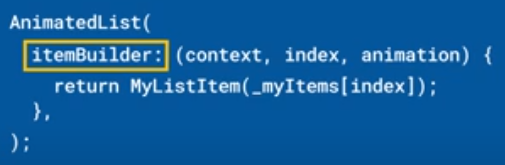
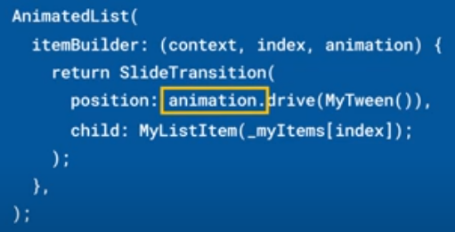
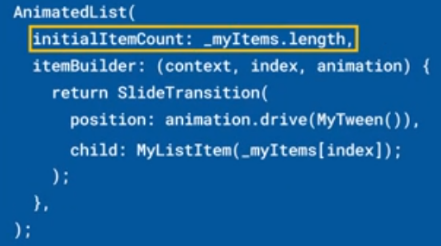
**AnimatedList**

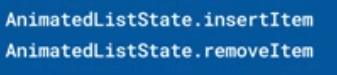
Se comienza agregando un itemBuilder con una animación:

A menos que la lista esté vacía se proporciona un initialItemCount:

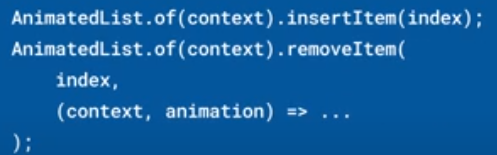


Hay que tener métodos AnimatedListState para insertar y para borrar:

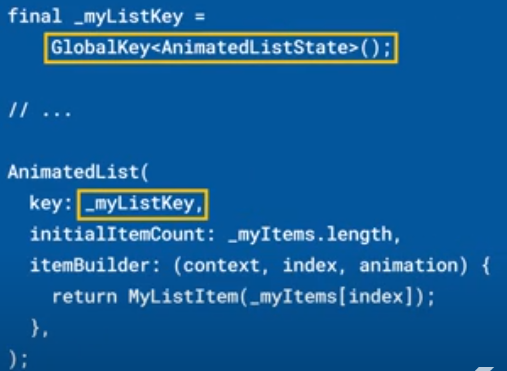


¿Cómo controlar AnimatedListState?. Hay dos métodos:

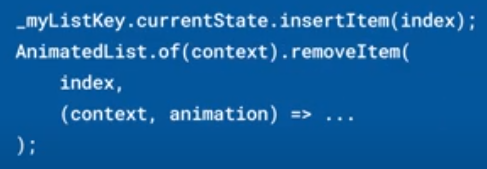
* Desde el código de la lista desde uno de los ítems se usa AnimateList.of(context):



* Desde cualquier otro lugar se usa una key

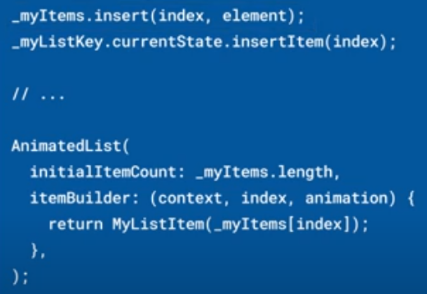


y así se puede llamar a insertItem o removeItem desde cualquier lugar



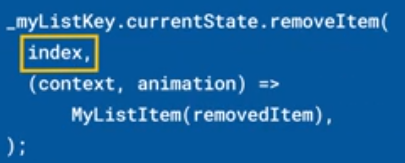
Con insertItem se indica que hay un nuevo elemento en la lista y que se debe iniciar la animación

AnimatedList usa el itemBuilder ya definido previamente

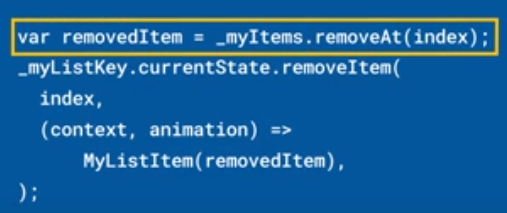


Con removeItem es similar.

Se le dice a AnimatedList que el ítem index será eliminado

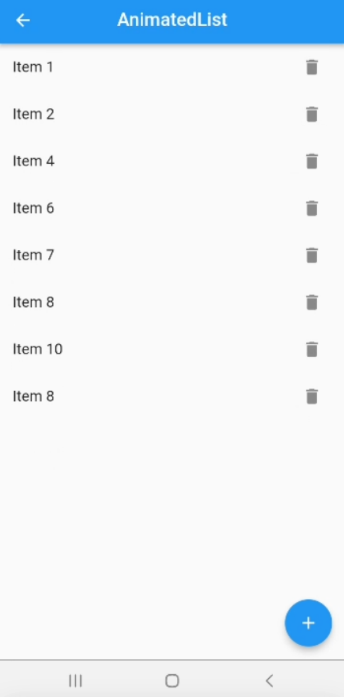


Es importante quitar el ítem de la estructura de datos subyacente:



|  |  |
| --- | --- |
| **Código** | **Comentarios** |
| import 'package:flutter/material.dart';  class AnimatedListScreen extends StatefulWidget {    const AnimatedListScreen({Key? key}) : super(key: key);    @override    State<AnimatedListScreen> createState() => \_AnimatedListScreenState();  }  class \_AnimatedListScreenState extends State<AnimatedListScreen> {    final GlobalKey<AnimatedListState> \_listKey = GlobalKey<AnimatedListState>();    final List<String> \_items = [      'Item 1',      'Item 2',      'Item 3',      'Item 4',      'Item 5',      'Item 6',      'Item 7',      'Item 8',      'Item 9',      'Item 10'    ];  //--------------------------------------------------    void \_addItem() {      final newIndex = \_items.length;      final newItem = 'Item ${newIndex + 1}';      setState(() {        \_items.add(newItem);      });      \_listKey.currentState!.insertItem(newIndex);    }  //--------------------------------------------------    void \_removeItem(int index) {      final removedItem = \_items[index];      setState(() {        \_items.removeAt(index);      });      \_listKey.currentState!.removeItem(        index,        (context, animation) => buildItem(removedItem, animation),      );    }  //--------------------------------------------------    Widget buildItem(String item, Animation<double> animation) {      return SizeTransition(        sizeFactor: animation,        child: ListTile(          title: Text(item),          trailing: IconButton(            icon: const Icon(Icons.delete),            onPressed: () =>                \_removeItem(\_items.indexOf(item)),          ),        ),      );    }  //--------------------------------------------------    @override    Widget build(BuildContext context) {      return Scaffold(        appBar: AppBar(          title: const Text('AnimatedList'),          centerTitle: true,        ),        body: AnimatedList(          key: \_listKey,          initialItemCount: \_items.length,          itemBuilder: (context, index, animation) {            return buildItem(\_items[index], animation);          },        ),        floatingActionButton: FloatingActionButton(          onPressed: \_addItem,          child: const Icon(Icons.add),        ),      );    }  } | Se define el key  Función para agregar un ítem  Función para borrar un ítem |

Así se ve:



- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -