

Flutter 2 Start

il più completo videocorso in italiano su Flutter

Fudeo

01

Cos'è un'app

02

Insieme

1. Schermate
2. Componenti
3. Servizi nativi (gps)
4. Chiamate HTTP

1.Schermate e 2.Componenti

Rappresentano la parte grafica di un app.

Le schermate (o pagine) sono alla fine dei contenitori di componenti.

I componenti (o widget) sono alla fine dei pezzi di schermate.

3. Servizi nativi

Sono dei servizi (o API) offerti dalla piattaforma sottostante.

Flutter può essere usato per creare app su:

- iOS / Android: accedere al GPS
- Web: interagire con JavaScript
- Desktop: accedere al FileSystem

4. Chiamate HTTP

(Quasi) ogni app ha bisogno di accedere ad internet, per:

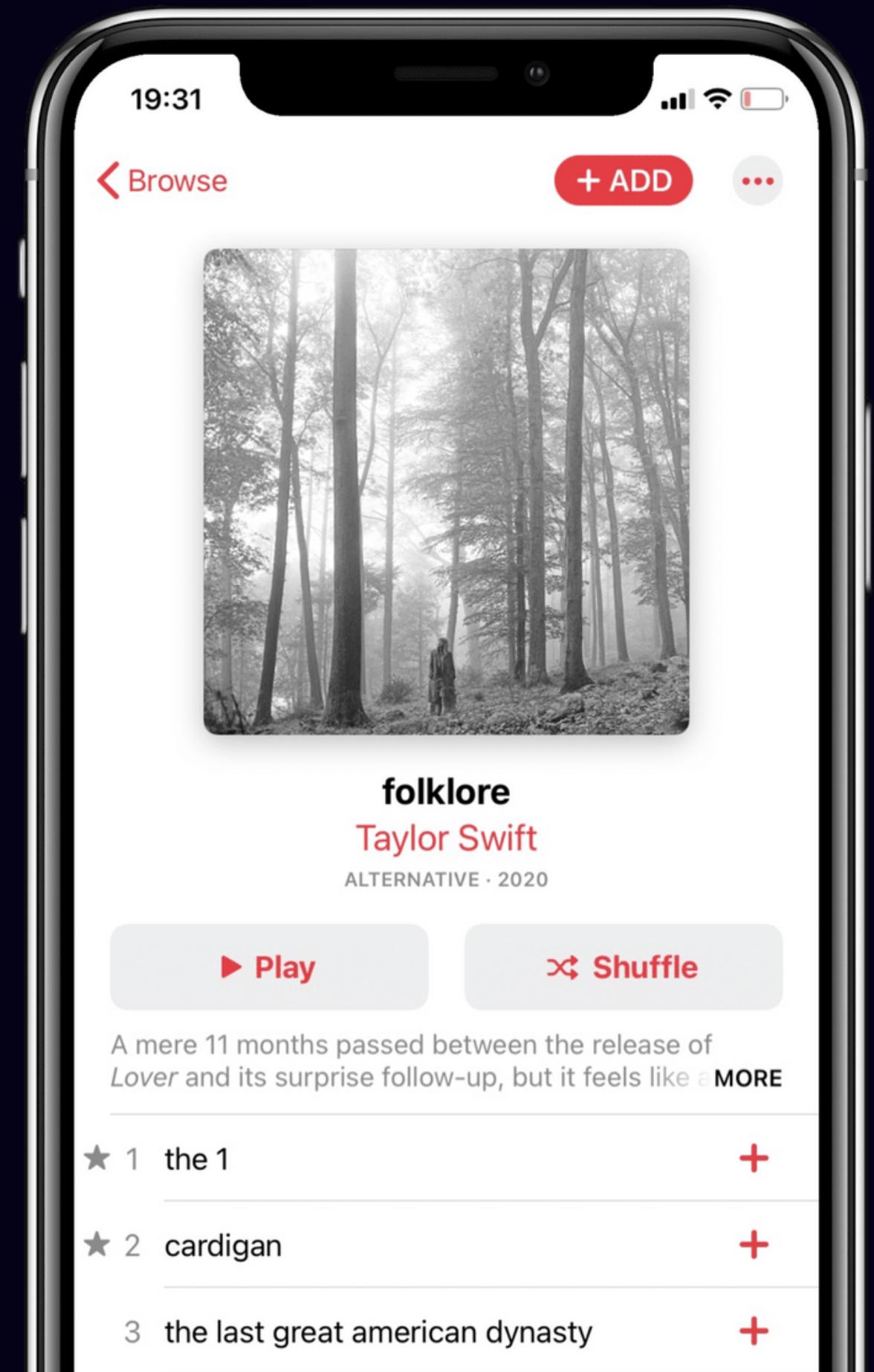
- Autenticazione
- Scaricare dati
- Ricevere notifiche
- Elaborare acquisti
- Trasmettere log

