma más básica ya están predefinidos y son comprendidos; estos pueden ser figuras geométricas, Regulares, etc. Ejemplo, en el iPhone, la distancia y forma de los íconos.
* **El código cromático.** Enmarca los esquemas de color que se utilizan en determinados diseños. Encierra la selección de intensidad, contraste, luminosidad y demás atributos que contienen la cromaticidad.
* **El código tipográfico.** Se entiende como las propiedades y atributos de los textos y sus caracteres, determinando las particularidades tales como tamaño, forma, peso y orientación.
* **El código fotográfico.** Define las imágenes y fotografías que serán trabajadas y abordadas bajo el mismo patrón, determinadas por el tipo de toma, por el encuadre y por las escalas, entre otras.
* **Códigos visuales.** Los códigos visuales están compuestos por todos los elementos mencionados anteriormente pero ya integrados y estructurados bajo un estilo particular alfabeto, tipografía, iconos, esquemas, ilustración, fotografía.
* **Códigos secuenciales.** Está compuesto por animaciones, video, zonas sensibles, estructuras de interacción.

Nota. <https://bit.ly/3AIoC6X>

En este mismo sentido y habiendo conocido los elementos gráficos más utilizados, es importante comprender cómo deben ser distribuidos en el espacio de trabajo, con el fin de realizar una correcta organización de los elementos necesarios para una buena visualización y navegabilidad.

La “ley de la buena forma”, demuestra que las personas usuarias buscan establecer su apreciación de la manera más clara y simple posible, minimizando todo a formas constantes y sencillas. Por esta razón los usuarios interpretan mejor lo que está sucediendo y el mensaje que se está ofreciendo en la interfaz si la información está organizada en formas sencillas.

De acuerdo con Sosa y Siegel (2015): las organizaciones en estructuras simples y jerarquizando los elementos de la interfaz, contribuyen a que el usuario pueda organizar y comprender mejor la información. Esto favorece la toma de decisiones para realizar una tarea. La utilización del contraste de color, por ejemplo, logra separar grandes bloques de información.

## Legibilidad y contraste

Es importante definir los colores y los tamaños del texto, al momento de realizar una lectura; el contraste debe permitir realizar esta actividad con facilidad. También se deben considerar temas como interlineados, espacios, entre otras, para realizar una lectura cómoda y fácil.

* **Manejo y estructura de la información.** Es fundamental el manejo de la organización de la información, de manera jerárquica iniciando con los títulos, los subtítulos, los pies de foto y algunos otros tipos de textos. Deben tener, desde luego, variadas configuraciones que permitan distinguir un texto de otro, de forma sencilla y que, también, sean de fácil reconocimiento para el usuario. Todos los elementos utilizados en la aplicación como los textos, los gráficos, íconos y todos los elementos adicionales, deben tener una consistencia y equilibrio, favoreciendo así la consistencia de estilo al sistema.
* **Familias tipográficas.** De acuerdo con Sosa y Siegel (2015): entre más completa esté una familia tipográfica se tendrá un mayor campo de acción, ya que se dispondrá de más variantes que se adecuen a textos largos, títulos, números en el cuerpo del texto, números en encabezados, pies de página, etcétera, estableciendo así los niveles jerárquicos dentro de la interfaz gráfica de usuario.
* **Tipografías limpias.** Cuello y Vittone (2013), recomiendan el uso de tipografías limpias abiertas como por ejemplo la fuente llamada “sans-serif”, porque es mucho más fácil visualizarlas dentro de las pequeñas pantallas que proporcionan los “smartphones” para la realización de lecturas largas.
* **Tamaño de tipografía en cada S.O.** El tamaño de la tipografía se mide en “scaled Pixels” o píxeles escalados; el tamaño utilizado en aplicaciones móviles va desde 12sp hasta 22sp. Es así como los títulos principales podrían medir hasta 34px. De esta manera, el tamaño en las etiquetas, dentro de botones importantes, es de 28px, aproximadamente. Partiendo de allí, va en disminución jerárquicamente hasta alcanzar el tamaño más pequeño que es muy cercano a los 14px. No obstante, la recomendación es no usar dimensiones que sean inferiores a los 20px en los textos que sean de lectura.

## Retícula

Estos elementos se utilizan para determinar la zona donde estará ubicada la información. Define las áreas y la división del espacio disponible, estableciendo las proporciones espacios y tamaños que tendrán los diferentes componentes de la interfaz.

Una retícula bien estructurada permite asegurar que los elementos contenidos podrán ser accedidos y tocados con el dedo de forma clara y sin problemas. Este es un tema fundamental para el correcto funcionamiento de la aplicación; definir correctamente el espacio de contacto de cada elemento dentro de la pantalla. Se debe, entonces, usar para el espaciado y la separación, un módulo de 8dp. En los márgenes laterales es muy común tener 16dp.

## Color

Cuello y Vittone (2013) definen los colores en categorías. Especialmente en el diseño gráfico para páginas web y aplicaciones móviles se utilizan, sobre todo, los llamados colores reservados que son: el rojo, el amarillo y el verde; incluyendo los colores señaléticos como el blanco, negro y azul. Estos colores tienen la particularidad que son, al mismo tiempo, color y signo porque implican, bajo su propio contexto, un significado.

* **Rojo.** Se utiliza para identificar errores y generar alertas importantes; consecuentemente indica peligro y esta tonalidad llama la atención rápidamente.
* **Amarillo.** Se emplea para indicar prevención; marca que las actividades a desarrollar o la acción a realizar implica la toma de una decisión que tendrá una consecuencia, generando una alerta al usuario.
* **Verde.** Se utiliza especialmente para mostrar mensajes de éxito y revalidación, indicando que una acción o actividad se ha realizado correctamente.

## Ícono

Los íconos son los elementos más utilizados en el ámbito del desarrollo de aplicaciones móviles ya que, por su tamaño y composición, permiten una fácil ubicación e interacción. Estos deben ser diseñados cuidadosamente para que las visualizaciones transmitan el mensaje correcto y el usuario entienda, de manera intuitiva, su intencionalidad. Pierce categorizó a los signos en tres grupos: los íconos, los índices y los símbolos.

* **Ícono.** El signo se parece a lo que representa. Ejemplo: uno de los íconos más populares es el de “like” de aprobación, que se utiliza en Facebook.
* **Índice.** Hay, de manera directa, una relación entre el signo y el objeto. Ejemplo: en los correos electrónicos, el lápiz posibilita identificar la acción de escribir un nuevo mensaje.
* **Símbolo.** Entre el signo y el significado, no existe conexión lógica; mayormente, se parte de convenciones de orden social, por ello su efectividad depende del nivel de saberes o conocimientos, además de experiencia, que el receptor tenga. Ejemplo: el avión de papel que nos indica enviar el mensaje o la nube que representa guardar la información en un repositorio remoto.

Nota. <https://bit.ly/3wr4sLA>

Habría que agregar también que los íconos son un lenguaje de signos y diseñar un signo implica, semióticamente, tres dimensiones. Según Royo (2004), estas dimensiones son:

* **Dimensión semiótica.** Es la relación directa de la imagen con su significado explícito. Existe una relación directa entre el ícono, la imagen y el mensaje. Como, por ejemplo: la papelera de reciclaje.
* **Dimensión semántica.** Define un sistema visual en el cual todos los elementos están relacionados y permite identificarlos como una sola estructura.
* **Dimensión funcional.** Busca relacionar el icono con la utilidad que el usuario puede identificar, relacionando su significado de manera directa. Respondiendo a qué tan legible es el ícono para los usuarios y la acción a gestionar.

## Ícono de lanzamiento

Se propone el ícono de lanzamiento como el símbolo que identifique a la aplicación. Este ícono será el referente gráfico que relaciona a los usuarios con el producto. Es de suma importancia generar un elemento gráfico con la mejor definición simbólica posible, que sea distintiva, que tenga elementos de originalidad, que también favorezca hacer distinción de la “app” con respecto a las demás, que tenga visibilidad y resalte en el “stock” de la tienda de descargas y en la lista de aplicaciones que hayan sido instaladas, por el usuario, en su teléfono inteligente.

En definitiva, las características de orden visual, han de comunicar expresa y efectivamente, el objetivo principal que tenga la aplicación y el estilo que define a la aplicación.

# Usabilidad y funcionalidad de las aplicaciones

Para realizar el diseño de una oportuna interfaz gráfica para aplicaciones móviles, es necesario tener claros algunos parámetros importantes que se han utilizado y que mejoran el rendimiento y la usabilidad.

* **Eficacia o efectividad.** La aplicación se considera efectiva cuando los usuarios logran concretar y realizar los objetivos específicos para los cuales fue desarrollada la aplicación. Se puede medir con la tasa de errores detectados y la cantidad de soluciones ofrecidas.
* **Eficiencia.** Se trata de la correlación que existe entre el esfuerzo usado, con la precisión y la certeza con la que las personas usuarias de apps alcanzan a realizar sus actividades. Los esfuerzos y los recursos que se usen, han de ser menores y la eficiencia ha de ser mayor. Si se pretende valorar la eficiencia, algunos indicadores podrían ser el tiempo de tardanza en finalizar una tarea y el tiempo de aprendizaje.
* **Satisfacción.** Es la respuesta positiva entregada por el usuario o cliente por la realización correcta y eficiente de las actividades se puede medir con la periodicidad que este la realiza y sus muestras positivas hacia el uso del producto.

Nota. <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-1:v1:en>

**Funcionalidad de interfaces**

Al comprender la importancia que tiene el concepto de usabilidad, en la etapa de diseño, se evitarán muchos inconvenientes y retrasos porque se identificarán, a tiempo, las características principales que debe tener la aplicación.

