

Flutter Multiplataforma + UI Builder amb FlutterFlow

Ainhoa Sanchez Salvago
SALESIANS SARRIÀ | S2AM

ÍNDEX

1.	COMPRENDRE QUE EL FLUTTER ÉS MULTIPLATAFORMA	2
1.1.	DIFERÈNCIES ENTRE DART NATIVE I DART WEB.....	2
1.2.	QUÈ VOL DIR JIT I AOT? PER A QUÈ SERVEIXEN CADASCUNA D'ELLES?.....	2
2.	CREAR UNA APLICACIÓ PER WINDOWS AMB FLUTTER.....	2
3.	CREAR I ENTENDRE UNA UI AMB FLUTTERFLOW.IO.....	5
3.1.	APP AURORES BOREALS.....	5
3.2.	LA MEVA APP	6
3.2.1.	Home Page	7
3.2.2.	Lectures preferides.....	8
3.2.3.	Informació	10
3.2.4.	Lectures Pendents.....	10
4.	Videos	10

1. COMPRENDRE QUE EL FLUTTER ÉS MULTIPLATAFORMA

1.1. DIFERÈNCIES ENTRE DART NATIVE I DART WEB

El Dart Native s'utilitza per desenvolupar aplicacions que s'executen directament sobre el sistema operatiu com aplicacions d'escriptori o aplicacions mòbils. En aquest cas, el codi Dart es compila a codi màquina natiu, això fa que l'aplicació pugui funcionar d'una manera més eficient i amb accés directe a les funcionalitats del sistema operatiu.

En canvi, Dart Web s'utilitza per crear aplicacions que funcionen dins d'un navegador Web. Com que els navegadors no entenen Dart directament el codi s'ha de compilar amb JavaScript abans de poder executar-se.

En resum, Dart Native està pensat per a aplicacions d'alt rendiment que s'executen directament sobre el dispositiu, mentre que Dart Web està orientat a aplicacions que han de funcionar dins del navegador.

1.2. QUÈ VOL DIR JIT I AOT? PER A QUÈ SERVEIXEN CADASCUNA D'ELLES?

JIT i AOT són dos mètodes de compilació de codi que serveixen per a optimitzar el rendiment dels programes. D'una banda, tenim el JIT que vol dir “Just in time”, compila el codi durant l'execució del programa. D'altra banda, el AOT que vol dir “Ahead of Time”, compila el codi complet abans que s'executi. Tot i que el mètode JIT és més ràpid per al desenvolupament, el AOT produeix un rendiment més ràpid i una aplicació més lleugera al final.

2. CREAR UNA APLICACIÓ PER WINDOWS AMB FLUTTER

Per tal de comprovar que l'aplicació de l'anterior pràctica funciona també bé amb un ordinador Windows, ho executarem una altra vegada però aquest cop escollint el nostre dispositiu Windows.