I componenti

02

Frammenti di grafica

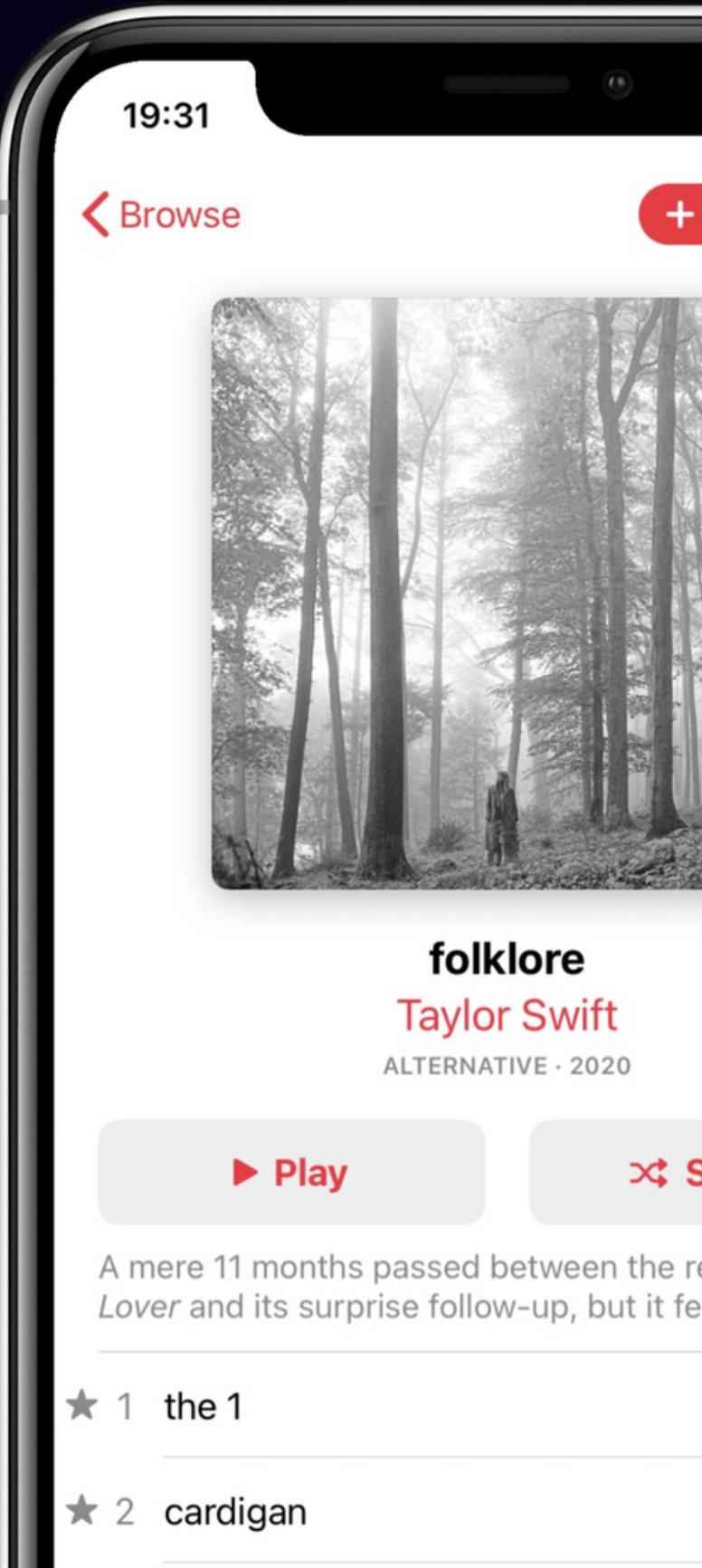
F



04

Frammenti di grafica

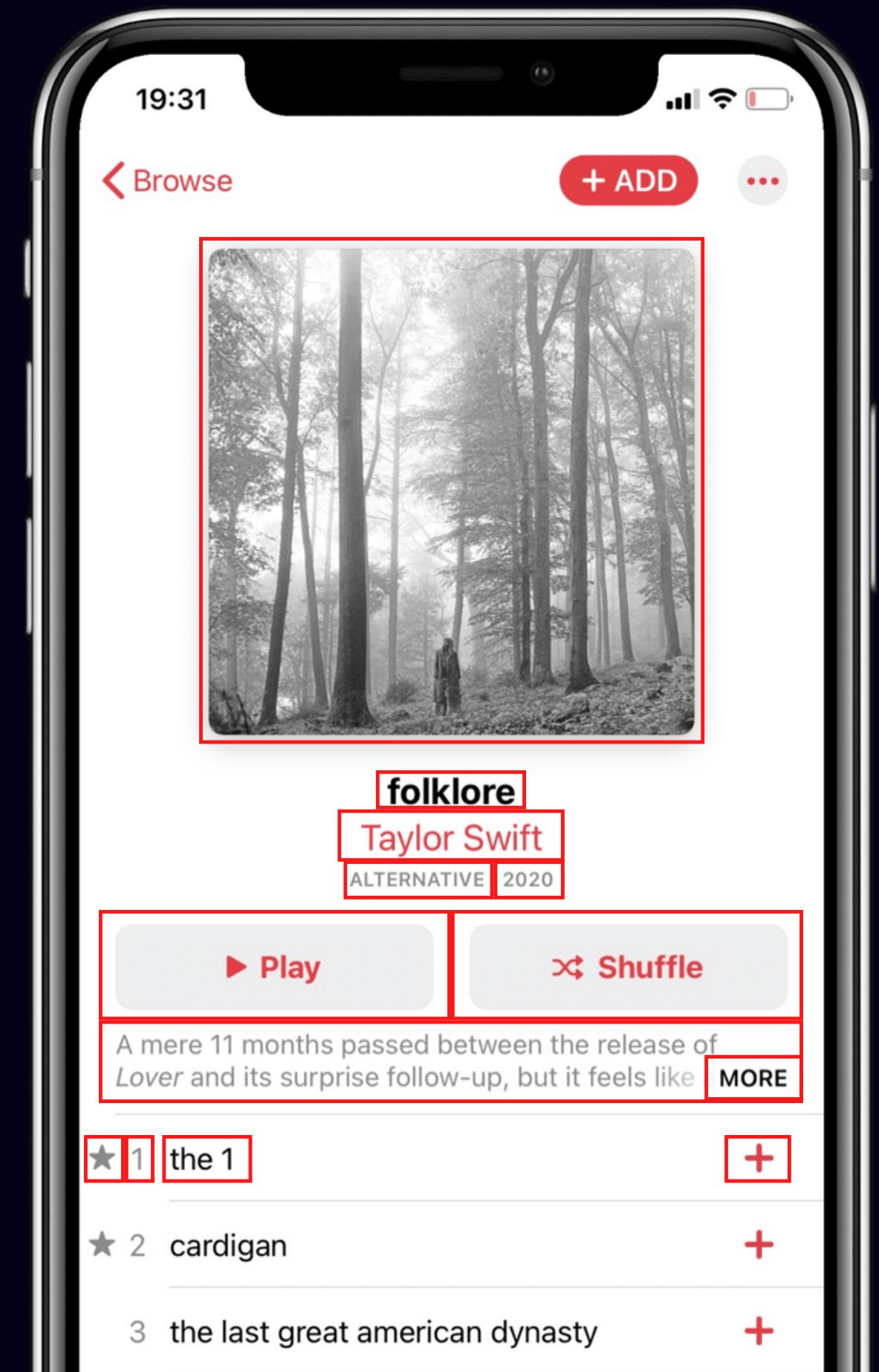
- Una **struttura** (testo, immagine)
- Uno **stile** (colori, bordi, margini)
- Una **logica** (click, doppio click, scroll)





