



# **FUN**damentals & Deep Dive

FAQ,

Tips and Tricks

Why Flutter? ... wait what's Flutter?...

**FUN**damentals and Hands on...



# Anilson Monteiro

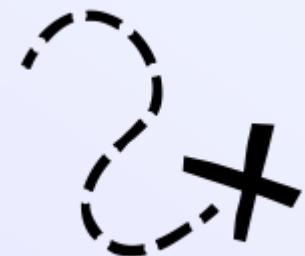
From Telecommunications Degree to  
**Full Stack Developer** by <Academia de Código>  
**Cloud Enthusiast**   
**JS Enthusiast**   
**Lifelong Learner**



# What we'll do today?

## Session 1 → Fundamentals

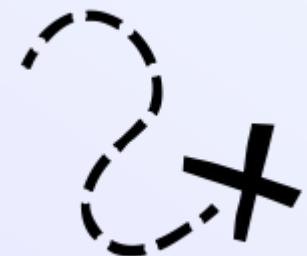
1. Flutter FAQ
2. Dart and OOP Programming - A Quick Tour
3. JS to Dart - A Quick Tour
4. Setup and Installation & Intro to Flutter Development
5. Flutter: The Basics - Recap



# What we'll do today?

## **Session 2 → Deep Dive**

6. Hands On & Mini Projects
7. Deployment
8. What else?
9. A little Challenge



# Requirements

- Programing Basics
- **Developer Spirit ;)**

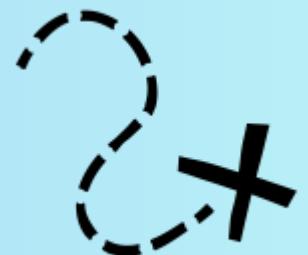
# FUNdamentals

Let's have fun...!

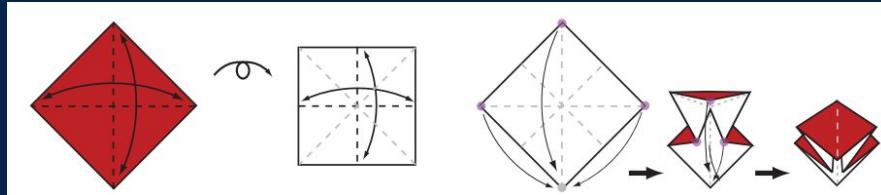


# Session 1 → Fundamentals

1. Flutter FAQ
2. Dart and OOP Programming - A Quick Tour
3. JS to Dart - A Quick Tour
4. Setup and Installation & Intro to Flutter Development
5. Flutter: The Basics - Recap



“Mastering the fundamentals  
is **challenging**, requires  
**practice**, but can be **fun**.”

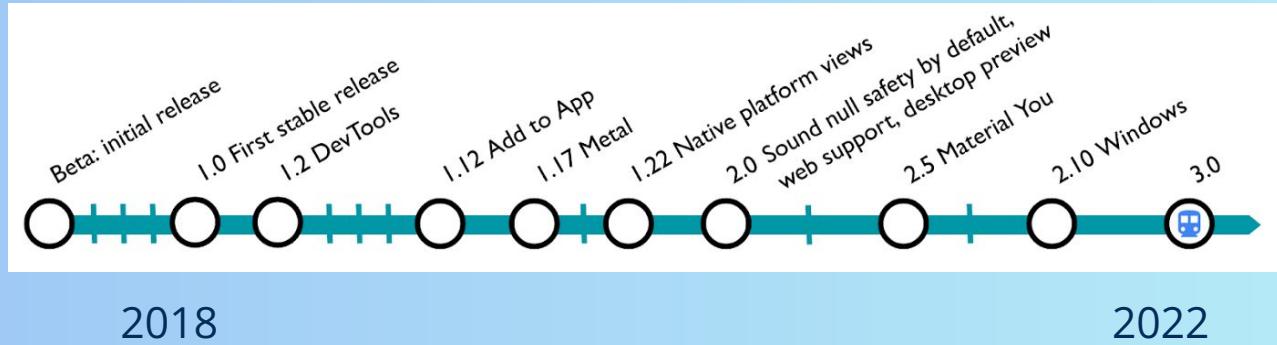


#1

# Flutter FAQ

Why Flutter? Wait...What's  
Flutter? Is it that? Who build  
Flutter? What can i do with it?  
Why Dart?

# Some news about Flutter



# Some news about Flutter



The image features three stylized blue bird characters with white bellies and brown feet. One bird is standing on the left, another is stacked in the middle, and a third is perched on top of the middle one, holding a small red cube. To the right of the birds is a large, three-dimensional number '3' composed of various colored blocks (blue, yellow, red, purple) of different sizes and shapes, some with small icons like a plus sign or a gear. The background is plain white.

Google I/O

- Today there are over 500,000 apps built with Flutter.
- Ready for production on **all desktop platforms**
- **Linux** and **macOS** have reached **stable**

Collaboration

Mac and  
Cross Platform  
(Android, iOS)

Interactivity

Native features

Customizations

# What is Flutter?

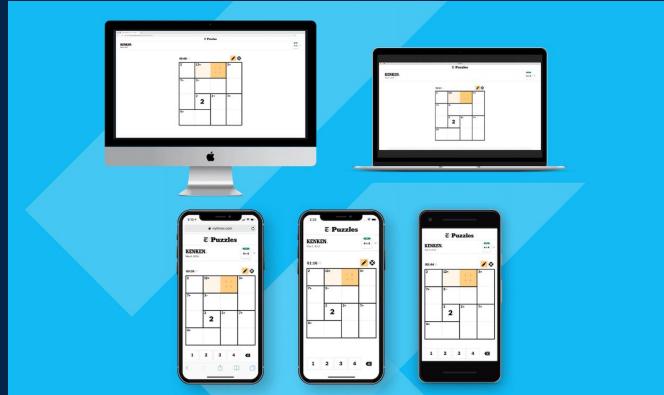
- Flutter is Google's portable UI toolkit for crafting beautiful, natively compiled applications for mobile, web, and desktop from a single codebase.