Li donem a executar i veiem que ens dona l'opció d'escollir el dispositiu. Un cop fet això, en el meu cas em dona un error. Per veure exactament de quin error es tracta, executarem “flutter doctor”.

```

File Edit Selection View Go Run Terminal Help
EXPLORER FLUTTER_APPLICATION_1
  > .dart_tool
  > idea
  > android
  > build
  > ios
  > lib
    > main.dart
    > linux
    > macos
  > test
    > widget_test.dart
  > web
    > windows
      > .gitignore
      > .metadata
      > analysis_options.yaml
      > flutter_application_1.yaml
      > pubspec.lock
      > pubspec.yaml
      > README.md
  > OUTLINE
  > TIMELINE
  > DEPENDENCIES
  > JAVA PROJECTS
  > Java Ready

main.dart x Select a device to use
Current Device
Available Devices
lib > main.dart
12   class MyWidget {
13     Widget
14     reti
15     t
16     t
17     // This is the theme of your application.
18     // TRY THIS: Try running your application with "flutter run". You'll see
19     // the application has a purple toolbar. Then, without quitting the app,
20     // try changing the seedColor in the colorScheme below to Colors.green
21     // and then invoke "hot reload" (save your changes or press the "hot
22     // reload" button in a Flutter-supported IDE, or press "r" if you used
23     // the command line to start the app).
24     //
25     // Notice that the counter didn't reset back to zero; the application
26     // state is not lost during the reload. To reset the state, use hot
27     // restart instead.
28     //
29     // This works for code too, not just values: Most code changes can be
30     // tested with just a hot reload.
31     colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
32     ), // ThemeData
33     home: const MyHomePage(title: 'Flutter Demo Ainhoa Sánchez SZAM'),
34   ); // MaterialApp
35 }
36 }
37
38 class MyHomePage extends StatefulWidget {
39   const MyHomePage({super.key, required this.title});
40
41   // This widget is the home page of your application. It is stateful, meaning
42   // that it has a State object (defined below) that contains fields that affect
43   // how it looks.
44
45   // This class is the configuration for the state. It holds the values (in this
46   // case the title) provided by the parent (in this case the App widget) and
47   // used by the build method of the State. Fields in a Widget subclass are
48   // always marked "final".
49
50   final String title;
51
52   @override
53   State<MyHomePage> createState() => _MyHomePageState();
54 }

Ln 30, Col 42 Spaces: 2 UTF-8 CR/LF ⓘ Dart ⌂ Go Live Windows (windows-x64) ⌂ Prettier ⌂

```

```

File Edit Selection View Go Run Terminal Help
EXPLORER FLUTTER_APPLICATION_1
  > .dart_tool
  > idea
  > android
  > build
  > ios
  > lib
    > main.dart
    > linux
    > macos
  > test
    > widget_test.dart
  > web
    > windows
      > .gitignore
      > .metadata
      > analysis_options.yaml
      > flutter_application_1.yaml
      > pubspec.lock
      > pubspec.yaml
      > README.md
  > OUTLINE
  > TIMELINE
  > DEPENDENCIES
  > JAVA PROJECTS
  > Java Ready

main.dart x Flutter: Run Flutter Doctor recently used ⓘ
Flutter New Project
Flutter Upgrade
Accounts: Manage Extension Account Preferences
Accounts: Manage Trusted Extensions For Account
Accounts: Manage Trusted MCP Servers For Account
Add Cursor Above
Add Cursor Below
Add Cursors to Bottom
Add Cursors to Line Ends
Add Cursors to Top
Add Data Breakpoint at Address
Add Function Breakpoint
Add Line Comment
Add Selection to Next Find Match
Ctrl + K Ctrl + C Ctrl + D
Ctrl + Alt + UpArrow
Ctrl + Alt + DownArrow
Shift + Alt + I
// restart instead.
//
// This works for code too, not just values: Most code changes can be
// tested with just a hot reload.
colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
), // ThemeData
home: const MyHomePage(title: 'Flutter Demo Ainhoa Sánchez SZAM'),
); // MaterialApp
}
}

class MyHomePage extends StatefulWidget {
const MyHomePage({super.key, required this.title});
}

// This widget is the home page of your application. It is stateful, meaning
// that it has a State object (defined below) that contains fields that affect
// how it looks.

// This class is the configuration for the state. It holds the values (in this
// case the title) provided by the parent (in this case the App widget) and
// used by the build method of the State. Fields in a Widget subclass are
// always marked "final".

final String title;

@Override
State<MyHomePage> createState() => _MyHomePageState();
}

Ln 30, Col 42 Spaces: 2 UTF-8 CR/LF ⓘ Dart ⌂ Go Live Windows (windows-x64) ⌂ Prettier ⌂