Frammenti di grafica

COMPONENTI



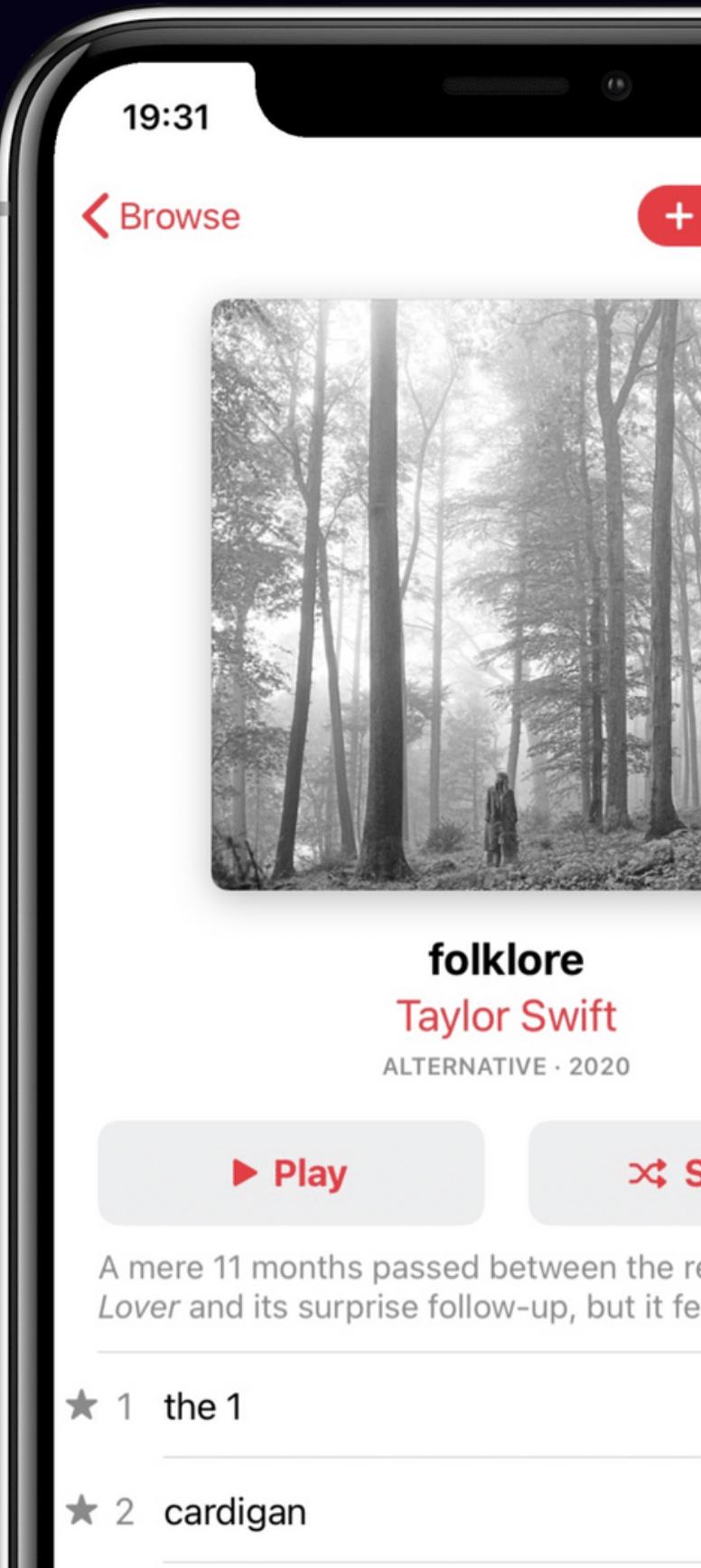
06

Web vs Flutter

07

Web

- **Struttura:** file HTML
- **Stile:** file CSS
- **Logica:** file JavaScript



08

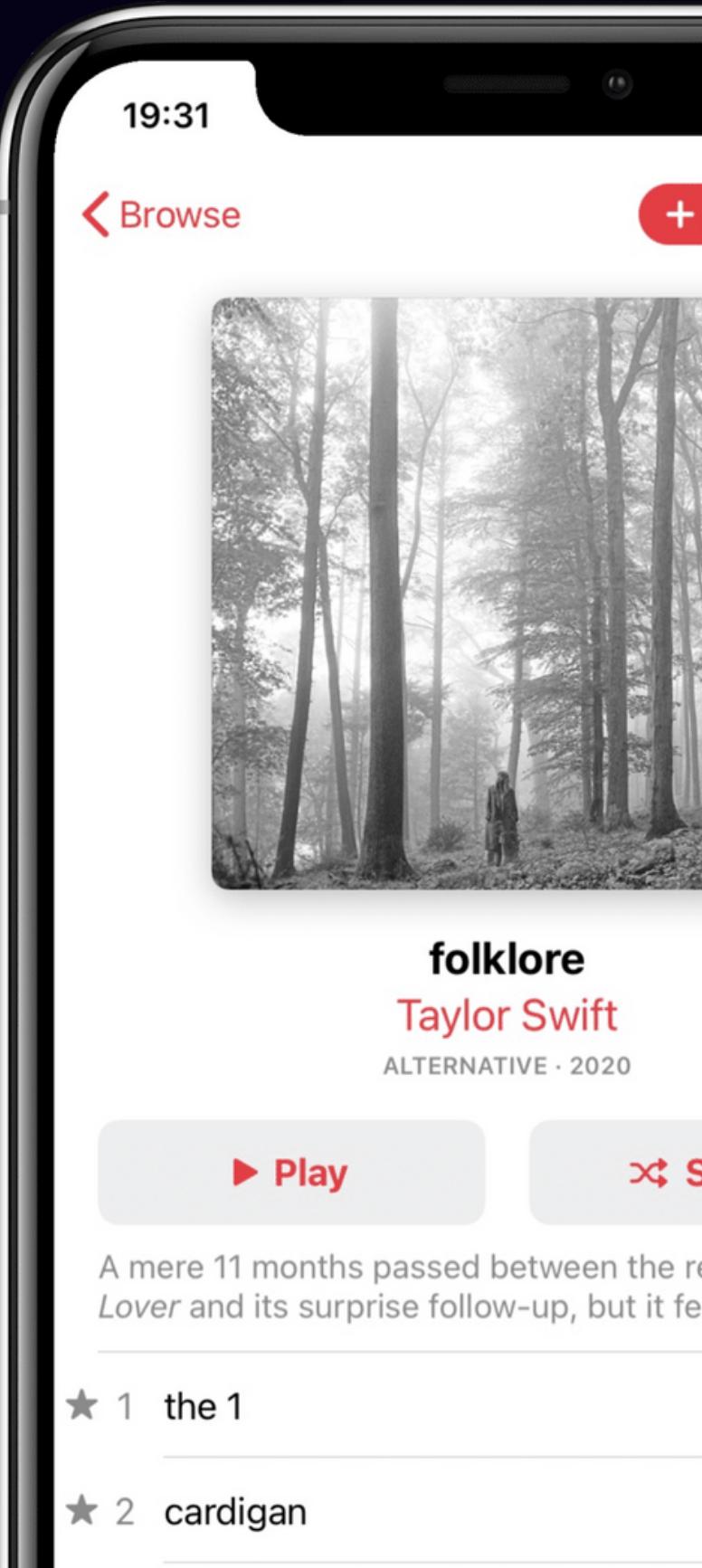
Web

- **3** linguaggi diversi
- **3** file diversi
- **3** sviluppatori diversi (scherzo!)

09

Flutter

- **Struttura:** stesso file Dart
- **Stile:** stesso file Dart
- **Logica:** stesso file Dart



10

Flutter

Un componente Flutter contiene al suo interno tutto:

- Struttura
- Stile
- Logica

Web e Flutter: codice

12

Web

```
<button id="play">Play</button>
```

```
#play {  
    background-color: gray;  
    color: red;  
}
```

```
document  
    .getElementById("play")  
    .addEventListener("click", function() {  
        console.log("Play cliccato");  
    })
```

13

Flutter

```
ElevatedButton(  
    style: ElevatedButton.styleFrom(  
        primary: Colors.grey,  
        textStyle: TextStyle(  
            color: Colors.red,  
        ),  
    ),  
    onPressed: () {  
        print("Play cliccato");  
    },  
    child: Text("Play"),  
)
```

14

Web

```
<button id="play">Play</button>  
  
#play {  
  background-color: gray;  
  color: red;  
}  
  
document  
  .getElementById("play")  
  .addEventListener("click", function() {  
    console.log("Play cliccato");  
})
```

Flutter

```
ElevatedButton(  
  style: ElevatedButton.styleFrom(  
    primary: Colors.grey,  
    textStyle: TextStyle(  
      color: Colors.red,  
    ),  
    ),  
  onPressed: () {  
    print("Play cliccato");  
  },  
  child: Text("Play"),  
)
```

15

Web

```
<button id="play">Play</button>  
  
#play {  
  background-color: gray;  
  color: red;  
}  
  
document  
  .getElementById("play")  
  .addEventListener("click", function() {  
    console.log("Play cliccato");  
  })
```

Flutter

```
ElevatedButton(  
  style: ElevatedButton.styleFrom(  
    primary: Colors.grey,  
    textStyle: TextStyle(  
      color: Colors.red,  
    ),  
    ),  
  onPressed: () {  
    print("Play cliccato");  
  },  
  child: Text("Play"),  
)
```

16

Web

```
<button id="play">Play</button>

#play {
  background-color: gray;
  color: red;
}

document
  .getElementById("play")
  .addEventListener("click", function() {
    console.log("Play cliccato");
  })

```

Flutter

```
ElevatedButton(
  style: ElevatedButton.styleFrom(
    primary: Colors.grey,
    textStyle: TextStyle(
      color: Colors.red,
    ),
  ),
  onPressed: () {
    print("Play cliccato");
  },
  child: Text("Play"),
)
```

Come creiamo componenti

1. In un file .dart
2. Usando il concetto di classe
3. Che estende un StatelessWidget
4. All'interno del metodo build
5. Che ritorna un componente di sistema