# What kinds of apps can I build with Flutter?

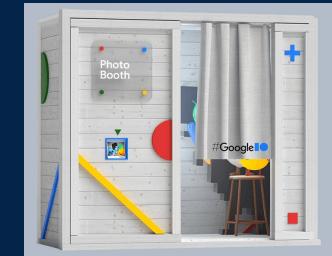
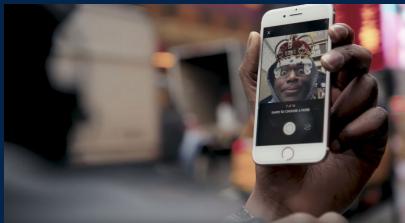


Hamilton Musical event App



New York Times puzzles

"Rather than re-writing the puzzle app for every different platform, they can just write it once with Flutter."



I/O Photo Booth for Google I/O 2021

# How much development experience do I need to use Flutter?

- “Flutter is approachable to programmers familiar with **object-oriented** concepts (classes, methods, variables, etc) and imperative programming concepts (loops, conditionals, etc).
- We have seen people with very **little programming experience** learn and use Flutter for **prototyping** and app development”.

# Who makes Flutter?

Flutter is an **open source** project, with contributions from **Google** and other companies and **individuals**.

**<https://github.com/flutter/flutter>**

# How does Flutter run my code on Android?

- The engine's C and C++ code are compiled with Android's NDK. The Dart code (both the SDK's and yours) are **ahead-of-time (AOT) compiled into native**, ARM, and x86 libraries.
- During **debug mode**, Flutter uses a **virtual machine (VM)** to run its code in order to enable stateful **hot reload**, a feature that lets you make changes to your running code **without recompilation**.

# Concepts

- **AOT (Ahead of Time) Compilation**
  - ahead-of-time compilation(AOT compilation) is the act of compiling an often **higher-level programming language** into an often **lower-level language** before execution of a program, usually at build-time, to reduce the amount of work needed to be performed at run time
- **Compilation vs Transpilation** (Source-to-source compiler or a source-to-source translator)
  - a type of **translator** that takes the **source code** of a program written in a programming language as its input and produces an equivalent source code in the same or a **different programming language**

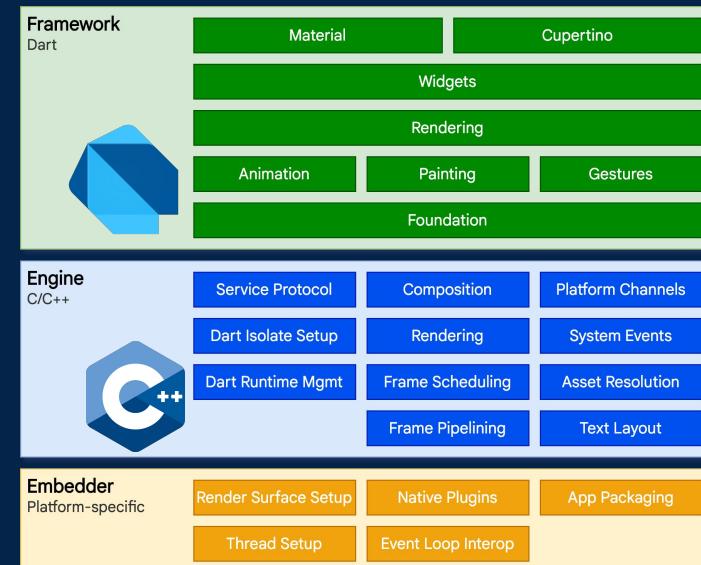
# Does Flutter run on the web?

- You can **compile** existing Flutter code to work on the web.
- A release build uses **dart2js** to produce a **single JavaScript file** main.dart.js.



# What language is Flutter written in?

- **Dart**, a fast-growing modern language optimized for client apps. The underlying graphics framework and the Dart virtual machine are implemented in **C/C++**
- Flutter is designed as an extensible, **layered** system.



<https://docs.flutter.dev/resources/architectural-overview>

# Where is Flutter's markup language? Why doesn't Flutter have a markup syntax?

- Flutter UIs are built with an **imperative**, object-oriented language (Dart, the same language used to build Flutter's framework). Flutter **doesn't** ship with a **declarative markup**.
- **UIs dynamically built with code** allow for more **flexibility**. it's difficult for a **rigid markup system** to express and produce **customized** widgets with bespoke **behaviors**.

