

Workshop 101

ALPHA FLUTTER BARCELONA

Supported by:



#alphaflutterbarcelona



CONTENTS

1. Alpha Health
2. Flutter key concepts
3. Dart
4. Room for improvement
5. Comparison
6. Workshop
7. Time to code



ALPHA

Alpha gathers some of the world's brightest minds to bring **breakthrough technology** and radical solutions to the **planet's biggest challenges**.

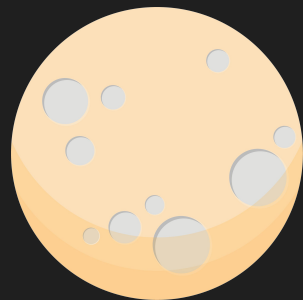
Alpha is different from a corporate R&D group or incubator. Our projects are so **ambitious** that they could take a decade to become successful, and they span subject areas well beyond Telefonica's core businesses.

ALPHA HEALTH

Alpha Health is the factory's **first Moonshot**.

Millions of people live every day with a **chronic disease**, such as diabetes, heart disease, and mental health difficulties, with millions more at risk of them. These diseases are now the **leading cause of death**, and **cost healthcare systems \$5trn a year**.

Our work aims to **encourage behaviour changes** that will radically reduce the burden of chronic disease globally.



OUR MOON

...

OUR SHOT

...





DAVID ANAYA
@danaya77



ROBERTO HUERTAS
@robertohuertasm

ionic
javascript
rxjs web dart
reactive rem!x
angular rust
jest node
flutter redux
stream alpha

“Flutter allows you to build beautiful native apps with great performance on iOS and Android from a single codebase.”

flutter.io



CONCEPTS

- Single codebase
- iOS and Android
- Beautiful designs
- Great performance
- **Fast development**



SINGLE CODEBASE

- Uses dart
<https://www.dartlang.org/>
- Access to platform and services



IOS AND ANDROID

- Platform conventions
- Fonts, buttons, tabs, look & feel, animations



BEAUTIFUL DESIGNS

- UI elements, rendering and animations within the framework
- Control every pixel on the screen
- Material design and cupertino implementations
<https://flutter.io/docs/development/ui/widgets/material>
<https://flutter.io/docs/development/ui/widgets/cupertino>



GREAT PERFORMANCE

- Compiles to native ARM code
- Own render engine (Skia)
- 60fps to 120fps (when supported)



FAST DEVELOPMENT

- Stateful hot reload ($< 1s$)
- JiT and AoT compilation



Dart is a
general-purpose
programming
language originally
developed by
Google.

- **Productive**
Typed language
Thousands of packages
- **Portable**
Compiles to ARM for native apps
Transpiles to JavaScript for web apps
- **Approachable**
Object oriented
Similar to C or Java
- **Reactive**
Futures and Streams
rxDart (Observables and Subjects)



ROOM FOR IMPROVEMENT

<https://github.com/flutter/flutter/wiki/Roadmap>

- Error messages
- Better integration with Google Services
- IDE support (vscode)
- JSON serialization
- Standardization
- Layout composition



| | FLUTTER | PWA | NATIVE | CORDOVA | REACT NATIVE |
|--------------------|----------------|------------|---------------------------------|-----------------------|-----------------------|
| Tech | Dart + Flutter | Javascript | Java/Kotlin Swift/ObjectiveC | Javascript (Ionic) | Javascript + React |
| Performance | ☆☆☆ | ☆ | ☆☆☆ | ☆ | ☆☆ |
| Device features | Device | Browser/OS | Device | Browser/OS | Device |
| Community | ☆☆ | ☆☆☆ | ☆☆☆ | ☆☆☆ | ☆☆☆ |
| Tooling | ☆☆☆ | ☆☆ | ☆☆☆ | ☆☆ | ☆☆ |



WORKSHOP

1. Dart basics
2. VSCode
3. Widgets
4. Application scaffold
5. Flutter composition style
6. Scoped model
7. Hero

TIME TO CODE!

<https://github.com/davidanaya/alpha-flutter-workshop-app>

```
import 'package:alpha_flutter_workshop_app/src/models/repo.dart';
import 'package:alpha_flutter_workshop_app/src/scoped-models/github_api_provider.dart';
import 'package:alpha_flutter_workshop_app/src/widgets/repo_tile.dart';
import 'package:flutter/material.dart';
import 'package:scoped_model/scoped_model.dart';

You, 20 hours ago | 1 author (You)

class MemberReposScreen extends StatelessWidget {
  final Member member;

  MemberReposScreen({@required this.member});
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text(member.login)),
      body: ScopedModelDescendant<GithubApiProvider>(
        builder: (context, child, GithubApiProvider provider) {
          return FutureBuilder(
            future: provider.getReposFromMember(member.login),
            builder: (context, AsyncSnapshot<List<Repo>> snapshot) =>
              _buildList(snapshot.hasData ? snapshot.data : []); // FutureBuilder
          ); // ScopedModelDescendant // Scaffold
        }
      ); // ListView.builder
    ); // ListView.builder
  }

  Widget _buildList(List<ListItem> repos) {
    final items = new List<Repo>()
      ..addAll(repos.map((memberRepos) => memberRepos.items));
    return ListView.builder(
      itemCount: items.length,
      itemBuilder: (context, index) {
        final item = items[index];

        if (item is Repo) {
          return RepoTile(item);
        } else if (item is MemberAvatar) {
          return _buildImage(item.avatarUrl);
        }
      },
    ); // ListView.builder
  }

  Widget _buildImage(String imageUrl) {
    return Hero(
      tag: member.login,
      child: Container(
        padding: EdgeInsets.all(32.0), child: Image.network(imageUrl)), // Container
    ); // Hero
  }
}
```

```
import 'dart:async';
import 'dart:convert';

import 'package:alpha_flutter_workshop_app/src/models/member.dart';
import 'package:alpha_flutter_workshop_app/src/models/repo.dart';
import 'package:http/http.dart' as http;
import 'package:scoped_model/scoped_model.dart';

You, 21 hours ago | 1 author (You)

class GithubApiProvider extends Model {
  Future<List<Member>> getMembers() async {
    final response =
      await http.get('https://api.github.com/orgs/Alpha-health/members');
    final List<dynamic> parsedList = json.decode(response.body);
    final membersList =
      parsedList.map((parsed) => Member.fromJson(parsed)).toList();
    return membersList;
  }

  Future<List<Repo>> getReposFromMember(String memberName) async {
    final response =
      await http.get('https://api.github.com/users/$memberName/repos');
    final List<dynamic> parsedList = json.decode(response.body);
    final reposList =
      parsedList.map((parsed) => Repo.fromJson(parsed)).toList();
    return reposList;
  }
}
```