1. Usabilidad y funcionalidad



[**Enlace de reproducción del video**](https://www.youtube.com/watch?v=jV5krdruvd0)

|  |
| --- |
| **Síntesis del video: Usabilidad y funcionalidad** |
| En el diseño de apps para dispositivos móviles, se hace fundamental que los principios y criterios técnicos al diseñar la interfaz, aseguren la calidad, rendimiento y usabilidad útil y familiar para los usuarios.  Para despertar interés en el usuario y fomentar actitudes positivas relacionadas con la percepción de productos y servicios, se emplea el diseño estético, con creaciones y composiciones naturales e integrales, que creen sentimientos de lealtad y aceptación.  La interfaz debe favorecer al usuario, de manera que este use su intuición en la búsqueda y navegación.  Debe conceder a los usuarios la toma de sus propias decisiones, utilizando componentes de estado actualizados y de fácil acceso.  El uso de colores es muy importante, pero no deben dejarle de lado señales secundarias que garanticen transmitir el mensaje claro y completo.  Los valores o configuraciones por defecto deben ser apropiados y en lo posible deben ser los más comunes.  Es ideal que los elementos de control estén siempre visibles y de fácil acceso, sin dejar de lado que en las aplicaciones móviles es poco práctico exponer elementos que no son necesarios todo el tiempo. |

# Diseño centrado en el usuario

Contar con información previa sobre las necesidades de consumo, estilos y tendencias en el plano tecnológico, entre otras, son factores clave en el diseño centrado en el usuario.

Esta información es relevante para la fase de producción y soluciones de diseño. En ella, por ejemplo, se determina cómo será la estructura principal de la aplicación, definiendo cada una de las funciones y tareas que debe realizar la aplicación móvil para cumplir con los requerimientos solicitados e identificados con el cliente o los usuarios.

## Fases del diseño de las aplicaciones

Existen cuatro fases establecidas para el diseño de aplicaciones móviles, definidas por la norma ISO 13407, que integran el proceso del DCU, diseño centrado en el usuario y que están, además, concebidas bajo seis principios que deben caracterizar una interfaz centrada en el usuario. Las dos primeras fases tienen especial intención de documentar las necesidades y requerimientos del usuario.

1. Comprender y hacer específico el contexto de uso.
2. Especificar los requisitos de usuario y de la organización.
3. Producir soluciones de diseño.
4. Evaluar los diseños en base a los requisitos.

## Principios del diseño centrado en usuarios

Los seis principios que deben caracterizar una interfaz centrada en el usuario apuntan, sin lugar a dudas, a una satisfacción plena del usuario con las aplicaciones que tenga a su disposición:

* Planificación.
* Especificación de requisitos.
* Especificación del contexto de uso.
* Genera la solución del diseño.
* Cumple con los requerimientos.
* Evaluación.

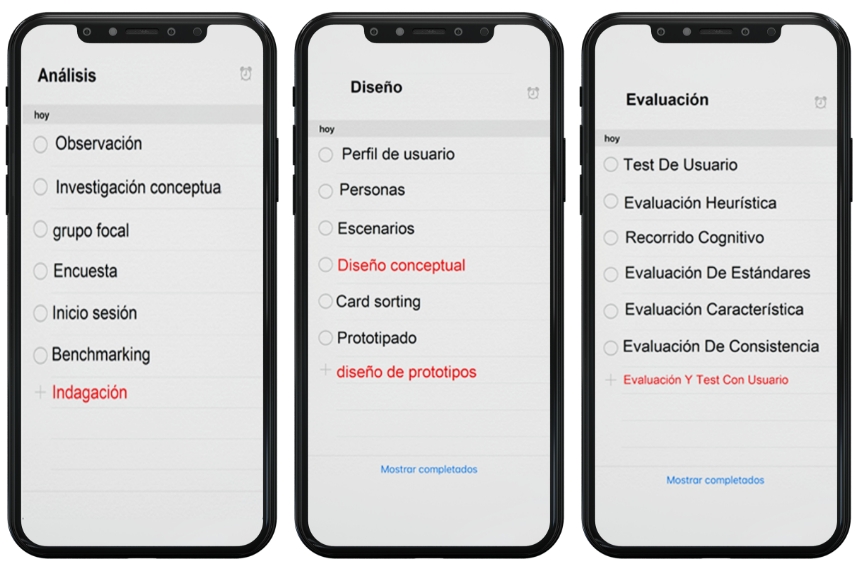
Nota. Presentación Introducción Al Desarrollo De Aplicaciones Móviles. (2020)

Esta responsabilidad con las personas que hacen uso de los teléfonos inteligentes y los servicios de las aplicaciones responde, además, a las demandas de la dinámica de consumo actual, a la creciente oferta de servicios que conectan a los individuos con el mundo, con las otras personas, con sus intereses y que, entre otras, dinamizan y facilitan su vida y actividades diarias.

## Técnicas para el diseño centrado en usuarios

Conocer y comprender las necesidades, limitaciones, comportamiento y características del usuario, sumado al modelo de las tres etapas; análisis, diseño y evaluación, desarrollado en cuatro fases específicas, ya mencionadas, abren la puerta para considerar las técnicas para un óptimo diseño de aplicaciones centradas en el usuario.

1. Técnica por fases para el desarrollo de un diseño centrado en el usuario



Nota. Presentación Introducción Al Desarrollo De Aplicaciones Móviles. (2020).

* **Conocimiento directo y real.** El diseño se centra en el conocimiento directo y real de los usuarios las tareas a realizar y el entorno. Es preciso comprender a los usuarios, conocer lo que quieren hacer con el sistema y comprender el entorno en el que se va a utilizar el sistema.
* **Usuarios involucrados.** Los usuarios son partícipes a través del diseño y desarrollo. Este principio permite que el equipo de trabajo involucre de manera directa a los usuarios en todas las fases, no solo en la ejecución de un “focus group” en el inicio del diseño o mediante la aplicación de una encuesta al final de la implementación sino durante todo el proceso de desarrollo.
* **Evaluación centrada en el usuario.** El diseño es depurado y pulido por la evaluación centrada en el usuario. Las pruebas deben llevarse durante todo el transcurso de diseño, por lo que también sirve para probar diseños preliminares, como prototipos en papel.
* **Proceso interactivo.** Es complejo que los usuarios manifiesten de manera clara y sencilla lo que quieren de un sistema. Por esa razón para definir lo que requieren los usuarios, se debe mostrar algo que probablemente no quieren y luego, se irá expresando cómo mejorarlo.
* **Consideración de emociones del usuario.** El diseño se debe ocupar de toda la experiencia del usuario. Este principio indica que el diseño ya no solo busca hacer a la interfaz usable y simple, sino que debe comprender los aspectos de percepción y emoción agrupados a la experiencia del usuario.
* **Participación de expertos.** El equipo de diseño incluye habilidades y perspectivas multidisciplinares. Los conocimientos del diseñador y del programador no son los únicos necesarios para el correcto funcionamiento de la aplicación; se debe incluir la opinión de expertos en accesibilidad, el equipo de “marketing”, escritores técnicos, y especialmente la de los usuarios finales.

## Ventajas del DCU

El diseño por fases y la observancia de los principios del diseño centrado en el usuario, favorecen un alto nivel de cumplimiento de expectativas no solo para quien hace uso de las aplicaciones y sus servicios sino también para las compañías u operadores que las ofrecen. Sin embargo, hay algunas ventajas que se presentan con anterioridad a la oferta y uso de la aplicación móvil.

* **Indagación.** La indagación se realiza en la etapa de Especificación del contexto de uso y de los requisitos contextualizando el uso e identificando los requerimientos. Está enfocada en el levantamiento de información que permita definir las características principales del producto o servicio se trabaja con los métodos cualitativos de observación, entrevistas, dinámicas de grupo y métodos cuantitativos encuestas y cuestionarios.
* **Diseño conceptual.** El diseño conceptual parte de la recolección y análisis realizado en la indagación.

Permite entender y analizar las características de los usuarios y sus necesidades definiendo el uso que le darán a la aplicación móvil es importante esta información porque permite orientar el diseño que se debe implementar para suplir la necesidad.

* **Diseño de prototipos.** En esta fase se elaboran los prototipos de baja y alta resolución. Son actividades propias de los diseñadores, estos diseños permiten visualizar la solución de la interfaz. En este punto son relevantes los conocimientos sobre comunicación, semiótica y retórica.
* **Inspección y test con usuarios.** Se utilizan las pruebas de usabilidad y test de navegación aplicados a los usuarios que permiten obtener información sobre los diseños para mejorarlos y validarlos.

# Análisis jerárquico de tareas

El análisis jerárquico de tareas HTA, por sus siglas en inglés: “Hierarchical Task Analysis”, permite establecer y describir las acciones por realizar, paso a paso, de manera secuencial, identificando y describiendo cada una de las tareas y actividades, enfocando en las operaciones y procedimientos.

Se prevé que, con el cumplimiento de una serie de tareas concretas, las personas, programas u organizaciones alcanzarán el logro de determinado objetivo o meta. Así mismo, el análisis jerárquico de tareas contempla la creación de un entorno suficiente y válido para poder ejecutar las acciones planeadas. El diseño de aplicaciones móviles, el desarrollo de la interfaz y todos los demás elementos que integran el proceso de programación de las mismas no escapa a este HTA.

## Secuencia

Una secuencia es un conjunto de elementos que se ordenan en una determinada sucesión, unos detrás de otros o unos delante de otros. Estas secuencias, por lo general, se presentan como un conjunto de información ordenada. En ese sentido, se habla de tres variables de una secuencia: Selección, Interacción y Tarea unitaria. La primera de ellas, la Selección se refiere a cuando dentro de un conjunto de tareas se tiene que elegir una de ellas; por su parte, la Iteración tiene que ver con la repetición de un subconjunto de tareas y, finalmente la Tarea unitaria se trata de una actividad única que no tiene actividades subordinadas.