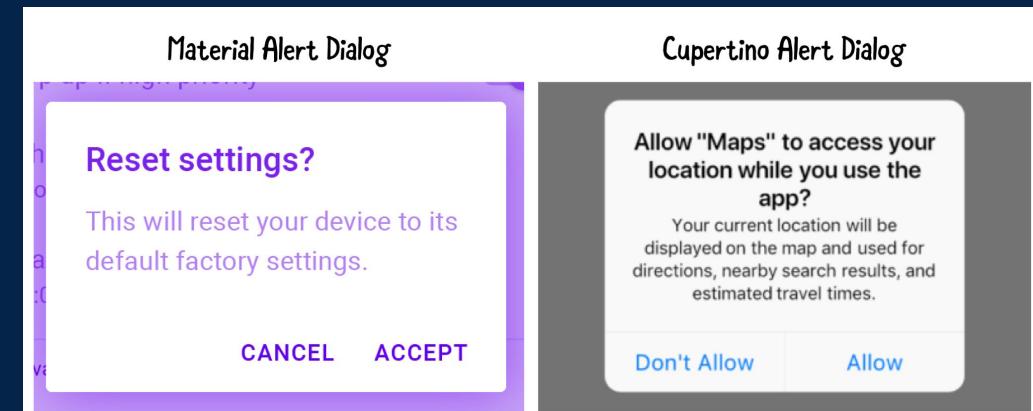
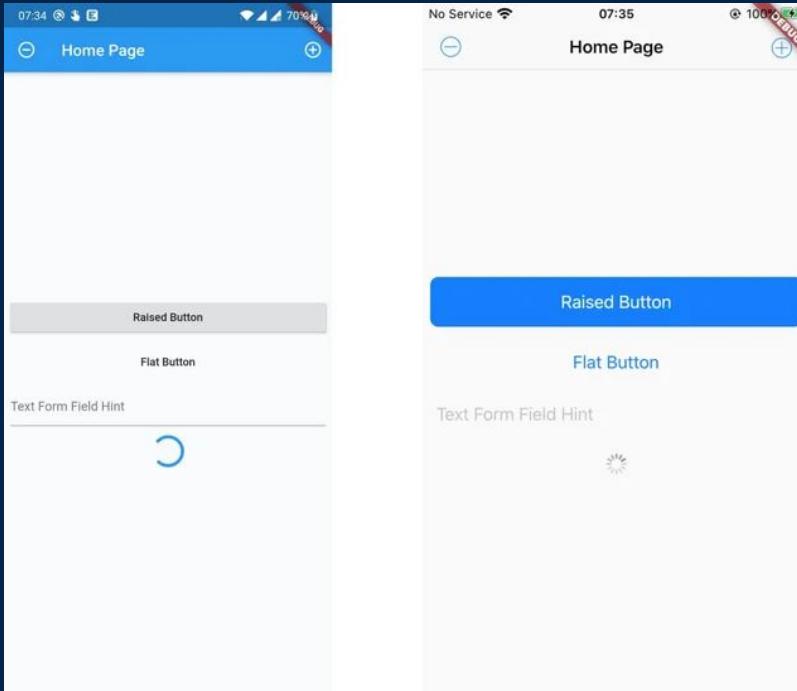


# Why would I want to share layout code across iOS and Android?

Can I extend and customize the bundled widgets?

- More and more, we see mobile app layouts and designs evolving to be more **brand-driven** and **unified across platforms**. This implies a strong motivation to **share layout and UI** code across iOS and Android.
- The **brand identity** and customization of the app's aesthetic design is now becoming **more important** than strictly adhering to **traditional platform aesthetics**.

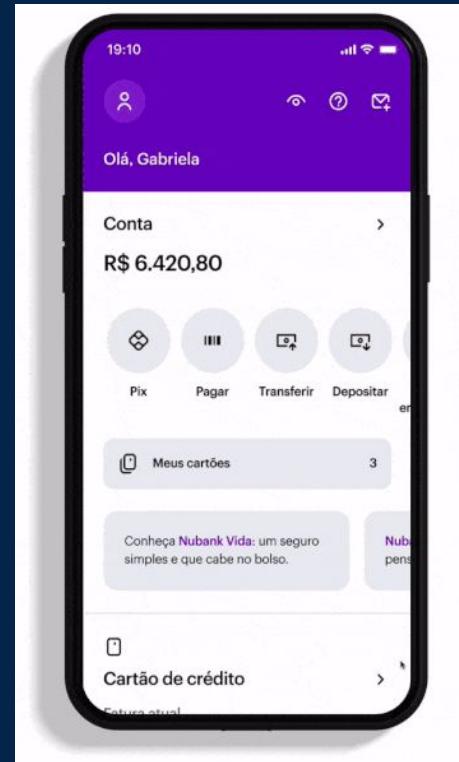
# Composition, Aesthetic



# Composition, Aesthetic

you can also create  
**pixel-perfect**  
experiences that match the  
Android and iOS design  
languages with Flutter.

For **designers** Flutter's  
ability to turn **concepts**  
**into production code**  
without the compromises  
imposed by typical  
frameworks



“...our own design system, in which product teams can extend and customize their UI components.

“

<https://building.nubank.com.br/scaling-with-flutter/>

# #2

## Dart and OOP Programming

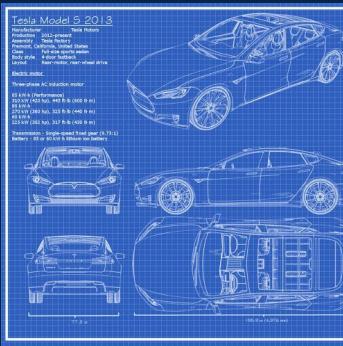
### - A Quick Tour

Dart is all about OOP...