```

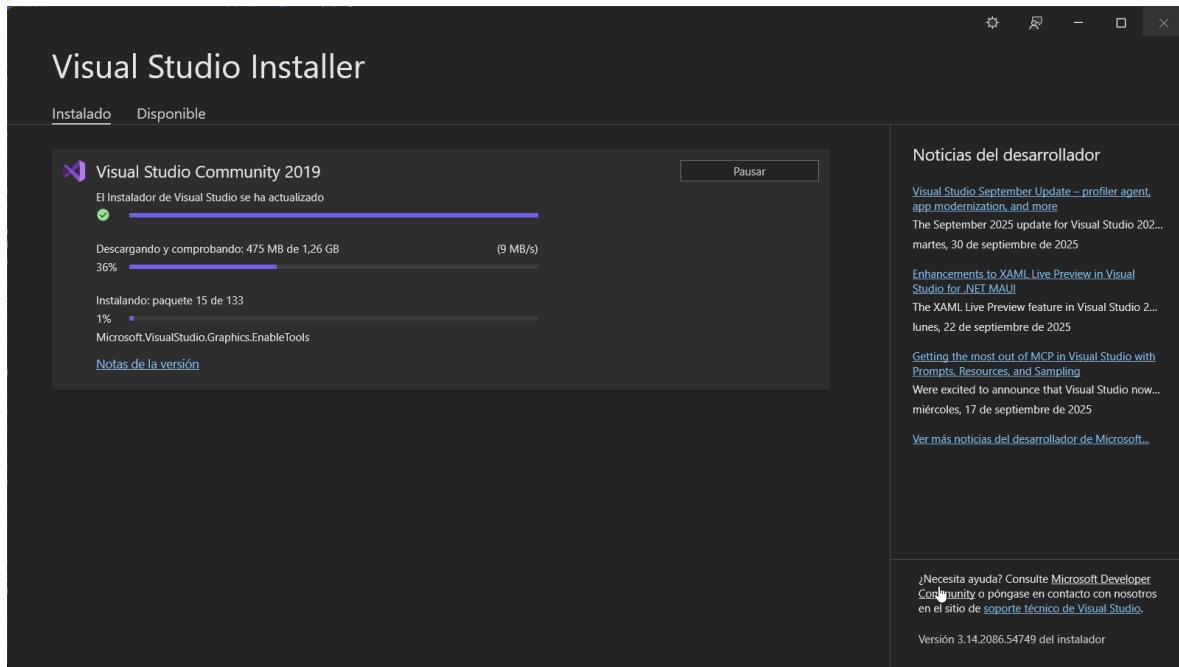
I veiem que ens diu que faltén alguns components necessaris per tal d'executar-ho.

```

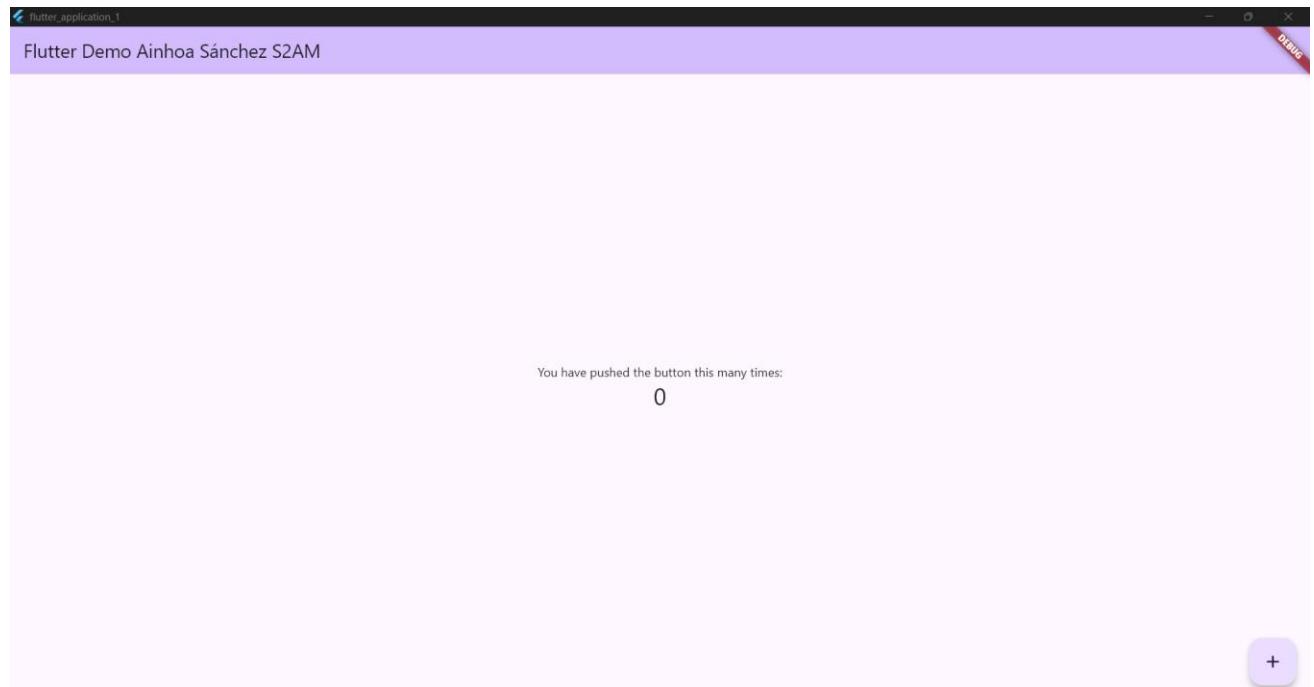
[!] Visual Studio - develop Windows apps (Visual Studio Community 2019 16.11.42) [218ms]
  • Visual Studio at C:\Program Files (x86)\Microsoft Visual Studio\2019\Community