18

1. File .dart

main.dart

2. Usando una classe

main.dart

```
class HomePage {  
}
```

3. Estende StatelessWidget

main.dart

```
class HomePage extends StatelessWidget {  
}
```

4. Dentro metodo build

main.dart

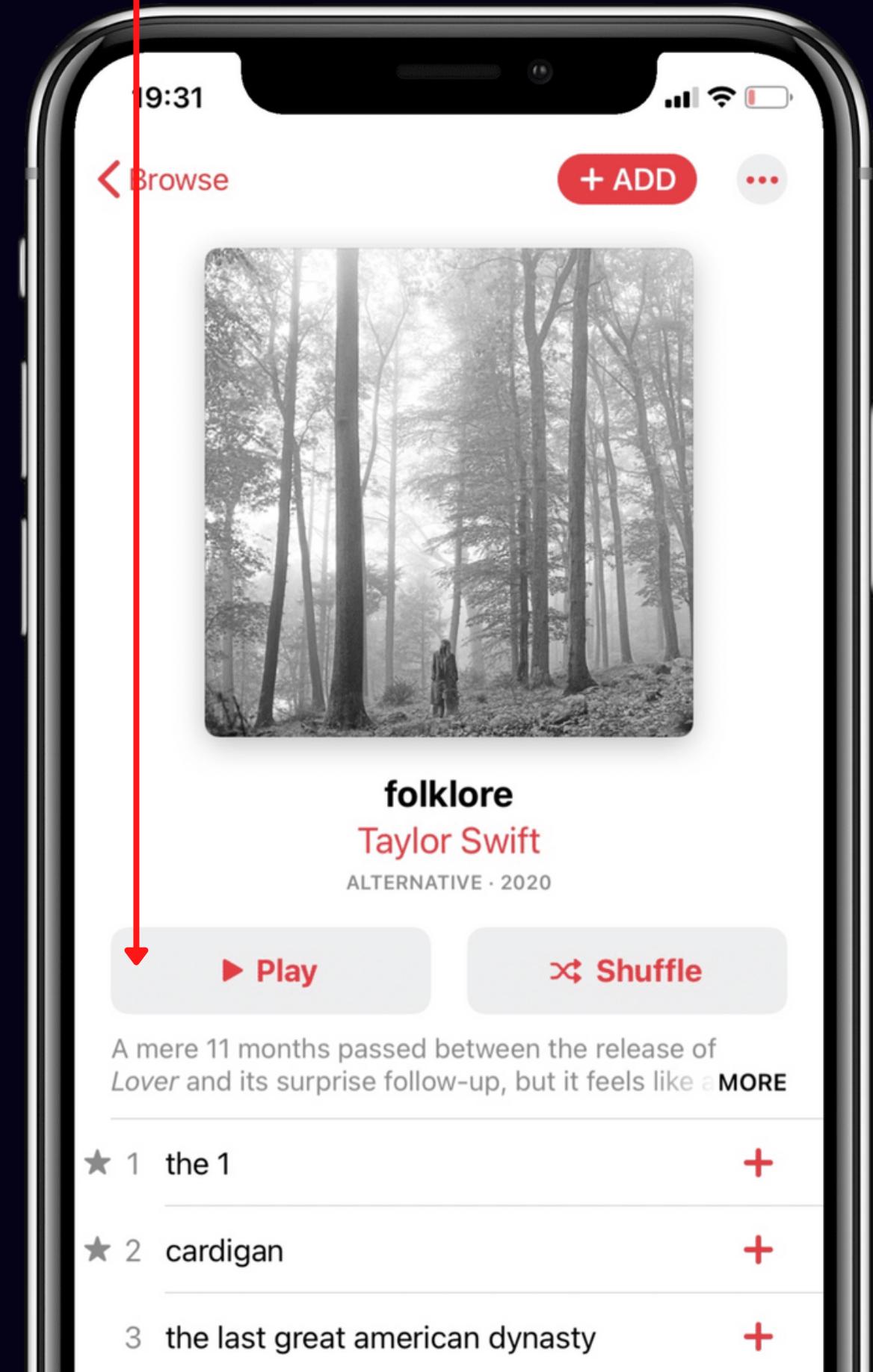
```
class HomePage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
  
  }  
}
```

5. Ritornando il componente

main.dart

```
class HomePage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Text("Welcome!");  
  }  
}
```