En el marco de la programación de aplicaciones para dispositivos móviles, la Secuencia busca responder:

* **Necesidad de información concreta.**

Por ejemplo: “¿Qué costo tiene este producto?”

* **Necesidad de información orientada a problemas.**

Por ejemplo: “¿Cuál es la relación entre la usabilidad y la arquitectura de información?”

* **Necesidad de información exploratoria.**

Por ejemplo: “Quiero escoger un buen hotel para este fin de semana”.

* **Necesidad de información sobre búsquedas previas,** es decir, localizar información que ya había localizado previamente.

**Análisis de una tarea en Gmail con método HTA**

Para ampliar los conocimientos sobre el análisis jerárquico de tareas, lo invitamos a ver el PDF Análisis de una tarea en Gmail con método HTA, el cual se encuentra en la carpeta Anexos.

## Perceptiva

La perceptiva se refiere a todos los elementos que hacen parte de la configuración del entorno, como los colores, las tipografías, logotipos y demás elementos gráficos que componen la interfaz. La perceptiva puede ser funcional, jerárquica o secuencial. Funcional, en tanto la atribución del rol que juegan dichos elementos, es decir, para qué sirven. Jerárquica porque busca distinguir cuáles son prescindibles y cuáles esenciales dentro del grupo de actividades. Por último, será secuencial en tanto toma información del flujo normal y lógico del proceso, permitiendo precisar al usuario una secuencia para lograr la realización de una actividad.

Estos son los pasos básicos para realizar una estructura o una arquitectura de información:

* **Revisión del material previo.** Una exploración de los requerimientos y una investigación de los productos que se asimilen al producto a diseñar. En palabras de Norman (1990): “las pistas acerca de cómo funcionan las cosas proceden de su estructura visible: en particular de sus prestaciones y limitaciones y de su topografía”.
* **Identificación de objetos.** Es importante identificar cada uno de los componentes gráficos y de información que contendrá la aplicación móvil, generando una lista ordenada y clasificada por elementos a utilizar.
* **Evaluación del contenido.** Es fundamental realizar una revisión previa de los objetos clasificados generando un inventario que servirá también como un repositorio de los elementos que componen el diseño.
* **Aplicar la técnica de “card sorting”.** Los resultados de esta técnica suelen ser representados en dendrogramas. Es una representación gráfica o diagrama de datos, en forma de árbol, que organiza los datos en subcategorías que se van dividiendo en otras hasta llegar al nivel de detalle deseado.

## La hipermedia

La hipermedia es entendida como una estructura ligada a la recuperación de la información y al diseño de los sistemas de búsqueda, de ayuda y de navegación; estos sistemas son las estrategias con las que el usuario cuenta para localizar la información y asimilar, y luego reformular su necesidad, hasta que considere que ha quedado resuelta.

* **Búsqueda.** El usuario utiliza la opción de búsqueda y analiza e interacciona con los resultados.
* **Navegación.** Se presenta cuando el usuario explora utilizando los enlaces hipertextuales.
* **Ayuda.** Es importante generar herramientas que permitan al usuario recurrir por una asistencia que le permita solicitar una orientación y le señala en dónde se encuentra el contenido que necesita o cuál es el procedimiento a seguir para lograr un objetivo.

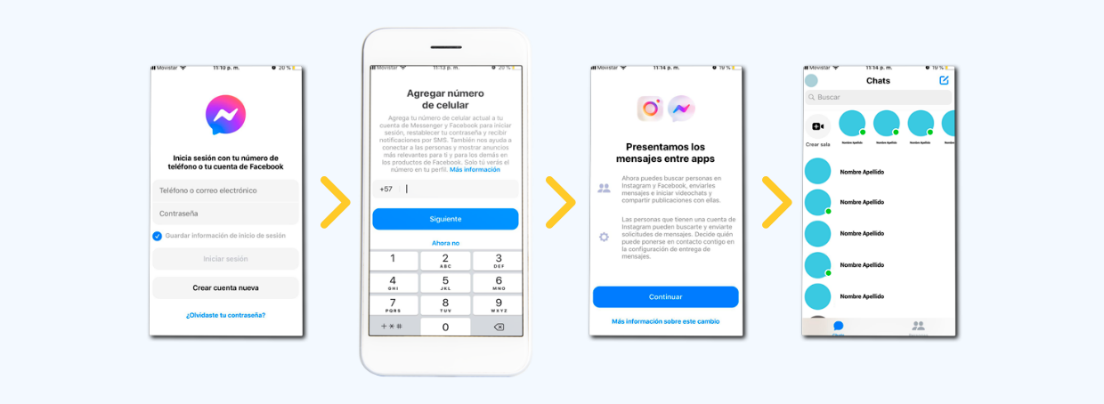
# Estructuras de navegación

Los servicios ideales que ha de prestar una aplicación móvil o cualquier sitio web, suponen para el usuario, la posibilidad de derivar de un punto a otro dentro del entorno, incluso fuera de él hacia otros entornos, para acercarse o acceder directamente a información, acciones, servicios, material, entre otras. Royo (2004) clasifica a las estructuras de navegación en tres tipos: lineales, jerárquicas y aleatorias.

**Estructuras lineales**

Generan una lectura secuencial y son usadas para tareas muy bien definidas, por ejemplo, los tutoriales que muchos programas y apps ejecutan en su primera ejecución, para instruir al usuario sobre su funcionamiento.

1. Estructura lineal Messenger secuencia lineal para iniciar sesión



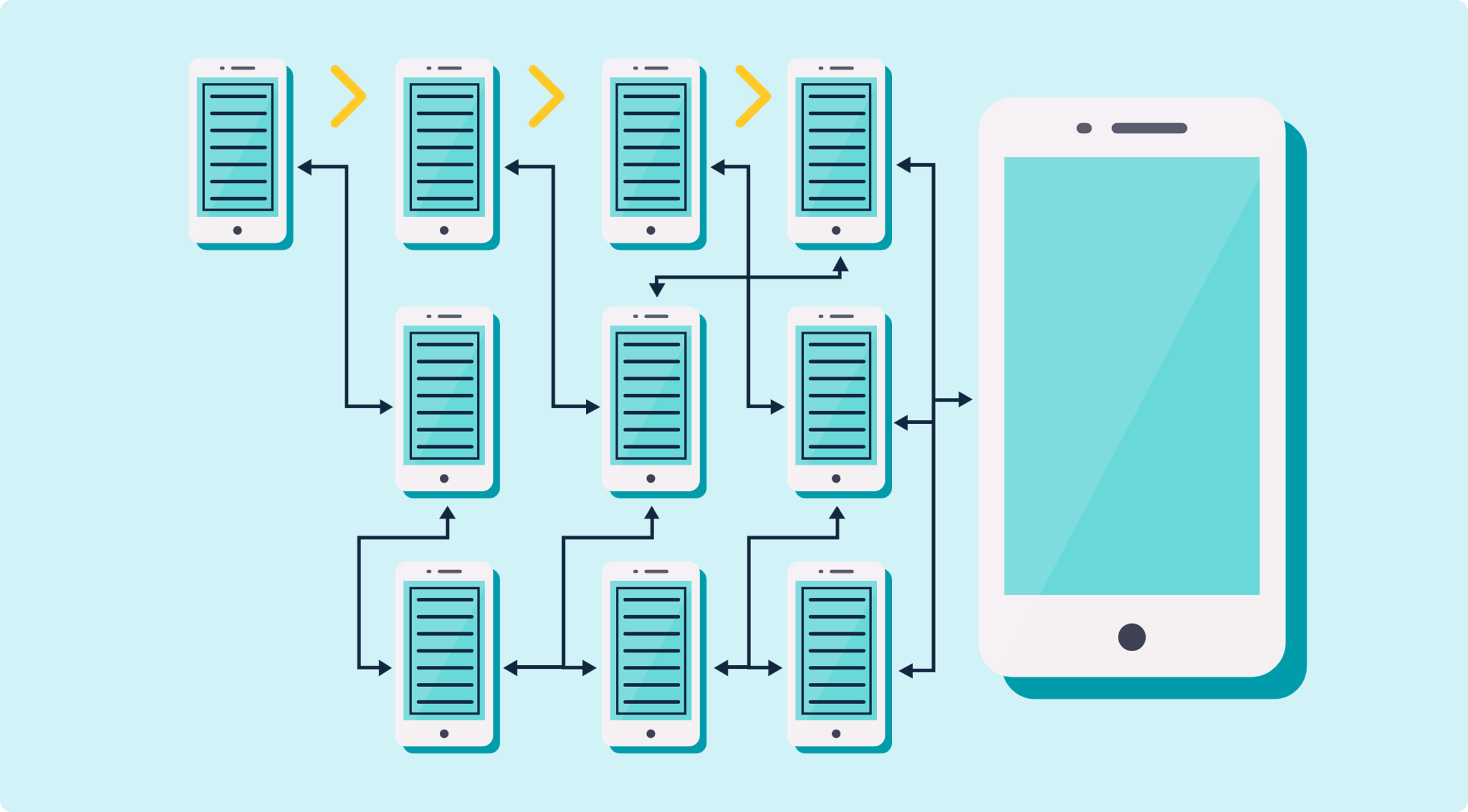
Nota. Presentación Introducción Al Desarrollo De Aplicaciones Móviles. (2020)

**Estructuras jerárquicas**

Estas estructuras permiten elegir un número de posiciones antes de pasar al siguiente nivel el cual también presenta diferentes opciones. Antes de continuar se maneja de manera ordenada ofreciendo una organización lógica y secuencial que permite identificar, de manera sencilla, dónde se encuentra la información. Existen varios tipos de secuencias que se adaptan y se utilizan según las necesidades de diseño.