  • Visual Studio Community 2019 version 16.11.35425.106
X Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the "Desktop development with C++" workload, and include these components:
  MSVC v142 - VS 2019 C++ x64/x86 build tools
    - If there are multiple build tool versions available, install the latest
  C++ CMake tools for Windows
  Windows 10 SDK

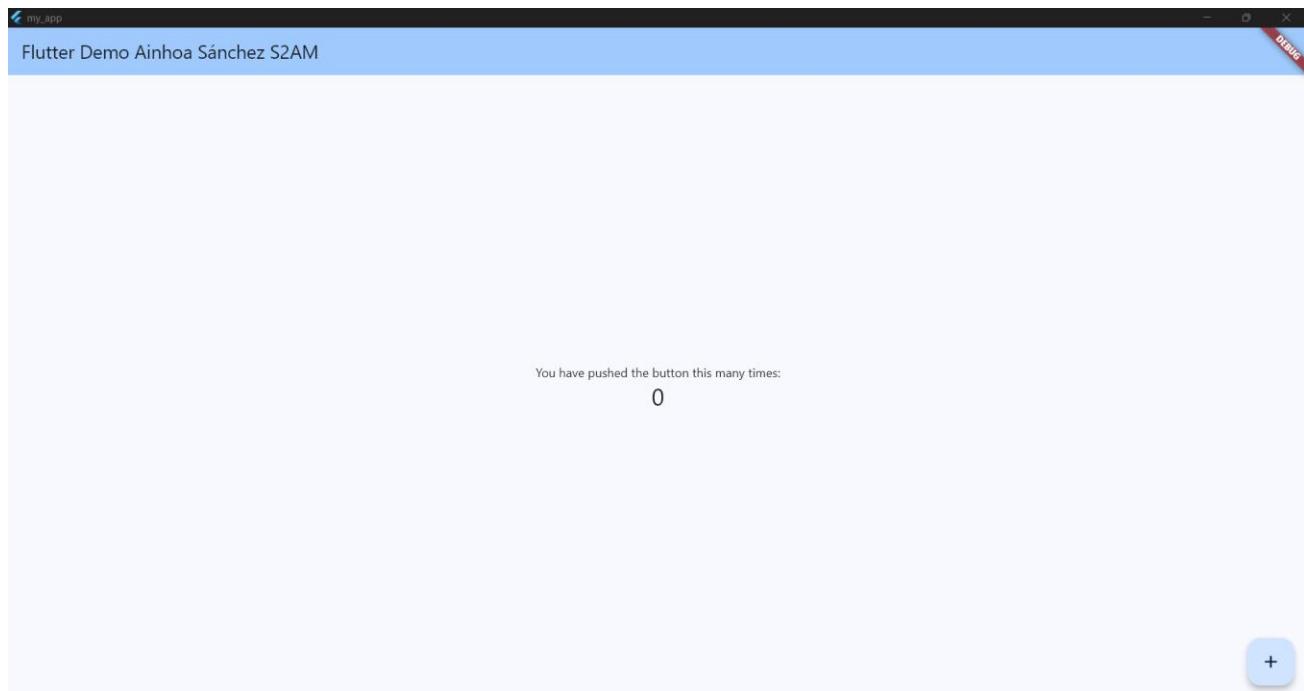
```

