

# Desarrollando y diseñando aplicaciones móviles con Flutter

Decena, Facundo – Maguire, Margarita – Merenda, Franco –  
Pellegrino, Maximiliano

InforSanLuis 2019

# Layout y diseño de UI

# Layouts

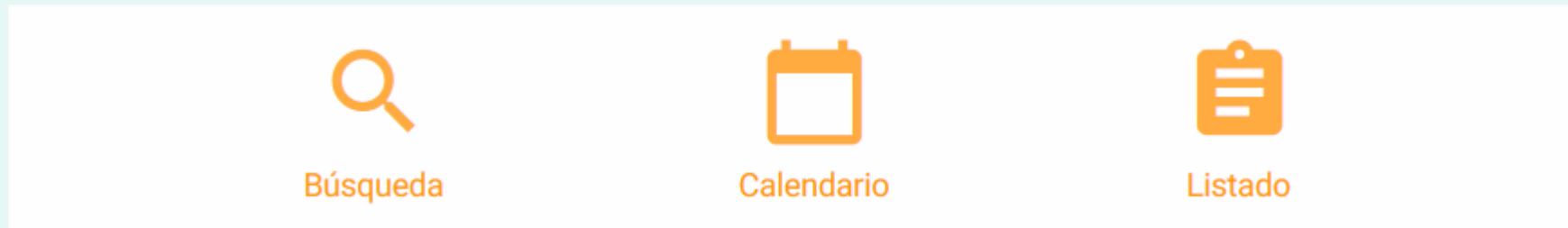
Casi todo en Flutter es un widget. Imágenes, iconos, texto, botones, todo lo que se ve en una app es un widget. Incluso las cosas que no podemos ver son widgets, como las filas, columnas, grillas, alineaciones, etc.

Los layouts se crean componiendo widgets para crear widgets más complejos.

Veamos un ejemplo.

# Layouts

Tenemos un ejemplo de una posible bottom bar para una aplicación. Parece complicada, pero probemos desagregándola en partes más pequeñas.



# Layouts

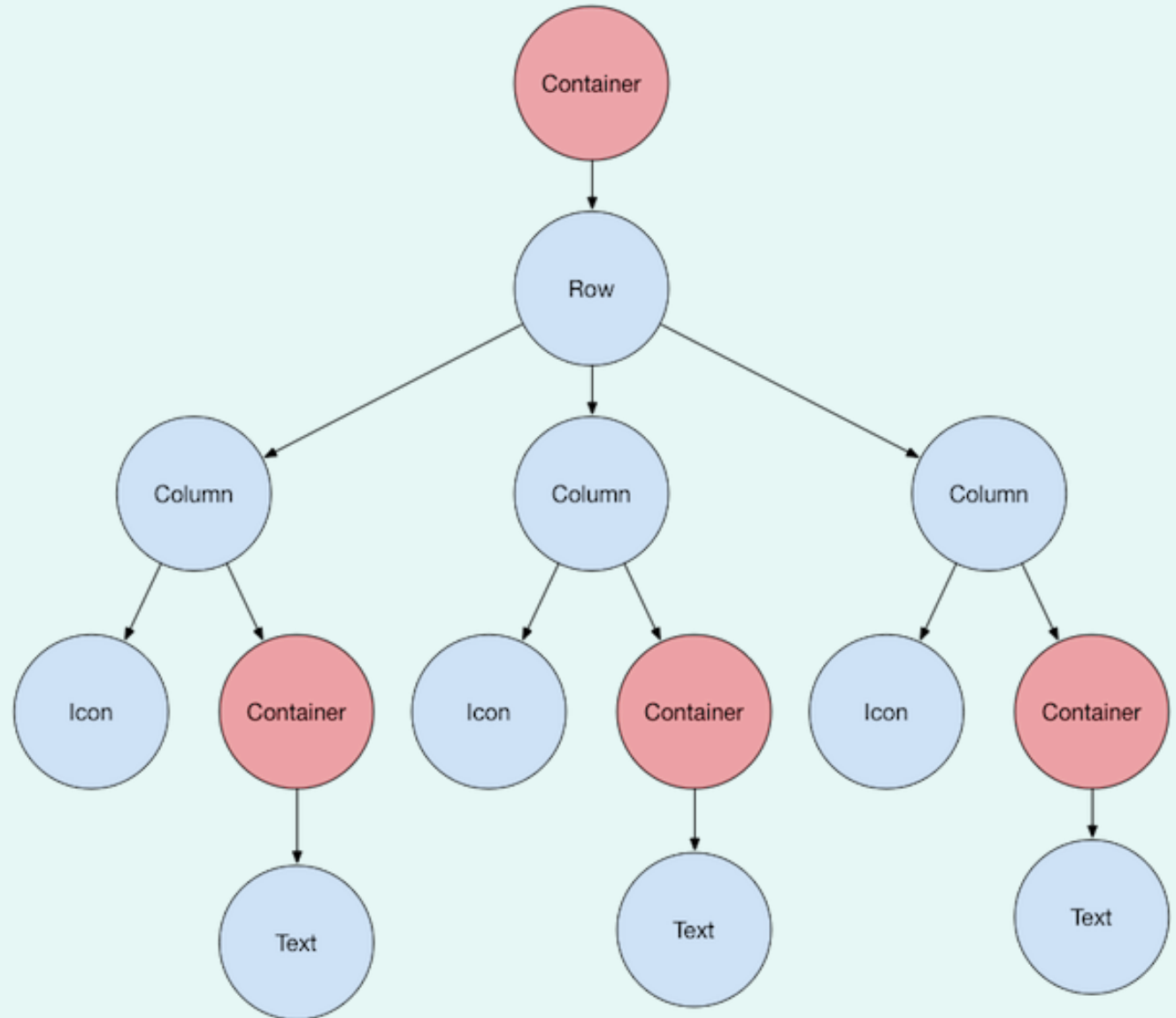
Tenemos un ejemplo de una posible bottom bar para una aplicación. Parece complicada, pero probemos desagregándola en partes más pequeñas.