Estas estructuras permiten elegir un número de posiciones antes de pasar al siguiente nivel el cual también presenta diferentes opciones. Antes de continuar se maneja de manera ordenada ofreciendo una organización lógica y secuencial que permite identificar, de manera sencilla, dónde se encuentra la información. Existen varios tipos de secuencias que se adaptan y se utilizan según las necesidades de diseño.

1. Estructura jerárquica

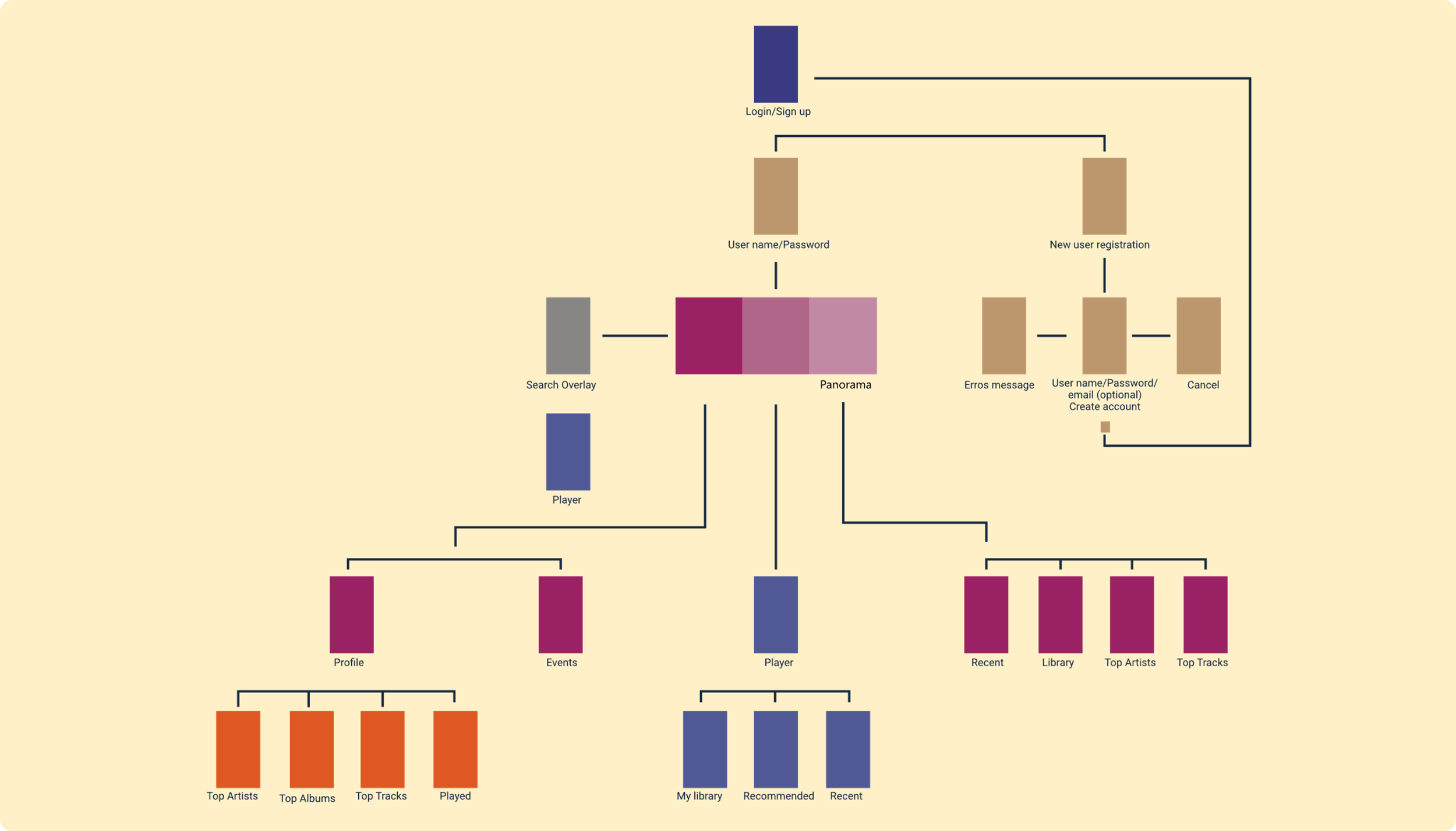


Nota. Presentación Introducción Al Desarrollo De Aplicaciones Móviles. (2020)

**Estructuras aleatorias**

Estas estructuras son utilizadas especialmente en el diseño de videojuegos o sistemas abiertos de interacción, donde el aprendizaje es complejo y se lucha contra el sistema o contra otros jugadores en red. Por ejemplo, al cargar una imagen o un escenario diferente, en cada actualización de la pantalla.

1. Estructura de una app



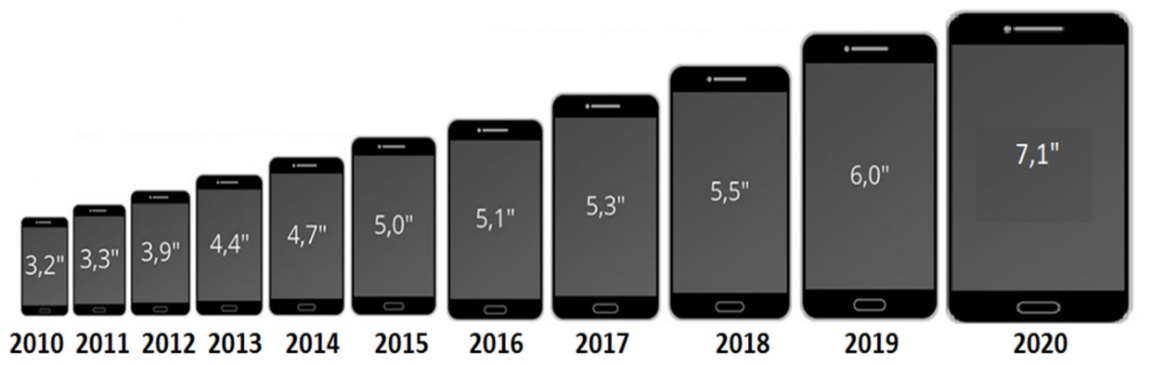
Nota. <http://appdesignbook.com/es/>

Se presenta una estructura jerárquica equilibrada de una app, desde el momento que el cliente se loguea o si es nuevo y necesita registro, hasta llegar a la información que desea.

# Dispositivos y visualización

El elemento primordial de la visualización en un dispositivo es la resolución de pantalla; para comprender qué es la resolución, es importante definir qué son píxeles: estos definen e indican cuál es la resolución de una pantalla. De la cantidad total de píxeles del teléfono, tableta, televisor o monitor, dependen las resoluciones de vídeo que se pueda llegar a alcanzar.

1. Tamaño de dispositivos y visualización



Nota. Presentación Introducción Al Desarrollo De Aplicaciones Móviles. (2020)

Se presenta cómo ha evolucionado el tamaño de las pantallas de los dispositivos móviles desde el año 2010 hasta el año 2020.

**Píxeles**

La pantalla viene dividida en “superficie de contacto”; elemento de imagen que consiste en puntos diminutos presentes en toda el área de la pantalla, y permiten que se visualice una imagen. Se puede entender un píxel como el tamaño más pequeño que una imagen puede tener.

**Imagen y dimensiones de la pantalla**

Una manera clara de comprenderlo es: si tenemos una resolución HD de 1280 x 720, quiere decir que la resolución de pantalla está conformada por 1280 columnas de 720 píxeles por columna. Es como una matriz. Por regla general, el primer número hace referencia a la anchura; el segundo, a la altura de la pantalla.

**Estándar de resolución**

Las resoluciones de las pantallas están definidas por estándares de resoluciones VGA, XGA, HD, Full HD, 1440p y 4K, entre otros, cuanto mayor sea la resolución, mejor será la calidad.

Tamaño de la pantalla

De forma predeterminada, el tamaño de la pantalla se mide en pulgadas (“inch”, en inglés). Cada pulgada equivale a 2,54 centímetros o 25,4 mm y también puede ser representada con comillas posteriores a la cifra, por ejemplo: 3,2″ (3,2 pulgadas).

# Prototipo de aplicaciones móviles

Realizar el prototipo de la aplicación móvil permite visualizar de manera clara la estructura gráfica que va a tener la aplicación; permite identificar cada uno de los elementos gráficos que se utilizarán en la app. Es el primer acercamiento que se tiene de la apariencia que adquirirá el “front-end”; esta es una de las técnicas utilizadas para representar cómo va a quedar la aplicación a desarrollar. Es el primer diseño y la primera actividad que podrán evaluar los clientes o usuarios. Se puede realizar a mano alzada o utilizando una herramienta de maquetación o prototipado; dichas herramientas son fáciles de utilizar y no hay necesidad de programar o manejar algún lenguaje de programación.

Ejecutar esta actividad ahorrará mucho tiempo y trabajo, ya que permite ver, de manera muy fiel, cómo va a quedar la app y si es necesario optimizar el diseño o cambiar la apariencia y usabilidad de la misma. Este prototipo será la primera entrega que se comparta a los clientes para su validación.