Anem al “Visual Studio installer” i cliquem sobre l’opció de modificar, afegim l’opció de “Desarrollo d’escritori con C++”



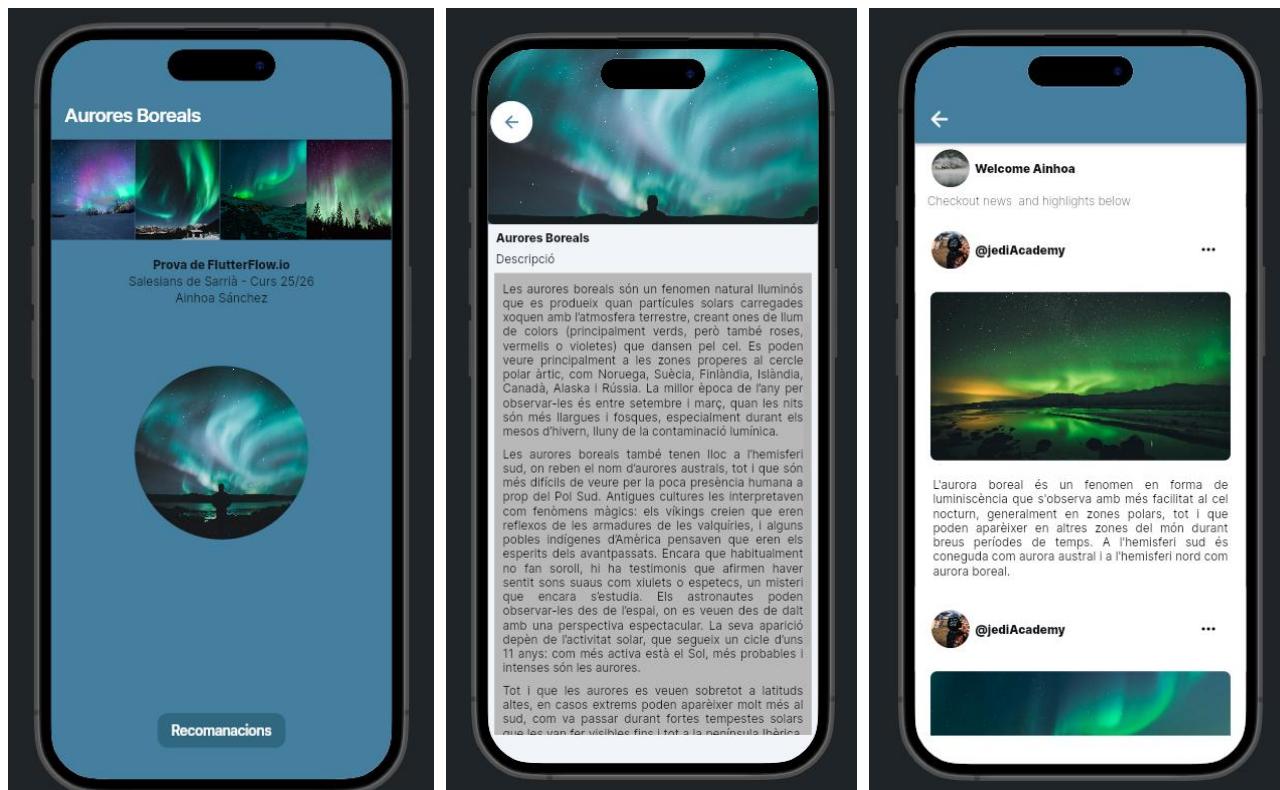
Un cop acabi d'instal·lar-se tot, tornarem a executar el “flutter doctor” per comprovar que tot està correctament i si veiem que no hi ha mes errors, podrem executar la nostra aplicació sense cap problema. Aquest mateix procés funciona tant amb el Visual Studio Code com amb l’Android Studio.