# Layouts

Diagrama del árbol de Widgets para la bottom bar anterior

A lo que vimos se agregan los Container que nos ayudan a customizar el widget hijo del mismo (márgenes, fondo, etc).



# Layouts

Podemos dividir estos widgets en dos categorías:

- **Widgets visibles:** son los que efectivamente nos muestran información en la interfaz. Imágenes, íconos, texto, etc.
- **Widgets de layout o diseño:** son los que nos ayudan a alinear, delimitar, organizar los demás widgets. Además poseen como atributo **child** o **children**, que les permiten anidarles Widgets.
  - **child:** Container, Center, Padding, etc.
  - **children:** Column, Row, Stack, ListView, etc.

# Layouts

La estructura básica de una aplicación que vamos a utilizar es la del Scaffold, que nos proporciona una barra superior, barra inferior, cuerpo de la página, etc, como vimos anteriormente. Si no se quiere seguir la estructura de Material se puede optar por otro widget de layout, como por ejemplo un Container.

Dentro del cuerpo de nuestra aplicación vamos a organizar los elementos que queremos mostrar, de manera vertical y horizontal como acabamos de ver en el ejemplo de la bottom bar. Para eso utilizaremos Column y Row.



# Layouts

- Column y Row son los dos patrones de layout más utilizados.
- Ambos utilizan una lista de widgets hijos para mostrar.
- A su vez, cada hijo puede ser un Row, un Column, o cualquier otro Widget.
- Podemos especificarles como alinear los hijos, vertical y horizontalmente.
- Los widgets hijos se pueden estirar o contraer.
- Se puede especificar como estos widgets hijos usan el espacio.

# Layouts

Row

2 children

child: new Column

child: new Image

Column

4 children

## Strawberry Pavlova

Pavlova is a meringue-based dessert named after the Russian ballerine Anna Pavlova. Pavlova features a crisp crust and soft, light inside, topped with fruit and whipped cream.



170 Reviews



PREP:

25 min



COOK:

1 hr

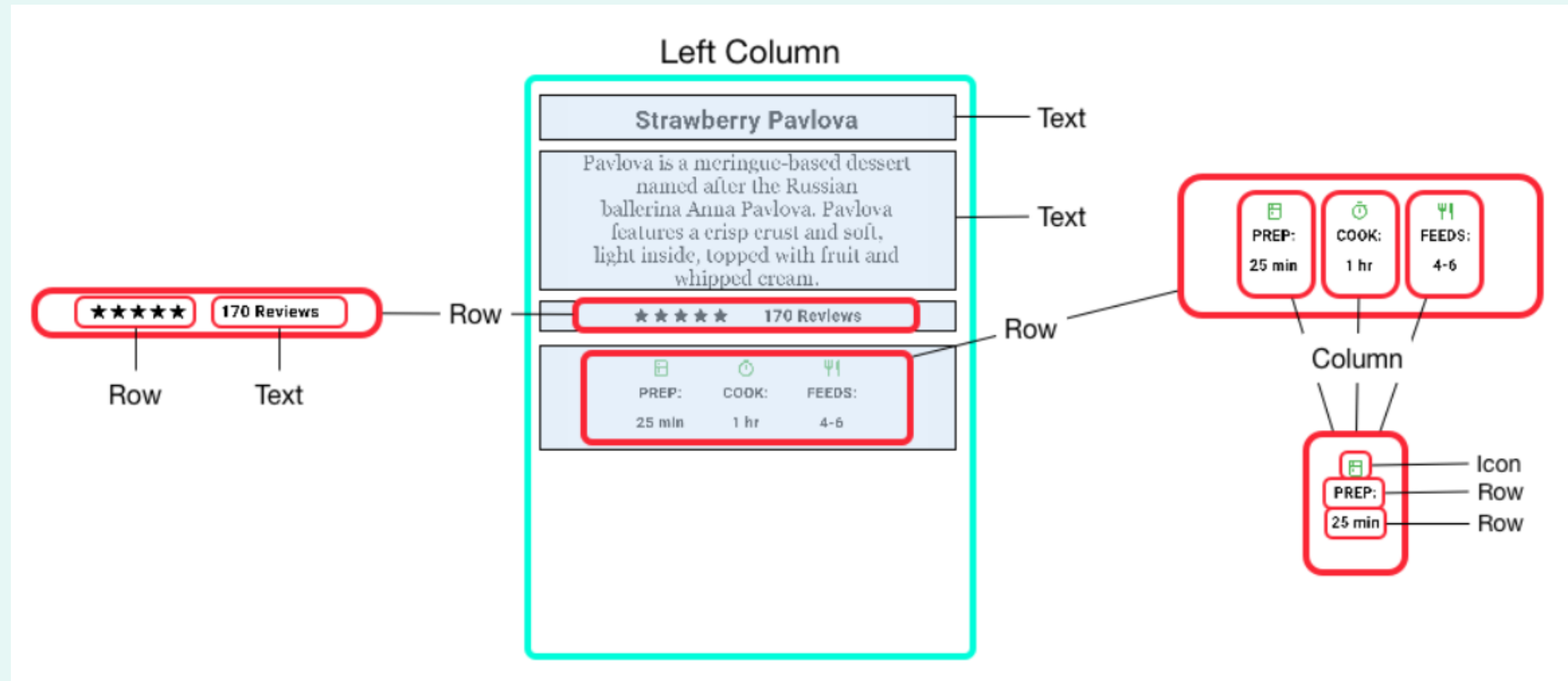


FEEDS:

4-6



# Layouts



# Layouts

Complementándose con Column y Row tenemos los widgets **Expanded** y **Flexible**, muy útiles para organizar el espacio y en el desarrollo de responsive UI.

- **Expanded:** Hace que el widget ocupe todo el espacio que tenga disponible.
- **Flexible:** No fuerza al widget a ocupar todo el espacio disponible, sino que le permite ocupar el tamaño máximo que desee.

# Layouts

## Layouts in Flutter

- <https://flutter.dev/docs/development/ui/layout>

## Building Layouts Tutorial

- <https://flutter.dev/docs/development/ui/layout/tutorial>

## Adding assets and images

- <https://flutter.dev/docs/development/ui/assets-and-images>

# Vocabulario de Diseño

# ¿Qué es un Vocabulario de Diseño?

Un vocabulario de diseño es un esquema o estilo que guía el diseño de diferentes productos. Se utilizan cuando se quiere que una suite de productos tenga un look and feel consistente y único.

Describen las paletas de colores, formas, texturas, materiales, patrones, layouts, etc. Luego, esto se utiliza en los diferentes objetos de la suite.