if in Java everything is objects, in Flutter  
everything is widgets



# The principles of OOP



- Class

4 PILLARS

Encapsulation

Abstraction

Inheritance

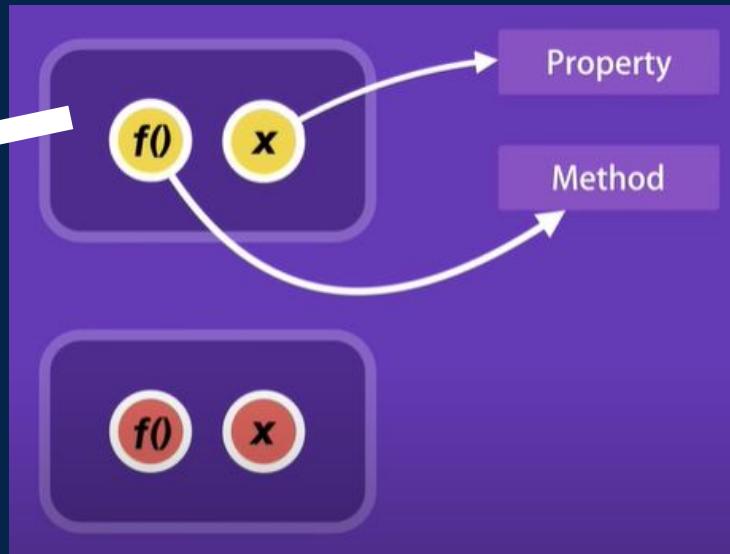
Polymorphism



- Object

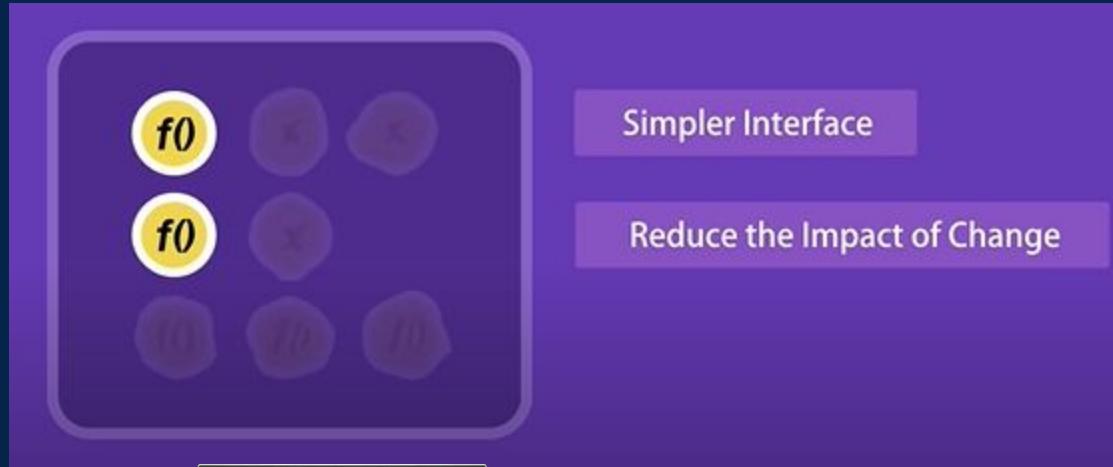
# Object Oriented Programming

- Programming paradigms are **different ways** or styles in which a given **program** or **programming language** can be **organized**.
- **objects**



# Object Oriented Programming

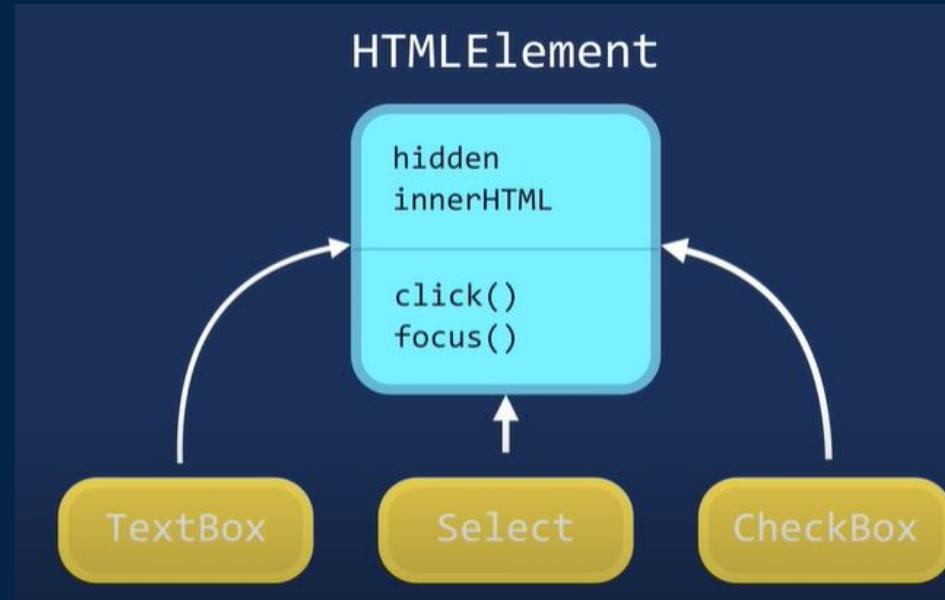
## - Abstraction, Encapsulation...



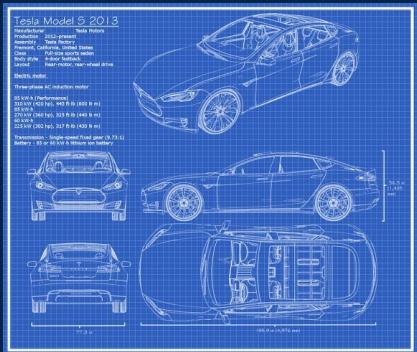
# Object Oriented Programming

## - Inheritance

- **IS-A Relationship**
- **Reduce redundant code**



# How do we start... with class



```
class Car{  
...  
}
```

`constructor()`



`let myCar = new Car()`



#3

## JS to Dart - A Quick Tour

Coming from JS? Let's see that Dart is not  
that monster after all



# Dart for JavaScript Developers

- Lab walkthrough
  - [Entry point](#)
  - Printing to the console
  - Variables
  - Default value
  - Functions



## Entry point

JavaScript doesn't have a pre-defined entry function—you define the entry point.

```
// JavaScript
function startHere() {
  // Can be used as entry point
}
```

In Dart, every app must have a top-level `main()` function that serves as the entry point to the app.

```
// Dart
void main() {}
```

Try it out in [DartPad](#).