3. CREAR I ENTENDRE UNA UI AMB FLUTTERFLOW.IO

3.1. APP AURORES BOREALS

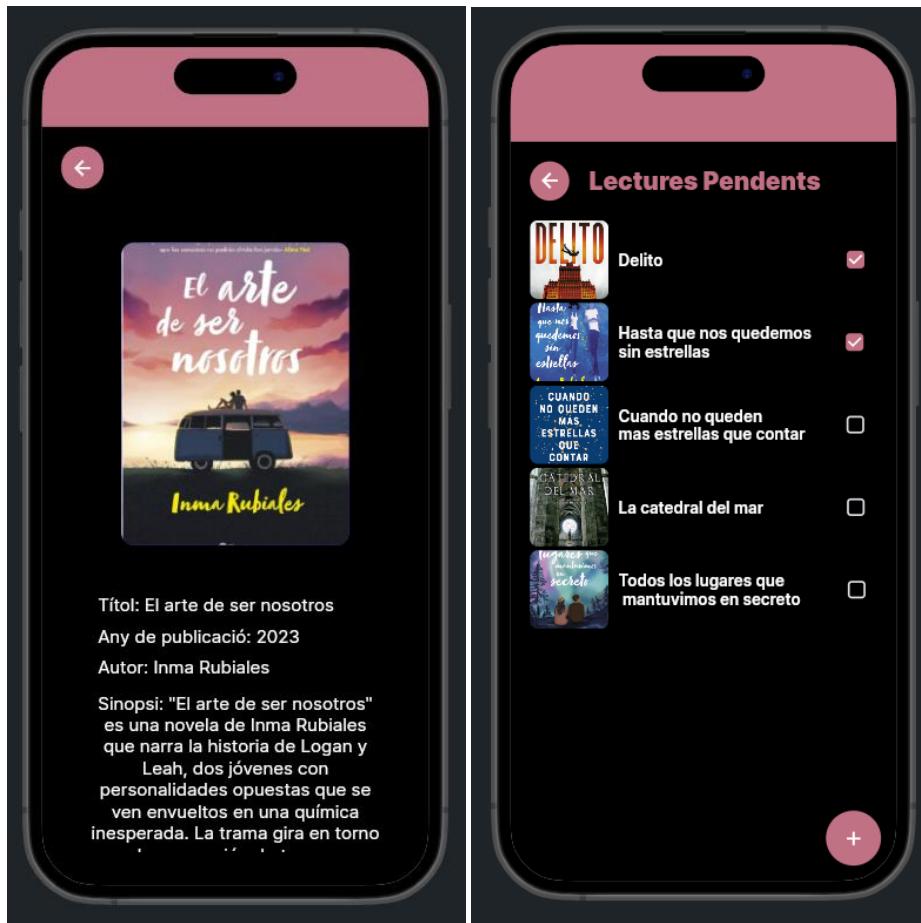


Aquesta aplicació funciona de la següent manera, a la pantalla principal trobem un seguit d'imatges les quals es poden anar a passant. En segon lloc veiem una imatge circular la qual podem clicar i se'n obra la segona pantalla, on trobem una descripció que ens parla de les aurores boreals i al fer "scroll" cap amunt podem seguir llegint. Finalment, si des de la pantalla principal cliquem sobre el botó de recomanacions se'n obre la tercera pantalla on trobem el que sembla una xarxa social, on diferents usuaris pengen imatges i comentaris amb recomanacions per altres usuaris.