1. Herramientas para la creación de prototipos

| Herramienta | Descripción |
| --- | --- |
| Balsamiq | Esta es una de las herramientas más utilizadas para la creación de “wireframes” y “mockups”. Es fácil de usar y resulta ser muy intuitiva para el usuario, facilita el diseño para lograr un buen “mockup” y no permite interacciones con el prototipo. |
| Lucidchart | Presenta una interfaz que opera arrastrando y soltando los elementos para crear el “mockups”. Se puede trabajar en grupo, compartiendo el proyecto con otros usuarios. |
| MarvelApp | Su ventaja es la accesibilidad a distintas plataformas, como puede ser PC, móvil o web. Para trabajo colaborativo, se sincroniza con Dropbox. Se pueden exportar prototipos como si fuera un APK; es útil para desarrolladores Android. |
| UXPin | Esta herramienta cuenta con muchas opciones para “wireframing”. Posibilita la conversión de un boceto a un “wireframe” digital. No solo es la imagen o el pantallazo. No es de uso gratuito. |
| Protoshare | Facilita el trabajo en equipo, brinda gran cantidad de opciones y es una herramienta clara en el diseño de la interfaz. |
| Proto.io | Presenta un diseño simple, fácil de usar. Es una herramienta muy valorada para el prototipo; cuenta con numerosas opciones gráficas para integrar interfaces para iOS, Android, y Windows Phone. |
| iPhoneMockup | Esta herramienta se presenta como una buena opción para los que buscan pasar su diseño del papel al plano digital, de manera sencilla. Cuenta con varios estilos y diseños que permiten utilizar gran cantidad de elementos, diseñando un “mockup” para iOS fácil y rápido. |
| Wirify | Se basa en el trabajo utilizando un “script”, que tiene la posibilidad de convertir la interfaz en un sitio web, en un “wireframe”. Esta herramienta se usa para las creaciones entorno desarrollado, permitiendo la publicación y fácil acceso para la evaluación de interfaz y su funcionamiento. |
| Pidoco | Esta herramienta permite generar un enlace URL del prototipo diseñado, que puede ser compartido con los usuarios o el cliente, para realizar una retroalimentación. |
| Wireframe.cc | Esta herramienta es fácil de usar, con un concepto minimalista, permite con mucha facilidad cambiar los diseños sobre diferentes plataformas, computadores de escritorio, portátiles, “tablets” o “smartphones”. |
| Gliffy | Esta herramienta es gratuita y permite realizar trabajos “online” para la creación de mockups de un sitio web o aplicación móvil. Utiliza librerías de esquemas; también cuenta con una API que permite la integración de las aplicaciones a diferentes elementos externos. |
| Cacoo | Facilita la creación de “mockups” para sitios web o apps. Tiene una opción gratuita que solo permite exportar el proyecto final en PNG. Es ideal si el objetivo es diseñar la estructura del proyecto y tener una visión completa de la aplicación y su funcionalidad. Permite el trabajo colaborativo en tiempo real, muy útil para los grupos de trabajo interdisciplinares. |
| Mockflow | Es una herramienta interactiva que permite la creación del sistema de mapas de las páginas y pantallas a utilizar en el proyecto. Realiza exportación en varios formatos, cuenta con varias librerías y diferentes plantillas que permiten elegir la más adecuada para representar y organizar el contenido de la aplicación. |
| Moqups | Moqups es una herramienta similar a Balsamiq. Está desarrollada bajo HTML5, lo que le convierte en una herramienta rápida y sencilla. |

Nota. Tomada de Introducción al desarrollo de aplicaciones móviles (2020).

# Lenguaje de programación para desarrollo de “apps”

Para elegir el mejor lenguaje de programación, es importante tener claro qué tipo de aplicación se desea desarrollar y, de esa manera, seleccionar la mejor herramienta, que supla las necesidades y brinde ventajas, según los requerimientos.

Cada lenguaje cuenta con unas características y funciones especiales que brindan ventajas según el tipo del proyecto por desarrollar.

## Lenguajes utilizados para desarrollo Android

Android es el sistema operativo más utilizado e implementado en dispositivos móviles; su popularidad está fundamentada en su versatilidad y su rápida adaptación. Cuenta con un entorno de desarrollo integrado llamado Android Studio para el desarrollo de aplicaciones Android. Su primera versión estable fue publicada en diciembre de 2014. Kotlin es su principal lenguaje de desarrollo, pero también se puede utilizar Java y C ++.

**Java**

Java es el lenguaje de programación más utilizado, cuenta con gran soporte por la extensa comunidad de desarrolladores que utilizan este lenguaje como su principal herramienta. En este momento, se puede usar Java con los programas como Netbeans, Eclipse y uno de los más utilizados en el desarrollo de aplicaciones móviles, Android Studio.

Para afianzar su comprensión sobre lenguajes de programación Kotlin y conocer mejor sus particularidades, le invitamos a visitar la página [Descripción general de Kotlin](https://developer.android.com/kotlin/overview?hl=es).

## Lenguajes iOS

iOS es otro de los sistemas operativos con mucha demanda y reconocimiento en el mercado tecnológico. Su entorno es bastante aceptado entre millones de usuarios y para él se crean y desarrollan múltiples “apps”.

1. **Swift.** Swift es el lenguaje utilizado en el desarrollo de software para iOS. Apple ha añadido algunas características geniales a este lenguaje, tales como la sintaxis simplificada y la capacidad de localizar con precisión los errores del programador. Algunas de sus características principales son:

* Permite trabajar con valores de retorno múltiples.
* Manejo de errores nativos usando “try ... catch” y “throw”.
* Es posible realizar métodos, extensiones y protocolos facilitando la programación.
* En sus sintaxis no es necesario escribir los puntos y comas para finalizar la línea.
* Iteración rápida y concisa con colecciones.
* Utiliza el paradigma de programación funcional, que se fundamenta en el uso de verdaderas funciones matemáticas.

1. **Objective-C.** Este lenguaje fue desarrollado para trabajar y realizar proyectos iOS. Es compatible con los Sistemas Operativos OS X e iOS exclusivamente, y brinda la posibilidad de utilizar el paradigma de la programación orientada a objetos. Sus características principales son:

* Se fundamenta en el lenguaje de programación C; hereda su sintaxis primitiva e instrucciones de control del flujo, definición de clases y métodos.
* Utiliza la tecnología “Grand Central Dispatch” (GCD), que permite una mejor estructuración de tareas simultáneas o asíncronas.
* Tiene soporte para gráficos de objetos y literales a nivel de código.

## Lenguajes multiplataforma (“Cross-Platform”)

Los lenguajes multiplataforma se denominan de esta manera por la sencilla razón que son compatibles con varias y distintas plataformas. Esta particularidad aplica no solo para los lenguajes de programación sino, también, para los mismos sistemas operativos, aplicaciones, programas o “software”.

1. **JavaScript.** JavaScript cuenta con una larga trayectoria que se remonta a los inicios de la “World Wide Web”. Este es un lenguaje muy utilizado para el desarrollo “front-end”, permite trabajar fácilmente el diseño y desarrollo de páginas web y aplicaciones móviles; se puede lograr y alcanzar una buena e interactiva experiencia de usuario.

Se considera como uno de los mejores lenguajes para realizar el desarrollo “front-end” de aplicaciones móviles; también es posible trabajar back end con la ayuda de “frameworks”, como Node JS.

1. **¿A quién pertenece?** JavaScript es propiedad de la fundación Mozilla; a su alrededor se ha generado gran cantidad de “frameworks” que lo hacen más asequible y fácil. El desarrollo de estas son algunas de las mejores y más utilizadas herramientas.
2. **Características de JavaScript.** Las características más relevantes que podemos mencionar son (Seydnejad, 2016):

* Tiene la comunidad más grande de desarrolladores que brinda soporte a JavaScript y sus “frameworks”.
* Las herramientas de JavaScript para desarrollo móvil siguen creciendo y evolucionando, adaptándose a las necesidades del mercado.
* Cuenta con gran cantidad de herramientas enfocadas en el desarrollo de aplicaciones móviles.
* JavaScript es un lenguaje multiplataforma que corre y se ejecuta en la mayoría de los dispositivos móviles.
* Soporta programación orientada a objetos.

1. **“Frameworks” de JavaScript.** Estas herramientas están desarrolladas específicamente para el desarrollo y trabajo de aplicaciones móviles, como Ionic2 y React Native. Tiene entornos de desarrollo que facilitan el desarrollar apps multiplataforma. Con estos “frameworks” y sus librerías, se pueden generar todo tipo de aplicaciones móviles. Su enfoque principal es el desarrollo híbrido, esto quiere decir que, realizando una sola versión de la aplicación, se podrá implementar y funcionará correctamente en iOS o Android.
2. **“Frameworks” más usados.** Estos son algunos de los “frameworks” más utilizados en el desarrollo de aplicaciones móviles híbridas:

* React
* Native JS
* Ionic JS
* NativeScript
* React JS
* Angular JS
* Ember JS
* Vue JS

1. **TypeScript.** TypeScript es un subconjunto de JavaScript que ofrece mayor y mejor soporte para el desarrollo de apps a gran escala. Este lenguaje de programación está sustentado, soportado y mantenido por Microsoft. TypeScript fue creado con el fin de permitir la escritura de código para aplicaciones móviles multiplataforma, utilizando “frameworks” como NativeScript.
2. **Aportes y características.** TypeScript ayuda a compilar el código depurado y dejar un código simple y limpio para que se ejecute en cualquier dispositivo que soporte JavaScript. Sus características principales son:

* Cuenta con soporte para las últimas versiones de JavaScript.
* Se integra con el soporte de comandos NPM, que son los gestores de paquetes para JavaScript, sobre la consola de comandos.
* Cuenta con soporte en los principales editores de código, como Sublime Text, Visual Studio Code, WebStorm.

## Otros lenguajes de programación

Existen otros lenguajes para crear aplicaciones móviles, como C#, C y C++. Estos lenguajes pertenecen a Microsoft tienen un muy alto potencial. También encontramos a Python y Ruby, que permiten el desarrollo de aplicaciones multiplataforma.