#4

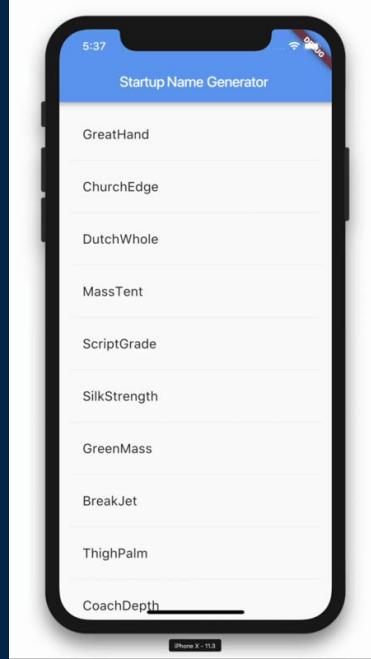
# Setup and Installation & Intro to Flutter Development

How to setup the environment for Flutter  
Development



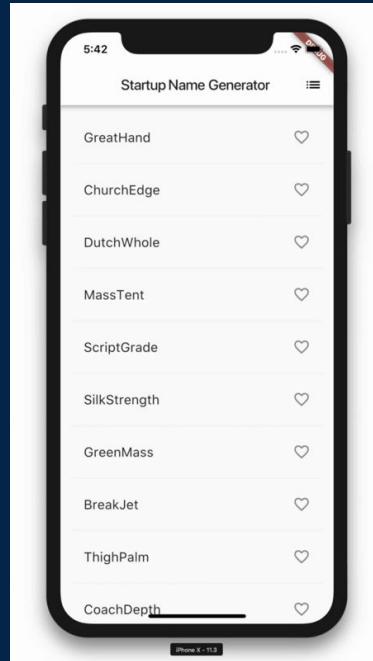
# Hello World - Part 1

- Lab walkthrough
  - Set up your Flutter environment
  - Create the starter Flutter app
  - Use an external package
  - Add a stateful widget
  - Create an infinite scrolling ListView



# Hello World - Part 2

- Lab walkthrough
  - Add icons to the list
  - Add interactivity
  - Navigate to a new screen
  - Change the UI using themes



# #5 RECAP

What did you just learn?



# Similarities of Web Development and Flutter development

- Lab Walkthrough
- Performing basic layout operations
  - Styling and aligning text
  - Setting background color
  - Centering components
  - Setting container width
- Manipulating position and size
  - Setting absolute position
  - Rotating components
  - Scaling components
  - Applying a linear gradient

- The HTML document starts with `<!DOCTYPE html>`, and the CSS box model is used for consistency with the Flutter model.

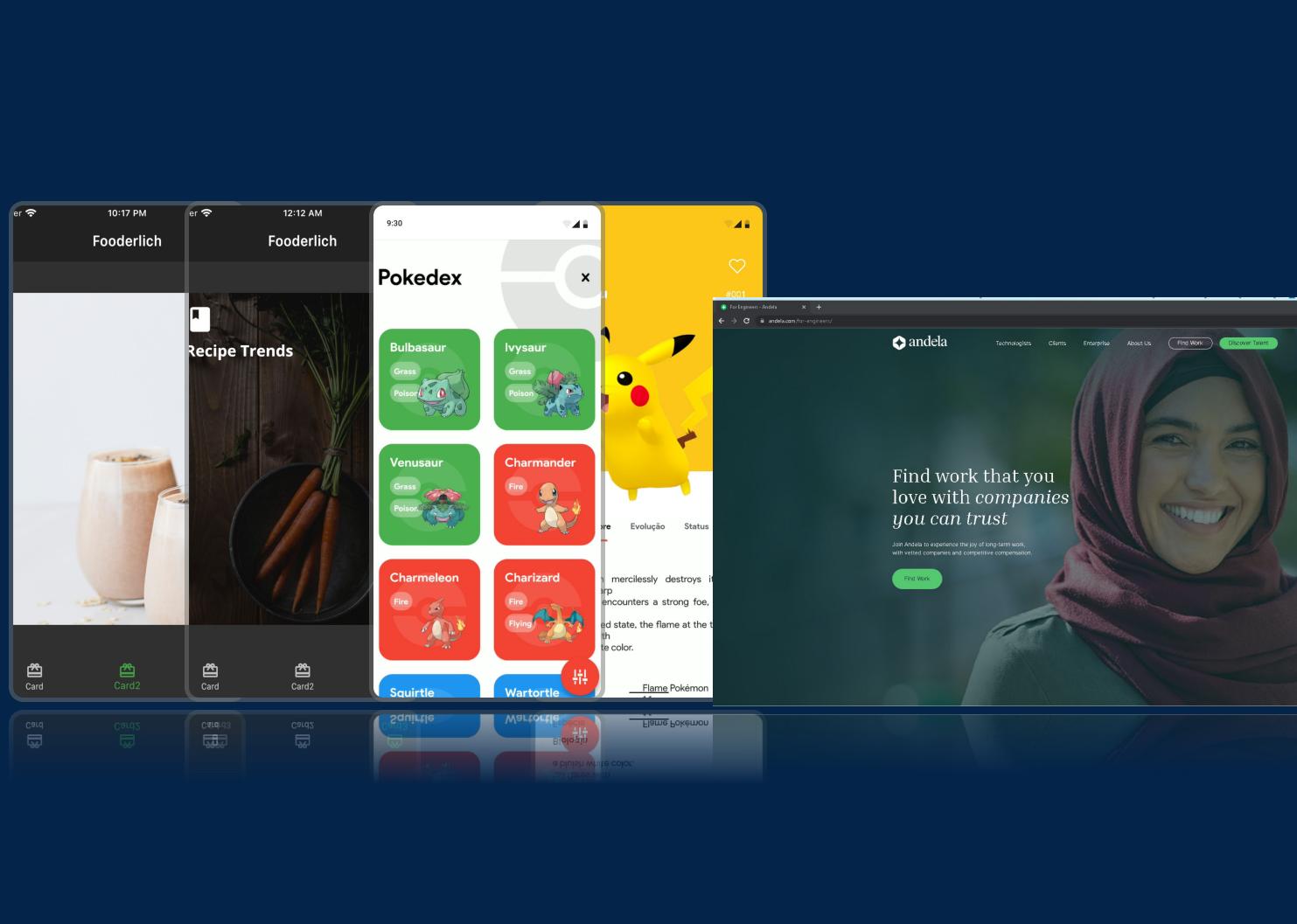
```
{  
  box-sizing: border-box;  
}
```