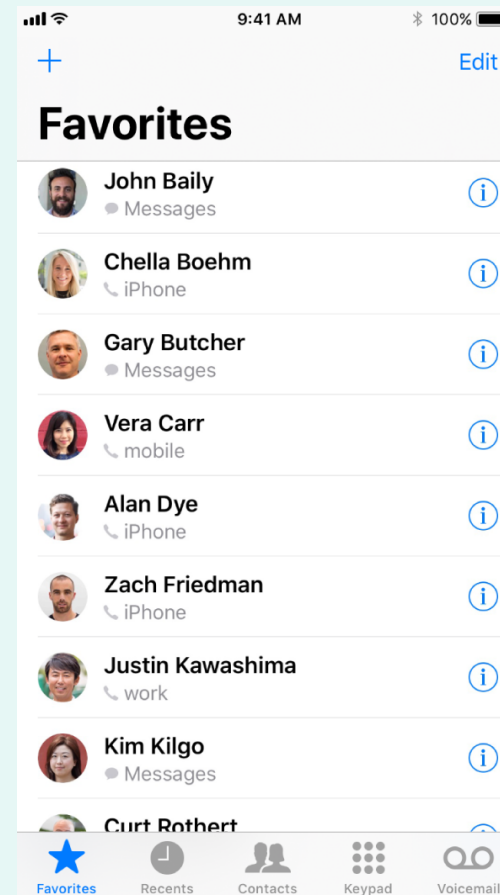
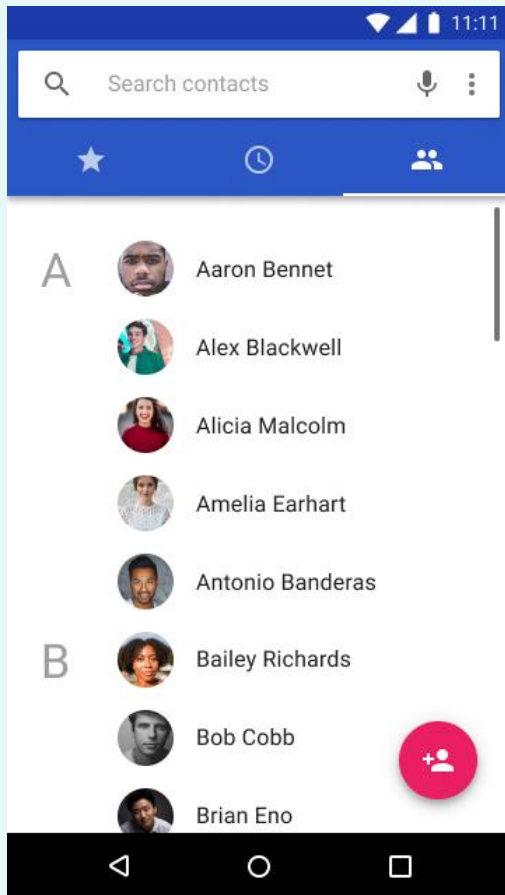
En el contexto de las GUI, las guías de diseño de interfaces hombre-máquina pueden ser vistas como un vocabulario de diseño para aplicaciones.

# Material vs iOS

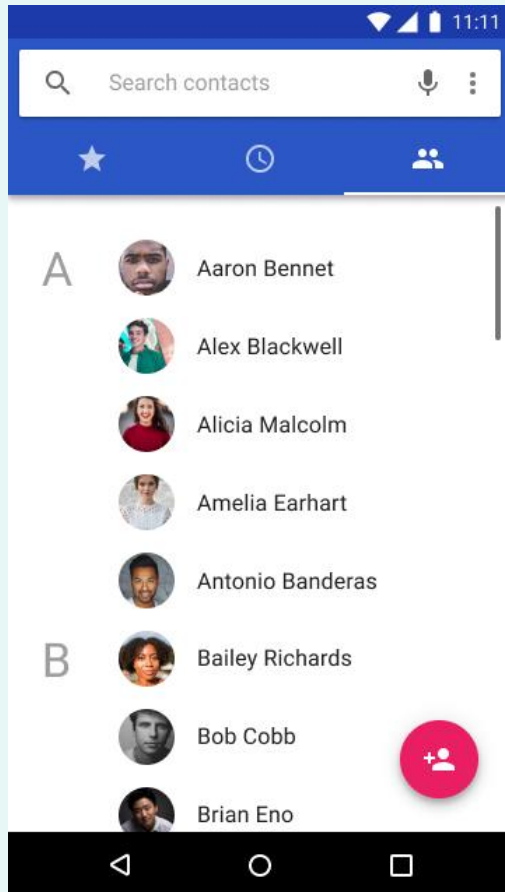
- **Material Design:** Es el estilo que utiliza Google para sus productos, además de ser el estilo que llevan la mayoría de aplicaciones para Android. Está inspirado en materiales de la vida real, como el papel, y en como se superponen, doblan, manipulan, etc.
- **Human Interface (iOS):** Es el estilo diseñado por Apple para sus diferentes productos. Está diferenciado según la plataforma (iOS, macOS, tvOS, etc). Busca ser claro, legible, enfocado en las funcionalidades, más minimalista y plano.