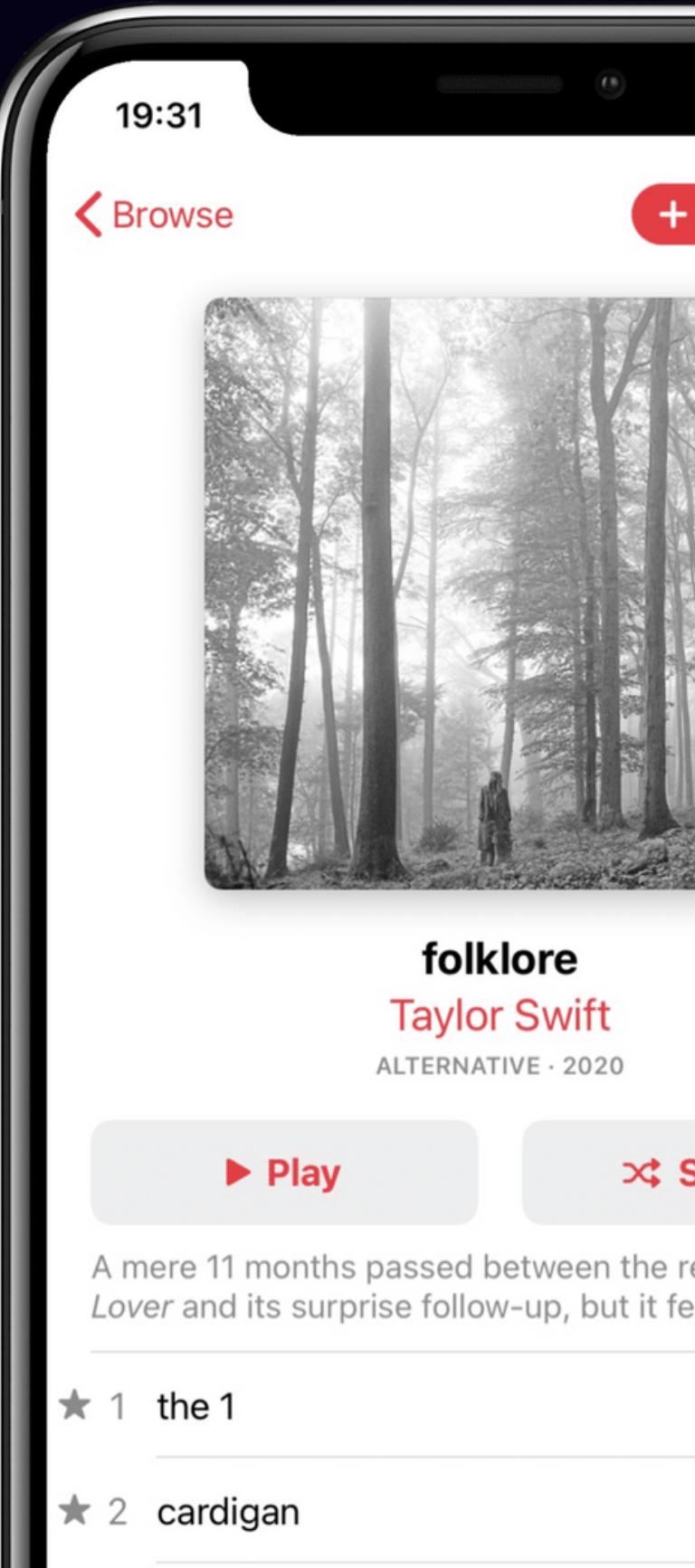
F



24

Cos'è una schermata?

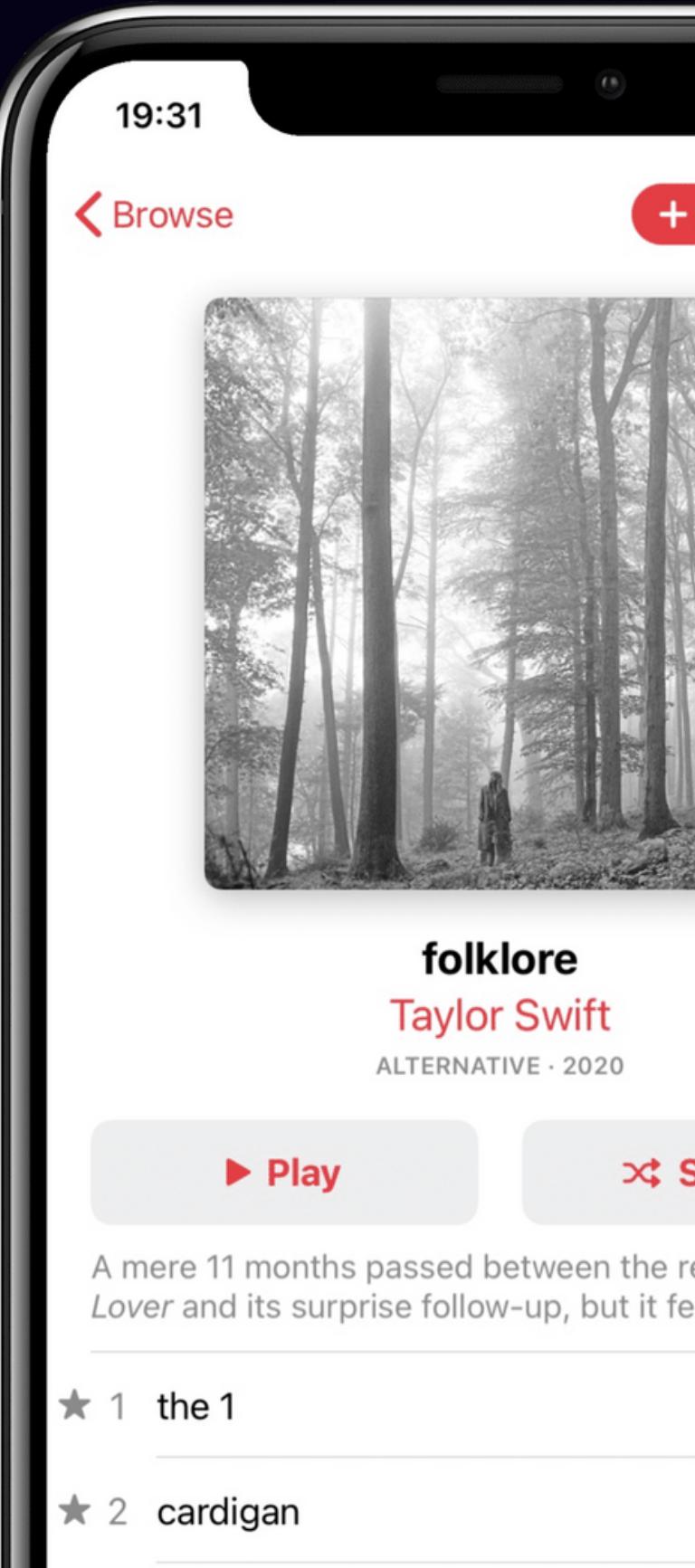
— Frammenti di grafica, uniti insieme in modo armonioso e funzionale.



25

Cos'è un componente?