* **C#.** C# es el idioma de Windows Mobile. Es muy similar a C++ y Java. Microsoft ha adoptado algunas de las características de Java para simplificar su arquitectura, manteniendo el C++ como diseño. También tiene una gran y activa comunidad de desarrolladores.
* **Python.** Python es un lenguaje que va tomando cada vez más popularidad; es fácil de aprender y fácil de comprender. Los creadores lograron trabajar en mantener una sintaxis simple y clara. Esto ayuda a los desarrolladores novatos a mantener altos niveles de productividad. Para desarrollar aplicaciones móviles multiplataforma, se utiliza Kivy.

Todos estos lenguajes de programación de aplicación nativa son compilados, en lugar de interpretados.

# Navegación

Lo que define la interacción del usuario con la “app”, su interfaz y el entorno general son los patrones y estructuras de navegación, planteando la manera en la cual el usuario recorrerá la aplicación, estableciendo el entorno gráfico y la interacción que realizará con los contenidos.

1. **Pestañas.** Las pestañas se utilizan para realizar el cambio entre pantallas y se trabaja sobre un mismo nivel jerárquico. Es importante resaltar o remarcar la pestaña principal y no usarlas para incluir otras acciones distintas a la navegación, para que el usuario tenga la comodidad y facilidad de tener un punto de partida. Google sugiere no emplear más de siete pestañas, pero en iOS, por ejemplo, se muestra un máximo de cinco pestañas.
2. **Lista.** Por lo general, las listas son muy similares en los tres sistemas operativos más populares: Android, iOS y Windows Phone. Las listas pueden estar integradas por texto, imágenes, botones y elementos gráficos que permitan identificar contenidos. Es importante siempre jerarquizar los contenidos en las listas. Google recomienda la agrupación de ítems relacionados para facilitar la comprensión del usuario al momento de realizar la navegación y la toma de decisiones.
3. **Menú tipo cajón.** Por lo general, este menú se encuentra ubicado y se despliega en el lado izquierdo, permite la visualización de los contenidos por medio de una lista que está compuesta por un título principal y subtítulos contenedores de información; permanecen ocultos y sólo se despliegan cuando son solicitados. Una de las grandes ventajas es el aprovechamiento del espacio, presentando una manera cómoda de navegar sobre contenidos nutridos o con gran cantidad de información.
4. **Acciones.** Se trata de los íconos que aparecen, por lo general, en la parte superior de las pestañas y se mantienen visibles a lo largo del recorrido. Representan las opciones más relevantes de la aplicación y permiten la toma de decisiones, por ejemplo: la opción salir, siguiente, y otras más. Dichas acciones deben ubicarse de acuerdo con su jerarquía y funcionamiento; todas las acciones pueden ser representadas por signos; la mayoría de estas se pueden realizar solamente en determinadas páginas, sin embargo, pueden existir excepciones cuando sea necesario que una acción esté siempre visible. (Cuello y Vittone, 2013)
5. **Cuadros de diálogo.** Se trata de casos en los que se debe interrumpir al usuario para que tome una decisión o para brindar una información necesaria. Cuando estos cuadros están visibles en pantalla, el usuario no puede hacer otra cosa en el resto de la aplicación; estos cuadros, por lo general, bloquean la pantalla hasta que el usuario toma una decisión.

El cuadro de diálogo busca reafirmar la decisión tomada por el usuario. Presenta dos opciones, la intención principal es evitar que el usuario cometa errores al momento de tomar decisiones apresuradas. Por esta razón, se solicita, generalmente, corroborar la decisión tomada.

1. **Barra de la app.** Esta barra se puede ubicar en la parte superior o inferior y contiene los íconos más importantes de la aplicación. Por lo general, se encuentra visible y es de fácil acceso para el usuario; en ella se pueden encontrar opciones como compartir o buscar. La barra de la app contribuye a generar un orden claro y lógico de la presentación, proporcionando un despliegue de acciones que no son tan comunes o que se utilizan con poca frecuencia. Es uno de los elementos más relevantes en el diseño, pero no siempre está incluida o hace parte de todas las aplicaciones móviles; puede ser implementada únicamente en las pantallas que lo requieran. (Cuello y Vittone, 2013)

* Despliegue de acciones. Estas funciones son unos elementos extras donde se organizan actividades a las que se recurre con poca frecuencia; por lo general, se encuentran ocultas a la espera de ser desplegadas.
* En Android, se ubican dentro de un botón de tres puntos verticales que las abre en forma de lista.
* Buscar. Se utiliza en algunas apps, sobre todo en aquellas que tienen gran cantidad de datos; es un elemento que facilita realizar algunas actividades a los usuarios.
* En todos los SO, es común encontrarlo como un cuadro o botón de búsqueda en la parte superior; se localiza dentro de la barra de acciones.

# Componentes gráficos

Antes de iniciar la elaboración del boceto y el “mockup”, se debe tener la idea clara de lo que se necesita desarrollar. El informe de diseño proporciona esta información, de manera clara y concisa.

Entonces, es importante conocer las opciones gráficas con las cuales se cuenta para el diseño de la interfaz. Asimismo, considerar el uso, anatomía, comportamiento, tematización y especificaciones de cada uno de los principales componentes de la interfaz gráfica de una aplicación móvil.

1. Principales componentes de una interfaz gráfica

| Componente | Ubicación |
| --- | --- |
| Barras de la aplicación: | parte inferior |
| Barras de la aplicación: | parte superior |
| “Banners” | No aplica |
| Barras navegación inferior | No aplica |

| Componente | Inglés |
| --- | --- |
| Botones: | “Buttons” |
| Botón de acción flotante: | “Buttons: floating action button” |
| Tarjetas: | “Cards” |
| Casillas de verificación: | “Checkboxes” |
| “Chips”: | “Chips” |
| Tablas de datos: | “Data tables” |
| Diálogos: | “Dialogs” |
| Divisores: | “Dividers” |
| Listas de imágenes: | “Image lists” |
| Listas: | “Lists” |
| Menús | “Menu” |
| Cajón de navegación: | “Navigation drawer” |
| Carril de navegación: | “Navigation rail” |
| Indicadores de progreso: | “Progress indicators” |
| Botones de opción: | “Radio buttons” |
| Hojas- abajo: | “Sheets- bottom” |
| Listas: | “Lists” |
| Hojas- lado: | “Sheets- side” |
| Deslizadores: | “Sliders” |
| “Snackbars”: | “Snackbars” |
| Interruptores: | “Switches” |
| Pestañas: | “Tabs” |
| Campos de texto: | “Text fields” |
| Recolectores de tiempo: | “Time pickers” |

Nota. Tomada de Introducción al desarrollo de aplicaciones móviles (2020).

# Gestos

Los gestos son los tipos de movimientos y pulsaciones que el usuario realiza sobre la pantalla del smartphone para activar las funciones de los diferentes elementos que hacen parte de la interfaz gráfica de usuario. Estos gestos y movimientos permiten la interacción con la aplicación.

**Principales gestos**

Los principales gestos y los más utilizados son tocar, arrastrar, o deslizar, ver, salir, entre otros. Para realizar estos movimientos, se requiere como máximo de dos dedos; para la mayoría de los usuarios, estos gestos son muy comunes y populares.

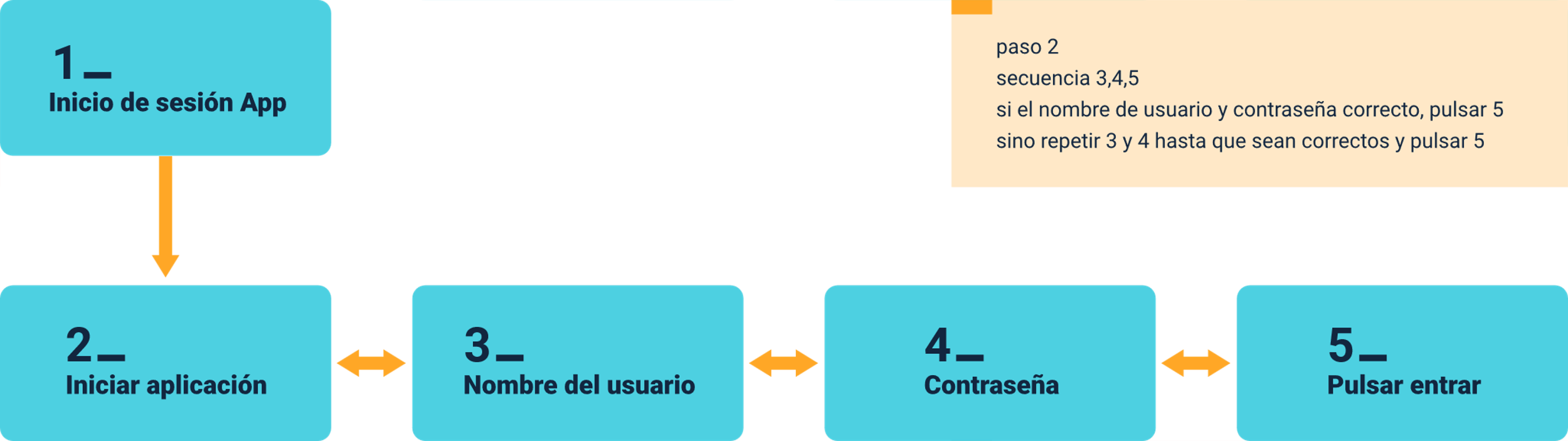
1. Descripción principales gestos

| # | Gestos | Android | iOS | Windows Phone |
| --- | --- | --- | --- | --- |
| 1 | Tocar. Tocar la superficie con la yema del dedo. | Selecciona la acción primaria. | Selecciona la acción primaria. | Selecciona la acción primaria. |
| 2 | Arrastrar. Mover sin perder el contacto. | Archivar, borrar o mover elementos. | Archivar botón de eliminar o mover elementos. | Cambiar a otras pestañas. |
| 3 | Deslizar. Deslizar perdiendo el contacto. | Cambiar a otras pestañas. | Desplazarse por el contenido. | Cambiar pestañas. |

# Estructura de navegación

La estructura de navegación define el paso a paso que se debe seguir para cumplir o realizar una actividad. Consiste en determinar la manera en que el usuario explorará los contenidos a través de las pantallas, íconos hipervínculos y demás elementos gráficos que permiten la interacción con la aplicación. Debe ser intuitiva y predecible.