- In Flutter, the default styling of the 'Lorem ipsum' text is defined by the syntax simple:

```
TextStyle bold24Roboto = TextStyle(  
  color: Colors.white,  
  fontSize: 24,  
  fontWeight: FontWeight.bold,  
);
```

Coming next...

# Deep dive



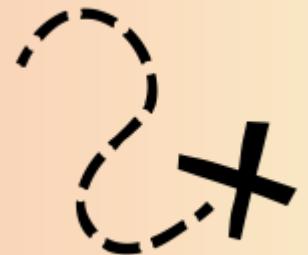
# Deep dive

Navigating in deep waters...!



# Session 2 → Deep Dive

- 6. Hands On & Mini Projects
- 7. Deployment
- 8. What else?
- 9. A little Challenge

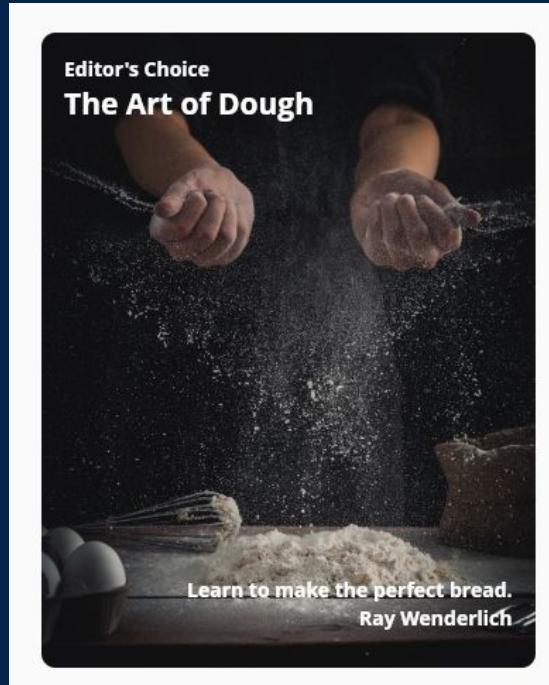


#6

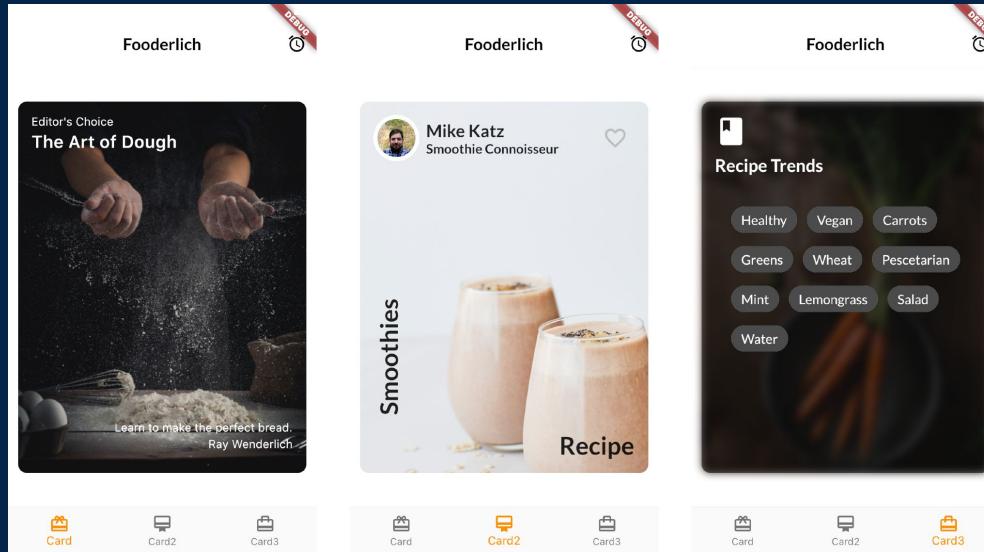
## Hands on Projects

Putting it all together and building  
something cool

# Project #1 - Fooderlich



# Challenge#1 - Fooderlich



# Recap: Dart for JavaScript Developers

- Lab walkthrough
  - Asynchronous programming



## Asynchronous programming

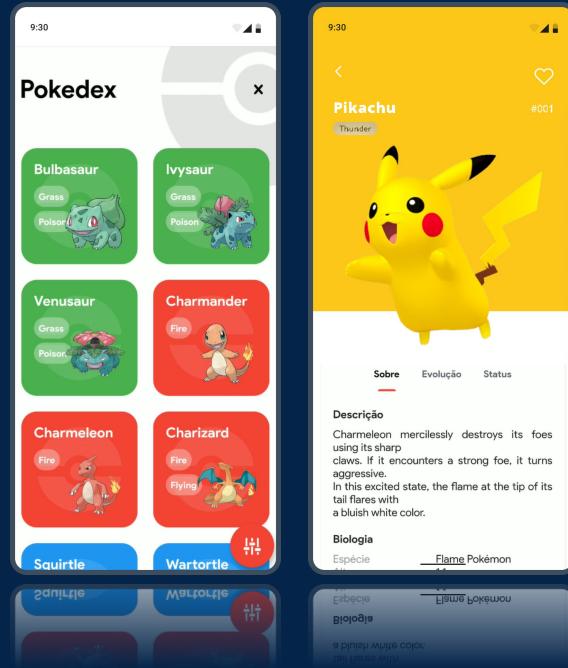