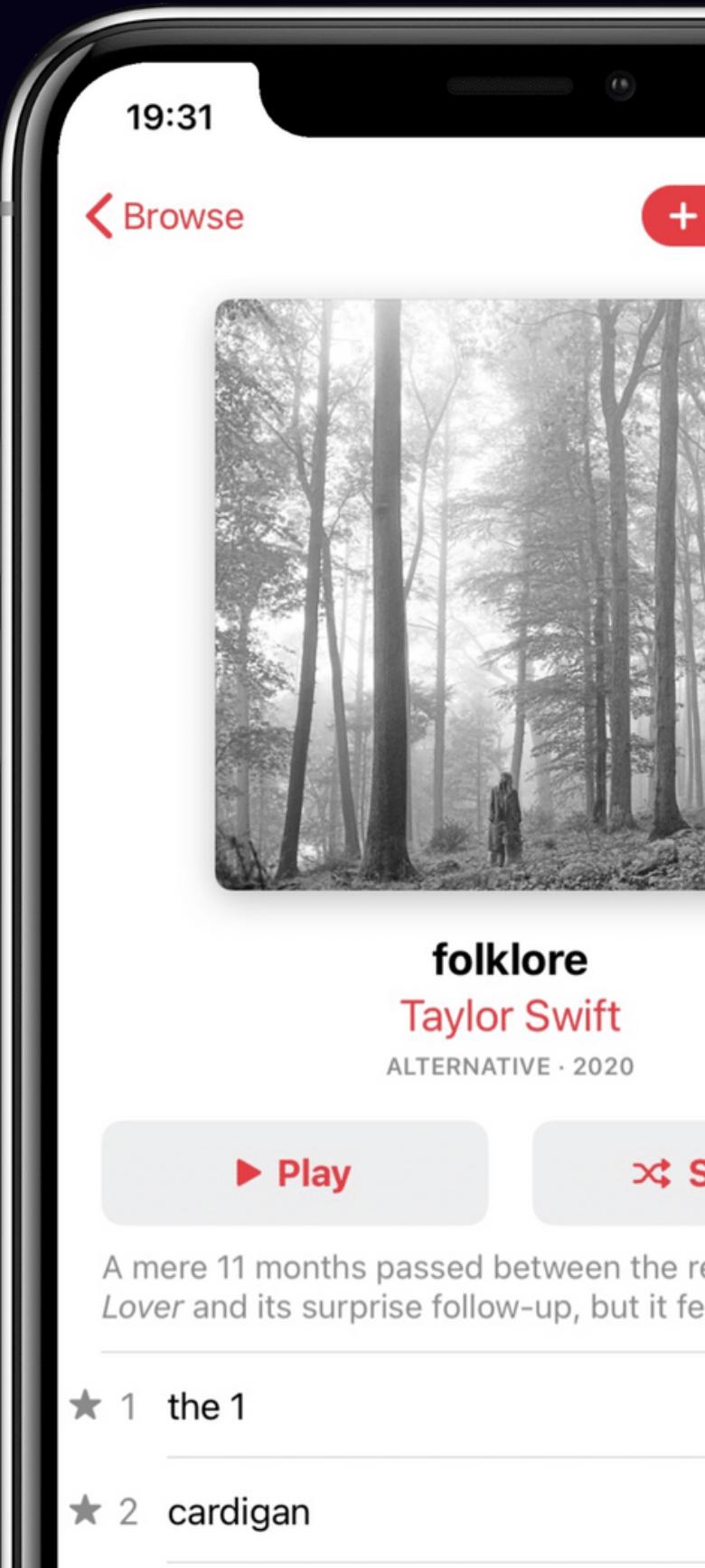
— Frammento di una grafica.



Componente dentro componente

Prendiamo il caso del pulsante Play di fianco:

- E' un componente: Sì
- Contiene al suo interno altri componenti: Sì
 - Icona Play
 - Testo "Play"



Per ricapitolare

- Una **schermata** è un insieme di frammenti di grafica.
- Un **componente** è un frammento di grafica.
- Quindi una schermata è un insieme di componenti.

Per ricapitolare

- Un **componente** può contenere dentro altri componenti.
- Quindi un **componente** può essere un **insieme di componenti**.

Per ricapitolare

- Quindi una schermata è un insieme di componenti.
- Quindi un componente può essere un insieme di componenti.

Schermata = Componente

Si. Una schermata in Flutter non è altro che:

- Un componente,
 - Con una struttura ben precisa
 - Degli attributi predefiniti
- Dentro di essa altri componenti

App = Componente

Ancora Si. Un'app in Flutter è un componente ben preciso, chiamato *MaterialApp*

App = Componente

Ancora Si. Un'app in Flutter è un componente ben preciso, chiamato *MaterialApp*, che contiene al suo interno varie schermate (di nuovo componenti)

App = Componente

Ancora Si. Un'app in Flutter è un componente ben preciso, chiamato *MaterialApp*, che contiene al suo interno varie schermate (di nuovo componenti), che contengono a loro volta altri componenti.

Esplosione di componenti!

Vediamone alcuni.