# Material vs iOS



# Material vs iOS

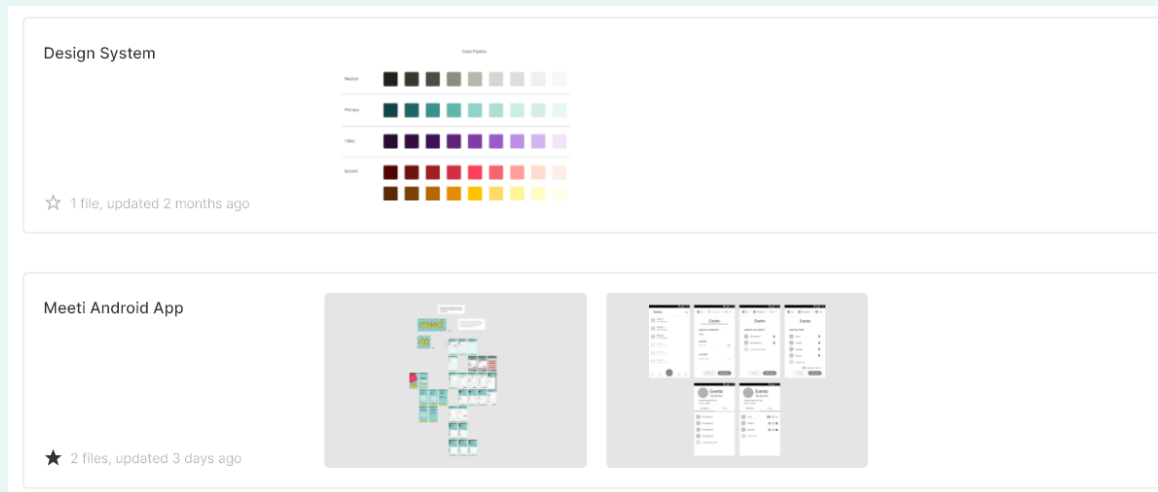


Este va a ser el estilo que vamos a usar nosotros. Todas las componentes, paletas de color, íconos, etc, van a estar en el paquete 'flutter/material.dart'. Importándolo en nuestro proyecto tendremos todo lo necesario para desarrollar una aplicación que se vea acorde al sistema operativo Android.

# Diseño de UI con Figma

# ¿Qué es Figma?

Figma es una herramienta de diseño colaborativo, que trabaja enteramente online y en el navegador. Además tiene versión de escritorio para Windows. Se pueden asignar roles en los proyectos (editar, solo ver, etc), tener organizadas diferentes carpetas, ver cambios en tiempo real y más.



# Empezando a diseñar

Empezar de cero a prototipar una aplicación es terriblemente difícil, por eso vamos a trabajar con archivos de base para Material, jugando con las componentes, colores, elevaciones, etc.

La propuesta es que puedan diseñar una o dos pantallas para una aplicación, sea una idea suya o inspirados en ejemplos que les vamos a dar a continuación.

# Empezando a diseñar

- **Idea 1:** Se quiere armar un formulario de registro para una aplicación. Constará de una variedad de campos a rellenar, y que una vez terminada la carga de los mismos, que se muestren en una página de usuario.
- **Idea 2:** Se quiere diseñar una página para compra de insumos para la pc. Se precisa un catálogo de productos y un carrito de compra para conocer lo que lleva elegido hasta el momento.

# Empezando a diseñar

- **Idea 3:** Se quiere diseñar una página para compra de insumos para la pc. Se necesita detallar los diferentes pasos de pago de una compra, como detallar el método de pago, envío de ser necesario, y los pasos que se crean necesarios

¡Sean creativos! Pueden basarse en alguna de las ideas, inspirarse con diseños en internet, o incluso repensar alguna idea propia que tengan por ahí.

# Empezando a diseñar

Algunas consideraciones básicas a la hora de plantear nuestro diseño:

- Elegir una **paleta de color** que exprese lo que queremos transmitir con la app (Seguridad, simpatía, simplicidad, etc).
- Tipografías **legibles**, pensadas para leer mucho texto.
- Pensar en la **información** que queremos mostrar en las pantallas y como la queremos distribuir.
- Ponerse del lado del **usuario** que va a utilizarla, visualizar el público al que va destinada.
- **Menos** siempre **es más**, ¡no sobrecarguemos de información!



# Herramientas útiles

Guía de Material Design

- <https://material.io/design/>

Recursos de Material Design

- <https://material.io/resources/>

Color Tool, para jugar con los colores de una Material App

- <https://material.io/resources/color/>

Estudios de aplicaciones, para inspirarse

- <https://material.io/design/material-studies/>

# Herramientas útiles

Color Wheel de Adobe

- <https://color.adobe.com/es/create>

Google Fonts

- <https://fonts.google.com/>

FontPair, ayuda a elegir un par de tipografías que combinen

- <https://fontpair.co/>

Pinterest, para buscar inspiración

- <https://www.pinterest.es/>