### Futures

Like JavaScript, Dart supports single-threaded execution. In JavaScript, the Promise object represents the eventual completion (or failure) of an asynchronous operation and its resulting value.

Dart uses `Future` objects to handle this.

```
// JavaScript
class Example {
  _getIPAddress() {
    const url = 'https://httpbin.org/ip';
    return fetch(url)
      .then(response => response.json())
      .then(responseJson => {
        const ip = responseJson.origin;
        return ip;
      });
  }
}
```

# Project #2 - Pokédex

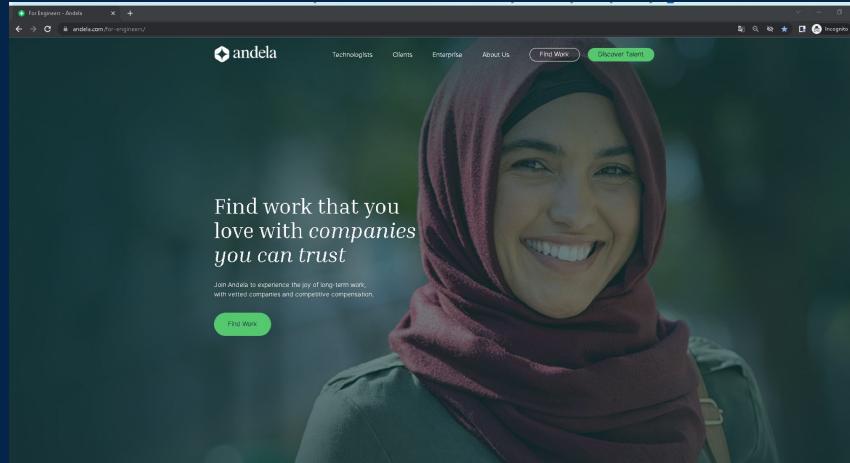


#8

## What else is out there?

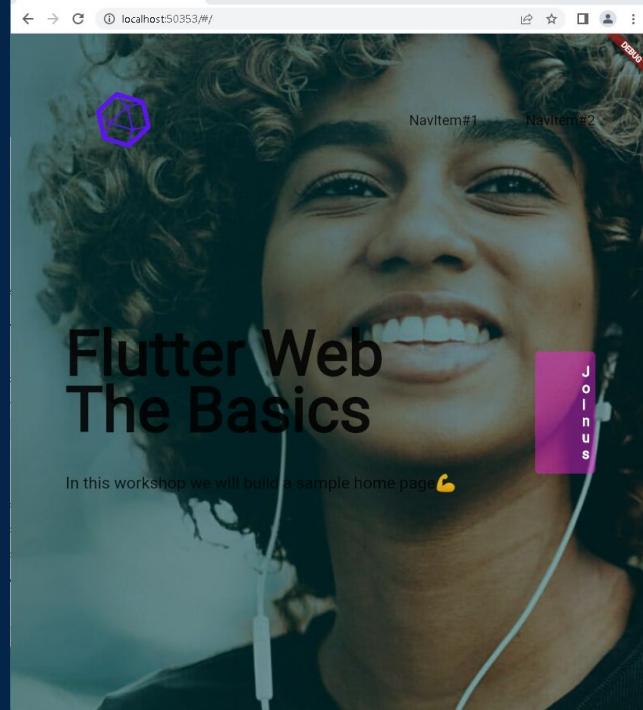
There is more to do on the web

# Project #3 - Andela Web Clone



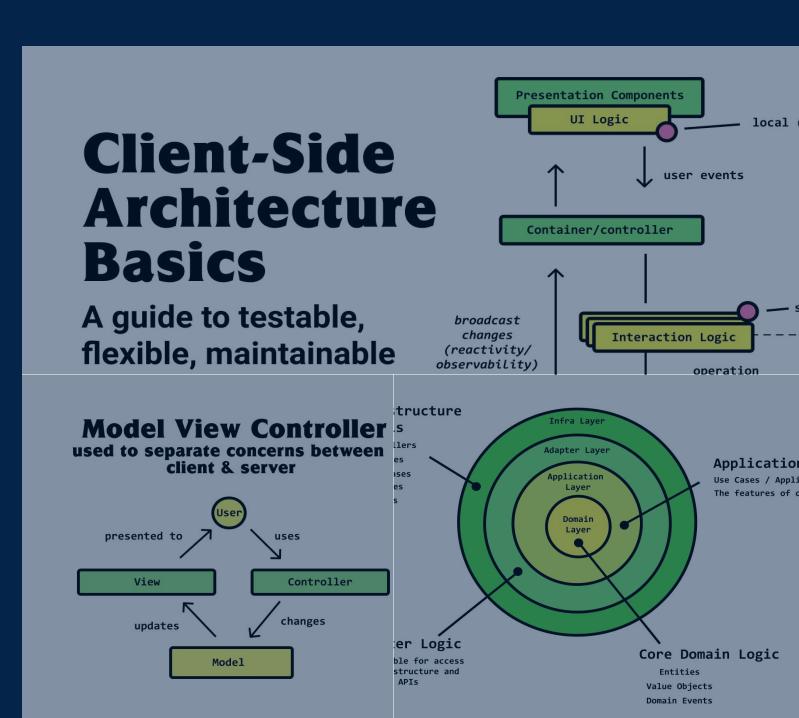
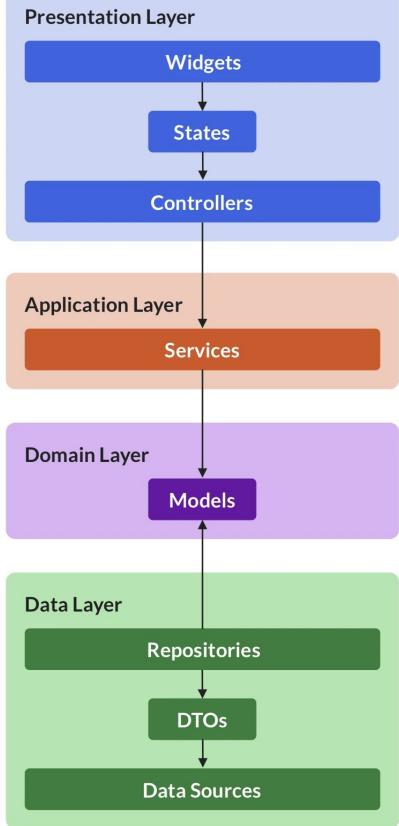
<https://andela.com/for-engineers/>