Struttura app

02

App: MaterialApp

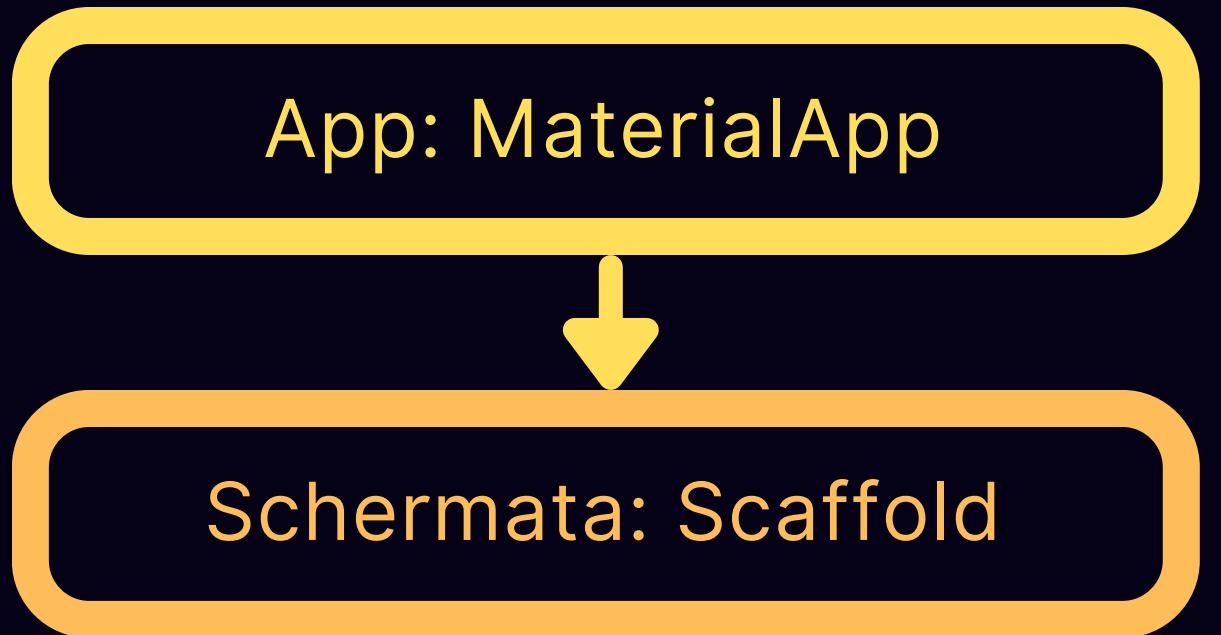
MaterialApp

La nostra app è un istanza di questo componente.

03

Scaffold

Una schermata è un'istanza di questo componente.



04

AppBar

Una schermata ha (quasi) sempre una barra del titolo.

App: MaterialApp

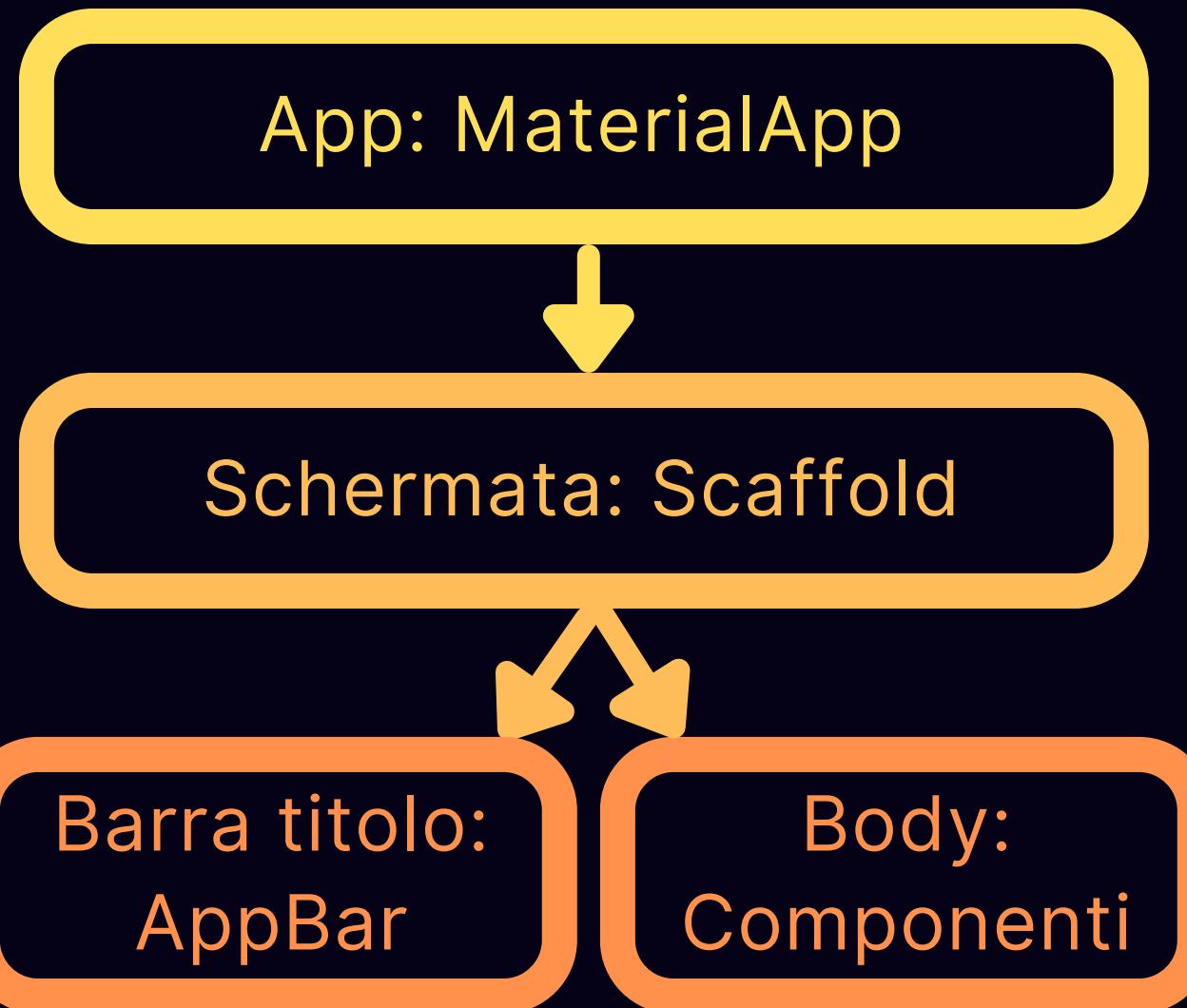
Schermata: Scaffold

Barra titolo:
AppBar

05

Body

Una schermata ha sempre un contenuto.



06

Progetto 1

Andiamo a realizzare una semplice app con questa struttura:

- MaterialApp
- Scaffold
- AppBar
 - Title
 - CenterTitle
- Text
- Center
- TextStyle