1. Estructura de navegación para inicio de sesión



(Raffino, 2020)

La estructura de navegación desde el inicio hasta ingresar es la siguiente:

* Paso 1: inicio de sesión en la app.
* Paso 2: iniciar la aplicación.
* Paso 3: nombre del usuario.
* Paso 4: contraseña.
* Paso 5: pulsar entrar.

Hay que tener en cuenta que si el paso 3 o 4 con incorrectos, se repite hasta que sean correctos y continúe con el paso 5.

* **Navegación con pantallas deslizantes.** Puede ser por medio del gesto “deslizar” o bien pulsando sobre la pestaña. Subnivel de navegación con pestañas deslizantes. Subnivel de navegación con pantallas que no deslizan, por lo tanto, dependen de un elemento pulsable.
* **La estructura o mapa de navegación.** Define los caminos que debe tomar el usuario. Estas posibles rutas deben estar definidas y muy claras en el diseño, deben presentar el esquema de la interfaz indicando los principales conceptos incluidos en el espacio de la información y las interrelaciones que existen entre ellos.
* **Recorridos rápidos, accesos fáciles.** Es una representación completa de la aplicación que sirve para orientar al usuario en el recorrido, facilitando el acceso directo al lugar que le interese. El mapa de navegación se puede representar de forma textual, también en forma gráfica o con una combinación de ambas.
* **Comprensión modular de la app.** Para ofrecer una visión global, es muy útil mostrar un mapa de todo el contenido, poder verlo en una sola pantalla y que esté accesible, pero es complejo lograr este cometido; por esa razón, es importante dividir el mapa por actividades, para representarlo por módulos o en pequeños procesos que permitan comprender, de manera particular, cada uno de los componentes de la aplicación. (Cuello y Vittone, 2013)

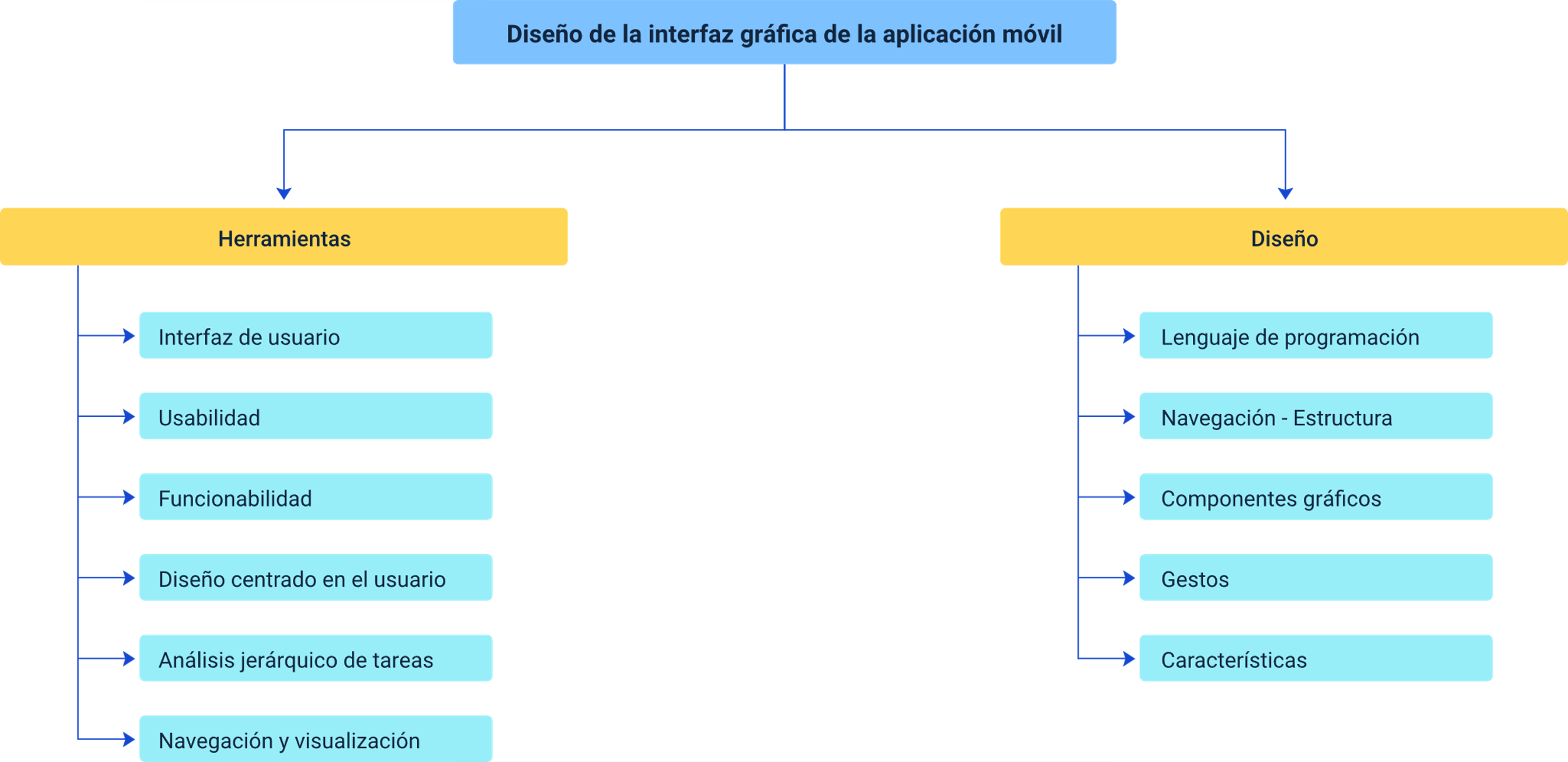
# Características de una apropiada interfaz de usuario

La tecnología actual cuenta, en su gran mayoría, con interfaces de usuario que permiten la comprensión e interacción de su funcionamiento. El diseño de una interfaz busca experiencias que sean intuitivas y funcionales para todos los que realizan una acción en la app o, sencillamente, interactúan con ella. Para ello se deben tener en cuenta algunas características que son claves para tener un resultado óptimo del proyecto.

* **Simplicidad.** Esta característica permite que el usuario pueda usar de forma fácil, fluida e intuitiva la aplicación. Dicha simplicidad se basa en el diseño minimalista, evitando saturar el entorno de elementos que no se requieran o que sean poco funcionales. Siempre enfocada en que cada elemento cumpla su objetivo y mejore la experiencia del usuario.
* **Claridad.** La claridad se basa en ofrecer los mensajes precisos, en el momento oportuno, mejorando y utilizando un lenguaje justo y conciso, que no se preste para ambigüedades, que permita al usuario una navegabilidad rápida, sin grandes obstáculos, evitando retrasos y, por consiguiente, una insatisfactoria experiencia para el usuario.
* **Coherencia.** Para mantener la coherencia, se debe trabajar específicamente con patrones que definan los tipos de elementos a utilizar, como las imágenes y sus características, los tipos de las fuentes, el lenguaje y su enfoque, el tono y la fluidez de la comunicación, el uso de los colores institucionales o representativos de la aplicación, la ubicación del menú y el logo y los íconos.
* **Familiaridad.** Se deben utilizar elementos comunes, por ejemplo, elementos gráficos universales, los íconos que ya hacen parte del lenguaje simbólico que, en su gran mayoría, los usuarios ya reconocen porque son utilizados en otras aplicaciones y hacen parte de la cotidianidad tecnológica. También es importante trabajar con la ubicación de estos elementos, ya que se puede trabajar bajo una estructura común conocida por los usuarios y utilizada en otras aplicaciones móviles.
* **Rapidez.** La fluidez en el desplazamiento por la aplicación y en la ejecución de acciones se logra cuando se integran las cuatro características: simplicidad, claridad, coherencia y familiaridad. Los usuarios suelen estar apurados, no cuentan con mucho tiempo para la ejecución de las actividades y ese es un valor agregado que les satisface en gran medida; ello se puede identificar en la aceptación o desaprobación de las aplicaciones.

Síntesis

A continuación, se presenta una síntesis de la temática estudiada en el componente formativo.



El esquema presenta la síntesis de la temática estudiada en el componente formativo, comenzando por el diseño de la interfaz gráfica de la aplicación móvil, compuesto por:

* Herramientas: interfaz de usuario, usabilidad, funcionabilidad, diseño centrado en el usuario, análisis jerárquico de tareas y navegación y visualización.
* Diseño: lenguaje de programación, navegación y su estructura, componentes gráficos, gestos y características.