3.2. LA MEVA APP

Per tal de poder practicar amb Flutter Flow creant una app des de zero, he decidit fer el disseny d'una App de llibres, amb l'objectiu de que l'usuari pugui tenir un llistat de llibres que li agraden cadascun amb la seva informació i una altre llistat on va afegint els llibres que vol llegir.





3.2.1. Home Page

En primer lloc, a la pantalla principal trobem un text amb un tamany de 25 i de color rosa. Aquest té un distància superior de 40.

```
child: Column(
  mainAxisSize: MainAxisSize.max,
  children: [
    Padding(
      padding: EdgeInsetsDirectional.fromSTEB(0, 40, 0, 0),
      child: Text(
        'Benvinguda Ainhoa!',
        textAlign: TextAlign.center,
        style: FlutterFlowTheme.of(context).bodyLarge.override(
          font: GoogleFonts.inter(
            fontWeight: FontWeight.w900,
            fontStyle:
              FlutterFlowTheme.of(context).bodyLarge.fontStyle,
          ),
          color: Color(0xFFC07184),
          fontSize: 25,
          letterSpacing: 0.0,
          fontWeight: FontWeight.w900,
          fontStyle:
            FlutterFlowTheme.of(context).bodyLarge.fontStyle,
        ),
      ),
    ),
  ],
),
```

Seguidament, trobarem una imatge circular que té una distància superior de 80 i una inferior de 70. El “circle image” té una mida de 200x200.

```
Padding(  
    padding: EdgeInsetsDirectional.fromSTEB(0, 80, 0, 70),  
    child: Container(  
        width: 200,  
        height: 200,  
        clipBehavior: Clip.antiAlias,  
        decoration: BoxDecoration(  
            shape: BoxShape.circle,  
        ),  
        child: Image.network(  
            'https://cdn-icons-png.flaticon.com/512/5550/5550463.png',  
            fit: BoxFit.cover,  
        ),  
    ),  
,  
)
```

En tercer lloc, tenim un container que conte dos botons, un que ens porta a la pagina de lectures preferides i l'altre que ens porta a la pàgina de lectures pendents. Com els dos botons són iguals, tenen tots dos el mateix codi. Podem veure que tenen una mida de 120x230 i que tenen un borderColor de mida 4.

```
child: Column(
  mainAxisSize: MainAxisSize.max,
  children: [
    FFButtonWidget(
      onPressed: () async {
        context.pushNamed(PendentsWidget.routeName);
      },
      text: 'Pendents',
      options: FFButtonOptions(
        width: 120,
        height: 230,
        padding: EdgeInsetsDirectional.fromSTEB(
          16, 0, 16, 0),
        iconPadding: EdgeInsetsDirectional.fromSTEB(
          0, 0, 0, 0),
        color: Color(0xFFC07184),
        textStyle: FlutterFlowTheme.of(context)
          .titleSmall
          .override(
            font: GoogleFonts.interTight(
              fontWeight:
                FlutterFlowTheme.of(context)
                  .titleSmall
                  .fontWeight,
            ),
            color: Colors.white,
            letterSpacing: 0.0,
            fontWeight:
              FlutterFlowTheme.of(context)
                .titleSmall
                .fontWeight,
            fontStyle:
              FlutterFlowTheme.of(context)
                .titleSmall
                .fontStyle,
          ),
          elevation: 0,
          borderSide: BorderSide(
            color: Color(0xFF90364C),
            width: 4,
          ),
        ),
      ),
    ],
  ),
);
```

3.2.2. Lectures preferides

En aquesta pantalla, el primer que trobem després de l'AppBar és una fila la qual conté un “Icon Button” i un text. Seguidament, trobem una “List View” la cual dins conté un container que té dins un seguit de files cadascuna amb una imatge (la portada del llibre) i el seu títol.

```
appBar: AppBar(
  backgroundColor: Color(0xFFC07184),
  automaticallyImplyLeading: false,
  actions: [],
  centerTitle: true,
  elevation: 0,
),
```