# Challenge#2 - How to make it Responsive



# Project Structure and Architecture

## - A quick note



- Client-Side Architecture
- SOLID Principles
- Clean Code
- Clean Architecture
- Refactoring

On the next episodes...

#9

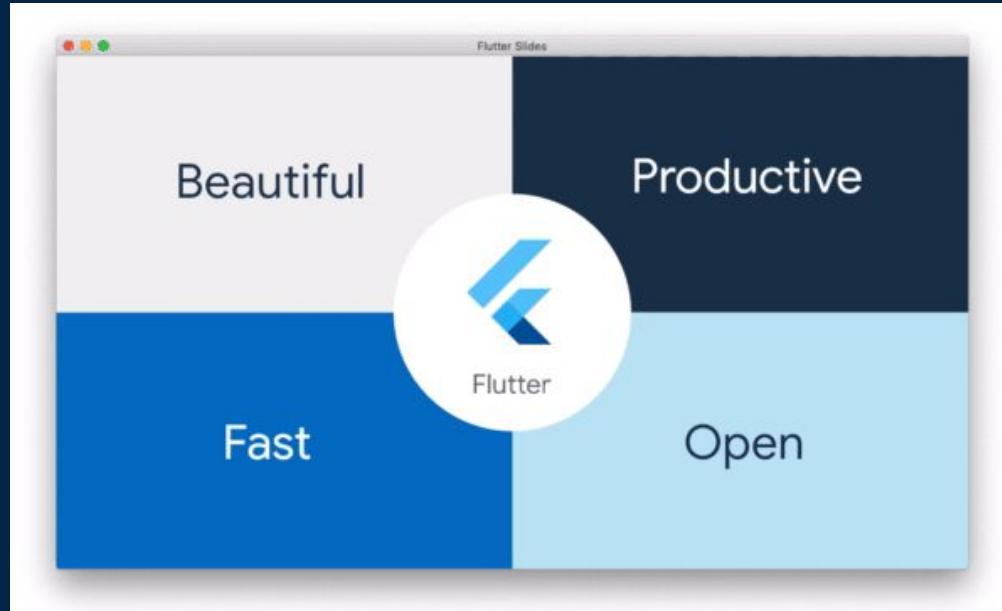
## A little Challenge

We saw how to develop for mobile and Web  
Now is your turn to share with us something  
from the desktop... yes it's possible too!



# Flutter Slides - Demo

- Flutter Desktop demo



#10

The end

Well, not yet...

# A quick look into the community

- Things we can find on the Flutter community
  - 
  - The Boring Flutter Show
  - The widget of the week
  - Animations - Community events
  - Book club - Community events
  - #30daysofcode
  - Puzzle Hackathon

# A quick look into the community

What else can i do with flutter??



<https://puzzle.renefloor.nl/>

<https://flutter-rush.web.app/>

<https://fph-planets.web.app/#/>

<https://devpost.com/software/beautiful-puzzle>

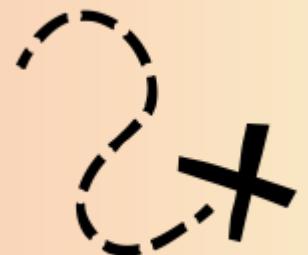
<https://devpost.com/software/flutter-rush-puzzle>

<https://devpost.com/software/planets-m6f8nx>

<https://flutterhack.devpost.com/>

# Final thoughts

- **Learn** the most you can
- Don't skip **steps**
- **Celebrate** the little victories
- Don't give up on **first bug**
- Look and **ask for help** (do your homework first)
- **Give back** and support the community (there is  
plenty of to do it)
- **Teach** what you learn (even if it's a hello world)
- Enjoy the **path** and
- have **FUN**damentals in mind
- Always look for the **next challenge**
- **Follow a community**
- **Learn together** (If you want to go **fast**, go **alone**. If  
you want to go **far**, go **together**)



Are you still  
here?



Don't loose  
next episodes...  
Thank you 



[linkedin.com/in/anilsonmonteiro](https://linkedin.com/in/anilsonmonteiro)



# Links and additional resources



[github.com/kotxiposix/workshop](https://github.com/kotxiposix/workshop)