Material complementario

|  |  |  |  |
| --- | --- | --- | --- |
| Tema | Referencia | Tipo de material | Enlace del recurso |
| 4.2 Principios del diseño centrado en usuarios | Campus Party. (2012). CPQuito2 - Principios de diseño de experiencia de usuario para Mobile. | Video | <https://www.youtube.com/watch?v=vVZ2F89MiyI> |
| 4.2 Principios del diseño centrado en usuarios | Jose Rafael Rojas B. (2021). Diseño y Desarrollo de Aplicaciones Móviles 01 - Introducción. | Video | <https://youtu.be/2STnjpA8XQc> |
| 8. Prototipo de aplicaciones móviles | Rivera, K. (2020, 24 junio). Fastest way to design a Mobile App : Using Balsamiq Mockups. | Video | <https://www.youtube.com/watch?v=RKr9u4S0W7g> |
| 8. Prototipo de aplicaciones móviles | yReyALD. (2020, 22 octubre). BALSAMIQ MOCKUPS / Instalación y maquetación de una aplicación móvil - FÁCIL (música GOT de fondo). | Video | <https://www.youtube.com/watch?v=idX3aFx67Vc> |
| 8. Prototipo de aplicaciones móviles | Programando Ideas. (2020, 17 junio). MOOC 3 - Apps para dispositivos móviles 2020 | Planificación del desarrollo de apps [Vídeo]. YouTube. | Video | <https://youtu.be/1qhScXdYGdM> |
| 11. Componentes gráficos | Material Design. (s. f.). Components. | Página web | <https://material.io/components?platform=android> |
| 11. Componentes gráficos | Material Design. (s. f.). Banners. | Página web | <https://m2.material.io/components/banners#anatomy> |

Glosario

**Antipatrón:** un antipatrón de diseño es un patrón de diseño que, invariablemente, conduce a una mala solución para un problema.

**“Benchmarking”:** se trata del proceso de creación, recopilación, comparación y de análisis de indicadores, que posibilite hacer medición del avance de los procesos y de las funciones de mayor importancia, dentro de una organización. Tales indicadores son conocidos como “benchmarks” y suelen servir como un estándar de éxito de las compañías.

**Estándar ISO 13407: “**Human-centred design processes for interactive systems” [ISO99] constituye un marco que sirve de guía para conseguir el desarrollo de sistemas interactivos usables incorporando el DCU durante el ciclo de vida del desarrollo.

**Hipermedia:** la hipermedia ayuda a hacer la designación del conjunto de métodos o de procedimientos del proceso de escritura, diseño o composición de contenidos que conforman soportes. Estos pueden ser: imágenes, videos, textos, audios, mapas y otros soportes emergentes, de información; así, lo que se obtiene como resultado, también, da la probabilidad de nuevas interacciones para los usuarios.

**Hipertexto:** es un sistema de organización y presentación de datos que se basa en la vinculación de fragmentos textuales o gráficos a otros fragmentos, lo cual permite al usuario acceder a la información no necesariamente de forma secuencial sino desde cualquiera de los distintos ítems relacionados.

**Ley de Fitts:** hace referencia a la rapidez y precisión con que ocurre el movimiento de los músculos en los humanos, cuando se está apuntando a un objetivo físico. Esta ley de fitts es utilizada para, por ejemplo, modelar el acto de apuntar con la mano o los dedos y, en el contexto de los ordenadores, para apuntar o dirigir mientras se hace uso de un mouse.

**Mapas de navegación:** son esquemas de organización de la información de una aplicación móvil, los cuales muestran las bases principales de la aplicación misma, expresan las relaciones de jerarquía y secuencia, permiten elaborar escenarios de comportamientos de los usuarios; son representados de forma textual, gráfica o ambas combinadas.

**“Mockup”:** en la fabricación y el diseño, el “mockup” es una maqueta a escala o de tamaño completo de un diseño o dispositivo, que se utiliza para la enseñanza, demostración, evaluación de diseño, promoción y otros fines.

**Multiplataforma:** indica la capacidad o características con las que cuentan algunas aplicaciones móviles o “software” de escritorio; permiten la interoperabilidad y ejecución sobre diferentes sistemas operativos o plataformas. Algunos lenguajes de programación que facilitan la creación de aplicaciones multiplataforma son Java, C++, JavaScript.

**Navegabilidad:** es la facilidad con la que un usuario puede desplazarse por todas las páginas o ventanas que componen un sitio web o una App. Para lograr este objetivo, el sitio web o la aplicación debe proporcionar un conjunto de recursos y estrategias de navegación diseñados para conseguir un resultado óptimo en la localización de la información y en la orientación para el usuario.

**Pixel:** es la abreviatura de “Picture Element” y puede definirse como la unidad más pequeña de color que componen una imagen digital, ya sea una fotografía, vídeo o un fotograma.

**Skeumorfismo:** su definición estricta es objeto evolucionado/derivado que mantiene detalles del diseño del objeto original con el objetivo de hacerlo más familiar. Viene de las palabras en griego “skeuos” (herramienta o contenedor) y “morphe” (“shape”).

Referencias bibliográficas

Cuello, J. y Vittone, J. (2013). Diseñando apps para móviles.

Desarrollador Android. (2015). Action Bar (Barra de Acciones). <https://desarrollador-android.com/diseno/pautas-2/action-bar-barra-de-acciones/>

Duckett, J. (2014). JavaScript & jQuery. Interactive front-end Web development. John Wiley & Sons.

Fernández Casado, Pablo E, 2019. Usabilidad web, teoría y uso. Bogotá, Madrid: Ediciones de la U, Ra-Ma Editorial.

Fernández, P. (2019). Usabilidad web, teoría y uso. RA-MA Editorial.

Gauchat, D. (2017). El gran libro de HTML5, CSC3 y JavaScript. Marcombo.

Gothelf, J., Seiden, J., & López Manzano, J. (2014). Lean UX. Logroño: UNIR Editorial.

Gualtetiero, T. y Salvaggio, A. (2019). JavaScript: Guía completa. Alfaomega.

Jacobson, I., Booch G., Rumbaugh, J.(2007). El lenguaje unificado de modelado: manual de referencia. Madrid: Pearson Educación

Kendall, K., Kendall, J., & Kendall, J. (2011). Análisis y diseño de sistemas. México, DF: Pearson Educación.

Kendall, K. y Kendall, J. (2011). Análisis y diseño de sistemas. Pearson Educación.

Klein, L. (2014). Experiencia de usuario para lean startups. Logroño: UNIR Editorial.

Klein, L. (2014). Experiencia de usuario para lean startups. UNIR Editorial.

Krug, S. (2015). No me hagas pensar, "actualización". Madrid: Anaya Multimedia.

Krug, S. (2015). No me hagas pensar. Actualización. Anaya Multimedia.

Luján Castillo, José Dimas, 2017. HTML5, CSS y Javascript. Crea tu web y apps con el estándar de desarrollo. Bogotá: Alfaomega Colombiana RC Libros.

Luján, J. (2017). HTML5, CSS y Javascript. Crea tu web y apps con el estándar de desarrollo. Alfaomega.

Norman, D. (2013). The design of everyday things. Basic Books.

Pohl, K. & Rupp, C. (2011). Requirements engineering fundamentals. Santa Barbara, CA: Rocky Nook.

Pohl, K. y Rupp, C. (2011). Requirements engineering fundamentals. Rocky Nook.

Pressman, R., Campos Olguín, V., & Enríquez Brito, J. (2010). Ingeniería del software. México: McGraw-Hill.

Pressman, R., Campos, V. y Enríquez, J. (2010). Ingeniería del software. McGraw-Hill.

Raffino, M. (2020). Interfaz - Qué es, concepto, tipos, características y ejemplos. Concepto.de. <https://concepto.de/interfaz/>

Recio García, Juan Antonio, 2018. HTML5, CSS3 y JQuery Curso práctico. Bogotá: Ediciones de la U.

Recio, J. (2018). HTML5, CSS3 y JQuery Curso práctico. Ediciones de la U.

Seydnejad, S. (2016). Modular Programming with JavaScript. Packt Publishing.

Sommerville I. (2011). Ingeniería del software. México: Addison-Wesley

Sommerville I. (2011). Ingeniería del software. Addison-Wesley

Whitten, J., Bentley, L., Randolph, G., Rico, M. y Orozco, M. (2008). Análisis de sistemas. McGraw - Hill Interamericana.

Créditos

| Nombre | Cargo | Regional y Centro de Formación |
| --- | --- | --- |
| Claudia Patricia Aristizabal | Responsable del Ecosistema | Dirección General |
| Rafael Neftalí Lizcano Reyes | Responsable de Línea de Producción | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Henry Eduardo Bastidas Paruma | Experto temático | Centro de teleinformática y producción industrial - Regional Cauca |
| Jonathan Guerrero Astaiza | Experto temático | Centro de teleinformática y producción industrial - Regional Cauca |
| Peter Emerson Pinchao Solis | Experto temático | Centro de teleinformática y producción industrial - Regional Cauca |
| Zulema Yidney León Escobar | Experta temática | Centro de teleinformática y producción industrial - Regional Cauca |
| Oscar Absalón Guevara | Diseñador instruccional | Centro de gestión industrial - Regional Bogotá |
| Alix Cecilia Chinchilla Rueda | Diseñadora instruccional | Centro de gestión industrial - Regional Bogotá |
| José Gabriel Ortiz Abella | Evaluador instruccional | Centro para la industria y la comunicación gráfica - Regional Distrito Capital |
| Julia Isabel Roberto | Evaluadora instruccional | Centro para la industria y la comunicación gráfica - Regional Distrito Capital |
| Ana Catalina Córdoba Sus | Metodólogo para formación virtual | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Carmen Alicia Martínez Torres | Animador y Productor Multimedia | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Wilson Andrés Arenales Cáceres | Storyboard e ilustración | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Camilo Andrés Bolaño Rey | Locución | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Yerson Fabian Zarate Saavedra | Diseñador de Contenidos Digitales | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Francisco José Lizcano Reyes | Desarrollador Full-stack | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Daniel Ricardo Mutis Gómez | Evaluador para Contenidos Inclusivos y Accesibles | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Zuleidy María Ruíz Torres | Validador de Recursos Educativos Digitales | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Luis Gabriel Urueta Álvarez | Validador de Recursos Educativos Digitales | Centro Industrial del Diseño y la Manufactura - Regional Santander |