En la següent imatge podem veure que dins primera fila de la pàgina trobem la mida del botó i que quan el pressionem ens portarà a la pantalla principal (la pantalla anterior)

```
child: Column(
  mainAxisAlignment: MainAxisAlignment.max,
  children: [
    Padding(
      padding: EdgeInsetsDirectional.fromSTEB(20, 20, 0, 4),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.max,
        children: [
          FlutterFlowIconButton(
            borderRadius: 20,
            buttonSize: 40,
            fillColor: Color(0xFFC07184),
            icon: Icon(
              Icons.arrow_back,
              color: FlutterFlowTheme.of(context).info,
              size: 24,
            ),
            onPressed: () async {
              context.pushNamed(HomePageWidget.routeName);
            },
          ),
        ],
      ),
    ),
  ],
),
```

En segon lloc, podem veure una de les files, la qual té una imatge associada de mida 80x80 i un text el qual és el títol del llibre. I el onTap() ens indica que al pressionar sobre la primera fila del container, se'ns obrirà una pantalla d'informació, la qual ens parla una mica més a fons sobre el llibre seleccionat.

<pre>child: Column(mainAxisAlignment: MainAxisAlignment.max, children: [Align(alignment: AlignmentDirectional(-1, 0), child: Padding(padding: EdgeInsetsDirectional.fromSTEB(20, 20, 20, 2), child: InkWell(splashColor: Colors.transparent, focusColor: Colors.transparent, hoverColor: Colors.transparent, highlightColor: Colors.transparent, onTap: () async { context.pushNamed(InformacioWidget.routeName); },),),),],),</pre>	<pre>child: Row(mainAxisAlignment: MainAxisAlignment.max, children: [Align(alignment: AlignmentDirectional(-1, 0), child: ClipRRect(borderRadius: BorderRadius.circular(8), child: Image.asset('assets/images/elartedesernosotros.png', width: 80, height: 80, fit: BoxFit.cover,),),),],),</pre>
--	---

3.2.3. Informació

Aquesta pantalla està composta d'una “flip Card” i a sota, tenim diversa informació sobre el llibre.

```
FlipCard(
  fill: Fill.fillBack,
  direction: FlipDirection.HORIZONTAL,
  speed: 400,
  front: Align(
    alignment: AlignmentDirectional(0, 0),
    child: Padding(
      padding: EdgeInsetsDirectional.fromSTEB(0, 55, 0, 0),
      child: Container(
        width: 230.6,
        height: 307.1,
        decoration: BoxDecoration(
          color: FlutterFlowTheme.of(context).primary,
          image: DecorationImage(
            fit: BoxFit.cover,
            image: Image.asset(
              'assets/images/elartedesnosotros.png',
            ).image,
          ),
          borderRadius: BorderRadius.circular(12),
        ),
      ),
    ),
  ),
  back: Align(
    alignment: AlignmentDirectional(0, 0),
    child: Padding(
      padding: EdgeInsetsDirectional.fromSTEB(0, 55, 0, 0),
      child: Container(
        width: 230.6,
        height: 307.1,
        decoration: BoxDecoration(
          color: FlutterFlowTheme.of(context).primary,
          image: DecorationImage(
            fit: BoxFit.cover,
            image: Image.asset(
              'assets/images/contraportada.png',
            ).image,
          ),
          borderRadius: BorderRadius.circular(12),
        ),
      ),
    ),
  ),
)
```

3.2.4. Lectures Pendants

Aquesta pantalla té exactament la mateixa estructura, ja comentada, de la pantalla de lectures preferides. L'única diferència és que en cada fila hi ha, a més a més, un “check box”.

4. Videos

https://drive.google.com/file/d/1On8qeC9qY71Nsln0vsZoXfc_t84PF_yI/view?usp=drive_link

https://drive.google.com/file/d/1CLB-2_sVAj3173k-25numPkg0WT3Vsef/view?usp=drive_link

AINHOA SÁNCHEZ

SALESIANS SARRIÀ

06/10/2025